THE CORE PAPERS

Realization Calculus

Projectors, Transfer and Arithmetic

Komorebi P.

THE CORE PAPERS

Realization Calculus

Projectors, Transfer and Arithmetic

Komorebi P.

A mathematical monograph in twelve chapters.

INTRODUCTION

Objects, operators and arithmetic

Realization Calculus studies how mathematical objects are generated from seeds, resolved into packets and related by operators. Projectors specify the components retained in a description. Transfer records their interaction and the passage from one resolution to another.

The first four chapters develop the Projector Resolved Seed Calculus. The argument then moves through multiresolution signals, evolution and memory, quantum dynamics and packet operator algorithms. The final chapters examine congruences, residual seeds, packet geometry and patching support.

Three questions recur. Which objects can be generated from the available seeds? Which relations survive a change of resolution? What additional hypotheses permit a compatible description to be realized in the intended analytic or arithmetic setting? Each construction is read with its own domain and assumptions.

The arithmetic conclusions retain their conditional character. Conjecture 66 in Chapter XI remains open. Chapter XII requires suitable residual and deformation data, patching and full support, a compatible Hecke point over the same coefficient field, classicality and an attached representation. A universal supply of these inputs is not a conclusion of the chapter.

The mathematical text and visual studies form the main sequence. The formal companions and glossary follow, with sources and references collected at the end.

READING MAP

Following the argument

The main sequence

Read Chapters I-XII in order for the development from seed constructions to conditional arithmetic realization. Each chapter opening connects its subject to the surrounding chapters and identifies the mathematical text and visual studies that follow.

Visual studies

The visual material stays with the chapter whose objects it illustrates. Figures and examples retain their stated inputs and ranges. A chapter identifier accompanies a global plate reference, as in Plate VIII.32.

Formal companions

The Lean guides are collected at the back. They identify the statements, hypotheses and source correspondences in their own formal scope. The machine-native treatment of Chapter I appears with its formal companion.

The back matter

Further studies retain their own hypotheses and open questions. The glossary gives short definitions with links to the relevant treatment. Sources and references are grouped by chapter and treatment at the end of the book.

Begin with Chapter I 1

Open the formal companions 1588

Open the glossary 2647

NOTATION AND REFERENCES

Conventions across the book

Local notation

Each chapter specifies its ambient spaces, scalar fields, domains and index sets. A symbol reused elsewhere need not denote the same object. Read every operator with its stated domain and codomain.

Finite and infinite settings

Finite algebra, limits and infinite-dimensional constructions have different assumptions. A formal identity does not by itself extend an operator to a larger space or supply continuity. The passage between settings is made where the text provides it.

Exact and numerical statements

Exact identities and numerical examples retain their stated roles. Numerical illustrations support the specified comparisons within their parameter ranges.

References within the volume

Continuous page numbers and PDF bookmarks navigate this book. Theorem, equation, figure and section numbers remain local to their named treatment unless a chapter prefix is given. Cite the chapter and treatment with the local number.

Citation labels

Bibliographies and source registers are collected at the end of the book. Their citation labels retain the numbering or short names used in the corresponding chapter and treatment. The accompanying electronic material preserves the supporting sources.

CONTENTS

Chapters I-VI

I Projector Resolved Seed Calculus I 1
Typed operator systems, seed rank, and exact packet coordinates

II Projector Resolved Seed Calculus II 98
Refinement towers, weighted transfer, and integral separation of operator packets

III Projector Resolved Seed Calculus III 177
Typed transfer, nonlinear packets, and obstruction theory

IV Projector Resolved Seed Calculus IV 279
Cyclic analytic functions, reciprocal families, and spectral zero data

V Multiresolution Signal Calculus 382
Exact coordinates, singular packets, and stable reconstruction

VI Transfer Semigroups and Memory 510
Probabilistic refinement, fractional operators, and exact hidden-state elimination

CONTENTS

Chapters VII-XII

VII Quantum Seeds and Effective Dynamics 659
Symmetry, geometric transport, quantum channels, and bounded refinement towers

VIII Packet Operator Algorithms 813
Exact compilation and GPU realization

IX Euler Congruences 930
Completion, factorial strengthening, and sharp dyadic geometry

X Universal Residual Seeds 1169
Deformation forests, transport categories, and conditional system completion

XI Tame and Wild Packet Geometry 1259
For local Langlands

XII Patching Support and Hecke Packets 1417
Conditional automorphic completion

CONTENTS

Further studies and reference

Further studies 1482

Arithmetic realization and its obstructions 1483

Visual studies of arithmetic realization 1551

Glossary 2647

Sources and references 2665

The scope of the companions

A written proof, a formal declaration and a finite computation answer different questions. The companions state their own coverage and the assumptions on which it depends.

Using the glossary

The glossary is a navigation aid. Its definitions are intentionally brief, and its references lead to the fuller statements and hypotheses in the text.

CONTENTS

Formal companions

I Projector Resolved Seed Calculus I 1588

II Projector Resolved Seed Calculus II 1721

III Projector Resolved Seed Calculus III 1768

IV Projector Resolved Seed Calculus IV 1815

V Multiresolution Signal Calculus 1936

VI Transfer Semigroups and Memory 1992

VII Quantum Seeds and Effective Dynamics 2054

VIII Packet Operator Algorithms 2132

IX Euler Congruences 2220

X Universal Residual Seeds 2350

XI Tame and Wild Packet Geometry 2384

XII Patching Support and Hecke Packets 2617

RIGHTS AND ATTRIBUTION

Use and attribution

Original material

Original writing, figures and editable writing sources are licensed under Creative Commons Attribution 4.0 International. Original software and machine-readable material are licensed under the MIT License. The accompanying files contain the full license texts.

Third-party material

Third-party sources, fonts and other incorporated material retain their own terms. Their notices accompany the collection and apply alongside the licenses for original material.

Attribution

Komorebi P., Realization Calculus: Projectors, Transfer and Arithmetic. When citing an individual result, include its chapter, named treatment and local result number.

CHAPTER I

Projector Resolved Seed Calculus I

Typed operator systems, seed rank, and exact packet coordinates

The opening chapter introduces the seed and operator language used throughout the book.

Seeds, operators and projectors provide the linear language of the series. Exact packet coordinates make the generated objects and the maps between them visible.

IN THIS CHAPTER

Mathematical text 2

Formal companion 1588

Sources and references 2666

Scope

The correspondence concerns the stated proof-level constructions. Noncomputable existence results do not provide executable solvers.

The next chapter compares these descriptions across refinement levels and asks which structures survive that passage.

Overview

This chapter begins with a practical mathematical question. Once a family of operators has been fixed, what can we generate from a few starting vectors, and how can we keep track of the resulting object? The answer has two parts. Seed rank measures how many starting vectors are needed. Packet coordinates describe the components of vectors and the transfers between them.

The construction starts with a typed operator grammar and a concrete realization of that grammar. From there we study generated hulls, presentations by seeds and relations, and the number of generators required by a module. The generator results include reduction modulo the Jacobson radical, the semisimple multiplicity formula over division rings, and the formula for a single operator over an arbitrary field.

We then resolve vectors and operators using complete families of idempotents. Finite packet reconstruction is exact. Countable reconstruction requires the stated convergence assumptions. Transfer blocks record the action from each input packet to each output packet, and their multiplication exposes both admissible paths and cancellation. Later sections develop quotient descent, coordinate changes, completion, and transport through realization functors.

Worked quiver, coordinate ring, matrix, graph, and shift models show how the same language is used in different settings. The visual atlas follows these models through complete calculations. Stable statement identifiers connect this text to the existing machine companion, while the separate Lean guide specifies the scope of the formal certificates. The written proofs and the imported classical results retain their stated hypotheses.

Reader's orientation

Status class	Meaning
Definition	Introduced structure or notation.
Theorem, proposition, corollary, lemma	Proved in this manuscript under the printed hypotheses.
Classical input	Imported standard theorem, stated explicitly because later proofs depend on it.
Boundary	A limitation, counterexample, or failure of an implication that is part of the formal scope.

Formal and computational companions. The formal companion explains the Lean development, its build procedure, theorem mapping, and the hypotheses of each verified statement. It distinguishes the 35-row foundational contract from the correspondence for the 108-record mathematical inventory. The computational companion presents those stable records, dependencies, and executable interfaces. Both companions are collected in the back matter. The mathematics below retains the full scope of its stated classical inputs and written proofs.

A route through this section

The argument has four stages. Reading them in order makes the later examples easier to follow.

1. **Grammar and realization.** Specify which compositions of operators are legal, then distinguish that abstract syntax from the operators that actually act on a chosen space.
2. **Generation.** Choose one or more seeds and ask which part of the realization is generated by their operator orbits. The minimal number of required seeds is the seed rank.
3. **Packet coordinates.** Resolve the space by a complete family of idempotents. Every vector and every operator then has exact packet coordinates.
4. **Transport.** Study how those coordinates behave under products, graphs, morphisms, quotients, completion, and realization functors.

The first two layers describe what can be generated. The last two describe how the generated object is resolved and transported.

The basic conceptual chain

$$\begin{aligned} &\text{typed grammar} \longrightarrow \text{realized algebra} \longrightarrow \text{seed hull} \\ &\longrightarrow \text{seed rank} \longrightarrow \text{packet coordinates} \\ &\longrightarrow \text{transfer blocks} \longrightarrow \text{functorial realization.} \end{aligned}$$

Each object in this chain has a different role. A grammar is a presentation of allowed operator words. A realized algebra is the image of that grammar in an endomorphism algebra. A seed hull is an orbit span inside one particular module. A packet system is an exact coordinate decomposition and need not be orthogonal. A realization functor is required to preserve a declared mathematical profile, not merely the vocabulary.

Four questions answered by Chapter I

Question	Formal object	Principal answer
What operations are legal?	Quiver with relations or a unital operator algebra	Typed realizations are modules over the quotient path algebra.
What can a finite seed family generate?	Evaluation map $A^{\oplus r} \to M$	The generated hull is a quotient of a finite free module.
How many seeds are required?	$\operatorname{srank}_A(M)$	For a finitely generated module, pass to $M/\operatorname{Jac}(A)M$. When the quotient algebra is semisimple Artinian, compute its simple multiplicities.
How are vectors and operators resolved?	Complete idempotents (P_α)	$X \cong \bigoplus_\alpha P_\alpha X$ and $T \leftrightarrow (P_\beta T P_\alpha)$.

Notation map

Symbol	Name	Meaning
$\mathfrak{G} = (Q, I)$	Typed grammar	A quiver with finite vertex set and a two-sided ideal of source and target compatible relations.
$\Lambda_{\mathfrak{G}}$	Grammar algebra	The quotient path algebra $\Bbbk Q/I$.
$\mathcal{A}_X$	Realized algebra	The image of the grammar algebra in $\operatorname{End}_\Bbbk(\lvert X\rvert)$.
$S \subseteq M$	Seed family	The vectors from which the operator orbit is generated.
$\operatorname{Hull}_A(S)$	Generated hull	The submodule AS, or its declared topological closure.
$\operatorname{srank}_A(M)$	Seed rank	The smallest cardinality of a generating family. A finite value is the minimum number of seeds.
$\mathcal{P} = (P_\alpha)$	Packet system	Pairwise annihilating idempotents summing to the identity.
X_α	Packet space	The range $P_\alpha X$.
$\mathcal{A}_\mathcal{P}$	Analysis	The map $x \mapsto (P_\alpha x)_\alpha$.
$\mathcal{S}_\mathcal{P}$	Synthesis	The map $(x_\alpha) \mapsto \sum_\alpha x_\alpha$.
$T_{\beta\alpha}$	Transfer block	The source and target block $P_\beta T P_\alpha : X_\alpha \to X_\beta$.
$\operatorname{Supp}_\mathcal{P}(x)$	Packet support	The packet indices on which x has a nonzero coordinate.
psrank	Rank for seeds confined to individual packets	The minimal number of generating seeds each supported in a single packet.

Small examples to keep in view

The following small systems recur throughout the explanatory boxes. The worked models and visual atlas later develop these ideas in complete calculations.

R1: one operator with several primary blocks

Let T act on a finite dimensional vector space. Regard X as a $\Bbbk[t]$-module by $t \cdot x = Tx$. If T has three Jordan blocks for the same eigenvalue, then three seeds are necessary and sufficient for that primary component. The largest number of blocks over any irreducible polynomial is the seed rank.

R2: a matrix algebra with multiplicity

Let $A = M_2(\Bbbk)$ act diagonally on $M = (\Bbbk^2)^{\oplus 3}$. One seed can contribute at most two independent multiplicity directions, because the regular module contains two copies of the simple module. Hence

$$\operatorname{srank}_A(M) = \left\lceil \frac{3}{2} \right\rceil = 2.$$

This is the simplest nontrivial instance of the semisimple multiplicity formula.

R3: an exact but oblique packet split

On $X = \Bbbk^2$, set

$$P_1 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}, \qquad P_2 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Then $P_i^2 = P_i$, $P_1P_2 = P_2P_1 = 0$, and $P_1 + P_2 = I$. The ranges form a direct sum, but over $\mathbb{R}$ the projectors are not orthogonal in the Euclidean inner product. Exact reconstruction and numerical conditioning are therefore separate questions.

R4: a directed packet system

Let $X = X_1 \oplus X_2 \oplus X_3$. Suppose the generating operators permit transfers $1 \to 2$ and $2 \to 3$, with no path entering packet 1. Packet 1 is the unique source component. If the corner action on X_1 is cyclic and the propagation hypotheses of theorem 9.7 hold, one seed in X_1 can generate all three packets.

Suggested reading paths

- For the minimal algebraic foundation, read Sections 1 to 5, then Sections 7 and 8.
- For finite dimensional representation theory, concentrate on Sections 4 to 6.
- For signal, wavelet, or quantum applications, read Sections 7 and 8 and the results on tensor products carefully.
- For graphs and applications involving residual transport, read Sections 8 to 10.
- For later categorical claims, read Sections 10 and 11 together with the final semantic realization criterion.

Scope of the argument

The chapter does not claim that every useful decomposition is automatically discovered, that every packet split is numerically stable, or that exact coordinates solve a nonlinear problem. Chapter I establishes the linear language and its exact transport laws. Refinement, obstruction theory, nonlinear trees, arithmetic valuations, and continuum limits belong to later chapters.

1 Coefficient conventions, logical status, and scope

A route through this section

This section sets out the conventions used throughout the chapter. It explains when a statement concerns algebra, when topology enters the argument, which results are taken as classical input, and how the stable record IDs connect this chapter to later work. Keeping these settings distinct makes the hypotheses of each result easier to follow.

1.1 Ambient regimes and theorem records

Definition 1.1 (Coefficient and module regimes)**.** All modules are left modules unless stated otherwise. The algebraic regime consists of a unital ring A and a unital left module M over A. When linear dimensions, tensor products, quivers, or operator blocks are used, A is a unital algebra over $\Bbbk$, where $\Bbbk$ is a field, and M is a vector space over $\Bbbk$. The topological regime consists of a Hausdorff topological vector space over a complete valued field, together with a unital algebra of continuous linear operators. Every infinite sum, completion, quotient topology, and closure below is interpreted only in the regime explicitly declared in its statement.

Conventions for empty families. For finite families of nonnegative integers we use $\max \emptyset = 0$. The height of an empty finite partially ordered set is 0, as is the height of a nonempty poset with no strict comparable pair. Thus the height in the theorem for the support preorder counts strict steps.

Definition 1.2 (Canonical theorem record)**.** A canonical theorem record is the tuple

$$(\mathrm{ID}, \mathrm{status}, \mathrm{hypotheses}, \mathrm{statement}, \mathrm{dependencies}, \mathrm{proof}, \mathrm{exports}, \mathrm{boundary}).$$

A later chapter cites the stable record ID. A domain specialization may add domain hypotheses, but it does not repeat a generic proof from this chapter.

Boundary 1.3 (Symbolic scope). This chapter proves the algebraic and topological statements printed below. It does not assert historical novelty, independent external certification, automatic discovery of useful packet systems, nonlinear obstruction theory, multilevel refinement, arithmetic congruences, quantum gravity, automorphy, or local or global Langlands reciprocity.

2 Typed operator grammars, realizations, and seed systems

A route through this section

An operator grammar describes which compositions are allowed and which relations they satisfy. A realization gives this syntax a concrete action on vector spaces. Its realized algebra consists of the operators that actually occur in that action. We can then ask which vectors serve as seeds.

2.1 Typed syntax and module realizations

Definition 2.1 (Typed operator grammar)**.** Let $Q = (Q_0, Q_1, s, t)$ be a quiver with finite vertex set. A typed operator grammar over $\Bbbk$ is a pair

$$\mathfrak{G} = (Q, I),$$

where $I \lhd \Bbbk Q$ is an ideal on both sides generated by linear combinations over $\Bbbk$ of paths having common source and common target. Write

$$\Lambda_{\mathfrak{G}} := \Bbbk Q / I.$$

A realization of $\mathfrak{G}$ is a family $(X_v)_{v \in Q_0}$ and linear maps

$$X_a : X_{s(a)} \longrightarrow X_{t(a)} \qquad (a \in Q_1)$$

that annihilate every relation in I. A morphism $f : X \to Y$ of realizations is a family of linear maps $f_v : X_v \to Y_v$ satisfying

$$f_{t(a)} X_a = Y_a f_{s(a)} \qquad (a \in Q_1).$$

Definition 2.2 (Realization on a single space)**.** For a typed realization X, define its realization on a single space by

$$|X| := \bigoplus_{v \in Q_0} X_v.$$

Every vertex idempotent acts as a coordinate projector on $|X|$, and every path acts as a block operator from its source space to its target space.

Theorem 2.3 (Typed realization and module equivalence)**.** *The category of realizations of $\mathfrak{G}$ is equivalent to the category of unital left modules over $\Lambda_{\mathfrak{G}}$. Explicitly, a realization X determines the collapsed space $|X|$ of definition 2.2 with the vertex idempotent e_v acting as the coordinate projector onto X_v and each arrow acting in the block between its declared source and target. Conversely, a unital module M over $\Lambda_{\mathfrak{G}}$ determines the realization*

$$M_v = e_v M, \qquad M_a(x) = ax.$$

The two constructions are inverse up to canonical natural isomorphism.

Proof. Because Q_0 is finite, $1 = \sum_{v \in Q_0} e_v$ in $\Bbbk Q$, the e_v are pairwise orthogonal, and the same identities hold in $\Lambda_{\mathfrak{G}}$. Given a realization, define the action of a path by composition of its arrow maps and extend linearly. Every element of I acts as zero, so the action factors through $\Lambda_{\mathfrak{G}}$. The vertex idempotents give

$$|X| = \bigoplus_v e_v |X| = \bigoplus_v X_v.$$

Conversely, if M is unital, then

$$M = 1M = \sum_v e_v M.$$

Orthogonality makes the sum direct. For an arrow $a : v \to w$, the identities $a = e_w a e_v$ imply

$$a(e_v M) \subseteq e_w M.$$

The relations in I act as zero because M is a module over $\Lambda_{\mathfrak{G}}$. Applying both constructions returns the original vertex spaces and arrow maps, or the original module decomposition, through the evident identity maps. These identifications commute with morphisms, giving an equivalence of categories. □

How to read the result

The equivalence does not claim that a quiver presentation is unique. It says that once a grammar is fixed, working with typed arrows and working with a module over the quotient path algebra carry the same information. The typed description is often easier for constructing an operator system. The module description is often easier for proving generation and quotient theorems.

A grammar with two vertices

Take two vertices and one arrow $a : 1 \to 2$. A realization is a pair of spaces and a linear map

$$X_1 \xrightarrow{X_a} X_2.$$

After collapse, $|X| = X_1 \oplus X_2$, the vertex idempotents are the two coordinate projectors, and a is a single block outside the diagonal. This elementary case already contains the viewpoint of packets and transfer blocks used later.

2.2 The realized action and faithful reduction

Definition 2.4 (Realized operator algebra)**.** For a realization X, the realized operator algebra is

$$\mathcal{A}_X := \mathrm{im}(\Lambda_{\mathfrak{G}} \longrightarrow \mathrm{End}_{\Bbbk}(|X|)) .$$

It records the actual operators in the realization and may be a proper quotient of the abstract grammar algebra.

Proposition 2.5 (Action factorization and hull invariance)**.** *Let $S \subseteq |X|$. The generated submodule under the grammar algebra equals the generated submodule under the realized operator algebra:*

$$\Lambda_{\mathfrak{G}} S = \mathcal{A}_X S.$$

Consequently, every statement about seed rank depends only on the realized action, not on redundant relations in a chosen presentation of the grammar.

Proof. The action map $\rho : \Lambda_{\mathfrak{G}} \twoheadrightarrow \mathcal{A}_X$ is surjective by definition and satisfies $\lambda s = \rho(\lambda) s$. Hence every grammar orbit vector lies in $\mathcal{A}_X S$. The reverse inclusion holds because every element of $\mathcal{A}_X$ is $\rho(\lambda)$ for some λ. □

The central idea

Replacing the abstract grammar algebra by its image $\mathcal{A}_X$ removes syntax that acts as zero. This is a faithful reduction of the action. It does not change any orbit, hull, or computation of seed rank.

2.3 Seeds, addressed hulls, and natural presentations

Definition 2.6 (Linear seed system)**.** A linear seed system is a tuple

$$\mathfrak{S} = (X, \mathcal{A}, S, \mathcal{P}, \tau),$$

where X is a unital left module over $\mathcal{A}$, $\mathcal{A}$ is fixed independently of the target element, $S = (s_1, \dots, s_r)$ is a finite distinguished seed family, $\mathcal{P}$ is either empty or a declared packet system, and τ is either the algebraic convention or a specified Hausdorff topology in which the declared operator action is continuous.

Definition 2.7 (Generated hull)**.** For $S \subseteq X$, define

$$\mathrm{Hull}_{\mathcal{A}}(S) := \sum_{s \in S} \mathcal{A}s.$$

In the topological regime define

$$\mathrm{Hull}_{\mathcal{A},\tau}(S) := \overline{\mathrm{Hull}_{\mathcal{A}}(S)}^{\tau}.$$

For a singleton $S = \{s\}$, write $\mathrm{Hull}_{\mathcal{A}}(s)$.

The algebraic hull uses finite sums of addressed seed vectors. In the topological setting, taking its closure may add limits that no finite expression reaches. Thus algebraic generation and dense generation answer different questions, even when they use the same seeds and operators.

Definition 2.8 (Natural presentation)**.** A seed presentation is natural relative to a declared grammar when the primitive operators and relations are fixed independently of the target, the relation system is finite or effectively enumerable, the topology is intrinsic to the realization, no primitive operator directly stores the complete target, and the presentation exposes a reusable invariant, theorem, or computation not encoded by a primitive chosen for the target.

Proposition 2.9 (Algebra generation and module generation are distinct)**.** *Every nonzero unital algebra A, regarded as a left module over itself, has*

$$\mathrm{srank}_A(A) = 1,$$

with cyclic seed 1_A. This conclusion is independent of the number of algebra generators required to present A as an algebra.

Proof. For every $a \in A$, one has $a = a1_A$. Hence $A1_A = A$. The number of operations needed to generate A as an algebra is a different invariant and does not enter the module calculation. □

Boundary 2.10 (Operators chosen for a target). For a fixed seed s and target x, the existence of a specially defined operator A_x satisfying $A_x s = x$ does not establish a natural seed calculus unless A_x belongs to a grammar that is fixed independently of the target and satisfies definition 2.8.

Scope of the argument

An operator may be chosen because it sends the current seed to a desired target. Such a choice proves only an existence statement for that one target unless the operator already belongs to the fixed algebra. The algebra must be declared before the question of seed rank is asked. Otherwise the acting system changes with the answer one wants to obtain.

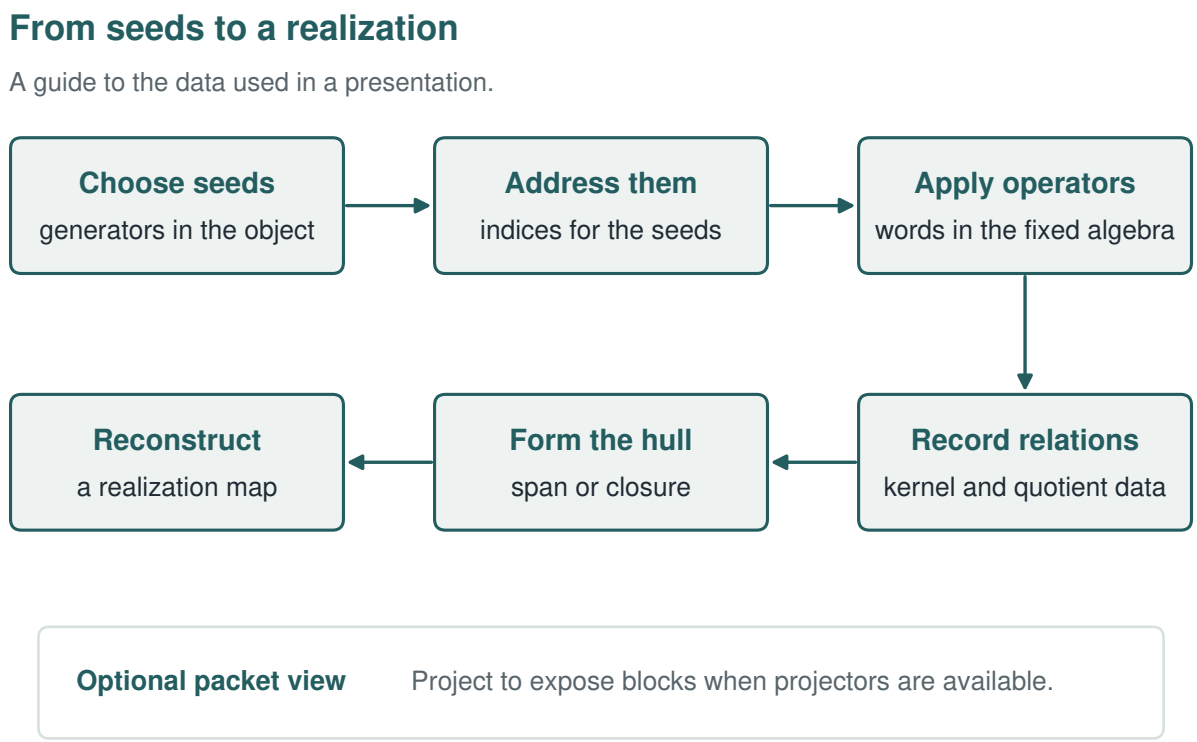


Figure 1. Schematic organization of the seed, operator, relation, and reconstruction data. This is a guide to the presentation, not an additional formal theorem.

3 Free presentations by seeds and relations

A route through this section

A seed family gives a canonical map from a finite free module. The generated hull is exactly its image, and the relations among addressed seeds form the kernel. This converts orbit generation into a presentation problem.

3.1 Finite free covers

Definition 3.1 (Seed evaluation map)**.** Let A be a unital ring, let M be a left module over A, and let $S = (s_1, \dots, s_r) \in M^r$. Define the linear map over A

$$\varepsilon_S : A^{\oplus r} \longrightarrow M, \qquad (a_1, \dots, a_r) \longmapsto \sum_{j=1}^{r} a_j s_j.$$

Theorem 3.2 (Finite free presentation)**.** *For the map in definition 3.1,*

$$\operatorname{im} \varepsilon_S = \operatorname{Hull}_A(S), \qquad N_S := \operatorname{Ker} \varepsilon_S \leq A^{\oplus r},$$

and the induced map is an isomorphism of left modules over A

$$\boxed{A^{\oplus r}/N_S \cong \operatorname{Hull}_A(S).}$$

Consequently, S generates M if and only if ε_S is surjective.

Proof. The image consists exactly of finite sums $\sum_j a_j s_j$, which is $\operatorname{Hull}_A(S)$. Because ε_S is linear over A, its kernel is a left submodule. The first isomorphism theorem gives

$$A^{\oplus r}/N_S \cong \operatorname{im} \varepsilon_S = \operatorname{Hull}_A(S).$$

The final equivalence is the equality $\operatorname{im} \varepsilon_S = M$. □

How to read the result

The theorem converts the geometric language of addressed seed orbits into the algebraic language of generators and relations. A word in the operators is represented by an element of the free module. Two addressed expressions represent the same vector exactly when their difference lies in the kernel. The generated hull is therefore not merely spanned by the free module. It is its canonical quotient.

A cyclic nilpotent chain

Let $A = \Bbbk[t]$ and $M = \Bbbk[t]/(t^3)$. The residue class $s = [1]$ is one seed and

$$\varepsilon_s : A \to M, \qquad f(t) \mapsto f(t)[1]$$

has kernel (t^3). Hence $M \cong A/(t^3)$, and the addressed vectors $s, ts, t^2 s$ are the visible orbit chain.

Corollary 3.3 (Characterization by minimal free covers)**.** *For every finitely generated left module M over A,*

$$\boxed{\operatorname{srank}_A(M) = \min\{r \geq 0 : \textit{there exists a surjection } A^{\oplus r} \twoheadrightarrow M\}.}$$

Proof. A seed family of size r gives a surjection by theorem 3.2. Conversely, the images of the standard basis vectors under any surjection $A^{\oplus r} \twoheadrightarrow M$ form a generating family of size at most r. □

Corollary 3.4 (Presentation with one seed). *For $s \in M$, the annihilator*

$$\mathrm{Ann}_A(s) := \{a \in A : as = 0\}$$

is a left ideal and

$$\boxed{A/\mathrm{Ann}_A(s) \cong \mathrm{Hull}_A(s).}$$

Proof. The evaluation map for one seed is $a \mapsto as$. Its kernel is $\mathrm{Ann}_A(s)$, which is therefore a left ideal, and theorem 3.2 gives the quotient isomorphism. □

3.2 One seed and the distinction between module and algebra relations

Proposition 3.5 (Criterion for the annihilator to be an ideal on both sides). *For $s \in M$, the left ideal $L = \mathrm{Ann}_A(s)$ is an ideal on both sides if and only if*

$$L\ \mathrm{Hull}_A(s) = 0.$$

If s is cyclic, these conditions are equivalent to

$$\mathrm{Ann}_A(s) = \mathrm{Ann}_A(M).$$

Proof. The left ideal L is an ideal on both sides exactly when $ab \in L$ for every $a \in L$ and $b \in A$. This is equivalent to

$$0 = (ab)s = a(bs) \qquad (a \in L,\ b \in A).$$

By additivity, the last identity is equivalent to $L\,\mathrm{Hull}_A(s) = 0$.

If s is cyclic, then $\mathrm{Hull}_A(s) = M$. Hence the preceding condition is $L \subseteq \mathrm{Ann}_A(M)$. The reverse inclusion always holds because an operator annihilating all of M annihilates s. Thus equality follows. Conversely, $\mathrm{Ann}_A(M)$ is an ideal on both sides, so equality implies that L is an ideal on both sides. □

Why a seed annihilator need not be an ideal on both sides

Let $A = M_2(\Bbbk)$ act on $M = \Bbbk^2$, and let $s = e_1$. Then $\mathrm{Ann}_A(s)$ consists of matrices whose first column is zero. It is a left ideal because left multiplication preserves a zero first column. It is not generally a right ideal. Thus the natural quotient for one seed is a quotient of the left regular module, not automatically a quotient algebra.

3.3 Topological completion of a presentation

Theorem 3.6 (Completion of a dense free presentation). *Let X be a complete Hausdorff locally convex space, let $\mathcal{A} \subseteq \mathrm{End}(X)$ act continuously, and let $S = (s_1, \dots, s_r)$. Put*

$$E := \mathcal{A}^{\oplus r}/N_S$$

algebraically and let $j : E \to X$ be the injective map induced by ε_S. Give E the locally convex topology transported from $j(E)$, equivalently the topology generated by

$$\tilde{p}(e) := p(j(e))$$

for the continuous seminorms p on X. Every $a \in \mathcal{A}$ acts continuously on E in the transported topology and therefore extends to $\widehat{E}$. If $\mathrm{Hull}_{\mathcal{A}}(S)$ is dense in X, then j extends uniquely to a topological isomorphism of modules over $\mathcal{A}$

$$\boxed{\widehat{E} \cong X.}$$

Proof. By theorem 3.2, j is an algebraic isomorphism of modules over $\mathcal{A}$ from E onto $\mathrm{Hull}_{\mathcal{A}}(S)$. The transported seminorms make it a topological isomorphism onto that subspace. For $a \in \mathcal{A}$ and a continuous seminorm p on X, continuity of $a : X \to X$ supplies continuous seminorms controlling $p(aj(e))$, so the induced action on E is continuous and extends uniquely to $\widehat{E}$. The subspace is dense by hypothesis. A Cauchy net in E is mapped to a Cauchy net in X, which converges because X is complete. This defines a continuous extension $\hat{j} : \widehat{E} \to X$. For any $x \in X$, choose a net (e_λ) in E with $j(e_\lambda) \to x$. The transported topology makes (e_λ) Cauchy, so it converges to some $\xi \in \widehat{E}$, and continuity gives $\hat{j}(\xi) = x$. Thus $\hat{j}$ is surjective. If $\hat{j}(\xi) = 0$, represent ξ by a Cauchy net (e_λ). Then $j(e_\lambda) \to 0$, so every defining seminorm $\tilde{p}(e_\lambda)$ tends to zero. Thus $e_\lambda \to 0$ in the completion and $\xi = 0$. Hence $\hat{j}$ is bijective. Its inverse is continuous by the universal property of completion, or directly because the topology on E was transported from X. □

The topology in this completion theorem comes from the realized image. It is not an independently chosen topology on the free algebra. This choice ensures that the completion recovers the given space, while leaving separate the question of whether another natural topology on the grammar produces the same completion.

What we have established

A seed family always gives a free presentation of its hull. The kernel stores all linear relations among addressed seed words. One seed gives a cyclic module, but only special annihilators permit an algebra quotient. Dense generation is handled by completing the presented image rather than by silently replacing algebraic span with closure.

4 Seed rank and functorial calculus

A route through this section

For a finitely generated module, seed rank is the smallest number of seeds needed to generate it. We first examine how this number behaves when the acting algebra changes or the module is replaced by a quotient. We then give bounds for exact sequences, direct sums, and tensor products. Quotients with simple actions provide useful witnesses for lower bounds.

4.1 Definition, cyclicity, and monotonicity

Definition 4.1 (Seed rank)**.** For a unital left module M over A, define

$$\operatorname{srank}_A(M) := \min\{|S| : AS = M\}.$$

For a topological module (X, τ), define

$$\operatorname{srank}_{\mathcal{A},\tau}(X) := \min\{|S| : \overline{\mathcal{A}S}^{\tau} = X\}.$$

Set $\operatorname{srank}_A(0) = 0$. When the value is finite, it is the minimal number of module generators.

Proposition 4.2 (Cyclic characterization)**.** *For $M \neq 0$,*

$$\operatorname{srank}_A(M) = 1 \quad \Longleftrightarrow \quad M \text{ contains a cyclic vector}.$$

Proof. Both conditions say that there is $m \in M$ with $Am = M$. □

Proposition 4.3 (Monotonicity under enlargement of the acting algebra)**.** *Let $B \subseteq A \subseteq \operatorname{End}_{\Bbbk}(X)$ be unital operator algebras with the same identity. Then*

$$\boxed{\operatorname{srank}_A(X) \leq \operatorname{srank}_B(X).}$$

The same statement holds for topological seed rank when both algebras act continuously.

Proof. For every $S \subseteq X$, one has $BS \subseteq AS$. Hence every generating family over B is also a generating family over A. Taking closures gives the topological statement. □

How to read the result

Enlarging the acting algebra can only create more orbit directions, so it can only lower the number of seeds required. This is why every statement about seed rank must name the acting algebra. The same vector space may be cyclic for a large algebra and require many seeds for a smaller one.

4.2 Quotients, isomorphisms, and exact sequences

Theorem 4.4 (Quotient monotonicity)**.** *Let $\varphi : A \to B$ be a unital ring homomorphism and let*

$$\Phi : M \twoheadrightarrow N$$

be a surjective additive map satisfying

$$\Phi(am) = \varphi(a)\Phi(m) \qquad (a \in A,\ m \in M).$$

Then

$$\boxed{\operatorname{srank}_B(N) \leq \operatorname{srank}_A(M).}$$

If M and N are topological modules, Φ is continuous and surjective, and the actions are continuous, then

$$\operatorname{srank}_{B,\tau_N}(N) \leq \operatorname{srank}_{A,\tau_M}(M).$$

Proof. Let S generate M. For every $n \in N$, choose $m \in M$ with $\Phi(m) = n$ and write $m = \sum_i a_i s_i$. Then

$$n = \sum_i \varphi(a_i)\Phi(s_i),$$

so $\Phi(S)$ generates N.

For the topological statement, let S generate M densely. Given $n = \Phi(m)$, choose a net $m_\lambda \in AS$ converging to m. Continuity gives

$$\Phi(m_\lambda) \longrightarrow n,$$

and every $\Phi(m_\lambda)$ lies in $B\Phi(S)$. Thus $B\Phi(S)$ is dense in N. □

A quotient can strictly lower seed rank

Let $A = \Bbbk$ act by scalars on $M = \Bbbk^3$. Then $\operatorname{srank}_A(M) = 3$. Quotienting by a subspace of dimension one gives a scalar module of dimension two, whose seed rank is 2. Quotient monotonicity is therefore often strict.

Corollary 4.5 (Isomorphism invariance)**.** *An intertwining module isomorphism preserves algebraic seed rank. A topological intertwining isomorphism preserves topological seed rank.*

Proof. Apply theorem 4.4 to the isomorphism and its inverse. □

Theorem 4.6 (Short exact sequence bounds)**.** *Let*

$$0 \longrightarrow L \xrightarrow{\iota} M \xrightarrow{\pi} N \longrightarrow 0$$

be a short exact sequence of left modules over A. Then

$$\boxed{\operatorname{srank}_A(N) \leq \operatorname{srank}_A(M) \leq \operatorname{srank}_A(L) + \operatorname{srank}_A(N).}$$

Proof. The lower inequality is quotient monotonicity. For the upper inequality, choose generating families $(\ell_i)_{i\in I}$ of L and $(n_j)_{j\in J}$ of N with the least possible cardinalities. Choose a lift $m_j \in M$ of each n_j. For any $m \in M$, a finite subset $F \subseteq J$ and coefficients $a_j \in A$ give

$$\pi(m) = \sum_{j\in F} a_j n_j.$$

The vector $m - \sum_{j\in F} a_j m_j$ belongs to $\iota(L)$, so it is a finite linear combination of the $\iota(\ell_i)$. The union of the lifted family and the image of the generating family of L therefore generates M. Its cardinality is at most $|I| + |J|$, giving the upper bound. This argument uses finite sums for each vector even when either generating family is infinite. □

Why the hypotheses matter

For a short exact sequence, a generating family for M always maps to a generating family for the quotient. It need not generate the submodule L. For the upper bound, choose generators of L separately and combine them with lifts of quotient generators.

4.3 Direct sums under shared and independent actions

Proposition 4.7 (Direct sum bounds for a shared action)**.** *If one algebra A acts diagonally on $M \oplus N$, then*

$$\boxed{\max\{\operatorname{srank}_A(M), \operatorname{srank}_A(N)\} \le \operatorname{srank}_A(M \oplus N) \le \operatorname{srank}_A(M) + \operatorname{srank}_A(N).}$$

Proof. The coordinate projections are intertwining quotients and give the lower bound. For the upper bound, embed minimal generating families of M as $(m_i, 0)$ and of N as $(0, n_j)$. Their union generates by finite sums and has cardinality at most the sum of the two seed ranks. □

Theorem 4.8 (Direct sum formula for independent actions)**.** *Let M be a left module over A and N a left module over B. Under the action of the product algebra*

$$(a, b)(m, n) = (am, bn),$$

one has

$$\boxed{\operatorname{srank}_{A\times B}(M \oplus N) = \max\{\operatorname{srank}_A(M), \operatorname{srank}_B(N)\}.}$$

Proof. The coordinate projections give the lower bound. Let r be the maximum on the right, regarded as a cardinal, and choose an index set I of cardinality r. Choose minimal generating families for M and N, index them by subsets of I, and fill any unused positions with zeros. Write the resulting families as $(m_i)_{i\in I}$ and $(n_i)_{i\in I}$. For any $m \in M$ and $n \in N$, their finite expressions involve a common finite subset $F \subseteq I$ after zero coefficients have been added. Thus

$$m = \sum_{i\in F} a_i m_i, \qquad n = \sum_{i\in F} b_i n_i,$$

and

$$(m, n) = \sum_{i\in F} (a_i, b_i)(m_i, n_i).$$

The family of paired vectors generates and has cardinality at most r. The empty family handles the case in which both modules are zero. □

4.4 Tensor generation and witnesses for lower bounds

Proposition 4.9 (Tensor product bound)**.** *Let A and B be unital algebras over $\Bbbk$, let M be a left module over A, and let N be a left module over B. Under the natural $A \otimes_\Bbbk B$ action,*

$$\boxed{\operatorname{srank}_{A\otimes B}(M \otimes_\Bbbk N) \le \operatorname{srank}_A(M) \operatorname{srank}_B(N).}$$

Proof. Choose minimal generating families $(m_i)_{i\in I}$ for M and $(n_j)_{j\in J}$ for N. Each vector in either module is a finite linear combination of its generators. Bilinearity therefore expresses every simple tensor as a finite sum of terms

$$(am_i) \otimes (bn_j) = (a \otimes b)(m_i \otimes n_j).$$

Simple tensors span the algebraic tensor product, so the family $(m_i \otimes n_j)_{(i,j)\in I\times J}$ generates it. This family has cardinality at most $|I||J|$, which is the claimed bound. The proof also covers an empty generating family, when the corresponding module and tensor product are zero. □

> **How to read the result**
>
> The tensor product bound is constructive. Tensor the chosen seed families and act independently in the two factors. Equality can fail because a single tensor seed may contain several useful components, or because the tensor product algebra acquires additional relations. Later quantum applications should therefore cite the bound, not assume multiplicativity.

Boundary 4.10 (Tensor rank is not generally multiplicative). The inequality in proposition 4.9 can be strict. Equality requires additional hypotheses on the acting algebras and their module multiplicities and is not exported as a universal law.

Proposition 4.11 (Monotonicity under extension of scalars)**.** *Let $K/\Bbbk$ be a field extension, let A be a unital algebra over $\Bbbk$, and let M be a finitely generated left module over A. Set*

$$A_K := K \otimes_{\Bbbk} A, \qquad M_K := K \otimes_{\Bbbk} M.$$

Then

$$\boxed{\operatorname{srank}_{A_K}(M_K) \le \operatorname{srank}_A(M).}$$

Every generating family $S \subseteq M$ extends to the generating family $1 \otimes S \subseteq M_K$.

Proof. Let $S = \{s_1, \dots, s_r\}$ generate M. Every pure tensor $\lambda \otimes m \in M_K$ has an expression

$$m = \sum_{i=1}^{r} a_i s_i,$$

so

$$\lambda \otimes m = \sum_{i=1}^{r} (\lambda \otimes a_i)(1 \otimes s_i).$$

Pure tensors span M_K, hence $1 \otimes S$ generates M_K over A_K. Taking the minimum over S proves the inequality. □

> **Why the hypotheses matter**
>
> The proposition gives an inequality in one direction. A smaller generating family found after extending scalars need not descend to the original field. Descent requires additional structure. The chapter therefore exports monotonicity, not an unconditional equality.

4.5 Quotient witnesses for lower bounds

Definition 4.12 (Witness for seed rank)**.** A surjective intertwiner $\Phi : M \twoheadrightarrow N$ is a witness for seed rank of order r when $\mathrm{srank}(N) = r$. It certifies $\mathrm{srank}(M) \geq r$.

Proposition 4.13 (Scalar quotient witness)**.** *Let M be a module over A with coefficient field $\Bbbk$. Suppose M has a quotient N of dimension r on which A acts through a character*

$$\chi : A \longrightarrow \Bbbk.$$

Then

$$\boxed{\mathrm{srank}_A(M) \geq r.}$$

Proof. For every $n \in N$, the cyclic submodule An is contained in the scalar span $\Bbbk n$. Therefore q seed images span a subspace of dimension at most q. Generating the quotient of dimension r requires at least r seeds. Quotient monotonicity transfers the lower bound to M. □

What we have established

Seed rank is functorial under quotients and isomorphisms, subadditive across extensions, given by the maximum for independent product actions, and bounded above under tensor products. Lower bounds come from quotients on which the action simplifies. Scalar extension cannot increase seed rank, but descent of a reduced seed family is a separate problem.

5 Generator number modulo the Jacobson radical

A route through this section

For a finitely generated module, Nakayama's lemma allows us to count generators after passing to the quotient by the Jacobson radical. When the algebra has finite dimension, this quotient is semisimple and its multiplicities give an explicit answer. We begin with the radical reduction and then calculate those multiplicities.

5.1 Reduction to the module top

Classical Input 5.1 (Nakayama lemma)**.** Let A be a unital ring, let $J \subseteq \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. If $M = N + JM$ for a submodule $N \leq M$, then $M = N$.

Theorem 5.2 (Generator number is detected modulo the Jacobson radical)**.** *Let A be a unital ring, let $J = \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. A family $S \subseteq M$ generates M if and only if its image generates M/JM as a module over A/J. When A/J is semisimple Artinian, this quotient is the usual semisimple top. No semisimplicity of A/J is required for the general equality. Consequently,*

$$\boxed{\mathrm{srank}_A(M) = \mathrm{srank}_{A/J}(M/JM).}$$

Proof. If S generates M, its image generates every quotient. Conversely, suppose the image of S generates M/JM and put $N = AS$. Then

$$M = N + JM.$$

Nakayama's lemma gives $M = N$. Thus the same finite families generate before and after passage to the top, and the minimal cardinalities agree. □

How to read the result

The theorem says that nilpotent or radical directions can lengthen orbit chains but cannot reduce the number of independent directions in the module top that must be seeded. Once the images of the seeds generate M/JM, Nakayama's lemma forces the entire module to be generated.

A local nilpotent algebra

Let $A = \Bbbk[\varepsilon]/(\varepsilon^3)$ and $M = A^{\oplus 2}$. The radical is $J = (\varepsilon)$, and

$$M/JM \cong \Bbbk^2.$$

Therefore two seeds are necessary and sufficient. The extra nilpotent layers εM and $\varepsilon^2 M$ are generated automatically once the two top directions are seeded.

Corollary 5.3 (Local algebra formula)**.** *Let A be a local ring with Jacobson radical J and residue division ring $D = A/J$. For every finitely generated left module M over A,*

$$\boxed{\operatorname{srank}_A(M) = \dim_D(M/JM),}$$

where the dimension is that of a left vector space over D.

Proof. The module M/JM over A/J is a left vector space over D. Its minimal number of generators is its dimension over D. □

5.2 Semisimple multiplicity calculus

Classical Input 5.4 (Wedderburn and Artin decomposition)**.** Every semisimple Artinian ring is isomorphic to a finite product

$$A \cong \prod_{i=1}^{q} M_{d_i}(D_i)$$

of full matrix rings over division rings.

Lemma 5.5 (Multiplicities in the regular module)**.** *Let*

$$A = \prod_{i=1}^{q} M_{d_i}(D_i)$$

and let $S_i = D_i^{d_i}$ denote the natural simple left module for the ith factor, extended by zero across the other factors. Then, as a left module over A,

$$\boxed{A \cong \bigoplus_{i=1}^{q} S_i^{\oplus d_i}.}$$

Proof. For one factor $M_d(D)$, let e_{jj} be the diagonal matrix units. The left regular module decomposes as

$$M_d(D) = \bigoplus_{j=1}^{d} M_d(D)e_{jj}.$$

Each minimal left ideal $M_d(D)e_{jj}$ is isomorphic to the natural column module D^d. Taking the direct product of factors gives the stated decomposition. □

Theorem 5.6 (Semisimple seed rank formula over arbitrary division rings)**.** *Let*

$$A \cong \prod_{i=1}^{q} M_{d_i}(D_i)$$

be semisimple Artinian and let

$$M \cong \bigoplus_{i=1}^{q} S_i^{\oplus m_i}$$

be a finitely generated left module over A. Then

$$\boxed{\operatorname{srank}_A(M) = \max_{1 \le i \le q} \left\lceil \frac{m_i}{d_i} \right\rceil.}$$

If A is any algebra of finite dimension over a field, the same formula applied to the decomposition of $M/\operatorname{Jac}(A)M$ over $A/\operatorname{Jac}(A)$ computes $\operatorname{srank}_A(M)$.

Proof. By corollary 3.3, M is generated by r elements exactly when it is a quotient of $A^{\oplus r}$. By lemma 5.5,

$$A^{\oplus r} \cong \bigoplus_{i} S_i^{\oplus r d_i}.$$

Every short exact sequence of modules over a semisimple ring splits. Therefore a quotient isomorphic to M exists if and only if

$$m_i \le r d_i \qquad \text{for every } i.$$

The smallest such r is the displayed maximum. For a nonsemisimple algebra of finite dimension, theorem 5.2 reduces the calculation to the semisimple module $M/\operatorname{Jac}(A)M$ over $A/\operatorname{Jac}(A)$. □

How to read the result

For the factor $M_{d_i}(D_i)$, one free generator contributes d_i copies of the corresponding simple module. If the top of M contains m_i copies, at least $\lceil m_i/d_i \rceil$ free generators are required. The worst simple type controls the global answer.
This count concerns copies of a simple module, rather than the number of coordinates in a chosen basis. A single seed can reach several copies through the matrix action. The regular module records precisely how many copies one free generator can supply.

Multiplicity consumes seed capacity

For $A = M_2(\Bbbk)$ acting on $(\Bbbk^2)^{\oplus 3}$, the regular module contributes two copies of the natural simple module per generator. One generator cannot cover all three multiplicity directions. Two can, giving seed rank 2.

Corollary 5.7 (Algebraically closed multiplicity formula)**.** *Let $\Bbbk$ be algebraically closed, let A be a unital algebra of finite dimension over $\Bbbk$, put $J = \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. Assume compatible decompositions indexed by an integer $q \geq 0$:*

$$A/J \cong \prod_{i=1}^{q} M_{d_i}(\Bbbk), \qquad M/JM \cong \bigoplus_{i=1}^{q} (\Bbbk^{d_i})^{\oplus m_i},$$

where the d_i are positive integers and the m_i are nonnegative integers. The first isomorphism is an isomorphism of algebras over $\Bbbk$. The second is an isomorphism of left modules over A/J, using the natural column module for each matrix factor and zero action from the other factors. Then

$$\mathrm{srank}_A(M) = \max_{1 \leq i \leq q} \left\lceil \frac{m_i}{d_i} \right\rceil,$$

with $\max \emptyset = 0$ when $q = 0$.

Proof. By theorem 5.2, $\mathrm{srank}_A(M) = \mathrm{srank}_{A/J}(M/JM)$. Apply theorem 5.6 to the displayed decompositions of A/J and M/JM, with every $D_i = \Bbbk$. This gives the stated maximum. If $q = 0$, the unital A/J-module M/JM is zero, and theorem 5.2 gives seed rank zero. □

Corollary 5.8 (Full matrix algebra)**.** *For the natural module $S = D^d$ over $M_d(D)$,*

$$\mathrm{srank}_{M_d(D)}(S) = 1.$$

More strongly, every nonzero vector of S is cyclic.

Proof. The formula gives $\lceil 1/d \rceil = 1$. Directly, let $v \neq 0$ and choose a coordinate $v_j \neq 0$. For any target $w = (w_i)$, the matrix whose jth column has entries $w_i v_j^{-1}$ and whose other columns vanish sends v to w. □

Corollary 5.9 (Multiplicity obstruction)**.** *For the diagonal action of $M_d(D)$ on $S^{\oplus m}$,*

$$\boxed{\mathrm{srank}_{M_d(D)}(S^{\oplus m}) = \left\lceil \frac{m}{d} \right\rceil.}$$

Proof. Apply theorem 5.6 to the single simple type with multiplicity m. □

What we have established

For finitely generated modules, the minimal number of seeds is detected on the quotient by the Jacobson radical. In the semisimple top, each generator has a fixed multiplicity capacity d_i in simple type i, and the largest capacity ratio determines the exact seed rank.

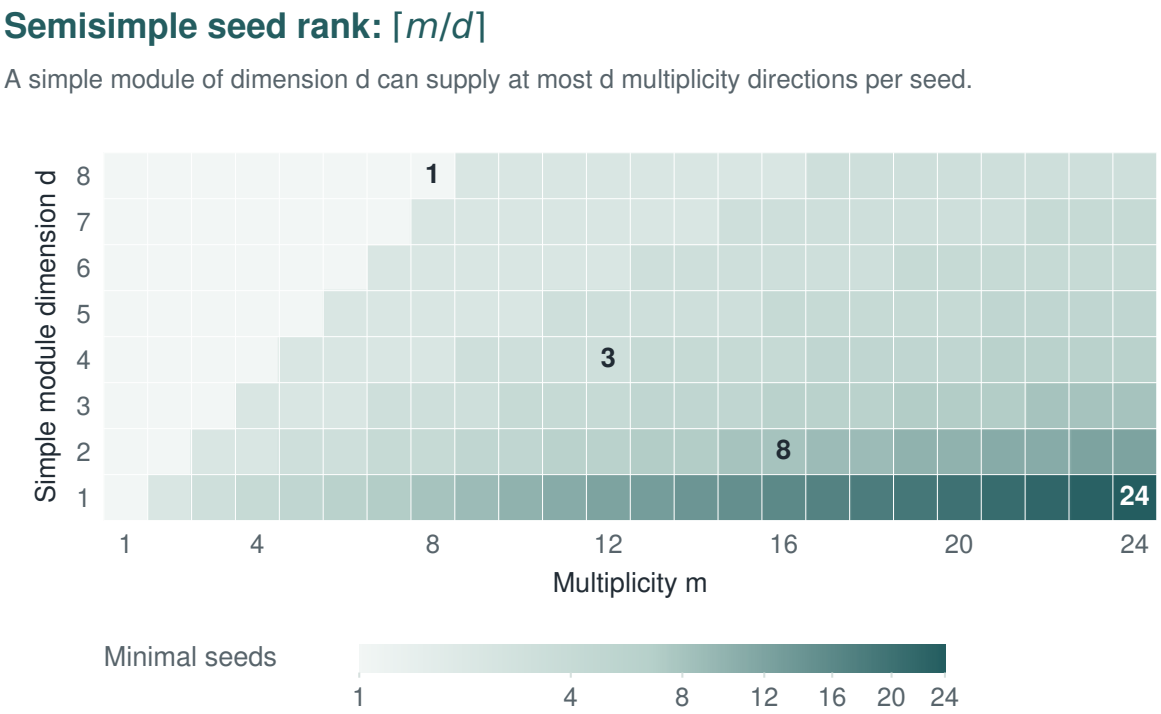


Figure 2. The semisimple multiplicity formula for one simple block. A seed can expose at most d independent multiplicity directions, so m copies require $\lceil m/d \rceil$ seeds.

6 Modules for a single operator over arbitrary fields

A route through this section

One operator is equivalent to a module over $\Bbbk[t]$. Primary decomposition isolates the irreducible polynomial sectors, and the quotient by $p(T)$ counts how many cyclic chains begin in each sector.

6.1 Primary components and generator counts

Classical Input 6.1 (Primary decomposition for polynomial modules)**.** Every module X over $\Bbbk[t]$ with finite dimension over the coefficient field decomposes as

$$X = \bigoplus_{p \in \mathrm{Irr}(\Bbbk[t])} X_p,$$

where $\mathrm{Irr}(\Bbbk[t])$ denotes the monic irreducible polynomials, only finitely many components are nonzero, and every nonzero X_p is a finite direct sum of modules $\Bbbk[t]/(p^e)$.

Theorem 6.2 (Seed rank for a single operator). *Let X have finite dimension over $\Bbbk$ and let $T \in \mathrm{End}_{\Bbbk}(X)$. Regard X as a module over $\Bbbk[t]$ by $t \cdot x = Tx$. For every monic irreducible p, let X_p be the primary component for p and set*

$$\mu_p(T) := \dim_{\Bbbk[t]/(p)}(X_p/p(T)X_p).$$

Then

$$\boxed{\mathrm{srank}_{\Bbbk[T]}(X) = \max_p \mu_p(T).}$$

Moreover, $\mu_p(T)$ is the number of cyclic primary summands for p in the decomposition into elementary divisors of X_p.

Proof. Let the minimal polynomial be

$$m_T(t) = \prod_p p(t)^{e_p}.$$

The acting algebra is

$$\Bbbk[T] \cong \Bbbk[t]/(m_T) \cong \prod_p R_p, \qquad R_p := \Bbbk[t]/(p^{e_p}),$$

by the Chinese remainder theorem. The factor R_p acts on X_p and annihilates every other primary component. Hence the formula for a direct sum with independent actions gives

$$\mathrm{srank}_{\Bbbk[T]}(X) = \max_p \mathrm{srank}_{R_p}(X_p).$$

Each R_p is local with maximal ideal generated by the class of p and residue division field $K_p = \Bbbk[t]/(p)$. By corollary 5.3,

$$\mathrm{srank}_{R_p}(X_p) = \dim_{K_p} X_p/p(T)X_p = \mu_p(T).$$

Finally, every summand $\Bbbk[t]/(p^e)$ contributes one copy of K_p after quotienting by p, so the dimension equals the number of summands in the elementary divisor decomposition. □

How to read the result

For one operator, the quotient $X_p/p(T)X_p$ counts the number of independent primary chains for the irreducible polynomial p. Over an algebraic closure these become Jordan chains. The seed rank is the largest chain count among the primary sectors, not the largest chain length.

Block count versus block size

A nilpotent operator with Jordan block sizes $(5, 1, 1)$ has seed rank 3, even though its largest block has length 5. Three seeds are needed because there are three independent chains. Once their initial vectors are chosen, repeated application of T generates every position in those chains.

6.2 Invariant factors, Jordan blocks, and cyclicity

Corollary 6.3 (Invariant factor count)**.** *If $X \neq 0$ and the rational canonical decomposition is written with nonconstant invariant factors*

$$X \cong \bigoplus_{j=1}^{r} \mathbb{k}[t]/(f_j), \qquad f_1 \mid f_2 \mid \cdots \mid f_r,$$

then

$$\boxed{\operatorname{srank}_{\mathbb{k}[T]}(X) = r.}$$

Proof. For each irreducible p, the number of invariant factors divisible by p equals the number of elementary divisors for p. Because $f_1 \mid \cdots \mid f_r$ and $f_1 \neq 1$ after zero summands are removed, some irreducible divisor of f_1 divides all r invariant factors. Hence $\max_p \mu_p(T) = r$. □

Corollary 6.4 (Formula for geometric multiplicity)**.** *If $\mathbb{k}$ is algebraically closed, then*

$$\boxed{\operatorname{srank}_{\mathbb{k}[T]}(X) = \max_{\lambda \in \operatorname{Spec}(T)} \dim_{\mathbb{k}} \operatorname{Ker}(T - \lambda I).}$$

Proof. The irreducibles are $p = t - \lambda$. On the primary component for λ, the quotient by $(T - \lambda I)X_\lambda$ has dimension equal to the number of Jordan blocks, which is also $\dim \operatorname{Ker}(T - \lambda I)$. □

Corollary 6.5 (Criterion for a cyclic matrix)**.** *For $X \neq 0$ of finite dimension, the following are equivalent:*

(i) *T has a cyclic vector.*

(ii) $\operatorname{srank}_{\mathbb{k}[T]}(X) = 1$.

(iii) *Every nonzero primary component for p is cyclic.*

(iv) *The rational canonical form has one invariant factor.*

(v) *The minimal polynomial equals the characteristic polynomial.*

(vi) *After scalar extension to an algebraic closure, every eigenvalue has geometric multiplicity one.*

Proof. The equivalence of (i) and (ii) is proposition 4.2. By theorem 6.2, (ii) holds exactly when every $\mu_p(T)$ belonging to a nonzero primary component equals one, which is (iii). Relatively prime primary cyclic modules combine into one cyclic module by the Chinese remainder theorem, so (iii) is equivalent to (iv). In rational canonical form, one invariant factor means that the unique invariant factor is both the minimal and characteristic polynomial, proving (iv)⇔(v). Scalar extension preserves the equality of minimal and characteristic polynomials. Over an algebraic closure, that equality is equivalent to one Jordan block for each eigenvalue, which is (vi). □

> **What we have established**
>
> A single operator is cyclic exactly when each primary sector has one cyclic chain. Over an algebraic closure this is equivalent to one Jordan block per eigenvalue and to equality of the minimal and characteristic polynomials.

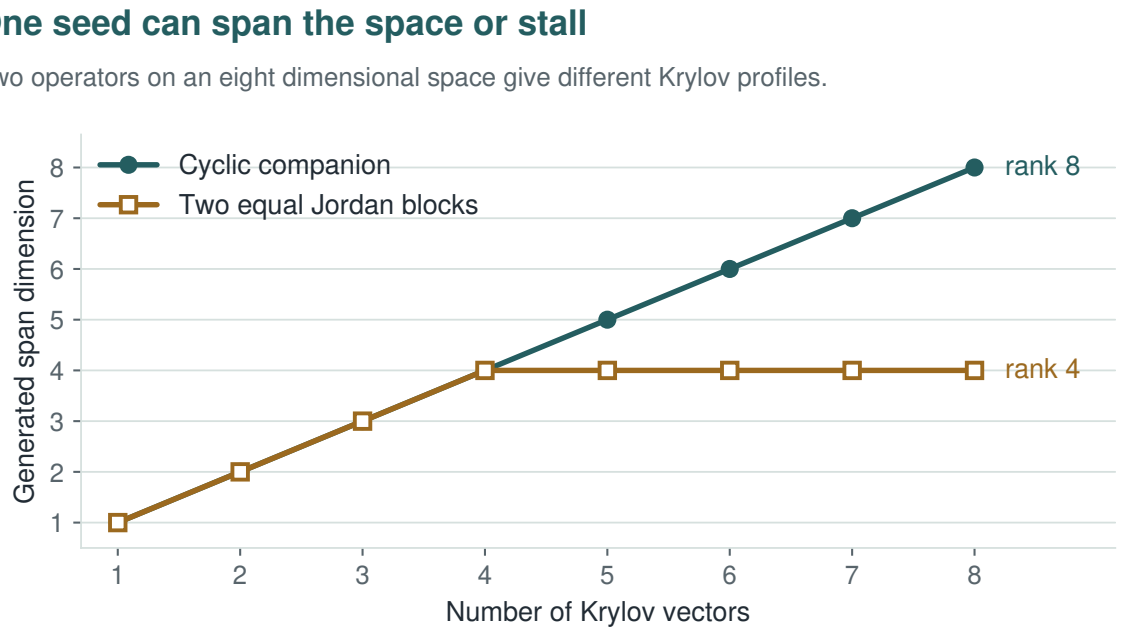


Figure 3. Krylov rank profiles for a cyclic companion operator and for two equal primary blocks. The second orbit stalls before reaching the ambient dimension because one seed cannot separate the repeated blocks.

7 Exact finite and countable packet coordinates

A route through this section

Packets give exact coordinates through idempotent maps. For finitely many packets, the reconstruction is algebraic. For countably many packets, we must also specify how the series converges and bound the coordinate maps. We begin with finite reconstruction, then consider packet support, tensor products, duality, and the universal mapping property before turning to convergence.

7.1 Finite exact packets

Definition 7.1 (Finite packet system)**.** A finite packet system on a vector space X over $\mathbb{k}$ is a family

$$\mathcal{P} = (P_\alpha)_{\alpha \in I} \subseteq \mathrm{End}_{\mathbb{k}}(X)$$

satisfying

$$P_\alpha^2 = P_\alpha, \qquad P_\alpha P_\beta = 0 \quad (\alpha \neq \beta), \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

Set $X_\alpha = P_\alpha X$. Zero packet spaces may be removed without changing any identity. No orthogonality is assumed unless stated.

Definition 7.2 (Packet support and packet pure vectors)**.** For a finite packet system $\mathcal{P} = (P_\alpha)_{\alpha\in I}$ and $x \in X$, define

$$\operatorname{Supp}_{\mathcal{P}}(x) := \{\alpha \in I : P_\alpha x \neq 0\}.$$

The vector x is *packet pure* if $|\operatorname{Supp}_{\mathcal{P}}(x)| \leq 1$. A finite family is packet pure if each of its members is packet pure, although different members may lie in different packets.

How to read the result

Packet support records which packet coordinates are present. A seed may lie in one packet, have coordinates in several packets, or be zero. Later, when we count the seeds needed for generation, this distinction matters because a seed with several nonzero coordinates can meet several graph sources at once.

Lemma 7.3 (Finite packet aggregation)**.** *Let $\sigma : I \twoheadrightarrow J$ be a surjection and define*

$$Q_j := \sum_{\sigma(\alpha)=j} P_\alpha.$$

Then $(Q_j)_{j\in J}$ is a finite packet system and

$$Q_j X = \bigoplus_{\sigma(\alpha)=j} X_\alpha.$$

Proof. Pairwise annihilation of the P_α gives

$$Q_j^2 = Q_j, \qquad Q_j Q_k = 0 \quad (j \neq k).$$

Summing over j recovers $\sum_\alpha P_\alpha = I$. The range statement follows from the directness forced by pairwise annihilation. □

Definition 7.4 (Packet analysis and synthesis)**.** For a finite packet system, define

$$\mathcal{A}_{\mathcal{P}} : X \longrightarrow \bigoplus_{\alpha\in I} X_\alpha, \qquad x \longmapsto (P_\alpha x)_\alpha,$$

and

$$\mathcal{S}_{\mathcal{P}} : \bigoplus_{\alpha\in I} X_\alpha \longrightarrow X, \qquad (x_\alpha)_\alpha \longmapsto \sum_\alpha x_\alpha.$$

Theorem 7.5 (Exact packet coordinate isomorphism)**.** *For every finite packet system,*

$$\boxed{\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}} = I_X, \qquad \mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}} = I_{\oplus X_\alpha}.}$$

Hence

$$\boxed{X \cong \bigoplus_{\alpha\in I} X_\alpha.}$$

Proof. For $x \in X$,

$$\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}}x = \sum_{\alpha} P_\alpha x = x.$$

If $x_\alpha \in X_\alpha$, then $P_\beta x_\alpha = \delta_{\alpha\beta} x_\alpha$, so

$$\mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}}(x_\alpha)_\alpha = \left(P_\beta \sum_{\alpha} x_\alpha\right)_\beta = (x_\beta)_\beta.$$

Thus analysis and synthesis are inverse linear maps. □

The central idea

Analysis records every packet coordinate and synthesis adds them back. The theorem gives an exact change of coordinates. Its proof uses the idempotent identities and does not require orthogonality. Questions about norms, energy, and conditioning require further structure.

Corollary 7.6 (Packetwise equality)**.** *For $x, y \in X$,*

$$x = y \quad \Longleftrightarrow \quad P_\alpha x = P_\alpha y \text{ for every } \alpha.$$

Proof. The forward implication is immediate. For the reverse implication, sum the packet equalities and use $\sum_\alpha P_\alpha = I$. □

Corollary 7.7 (Universal mapping property of finite packet coordinates)**.** *Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ be a finite packet system on X, and let Y be a vector space over $\Bbbk$.*

1. *Every family of linear maps $f_\alpha : X_\alpha \to Y$ determines a unique linear map $f : X \to Y$ with $f|_{X_\alpha} = f_\alpha$, namely*
$$f = \sum_{\alpha} f_\alpha P_\alpha.$$
2. *Every family of linear maps $g_\alpha : Y \to X_\alpha$ determines a unique linear map $g : Y \to X$ with $P_\alpha g = g_\alpha$, namely*
$$g = \sum_{\alpha} g_\alpha.$$

Equivalently,

$$\mathrm{Hom}(X, Y) \cong \prod_{\alpha} \mathrm{Hom}(X_\alpha, Y), \qquad \mathrm{Hom}(Y, X) \cong \prod_{\alpha} \mathrm{Hom}(Y, X_\alpha),$$

where the products are finite.

Proof. For $x = \sum_\alpha P_\alpha x$, any extension of the f_α must satisfy

$$f(x) = \sum_{\alpha} f_\alpha(P_\alpha x),$$

which proves both existence and uniqueness of the first formula. For the second, the sum $g = \sum_\alpha g_\alpha$ satisfies

$$P_\beta g = \sum_{\alpha} P_\beta g_\alpha = g_\beta$$

because $g_\alpha(Y) \subseteq X_\alpha$. Uniqueness follows from $\sum_\alpha P_\alpha = I$. □

7.2 Tensor products and dual packets

Theorem 7.8 (Tensor product packet systems). *Let $\mathcal{P} = (P_\alpha)_{\alpha\in I}$ be a finite packet system on X, and let $\mathcal{Q} = (Q_\gamma)_{\gamma\in J}$ be a finite packet system on Y. Then*

$$\mathcal{P} \boxtimes \mathcal{Q} := (P_\alpha \otimes Q_\gamma)_{(\alpha,\gamma)\in I\times J}$$

is a finite packet system on the algebraic tensor product $X \otimes_{\mathbb{k}} Y$, with packet spaces

$$(P_\alpha \otimes Q_\gamma)(X \otimes Y) = X_\alpha \otimes Y_\gamma.$$

Hence

$$\boxed{X \otimes Y \cong \bigoplus_{(\alpha,\gamma)\in I\times J} X_\alpha \otimes Y_\gamma.}$$

Proof. For every pair of packet indices,

$$(P_\alpha \otimes Q_\gamma)^2 = P_\alpha^2 \otimes Q_\gamma^2 = P_\alpha \otimes Q_\gamma.$$

If $(\alpha,\gamma) \neq (\beta,\delta)$, then either $P_\alpha P_\beta = 0$ or $Q_\gamma Q_\delta = 0$, so the corresponding tensor idempotents annihilate each other. Finally,

$$\sum_{\alpha,\gamma} P_\alpha \otimes Q_\gamma = \left(\sum_\alpha P_\alpha\right) \otimes \left(\sum_\gamma Q_\gamma\right) = I_X \otimes I_Y.$$

The range is generated by pure tensors $P_\alpha x \otimes Q_\gamma y$, hence equals $X_\alpha \otimes Y_\gamma$. Exact reconstruction follows from theorem 7.5. □

Proposition 7.9 (Dual packet systems). *Let X be finite dimensional and let $\mathcal{P} = (P_\alpha)$ be a finite packet system. The algebraic dual maps*

$$P_\alpha^* : X^* \to X^*, \qquad P_\alpha^*(\varphi) = \varphi \circ P_\alpha,$$

form a finite packet system on X^*. Its packet with index α is canonically isomorphic to X_α^*, and

$$X^* \cong \bigoplus_\alpha X_\alpha^*.$$

Proof. Because dualization reverses products,

$$P_\alpha^* P_\beta^* = (P_\beta P_\alpha)^* = \delta_{\alpha\beta} P_\alpha^*.$$

Also $\sum_\alpha P_\alpha^* = I_{X^*}$. Thus the dual maps form a packet system. Restriction gives a map from $P_\alpha^* X^*$ to X_α^*. It is injective because a covector in the range of P_α^* vanishes on every other packet, and it is surjective because any covector on X_α extends to X through the direct sum decomposition. □

Tensoring the oblique packet split

Tensor the two oblique packets in Running Example R3 with the coordinate packets of $\mathbb{k}^3$. The result has six exact packets. No orthogonality is created by tensoring, but the analysis and synthesis identities remain exact because they are idempotent identities.

7.3 Banach stability and coordinate conditioning

Exact reconstruction answers whether the coordinates retain the whole vector. Stability asks how errors in those coordinates affect the reconstructed vector. A finite packet system answers the first question through its idempotent identities. On a Banach space, boundedness of analysis and synthesis answers the second. The chosen norm on the coordinate space matters, so we fix it before comparing operator sizes.

Theorem 7.10 (Finite Banach packet isomorphism)**.** *Let X be Banach and let $(P_\alpha)_{\alpha\in I}$ be a finite family of bounded pairwise annihilating idempotents with sum I_X. Then every range $X_\alpha = P_\alpha X$ is closed. For any finite product norm equivalent to the maximum norm on $\bigoplus_\alpha X_\alpha$, packet analysis and synthesis are bounded inverse operators.*

Proof. The range of a bounded idempotent is its space of fixed points,

$$P_\alpha X = \operatorname{Ker}(I - P_\alpha),$$

which is closed. Because I is finite, analysis and synthesis are bounded for the maximum and sum norms, respectively, and therefore for every equivalent finite product norm. Their inverse identities are algebraic and were proved in theorem 7.5. □

Definition 7.11 (Packet coordinate condition number)**.** Fix a finite product norm on $\bigoplus_\alpha X_\alpha$ equivalent to the maximum norm, as in theorem 7.10. Define

$$\kappa(\mathcal{P}) := \|\mathcal{A}_\mathcal{P}\|\,\|\mathcal{S}_\mathcal{P}\|.$$

Proposition 7.12 (Conditioned coordinate transport)**.** *For a finite Banach packet system with $X \neq 0$,*

$$\kappa(\mathcal{P}) \geq 1.$$

If $T \in \mathcal{B}(X)$ and

$$[T]_\mathcal{P} := \mathcal{A}_\mathcal{P} T \mathcal{S}_\mathcal{P},$$

then

$$\boxed{\|[T]_\mathcal{P}\| \leq \kappa(\mathcal{P})\|T\|, \qquad \|T\| \leq \kappa(\mathcal{P})\|[T]_\mathcal{P}\|.}$$

Proof. Since $I = \mathcal{S}_\mathcal{P}\mathcal{A}_\mathcal{P}$,

$$1 = \|I\| \leq \|\mathcal{S}_\mathcal{P}\|\|\mathcal{A}_\mathcal{P}\|.$$

The first displayed estimate follows from submultiplicativity. The inverse identity

$$T = \mathcal{S}_\mathcal{P}[T]_\mathcal{P}\mathcal{A}_\mathcal{P}$$

gives the second estimate. □

How to read the result

The quantity $\kappa(\mathcal{P})$ measures how strongly the chosen packet coordinates can amplify perturbations. An orthogonal Hilbert decomposition has the best possible behavior in the natural ℓ^2 sum norm. An oblique packet system can reconstruct every vector exactly and still have a large condition number.

Corollary 7.13 (Completion preserves a finite packet split)**.** *Let M be a dense normed subspace of a Banach space X. Suppose every bounded P_α leaves M invariant and the restricted projectors form a packet system on M. Then*

$$P_\alpha X = \overline{P_\alpha M}$$

and the packet decomposition of M extends uniquely to the completion X.

Proof. For $x \in X$, choose $m_n \in M$ with $m_n \to x$. Continuity gives $P_\alpha m_n \to P_\alpha x$, so $P_\alpha X \subseteq \overline{P_\alpha M}$. The reverse inclusion follows because $P_\alpha X$ is closed and contains $P_\alpha M$. The bounded extensions remain pairwise annihilating idempotents whose finite sum is the identity. □

7.4 Countable packets and unconditional convergence

For a finite family, synthesis adds a fixed number of coordinates. For a countable family, it must also take a limit. The algebraic idempotent identities still separate the coordinates, but they do not by themselves guarantee that the sum converges. The next definition includes unconditional convergence in the coordinate space itself, so synthesis is defined on precisely the sequences whose packet sums can be assembled in this way.

Definition 7.14 (Unconditional packet sum space)**.** Let $(X_n)_{n\ge1}$ be closed subspaces of a real or complex Banach space X. Define

$$\bigoplus_{n\ge1}^{u} X_n$$

to be the space of sequences $x = (x_n)$ with $x_n \in X_n$ for which $\sum_n x_n$ converges unconditionally in X. Equip it with

$$\|x\|_u := \sup_{F\Subset\mathbb{N}} \sup_{|a_n|\le1} \left\| \sum_{n\in F} a_n x_n \right\|.$$

Lemma 7.15 (Completeness of the unconditional packet sum)**.** *The normed space $\bigoplus_{n\ge1}^{u} X_n$ in definition 7.14 is Banach.*

Proof. Let $x^{(k)} = (x_n^{(k)})$ be Cauchy in $\|\cdot\|_u$. For every fixed n,

$$\|x_n^{(k)} - x_n^{(\ell)}\| \le \|x^{(k)} - x^{(\ell)}\|_u,$$

so $x_n^{(k)}$ converges to some $x_n \in X_n$ because X_n is closed. Given $\varepsilon > 0$, choose k_0 such that

$$\|x^{(k)} - x^{(\ell)}\|_u < \varepsilon \qquad (k, \ell \ge k_0).$$

For finite F and $|a_n| \le 1$, pass $\ell \to \infty$ in the finite sum to obtain

$$\left\| \sum_{n \in F} a_n (x_n^{(k)} - x_n) \right\| \le \varepsilon \qquad (k \ge k_0).$$

Hence $\|x^{(k)} - x\|_u \le \varepsilon$ once x is known to be unconditionally summable.

Choose such a $k \ge k_0$. Since $x^{(k)}$ is unconditionally summable, there is a finite F_0 such that every finite $F \subseteq \mathbb{N} \setminus F_0$ and every $|a_n| \le 1$ satisfy

$$\left\| \sum_{n \in F} a_n x_n^{(k)} \right\| < \varepsilon.$$

The preceding uniform estimate then gives

$$\left\| \sum_{n \in F} a_n x_n \right\| < 2\varepsilon.$$

Thus the tails of $\sum_n x_n$ are uniformly small under bounded scalar changes, which is the unconditional Cauchy criterion. Therefore $x \in \bigoplus^u X_n$, and the earlier estimate proves $x^{(k)} \to x$ in $\|\cdot\|_u$. □

Theorem 7.16 (Countable unconditional packet isomorphism)**.** *Let X be a real or complex Banach space and let $(P_n)_{n \ge 1} \subseteq \mathcal{B}(X)$ satisfy*

$$P_n P_m = \delta_{nm} P_n$$

and

$$x = \sum_{n \ge 1} P_n x$$

unconditionally for every $x \in X$. Assume

$$K := \sup_{F \in \mathbb{N}} \sup_{|a_n| \le 1} \left\| \sum_{n \in F} a_n P_n \right\| < \infty.$$

Then analysis and synthesis define inverse Banach space isomorphisms

$$\boxed{X \cong \bigoplus_{n \ge 1}^{u} P_n X.}$$

More precisely,

$$\|\mathcal{A}_{\mathcal{P}}\| \le K, \qquad \|\mathcal{S}_{\mathcal{P}}\| \le 1.$$

Proof. Each $P_n X = \operatorname{Ker}(I - P_n)$ is closed. For $x \in X$,

$$\|\mathcal{A}_{\mathcal{P}} x\|_u = \sup_{F, a} \left\| \sum_{n \in F} a_n P_n x \right\| \le K \|x\|.$$

For $y = (y_n) \in \bigoplus^u P_n X$, define

$$\mathcal{S}_{\mathcal{P}} y = \sum_n y_n.$$

Since the finite partial sums with every coefficient equal to one are among those used in $\|y\|_u$,

$$\|\mathcal{S}_{\mathcal{P}} y\| \le \|y\|_u.$$

The assumed expansion gives $\mathcal{S}_{\mathcal{P}} \mathcal{A}_{\mathcal{P}} x = x$. Conversely, boundedness of P_m permits termwise application to the unconditionally convergent series:

$$P_m \mathcal{S}_{\mathcal{P}} y = \sum_n P_m y_n = y_m.$$

Thus $\mathcal{A}_{\mathcal{P}} \mathcal{S}_{\mathcal{P}} = I$. Completeness of the coordinate space is lemma 7.15. □

Remark 7.17 (Uniform boundedness). If every expansion $\sum_n P_n x$ is unconditional, the operators obtained from finite partial sums with signs form a pointwise bounded family. The uniform boundedness principle therefore yields a finite constant K. We retain this constant explicitly in theorem 7.16 because later estimates use it to bound the coordinate map.

Boundary 7.18 (Infinite packet sums). For an infinite family, an identity such as $\sum_\alpha P_\alpha = I$ requires a declared convergence mode and the corresponding uniform boundedness statement. Pointwise, strong, norm, weak, and unconditional convergence are not interchangeable.

What we have established

Finite packet systems give exact coordinates in a direct sum, with no orthogonality requirement. Their universal property assembles arbitrary packetwise linear maps uniquely. Tensor products and duals preserve packet structure. In Banach spaces, the analysis and synthesis norms control conditioning. The countable theory developed here uses unconditional convergence and a uniform bound on finite partial sums with signs.

8 Transfer block algebra and support patterns

A route through this section

Once vectors are resolved into packets, every operator becomes a matrix of maps between packet spaces. To multiply operators, we sum over the possible intermediate packets. The blocks permitted to be nonzero define a relation on the packet indices. When that relation is a preorder, the resulting algebra has a triangular structure and a nilpotent ideal of strict transfers.

8.1 Exact operator coordinates

Definition 8.1 (Transfer block). For $T \in \mathrm{End}_{\Bbbk}(X)$ and a finite packet system $\mathcal{P}$, define

$$T_{\beta\alpha} := P_\beta T P_\alpha \in \mathrm{Hom}_{\Bbbk}(X_\alpha, X_\beta).$$

Theorem 8.2 (Exact operator reconstruction). *Every $T \in \mathrm{End}_{\Bbbk}(X)$ satisfies*

$$\boxed{T = \sum_{\alpha,\beta \in I} T_{\beta\alpha}.}$$

Proof. Insert the finite packet identity on both sides:

$$T = \left(\sum_{\beta} P_\beta\right) T \left(\sum_{\alpha} P_\alpha\right) = \sum_{\alpha,\beta} P_\beta T P_\alpha.$$

□

How to read the result

Operator reconstruction inserts the identity on both sides of T. The source index α records where a vector begins, and the target index β records where its image lands. The resulting matrix is exact, and each entry is itself a linear map.

Theorem 8.3 (Packet matrix multiplication). *For $S, T \in \mathrm{End}_{\Bbbk}(X)$,*

$$\boxed{(ST)_{\gamma\alpha} = \sum_{\beta \in I} S_{\gamma\beta} T_{\beta\alpha}.}$$

Proof. Insert $I = \sum_\beta P_\beta$ between S and T:

$$P_\gamma S T P_\alpha = \sum_{\beta} P_\gamma S P_\beta T P_\alpha.$$

□

The central idea

The intermediate packet index in block multiplication plays the role of a summed matrix index. Repeated multiplication therefore becomes a sum over packet paths. Chapter II adds weights and filtrations to this path expansion. Chapter III adds nonlinear trees.

Corollary 8.4 (Packet powers and paths). *For $m \geq 1$,*

$$(T^m)_{\beta_m \beta_0} = \sum_{\beta_1, \ldots, \beta_{m-1}} T_{\beta_m \beta_{m-1}} \cdots T_{\beta_1 \beta_0}.$$

The same formula holds for a product of nonidentical operators with the corresponding blocks in temporal order.

Proof. Induct on m using theorem 8.3. For a product of different operators, use the same induction with the specified operator at each step. □

Corollary 8.5 (Adjoint reversal). *If X is Hilbert, every P_α is orthogonal, and $T \in \mathcal{B}(X)$, then*

$$\boxed{(T_{\beta\alpha})^* = (T^*)_{\alpha\beta}.}$$

Proof. Because $P_\alpha^* = P_\alpha$,

$$(P_\beta T P_\alpha)^* = P_\alpha T^* P_\beta.$$

□

8.2 The full block algebra and intertwining equations

Corollary 8.6 (Tensor product transfer blocks). *Let $S \in \mathrm{End}(X)$ and $T \in \mathrm{End}(Y)$, with packet systems $\mathcal{P}$ and $\mathcal{Q}$. Relative to the tensor product packet system of theorem 7.8,*

$$\boxed{(S \otimes T)_{(\beta,\delta),(\alpha,\gamma)} = S_{\beta\alpha} \otimes T_{\delta\gamma}.}$$

Proof. By definition,

$$\begin{aligned}(S \otimes T)_{(\beta,\delta),(\alpha,\gamma)} &= (P_\beta \otimes Q_\delta)(S \otimes T)(P_\alpha \otimes Q_\gamma) \\ &= (P_\beta S P_\alpha) \otimes (Q_\delta T Q_\gamma) \\ &= S_{\beta\alpha} \otimes T_{\delta\gamma}.\end{aligned}$$

□

Theorem 8.7 (Blockwise intertwining equations). *Let $T \in \mathrm{End}(X)$, $S \in \mathrm{End}(Y)$, let $\mathcal{P} = (P_\alpha)_{\alpha\in I}$ and $\mathcal{Q} = (Q_\gamma)_{\gamma\in J}$ be finite packet systems, and let $U : X \to Y$ be linear. Define*

$$T_{\beta\alpha} = P_\beta T P_\alpha, \qquad S_{\gamma\delta} = Q_\gamma S Q_\delta, \qquad U_{\gamma\alpha} = Q_\gamma U P_\alpha.$$

Then

$$\boxed{UT = SU}$$

if and only if, for every $\gamma \in J$ and $\alpha \in I$,

$$\boxed{\sum_{\beta\in I} U_{\gamma\beta} T_{\beta\alpha} = \sum_{\delta\in J} S_{\gamma\delta} U_{\delta\alpha}.}$$

Proof. Apply Q_γ on the left and P_α on the right. Inserting $I_X = \sum_\beta P_\beta$ gives

$$Q_\gamma U T P_\alpha = \sum_\beta Q_\gamma U P_\beta T P_\alpha = \sum_\beta U_{\gamma\beta} T_{\beta\alpha}.$$

Inserting $I_Y = \sum_\delta Q_\delta$ gives

$$Q_\gamma S U P_\alpha = \sum_\delta Q_\gamma S Q_\delta U P_\alpha = \sum_\delta S_{\gamma\delta} U_{\delta\alpha}.$$

Thus $UT = SU$ implies every block equation. Conversely, if every block equation holds, then all blocks of $UT - SU$ vanish. Exact operator reconstruction yields $UT - SU = 0$. □

How to read the result

A global intertwiner is equivalent to a coupled finite system of packet equations. Later arguments about holonomy and representations use this theorem after the local blocks have been constructed. The equations state the exact compatibility those blocks must satisfy.

Theorem 8.8 (Full block algebra representation)**.** *The map*

$$\Theta_{\mathcal{P}} : \mathrm{End}_{\Bbbk}(X) \longrightarrow \prod_{\beta,\alpha\in I} \mathrm{Hom}_{\Bbbk}(X_\alpha, X_\beta), \qquad T \longmapsto (T_{\beta\alpha})_{\beta,\alpha},$$

is an algebra isomorphism when the codomain is equipped with matrix multiplication whose entries are operators.

Proof. Injectivity follows from exact reconstruction. Given arbitrary linear maps $B_{\beta\alpha} : X_\alpha \to X_\beta$, define

$$T\left(\sum_\alpha x_\alpha\right) := \sum_{\alpha,\beta} B_{\beta\alpha} x_\alpha.$$

The packet decomposition is unique and the index set is finite, so this defines a linear operator whose blocks are the prescribed maps. Thus $\Theta_{\mathcal{P}}$ is surjective. Compatibility with multiplication is theorem 8.3. □

Corollary 8.9 (Propagation of packet support)**.** *Let $T \in \mathrm{End}(X)$ and $x \in X$. Then*

$$\boxed{\mathrm{Supp}_{\mathcal{P}}(Tx) \subseteq \left\{\beta : \exists \alpha \in \mathrm{Supp}_{\mathcal{P}}(x) \text{ with } T_{\beta\alpha} \neq 0\right\}.}$$

Equality holds whenever, for every target β in the set on the right, the active contributions

$$(T_{\beta\alpha} P_\alpha x : \alpha \in \mathrm{Supp}_{\mathcal{P}}(x),\ T_{\beta\alpha} \neq 0)$$

have nonzero sum. In particular, equality holds if at most one active source packet feeds each such β and that source coordinate does not lie in the kernel of the corresponding block.

Proof. For every target packet β,

$$P_\beta T x = \sum_{\alpha\in\mathrm{Supp}_{\mathcal{P}}(x)} T_{\beta\alpha} P_\alpha x.$$

If every block in this sum is zero, then the coordinate with index β vanishes. This proves the inclusion. Equality is exactly the assertion that the displayed active sum is nonzero for every target in the set on the right. The final sufficient condition leaves at most one nonzero summand at each target, and kernel avoidance makes that summand nonzero. □

Corollary 8.10 (Dimension count)**.** *If X is finite dimensional and $d_\alpha = \dim X_\alpha$, then*

$$\dim \mathrm{End}_{\Bbbk}(X) = \sum_{\alpha,\beta} d_\alpha d_\beta = \left(\sum_{\alpha} d_\alpha \right)^2 .$$

Proof. Take dimensions in the block algebra isomorphism and use $\dim \mathrm{Hom}(X_\alpha, X_\beta) = d_\alpha d_\beta$. □

Proposition 8.11 (Block diagonal projection and packet commutant)**.** *Define*

$$\Delta_{\mathcal{P}}(T) := \sum_{\alpha} P_\alpha T P_\alpha .$$

Then $\Delta_{\mathcal{P}}$ is a linear idempotent on $\mathrm{End}(X)$ whose range is the block diagonal subalgebra. For $T \in \mathrm{End}(X)$, the following are equivalent:

(i) $TP_\alpha = P_\alpha T$ *for every* α.

(ii) $T_{\beta\alpha} = 0$ *whenever* $\beta \neq \alpha$.

(iii) $\Delta_{\mathcal{P}}(T) = T$.

Proof. Pairwise annihilation gives $\Delta_{\mathcal{P}}^2 = \Delta_{\mathcal{P}}$ and identifies its range with diagonal block families. If T commutes with all P_α, then for $\beta \neq \alpha$,

$$P_\beta T P_\alpha = P_\beta P_\alpha T = 0.$$

Thus (i) implies (ii). Exact reconstruction shows (ii) implies (iii). If $T = \sum_\alpha P_\alpha T P_\alpha$, then

$$TP_\gamma = P_\gamma T P_\gamma = P_\gamma T,$$

so (iii) implies (i). □

Lower triangular support

Take three packets of dimension one. Permit the diagonal blocks and transfers from a source with a lower index to a target with a higher index. The support relation is a total preorder. The blocks outside the diagonal form a nilpotent ideal, and every product of three strict transfers vanishes. This finite algebraic example anticipates the weighted triangular inverses developed in Chapter II.

8.3 Support patterns, preorders, and triangular structure

Definition 8.12 (Support relation and support space)**.** For a relation $R \subseteq I \times I$, with (β, α) interpreted as an allowed transfer $\alpha \to \beta$, define

$$\mathfrak{M}_R := \{T \in \mathrm{End}(X) : T_{\beta\alpha} = 0 \text{ whenever } (\beta,\alpha) \notin R\}.$$

Theorem 8.13 (Support pattern algebra criterion)**.** *Assume $X_\alpha \neq 0$ for every $\alpha \in I$. Then:*

(i) $I_X \in \mathfrak{M}_R$ *if and only if R is reflexive.*

(ii) $\mathfrak{M}_R\mathfrak{M}_R \subseteq \mathfrak{M}_R$ *if and only if R is transitive.*

(iii) $\mathfrak{M}_R$ *is a unital subalgebra if and only if R is a preorder.*

Proof. The identity has diagonal blocks I_{X_α} and no blocks outside the diagonal, so it belongs to $\mathfrak{M}_R$ exactly when $(\alpha, \alpha) \in R$ for every α.

Assume R is transitive. If a summand

$$S_{\gamma\beta}T_{\beta\alpha}$$

in the (γ, α) block of ST is nonzero, then (γ, β) and (β, α) lie in R. Transitivity gives $(\gamma, \alpha) \in R$, so every forbidden block of ST vanishes.

Conversely, suppose transitivity fails. Choose α, β, γ with

$$(\beta, \alpha) \in R, \qquad (\gamma, \beta) \in R, \qquad (\gamma, \alpha) \notin R.$$

Choose nonzero vectors $u \in X_\alpha$, $v \in X_\beta$, and $w \in X_\gamma$. Choose linear functionals $f \in X_\alpha^*$ and $g \in X_\beta^*$ with $f(u) = g(v) = 1$. Let T have the sole block $x \mapsto f(x)v$ from X_α to X_β, and let S have the sole block $y \mapsto g(y)w$ from X_β to X_γ. Then $S, T \in \mathfrak{M}_R$, but ST has a nonzero forbidden (γ, α) block. This proves necessity. Combining reflexivity and transitivity gives (iii). □

Corollary 8.14 (Bounded support patterns closed under $*$)**.** *Let X be Hilbert, let the packet projectors be orthogonal, assume $X_\alpha \neq 0$ for every $\alpha \in I$, and put*

$$\mathfrak{M}_R^{\mathrm{b}} := \mathfrak{M}_R \cap \mathcal{B}(X).$$

Then $\mathfrak{M}_R^{\mathrm{b}}$ is closed under adjoints if and only if R is symmetric. Hence $\mathfrak{M}_R^{\mathrm{b}}$ is a unital $*$ subalgebra of $\mathcal{B}(X)$ if and only if R is an equivalence relation.

Proof. Adjoint reversal sends an allowed (β, α) block to an (α, β) block. Symmetry is therefore sufficient. If symmetry fails, choose a nonzero map of rank one in an allowed nonsymmetric block. Its adjoint occupies the forbidden reverse block, proving necessity. A reflexive, transitive, symmetric relation is an equivalence relation. □

Theorem 8.15 (Preorder support algebra decomposition)**.** *Let R be a preorder on the finite set I. Define*

$$\alpha \sim \beta \quad \Longleftrightarrow \quad (\beta, \alpha) \in R \text{ and } (\alpha, \beta) \in R.$$

Let $\overline{I} = I/\!\sim$, which carries the induced partial order, and put

$$X_C := \bigoplus_{\alpha \in C} X_\alpha.$$

Let $\mathfrak{N}_R \subseteq \mathfrak{M}_R$ consist of operators whose blocks vanish whenever source and target belong to the same equivalence class. Then:

(i) $\mathfrak{N}_R$ *is a nilpotent ideal on both sides.*

(ii) If h is the maximal number of strict steps in a chain of $\overline{I}$, then

$$\mathfrak{N}_R^{h+1} = 0.$$

(iii) The map to the diagonal classes induces an algebra isomorphism

$$\boxed{\mathfrak{M}_R/\mathfrak{N}_R \cong \prod_{C\in\overline{I}} \mathrm{End}_{\Bbbk}(X_C).}$$

Proof. A nonzero block in $\mathfrak{N}_R$ moves from one equivalence class to a strictly larger class in the quotient partial order. The block multiplication law shows that products concatenate such moves. Therefore left or right multiplication by an element of $\mathfrak{M}_R$ preserves the property of having no blocks within a single class, so $\mathfrak{N}_R$ is an ideal on both sides. A product of $h+1$ elements of $\mathfrak{N}_R$ would require a strict chain of $h+1$ steps in $\overline{I}$, which does not exist. This proves nilpotence.

Let $P_C = \sum_{\alpha\in C} P_\alpha$. Define

$$\rho : \mathfrak{M}_R \longrightarrow \prod_C \mathrm{End}(X_C), \qquad \rho(T) = (P_C T P_C)_C.$$

If an intermediate class D contributes to $P_C S T P_C$, then the preorder contains transfers $C \to D$ and $D \to C$, forcing $C = D$ in the quotient poset. Hence

$$P_C S T P_C = (P_C S P_C)(P_C T P_C),$$

so ρ is an algebra homomorphism. Every collection of endomorphisms of the X_C defines an allowed operator that is block diagonal with respect to the classes, so ρ is surjective. Its kernel is exactly $\mathfrak{N}_R$. The first isomorphism theorem gives the quotient formula. □

What we have established

Every operator has an exact packet matrix. Products sum over intermediate packet indices, tensor products factor blockwise, and global intertwiners are equivalent to coupled block equations. A support pattern is closed under multiplication precisely when its relation is transitive. For a preorder pattern, the quotient is block diagonal and the ideal of strict transfers is nilpotent.

9 Transfer graphs and directed packet generation

A route through this section

The transfer graph records which blocks between packets occur in a declared generating family. Reachability constrains every word in that family. To obtain an exact seed count from the graph, we also need hypotheses that let generation propagate along its edges.

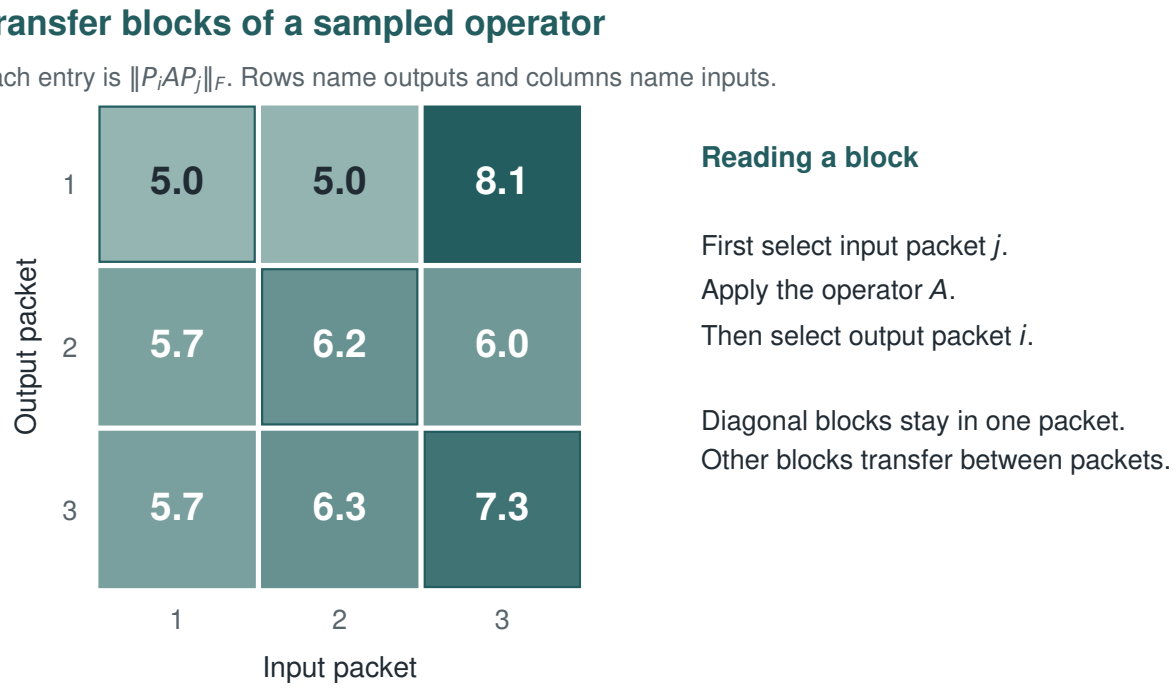


Figure 4. An operator represented by the norms of its blocks between packets. The diagonal entries preserve packets. Entries outside the diagonal describe transfers between packets.

9.1 Graphs generated by declared operators

Definition 9.1 (Transfer graph of a family generating the algebra)**.** Let $\mathcal{P} = (P_\alpha)_{\alpha\in I}$ be a finite packet system on X with every $P_\alpha \in \mathcal{A}$. Let $\mathcal{G} \subseteq \mathcal{A}$ satisfy

$$\mathcal{A} = \mathrm{alg}_{\Bbbk}(\{P_\alpha : \alpha \in I\} \cup \mathcal{G}),$$

where $\mathrm{alg}_{\Bbbk}(E)$ denotes the unital subalgebra over $\Bbbk$ generated by E. The transfer graph $\Gamma(\mathcal{G}, \mathcal{P})$ has vertex set I and an arrow

$$\alpha \longrightarrow \beta$$

when

$$P_\beta G P_\alpha \neq 0$$

for at least one $G \in \mathcal{G}$. Reachability includes paths of length zero.

Theorem 9.2 (Reachability support of the generated algebra)**.** *For every $A \in \mathcal{A}$,*

$$P_\beta A P_\alpha = 0 \qquad \textit{whenever } \beta \notin \mathrm{Reach}(\alpha).$$

Equivalently,

$$\mathcal{A} \subseteq \mathfrak{M}_{R_\Gamma}, \qquad R_\Gamma := \{(\beta, \alpha) : \beta \in \mathrm{Reach}(\alpha)\}.$$

Proof. It is enough to treat a word in the generators and packet projectors. Insert

$$I = \sum_{\gamma \in I} P_\gamma$$

between consecutive nonprojector generators. Every resulting (β, α) block is a finite sum of products

$$P_\beta G_m P_{\gamma_{m-1}} G_{m-1} \cdots P_{\gamma_1} G_1 P_\alpha.$$

A nonzero product requires every consecutive block to be nonzero, hence gives a directed path

$$\alpha \to \gamma_1 \to \cdots \to \gamma_{m-1} \to \beta.$$

Packet projectors only insert stationary steps. Thus a word has no block from α to an unreachable β. Linear combinations preserve the same zero pattern. □

How to read the result

The graph belongs to the declared generating family within $\mathrm{End}(X)$. Multiplying its generators concatenates paths, so a word in them cannot create a block outside graph reachability. Reachability alone gives less information in the other direction. A nonzero edge block may reach only a proper subspace of its target packet.

A path records a possible sequence of transfers. It does not guarantee that the composed transfer is nonzero. The image of one block may lie in the kernel of the next, and different path contributions may cancel when they are added. The reachability theorem therefore gives a restriction on support. To prove that a target packet is generated, we must also examine what the blocks do to vectors. Surjectivity along edges will provide that additional information below.

9.2 Reachability obstructions for seeds

Proposition 9.3 (Reachability obstruction for generated hulls)**.** *Let $I_0 \subseteq I$ and let*

$$S \subseteq \bigoplus_{\alpha \in I_0} X_\alpha.$$

Then

$$\boxed{\mathrm{Hull}_{\mathcal{A}}(S) \subseteq \bigoplus_{\beta \in \mathrm{Reach}(I_0)} X_\beta.}$$

In particular, no seed supported outside a source strongly connected component can generate a nonzero vector in that source component.

Proof. For $s \in S$, write $s = \sum_{\alpha \in I_0} P_\alpha s$. If β is unreachable from I_0, then for every $A \in \mathcal{A}$,

$$P_\beta A s = \sum_{\alpha \in I_0} P_\beta A P_\alpha s = 0$$

by theorem 9.2. The same holds for every finite linear combination of orbit vectors. □

9.3 Packet pure generation and source components

A strongly connected component groups packets that can reach one another. The condensation graph replaces each such group by a single vertex. It has no directed cycles between distinct components. A source component has no incoming edge from another component, so generation there must begin with seed coordinates already present in that component. The rank below records the effect of requiring each seed to lie in a single packet.

Definition 9.4 (Packet pure seed rank)**.** A vector is packet pure when it is zero or lies in one packet space X_α, equivalently when its packet support has cardinality at most one. Define

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) := \min\{|S| : S \text{ is packet-pure and } \mathcal{A}S = X\}.$$

For $C \subseteq I$, put

$$P_C := \sum_{\alpha \in C} P_\alpha, \qquad X_C := P_C X, \qquad \mathcal{A}_C := P_C \mathcal{A} P_C,$$

and define the local packet pure rank

$$r_C := \operatorname{psrank}_{\mathcal{A}_C,\mathcal{P}|_C}(X_C).$$

Proposition 9.5 (Basic packet pure rank bounds)**.** *For every A module resolved into packets with $P_\alpha \in A$,*

$$\boxed{\operatorname{srank}_A(X) \le \operatorname{psrank}_{A,\mathcal{P}}(X) \le \sum_{\alpha \in I} \operatorname{srank}_{P_\alpha A P_\alpha}(X_\alpha).}$$

The corner algebra on the right is unital on X_α with identity P_α.

Proof. Every packet pure generating family is an unrestricted generating family, which gives the first inequality. For each α, choose a minimal generating family $S_\alpha \subseteq X_\alpha$ for the corner action $P_\alpha A P_\alpha$. The union $S = \bigcup_\alpha S_\alpha$ is packet pure. Since every $x \in X$ has the unique decomposition $x = \sum_\alpha x_\alpha$ and every x_α lies in $(P_\alpha A P_\alpha) S_\alpha \subseteq AS$, the union generates X. Its cardinality is at most the displayed sum. □

Definition 9.6 (Packet system surjective along edges)**.** The module resolved into packets is surjective along edges relative to $\Gamma(\mathcal{G}, \mathcal{P})$ when, for every graph edge $\alpha \to \beta$, there exists $B \in \mathcal{A}$ such that the pure block

$$P_\beta B P_\alpha : X_\alpha \twoheadrightarrow X_\beta$$

is surjective.

Theorem 9.7 (Source component formula for packet pure seed rank)**.** *Assume I is finite, every packet space is nonzero, and the module resolved into packets is surjective along edges. Let $\mathfrak{C}_0$ be the set of source strongly connected components of the condensation graph of $\Gamma(\mathcal{G}, \mathcal{P})$. Then*

$$\boxed{\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = \sum_{C \in \mathfrak{C}_0} r_C.}$$

Proof. Let S be a packet pure generating family for X. Discard zero seeds, which do not change the generated hull, and fix a source component C. Seeds outside C cannot reach C by proposition 9.3. Moreover, for $s \in X_C$ and $A \in \mathcal{A}$,

$$P_C A s = (P_C A P_C) s \in \mathcal{A}_C s.$$

Thus the seeds of S lying in X_C must generate X_C under $\mathcal{A}_C$, and there are at least r_C of them. Summing over disjoint source components gives

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) \geq \sum_{C \in \mathfrak{C}_0} r_C.$$

For the reverse inequality, choose for each source component C a packet pure family S_C of size r_C generating X_C under $\mathcal{A}_C$, and put $S = \bigcup_C S_C$. Then every source component lies in $\mathcal{A}S$. Traverse the finite condensation graph in a topological order. Suppose all components preceding a component D that is not a source have already been shown to lie in $\mathcal{A}S$. Choose an incoming condensation edge from such a component to D, represented by a graph edge $\alpha \to \beta$ with α already generated and $\beta \in D$. Surjectivity along edges gives a pure block

$$B_{\beta\alpha} : X_\alpha \twoheadrightarrow X_\beta.$$

Because $X_\alpha \subseteq \mathcal{A}S$ and $\mathcal{A}S$ is invariant under $\mathcal{A}$, surjectivity implies $X_\beta \subseteq \mathcal{A}S$. For any other vertex $\gamma \in D$, choose a directed path inside the strongly connected component from β to γ. Successive surjective edge blocks propagate the full packet space along the path, so $X_\gamma \subseteq \mathcal{A}S$. Hence $X_D \subseteq \mathcal{A}S$. Induction over the condensation graph gives $X = \mathcal{A}S$, proving the upper bound. □

Why the hypotheses matter

The formula for source components uses local generation inside those components and surjectivity along the edges. Local generation supplies the starting packet spaces. Surjectivity then propagates their full spaces through the condensation graph. Without it, a path may exist while its blocks reach only a proper subspace.

Definition 9.8 (Packet propagating system)**.** A packet system that is surjective along edges is called packet propagating when every packet space X_α is cyclic under its diagonal corner algebra

$$P_\alpha \mathcal{A} P_\alpha.$$

Corollary 9.9 (Unweighted source component count)**.** *Assume the module resolved into packets is packet propagating and $X_\alpha \neq 0$ for every $\alpha \in I$. If c_0 denotes the number of source strongly connected components, then*

$$\boxed{\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = c_0.}$$

Proof. Fix a source component C and a vertex $\alpha \in C$. Choose a cyclic vector $s \in X_\alpha$ for $P_\alpha \mathcal{A} P_\alpha$. It generates all of X_α. Strong connectivity and surjectivity along edges then propagate the whole packet space to every vertex of C. Thus $r_C = 1$ for every source component. Apply theorem 9.7. □

Boundary 9.10 (Mixed support seeds can merge source requirements). The source component formulas concern packet pure seeds. They do not give the unrestricted seed rank. For example, let

$$\mathcal{A} = \Bbbk \times \Bbbk, \qquad X = \Bbbk \oplus \Bbbk$$

with the coordinate action and the two coordinate packet projectors. The transfer graph has two source components and

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = 2,$$

but the mixed vector $(1, 1)$ is cyclic, so

$$\operatorname{srank}_{\mathcal{A}}(X) = 1.$$

Why mixed seeds change the answer

Let the acting algebra be diagonal on $X = X_1 \oplus X_2$, with no transfer between the packets. Generation by packet pure seeds requires one seed in each source component, so psrank $= 2$. A single mixed vector $x_1 + x_2$ may generate both packets of dimension one under the diagonal action, so unrestricted seed rank can be 1.

What we have established

Graph reachability restricts which blocks can be nonzero in the algebra generated by the declared operators. Source components give the exact rank for packet pure seeds under the stated local generation and edge propagation hypotheses. Seeds with several nonzero packet coordinates have a different minimum count and may meet several source requirements at once.

10 Coordinate transport, morphisms, invariant exact sequences, and completion

A route through this section

This section follows packet data through changes of coordinates, maps between systems, and passage to a quotient. It then explains the additional assumptions needed when density, topology, or completion enters the argument.

10.1 Coordinate changes and packet compatible morphisms

Theorem 10.1 (Coordinate transport). *Let $U : X \to Y$ be a vector space isomorphism and let $\mathcal{P} = (P_\alpha)$ be a packet system on X. Define*

$$Q_\alpha := UP_\alpha U^{-1}.$$

Then $\mathcal{Q} = (Q_\alpha)$ is a packet system on Y. For $\widetilde{T} = UTU^{-1}$,

$$\boxed{Q_\beta \widetilde{T} Q_\alpha = U(P_\beta T P_\alpha)U^{-1}.}$$

Hence packet dimensions, zero patterns, and support preorders are invariant under coordinate transport. If a declared algebra generating family $\mathcal{G}$ is transported to $U\mathcal{G}U^{-1}$, then its transfer graph and reachability relation are also preserved.

Proof. Conjugation preserves products and sums, so

$$Q_\alpha Q_\beta = UP_\alpha P_\beta U^{-1} = \delta_{\alpha\beta} Q_\alpha, \qquad \sum_\alpha Q_\alpha = I_Y.$$

The block identity is direct multiplication:

$$Q_\beta \widetilde{T} Q_\alpha = UP_\beta T P_\alpha U^{-1}.$$

All listed invariants are determined by these transported blocks. □

How to read the result

An isomorphism changes the coordinates of each packet while preserving the direct sum structure. Conjugating the projectors and the operator together preserves every block identity. This is the linear prototype for the gauge changes used later in the series.

Definition 10.2 (Packet compatible operator morphism). Let $(X, \mathcal{A}, \mathcal{P})$ and $(Y, \mathcal{B}, \mathcal{Q})$ be packet resolved operator modules with packet index sets I and J. A packet compatible operator morphism is a triple

$$(\Phi, \varphi, \sigma)$$

consisting of a unital algebra homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$, a linear map $\Phi : X \to Y$ satisfying

$$\Phi(Ax) = \varphi(A)\Phi(x),$$

and a map $\sigma : I \to J$ satisfying

$$Q_j \Phi = \Phi P_{\sigma^{-1}(j)}, \qquad P_{\sigma^{-1}(j)} := \sum_{\sigma(\alpha)=j} P_\alpha.$$

An empty fiber is interpreted as the zero projector.

The packet index map records which source packets are grouped into each target packet. Its compatibility equation says that projecting after transport gives the same result as first collecting those source

packets. The intertwining equation separately ensures that the declared operator actions agree with the transport. Both conditions matter when comparing transfer blocks.

Definition 10.3 (Morphism of seeded systems)**.** A morphism

$$(X, \mathcal{A}, S, \mathcal{P}) \longrightarrow (Y, \mathcal{B}, S', \mathcal{Q})$$

is a packet compatible operator morphism satisfying the seed condition

$$\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S').$$

In the topological category all maps and actions are required to be continuous and the seed condition uses the topological hull when declared.

Theorem 10.4 (Naturality of generated hulls and transfer blocks)**.** *Let (Φ, φ, σ) be a packet compatible operator morphism. For every subset $S \subseteq X$,*

$$\boxed{\Phi(\mathrm{Hull}_{\mathcal{A}}(S)) = \mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S)) \subseteq \mathrm{Hull}_{\mathcal{B}}(\Phi(S)).}$$

For $T \in \mathcal{A}$ and $i, j \in J$,

$$\boxed{Q_j \varphi(T) Q_i \Phi = \sum_{\substack{\sigma(\beta)=j \\ \sigma(\alpha)=i}} \Phi P_\beta T P_\alpha.}$$

If Φ is surjective and S generates X, then $\Phi(S)$ generates Y under $\varphi(\mathcal{A})$ and hence under $\mathcal{B}$. If, in addition, the morphism is seeded, then S' generates Y.

Proof. Every element of $\Phi(\mathrm{Hull}_{\mathcal{A}}(S))$ is a finite sum of vectors

$$\Phi(As) = \varphi(A)\Phi(s),$$

so it belongs to $\mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S))$. Conversely, every generator $\varphi(A)\Phi(s)$ equals $\Phi(As)$, giving equality.

For the block formula, use intertwining and packet compatibility twice:

$$\begin{aligned} Q_j \varphi(T) Q_i \Phi &= Q_j \varphi(T) \Phi P_{\sigma^{-1}(i)} \\ &= Q_j \Phi T P_{\sigma^{-1}(i)} \\ &= \Phi P_{\sigma^{-1}(j)} T P_{\sigma^{-1}(i)} \\ &= \sum_{\substack{\sigma(\beta)=j \\ \sigma(\alpha)=i}} \Phi P_\beta T P_\alpha. \end{aligned}$$

If Φ is surjective and $\mathrm{Hull}_{\mathcal{A}}(S) = X$, the hull equality gives $Y = \mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S))$. If $\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S')$, then $\mathrm{Hull}_{\mathcal{B}}(\Phi(S)) \subseteq \mathrm{Hull}_{\mathcal{B}}(S')$, so S' generates Y. □

The central idea

Naturality gives two ways to compute the same transported block. One may apply the target operator between target projectors, or sum the source blocks whose packet indices map to those target indices and then apply the morphism. The theorem also identifies the image of the generated hull. These equalities make the comparison between the two systems precise.

Proposition 10.5 (Category of seeded packet systems)**.** *Packet compatible operator morphisms compose, seeded morphisms compose, and identity morphisms satisfy both definitions. Thus packet resolved operator modules form a category* **PktMod** *and seeded packet systems form a category* **SeedSys***.*

Proof. Let $(\Phi, \varphi, \sigma) : (X, \mathcal{A}, \mathcal{P}) \to (Y, \mathcal{B}, \mathcal{Q})$ and $(\Psi, \psi, \tau) : (Y, \mathcal{B}, \mathcal{Q}) \to (Z, \mathcal{C}, \mathcal{R})$ be morphisms. Intertwining is preserved because

$$\Psi\Phi(Ax) = \Psi(\varphi(A)\Phi x) = (\psi \circ \varphi)(A)\Psi\Phi x.$$

For k in the target packet set,

$$\begin{aligned} R_k \Psi\Phi &= \Psi Q_{\tau^{-1}(k)}\Phi \\ &= \Psi\Phi \sum_{\tau(\sigma(\alpha))=k} P_\alpha. \end{aligned}$$

Thus the composite packet map is $\tau \circ \sigma$. If the morphisms are seeded, then

$$\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S'), \qquad \Psi(S') \subseteq \mathrm{Hull}_{\mathcal{C}}(S'').$$

By theorem 10.4,

$$\Psi\Phi(S) \subseteq \Psi(\mathrm{Hull}_{\mathcal{B}}(S')) \subseteq \mathrm{Hull}_{\mathcal{C}}(\Psi(S')) \subseteq \mathrm{Hull}_{\mathcal{C}}(S'').$$

Identity maps satisfy the definitions with identity algebra and packet maps. □

10.2 Invariant subspaces and quotient packets

Passage to a quotient uses two separate invariance assumptions. The packet projectors must preserve the subspace for packet coordinates to descend. An operator must also preserve that subspace for its action, and hence its transfer blocks, to descend. The first condition is enough for the packet exact sequence. The second enters the corollary about transfer blocks.

Theorem 10.6 (Packet exact sequence for an invariant subspace)**.** *Let $K \leq X$ satisfy $P_\alpha K \subseteq K$ for every α. Define*

$$K_\alpha := P_\alpha K = K \cap X_\alpha$$

and

$$\overline{P}_\alpha(x + K) := P_\alpha x + K.$$

Then:

(i) The nonzero restrictions $P_\alpha|_K$ form a packet system on K.

(ii) The nonzero $\overline{P}_\alpha$ form a packet system on X/K.

(iii) For every α there is a short exact sequence

$$0 \longrightarrow K_\alpha \longrightarrow X_\alpha \longrightarrow \overline{P}_\alpha(X/K) \longrightarrow 0,$$

and

$$\overline{P}_\alpha(X/K) \cong X_\alpha/(X_\alpha \cap K) \cong (X_\alpha + K)/K.$$

(iv) The full exact sequence decomposes packetwise as the direct sum of these short exact sequences.

Proof. Restriction preserves idempotency, pairwise annihilation, and the identity sum on K. If $x \in K \cap X_\alpha$, then $x = P_\alpha x \in P_\alpha K$, proving $K_\alpha = K \cap X_\alpha$.

The quotient operator is well defined because replacing x by $x + k$ changes $P_\alpha x$ by $P_\alpha k \in K$. The idempotent and annihilation identities descend, and the sum is the identity on X/K. The map $X_\alpha \to X/K$ has image $(X_\alpha + K)/K$ and kernel $X_\alpha \cap K$, giving the short exact sequence and the quotient isomorphisms. Exact packet coordinates identify the original sequence with the direct sum over α. □

Packet descent to a quotient

Let $X = X_1 \oplus X_2$ and let $K \subseteq X_1$ be a subspace. The quotient has packets X_1/K and X_2. Any operator preserving K descends with the same organization by source and target packets, with blocks entering or leaving the first packet replaced by their induced quotient maps. An induced block can become zero even when its original block is nonzero, so the pattern of nonzero blocks can shrink. Any resulting zero packet is removed from the quotient packet system.

Theorem 10.7 (Packet descent through an invariant quotient)**.** *Under the hypotheses of theorem 10.6, the operators*

$$\overline{P}_\alpha(x + K) = P_\alpha x + K$$

are well defined pairwise annihilating idempotents with sum $I_{X/K}$, and the quotient map satisfies

$$\boxed{\pi P_\alpha = \overline{P}_\alpha \pi.}$$

After zero packets are removed, they form the canonical quotient packet system.

Proof. All assertions are the quotient part of theorem 10.6. The intertwining identity follows directly from the definition:

$$\overline{P}_\alpha \pi(x) = P_\alpha x + K = \pi P_\alpha(x).$$

□

Corollary 10.8 (Restriction and quotient of transfer blocks)**.** *Under the hypotheses of theorem 10.6, let $T \in \mathrm{End}(X)$ preserve K. Then every block $T_{\beta\alpha}$ maps K_α into K_β, and the induced quotient operator $\overline{T}$ satisfies*

$$\boxed{\overline{T}_{\beta\alpha}\,\pi = \pi T_{\beta\alpha},}$$

where $\pi : X \twoheadrightarrow X/K$.

Proof. If $x \in K_\alpha$, then $Tx \in K$, so

$$T_{\beta\alpha}x = P_\beta Tx \in K \cap X_\beta = K_\beta.$$

On the quotient,

$$\overline{T}_{\beta\alpha}\pi = \overline{P}_\beta \overline{TP}_\alpha \pi = \pi P_\beta T P_\alpha.$$

□

Corollary 10.9 (Topological packet exact sequence)**.** *If X is Banach, the P_α are bounded, and K is closed and packet invariant, then every packet sequence in theorem 10.6 is a short exact sequence of Banach spaces with bounded maps, and the quotient projectors are bounded.*

Proof. Closedness of K makes X/K Banach. The restricted projectors are bounded, and the quotient projectors are bounded because

$$\|\overline{P}_\alpha(x+K)\|_{X/K} \le \|P_\alpha\|\,\|x+K\|_{X/K}.$$

The packet spaces are closed ranges of bounded idempotents. □

10.3 Dense images, quotient topology, and completion

Algebraic generation uses finite linear combinations of operator images. Topological generation allows limits of those combinations in the declared topology. A continuous surjection carries this approximation property to the target, but identifying the target with a topological quotient requires a further assumption.

Theorem 10.10 (Dense realization theorem)**.** *Let X and Y be Hausdorff topological vector spaces, let $\Phi : X \to Y$ be continuous, surjective, and intertwining with respect to φ, and let S topologically generate X. Then $\Phi(S)$ topologically generates Y. If finite continuous packet systems satisfy*

$$Q_j \Phi = \Phi P_{\sigma^{-1}(j)},$$

then

$$\boxed{Q_j Y = \Phi(P_{\sigma^{-1}(j)} X)}$$

for every j.

Proof. Given $y = \Phi(x)$, choose a net $x_\lambda \in \mathrm{Hull}_{\mathcal{A}}(S)$ converging to x. Then

$$\Phi(x_\lambda) \to y,$$

and naturality places every $\Phi(x_\lambda)$ in $\mathrm{Hull}_{\mathcal{B}}(\Phi(S))$. Hence the latter hull is dense. For packets, one inclusion follows from compatibility. Conversely, if $y = Q_j y$, choose x with $\Phi(x) = y$. Then

$$y = Q_j \Phi(x) = \Phi P_{\sigma^{-1}(j)} x.$$

□

Theorem 10.11 (Quotient topology and completed realization)**.** *Let $\Phi : X \twoheadrightarrow Y$ be a continuous linear quotient map between Hausdorff topological vector spaces, and put $K = \operatorname{Ker}\Phi$. Then the induced map*

$$\widetilde{\Phi} : X/K \longrightarrow Y$$

is a topological isomorphism. If $D \subseteq X$ is a linear subspace whose image $\Phi(D)$ is dense in a complete space Y, then the completion of

$$D/(D \cap K)$$

in the topology transported from Y is canonically isomorphic to Y.

Proof. The algebraic first isomorphism theorem gives a bijection $\widetilde{\Phi}$. By definition of the quotient topology, a set $U \subseteq X/K$ is open exactly when its preimage in X is open. Because Φ is a quotient map, this is equivalent to $\widetilde{\Phi}(U)$ being open in Y. Thus $\widetilde{\Phi}$ is a homeomorphism. The second statement applies the dense completion argument of theorem 3.6 to the injective map $D/(D \cap K) \to Y$. □

Boundary 10.12 (Continuous surjections and quotient topology). Continuity and surjectivity transport density as in theorem 10.10. They do not by themselves identify the given topology on Y with the quotient topology from X. The stronger conclusion in theorem 10.11 requires the quotient map hypothesis or an equivalent open mapping condition.

Scope of the argument

A continuous surjective linear map need not be a quotient map in arbitrary topological vector spaces. The algebraic first isomorphism theorem gives a bijection on the quotient. A topological isomorphism requires the quotient topology to agree with the target topology. An applicable open mapping theorem can supply this agreement, as it does for continuous linear surjections between Banach spaces.

The completion statement uses the topology transported from the target space. This choice ensures that approximation in the quotient means approximation in the target. A topology inherited from the original subspace need not give the same completion unless the two topologies agree. The theorem therefore names the topology as part of its conclusion.

What we have established

Packet systems transport by conjugation, form a category under packet compatible seeded morphisms, restrict to invariant subspaces, descend to invariant quotients, and survive completion under explicit boundedness and quotient topology hypotheses.

11 Realization functors and semantic theorem transport

A route through this section

A later domain is a genuine realization only when the relevant structures and identities are preserved. The purpose of this section is to replace informal analogy by a precise preservation profile.

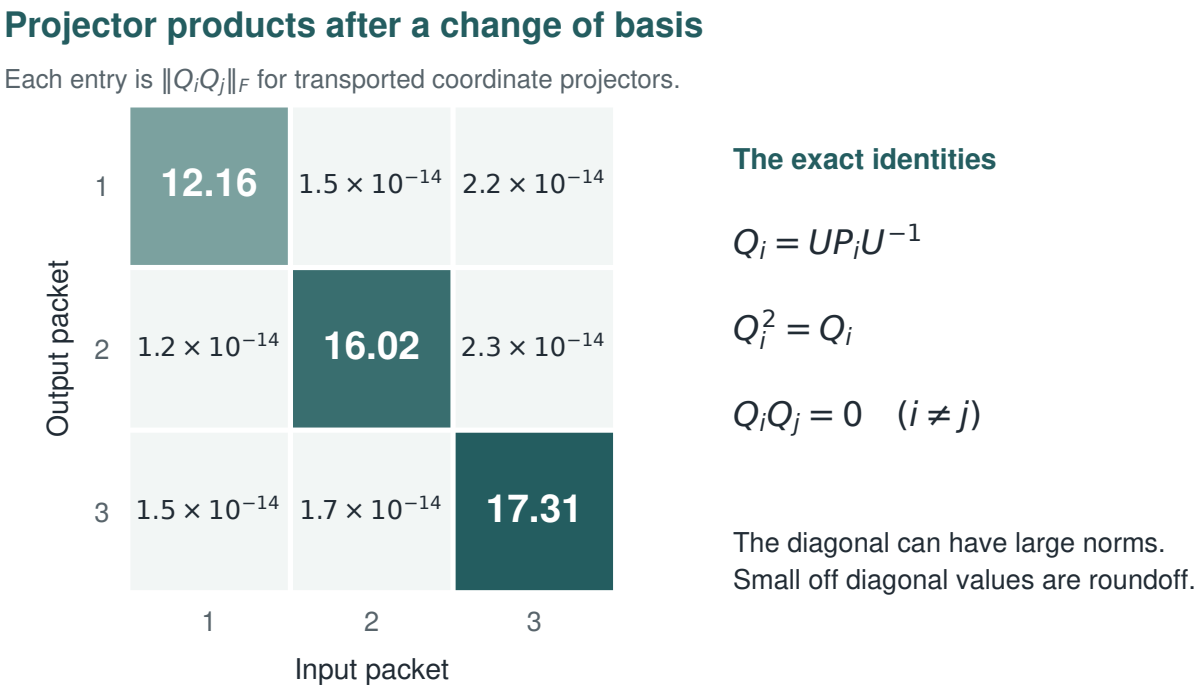


Figure 5. Numerical certificate for a nonorthogonal transported packet system. The diagonal products retain idempotent size while off diagonal products remain at roundoff level.

11.1 Preservation profiles

Definition 11.1 (Realization functor and preservation profile)**.** Let $\mathbf{SeedSys}_{\mathrm{abs}}$ be an abstract category of seed systems and let $\mathbf{C}_D$ be a domain category. A realization functor is a functor

$$\mathrm{Ev}_D : \mathbf{SeedSys}_{\mathrm{abs}} \longrightarrow \mathbf{C}_D$$

with a declared preservation profile specifying which of the following structures are preserved:

$$0,\ I,\ \text{finite sums},\ \text{composition},\ \text{biproducts},$$
$$\text{kernels},\ \text{cokernels},\ \text{quotients},\ \text{tensor products},$$
$$\text{packet idempotents},\ \text{transfer blocks},\ \text{topological completions}.$$

Preservation of a structure means preservation of its defining identity or universal property, not merely a similarity of notation.

Theorem 11.2 (Equational realization principle)**.** *Let $F : \mathbf{C} \to \mathbf{D}$ be an additive functor between preadditive categories. Any equality of morphisms constructed from declared generating morphisms by finite sums, additive inverses, identities, zero maps, and composition is preserved by F. Scalar multiplication by a coefficient ring is also preserved when the categories are enriched over that ring and F is linear for*

the declared enrichment. If a proof additionally uses a chosen biproduct, kernel, cokernel, quotient, tensor product, or completion and F preserves that construction in its declared profile, then the corresponding proof diagram is transported to $\mathbf{D}$.

Proof. Functoriality gives

$$F(I) = I, \qquad F(g \circ f) = F(g) \circ F(f),$$

and additivity gives

$$F(0) = 0, \qquad F(f + g) = F(f) + F(g), \qquad F(-f) = -F(f).$$

If the declared enrichment is present, linearity additionally gives $F(af) = aF(f)$ for every coefficient a. A structural induction on the construction of a morphism expression therefore carries every permitted equational proof to its image under F. If a step invokes a universal construction, preservation means that the image diagram satisfies the same universal property. The uniquely induced morphism in the source proof is therefore carried to the uniquely induced morphism in the target proof. Repeating this argument through the finite proof diagram transports the conclusion. □

How to read the result

The equational principle covers identities built from operations the functor preserves. Additive identities require additivity. Identities with scalar coefficients require linear enrichment. Quotients, tensor products, completions, kernels, or limits require their own preservation assumptions. The preservation profile records the assumptions needed to carry each step of the proof into the target category.

11.2 From equation transport to semantic realization

Proposition 11.3 (Realization export principle)**.** *Every canonical Chapter I identity whose proof uses only the operations listed in a realization functor's verified preservation profile remains valid after applying that functor. If the proof uses a domain specific hypothesis not encoded by the functor, that hypothesis must be verified separately before the transported statement is promoted to a theorem.*

Proof. The first assertion is theorem 11.2. The second is a logical boundary: a functor transports established identities and preserved universal constructions, but it does not create an unverified analytic, topological, positivity, finiteness, or integrality hypothesis. □

Definition 11.4 (Semantic realization criterion)**.** A domain theorem is a semantic realization of a canonical seed calculus theorem only when:

(i) An explicit realization functor or intertwining construction is given.

(ii) The preservation profile covers every construction used in the canonical proof.

(iii) Every additional analytic, finiteness, convergence, positivity, or integrality hypothesis is verified in the domain.

(iv) The resulting conclusion is stated in the domain's native objects.

Reuse of the words “seed”, “packet”, or “transfer” without these data is only syntactic reuse.

Scope of the argument

A change of nouns is not a realization theorem. Calling a frequency band, a ramification sector, and a symmetry sector a “packet” is only terminology until one exhibits the idempotents, proves reconstruction, identifies the transfer blocks, and checks that the relevant morphisms or limits preserve those structures.

What we have established

The preservation profile determines which equations and proof constructions pass to a new domain. Checking it gives later chapters a concrete test for transporting a theorem: identify the realization, check each operation used in the proof, and verify the additional domain hypotheses.

12 Canonical export interface and claim boundary

A route through this section

The export interface lets later chapters use the foundational proofs directly. Each chapter can cite the stable record ID, state its domain hypotheses, and prove the new step specific to its realization.

Definition 12.1 (Worked realization interface)**.** The foundational calculus is illustrated by four canonical algebraic realizations:

(i) A typed quiver reachability model.

(ii) An algebraic coordinate ring presentation.

(iii) A companion and repeated primary single operator model.

(iv) Path and cycle adjacency models.

The worked model sections below instantiate these four interfaces. A bilateral shift example records the analytic completion mechanism also used in Chapter V. Computational specifications connect the proofs with the verification companion and are not premises of the foundational theorems.

Definition 12.2 (Chapter I export object)**.** The canonical Chapter I export is

$$\mathsf{PSC}_I = \left(\begin{array}{c} \text{typed grammars and realized operator algebras,} \\ \text{seed systems, free presentations, and seed rank,} \\ \text{module-top, semisimple, and single-operator formulas,} \\ \text{finite and unconditional packet coordinates,} \\ \text{transfer blocks and support algebras,} \\ \text{reachability and packet-pure source generation,} \\ \text{morphisms, invariant exact sequences, quotients, completions,} \\ \text{realization functors and equational theorem transport} \end{array} \right).$$

12.1 Use across chapters

Chapter I export	Primary users	Later role
Typed grammar and module equivalence	II to XII	Gives each domain a legal operator syntax and a concrete module realization.
Free seed presentation	IV, IX, X	Converts generated families into quotient presentations and relation modules.
Jacobson top seed rank	VII, X, XII	Computes minimal generators from semisimple multiplicity data.
Single operator formula	IV, VI, IX	Resolves recurrence, companion, and spectral chain generator counts.
Exact packet coordinates	II to XII	Supplies analysis, synthesis, and packetwise equality.
Tensor and dual packet systems	V, VII, VIII, XI	Controls product systems, observables, and paired representations.
Transfer block multiplication	II to XII	Converts operator composition into sums over packet paths.
Block intertwining equations	III, VII, XI	Expresses global equivalence or gluing as local packet equations.
Support preorder algebra	II, VIII, X	Produces triangular transfer structure and nilpotent strict transfer ideals.
Source component seed rank	VIII, X	Identifies the packet pure source requirements under propagation hypotheses.
Packet exact sequences and quotient descent	III, VI, X, XII	Carries packets through constraints, reductions, and deformation quotients.
Semantic realization criterion	IV to XII	Specifies when a domain theorem is a true realization of the common calculus.

Boundary 12.3 (Exact limits of Chapter I). The existence of an exact packet decomposition does not imply that the decomposition is computationally discoverable, numerically well conditioned, statistically identifiable, dynamically invariant, or mathematically useful. The condition number in definition 7.11 quantifies only coordinate stability once the packets are supplied. Refinement towers, filtered transfer, nonlinear obstruction objects, automatic packet recovery, continuum limits, arithmetic lifts, and automorphic occupancy require additional theorems in later chapters.

13 A typed quiver model

A small quiver model shows how the path algebra records a direction that the available seeds cannot reach.

Let the quiver have vertices U, V and arrows

$$R : U \to U, \qquad A : U \to V, \qquad B : V \to U,$$

with the relation in the path algebra

$$BA = e_U,$$

whose realization is the operator identity $BA = I_U$. Realize

$$U = \Bbbk^2, \qquad V = \Bbbk^3,$$

by

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad A(u_1, u_2) = (u_1, u_2, 0), \qquad B(v_1, v_2, v_3) = (v_1, v_2).$$

Starting from the seed $e_1 \in U$, repeated application of R spans U, and A carries that span onto the first two coordinates of V. Any path from U to V ends with A and therefore lands in $\operatorname{im} A$. The third coordinate direction, which spans $\ker B$, is unreachable from a seed supported only at U. Therefore the generated representation is

$$U \oplus \operatorname{span}\{(1, 0, 0), (0, 1, 0)\} \subsetneq U \oplus V.$$

Adding the seed $(0, 0, 1) \in V$ generates the full representation.

Proposition 13.1 (Typed reachability certificate)**.** *In the preceding quiver realization, no seed supported at U can generate the full representation, whereas the two seeds*

$$e_1 \in U, \qquad e_3 \in V$$

do generate it.

Proof. The first claim follows because every path that starts at U and ends in V has final arrow A, so its image lies in $\operatorname{im} A$. For the second, e_1 and Re_1 span U. Their images under A span $\operatorname{im} A$, and e_3 spans $\ker B$, the missing complementary direction in V. □

This example shows why a typed grammar is more than a notational convenience. The obstruction lies in the subspace $\ker B \subset V$, which is unreachable from seeds supported at U, and is visible before any coordinate calculation.

The arrow from one vertex to another records which type can be reached. Its image records which directions can be reached within that type. Here the first seed reaches both vertices, yet misses one direction at the second vertex. The extra seed supplies that direction directly. This is why reachability of a vertex alone does not settle generation of its entire space.

14 Worked algebraic model: an elliptic coordinate ring

Let $\operatorname{char} \Bbbk \neq 2$ and let

$$A = \Bbbk[x, y]/(y^2 - x^3 + x).$$

The affine equation $y^2 = x^3 - x$ has classical discriminant 64, which is nonzero under this hypothesis. Standard algebraic geometry then identifies its smooth projective completion as an elliptic curve [9]. The Lean development checks the algebraic implication $2 \neq 0 \Rightarrow 64 \neq 0$, the quotient normal form, and the operator identities below. It does not formalize projective smoothness, genus, or the geometric identification of the discriminant. The packet computation itself uses only the displayed relation in the quotient ring. Every class has a unique normal form

$$f(x, y) = a(x) + yb(x).$$

Indeed, $y^2 - (x^3 - x)$ is monic of degree two in y, so division in $\Bbbk[x][y]$ gives a unique remainder of degree less than two in y. Concretely, every power y^m with $m \geq 2$ can be reduced using

$$y^2 = x^3 - x.$$

14.1 One module seed and two address operators

Take the seed

$$s = 1$$

and let M_x, M_y be multiplication by x and y. Then

$$M_x^i M_y^j s = x^i y^j,$$

so the algebra, regarded as a module over its multiplication algebra, is generated by one seed.

The two packet spaces are

$$A_0 = \Bbbk[x], \qquad A_1 = y\Bbbk[x].$$

Let

$$P_0(a + yb) = a, \qquad P_1(a + yb) = yb.$$

These idempotents are linear over $\Bbbk$, annihilate one another, and satisfy $P_0 + P_1 = I$. They are not module maps over A, because multiplication by y exchanges the two packets.

Identify the packet coefficients by

$$A_0 \oplus A_1 \longrightarrow \Bbbk[x] \oplus \Bbbk[x], \qquad (a, yb) \longmapsto (a, b).$$

In these ordered coefficient coordinates, with M_x and M_{x^3-x} denoting multiplication on $\Bbbk[x]$,

$$M_x = \begin{pmatrix} M_x & 0 \\ 0 & M_x \end{pmatrix}, \qquad M_y = \begin{pmatrix} 0 & M_{x^3-x} \\ I & 0 \end{pmatrix}.$$

The two rows of the second matrix record the outgoing packet coefficients. An even input enters the odd packet without an additional factor. An odd input returns to the even packet through the factor im-

posed by the curve equation. Applying the matrix twice therefore preserves each packet and multiplies its coefficient by that factor.

The relation is reconstructed as an operator identity:

$$M_y^2 = M_x^3 - M_x.$$

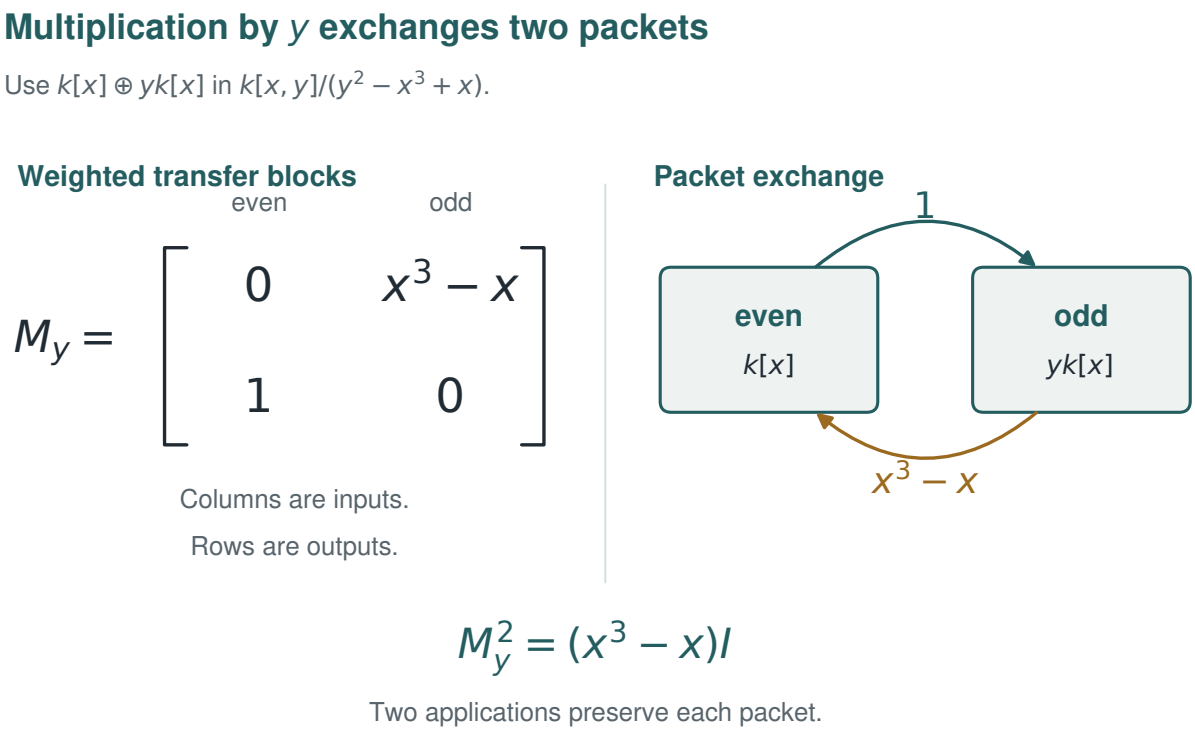


Figure 6. Multiplication by y exchanges the even and odd packets. The transfer from the odd packet to the even packet carries the relation weight $x^3 - x$.

14.2 Exact product law

Write

$$f = a_0 + ya_1, \qquad g = b_0 + yb_1.$$

Then

$$fg = (a_0b_0 + (x^3 - x)a_1b_1) + y(a_0b_1 + a_1b_0).$$

This is a convolution law for the two packets. It is already visible at the linear operator level because multiplication by a fixed element is a transfer operator. The even output collects the product of the two even coefficients and the weighted product of the two odd coefficients. The odd output collects the two products with one even and one odd coefficient. The quotient relation is thus retained in the weight rather than discarded during the decomposition.

Example 14.1 (Worked reduction)**.** Take

$$f = x^2 + y, \qquad g = x - y.$$

Then

$$\begin{aligned} fg &= x^3 - x^2y + xy - y^2 \\ &= x^3 - x^2y + xy - (x^3 - x) \\ &= x + (x - x^2)y. \end{aligned}$$

Thus

$$P_0(fg) = x, \qquad P_1(fg) = (x - x^2)y.$$

The symbolic certificate in the companion code performs the same quotient reduction exactly.

14.3 What this example establishes

The coordinate ring illustrates four distinct parts of the calculus:

(i) the unit is a module seed.

(ii) M_x and M_y are address operators.

(iii) the curve equation is an operator relation.

(iv) parity in y gives exact packet analysis, transfer, and reconstruction.

It does not imply that every algebra can be presented as an algebra generated by one element. It shows instead that unital algebras are cyclic as modules over their full multiplication algebra.

15 Worked signal model: a bilateral shift

Let $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$ and let

$$X = \ell^2(\mathbb{Z}; \mathbb{F})$$

with a family of bounded reindexing operators indexed by the integers

$$(S_m x)[n] = x[n - m] \qquad (m, n \in \mathbb{Z}).$$

Thus $S_0 = I$, $S_{m+r} = S_m S_r$, and S_1 is the usual bilateral shift. We use the standard conventions for Hilbert spaces and bounded operators of [3]. The impulse δ_0 satisfies

$$S_m \delta_0 = \delta_m$$

for every $m \in \mathbb{Z}$. Therefore sequences with finite support lie in its algebraic orbit hull and are dense in $\ell^2(\mathbb{Z})$.

Proposition 15.1 (Shift seed)**.** *For $n \in \mathbb{N}$, let $G(n)$ mean that some finite seed set with n elements has dense hull under the declared family $\{S_m : m \in \mathbb{Z}\}$. Then*

$$1 = \min\{n \in \mathbb{N} : G(n)\}.$$

Equivalently, one seed suffices and no empty seed family generates densely.

Proof. The orbit of δ_0 contains every standard basis vector. Their finite linear span is dense, so $G(1)$ holds. The hull of the empty seed set is zero, while $\ell^2(\mathbb{Z}; \mathbb{F})$ is nonzero, so $G(0)$ fails. Therefore 1 is the least admissible finite cardinality. □

For $N \geq 1$, define residue projectors

$$(P_{N,j}x)[n] = \begin{cases} x[n], & n \equiv j \pmod{N}, \\ 0, & \text{otherwise.} \end{cases}$$

They form an orthogonal packet system and satisfy

$$S_1 P_{N,j} = P_{N,j+1} S_1,$$

with indices modulo N. The shift is therefore a cyclic transfer between residue packets.

The seed claim concerns density, rather than equality with the algebraic hull. Finite linear combinations of shifted impulses give precisely the sequences with finite support. Taking their closure gives the whole Hilbert space. The packet decomposition and the choice of completion answer different questions, and both must be specified for this realization.

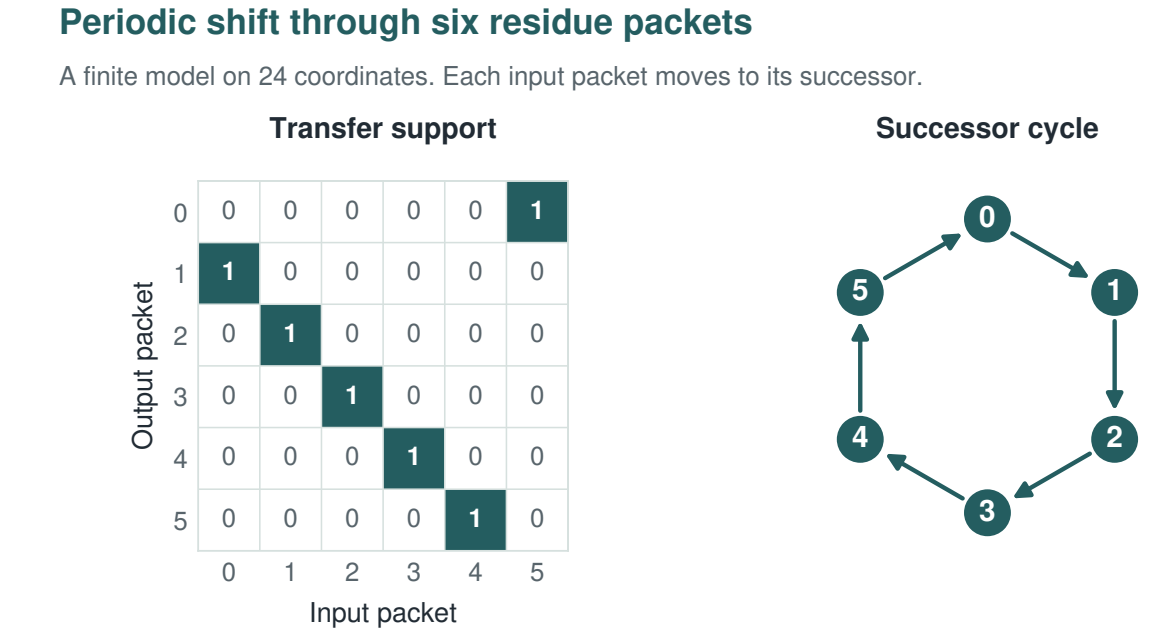


Figure 7. Support among residue packets in the periodic shift model on 24 coordinates with six packets. Each input residue transfers into the next output residue. The theorem for the infinite bilateral shift is established separately in the text.

Boundary 15.2. This section uses residue packets only to demonstrate the foundational transfer algebra. Nested refinement and structures for filter banks require additional definitions and are outside Chapter I.

16 Worked matrix models

16.1 The full matrix algebra

Let $X = \Bbbk^n$ and let $\mathcal{A} = M_n(\Bbbk) = \mathrm{End}_{\Bbbk}(X)$, with the full matrix algebra declared before a seed or target is chosen. By corollary 5.8, every nonzero seed is cyclic. This is the maximal linear grammar on X. It does not give a compression result. Under a smaller declared operator algebra, the same vector need not be cyclic.

16.2 Companion operators and repeated blocks

For one operator T, theorem 6.2 turns verified primary data into an exact certificate for the minimum number of seeds. The numerical companion compares:

(a) a nilpotent companion operator with one Jordan chain, for which the Krylov rank is $1, 2, \dots, n$.

(b) two equal Jordan blocks, for which one diagonal seed produces a chain only as long as one block and then stalls.

The observed stall is not numerical failure. The associated scalar quotient of dimension two certifies the lower bound in theorem 6.2. Identifying that quotient with the space that measures geometric multiplicity is the classical Jordan input stated above.

17 Worked graph model: adjacency algebras

Let G be a nonempty finite graph with adjacency matrix A_G, and regard $\Bbbk^{V(G)}$ as a module over $\Bbbk[A_G]$. The general result applies directly to a spectral certificate. For a symmetric real adjacency matrix, the classical spectral theorem supplies such data.

Within one eigenspace, every polynomial in the adjacency operator acts by a scalar. A seed therefore supplies at most one vector direction in that eigenspace. Distinct eigenvalues can be separated by projector polynomials, so the same seed may supply one direction to each of several eigenspaces. With a complete eigenbasis, the largest eigenspace determines how many seeds are needed, rather than the number of distinct eigenvalues.

Corollary 17.1 (Certified graph seed number)**.** *Let $\Bbbk$ be a field and let Λ be a nonempty finite set. Suppose there are pairwise distinct scalars $a_\lambda \in \Bbbk$, multiplicities $m_\lambda \in \mathbb{N}$, and an eigenbasis*

$$\{b_{\lambda,j} : \lambda \in \Lambda,\ 1 \le j \le m_\lambda\}$$

of $\Bbbk^{V(G)}$ satisfying

$$A_G b_{\lambda,j} = a_\lambda b_{\lambda,j}.$$

If $m_\lambda \le R$ for all λ and $m_{\lambda_*} = R$ for some λ_*, then R is the least natural number of adjacency seeds:

$$R = \min\left\{ r \in \mathbb{N} : \begin{array}{l} \text{there exist } s_1, \dots, s_r \text{ with} \\ \Bbbk^{V(G)} = \mathrm{span}_{\Bbbk}\{p(A_G)s_j : p \in \Bbbk[t],\ 1 \le j \le r\} \end{array} \right\}.$$

Proof. The distinct eigenvalue labels give Lagrange polynomials that project onto the individual eigenspace packets. Padding one basis vector from every packet into each of R seeds gives the upper

bound. Projection to the m_λ eigenspace coordinates is a surjective scalar quotient, so any generating family has at least m_λ members for every λ. The label λ_* gives the matching lower bound R. □

Boundary 17.2 (Input from spectral classification). For a finite undirected graph over $\mathbb{C}$, the spectral theorem for real symmetric matrices classically supplies a certificate of the displayed kind, and the numbers m_λ are the usual eigenvalue multiplicities. See [2]. The formal declaration verifies the formula for the number of seeds from the supplied eigenbasis and distinct eigenvalue labels. It does not derive diagonalizability from symmetry or implement an eigensolver.

17.1 Path graphs

For every field $\Bbbk$ and every $n \geq 1$, an endpoint of P_n is a cyclic vector for the adjacency operator. Consequently one is the least finite number of adjacency seeds:

$$1 = \min\left\{r \in \mathbb{N} : \exists s_1, \dots, s_r, \quad \Bbbk^n = \operatorname{span}_{\Bbbk}\{p(A_{P_n})s_j : p \in \Bbbk[t],\ 1 \leq j \leq r\}\right\}.$$

Over $\mathbb{C}$, the familiar spectral description has the distinct eigenvalues

$$\lambda_k = 2\cos\frac{k\pi}{n+1}, \qquad 1 \leq k \leq n,$$

and the normalized eigenvectors have endpoint coordinate proportional to

$$\sin\frac{k\pi}{n+1} \neq 0.$$

There is also a combinatorial proof valid over every field. Let

$$K_n = [e_1, Ae_1, \dots, A^{n-1}e_1].$$

Its (i, j) entry counts walks of length $j-1$ from the first vertex to vertex i. This entry is zero when $i > j$, while the diagonal entry is one because there is a unique shortest walk of length $i-1$ from vertex 1 to vertex i. Hence K_n is upper triangular with unit diagonal, so

$$\det K_n = 1 \qquad (n \geq 1).$$

Thus the endpoint is cyclic for every path size, not merely for the tested range.

17.2 Cycle graphs

Two adjacent vertex vectors generate the adjacency module of every simple cycle. Over $\mathbb{C}$, a Fourier quotient built from a primitive root of unity supplies a matching scalar obstruction of dimension two. Hence, for every $n \geq 3$, two is the least finite number of adjacency seeds:

$$2 = \min\left\{r \in \mathbb{N} : \exists s_1, \dots, s_r, \quad \mathbb{C}^n = \operatorname{span}_{\mathbb{C}}\{p(A_{C_n})s_j : p \in \mathbb{C}[t],\ 1 \leq j \leq r\}\right\}.$$

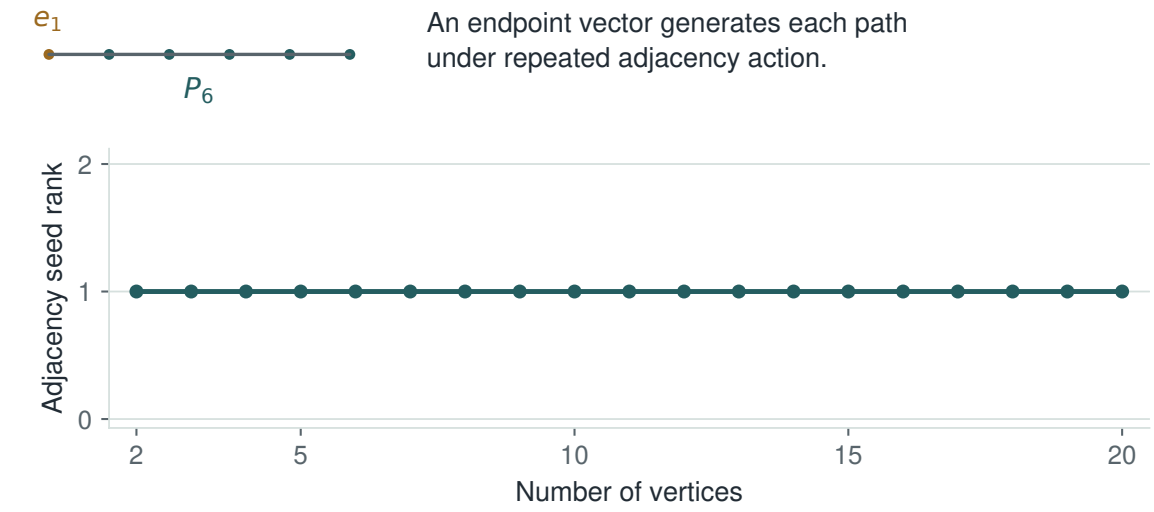


Figure 8. The endpoint Krylov certificate gives every nonempty path one adjacency seed.

To see the upper bound directly, index the vertices modulo n. The identity $Ae_j = e_{j-1} + e_{j+1}$ recovers successive vertex vectors from e_0 and e_1, over any field. For the lower bound over $\mathbb{C}$, let ζ be a primitive nth root of unity and define

$$q(x) = \left(\sum_{j=0}^{n-1} \zeta^j x_j, \sum_{j=0}^{n-1} \zeta^{-j} x_j \right).$$

Then $q(Ax) = (\zeta + \zeta^{-1}) q(x)$. The images of e_0 and e_1 are $(1, 1)$ and (ζ, ζ^{-1}), which are independent because $\zeta \neq \zeta^{-1}$ for $n \geq 3$. Thus q is a scalar quotient of dimension two, and proposition 4.13 gives the matching lower bound.

The usual spectral description writes the eigenvalues as

$$\lambda_k = 2 \cos \frac{2\pi k}{n}, \qquad 0 \leq k < n.$$

The values for k and $n - k$ coincide. Apart from the simple characters at $k = 0$ and, for even n, $k = n/2$, the remaining Fourier characters occur in pairs. Therefore the maximum eigenspace multiplicity is two, although some eigenspaces have multiplicity one, and the certified minimum above is two.

The path and cycle examples use nearly identical local operators. Their minimum finite seed numbers differ because the path has an endpoint Krylov basis, whereas the cycle has a Fourier quotient obstruction of dimension two.

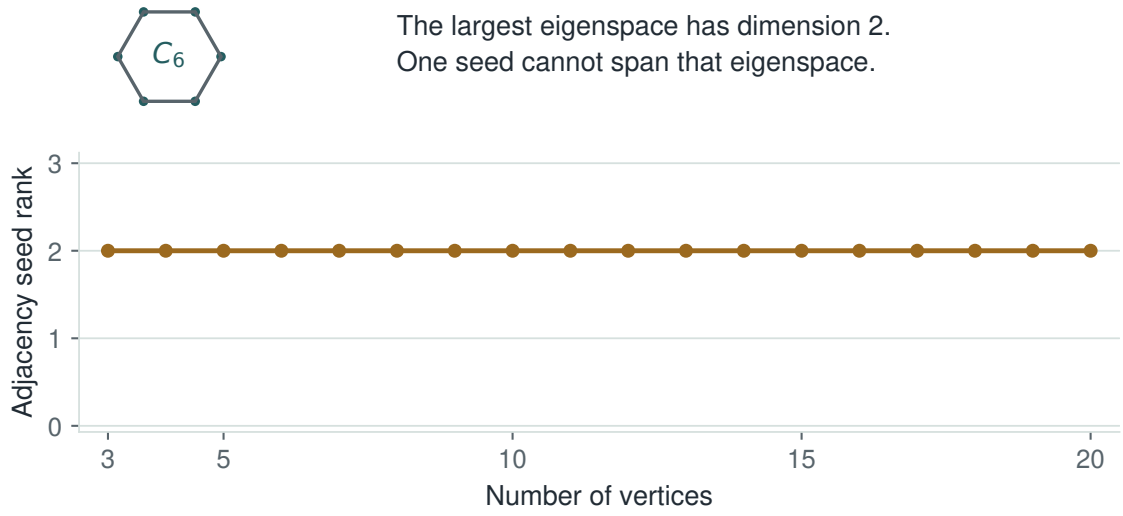


Figure 9. Cycle graphs have maximum eigenspace multiplicity two for the adjacency operator and therefore require two adjacency seeds, although one or two exceptional eigenspaces are simple.

18 The common operator language

The preceding examples can be written without changing the foundational notation.

Table 3. Foundational realizations of the common operator language.

Realization	Seed	Operator grammar	Packets	Completion or quotient
Coordinate ring	1	M_x, M_y	even/odd y degree	quotient by $y^2 - x^3 + x$
Shift system	δ_0	$(S_m)_{m\in\mathbb{Z}}$	residue classes of n	ℓ^2 closure
Full matrix module	$s \neq 0$	$M_n(\Bbbk)$	chosen idempotents	finite dimensional
Single operator	$s_1, \dots, s_r$	$\Bbbk[T]$	primary or spectral	finite dimensional
Graph adjacency	vertex seeds	$\Bbbk[A_G]$	eigenspaces	finite dimensional
Typed grammar	vertex seeds	path algebra $\Bbbk Q/I$	vertex idempotents	relation quotient

For a finite ordered seed family $S = (s_1, \dots, s_r)$, let $\varepsilon_S : \mathcal{A}^{\oplus r} \to X$ be the seed evaluation map of definition 3.1. Its kernel is the relation module. The free module modulo that kernel presents the generated hull by theorem 3.2. The invariant content is:

$$
\begin{aligned}
&\text{generated hull: } \operatorname{Hull}_{\mathcal{A}}(S),\\
&\text{seed complexity: } \operatorname{srank}_{\mathcal{A}}(X),\\
&\text{packet coordinates: } x \leftrightarrow (P_\alpha x)_\alpha,\\
&\text{operator coordinates: } T \leftrightarrow (P_\beta T P_\alpha)_{\beta,\alpha},\\
&\text{relation module: } N_S := \operatorname{Ker} \varepsilon_S \le \mathcal{A}^{\oplus r},\\
&\text{free presentation: } \mathcal{A}^{\oplus r}/N_S \cong \operatorname{Hull}_{\mathcal{A}}(S),\\
&\text{coordinate invariance: } P_\alpha \mapsto U P_\alpha U^{-1},\\
&\text{realization: } \Phi A = \varphi(A)\Phi.
\end{aligned}
$$

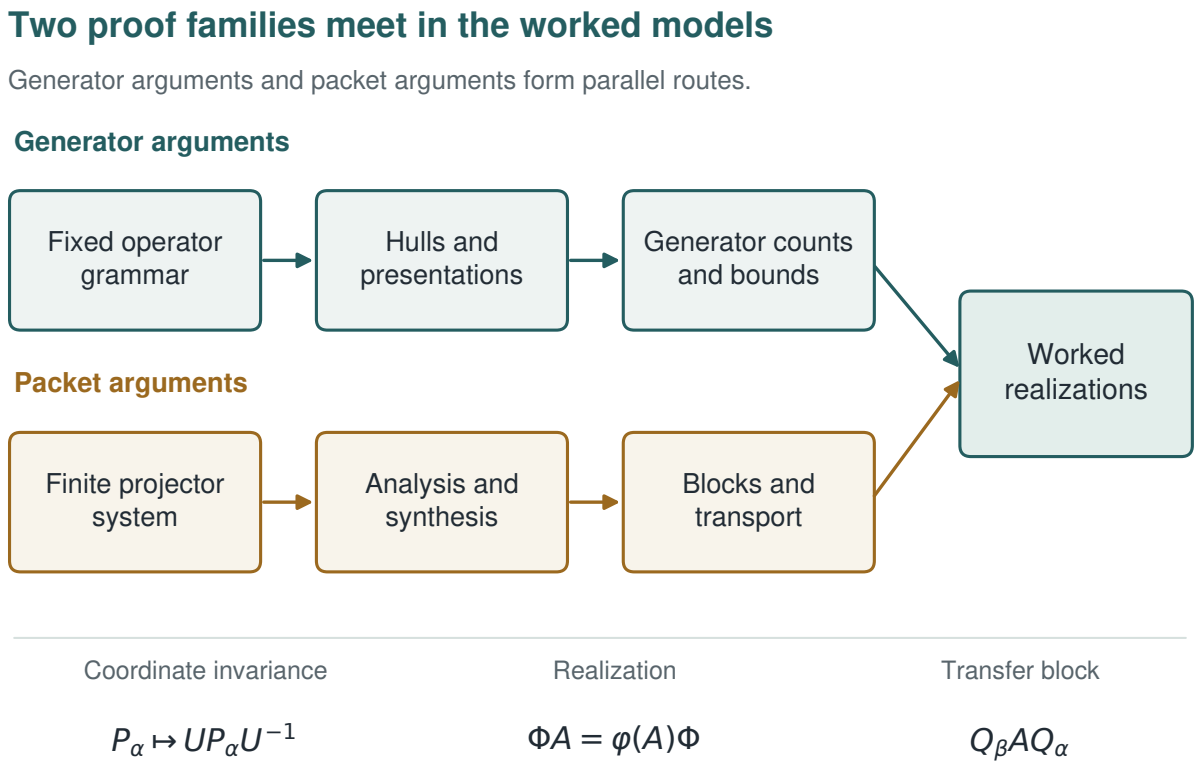


Figure 10. Structural overview of the two proof families. The generator and packet arguments meet in the worked realizations. This diagram is not an exhaustive graph of Lean declaration dependencies.

19 Computational certificates

The accompanying Python package is not used as a substitute for proof. It provides exact symbolic checks and finite numerical regression tests for formulas already proved. The computed ranges and random seeds are recorded explicitly. These checks do not extend a theorem beyond its hypotheses or certify an unrestricted numerical implementation. The journal companion also tests integer counts beyond the precision of floating point arithmetic and rejects the unsupported spectral shortcut for complex symmetric matrices.

19.1 Test suite

The test suite checks:

(i) projector idempotence, annihilation, and completeness.

(ii) exact vector analysis and synthesis.

(iii) exact operator reconstruction from all transfer blocks.

(iv) coordinate transport under a dense random similarity transform.

(v) cyclicity of a companion operator.

(vi) the obstruction that requires two seeds for repeated Jordan blocks.

(vii) the semisimple multiplicity formula in representative cases.

(viii) full orbit dimension for two representative nonzero seeds under the full matrix algebra.

(ix) exact reduction and packet multiplication in the elliptic coordinate ring.

(x) seed ranks of path and cycle adjacency modules over the reported finite ranges.

(xi) the cyclic residue transfer pattern of a finite shift.

19.2 Recorded results

The finite shift pattern recovered by the experiment is

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

which is the exact packet adjacency of a cyclic transfer among four phases.

19.3 Algorithms

The following finite constructions state mathematical specifications for the companion. Their Lean counterparts and exact scope are listed in the separate guide. They do not assert code extraction or a particular backend's performance.

Table 4. Representative finite checks recorded for the worked models. The computational companion distinguishes these records from its separately verified runs.

Quantity	Result
Projector idempotence residual	0
Projector annihilation residual	0
Projector completeness residual	0
Operator reconstruction residual	0
Residual for transported projectors, maximum	1.85×10^{-15}
Companion seed rank	1
Two equal Jordan blocks, seed rank	2
$M_3(\mathbb{C})$ acting on seven copies, seed rank	3
Sampled path graph seed ranks, $2 \le n \le 12$	1
Sampled cycle graph seed ranks, $3 \le n \le 12$	2
Full $M_4(\mathbb{R})$ orbit dimensions for two sampled nonzero seeds	$4, 4$
Automated tests	$33/33$ passed

Construction 19.1 (Closure specification for the orbit hull)**.** Let V have finite dimension, let $(O_i)_{i\in I}$ be a finite operator family, and let $S \subseteq V$. Define

$$W_0 = \operatorname{span}_{\Bbbk} S, \qquad W_{n+1} = W_n + \sum_{i\in I} O_i(W_n).$$

Then there is an $n \in \mathbb{N}$ such that

$$W_{n+1} = W_n \qquad \text{and} \qquad W_n = \operatorname{Hull}_{(O_i)}(S),$$

where the right side is the least subspace containing S and invariant under every O_i. Thus the ascending subspace iteration terminates and its stable value is exactly the finite operator hull.

Each strict enlargement increases dimension, so finite dimension prevents the iteration from continuing indefinitely. Equality at one step means that applying any declared operator keeps every vector in the current subspace. The stable subspace is therefore invariant, while every earlier step belongs to any invariant subspace that contains the seeds.

Construction 19.2 (Table specification for transfer blocks)**.** Given finite projector systems $(P_i)_{i\in I}$ and $(Q_j)_{j\in J}$ and a linear map T, form the table

$$B_{ji} = Q_j T P_i.$$

Its assembly reconstructs the input map exactly,

$$\sum_{j\in J}\sum_{i\in I} B_{ji} = T,$$

and its reported support relation is exactly $\{(i,j) : B_{ji} \neq 0\}$. The latter equality is the specification of the transfer graph output. It does not claim that Lean extracts a routine that decides the edge list for every coefficient field.

Construction 19.3 (Exact graph seed rank from a certificate)**.** For a finite square matrix A over a field K, define the noncomputable exact spectral quantity

$$r_{\mathrm{spec}}(A) = \max_{\mu \in \mathrm{Spec}_K(A)} \dim_K \ker(A - \mu I),$$

with value zero for an empty spectrum. Suppose an exact certificate supplies a finite eigenbasis indexed by labels and multiplicities, the eigenvalue equations, a multiplicity bound attained by one label, projector polynomials that extract the padded eigenbasis generators, coverage of every spectral value by a label, equality of the declared and actual eigenspace dimensions, and membership of the maximizing eigenvalue in the spectrum. Then $r_{\mathrm{spec}}(A)$ is the least natural number of seeds under polynomial orbits for A.

This is a verified certificate checker/specification. It does not construct the eigenbasis, prove that a symmetric matrix is diagonalizable, or implement an exact eigensolver. The companion Python code performs symbolic computations in the reported finite graph range. The path and cycle results for every size are proved separately above.

20 Reading guide: common confusions and diagnostic questions

20.1 Common confusions

Do not conflate	Correct distinction
Grammar with realized algebra	The grammar may contain relations or words that act as zero. Seed rank depends only on the image action.
Seed with basis vector	A seed is a generator under an algebra action. One seed may generate a module with large dimension.
Cyclic vector with eigenvector	A cyclic vector must generate the whole module under operator words. It is usually not an eigenvector.
Packet with orthogonal subspace	A packet is the range of an idempotent in a complete annihilating family. Orthogonality is optional.
Exact reconstruction with good conditioning	Algebraic inverse identities may coexist with large analysis and synthesis norms.
Nonzero transfer edge with surjective transfer	A nonzero block can have small rank. Graph reachability alone does not prove generation.
Seed rank for single packet seeds with unrestricted seed rank	Seeds supported in several packets may meet several source components simultaneously.
Continuous surjection with quotient isomorphism	The target topology must agree with the quotient topology.
Repeated notation with semantic realization	A true realization must preserve a declared collection of operations and identities.

20.2 Diagnostic questions for a new application

Before claiming that a new subject is a realization of Chapter I, answer the following.

1. What is the fixed operator grammar or acting algebra?
2. What is the concrete realization space or module?
3. Which seeds are declared before the target is chosen?
4. Is generation algebraic, topological, probabilistic, or approximate?
5. What are the idempotent packets, and in what sense do they sum to the identity?
6. Are the packets finite, countable unconditional, or merely formal?
7. What are the transfer blocks, and which blocks are known to vanish?
8. Which morphisms preserve the packet system?
9. Which quotient, completion, tensor, or limit operations must be preserved?
10. What exact theorem from Chapter I is being specialized, and what new domain theorem remains to be proved?

20.3 Theorem spine at a glance

free seed presentation	packet idempotents
$\Downarrow$	$\Downarrow$
minimal generator number	exact analysis/synthesis
$\Downarrow$	$\Downarrow$
reduction modulo the Jacobson radical	transfer block algebra
$\Downarrow$	$\Downarrow$
semisimple or primary computation	graphs, quotients, and realization

21 Notation and convention ledger

Address.
An operator word applied to a seed.

Analysis.
The map sending an object to all packet coordinates.

Block support.
The set of packet pairs (β, α) for which $P_\beta T P_\alpha \neq 0$.

Collapsed realization.
The direct sum of all typed vertex spaces.

Condensation graph.
The acyclic graph obtained by collapsing strongly connected components.

Generated hull.
The submodule generated by a seed family, or its declared closure.

Grammar.
A typed quiver and relation ideal describing legal operator words.

Packet.
The range of one idempotent in a complete annihilating family.

Packet pure.
Supported in at most one packet.

Preservation profile.
The operations a realization functor is required to preserve.

Realized algebra.
The image of the grammar algebra in the endomorphism algebra of a realization.

Seed rank.
The smallest cardinality of a generating family. For a finite value, this is the minimum number of seeds.

Source component.
A strongly connected component with no incoming edge in the condensation graph.

Synthesis.
The map adding packet coordinates into the original object.

Transfer block.
The map from one packet space to another obtained by sandwiching an operator between packet projectors.

Transfer graph.
The directed graph formed from nonzero blocks of a declared algebra generating family.

> **How to use the chapter after the first reading**
>
> For a concrete calculation, first reduce the action to the realized algebra. Next compute or bound seed rank using quotients, the module top, or primary decomposition. Then declare the packet system and use exact analysis and synthesis. Finally move all operator equations into transfer blocks, where support, graph, quotient, and functorial constraints become explicit.

22 Proof dependency spine

Canonical record	Immediate proof dependencies
PSC-I-T-TYPED-MODULE-EQUIVALENCE	Typed grammar and path algebra action.
PSC-I-T-FREE-PRESENTATION	Seed evaluation map and first isomorphism theorem.
PSC-I-T-COMPLETED-PRESENTATION	Free presentation and uniqueness of completion.
PSC-I-T-QUOTIENT-MONOTONICITY	Intertwining identity and generated hull.
PSC-I-T-EXACT-SEQUENCE-BOUNDS	Quotient monotonicity and lifting generators.
PSC-I-T-MODULE-TOP	Nakayama lemma.
PSC-I-T-SEMISIMPLE-SEED-RANK	Module top theorem and regular module multiplicities.
PSC-I-T-SINGLE-OPERATOR	Primary decomposition, local formula, and product action.
PSC-I-T-EXACT-PACKET-COORDINATES	Complete finite idempotent family.
PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS	Completeness of the unconditional coordinate space.
PSC-I-T-BLOCK-ALGEBRA	Reconstruction and block multiplication.
PSC-I-T-SUPPORT-PATTERN-ALGEBRA	Block multiplication and rank one necessity tests.
PSC-I-T-PREORDER-ALGEBRA-DECOMPOSITION	Preorder support criterion and finite chain nilpotence.
PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT	Generator word packet expansion.

Canonical record	Immediate proof dependencies
PSC-I-T-SOURCE-COMPONENT-SEED-RANK	Reachability obstruction and edge surjective propagation.
PSC-I-T-NATURALITY	Intertwining and packet aggregation.
PSC-I-T-PACKET-EXACT-SEQUENCE	Invariant packet projectors and first isomorphism theorem.
PSC-I-T-DENSE-REALIZATION	Continuity and naturality of hulls.
PSC-I-T-REALIZATION-EXPORT	Additive functor axioms and preserved universal constructions.

23 Conclusions and further use

The results establish the following tools for working with a declared operator action and packet system:

I.1. The two sided annihilator criterion is proved in proposition 3.5.

I.2. Generator detection modulo the Jacobson radical is proved in theorem 5.2.

I.3. The semisimple seed rank formula applies over arbitrary division rings, with the conventions for left and right modules kept explicit in theorem 5.6.

I.4. The single operator formula and its cyclic criteria apply over arbitrary fields in theorem 6.2 and corollary 6.5.

I.5. The countable unconditional packet theorem is proved in theorem 7.16.

I.6. The classification of support compatible with $*$ is proved in corollary 8.14.

I.7. The weighted source component formula gives the packet pure generation requirement under the propagation hypotheses in theorem 9.7.

I.8. Seed system composition, packet exact sequences, quotient block descent, dense realization, quotient completion, and semantic theorem transport are proved in proposition 10.5, theorems 10.6, 10.10, 10.11 and 11.2, and corollary 10.8.

I.9. Exact sequence bounds, monotonicity under enlargement of the acting algebra, coordinate stability, projection onto diagonal blocks, and preorder decomposition are established in theorems 4.6 and 8.15 and propositions 4.3, 7.12 and 8.11.

A Stable statement concordance

These identifiers preserve the expanded foundation's namespace of 108 records. The display numbers refer to Chapter I. The identifiers track mathematical statements, and their count does not measure how many theorems have been verified in Lean. Statements in the worked models retain their source labels and are mapped separately by the Lean guide.

Stable identifier	No.	Statement
PSC-I-D-COEFFICIENT-REGIME	1.1	Coefficient and module regimes
PSC-I-D-THEOREM-RECORD	1.2	Canonical theorem record
PSC-I-B-SYMBOLIC-SCOPE	1.3	Symbolic scope
PSC-I-D-TYPED-GRAMMAR	2.1	Typed operator grammar
PSC-I-D-COLLAPSED-REALIZATION	2.2	Collapsed one space realization
PSC-I-T-TYPED-MODULE-EQUIVALENCE	2.3	Typed realization and module equivalence
PSC-I-D-REALIZED-ALGEBRA	2.4	Realized operator algebra
PSC-I-P-ACTION-FACTORIZATION	2.5	Action factorization and hull invariance
PSC-I-D-SEED-SYSTEM	2.6	Linear seed system
PSC-I-D-GENERATED-HULL	2.7	Generated hull
PSC-I-D-NATURAL-PRESENTATION	2.8	Natural presentation
PSC-I-P-ALGEBRA-MODULE-GENERATION	2.9	Algebra generation and module generation are distinct
PSC-I-B-TARGET-DEPENDENT-OPERATORS	2.10	Target dependent operators
PSC-I-D-SEED-EVALUATION	3.1	Seed evaluation map
PSC-I-T-FREE-PRESENTATION	3.2	Finite free presentation
PSC-I-C-MINIMAL-FREE-COVER	3.3	Minimal free cover characterization
PSC-I-C-ONE-SEED-PRESENTATION	3.4	One seed presentation
PSC-I-P-ANNIHILATOR-TWO-SIDED	3.5	Two sided annihilator criterion
PSC-I-T-COMPLETED-PRESENTATION	3.6	Completion of a dense free presentation
PSC-I-D-SEED-RANK	4.1	Seed rank
PSC-I-P-CYCLIC-CHARACTERIZATION	4.2	Cyclic characterization
PSC-I-P-ACTING-ALGEBRA-MONOTONICITY	4.3	Monotonicity under enlargement of the acting algebra
PSC-I-T-QUOTIENT-MONOTONICITY	4.4	Quotient monotonicity
PSC-I-C-ISOMORPHISM-INVARIANCE	4.5	Isomorphism invariance
PSC-I-T-EXACT-SEQUENCE-BOUNDS	4.6	Short exact sequence bounds
PSC-I-P-SHARED-DIRECT-SUM	4.7	Shared direct sum bounds
PSC-I-T-INDEPENDENT-DIRECT-SUM	4.8	Independent direct sum formula
PSC-I-P-TENSOR-BOUND	4.9	Tensor product bound
PSC-I-B-TENSOR-NONEQUALITY	4.10	Tensor rank is not generally multiplicative
PSC-I-P-SCALAR-EXTENSION-MONOTONICITY	4.11	Scalar extension monotonicity
PSC-I-D-SEED-RANK-WITNESS	4.12	Seed rank witness
PSC-I-P-SCALAR-QUOTIENT-WITNESS	4.13	Scalar quotient witness
PSC-I-CI-NAKAYAMA	5.1	Nakayama lemma
PSC-I-T-MODULE-TOP	5.2	Generator number is detected modulo the Jacobson radical
PSC-I-C-LOCAL-ALGEBRA-FORMULA	5.3	Local algebra formula
PSC-I-CI-WEDDERBURN-ARTIN	5.4	Wedderburn and Artin decomposition
PSC-I-L-REGULAR-MODULE-MULTIPLICITIES	5.5	Regular module multiplicities
PSC-I-T-SEMISIMPLE-SEED-RANK	5.6	Semisimple seed rank formula over arbitrary division rings
PSC-I-C-ALGEBRAICALLY-CLOSED-FORMULA	5.7	Algebraically closed multiplicity formula

Stable identifier	No.	Statement
PSC-I-C-FULL-MATRIX	5.8	Full matrix algebra
PSC-I-C-MULTIPLICITY-OBSTRUCTION	5.9	Multiplicity obstruction
PSC-I-CI-PID-PRIMARY-DECOMPOSITION	6.1	Primary decomposition for polynomial modules
PSC-I-T-SINGLE-OPERATOR	6.2	Single operator seed rank
PSC-I-C-INVARIANT-FACTOR-COUNT	6.3	Invariant factor count
PSC-I-C-ALGEBRAICALLY-CLOSED-SINGLE-OPERATOR	6.4	Geometric multiplicity formula
PSC-I-C-CYCLIC-MATRIX	6.5	Cyclic matrix criterion
PSC-I-D-PACKET-SYSTEM	7.1	Finite packet system
PSC-I-D-PACKET-SUPPORT	7.2	Packet support and packet pure vectors
PSC-I-L-PACKET-AGGREGATION	7.3	Finite packet aggregation
PSC-I-D-ANALYSIS-SYNTHESIS	7.4	Packet analysis and synthesis
PSC-I-T-EXACT-PACKET-COORDINATES	7.5	Exact packet coordinate isomorphism
PSC-I-C-PACKETWISE-EQUALITY	7.6	Packetwise equality
PSC-I-C-PACKET-UNIVERSAL-PROPERTY	7.7	Universal mapping property of finite packet coordinates
PSC-I-T-TENSOR-PACKET-SYSTEM	7.8	Tensor product packet systems
PSC-I-P-DUAL-PACKET-SYSTEM	7.9	Dual packet systems
PSC-I-T-BANACH-PACKET-ISOMORPHISM	7.10	Finite Banach packet isomorphism
PSC-I-D-PACKET-CONDITION-NUMBER	7.11	Packet coordinate condition number
PSC-I-P-CONDITIONED-COORDINATES	7.12	Conditioned coordinate transport
PSC-I-C-COMPLETION-PACKET-SPLIT	7.13	Completion preserves a finite packet split
PSC-I-D-UNCONDITIONAL-PACKET-SPACE	7.14	Unconditional packet sum space
PSC-I-L-UNCONDITIONAL-COMPLETENESS	7.15	Completeness of the unconditional packet sum
PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS	7.16	Countable unconditional packet isomorphism
PSC-I-R-UNIFORM-BOUNDEDNESS	7.17	Uniform boundedness
PSC-I-B-INFINITE-PACKET-TOPOLOGY	7.18	Infinite packet sums
PSC-I-D-TRANSFER-BLOCK	8.1	Transfer block
PSC-I-T-OPERATOR-RECONSTRUCTION	8.2	Exact operator reconstruction
PSC-I-T-BLOCK-MULTIPLICATION	8.3	Packet matrix multiplication
PSC-I-C-BLOCK-POWERS	8.4	Packet powers and paths
PSC-I-C-ADJOINT-REVERSAL	8.5	Adjoint reversal
PSC-I-C-TENSOR-TRANSFER-BLOCKS	8.6	Tensor product transfer blocks
PSC-I-T-BLOCK-INTERTWINING	8.7	Blockwise intertwining equations
PSC-I-T-BLOCK-ALGEBRA	8.8	Full block algebra representation
PSC-I-C-PACKET-SUPPORT-PROPAGATION	8.9	Propagation of packet support
PSC-I-C-DIMENSION-COUNT	8.10	Dimension count
PSC-I-P-BLOCK-DIAGONAL-COMMUTANT	8.11	Block diagonal projection and packet commutant
PSC-I-D-SUPPORT-RELATION	8.12	Support relation and support space
PSC-I-T-SUPPORT-PATTERN-ALGEBRA	8.13	Support pattern algebra criterion
PSC-I-C-SUPPORT-PATTERN-STAR	8.14	Bounded support patterns closed under $*$
PSC-I-T-PREORDER-ALGEBRA-DECOMPOSITION	8.15	Preorder support algebra decomposition
PSC-I-D-TRANSFER-GRAPH	9.1	Transfer graph of an algebra generating family
PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT	9.2	Reachability support of the generated algebra
PSC-I-P-REACHABILITY-OBSTRUCTION	9.3	Reachability obstruction for generated hulls
PSC-I-D-DIRECTED-SEED-RANK	9.4	Packet pure seed rank
PSC-I-P-PACKET-PURE-RANK-BOUNDS	9.5	Basic packet pure rank bounds
PSC-I-D-EDGE-SURJECTIVE	9.6	Edge surjective packet system

Stable identifier	No.	Statement
PSC-I-T-SOURCE-COMPONENT-SEED-RANK	9.7	Source component formula for packet pure seed rank
PSC-I-D-PACKET-PROPAGATING	9.8	Packet propagating system
PSC-I-C-SOURCE-COMPONENT-COUNT	9.9	Unweighted source component count
PSC-I-B-PACKET-PURE-BOUNDARY	9.10	Mixed support seeds can merge source requirements
PSC-I-T-COORDINATE-TRANSPORT	10.1	Coordinate transport
PSC-I-D-PACKET-MORPHISM	10.2	Packet compatible operator morphism
PSC-I-D-SEED-MORPHISM	10.3	Morphism of seeded systems
PSC-I-T-NATURALITY	10.4	Naturality of generated hulls and transfer blocks
PSC-I-P-SEED-SYSTEM-CATEGORY	10.5	Category of seeded packet systems
PSC-I-T-PACKET-EXACT-SEQUENCE	10.6	Packet exact sequence for an invariant subspace
PSC-I-T-QUOTIENT-PACKET-DESCENT	10.7	Packet descent through an invariant quotient
PSC-I-C-QUOTIENT-PACKET-DESCENT	10.8	Restriction and quotient of transfer blocks
PSC-I-C-TOPOLOGICAL-PACKET-EXACT-SEQUENCE	10.9	Topological packet exact sequence
PSC-I-T-DENSE-REALIZATION	10.10	Dense realization theorem
PSC-I-T-QUOTIENT-TOPOLOGY-COMPLETION	10.11	Quotient topology and completed realization
PSC-I-B-TOPOLOGICAL-QUOTIENT	10.12	Continuous surjections and quotient topology
PSC-I-D-REALIZATION-FUNCTOR	11.1	Realization functor and preservation profile
PSC-I-T-REALIZATION-EXPORT	11.2	Equational realization principle
PSC-I-P-REALIZATION-EXPORT	11.3	Realization export principle
PSC-I-D-SEMANTIC-REALIZATION	11.4	Semantic realization criterion
PSC-I-ALG-EXAMPLE-SLOTS	12.1	Worked realization interface
PSC-I-EXPORT	12.2	Chapter I export object
PSC-I-B-FINAL-CLAIM-BOUNDARY	12.3	Exact limits of Chapter I

B Visual atlas

The following twenty-two plates present exact finite models, explanatory diagrams, and one analytic density example. Each plate follows a calculation, identity, or obstruction from the main text. All graphics are generated from mathematical source, and the cross-references refer to the mathematical text.

Plate	Route through the examples
1, 2	Fixed grammars, seed power, and the objects of the calculus.
3, 4	Two linked chains and a quotient proving the minimum seed count.
5, 6	Left ideal relations and typed reachability with a missing direction.
7, 8	Oblique exact coordinates and complete transfer block multiplication.
9, 10	Cancellation along paths and the algebra of admissible zero patterns.
11, 12	Coordinate transport, shared action, and independent control.
13, 14	Semisimple multiplicity budgets and primary extraction polynomials.
15, 16	Path endpoints and the cycle's two dimensional quotient obstruction.
17	A coordinate ring relation as a two packet product rule.
18, 19	Dense shift generation, residue coordinates, and address carry.
20, 21	Exact orbit hull stabilization and all tensor cross terms.
22	The scope of mathematical, formal, and computational evidence.

The seed examples can be read consecutively. The packet and graph plates also provide entry points for readers interested in coordinates or propagation. The displayed matrices are exact. Numerical illustrations in the earlier figures retain their explicitly stated finite ranges. Each diagram is accompanied by the equation or obstruction it represents, so it can be read alongside the relevant proof.

Plate I.1 / CHAPTER I / A VISUAL APPENDIX OF WORKED EXAMPLES

What can one seed actually do?

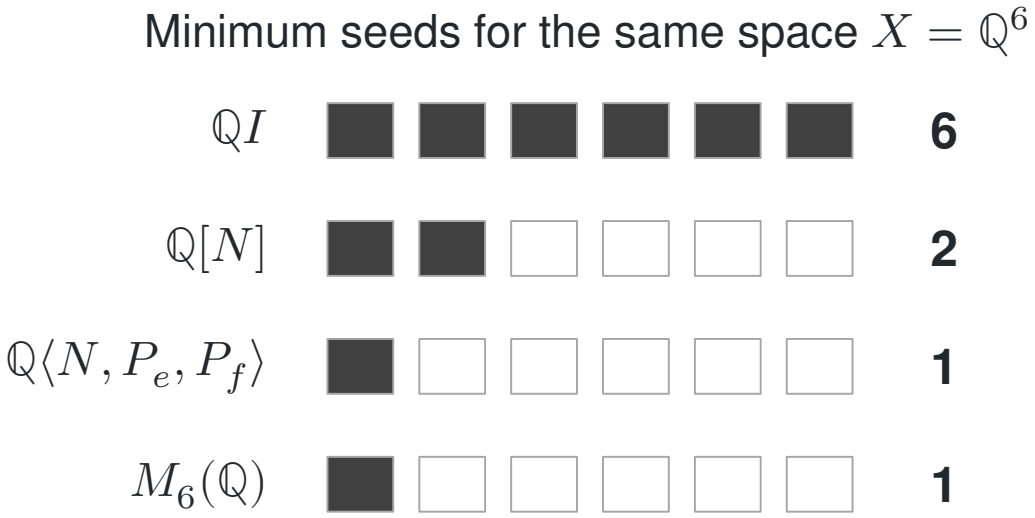
Main text: definition 4.1, proposition 4.3, and theorem 7.5

A seed does not carry a fixed amount of generative power by itself. The answer depends on the operators that are allowed to act on it. Chapter I makes that dependence part of the mathematical object rather than leaving it implicit.

Use the same space of dimension six throughout the first comparison:

$$X = \mathbb{Q}^6, \qquad N = J_3(0) \oplus J_3(0), \qquad s = e_0 + f_0.$$

Here each $J_3(0)$ moves a basis vector forward along a chain of three basis vectors. The projectors P_e, P_f select the two chains. Their formulas and the proofs of the minimum seed counts appear on Plate I.3 and Plate I.4.



A count of minimum generators, not a timing comparison. The ambient space is unchanged. Enlarging the acting algebra changes what one seed can separate.

Three questions to keep apart

Generation	Can the fixed grammar reach every vector by finite linear combinations of operator words?
Coordinates	Can every vector be split into packets and then reconstructed exactly?
Completion	When finite words do not suffice, does their span become dense in the specified topology?

A packet projector can be available as a way to *inspect* a vector without belonging to the *generation* grammar. Admitting it as a new primitive can lower seed rank. That distinction explains the second and third rows of the figure.

> **Read this as:** The useful question is not "How powerful is this seed?" It is "What does this seed generate under this particular, previously fixed algebra of operators?"

Plate I.2 / ORIENTATION / SYMBOLS WITH JOBS

An atlas of the objects

Main text: definitions 2.1, 2.6, 4.1, 7.1 and 8.1

Most of the chapter can be read by tracking a small collection of objects. The table distinguishes their mathematical roles and names a concrete instance used in this appendix.

Object	What it means	Instance here
X	The ambient vector space or module. It specifies the objects that might be generated.	$\mathbb{Q}^6$ with two chains of length three.
$S \subseteq X$	The selected seed family. A family can contain vectors with components in several packets.	$\{e_0 + f_0\}$ or $\{e_0, f_0\}$.
$\mathcal{A}$	The fixed unital algebra of admissible operators. It is not chosen anew for each target.	$\mathbb{Q}[N]$ versus $\mathbb{Q}\langle N, P_e, P_f\rangle$.
$\mathrm{Hull}_{\mathcal{A}}(S)$	The smallest invariant linear subspace containing the seeds.	$\mathrm{span}\{e_j + f_j : 0 \le j \le 2\}$.
P_α	A finite family of pairwise annihilating idempotents whose sum is I.	Chain projections, oblique projections, or residue masks.
X_α	The packet subspace $P_\alpha X$. It can have dimension greater than one.	One chain of dimension three.
$T_{\beta\alpha}$	The block $P_\beta T P_\alpha$: input α, output β.	A numerical entry when each coordinate packet has dimension one.
F/I_s	A module of operator words modulo relations that kill the seed.	A quotient of left modules. It need not be a quotient algebra.
U	An invertible change of realization coordinates.	A shear that makes orthogonal packets oblique.
Φ	An intertwiner that need not be invertible.	Projection from two chains to the first chain.
τ	The topology used when taking the closure of an algebraic hull.	The ℓ^2 norm for a bilateral shift.

Read products from right to left

In $P_\beta T P_\alpha x$, first select the input packet, then apply T, then select the output packet. In a matrix, this is why the output label belongs to the row and the input label to the column.

Where to begin

For seed rank, start with Plate I.3. For exact reconstruction, start with Plate I.7. For graphs, use Plate I.15 and Plate I.16. For completion in infinite dimensions, use Plate I.18 and Plate I.19. The scope of the formal proofs is collected on Plate I.22.

Plate I.3 / MODEL A / AN OPERATOR IN SIX DIMENSIONS

Two chains can stay linked

Main text: theorem 6.2. Matrices over $\mathbb{Q}$

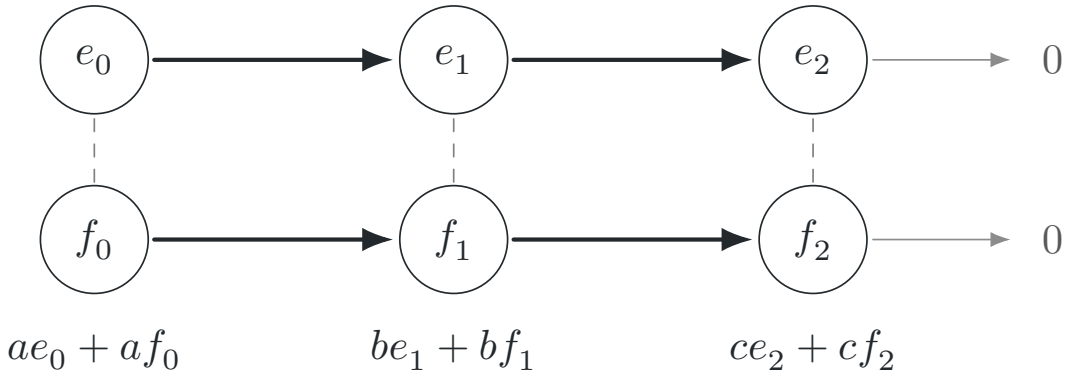
Use the ordered basis $(e_0, e_1, e_2, f_0, f_1, f_2)$. Define

$$Ne_0 = e_1, \quad Ne_1 = e_2, \quad Ne_2 = 0, \qquad Nf_0 = f_1, \quad Nf_1 = f_2, \quad Nf_2 = 0.$$

Starting with $s = e_0 + f_0$, the two chains advance together:

$$s = e_0 + f_0, \qquad Ns = e_1 + f_1, \qquad N^2 s = e_2 + f_2, \qquad N^3 s = 0.$$

$$Ne_j = e_{j+1}, \qquad Nf_j = f_{j+1} \quad (j = 0, 1)$$



The vertical dotted pairings record equal coefficients, not an additional allowed operator. Every polynomial in N uses the same coefficients on both chains.

Since $N^3 = 0$, every polynomial orbit vector has the form

$$p(N)s = a(e_0 + f_0) + b(e_1 + f_1) + c(e_2 + f_2).$$

The generated hull therefore has dimension three, even though the seed has a nonzero component in each packet of dimension three. The Krylov ranks are $1, 2, 3, 3, 3, 3$.

What changes when a projector becomes admissible?

Let P_e keep the coordinates along the e chain and P_f those along the f chain. If these are merely measurement maps, the preceding hull is unchanged. If they become allowed generators, then

$$P_e s = e_0, \qquad P_f s = f_0,$$

and powers of N now recover all six basis vectors. The same seed becomes cyclic for the larger algebra $\mathbb{Q}\langle N, P_e, P_f\rangle$.

> **Read this as:** Nonzero components in every packet are not enough. One must also be able to separate the independent directions inside and across those packets using the allowed operations.

Plate I.4 / MODEL A / A CERTIFICATE FOR THE LOWER BOUND

A quotient proves two seeds are necessary

Main text: proposition 4.13 and theorem 6.2

The failure of the particular seed on Plate I.3 does not prove that *every* single seed fails. A certificate for the lower bound does.

Define a surjective linear map

$$q : X \longrightarrow \mathbb{Q}^2, \qquad q\Big(\sum a_j e_j + \sum b_j f_j\Big) = (a_0, b_0).$$

The quotient keeps only the two chain heads

e_0	e_1	e_2	f_0	f_1	f_2
1	0	0	0	0	0
0	0	0	1	0	0

$$q(a_0, a_1, a_2, b_0, b_1, b_2) = (a_0, b_0)$$

An actual 2×6 quotient matrix. It forgets the advanced positions in both chains and retains two independent head coordinates.

Because $qN = 0$, one has, for every polynomial p,

$$q(p(N)x) = p(0)q(x).$$

Thus every operator acts as a *scalar* on the quotient. The image of the orbit of one seed stays on one line in $\mathbb{Q}^2$. It cannot span this quotient of dimension two.

Lower and upper bounds meet

If r seeds generated X, their images would generate the quotient. Since scalar multiplication cannot create a new independent direction, $r \geq 2$. Conversely, the two seeds e_0, f_0 generate their respective chains, so

$$\boxed{\operatorname{srank}_{\mathbb{Q}[N]}(X) = 2.}$$

This proof uses the displayed operator and quotient. It does not ask a computer to search every possible seed, and it does not need an existence theorem for Jordan form.

A general pattern worth recognizing

A surjective intertwiner $q : X \twoheadrightarrow Y$ turns any generating family of X into a generating family of Y. Therefore a quotient on which the grammar loses mixing power can certify a lower bound in the original, much larger space.

> **Read this as:** To rule out one seed, find a quotient in which its entire orbit is forced onto one line while the quotient still has at least two independent directions.

Plate I.5 / PRESENTATIONS / WHY A LEFT IDEAL APPEARS

Relations identify words, not just outputs

Main text: theorem 3.2, corollary 3.4, and proposition 3.5

Let $F = \Bbbk\langle z_1, \dots, z_r\rangle$ be the algebra of noncommuting operator words. Evaluation at a fixed seed is

$$\Phi_s : F \to X, \qquad f \mapsto f(T_1, \dots, T_r)s.$$

Two words represent the same generated vector exactly when their difference lies in $I_s = \operatorname{Ker} \Phi_s$. Consequently

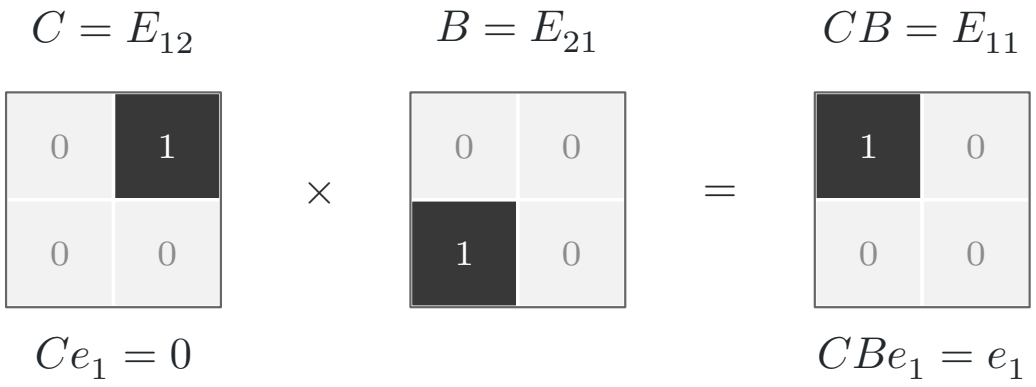
$$F/I_s \cong \operatorname{Hull}_F(s)$$

as *left modules*. The relation $f(T)s = 0$ implies $g(T)f(T)s = 0$ for every word g, which explains closure under multiplication on the left.

The distinction is visible in two dimensions

Use $X = \mathbb{Q}^2$, seed $e_1 = (1, 0)^{\mathsf{T}}$, and the full matrix algebra. Its seed annihilator is the set of matrices with zero first column:

$$I_{e_1} = \left\{ \begin{pmatrix} 0 & a \\ 0 & b \end{pmatrix} : a, b \in \mathbb{Q} \right\}.$$



The relation $Ce_1 = 0$ is destroyed by multiplying C on the right by B. The product CB sends the seed to itself.

For arbitrary D, $(DC)e_1 = D(Ce_1) = 0$. But $(CB)e_1 = e_1 \neq 0$. Thus I_{e_1} is a left ideal, but it is not stable under multiplication on the right. The quotient retains the action on generated vectors. Calling it a ring would lose the distinction that makes the presentation correct.

Several seeds give a module of relations

For $(s_1, \dots, s_m)$, evaluation is $(f_1, \dots, f_m) \mapsto \sum_j f_j(T)s_j$. Its kernel is a submodule of F^m and can include relations *between* seeds, not only relations annihilating each seed separately.

> **Read this as:** A presentation says which operator expressions are identified in a particular realization. A seed annihilator is not automatically an algebra relation ideal.

Plate I.6 / TYPED MODEL / $U = \mathbb{Q}^2$ AND $V = \mathbb{Q}^3$

A reached type can still hide a direction

Main text: proposition 13.1 and definition 9.4

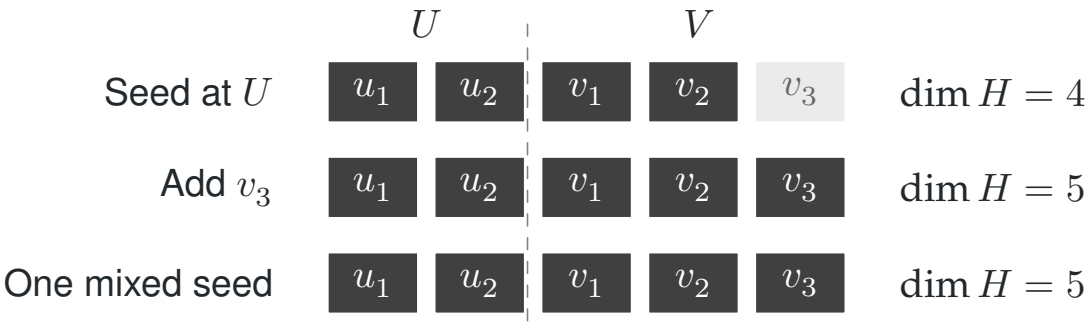
The chapter's typed model has a rotation $R : U \to U$, an inclusion $A : U \to V$, and a retraction $B : V \to U$:

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

The typed relation is $BA = I_U$. In the opposite order,

$$AB = \operatorname{diag}(1, 1, 0) \neq I_V.$$

The missing direction is $\operatorname{Ker} B = \mathbb{Q}v_3$, not an entirely missing vertex.



Dark slots indicate independent directions in the generated hull. The same two vertex types are present in every row, but their generated subspaces differ.

One seed supported at U

The vectors u_1, Ru_1 span U. Their images under A span $\mathbb{Q}v_1 + \mathbb{Q}v_2$. Every path from U to V ends with A, so it cannot supply v_3. The hull has dimension four. Adding the pure seed v_3 gives all five dimensions.

Why this does not prove an unrestricted minimum of two

The bound path algebra includes the vertex idempotents P_U, P_V. A mixed seed

$$s = u_1 + v_3 \in U \oplus V$$

satisfies $P_U s = u_1$ and $P_V s = v_3$. It therefore generates the entire representation by itself. The chapter proves that no seed *supported at* U is enough and that two displayed pure seeds suffice. It does not claim that two arbitrary mixed seeds are necessary.

> **Read this as:** Keep three notions separate: reaching a vertex, filling its entire vector space, and minimizing seeds when each seed must lie in a single packet. They answer different questions.

Plate I.7 / PACKET COORDINATES / AN OBLIQUE SPLITTING

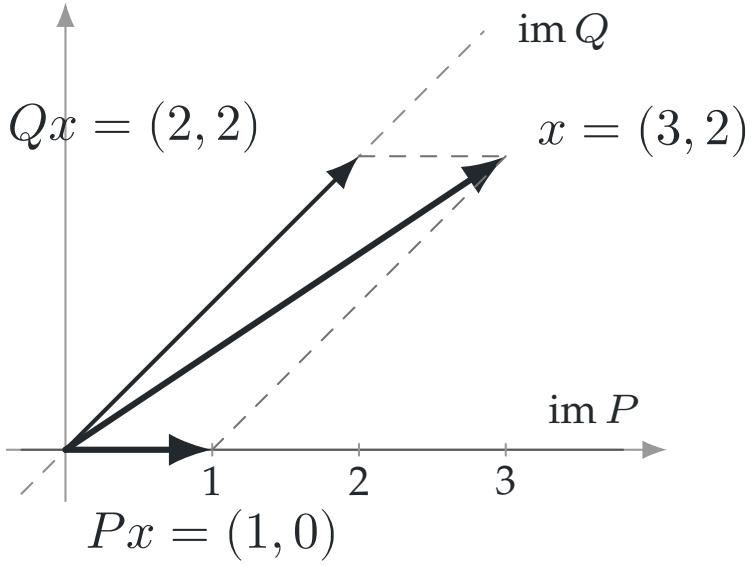
Exact does not mean orthogonal

Main text: theorem 7.5 and corollary 8.5

In $\mathbb{R}^2$, take

$$P = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}, \qquad Q = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

These maps satisfy $P^2 = P$, $Q^2 = Q$, $PQ = QP = 0$ and $P + Q = I$. Their images are the horizontal line and the diagonal line, respectively. They are not self-adjoint in the usual Euclidean inner product.



Actual Euclidean vector geometry for $x = (3, 2)$. The parallelogram reconstructs x from $Px = (1, 0)$ and $Qx = (2, 2)$. Neither component is an orthogonal projection of x.

Analysis records (Px, Qx) and synthesis adds the two vectors. For a packet tuple $(u, v) \in \operatorname{im} P \times \operatorname{im} Q$,

$$P(u + v) = u, \qquad Q(u + v) = v.$$

Both inverse identities follow from annihilation and completeness, without any angle condition.

What the algebra does not preserve

For the displayed x,

$$\|x\|^2 = 13, \qquad \|Px\|^2 + \|Qx\|^2 = 1 + 8 = 9.$$

There is no Parseval identity here. The cross term is nonzero because the packet directions are not perpendicular. Likewise, the simple formula for reversing adjoints of packet blocks requires self-adjoint projectors. It is not justified for this oblique pair.

> **Read this as:** Exact reconstruction is an algebraic inverse statement. Energy preservation and adjoint identities require additional geometric hypotheses.

Plate I.8 / OPERATOR COORDINATES / MULTIPLICATION AS A SUM

A transfer block has an input and an output

Main text: theorems 8.2 and 8.3 and corollary 8.4

Use the three coordinate packets of dimension one in $\mathbb{Q}^3$. The block $T_{ij} = P_i T P_j$ sends input packet j into output packet i. For this example, its nonzero scalar is the (i, j) matrix entry.

T

1	2	0
0	1	3
4	0	1

input column; output row

$T \cdot T$

T^2

1	4	6
12	1	6
8	8	1

sum over intermediate packets

All displayed entries are exact integers. The square is computed from the operator, not from a thresholded graph or a matrix of block norms.

Reconstruction retains every block

The identity $P_1 + P_2 + P_3 = I$ gives

$$T = \sum_{i,j=1}^{3} P_i T P_j.$$

Selecting only the diagonal would change the operator. The entries $2, 3, 4$ carry transfers between different packets.

Composition sums over intermediate packets

The $(1, 2)$ entry of the square has two contributing routes:

$$(T^2)_{12} = T_{11}T_{12} + T_{12}T_{22} + T_{13}T_{32} = 1 \cdot 2 + 2 \cdot 1 + 0 = 4.$$

For input 3 and output 1, only the route through packet 2 survives:

$$(T^2)_{13} = T_{12}T_{23} = 2 \cdot 3 = 6.$$

For packets of higher dimension, the same law uses actual linear maps as entries. The order of their composition still matters.

> **Read this as:** An edge says a block is nonzero. The block itself carries the coefficient, sign, rank and action. A transfer graph alone does not retain those data.

Plate I.9 / REACHABILITY / NECESSARY IS NOT SUFFICIENT

Two visible routes can cancel exactly

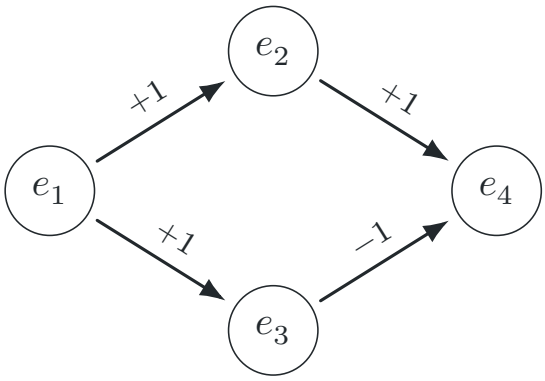
Main text: corollary 8.4 and proposition 9.3

Let a single operator D on $\mathbb{Q}^4$ act by

$$De_1 = e_2 + e_3, \qquad De_2 = e_4, \qquad De_3 = -e_4, \qquad De_4 = 0.$$

Every vertex of its transfer graph for the coordinate packets is reachable from vertex 1. Nevertheless,

$$D^2 e_1 = e_4 - e_4 = 0, \qquad D^2 = 0.$$



$$(D^2)_{41} = 1 - 1 = 0$$

$$[e_1, De_1, D^2 e_1]$$

1	0	0
0	1	0
0	1	0
0	0	0

$$\text{rank} = 2$$

The diamond is the actual weighted action of D. The two paths of length two exist, but their contributions to the same transfer block sum to zero.

What the seed generates

With the fixed algebra $\mathbb{Q}[D]$ and seed e_1,

$$\text{Hull}_{\mathbb{Q}[D]}(e_1) = \text{span}\{e_1, e_2 + e_3\}.$$

It has dimension two, not four. Both the independent difference $e_2 - e_3$ and the last coordinate e_4 are missing.

Why projecting in the middle changes the experiment

If P_2 becomes an admissible primitive, the word DP_2D satisfies

$$DP_2De_1 = DP_2(e_2 + e_3) = De_2 = e_4.$$

But P_2 was not an allowed generator of the original $\mathbb{Q}[D]$ grammar. Using it to argue that the original seed already reached e_4 would silently enlarge the algebra.

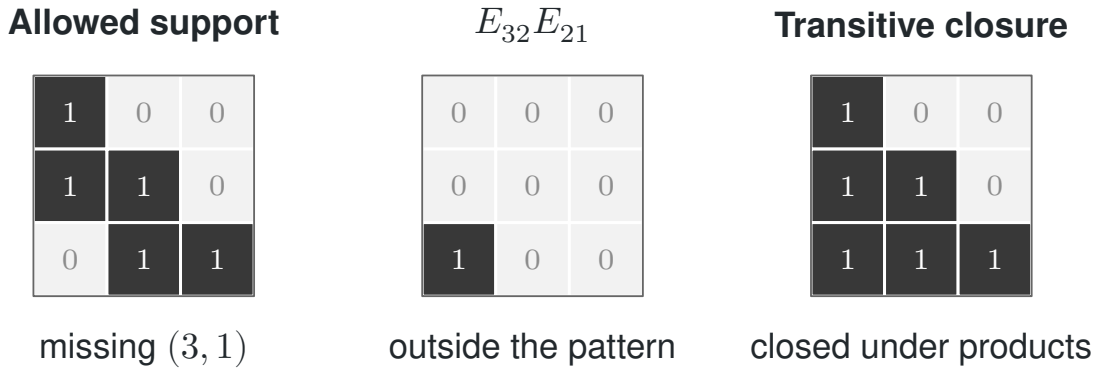
> **Read this as:** Unreachable nonzero packets obstruct generation. Reachability by itself does not prove generation: coefficients can cancel, blocks can have small images, and intermediate projectors may not be admissible.

Plate I.10 / SUPPORT ALGEBRAS / A UNIVERSAL TEST

Which zero patterns survive multiplication?

Main text: theorem 8.13. All packet spaces are nonzero

Suppose every matrix supported in a specified pattern is allowed. Then multiplication can create a new entry whenever two allowed edges compose. Let E_{ij} denote the matrix that sends e_j to e_i and kills the other basis vectors.



Here 1 means "allowed," not that every admissible operator has that numerical value. The middle matrix is an actual product that violates the original pattern.

The first pattern contains the diagonal and permits $1 \to 2$ and $2 \to 3$, but forbids $1 \to 3$. Yet

$$E_{32}E_{21} = E_{31} \neq 0.$$

Both factors obey the pattern and their product does not. Thus the whole supported matrix space is not an algebra.

Reflexivity and transitivity have different jobs

The identity has a nonzero diagonal block on every nonzero packet, so including the identity requires every packet to be related to itself. Multiplicative closure requires every composite pair. Consequently, for arbitrary algebraic block maps on nonzero packet spaces,

unital universal support algebra $\iff$ reflexive, transitive support relation.

The transitive closure in the figure is the lower triangular pattern. It does define a unital algebra.

Why this does not contradict cancellation

On Plate I.9, one particular operator has canceling paths. The support theorem asks whether multiplication works for *every* assignment of allowed block maps. Choosing two suitable maps of rank one removes the possibility of hiding a failed composite behind special cancellation.

> **Read this as:** A zero pattern is a structural rule only when it survives all permitted block assignments. A coincidence in one matrix is not a closure theorem.

Plate I.11 / CONJUGACY / STRUCTURE VERSUS RAW SPARSITY

Move the packets with the coordinates

Main text: theorems 10.1 and 10.4 and corollary 4.5

Start with $T = \mathrm{diag}(1, 2)$ and the two coordinate projections. Apply the invertible shear U. The transformed operator is $T^U = UTU^{-1}$, and the transformed projectors are $Q_i = UP_iU^{-1}$.

T

1	0
0	2

U

1	1
0	1

UTU^{-1}

1	1
0	2

$Q_1 = UP_1U^{-1}$

1	−1
0	0

$Q_2 = UP_2U^{-1}$

0	1
0	1

The transformed operator has a nonzero entry away from the diagonal in the standard basis. Relative to its transported, oblique packets, however, both blocks between distinct packets remain zero.

Indeed,

$$Q_1T^UQ_2 = Q_2T^UQ_1 = 0, \qquad Q_iT^UQ_j = U(P_iTP_j)U^{-1}.$$

A change in raw matrix sparsity is therefore not necessarily a change in the packet transfer graph. Conjugacy preserves the exact block identities and their zero pattern. A nonunitary shear need not preserve their Euclidean norms.

When the map is not invertible

For an intertwiner $\Phi A = \varphi(A)\Phi$, the hull identity uses the *image algebra*:

$$\Phi(\mathrm{Hull}_{\mathcal{A}}(S)) = \mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi S).$$

Replacing $\varphi(\mathcal{A})$ with a larger target algebra can enlarge the target hull. Seed rank need not be equal relative to that larger algebra. A surjective intertwiner gives the appropriate monotonicity inequality, not automatic equality.

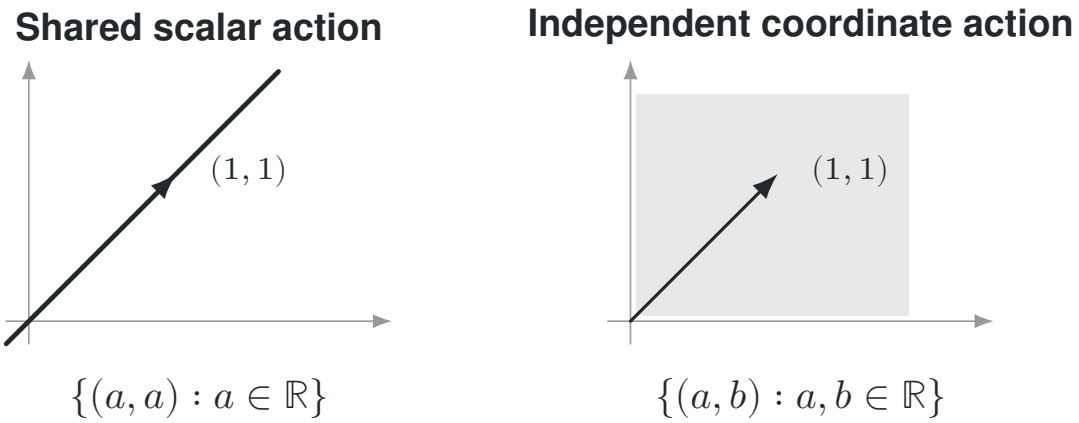
> **Read this as:** Transport the space, operators and packet maps together. The invariant is the collection of commuting algebraic relationships, rather than the appearance of the matrix.

Plate I.12 / DIRECT SUMS / ONE SEED CAN CARRY TWO COMPONENTS

Shared control and independent control differ

Main text: proposition 4.7 and theorem 4.8

Take $X = \Bbbk \oplus \Bbbk$ and $s = (1, 1)$. With one shared scalar acting on both coordinates, every orbit vector is (a, a). With independent scalars, the orbit contains every (a, b).



The geometric picture uses $\Bbbk = \mathbb{R}$. On the right, shading shows a finite window of the whole plane. On the left, only the diagonal line is generated. The algebraic distinction holds over any field.

The corresponding minimum seed numbers are

$$\operatorname{srank}_{\Bbbk I}(\Bbbk^2) = 2, \qquad \operatorname{srank}_{\Bbbk\times\Bbbk}(\Bbbk \oplus \Bbbk) = 1.$$

In the independent case, the algebra contains $(1, 0)$ and $(0, 1)$, which recover the two coordinates of s separately.

This is why the independent formula uses a maximum

For independent actions of $\mathcal{A}$ on X and $\mathcal{B}$ on Y, a common list of seed positions can carry the seeds for both spaces. Pad the shorter list with zeros. Independent operators then select the two components, giving

$$\operatorname{srank}_{\mathcal{A}\times\mathcal{B}}(X \oplus Y) = \max\{\operatorname{srank}_{\mathcal{A}}(X), \operatorname{srank}_{\mathcal{B}}(Y)\}.$$

With a shared action, this selection need not be available. The general bound is the sum.

One operator can supply independent control

Over $\mathbb{Q}$, $T = \operatorname{diag}(1, 2)$ has polynomial projectors $2I - T$ and $T - I$. Thus $\mathbb{Q}[T]$ already separates the two eigenspaces. If the eigenvalues coincide, these formulas no longer separate them. Distinct primary labels play the same role on Plate I.14.

> **Read this as:** Counting occupied components is not the same as counting seeds. One mixed seed may generate several components when the algebra can act on them independently.

Plate I.13 / SPLIT SEMISIMPLE MODEL / A CONSTRUCTIVE FORMULA

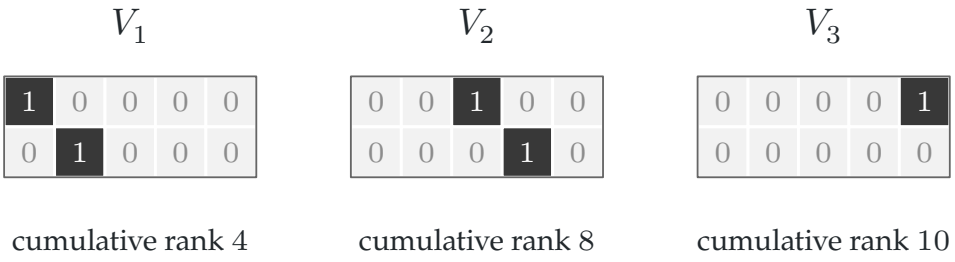
How much row space can one seed supply?

Main text: theorem 5.6 and corollary 5.7

Let $M_2(\mathbb{Q})$ act on $X = M_{2\times 5}(\mathbb{Q})$ by left multiplication. Think of the five columns as five copies of the same natural module of dimension two. One seed is a 2×5 matrix V.

Every row of AV is a linear combination of the two rows of V. Hence one seed exposes at most two independent directions in the multiplicity space of dimension five. Two seeds expose at most four. At least three are necessary.

Each row is one direction in the five-dimensional multiplicity space.



The three fixed seeds cover multiplicity directions $1, 2$, then $3, 4$, then 5. The exact dimensions of their cumulative matrix orbits are $4, 8, 10$.

An explicit synthesis, not just dimension counting

For

$$W = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix},$$

choose A_1 from columns $1, 2$ of W, A_2 from columns $3, 4$, and A_3 with first column equal to column 5 of W and second column zero. Then

$$A_1V_1 + A_2V_2 + A_3V_3 = W.$$

The same construction works for arbitrary entries of W, so the lower bound is attained:

$$\boxed{\operatorname{srank}_{M_2(\mathbb{Q})} M_{2\times 5}(\mathbb{Q}) = \lceil 5/2 \rceil = 3.}$$

Several split simple blocks

If the acting algebra and module are supplied in compatible product coordinates, the row budget is computed separately in each block and the formula for independent actions takes the maximum. For block data $(d, m) = (2, 5)$ and $(3, 4)$, the result is $\max(3, 2) = 3$, not $3 + 2$.

> **Read this as:** The formula measures multiplicity directions per seed. The formal development constructs compatible semisimple decompositions. The split formula uses the displayed algebra and module equivalences.

Plate I.14 / PRIMARY DATA / AN ACTUAL EXTRACTION CERTIFICATE

Distinct labels let one polynomial separate chains

Main text: theorem 6.2 and corollary 6.3

Keep the model in six dimensions, but compare two different operators:

$$N = J \oplus J, \qquad T = J \oplus (2I + J), \qquad J = J_3(0).$$

For the same mixed seed $s = e_0 + f_0$, the Krylov matrices are different:

Equal primary labels

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0

Krylov rank 3

Distinct primary labels

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
1	2	4	8	16	32
0	1	4	12	32	80
0	0	1	6	24	80

Krylov determinant 512

Exact rational calculations. Changing the primary label from 0 to 2 changes the operator, not merely its coordinates. The determinant on the right is $512 \neq 0$.

The missing separating operation is now a polynomial

Define

$$p_2(t) = \frac{3t^5 - 15t^4 + 20t^3}{16}, \qquad p_0(t) = 1 - p_2(t).$$

These polynomials satisfy

$$p_2 \equiv 0 \pmod{t^3}, \qquad p_2 \equiv 1 \pmod{(t-2)^3}.$$

Consequently $p_0(T) = P_e$ and $p_2(T) = P_f$. Applying them to s extracts both chain heads, after which polynomial powers generate all six directions. Unlike the model on Plate I.3, the separating maps already belong to the polynomial grammar.

> **Read this as:** Repeated labels can lock chains together. Distinct primary labels can admit polynomial extraction. Here the block data and extractor are explicit. No theorem asserting the existence of Jordan form is inferred from this example.

Plate I.15 / GRAPH MODEL / THE PATH CERTIFICATE

Walk counts make an endpoint cyclic

Main text: Worked graph model. Endpoint Krylov proof for every path size

Let A be the adjacency matrix of the path with five vertices. The (i, j) entry of

$$K = [e_1, Ae_1, A^2e_1, A^3e_1, A^4e_1]$$

counts walks of length $j - 1$ from vertex 1 to vertex i.



k	0	1	2	3	4
	1	0	1	0	2
	0	1	0	2	0
	0	0	1	0	3
	0	0	0	1	0
	0	0	0	0	1

unit diagonal; no entries below it

The entries are walk counts. At each new power, a new furthest vertex appears with coefficient one. The dark graph vertex is the seed.

A walk cannot reach vertex i in fewer than $i - 1$ steps. Therefore entries below the diagonal vanish. There is exactly one walk of length $i-1$ from the first vertex to vertex i: the shortest path. The diagonal entries are all one, so $\det K = 1$.

The same reason works at every size

For a path with any $n \geq 1$, the endpoint Krylov matrix is upper triangular with unit diagonal. This is a combinatorial proof of cyclicity over *every field*. It does not rely on numerical eigenvalues. The appendix checks the finite cases $1 \leq n \leq 12$ as implementation examples, separately from the chapter's proof for every path size.

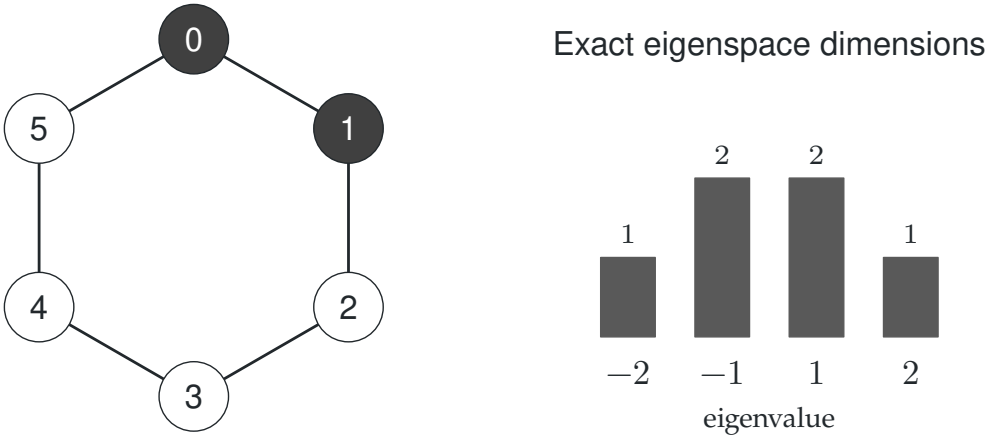
> **Read this as:** The endpoint distinguishes distances. Each extra operator power reveals one new coordinate before the finite space is exhausted.

Plate I.16 / GRAPH MODEL / A CYCLE WITH SIX VERTICES

Closing a path changes the seed obstruction

Main text: Worked graph model. Proof of the minimum seed count for complex cycles

For the adjacency matrix A of C_6, the distinct eigenvalues are $2, 1, -1, -2$, with multiplicities $1, 2, 2, 1$. A single vertex seed spans only four directions. Two adjacent vertex seeds span all six.



The two dark vertices are the displayed generating pair. The bars show exact eigenspace dimensions, not estimated spectral peaks.

A lower bound without a numerical eigensolver

The two rows of the following matrix are independent and obey $qA = q$:

$$q = \begin{pmatrix} 2 & 1 & -1 & -2 & -1 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 \end{pmatrix}.$$

Thus $q : \mathbb{Q}^6 \twoheadrightarrow \mathbb{Q}^2$ is a scalar quotient for the eigenvalue 1. Its extension over $\mathbb{C}$ gives the same obstruction of dimension two. The orbit of any single seed projects to a line, so no single seed can generate the full module.

The matching construction

From adjacent vertex vectors e_0, e_1, the identity

$$e_{j+1} = Ae_j - e_{j-1}$$

recovers successive vertices around the cycle. The lower and upper bounds therefore give two seeds for this example. The chapter proves the corresponding minimum for every complex simple cycle of size at least three.

> **Read this as:** Connectedness is not cyclicity for the adjacency grammar. If a directed cyclic shift were allowed instead, one vertex would generate the whole cycle. That would be a different operator algebra.

Plate I.17 / COORDINATE ALGEBRA / NORMAL FORM RATHER THAN A CURVE SKETCH

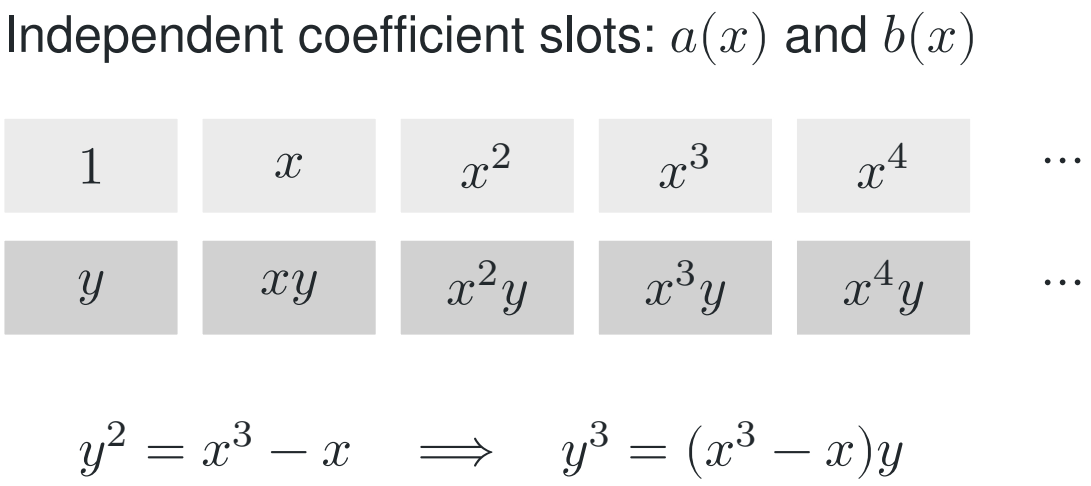
A relation becomes a packet product rule

Main text: Worked coordinate ring model. Normal form and exact multiplication

Work over $\mathbb{Q}$ in

$$A = \mathbb{Q}[x, y]/(y^2 - x^3 + x).$$

Because the relation is monic of degree two in y, each element has one unique form $a(x) + yb(x)$. The two packet coefficients are polynomials, not scalars.



A finite window of the two infinite coefficient lanes. Higher powers of y are rewritten using the relation rather than discarded. No geometric genus or smoothness claim is encoded by this picture.

Multiplication by y transfers between the lanes

In the ordered coefficient coordinates (a, b),

$$M_y = \begin{pmatrix} 0 & x^3 - x \\ 1 & 0 \end{pmatrix}, \qquad M_y^2 = (x^3 - x)I.$$

The entries denote multiplication maps on $\mathbb{Q}[x]$. For $f = a_0 + ya_1$ and $g = b_0 + yb_1$,

$$fg = (a_0b_0 + (x^3 - x)a_1b_1) + y(a_0b_1 + a_1b_0).$$

In particular,

$$(x^2 + y)(x - y) = x + (x - x^2)y.$$

One module seed does not mean one algebra generator

Starting from 1, the fixed multiplication operators M_x, M_y generate every monomial in normal form. Thus the algebra is cyclic as a module over its multiplication algebra. This does not remove either address operator, and does not say that the algebra is generated by one algebra element.

> **Read this as:** The relation controls an exact weighted transfer between coefficient lanes. Packet reconstruction is $a + yb$, while packet multiplication retains the weight $x^3 - x$.

Plate I.18 / INFINITE MODEL / THE BILATERAL SHIFT

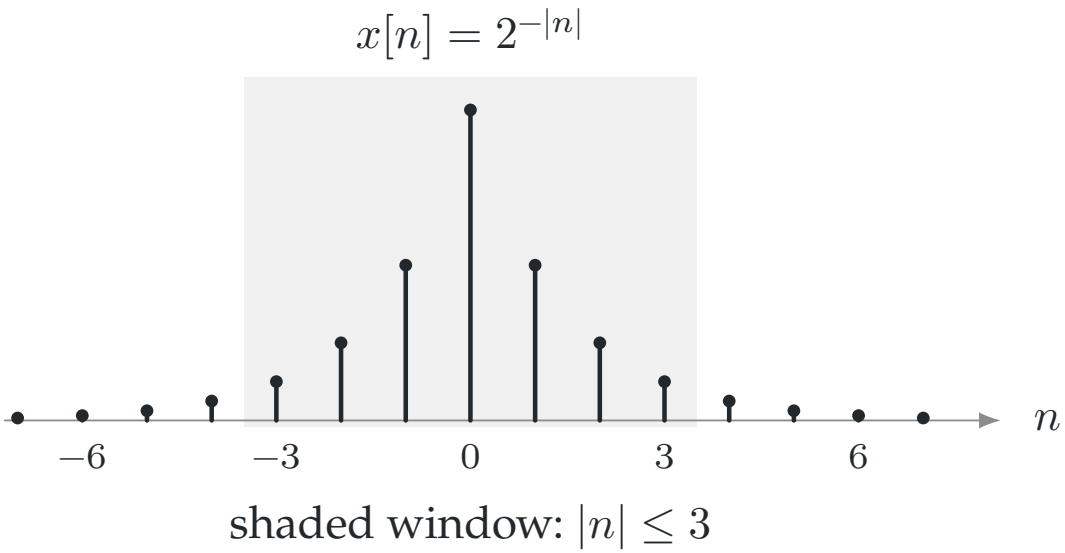
Density is not finite generation of each vector

Main text: proposition 15.1 and theorem 3.6

On $\ell^2(\mathbb{Z})$, declare every integer shift

$$(S_m x)[n] = x[n-m].$$

The seed δ_0 generates $\delta_m = S_m \delta_0$. Finite linear combinations give precisely the sequences with finite support. Their norm closure is all of $\ell^2(\mathbb{Z})$.



A finite window of the actual sequence $x[n] = 2^{-|n|}$. Outside every finite window, coefficients remain nonzero. The shaded window illustrates the truncation $x^{(3)}$, not a periodic wraparound.

This x cannot be a finite linear combination of shifted impulses: it has infinite support. But if $x^{(N)}$ agrees with x on $|n| \le N$ and is zero elsewhere, then

$$\left\|x - x^{(N)}\right\|_2^2 = 2 \sum_{n>N} 4^{-n} = \frac{2}{3}\, 4^{-N} \longrightarrow 0.$$

The exact squared errors for $N = 1, 3, 5$ are $1/6$, $1/96$, and $1/1536$. These are identities from a geometric series, not floating residuals.

What the topology contributes

The statement is that *one seed has dense hull*. It is not that every vector is a finite word expression. The completion theorem additionally specifies a normed presentation, a dense linear isometry into a Banach target, and the bounded intertwining needed to extend operators.

> **Read this as:** Completion adds limits in a declared topology. A finite numerical window can illustrate the approximants, but cannot be substituted for the infinite object or its convergence proof.

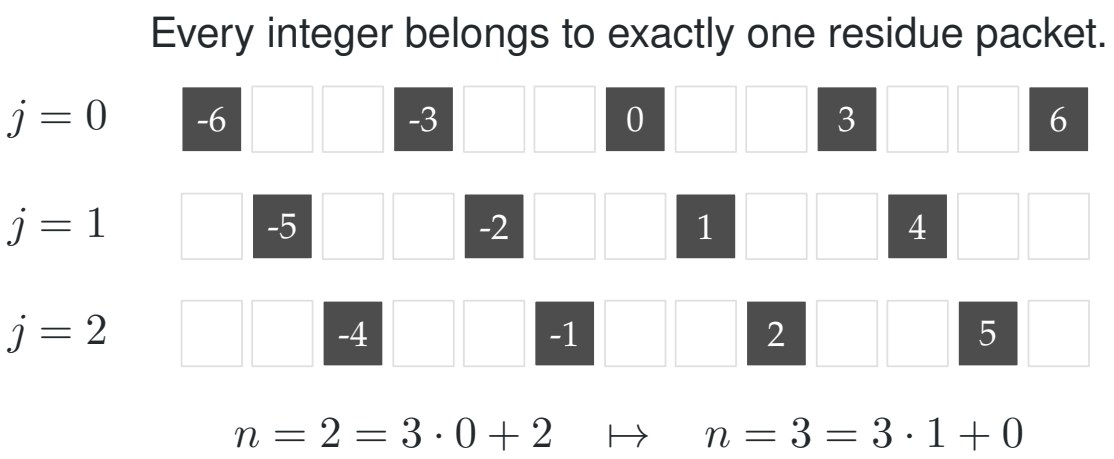
Plate I.19 / INFINITE MODEL / THREE EXACT SUBSEQUENCES

Residue packets retain an address carry

Main text: theorem 7.10. Residue packets

For $j = 0, 1, 2$, let P_j keep the entries with $n \equiv j \pmod 3$. These are bounded orthogonal projectors whose sum is the identity. The packet coordinates can be written as

$$x_j[k] = x[3k + j], \qquad k \in \mathbb{Z}.$$



Each occupied tile represents one integer address in a displayed window. The three rows partition the addresses exactly. Negative indices use the same residue convention.

What one shift does in these coordinates

Let $S = S_1$ and $z = Sx$. Substituting $3k + j$ into the shift definition gives

$$z_1[k] = x_0[k], \qquad z_2[k] = x_1[k], \qquad z_0[k] = x_2[k - 1].$$

The passage from phase 2 to phase 0 carries a shift of the coarse address. If $Lh[k] = h[k - 1]$, the exact packet matrix is

$$S_{\text{packets}} = \begin{pmatrix} 0 & 0 & L \\ I & 0 & 0 \\ 0 & I & 0 \end{pmatrix}.$$

Replacing L by a scalar 1 would preserve the phase graph but would lose the operator's action on the infinite sequences.

Synthesis keeps the positions

To reconstruct x, place $x_j[k]$ at its original address $3k + j$. In the ambient space formulation, this is the finite sum $x = P_0x + P_1x + P_2x$. Boundedness makes the same formulas valid after taking the ℓ^2 completion.

> **Read this as:** A cyclic packet graph is only the routing pattern. The block at the phase boundary still carries an actual shift operator.

Plate I.20 / EXACT COMPUTATION / AN INVARIANCE CERTIFICATE

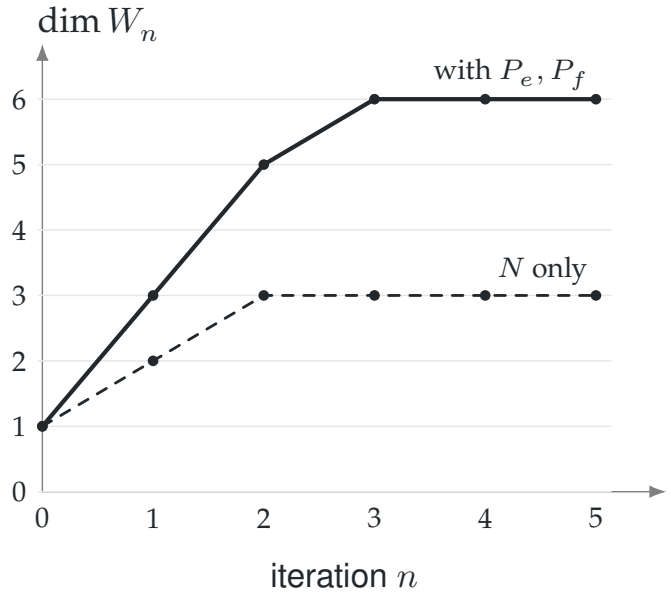
Stop when the subspace stops growing

Main text: construction 19.1

For a fixed finite operator family (O_i), start from $W_0 = \operatorname{span} S$ and iterate

$$W_{n+1} = W_n + \sum_i O_i W_n.$$

This is an ascending chain. When two consecutive subspaces agree, every generator maps the stable subspace into itself. The stable value is therefore the entire hull of operator words, not just the span of the words tested so far.



Exact dimensions for the same mixed seed in Model A. The dashed curve allows only N. The solid curve also admits the two chain projectors. Plateau values are repeated to show the certified stop.

Why the process stops in finite dimensions

If an inclusion is strict, the dimension rises by at least one. It cannot rise beyond $\dim X$. At equality, invariance and containment of the seeds prove both inclusions between the stable space and the least invariant hull.

What the companion code actually does

The new checker retains exact column bases using rational row reduction. It then verifies that adjoining the images of those bases under every generator does not increase rank. These examples use neither a cutoff on singular values nor a numerical tolerance.

The Lean statement is a mathematical termination and correctness specification. It is not a claim that this Python implementation for maintaining a basis was extracted from Lean or that a particular runtime bound was proved. The original numerical companion and its tolerances remain separate.

> **Read this as:** "No new directions" is a stopping certificate only after checking the full declared operator family. Stopping after a convenient selection of words is not the same test.

Plate I.21 / OPERATIONS ON SEED RANK / A RECTANGLE, NOT A DIAGONAL

Tensor products require every cross term

Main text: proposition 4.9. Algebraic tensor products

If (x_i) generates X under $\mathcal{A}$ and (y_j) generates Y under $\mathcal{B}$, then the candidate tensor seeds are *all* $x_i \otimes y_j$. For finite expansions,

$$x = \sum_i A_i x_i, \qquad y = \sum_j B_j y_j,$$

bilinearity gives the required double sum:

$$x \otimes y = \sum_i \sum_j (A_i \otimes B_j)(x_i \otimes y_j).$$

The coefficient rectangle

	$5f_1$	$7f_2$
$2e_1$	10	14
$3e_2$	15	21

$$10(e_1 \otimes f_1) + 14(e_1 \otimes f_2)$$

$$+15(e_2 \otimes f_1) + 21(e_2 \otimes f_2)$$

The exact expansion of $(2e_1 + 3e_2) \otimes (5f_1 + 7f_2)$. The terms away from the diagonal 14 and 15 cannot be omitted.

Keeping only the tensors with matching indices $e_1 \otimes f_1$ and $e_2 \otimes f_2$ spans a subspace of dimension two in a tensor product of dimension four. It misses both cross directions. Under scalar actions on $\mathbb{Q}^2$ and $\mathbb{Q}^2$, all four basis tensors are necessary and sufficient.

The resulting upper bound

Since simple tensors span the algebraic tensor product, the complete rectangular family proves

$$\boxed{\operatorname{srank}_{\mathcal{A}\otimes\mathcal{B}}(X \otimes Y) \le \operatorname{srank}_{\mathcal{A}}(X)\ \operatorname{srank}_{\mathcal{B}}(Y).}$$

This is not a universal equality claim. Changing the algebra changes the answer: the full independent matrix action can make a single nonzero pure tensor cyclic. Nor does the algebraic argument itself establish a theorem for completed topological tensor products.

> **Read this as:** The generator construction is indexed by a Cartesian product. Its proof must retain the finite double sum and the hypotheses on scalar actions that make the tensor action well defined.

Plate I.22 / PROOF BOUNDARY / RETURN TO THE MANUSCRIPT

What the pictures certify

Main text: Reading the theorem hypotheses and companion evidence

The mathematical statements in the main text have explicit hypotheses and written proofs. The Lean guide separately identifies the checked declarations, hypotheses, and coverage. The computational companion identifies the 108 stable foundation records and the checks executed on their structured representations. These are distinct forms of evidence.

Topic	Mathematical content	What must be retained
Presentations and hulls	Evaluation, module quotients, and generation under a fixed algebra.	The grammar must be fixed independently of the target.
Finite packets	Idempotence, annihilation, completeness and exact finite synthesis.	Orthogonality requires an additional hypothesis.
Semisimple formula	Multiplicities over division rings determine the minimal free cover.	The stated classical inputs and decomposition data.
Primary and graph formulas	Primary components and scalar quotients give sharp generator counts.	The coefficient field and multiplicity conventions.
Paths and cycles	Formulas for paths of every size and for complex cycles have separate proofs.	Finite tests illustrate rather than prove claims for every size.
Coordinate algebra	Normal form, multiplication, packet identities and a unit seed.	Smooth projective geometry needs additional classical input.
Completion	Density and bounded extraction in the stated topology.	Infinite sequences are distinct from periodic finite arrays.

Evidence travels with the examples

The supporting material includes mathematical source, exact model definitions, deterministic graphics, Python checks, Lean source and theorem mapping, and the computational record archive. The visual models use integer, rational, symbolic, or explicitly analytic calculations. The verification record distinguishes the execution evidence for each component.

> **Read this as:** The pictures make the hypotheses visible. Read the traditional proof, the formal contract, and the executable record at their respective stated scopes.

CHAPTER II

Projector Resolved Seed Calculus II

Refinement towers, weighted transfer, and integral separation of operator packets

The linear framework now becomes a language for comparing resolutions.

Refinement towers compare packet descriptions across levels. Weighted transfers and integral separation show which structures survive a change of resolution.

IN THIS CHAPTER

Mathematical text 99

Visual studies 152

Formal companion 1721

Sources and references 2668

Scope

Canonical values are defined for admissible data. A general existence theorem for admissible contours remains a separate question.

The next chapter moves from linear packet coordinates to typed nonlinear operations and analytic jets.

How to use this volume

Reading route. Exact packet coordinates and residue selectors lead to weighted inversion and rational walk series. Lattice gluing then gives separation weights and precision delays. Smith congruences and operator defects connect these formulas to finite computation.

Companions. The worked plates illustrate the constructions of this chapter. The formal companion in the back matter distinguishes the chapter-wide correspondence from the foundational 25-row walkthrough, with explicit evidence and scope.

Overview

A complete family of idempotents gives exact coordinates, but a single family does not record scale or integral visibility. We study towers in which every coarse idempotent is the sum of its fine descendants. The coordinate changes are invertible and coherent, and every operator block is the sum of its refined blocks. Residue selectors give a model over arbitrary rings. When roots of unity are available they agree with character averages, and coprime moduli factor through the Chinese remainder theorem. In a complete filtered algebra, a diagonal operator plus positive weight transfers has an inverse obtained by summing admissible weighted paths. The formula truncates exactly at finite precision, while finite transfer graphs have rational path series. For a rational spectral decomposition over a discrete valuation ring, resultants give denominator bounds. The exact denominator is the largest Smith exponent of the lattice gluing module. This yields a necessary and sufficient precision delay, monotonicity under refinement, and a full congruence solving formula. We close with exact formulas for the propagation of a coarsening defect through powers, resolvents, holomorphic functions, exponentials, and multilevel towers.

The exposition distinguishes general written proofs, formal specifications, and finite computational examples. The worked plates supply concrete models. The formal companion identifies the exact declarations and hypotheses that support each mapped statement.

1 Introduction

A complete family of pairwise annihilating idempotents decomposes a module into exact coordinate pieces. Many problems require more than one such decomposition. A signal may be split first into two residue classes and then into four. A filtered operator may be resolved into transfers that appear at different valuation depths. A rational spectral splitting may exist over the fraction field while remaining invisible on the original lattice. These are all refinement problems, but they involve different notions of scale.

The first part of the chapter concerns exact refinement. If a fine packet family refines a coarse family, then each coarse idempotent is the sum of its fine descendants. The coordinate spaces at the two levels are canonically isomorphic because both represent the whole module. The refinement relation adds more information. It tells us which fine coordinates belong to each coarse coordinate, and it gives a compatible decomposition of every operator block. This is the algebra behind polyphase coordinates and nested filter banks. Standard treatments of those subjects may be found in [10, 2].

The second part studies inversion in a complete filtered algebra. Suppose an operator has the form

$$T = D + N,$$

where D is invertible and every packet transfer in N raises the filtration by a positive amount. The ordinary Neumann series can then be expanded into weighted paths. The weight is not the number of edges. It is the total filtration gain along the path. At precision L, every path of weight at least L vanishes. A finite transfer graph also gives a rational generating function for the number of paths of each total weight. The relevant generating function facts are classical and are discussed in [9].

The third part concerns integral separation. Let V be a vector space over the fraction field of a discrete valuation ring $\mathcal{O}$, and let Λ be an $\mathcal{O}$ lattice in V. A rational projector E need not preserve Λ. Its separation weight is the least s such that

$$\pi^{s} E\Lambda \subseteq \Lambda.$$

A Bézout construction gives an upper bound through a polynomial resultant. That bound need not be exact because it does not see the position of the lattice. The exact answer is controlled by the finite length module

$$G_{\Lambda,\mathcal{P}} = \Lambda \Big/ \bigoplus_{\alpha} (\Lambda \cap V_{\alpha}),$$

where $V = \bigoplus_{\alpha} V_{\alpha}$ is the rational packet decomposition. The largest Smith exponent of this gluing module is exactly the common denominator needed to make every packet projector integral. This interpretation also proves that separation weight cannot decrease under refinement. Background on local fields, Smith normal form, and matrix polynomials may be found in [8, 6, 3].

The last part records how a defect between fine and coarse operators propagates. If

$$\Delta = CT_{f} - T_{c}C,$$

then exact formulas describe the corresponding defect for powers, resolvents, holomorphic functions, exponentials, and a chain of several refinement levels. These formulas are elementary, but collecting them in one place makes the later analytic uses transparent. The spectral and resolvent background follows the usual operator theory in [4].

Every theorem in this chapter concerns linear operators. An exact refinement retains the full object and changes its coordinates. Questions about nonlinear operation trees, nonlinear correction, and obstruction complexes require a separate theory.

2 Depth and compatibility

Packet, filtration, and separation depths describe three different kinds of scale. Compatibility is initially recorded by an operator defect. Keeping these notions separate makes the later formulas easier to read.

The first is packet depth. A finer packet family has more idempotents and a more detailed address set. The passage from residues modulo two to residues modulo four is an example. Packet depth is combinatorial. It says how many distinctions have been made in the coordinate system.

The second is filtration depth. A transfer in $F^r\mathcal{B}$ vanishes modulo $F^r\mathcal{B}$. This membership gives a lower bound on its depth. A nonzero transfer may first become visible only at a higher precision. Two paths with the same number of edges can have different filtration depths. A longer path can appear earlier when its edges have smaller weights.

The third is separation depth. A rational packet projector may require multiplication by π^s before it preserves the chosen lattice. This number depends on both the rational decomposition and the lattice. It can be zero even when the corresponding eigenvalues are close, and it can be positive even when the rational projector is simple.

The fourth notion is the compatibility defect. A fine operator and a separately chosen coarse operator need not commute with coarsening. Their discrepancy is

$$\Delta = CT_f - T_cC.$$

This operator records the failure of one model to represent another exactly. A chosen norm measures its size, while a chosen filtration can assign a depth. No integer compatibility depth is assigned without that additional structure.

The four notions meet in applications, but no one of them determines the others. An exact refinement tower may have zero coordinate error while its fine spectral packets have positive lattice weight. A filtered inverse may truncate exactly at a given depth while the coarse operator still has a nonzero compatibility defect. The chapter treats each quantity with its own theorem before combining them.

The central idea

Packet depth changes the address of a component. Filtration depth controls when a transfer survives a precision cutoff. Separation depth clears a rational denominator on a chosen lattice. The compatibility defect records a discrepancy between two operator models. These quantities can interact, but each needs its own hypothesis and its own calculation.

A simple two packet model already contains all four notions. Let

$$X = X_0 \oplus X_1, \qquad D = \begin{pmatrix} D_0 & 0 \\ 0 & D_1 \end{pmatrix}, \qquad N = \begin{pmatrix} 0 & N_{01} \\ N_{10} & 0 \end{pmatrix}.$$

A finer family may split X_0 and X_1 into smaller summands. The blocks N_{01} and N_{10} may enter at different filtration weights. Over a fraction field, the rational packet spaces may fail to split the chosen

lattice. A coarse operator on one selected summand may also fail to intertwine with the full matrix. The later sections isolate these four effects one at a time.

3 Packet systems and refinement towers

Packet coordinates split a module into pieces that can be recovered separately and then added back together. The construction uses idempotents and does not need an inner product.

Definition 3.1 (Packet system)**.** Let X be a module over a unital ring. A finite packet system on X is a family

$$\mathcal{P} = (P_\alpha)_{\alpha \in I} \subseteq \mathrm{End}(X)$$

of nonzero idempotents such that

$$P_\alpha P_\beta = 0 \quad \text{when } \alpha \neq \beta, \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

Write

$$X_\alpha = P_\alpha X, \qquad X^{\mathcal{P}} = \bigoplus_{\alpha \in I} X_\alpha.$$

The analysis and synthesis maps are

$$\mathcal{A}_{\mathcal{P}} x = (P_\alpha x)_\alpha, \qquad \mathcal{S}_{\mathcal{P}}(x_\alpha)_\alpha = \sum_\alpha x_\alpha.$$

They are inverse isomorphisms.

Definition 3.2 (Refinement)**.** Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ and $\mathcal{Q} = (Q_\beta)_{\beta \in J}$ be packet systems on the same module. We say that $\mathcal{Q}$ refines $\mathcal{P}$ when there is a surjection

$$\rho : J \longrightarrow I$$

such that

$$P_\alpha = \sum_{\rho(\beta) = \alpha} Q_\beta$$

for every $\alpha \in I$. The index β is then called a child of $\rho(\beta)$.

Theorem 3.3 (Children, parents, and transitivity)**.** *Suppose that $\mathcal{Q}$ refines $\mathcal{P}$ through ρ. Then*

$$P_\alpha Q_\beta = Q_\beta P_\alpha = \begin{cases} Q_\beta & \rho(\beta) = \alpha, \\ 0 & \rho(\beta) \neq \alpha. \end{cases}$$

Hence $Q_\beta X \subseteq P_{\rho(\beta)} X$. If a third packet system $\mathcal{R}$ refines $\mathcal{Q}$, then $\mathcal{R}$ refines $\mathcal{P}$ through the composite parent map.

Proof. Multiply the identity

$$P_\alpha = \sum_{\rho(\gamma)=\alpha} Q_\gamma$$

by Q_β on either side. Pairwise annihilation in the fine family leaves Q_β when $\rho(\beta) = \alpha$ and leaves zero otherwise. For transitivity, substitute the child expansion of each Q_β into the expansion of P_α and group the resulting terms by the composite parent. □

A refinement tower is a sequence $(\mathcal{P}_\ell)_{\ell\geq 0}$ with parent maps

$$\rho_{m,\ell} : I_m \longrightarrow I_\ell \qquad (m \geq \ell)$$

such that $\mathcal{P}_m$ refines $\mathcal{P}_\ell$ through $\rho_{m,\ell}$ for every $m \geq \ell$, and

$$\rho_{\ell,\ell} = \mathrm{id}_{I_\ell}, \qquad \rho_{n,\ell} = \rho_{m,\ell} \circ \rho_{n,m} \qquad (n \geq m \geq \ell).$$

Thus every parent map witnesses the stated refinement and the maps compose coherently. The next result shows that the coordinate changes form a coherent system.

Definition 3.4 (Coordinate change across levels)**.** For two packet systems $\mathcal{P}$ and $\mathcal{Q}$ on X, define

$$J_{\mathcal{Q}\leftarrow\mathcal{P}} = \mathcal{A}_{\mathcal{Q}}\mathcal{S}_{\mathcal{P}} : X^{\mathcal{P}} \longrightarrow X^{\mathcal{Q}}$$

and

$$C_{\mathcal{P}\leftarrow\mathcal{Q}} = \mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{Q}} : X^{\mathcal{Q}} \longrightarrow X^{\mathcal{P}}.$$

When $\mathcal{Q}$ refines $\mathcal{P}$, the first map resolves each coarse coordinate into its children and the second adds the children back to their parent.

The central idea

A coarse coordinate retains a whole vector in its coarse packet. Adding its children keeps the whole vector inside that packet. Fine analysis recovers each child by applying its projector. This explains why the two coordinate maps can be inverse even when the fine system has more packet labels.

Theorem 3.5 (Exact coordinate change and coherence)**.** *For any two packet systems on X,*

$$C_{\mathcal{P}\leftarrow\mathcal{Q}}J_{\mathcal{Q}\leftarrow\mathcal{P}} = I_{X^{\mathcal{P}}}, \qquad J_{\mathcal{Q}\leftarrow\mathcal{P}}C_{\mathcal{P}\leftarrow\mathcal{Q}} = I_{X^{\mathcal{Q}}}.$$

For three packet systems $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ on X,

$$J_{\mathcal{R}\leftarrow\mathcal{Q}}J_{\mathcal{Q}\leftarrow\mathcal{P}} = J_{\mathcal{R}\leftarrow\mathcal{P}},$$

$$C_{\mathcal{P}\leftarrow\mathcal{Q}}C_{\mathcal{Q}\leftarrow\mathcal{R}} = C_{\mathcal{P}\leftarrow\mathcal{R}}.$$

If $\mathcal{Q}$ refines $\mathcal{P}$ through ρ, then

$$\left(J_{\mathcal{Q}\leftarrow\mathcal{P}}x\right)_\beta = Q_\beta x_{\rho(\beta)}, \qquad \left(C_{\mathcal{P}\leftarrow\mathcal{Q}}y\right)_\alpha = \sum_{\rho(\beta)=\alpha} y_\beta.$$

Proof. The packet analysis and synthesis maps are inverse. Therefore

$$C_{\mathcal{P}\leftarrow\mathcal{Q}}J_{\mathcal{Q}\leftarrow\mathcal{P}} = \mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}}\mathcal{S}_{\mathcal{P}} = \mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}} = I.$$

The other inverse identity is the same calculation with the two systems exchanged. The coherence identities follow by inserting

$$\mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}} = I_X.$$

Now assume that $\mathcal{Q}$ refines $\mathcal{P}$. If $x = (x_\alpha)_\alpha$ with $x_\alpha \in P_\alpha X$, then

$$Q_\beta \sum_\alpha x_\alpha = Q_\beta x_{\rho(\beta)}$$

by theorem 3.3. The formula for C follows by synthesis followed by projection onto the parent packet. □

Remark 3.6 (Refinement and reduction). The maps in theorem 3.5 are isomorphisms because both coordinate spaces represent all of X. They do not reduce dimension and they do not discard packets. Approximation begins only after one chooses a truncation or compression rule.

Corollary 3.7 (Orthogonal Hilbert packets)**.** *If X is a real or complex Hilbert space and every packet system in the tower consists of orthogonal projections, equip each finite packet direct sum with the Hilbert inner product $\langle (x_\alpha), (y_\alpha)\rangle = \sum_\alpha \langle x_\alpha, y_\alpha\rangle$. Then analysis and synthesis are unitary. Every coordinate change in theorem 3.5 is unitary.*

Proof. For an orthogonal packet system,

$$\|x\|^2 = \sum_\alpha \|P_\alpha x\|^2.$$

Thus analysis is an isometry onto the full packet direct sum. Synthesis is its inverse. A coordinate change is a composition of two unitary maps. □

Every coarse packet is also uniquely reconstructed from its fine children. If vectors $x_\beta \in Q_\beta X$ sum to a coarse packet x, applying Q_γ recovers $x_\gamma = Q_\gamma x$. This is the uniqueness clause of the canonical nested-refinement statement.

3.1 Typed coordinate naturality

These coordinate changes are coherent through every finite tower. If three complete packet systems are denoted by a, b, c, then

$$J_{c,b}J_{b,a} = J_{c,a}, \qquad J_{a,a} = I, \qquad J_{b,a}^{-1} = J_{a,b}.$$

These laws hold for arbitrary exact packet systems representing the same ambient module. The parent maps supply the component formulas linking children to their parents. Coordinate coherence itself does not require them.

The same coherence applies to operators. Let P_a, P_b be packet systems on X, let Q_a, Q_b be packet systems on Y, where X and Y are modules over the same scalar ring, and let $T : X \to Y$ be a linear map over that ring. Its exact coordinate representation

$$\widehat{T}_{Q,P} = \mathcal{A}_Q T \mathcal{S}_P$$

preserves identities and typed composition, and the naturality square is

$$\boxed{J^{Y}_{Q_b,Q_a}\,\widehat{T}_{Q_a,P_a} = \widehat{T}_{Q_b,P_b}\,J^{X}_{P_b,P_a}.}$$

The identity, cocycle, composition, inverse, and naturality laws are all proved in Lean. The separate refinement hypotheses supply the local component formulas linking each child to its parent.

How to read the result

These identities concern complete coordinates of the same module. A quotient is a different operation: it identifies vectors that differ by an element of a chosen submodule. Later, finite precision will be expressed by a quotient that removes terms beyond the chosen filtration depth. This distinction explains why refinement is exactly reversible while precision truncation has a prescribed residual.

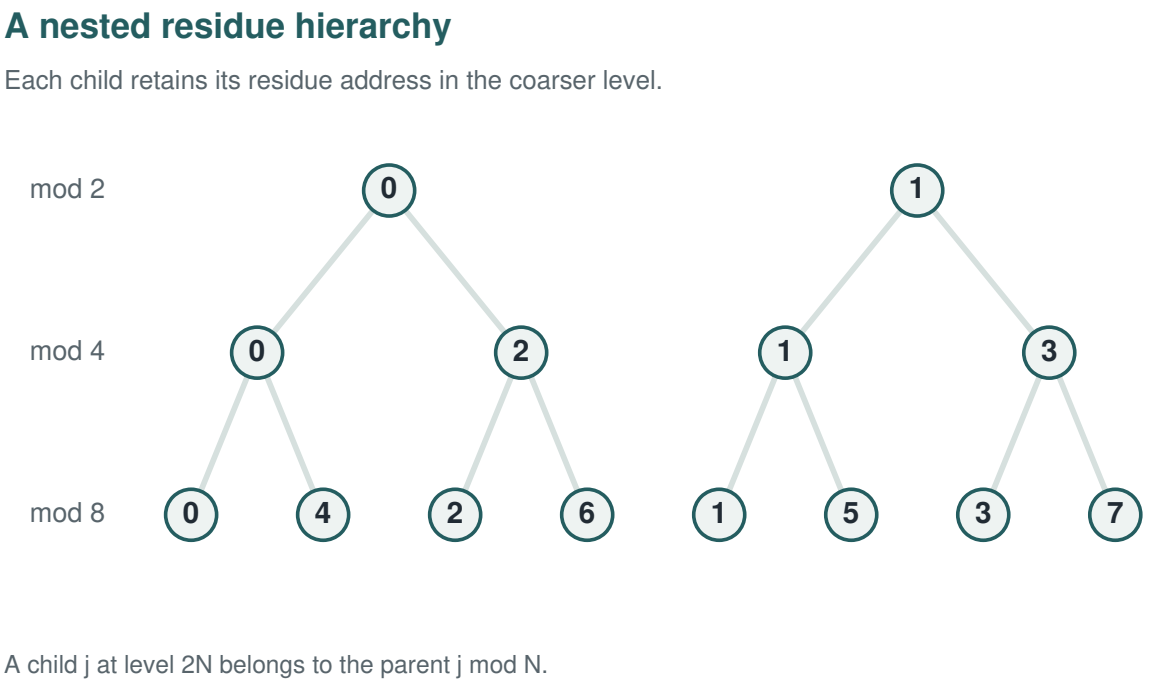


Figure 1. A nested residue hierarchy. Each residue modulo 2 is the union of two residues modulo 4, and each residue modulo 4 is the union of two residues modulo 8.

3.2 Operator blocks across refinement

For $T \in \mathrm{End}(X)$ and a packet system $\mathcal{P}$, write

$$T^{\mathcal{P}}_{\beta\alpha} = P_\beta T P_\alpha.$$

Theorem 3.8 (Refinement of operator blocks). *Suppose that $\mathcal{Q}$ refines $\mathcal{P}$ through ρ. Then*

$$T^{\mathcal{P}}_{ba} = \sum_{\substack{\rho(\beta)=b \\ \rho(\alpha)=a}} T^{\mathcal{Q}}_{\beta\alpha}.$$

The matrix of T in fine coordinates is related to the matrix in coarse coordinates by

$$[T]_{\mathcal{P}} = C_{\mathcal{P}\leftarrow\mathcal{Q}}[T]_{\mathcal{Q}}J_{\mathcal{Q}\leftarrow\mathcal{P}}.$$

Proof. Expand the two coarse projectors into their children:

$$P_b T P_a = \left(\sum_{\rho(\beta)=b} Q_\beta\right) T \left(\sum_{\rho(\alpha)=a} Q_\alpha\right).$$

Distributing the finite sums gives the first identity. For the coordinate identity, use

$$[T]_{\mathcal{P}} = \mathcal{A}_{\mathcal{P}} T \mathcal{S}_{\mathcal{P}}$$

and insert

$$I_X = \mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}}.$$

□

Corollary 3.9 (Support under refinement). *Under the hypotheses of theorem 3.8, a coarse block $T^{\mathcal{P}}_{ba}$ is zero if and only if every fine descendant block*

$$T^{\mathcal{Q}}_{\beta\alpha} \qquad (\rho(\beta) = b,\ \rho(\alpha) = a)$$

is zero.

Proof. If every fine block is zero, the coarse block is zero by theorem 3.8. Conversely, suppose $P_b T P_a = 0$. Multiply on the left by Q_β and on the right by Q_α. The child identities in theorem 3.3 give

$$Q_\beta T Q_\alpha = 0.$$

□

Thus refinement cannot hide an active transfer inside a zero coarse block. It can only reveal where the active transfer sits among the descendants.

3.3 Common refinements

Two packet systems need not be comparable. When their idempotents commute, their products give a common refinement.

Theorem 3.10 (Common refinement of commuting packet systems). *Let $\mathcal{P} = (P_\alpha)_{\alpha\in I}$ and $\mathcal{Q} = (Q_\beta)_{\beta\in J}$ be packet systems on X. Assume*

$$P_\alpha Q_\beta = Q_\beta P_\alpha$$

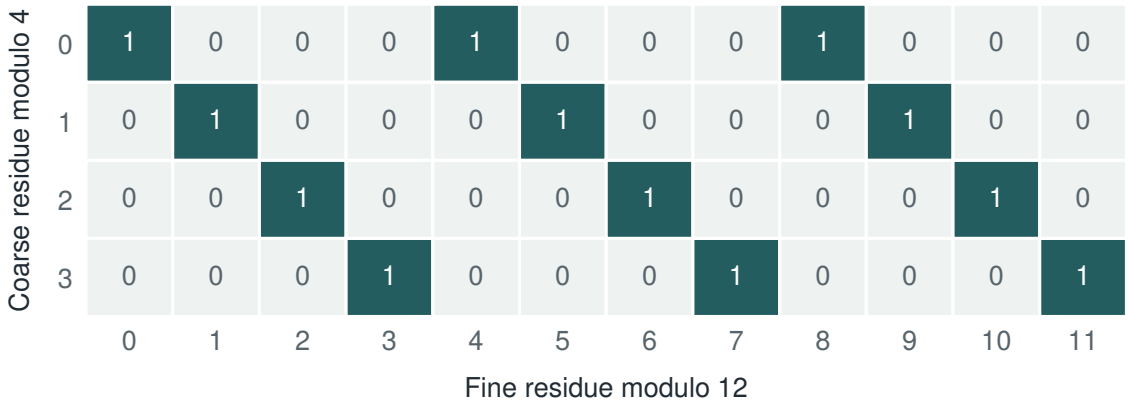


Figure 2. The incidence matrix $M_{r,s} = 1$ when the fine residue s belongs to the coarse residue r. Its rows implement exact aggregation from fine packets to coarse packets.

for all α and β. Remove the zero operators from the family

$$R_{\alpha\beta} = P_\alpha Q_\beta.$$

The remaining operators form a packet system that refines both $\mathcal{P}$ and $\mathcal{Q}$. The same statement holds for any finite collection of pairwise commuting packet systems.

Proof. Commutativity gives

$$R_{\alpha\beta}^2 = P_\alpha^2 Q_\beta^2 = R_{\alpha\beta}.$$

If $(\alpha, \beta) \neq (\alpha', \beta')$, then either $P_\alpha P_{\alpha'} = 0$ or $Q_\beta Q_{\beta'} = 0$, so the corresponding products annihilate one another. Finally,

$$\sum_{\alpha,\beta} R_{\alpha\beta} = \left(\sum_\alpha P_\alpha\right)\left(\sum_\beta Q_\beta\right) = I_X.$$

For a fixed α,

$$P_\alpha = P_\alpha \sum_\beta Q_\beta = \sum_\beta R_{\alpha\beta},$$

and the same argument works for a fixed β. Since each P_α and Q_β is nonzero, these identities ensure that every parent has a nonzero product child. The two parent maps on the supported product family are therefore surjective. Iteration proves the finite family statement. For the empty collection, use the supported identity family: (I_X) when $X \neq 0$ and the empty family when $X = 0$. □

Proposition 3.11 (Coarsest common refinement)**.** *Under the hypotheses of theorem 3.10, let $\mathcal{R}$ be a packet system that refines both $\mathcal{P}$ and $\mathcal{Q}$. Then $\mathcal{R}$ refines the nonzero product family $(P_\alpha Q_\beta)$. Hence that product family is the coarsest packet system that refines both $\mathcal{P}$ and $\mathcal{Q}$.*

Proof. Let R_γ be a packet in $\mathcal{R}$. Because $\mathcal{R}$ refines $\mathcal{P}$, there is a unique α with

$$P_\alpha R_\gamma = R_\gamma.$$

Likewise there is a unique β with

$$Q_\beta R_\gamma = R_\gamma.$$

Therefore

$$P_\alpha Q_\beta R_\gamma = R_\gamma.$$

Every packet of $\mathcal{R}$ lies in one product packet. Multiplying $\sum_\gamma R_\gamma = I_X$ by $P_\alpha Q_\beta$ and using the parent identities gives

$$P_\alpha Q_\beta = \sum_{\substack{\gamma:\ P_\alpha R_\gamma = R_\gamma \\ Q_\beta R_\gamma = R_\gamma}} R_\gamma.$$

Thus the packets with the same pair of parents reconstruct the product packet. If that product packet is nonzero, the sum has a child, so the parent map onto the supported product family is surjective. □

Example 3.12 (A two level split)**.** Let

$$X = X_{00} \oplus X_{01} \oplus X_{10} \oplus X_{11}.$$

Assume that all four summands are nonzero. The four coordinate projections form a fine packet system. The two sums

$$P_0 = Q_{00} + Q_{01}, \qquad P_1 = Q_{10} + Q_{11}$$

form a coarse packet system. If $x = (x_{00}, x_{01}, x_{10}, x_{11})$, the coarse coordinates are

$$(x_{00} + x_{01}, x_{10} + x_{11})$$

when the sums are interpreted inside the two coarse direct summands. Fine analysis recovers the four entries exactly. Nothing has been approximated.

4 Residue coordinates and coprime factorization

Residue packets sort a formal series by the congruence classes of its exponents. This construction works over every coefficient ring. The character formula describes the same selectors through an operator average and needs stronger hypotheses.

Definition 4.1 (Residue selector)**.** Let R be a commutative ring and let

$$F(z) = \sum_{n \geq 0} a_n z^n \in R[[z]].$$

For $N \geq 1$ and $0 \leq j < N$, define

$$(P_{N,j} F)(z) = \sum_{n \equiv j \pmod N} a_n z^n.$$

Proposition 4.2 (Residue packet algebra)**.** *For every commutative ring R, the operators $(P_{N,j})_{0\le j<N}$ satisfy the identities below. If R is nonzero, they form a packet system on $R[[z]]$:*

$$P_{N,i}P_{N,j} = \delta_{ij}P_{N,j}, \qquad \sum_{j=0}^{N-1} P_{N,j} = I.$$

Proof. Every nonnegative integer belongs to one residue class modulo N. A coefficient survives both $P_{N,i}$ and $P_{N,j}$ exactly when the two labels are equal to its residue. Summing all selectors preserves every coefficient once. □

4.1 Finite orthogonality certificates

The coefficient-mask definition of $P_{N,j}$ also makes sense over a commutative semiring: it only retains or sets coefficients to zero. We use that same definition in the following semiring statement.

Let U act on coefficient sequences by

$$(UF)(z) = F(\omega z).$$

If $\omega^N = 1$, then $U^N = I$.

Theorem 4.3 (Root-of-unity realization)**.** *Let R be a commutative semiring, let $N > 0$, and suppose $\omega, \omega^\vee, \eta \in R$ satisfy $\omega^N = 1$ and the finite $N \times N$ Fourier orthogonality table*

$$\eta \sum_{k=0}^{N-1} (\omega^\vee)^{jk}\omega^{rk} = \begin{cases} 1, & r = j, \\ 0, & r \neq j, \end{cases} \qquad 0 \le j, r < N.$$

Then

$$\boxed{P_{N,j} = \eta \sum_{k=0}^{N-1} (\omega^\vee)^{jk} U^k.}$$

In particular, if R is a field, $N > 0$, and ω is a primitive Nth root of unity, the certificate holds with $\omega^\vee = \omega^{-1}$ and $\eta = N^{-1}$.

Proof. On the monomial z^n, the right-hand side acts by

$$\eta \sum_{k=0}^{N-1} (\omega^\vee)^{jk}\omega^{nk}.$$

The identity $\omega^N = 1$ reduces this expression to the table entry indexed by $r = n \bmod N$. The finite certificate therefore gives exactly the coefficient mask defining $P_{N,j}$. Over a field, the primitive-root geometric sum supplies the certificate. □

The field or orthogonality condition is essential. A primitive element of order N in a ring with zero divisors does not by itself force the geometric sums away from the diagonal to vanish. For example, in $\mathbb{F}_3 \times \mathbb{F}_3$ the element $(-1, 1)$ has order two, but the second component of its nontrivial character sum is nonzero. The coefficient selector uses no division and remains valid without any Fourier certificate. This distinction becomes decisive at the dyadic prime.

4.2 The field specialization

Theorem 4.4 (Character realization)**.** *Let $\Bbbk$ be a field, assume that N is nonzero in $\Bbbk$, and suppose that $\Bbbk$ contains a primitive Nth root of unity ω. Define*

$$(UF)(z) = F(\omega z).$$

Then

$$P_{N,j} = \frac{1}{N} \sum_{k=0}^{N-1} \omega^{-jk} U^k.$$

Proof. On z^n, the right side acts by the scalar

$$\frac{1}{N} \sum_{k=0}^{N-1} \omega^{k(n-j)}.$$

This is one when $n \equiv j \pmod N$ and zero otherwise by the geometric sum identity in the field $\Bbbk$. □

Remark 4.5 (Two distinct constructions). The coefficient selector of definition 4.1 uses no division. The character formula in theorem 4.4 divides by N. At a prime that divides N, the coefficient selector may remain integral even when the character average does not.

Theorem 4.6 (Nested residue refinement)**.** *Let N, M be positive integers with N dividing M, and let $0 \le j < N$. Then*

$$P_{N,j} = \sum_{\substack{0 \le s < M \\ s \equiv j \pmod N}} P_{M,s}.$$

When R is nonzero, the packet system modulo M refines the packet system modulo N.

Proof. An exponent n satisfies $n \equiv j \pmod N$ exactly when its unique residue s modulo M satisfies $s \equiv j \pmod N$. □

Example 4.7 (The shift graph of residue packets)**.** Assume that R is nonzero and $N > 0$. Multiplication by z sends the residue packet $P_{N,j}R[[z]]$ into $P_{N,j+1}R[[z]]$, where the index is read modulo N. Its packet graph is the directed cycle

$$0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow N-1 \longrightarrow 0.$$

Multiplication by a polynomial supported on a set of residues gives the corresponding finite set of phase shifts. This is the simplest transfer graph behind a polyphase representation.

Theorem 4.8 (Coprime factorization)**.** *Let A and B be coprime positive integers. For each j modulo AB, write*

$$j_A = j \pmod A, \qquad j_B = j \pmod B.$$

Then

$$P_{AB,j} = P_{A,j_A} P_{B,j_B}.$$

Every pair (j_A, j_B) occurs for one residue j modulo AB.

Proof. A coefficient survives the product on the right exactly when its exponent satisfies both congruences. The Chinese remainder theorem gives one residue class modulo AB with those two coordinates. The needed form of the theorem is standard and may be found in [5]. □

Corollary 4.9 (Dyadic and odd coordinates)**.** *Write $N = 2^a M$ with M odd. Every residue packet modulo N is the product of one packet modulo 2^a and one packet modulo M.*

Proof. Apply theorem 4.8 to the coprime moduli 2^a and M. □

Example 4.10 (Residues modulo twelve)**.** The pair

$$(j \pmod 4, j \pmod 3) = (3, 1)$$

corresponds to $j = 7$ modulo 12. Hence

$$P_{12,7} = P_{4,3} P_{3,1}.$$

This factorization concerns addresses. It does not say that the two factors have the same integral spectral behavior.

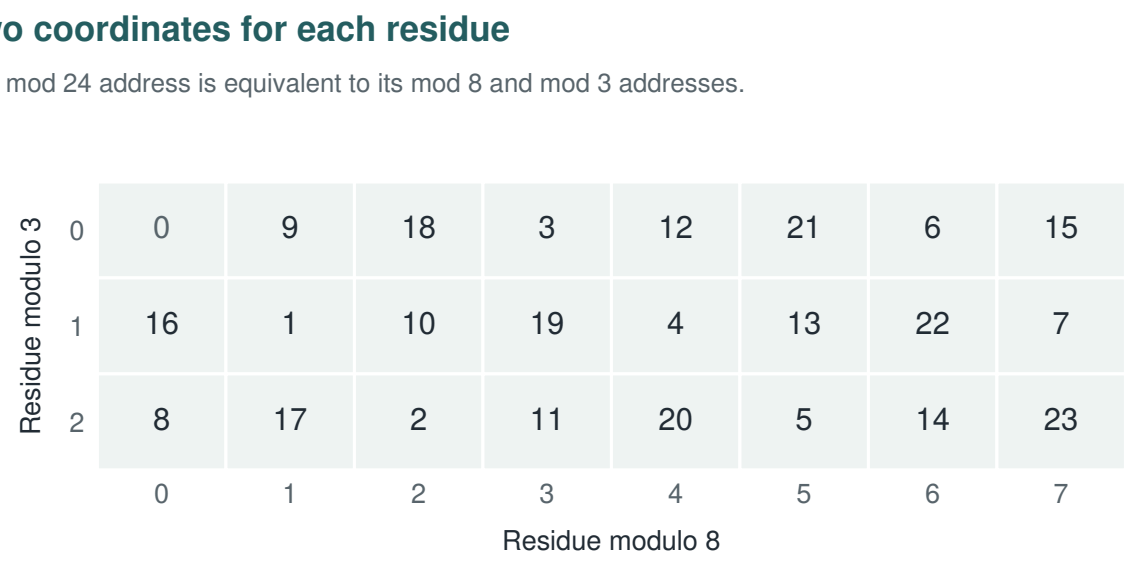


Figure 3. A composite residue packet is addressed by its dyadic and odd coordinates. The displayed mod-24 system is the product of mod 8 and mod 3.

5 Complete filtered transfer algebras

A filtration records how deeply a term lies in the chosen notion of precision. The weights record guaranteed filtration gains that add under multiplication. A discrete valuation ring provides the main example.

Definition 5.1 (Complete multiplicative filtration)**.** A complete filtered algebra is a unital algebra $\mathcal{B}$ with a decreasing sequence of two-sided ideals

$$\mathcal{B} = F^0\mathcal{B} \supseteq F^1\mathcal{B} \supseteq F^2\mathcal{B} \supseteq \cdots$$

such that

$$F^r\mathcal{B}\,F^s\mathcal{B} \subseteq F^{r+s}\mathcal{B}, \qquad \bigcap_{L\geq 0} F^L\mathcal{B} = 0,$$

and the natural map

$$\mathcal{B} \longrightarrow \varprojlim_L \mathcal{B}/F^L\mathcal{B}$$

is an isomorphism. A transfer term N_e has a declared positive integer weight $w(e) \in \mathbb{Z}_{>0}$ when

$$N_e \in F^{w(e)}\mathcal{B}.$$

In the filtration topology, reduction to $\mathcal{B}/F^L\mathcal{B}$ is continuous when the quotient has the discrete topology. Its kernel $F^L\mathcal{B}$ is therefore closed. In particular, a convergent sum of terms in $F^L\mathcal{B}$ remains in that ideal.

If $\mathcal{O}$ is a complete discrete valuation ring with uniformizer π and $\mathcal{B}$ is a separated and complete $\mathcal{O}$ algebra for the filtration

$$F^L\mathcal{B} = \pi^L\mathcal{B},$$

then this is a standard example. Finite matrix algebras over $\mathcal{O}$ satisfy these conditions. Completeness of $\mathcal{O}$ alone does not imply completeness of every $\mathcal{O}$ algebra. Basic facts about completeness in this setting are reviewed in [8].

Theorem 5.2 (Weighted inverse)**.** *Let $\mathcal{B}$ be a complete filtered algebra. Let $D \in \mathcal{B}$ be invertible with*

$$D, D^{-1} \in F^0\mathcal{B}.$$

Let E be a finite set and suppose

$$T = D + \sum_{e\in E} N_e, \qquad N_e \in F^{w(e)}\mathcal{B}, \qquad w(e) \geq 1.$$

Set

$$A_e = D^{-1}N_e.$$

For a word $\gamma = (e_1, \ldots, e_r)$, write

$$A_\gamma = A_{e_r}\cdots A_{e_1}, \qquad W(\gamma) = \sum_{j=1}^{r} w(e_j).$$

The empty word has weight zero and contributes the identity. Then T is invertible and

$$T^{-1} = \sum_{\gamma} (-1)^{|\gamma|} A_\gamma D^{-1}.$$

The series converges in the filtration, and the term indexed by γ belongs to $F^{W(\gamma)}\mathcal{B}$.

Proof. Put

$$A = \sum_{e \in E} A_e.$$

Since left multiplication by D^{-1} preserves the filtration, $A \in F^1\mathcal{B}$. Completeness gives the convergent Neumann series

$$(I + A)^{-1} = \sum_{r \geq 0} (-A)^r.$$

Because $T = D(I + A)$,

$$T^{-1} = (I + A)^{-1} D^{-1}.$$

Expanding each power of A gives the sum over words, with the rightmost edge acting first. Multiplicativity of the filtration gives

$$A_\gamma \in F^{W(\gamma)}\mathcal{B}.$$

For each L, only finitely many words have weight below L because E is finite and every edge has positive weight. Thus the word expansion converges to the same inverse. □

The preceding formula does not yet use packet blocks. When D is packet diagonal and the terms N_e are individual transfers, only words with compatible packet endpoints can contribute. These words become directed paths.

Theorem 5.3 (Block path formula). *Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ be a finite packet system in a complete filtered algebra of operators. Suppose*

$$D = \sum_\alpha P_\alpha D P_\alpha$$

belongs to $\mathcal{B}$ and is invertible on every packet. Assume that the assembled inverse

$$D^{-1} = \sum_\alpha D_\alpha^{-1} P_\alpha \in \mathcal{B} = F^0\mathcal{B}, \qquad D_\alpha = P_\alpha D P_\alpha,$$

where each packet inverse is extended by zero to the other packets. Let the edge set be finite, assume $w(e) \geq 1$ for every edge, and write

$$N = \sum_{e : \alpha \to \beta} N_e, \qquad N_e = P_\beta N_e P_\alpha, \qquad N_e \in F^{w(e)}\mathcal{B}.$$

For an edge $e : \alpha \to \beta$, put

$$A_e = (P_\beta D P_\beta)^{-1} N_e.$$

Then the block from $P_\alpha X$ to $P_\beta X$ of $(D + N)^{-1}$ is

$$P_\beta (D + N)^{-1} P_\alpha = \sum_{\gamma : \alpha \rightsquigarrow \beta} (-1)^{|\gamma|} A_\gamma D_\alpha^{-1},$$

where $D_\alpha = P_\alpha D P_\alpha$ and the sum runs over directed paths from α to β, including the empty path when $\alpha = \beta$.

Proof. The diagonal inverse is

$$D^{-1} = \sum_{\alpha} D_{\alpha}^{-1} P_{\alpha}.$$

The block $P_{\beta} D^{-1} N P_{\alpha}$ is nonzero only when there is a declared edge from α to β. Repeated block multiplication shows that the rth power of $D^{-1}N$ is the sum over all directed paths of length r, with the operator factors ordered along the path. Apply theorem 5.2 and multiply on the right by the packet diagonal inverse. □

Corollary 5.4 (Minimum path weight)**.** *For packets α and β, let $d(\alpha, \beta)$ be the least total weight of a directed path from α to β, with $d = \infty$ when no path exists. Use the convention $F^{\infty}\mathcal{B} = \bigcap_{L \geq 0} F^{L}\mathcal{B} = 0$. If $\alpha \neq \beta$, then*

$$P_{\beta}(D + N)^{-1} P_{\alpha} \in F^{d(\alpha,\beta)}\mathcal{B}.$$

If there is no path from α to β, the block is zero.

Proof. Every term in the block path formula is indexed by a path from α to β and lies in the filtration level given by its total weight. If no such path exists, the sum is empty. □

Example 5.5 (A longer path that appears earlier)**.** Consider three packets α, β, and γ. Suppose there is a direct edge from α to γ of weight three, together with edges

$$\alpha \longrightarrow \beta, \qquad \beta \longrightarrow \gamma$$

of weight one each. The two edge route has total weight two, so it contributes to the inverse before the direct edge does. This is why the truncation rule uses total weight rather than path length. The lower bound in corollary 5.4 need not be an equality because several paths of the same weight may cancel.

> **The central idea**
>
> Word length counts how many transfers are composed. Weight counts how much filtration gain those transfers guarantee. The sign in the inverse expansion depends on length, while the precision cutoff depends on weight. A short word can therefore disappear at a chosen precision while a longer word still contributes.

Corollary 5.6 (Acyclic transfer graph)**.** *If the packet transfer graph is acyclic, then $D^{-1}N$ is nilpotent. The inverse in theorem 5.3 is a finite sum and does not require completeness of the filtration.*

Proof. A finite acyclic directed graph has a bound on path length. Every sufficiently long product of transfer blocks is therefore zero. The Neumann series terminates. □

Corollary 5.7 (Stability of inversion at finite precision)**.** *Let T and $\widetilde{T}$ be invertible elements of $F^{0}\mathcal{B}$. If*

$$T - \widetilde{T} \in F^{L}\mathcal{B},$$

then

$$T^{-1} - \widetilde{T}^{-1} \in F^{L}\mathcal{B}.$$

Proof. Use the resolvent identity at zero in the form

$$T^{-1} - \widetilde{T}^{-1} = T^{-1}(\widetilde{T} - T)\widetilde{T}^{-1}.$$

Both inverse factors belong to $F^0\mathcal{B}$, so the product lies in $F^L\mathcal{B}$. □

Corollary 5.8 (Exact finite precision truncation)**.** *Under the hypotheses of theorem 5.2, define*

$$R_L = \sum_{W(\gamma)<L} (-1)^{|\gamma|} A_\gamma D^{-1}.$$

Then

$$T^{-1} - R_L \in F^L\mathcal{B},$$

$$TR_L - I \in F^L\mathcal{B}, \qquad R_L T - I \in F^L\mathcal{B}.$$

Thus R_L is the exact inverse in $\mathcal{B}/F^L\mathcal{B}$.

Proof. Every omitted word has weight at least L, so the difference from the full inverse lies in $F^L\mathcal{B}$. Multiplication by T or by its inverse preserves membership in $F^L\mathcal{B}$ under the stated filtration assumptions. The two residual statements follow. □

Example 5.9 (A triangular transfer)**.** Let $\mathcal{O}$ be complete with uniformizer π, and consider

$$T = \begin{pmatrix} 1 & \pi^a \\ 0 & 1 \end{pmatrix}$$

with $a \geq 1$. The only nontrivial transfer has weight a, and its square is zero. Hence

$$T^{-1} = \begin{pmatrix} 1 & -\pi^a \\ 0 & 1 \end{pmatrix}.$$

At precision $L \leq a$, the transfer vanishes and the inverse is the identity modulo π^L. At depth $a+1$, the transfer between the two packets first becomes visible.

Two compatible word conventions. The path expansion above labels a word in chronological order and writes its operator product in reverse order, so the first edge acts first. The following finite-word contract uses written multiplication order. Reversing all word labels identifies the sums over all words. Individual path labels must retain their stated convention. The two conventions do not change a weight or an inverse.

5.1 Finite quotient and normed-ring realizations

5.1.1 Normalized edges and written-word convention

Let $\mathcal{O}$ be a complete discrete valuation ring with uniformizer π, fraction field K, and filtration

$$F^m\mathcal{B} = \pi^m\mathcal{B}$$

on an $\mathcal{O}$-algebra $\mathcal{B}$. The finite quotient statements below do not require $\mathcal{B}$ itself to be complete. Suppose

$$T = D + H,$$

where D is invertible and the perturbation H raises filtration. The symbol K is reserved for the fraction field.

A packet decomposition resolves $H = \sum_{e \in E} H_e$ and assigns a certified filtration gain

$$D^{-1}H_e \in F^{w(e)}\mathcal{B}, \qquad w(e) \in \mathbb{Z}_{>0}.$$

Each e is a directed transfer edge. For an ordered transfer word

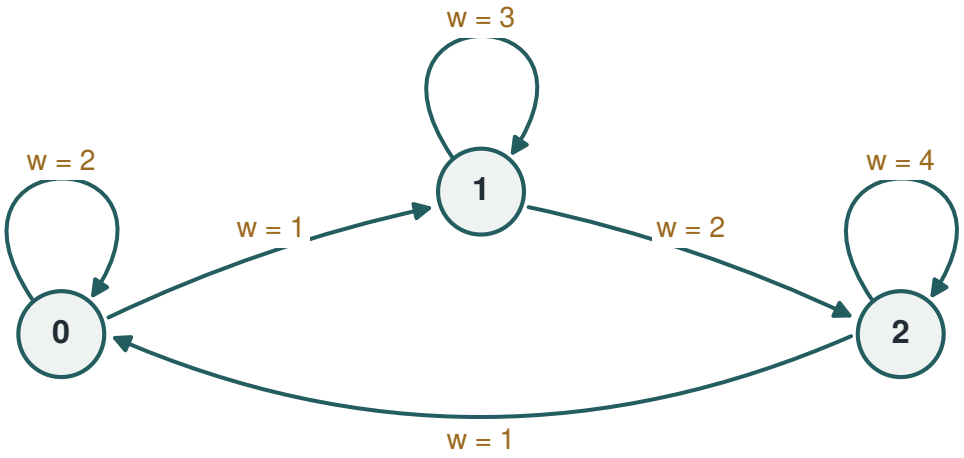
$$\gamma = (e_1, \dots, e_r),$$

define

$$W(\gamma) = \sum_{j=1}^{r} w(e_j), \qquad A_\gamma = A_{e_1} \cdots A_{e_r}, \qquad A_e = D^{-1}H_e.$$

Weighted transfer graph

Edge weights measure filtration gain. Every edge has multiplicity one.



The weighted walk series below starts at packet 0 and ends at packet 2.

Figure 4. A finite packet-transfer graph. Edge weights record filtration gain rather than merely the number of operator compositions.

Theorem 5.10 (Certified weighted inverse). *Let R be a possibly noncommutative ring, let D^{-1} be a supplied two-sided inverse of D, and put $A = D^{-1}H$.*

(a) If $A^r = 0$, then $D + H$ has the two-sided inverse

$$\boxed{(D+H)^{-1} = \left(\sum_{k=0}^{r-1} (-A)^k\right) D^{-1}.}$$

(b) If R is a complete normed ring and $\|A\| < 1$, then $\sum_{k\geq 0}(-A)^k$ converges and

$$\boxed{(D+H)^{-1} = \left(\sum_{k\geq 0}(-A)^k\right) D^{-1}.}$$

Proof. Factor

$$T = D(I+A), \qquad A = D^{-1}H.$$

For a finite cutoff r, multiplication on either side gives the noncommutative telescoping identity

$$(I+A)\sum_{k=0}^{r-1}(-A)^k = \left(\sum_{k=0}^{r-1}(-A)^k\right)(I+A) = 1-(-A)^r.$$

This proves (a). In (b), the strict norm bound makes the geometric series summable. Continuity of multiplication passes the same identities to the limit. Multiplication by D^{-1} gives the displayed inverse in both orders. □

Corollary 5.11 (Finite precision). *Let $F^\bullet R$ be a multiplicative decreasing filtration and let $q_L : R \to R_L$ be a ring homomorphism that kills $F^L R$. Suppose*

$$H = \sum_{e\in E} H_e, \qquad D^{-1}H_e \in F^{w(e)}R, \qquad w(e) \geq 1,$$

with E finite. Then $q_L(D^{-1}H)^L = 0$, the finite sum in theorem 5.10(a) is a two-sided inverse after applying q_L, and its word expansion may discard every word of weight at least L. More explicitly, for a length cutoff N, define

$$S^{\mathrm{pr}}_{L,N} = \sum_{n=0}^{N-1}(-1)^n \sum_{\substack{\gamma\in E^n \\ W(\gamma)<L}} A_\gamma.$$

Lean proves directly that $q_L(S^{\mathrm{pr}}_{L,N})$ equals the mapped geometric inverse whenever $q_L(A^N) = 0$, and hence is a two-sided inverse of $q_L(I+A)$, where $A = \sum_e A_e = D^{-1}H$. With the supplied two-sided inverse D^{-1} of D, this gives the explicit two-sided inverse

$$\boxed{q_L\left(S^{\mathrm{pr}}_{L,N}D^{-1}\right)} \quad \textit{of} \quad q_L(D+H).$$

If every edge weight is at least m and $L \leq mN$, then the nilpotence condition follows from the filtration data because $m|\gamma| \leq W(\gamma)$. For $m > 0$, the executable choice $N = \lceil L/m\rceil$ satisfies $L \leq mN$. This elementary ceiling inequality is also proved in Lean. The choice $m = 1, N = L$ gives the standard cutoff. In the DVR example $q_L : R \to R/\pi^L R$, this reads

$$q_L(T)^{-1} = q_L\left(\sum_{W(\gamma)<L}(-1)^{|\gamma|}A_\gamma D^{-1}\right).$$

Here $A_\gamma = (D^{-1}H_{e_1})\cdots(D^{-1}H_{e_r})$ in the written order. Multiplicativity places it in $F^{W(\gamma)}R$, so the quotient kills it when $W(\gamma) \geq L$. Positive weights also imply $|\gamma| \leq W(\gamma)$, making the retained search finite.

How to read the result

Using the standard length cutoff, every discarded word has weight at least the chosen depth and vanishes in the quotient. The retained sum therefore satisfies both inverse identities exactly at that precision. The earlier truncation theorem states the corresponding residual bounds in the complete filtered algebra itself.

The inherited finite-word contract proves the result in the finite quotients above and in the contraction specialization for complete normed rings. The complete filtered statement has its own separation and inverse-limit completeness hypotheses. Its Lean coverage is recorded separately in the correspondence.

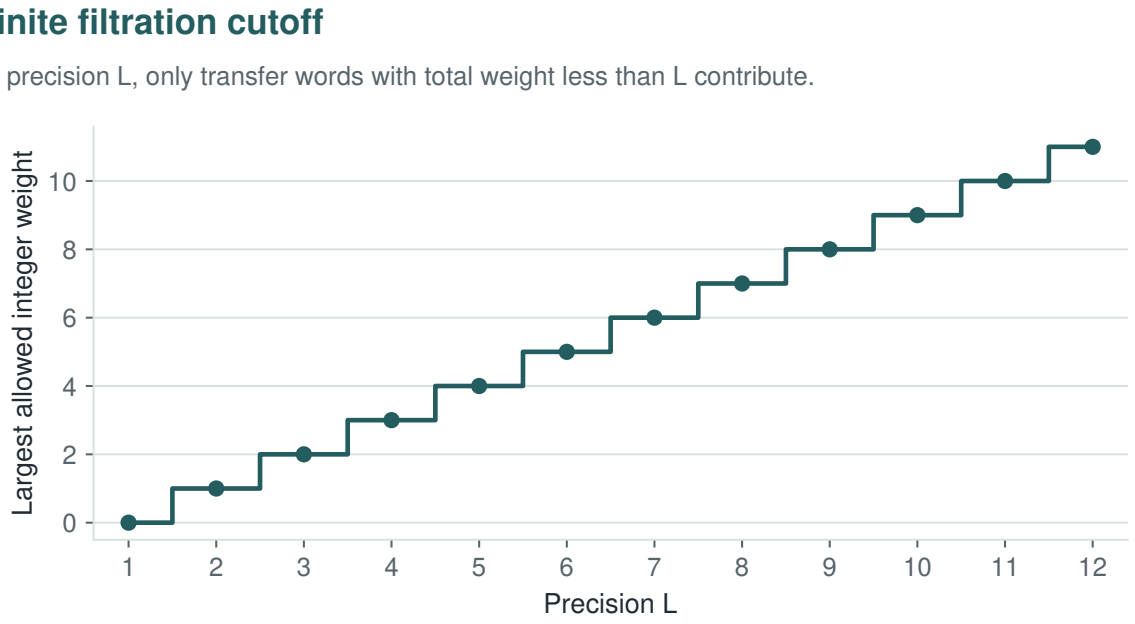


Figure 5. At precision L, weighted inversion requires only transfer words whose cumulative weight is below L.

6 Rational weighted transfer series

A finite weighted graph records which path terms can occur in the inverse. Collecting paths by total weight turns this support information into a generating function. The finite adjacency matrix makes that function rational.

Definition 6.1 (Weighted transfer series)**.** Let G be a finite directed multigraph. Give every edge e a positive integer weight $w(e)$ and a scalar multiplicity $m(e)$ in a field $\Bbbk$. Define the polynomial adjacency matrix

$$A(t)_{ij} = \sum_{e:i\to j} m(e)t^{w(e)}.$$

For column vectors u and v, define

$$Z_{u,v}(t) = \sum_{n\geq 0} a_n t^n,$$

where

$$a_n = \sum_{i,j} \sum_{\substack{\gamma: i\rightsquigarrow j \\ W(\gamma)=n}} u_i v_j \prod_{\nu=1}^{|\gamma|} m(e_\nu).$$

Here $\gamma = (e_1, \dots, e_{|\gamma|})$ is a directed walk. Vertices and edges may repeat, and the empty walk has weight zero and multiplicity one. Thus the endpoint vectors weight the start and end of each walk, rather than only selecting subsets of vertices.

Theorem 6.2 (Rationality of the transfer series)**.** *The weighted transfer series satisfies*

$$Z_{u,v}(t) = u^{\mathsf{T}}(I - A(t))^{-1}v.$$

It is a rational function whose denominator divides

$$\det(I - A(t)).$$

Proof. Since every entry of $A(t)$ has zero constant term, the inverse exists in the formal power series ring and

$$(I - A(t))^{-1} = \sum_{r\geq 0} A(t)^r.$$

The (i, j) entry of $A(t)^r$ is the sum of the monomials attached to all paths of length r from i to j. Summing over r records paths of every length. Multiplication by the endpoint vectors gives the stated series. The adjugate formula gives

$$(I - A(t))^{-1} = \frac{\operatorname{adj}(I - A(t))}{\det(I - A(t))},$$

so every entry is rational. □

Corollary 6.3 (Recurrence for path counts)**.** *Write*

$$Z_{u,v}(t) = \frac{P(t)}{Q(t)}, \qquad Q(t) = 1 + q_1 t + \cdots + q_d t^d.$$

Use the convention $\deg 0 = -\infty$*. Then, for every nonnegative integer* $n > \deg P$*,*

$$a_n + q_1 a_{n-1} + \cdots + q_d a_{n-d} = 0.$$

Proof. Compare coefficients in the identity

$$Q(t)Z_{u,v}(t) = P(t).$$

Beyond the degree of P, the coefficient on the right is zero. □

In the recurrence, set $a_k = 0$ when $k < 0$. This is the coefficient convention for the formal power series $Z_{u,v}(t)$ and makes the first recurrence equations meaningful even when $n < d$.

Corollary 6.4 (Several independent weights)**.** *Suppose every edge has a weight vector*

$$w(e) = (w_1(e), \dots, w_r(e)) \in \mathbb{N}^r \setminus \{0\}.$$

Replace $t^{w(e)}$ by

$$t_1^{w_1(e)} \cdots t_r^{w_r(e)}.$$

The resulting multigraded transfer series is rational in $t_1, \dots, t_r$.

Proof. The same matrix inverse proof works over the formal power series ring in several variables. The adjugate formula gives rationality. □

The multigraded form is useful when dyadic depth, odd depth, and another filtration are tracked separately rather than collapsed into one number.

Example 6.5 (A transfer series with one loop)**.** Take vertices 0, 1, and 2. Give the edge $0 \to 1$ weight one, the loop $1 \to 1$ weight two, and the edge $1 \to 2$ weight one. Every path from 0 to 2 uses the first and last edges and may traverse the loop any number of times. Hence

$$Z_{0,2}(t) = t^2(1 + t^2 + t^4 + \cdots) = \frac{t^2}{1 - t^2}.$$

There is one path at every positive even weight and no path at an odd weight. The denominator records the increase of two units produced by the loop.

6.1 Automatic finite-graph construction

Let G be a finite directed graph with positive integer edge weights. Define the polynomial adjacency matrix

$$A(t)_{ij} = \sum_{e : i \to j} t^{w(e)},$$

where parallel edges are distinct elements of the finite edge type and are therefore counted with their full multiplicity.

For fixed start and target vertices s, t, define

$$Z_{s,t}(t) = \sum_{n \geq 0} a_n(s,t) t^n,$$

where $a_n(s,t)$ counts directed walks from s to t with total weight n. Vertices and edges may repeat. Thus "path zeta" is retained as the name of the series and Lean API, but the walks are not required to visit each vertex only once.

Theorem 6.6 (Automatic finite-graph transfer-zeta rationality). *Assume the vertex and edge sets are finite and every edge has positive integer weight. For every n, exact enumeration of edge tuples of lengths at most n, followed by endpoint and total-weight filtering, returns $a_n(s,t)$. These coefficients obey the recurrence obtained by removing the first edge*

$$a_n(s,t) = \mathbf{1}_{s=t,\, n=0} + \sum_{\substack{e:\, \mathrm{source}(e)=s \\ w(e)\le n}} a_{n-w(e)}(\mathrm{target}(e), t).$$

Let

$$T(t) = I - A(t), \qquad Q(t) = \det T(t), \qquad P_{s,t}(t) = \mathrm{adj}(T(t))_{s,t}.$$

Then $Q(0) = 1$, hence $Q \neq 0$ and Q is a unit in $\mathbb{Z}[[t]]$, and the independently enumerated walk series satisfies

$$\boxed{Z_{s,t}(t) = Q(t)^{-1} P_{s,t}(t), \qquad Q(t) Z_{s,t}(t) = P_{s,t}(t)} \quad \text{in } \mathbb{Z}[[t]].$$

Proof. Every edge has weight at least one, so a word of weight n has length at most n. The enumerator therefore searches a finite set containing every contributing walk, and its filters are exactly the defining predicates for a_n. Removing the first edge gives the displayed disjoint decomposition into the empty diagonal walk and the first-edge tails. In matrix form this is

$$Z = I + AZ, \qquad (I - A)Z = I.$$

Positive weights give $A(0) = 0$, so $T(0) = I$ and $Q(0) = \det(I) = 1$. The two adjugate identities

$$T\,\mathrm{adj}(T) = QI, \qquad \mathrm{adj}(T)T = QI$$

remain valid after inclusion into $\mathbb{Z}[[t]]$. Multiplying the adjugate by the formal-series inverse of Q constructs a two-sided inverse of T. Uniqueness of the inverse identifies that construction with the independently enumerated matrix Z, yielding the entrywise formulas. □

This is the finite transfer-matrix method of [9], but here its determinant, adjugate, normalization, recurrence from the first edge, and identification with the enumerated series are all formalized. The older API accepting an independently supplied normalized coefficient identity $QZ = P$ is retained for external certificates. It is no longer needed to obtain rationality for the finite graph defined above. The Python companion performs an exact symbolic construction as an independent executable check.

Corollary 6.7 (Linear recurrence). *With the canonical polynomials from theorem 6.6, write*

$$Q(t) = \sum_k q_k t^k, \qquad Z_{s,t}(t) = \sum_{n\ge 0} a_n(s,t) t^n.$$

Then the walk counts satisfy the recurrence with constant coefficients determined by Q after the finite initial range contributed by $P_{s,t}$, with $\deg 0 = -\infty$:

$$\boxed{\sum_{k=0}^{n} q_k a_{n-k}(s,t) = 0 \qquad (n > \deg P_{s,t}).}$$

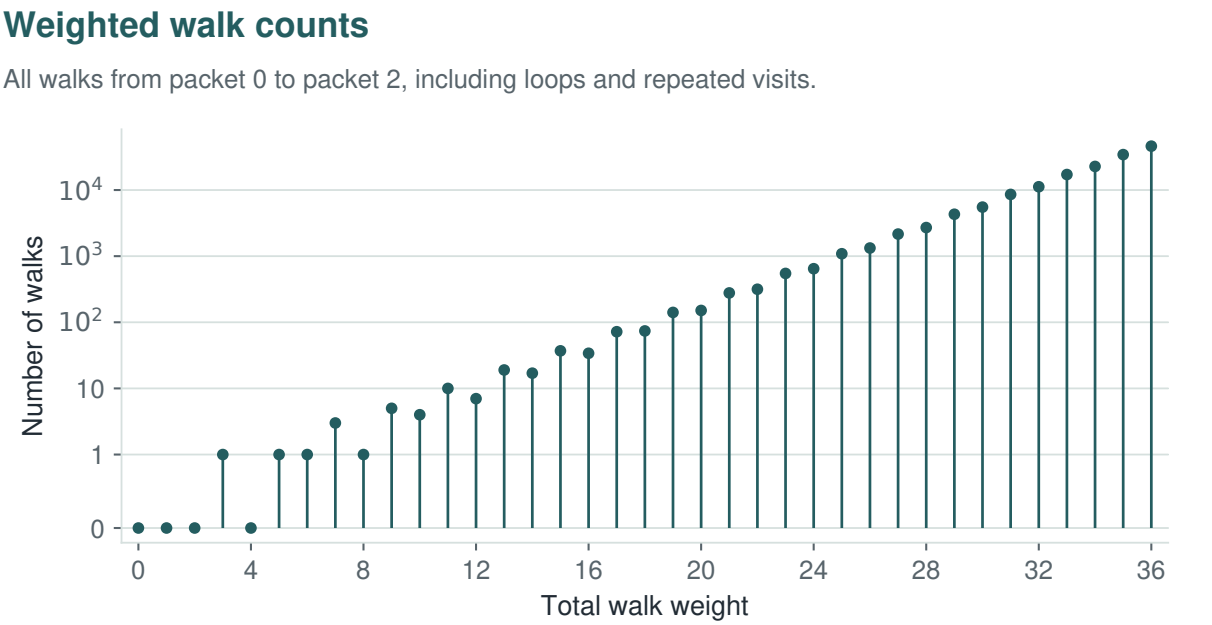


Figure 6. The walk counts for one weighted graph. The exact generating function is rational, so these counts satisfy a finite linear recurrence.

Example 6.8 (A computed transfer zeta)**.** For the graph used in the numerical companion, the walk series from packet 0 to packet 2 is

$$Z_{0,2}(t) = \frac{t^3}{1 - t^2 - t^3 - 2t^4 + t^5 + t^6 + t^7 - t^9}.$$

The first nonzero coefficients are

$$1, 1, 1, 3, 1, 5, 4, 10, 7, 19, 17, 37, \ldots$$

at their corresponding weights. Direct enumeration through weight 30 agrees coefficient by coefficient with the rational expansion.

7 Rational spectral projectors and resultant bounds

Let $\mathcal{O}$ be a discrete valuation ring with fraction field K and uniformizer π. Let Λ be a finite free $\mathcal{O}$ module and put

$$V = K \otimes_{\mathcal{O}} \Lambda.$$

Suppose $A \in \mathrm{End}_{\mathcal{O}}(\Lambda)$.

Theorem 7.1 (Bézout spectral projector)**.** *Assume that*

$$\chi_A(x) = f(x)g(x)$$

with monic polynomials $f, g \in \mathcal{O}[x]$ that are coprime in $K[x]$. Choose $u, v \in \mathcal{O}[x]$ and a nonzero $r \in \mathcal{O}$ such that

$$u(x)f(x) + v(x)g(x) = r.$$

Then

$$E_f = r^{-1}v(A)g(A)$$

is the projector of V onto Ker $f(A)$ *along* Ker $g(A)$. *In particular,*

$$E_f^2 = E_f, \qquad rE_f \in \mathrm{End}_{\mathcal{O}}(\Lambda).$$

Proof. Cayley Hamilton gives

$$f(A)g(A) = 0.$$

If $x \in \mathrm{Ker}\, f(A)$, then the Bézout identity gives

$$v(A)g(A)x = rx,$$

so $E_f x = x$. If $x \in \mathrm{Ker}\, g(A)$, then $E_f x = 0$. The same identity gives the direct sum decomposition

$$V = \mathrm{Ker}\, f(A) \oplus \mathrm{Ker}\, g(A).$$

Indeed, every $x \in V$ is the sum of

$$r^{-1}v(A)g(A)x \in \mathrm{Ker}\, f(A)$$

and

$$r^{-1}u(A)f(A)x \in \mathrm{Ker}\, g(A).$$

Thus E_f is the stated projector and is idempotent. The formula for rE_f has coefficients in $\mathcal{O}$, so it preserves Λ. □

Theorem 7.2 (Several coprime spectral packets)**.** *Assume*

$$\chi_A(x) = \prod_{i=1}^{r} f_i(x),$$

where the monic factors $f_i \in \mathcal{O}[x]$ are pairwise coprime in $K[x]$. Put

$$g_i = \prod_{j \neq i} f_j.$$

There are complementary projectors E_i with

$$\mathrm{im}\, E_i = \mathrm{Ker}\, f_i(A), \qquad \sum_i E_i = I, \qquad E_i E_j = 0 \quad (i \neq j).$$

One may choose E_i so that

$$\mathrm{Res}(f_i, g_i)E_i \in \mathrm{End}_{\mathcal{O}}(\Lambda)$$

up to multiplication of the resultant by a unit.

Proof. Apply theorem 7.1 to the factorization $\chi_A = f_i g_i$. The resulting projector is one on $\operatorname{Ker} f_i(A)$ and zero on every $\operatorname{Ker} f_j(A)$ with $j \neq i$. The direct sum decomposition follows from the polynomial Chinese remainder theorem. The orthogonality and sum identities follow by evaluating the corresponding idempotents in $K[x]/(\chi_A)$. The Sylvester Bézout relation for (f_i, g_i) gives the stated denominator bound. □

Corollary 7.3 (Resultant denominator bound)**.** *Under the hypotheses of theorem 7.1, one may take*

$$r = \operatorname{Res}(f, g)$$

up to multiplication by a unit. Therefore

$$\pi^{\operatorname{val}_\pi(\operatorname{Res}(f,g))} E_f \Lambda \subseteq \Lambda.$$

Proof. The Sylvester matrix construction gives a Bézout relation with right side equal to the resultant. Apply theorem 7.1. Standard facts about resultants may be found in [5, 3]. □

Example 7.4 (A resultant bound that is not exact)**.** Let

$$A = \operatorname{diag}(0, \pi^s)$$

on the standard lattice $\Lambda = \mathcal{O}^2$. The factors x and $x - \pi^s$ have resultant $-\pi^s$, up to the harmless choice of resultant order, but the two spectral projectors are the integral coordinate projectors. Their exact separation weight is zero. The resultant sees the closeness of the two eigenvalues. It does not see that the chosen lattice already splits along their eigenspaces.

> **How to read the result**
>
> A resultant bounds the denominator through the characteristic factors. The exact separation weight also depends on how the lattice sits inside the rational packet spaces. These are two stages of the calculation. First construct a rational projector with a known clearing denominator. Then measure the gluing of the chosen lattice to find the least denominator that actually works.

7.1 Evaluated certificates and explicit lattice action

7.1.1 Minimal evaluated Bézout identities

Let $\mathcal{O}$ be a DVR, $K = \operatorname{Frac}(\mathcal{O})$, and $A \in \operatorname{End}_{\mathcal{O}}(\Lambda)$ for a finite free lattice Λ. Suppose the characteristic polynomial factors over $\mathcal{O}$ as

$$\chi_A(x) = f(x)g(x),$$

where f and g are coprime over $K[x]$.

Theorem 7.5 (Evaluated Bézout spectral projector)**.** *Let $F, G, U, V \in \operatorname{End}_K(\Lambda \otimes_{\mathcal{O}} K)$ and let $r \in K^\times$. Suppose*

$$UF + VG = rI, \qquad FG = 0, \qquad FG = GF, \qquad GU = UG.$$

Put $N = VG$. Then

$$\boxed{N^2 = rN.}$$

Consequently the operator

$$\boxed{E = r^{-1}VG}$$

is idempotent, acts as the identity on $\ker F$*, and acts as zero on* $\ker G$*. If* V *and* G *preserve* Λ*, then*

$$rE = VG \in \mathrm{End}_{\mathcal{O}}(\Lambda).$$

Proof. The Bézout identity gives $N = rI - UF$. The two stated commutation relations and $FG = 0$ give

$$N(UF) = VGUF = VUGF = VUFG = 0.$$

Therefore

$$N^2 = N(rI - UF) = rN.$$

Since $r \neq 0$, $E^2 = r^{-2}N^2 = E$. If $Fx = 0$, the evaluated Bézout identity gives $VGx = rx$, hence $Ex = x$. If $Gx = 0$, then $Ex = 0$. The lattice assertion follows directly from $rE = VG$. □

The equality $N^2 = rN$ before division uses neither $r \neq 0$ nor the other pairwise commutation relations that hold in the polynomial application. Normalizing to E uses $r \neq 0$. Its action on $\ker F$ uses only the Bézout identity and $r \neq 0$, while vanishing on $\ker G$ follows immediately from $N = VG$. The bundled Lean certificate records all pairwise commutations as convenient evaluated polynomial data, but the minimal theorems expose only the hypotheses they consume.

For the polynomial application, one supplies $F = f(A)$, $G = g(A)$, $U = u(A)$, and $V = v(A)$. A Bézout or resultant identity $uf + vg = r$ evaluates to the first certificate, while $(fg)(A) = 0$ supplies the second. When $fg = \chi_A$, this last equality is Cayley Hamilton. Standard Bézout, resultant, and DVR facts are recalled from [5, 8]. The inherited certificate theorem begins with the evaluated certificate. It does not infer exact action merely from nilpotence of $f(A)$ or $g(A)$, which would be insufficient.

Definition 7.6 (Separation weight)**.** For a rational projector E and a specified $\mathcal{O}$ lattice Λ, provided at least one clearing exponent exists, define

$$s_{\pi,\Lambda}(E) = \min\{s \geq 0 : \pi^s E(\Lambda) \subseteq \Lambda\}.$$

If the evaluated certificate arises from integral v, g and $r = u\pi^s$ with u^{-1} preserving the lattice, then theorem 7.5 gives the upper bound

$$s_{\pi,\Lambda}(E) \leq s = \mathrm{val}_\pi(r).$$

Indeed,

$$\boxed{\pi^s E = u^{-1}VG}$$

preserves Λ. The minimal Lean theorem needs only the lattice action of u^{-1}. The bundled wrapper for lattice units records both u and u^{-1}. A nonzero scalar in the fraction field need not be an integral unit. Existence of a clearing exponent is an explicit hypothesis. Once one exists, the least one is obtained from the least element property of the natural numbers. Clearing exponents are upward closed provided multiplication by π preserves the lattice.

7.1.2 Certificate-level precision delay

Theorem 7.7 (Projector precision delay)**.** *Let E be a rational projector with $\pi^s E(\Lambda) \subseteq \Lambda$. If*

$$x \equiv y \pmod{\pi^{m+s}\Lambda},$$

then

$$Ex \equiv Ey \pmod{\pi^m \Lambda}.$$

Thus $m+s$ input digits suffice to determine m output digits of the projected packet.

Proof. Write $x - y = \pi^{m+s} z$. Since $\pi^s E(z) \in \Lambda$,

$$E(x-y) = \pi^m(\pi^s E(z)) \in \pi^m \Lambda.$$

□

The theorem gives a precision budget for delayed separation. A packet with weight s can be reconstructed at the requested output precision by retaining s additional input digits.

8 Exact lattice separation

The exact denominator belongs to the pair consisting of the rational packet decomposition and the integral lattice.

Definition 8.1 (Lattice packet data)**.** Let

$$V = \bigoplus_{\alpha \in I} V_\alpha$$

be a finite direct sum over K, with packet projectors E_α. Let $\Lambda \subseteq V$ be an $\mathcal{O}$ lattice. Put

$$\Lambda_\alpha = \Lambda \cap V_\alpha, \qquad \Lambda_{\text{split}} = \bigoplus_\alpha \Lambda_\alpha,$$

and define the gluing module

$$G_{\Lambda,\mathcal{P}} = \Lambda / \Lambda_{\text{split}}.$$

The separation weight of E_α is

$$s_\Lambda(E_\alpha) = \min\{s \geq 0 : \pi^s E_\alpha \Lambda \subseteq \Lambda\}.$$

The global separation weight is

$$s_\Lambda(\mathcal{P}) = \max(\{0\} \cup \{s_\Lambda(E_\alpha) : \alpha \in I\}).$$

For a finitely generated torsion $\mathcal{O}$ module G, write $\exp_\pi(G)$ for the least s such that $\pi^s G = 0$.

The gluing module $G_{\Lambda,\mathcal{P}}$ is a finitely generated torsion $\mathcal{O}$ module, hence has finite length, because Λ_{split} and Λ are full lattices in the same vector space. To see that the split lattice is full, clear denominators of bases of the rational summands. Each $\Lambda \cap V_\alpha$ then spans V_α, so their direct sum spans V.

Theorem 8.2 (Gluing module and exact separation weight). *With the notation of definition 8.1, the map*

$$\Phi : G_{\Lambda,\mathcal{P}} \longrightarrow \bigoplus_{\alpha \in I} E_\alpha \Lambda / \Lambda_\alpha, \qquad [x] \longmapsto (E_\alpha x + \Lambda_\alpha)_\alpha$$

is injective, and every coordinate projection of its image is surjective. We also have

$$\boxed{\exp_\pi(G_{\Lambda,\mathcal{P}}) = s_\Lambda(\mathcal{P}).}$$

If

$$G_{\Lambda,\mathcal{P}} \cong \bigoplus_{j=1}^{r} \mathcal{O}/\pi^{a_j}\mathcal{O}$$

with $a_1 \le \cdots \le a_r$, then

$$s_\Lambda(\mathcal{P}) = \max(\{0\} \cup \{a_j : 1 \le j \le r\}).$$

Proof. If $x \in \Lambda$ maps to zero, then $E_\alpha x \in \Lambda_\alpha$ for every α. Since

$$x = \sum_\alpha E_\alpha x,$$

we have $x \in \Lambda_{\text{split}}$. Thus Φ is injective. The coordinate map onto $E_\alpha\Lambda/\Lambda_\alpha$ is surjective by the definition of $E_\alpha\Lambda$.

Let $s = s_\Lambda(\mathcal{P})$. For every $x \in \Lambda$ and every α,

$$\pi^s E_\alpha x \in \Lambda_\alpha.$$

Hence

$$\pi^s x = \sum_\alpha \pi^s E_\alpha x \in \Lambda_{\text{split}},$$

so $\pi^s G_{\Lambda,\mathcal{P}} = 0$. This proves

$$\exp_\pi(G_{\Lambda,\mathcal{P}}) \le s.$$

If $d < s$, then for some α there is an $x \in \Lambda$ with

$$\pi^d E_\alpha x \notin \Lambda_\alpha.$$

The class $\pi^d[x]$ is therefore nonzero in the gluing module. Thus π^d does not annihilate the gluing module, and equality follows. The final formula is the usual description of the exponent through Smith normal form. □

Corollary 8.3 (Total gluing and worst precision). *If*

$$G_{\Lambda,\mathcal{P}} \cong \bigoplus_{j=1}^{r} \mathcal{O}/\pi^{a_j}\mathcal{O},$$

then

$$\operatorname{length}_{\mathcal{O}}(G_{\Lambda,\mathcal{P}}) = \sum_{j=1}^{r} a_j, \qquad s_\Lambda(\mathcal{P}) = \max(\{0\} \cup \{a_j : 1 \le j \le r\}).$$

Thus the lattice index measures the total amount of gluing, while the separation weight measures the worst precision loss in one direction.

Proof. The length and exponent of a finitely generated torsion module are read directly from its Smith decomposition. Apply theorem 8.2. □

Corollary 8.4 (Two complementary packets)**.** *Suppose*

$$V = V_1 \oplus V_2$$

with projector E onto V_1 and complementary projector $I - E$. Then

$$\Lambda/(\Lambda_1 \oplus \Lambda_2) \cong E\Lambda/\Lambda_1 \cong (I-E)\Lambda/\Lambda_2.$$

Consequently

$$s_\Lambda(E) = s_\Lambda(I-E) = \exp_\pi(\Lambda/(\Lambda_1 \oplus \Lambda_2)).$$

Proof. Map $x \in \Lambda$ to $Ex + \Lambda_1$. Its kernel consists of the x for which $Ex \in \Lambda_1$. In that case

$$(I-E)x = x - Ex \in \Lambda \cap V_2 = \Lambda_2,$$

so the kernel is $\Lambda_1 \oplus \Lambda_2$. The map is surjective by definition. The complementary map is handled in the same way. The equality of weights follows from theorem 8.2. □

Corollary 8.5 (Integral splitting criterion)**.** *Every packet projector preserves Λ if and only if*

$$\Lambda = \bigoplus_{\alpha} (\Lambda \cap V_\alpha).$$

Equivalently,

$$s_\Lambda(\mathcal{P}) = 0 \quad \Longleftrightarrow \quad G_{\Lambda,\mathcal{P}} = 0.$$

Proof. If every projector preserves Λ, then

$$x = \sum_{\alpha} E_\alpha x$$

expresses every $x \in \Lambda$ as a sum of vectors in Λ_α. The converse is immediate because each projector is the identity on one direct summand and zero on the others. The final equivalence follows from theorem 8.2. □

8.1 Refinement and separation weight

A finer rational packet system imposes more integral splitting conditions.

Theorem 8.6 (Gluing exact sequence under refinement)**.** *Let $\mathcal{Q} = (Q_\beta)$ refine $\mathcal{P} = (P_\alpha)$ on V, and let Λ be a lattice. Define*

$$\Lambda_{\mathcal{P}} = \bigoplus_\alpha (\Lambda \cap P_\alpha V), \qquad \Lambda_{\mathcal{Q}} = \bigoplus_\beta (\Lambda \cap Q_\beta V).$$

Then

$$\Lambda_{\mathcal{Q}} \subseteq \Lambda_{\mathcal{P}} \subseteq \Lambda$$

and there is a short exact sequence

$$0 \longrightarrow \Lambda_{\mathcal{P}}/\Lambda_{\mathcal{Q}} \longrightarrow \Lambda/\Lambda_{\mathcal{Q}} \longrightarrow \Lambda/\Lambda_{\mathcal{P}} \longrightarrow 0.$$

In particular,

$$\boxed{s_\Lambda(\mathcal{P}) \le s_\Lambda(\mathcal{Q}).}$$

Proof. Every fine packet space lies in its parent packet space, so every element of $\Lambda_{\mathcal{Q}}$ belongs to $\Lambda_{\mathcal{P}}$. The displayed exact sequence is the standard quotient sequence for nested submodules. The exponent of a quotient cannot exceed the exponent of the module from which it is obtained. Apply theorem 8.2 to the two gluing modules. □

Example 8.7 (Strict growth under refinement)**.** Take the standard lattice $\mathbb{Z}_2^2$ and the involution from proposition 9.1. The one packet system (I) has weight zero. Refining it into the two rational sign packets produces weight one. The coarse lattice is already split because there is only one summand. The fine eigenspaces are glued together by one dyadic digit.

Remark 8.8 (Strict growth can occur). A coarse packet family may preserve the lattice even though a fine spectral split does not. Refinement can reveal rational directions that are glued together integrally. The weight records the extra depth needed to separate them.

8.2 Exact precision delay

The separation weight has an operational meaning. It is the exact number of extra input digits needed to recover a projected packet uniformly.

Theorem 8.9 (Necessary and sufficient precision delay)**.** *Let E be a rational projector on V, let Λ be a lattice, and let*

$$s = s_\Lambda(E).$$

For every $m \ge 0$, the rule

$$[x]_{m+s} \longmapsto [Ex]_m$$

from $\Lambda/\pi^{m+s}\Lambda$ to $V/\pi^m\Lambda$ is well defined. Here m is an integer. If d is an integer with $0 \le d < s$, then for every integer $m \ge 0$ there are $x, y \in \Lambda$ such that

$$x \equiv y \pmod{\pi^{m+d}\Lambda}$$

but

$$Ex \not\equiv Ey \pmod{\pi^m\Lambda}.$$

Thus s extra digits are both sufficient and necessary for a uniform precision guarantee.

Proof. If

$$x - y = \pi^{m+s} z,$$

then

$$E(x-y) = \pi^m(\pi^s E z) \in \pi^m \Lambda.$$

This proves that the rule is well defined. Now take $d < s$. By the minimality of s, there is a $z \in \Lambda$ with

$$\pi^d E z \notin \Lambda.$$

Set

$$x = \pi^{m+d} z, \qquad y = 0.$$

Then $x \equiv y$ modulo $\pi^{m+d}\Lambda$, while

$$E(x-y) = \pi^m(\pi^d E z) \notin \pi^m \Lambda.$$

□

The central idea

The necessity statement is uniform over the lattice. A particular input may need fewer extra digits, but a smaller delay cannot recover every projected packet at the prescribed precision. This is why the least clearing exponent gives an exact precision guarantee rather than only a sufficient budget.

Example 8.10 (The same rational split with two lattices)**.** Let

$$V = Ke_1 \oplus Ke_2$$

and let E project onto Ke_1. On the split lattice

$$\Lambda_0 = \mathcal{O}e_1 \oplus \mathcal{O}e_2,$$

the weight is zero. Fix $s \geq 1$ and consider

$$\Lambda_s = \mathcal{O}e_1 + \mathcal{O}(\pi^{-s}e_1 + e_2).$$

Then

$$\Lambda_s \cap Ke_1 = \mathcal{O}e_1, \qquad \Lambda_s \cap Ke_2 = \pi^s \mathcal{O}e_2.$$

The gluing module is isomorphic to $\mathcal{O}/\pi^s\mathcal{O}$, so

$$s_{\Lambda_s}(E) = s.$$

The rational decomposition has not changed. Only the lattice has changed.

The example can be realized by an integral operator. Define

$$Ae_1 = 0, \qquad Ae_2 = \pi^s e_2.$$

In the basis

$$e_1, \quad \pi^{-s}e_1 + e_2,$$

the matrix of A is

$$\begin{pmatrix} 0 & -1 \\ 0 & \pi^s \end{pmatrix}.$$

Its eigenvalue factors have resultant equal to π^s up to a unit, and the exact projector weight on Λ_s is also s.

9 The dyadic sign split

The smallest example already shows why coefficient selectors and spectral averages must be kept distinct.

Proposition 9.1 (Sign projectors over a lattice)**.** *Let*

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$$

act on $\mathbb{Z}_p^2$. Over $\mathbb{Q}_p$, the sign projectors are

$$E_+ = \frac{I + A}{2} = \begin{pmatrix} 1 & 1/2 \\ 0 & 0 \end{pmatrix},$$

$$E_- = \frac{I - A}{2} = \begin{pmatrix} 0 & -1/2 \\ 0 & 1 \end{pmatrix}.$$

For every odd prime p, both projectors preserve $\mathbb{Z}_p^2$. At $p = 2$,

$$s_2(E_+) = s_2(E_-) = 1.$$

The factors $x - 1$ and $x + 1$ have resultant 2.

Proof. The displayed matrices are complementary idempotents. The scalar 2 is a unit in $\mathbb{Z}_p$ for odd p, so both matrices are integral there. Over $\mathbb{Z}_2$, multiplying by 2 clears the denominators, while neither projector preserves the standard lattice. Thus both weights are one. The resultant of the two linear factors is their root difference, which is 2. □

Modulo 2, the two eigenvalues become equal. The rational sign packets remain distinct over $\mathbb{Q}_2$, but the standard lattice does not split into their intersections until one more dyadic digit is retained.

Proposition 9.2 (Integral character averages away from the group order)**.** *Let $\mathcal{O}$ be a discrete valuation ring, let N be a unit in $\mathcal{O}$, and suppose $\mathcal{O}$ contains a primitive Nth root of unity ω. If an integral operator U satisfies $U^N = I$, then*

$$E_j = \frac{1}{N} \sum_{k=0}^{N-1} \omega^{-jk} U^k$$

preserves every U stable $\mathcal{O}$ lattice. At primes dividing N, this formula alone gives no integral projector.

Proof. Every scalar coefficient in the displayed sum belongs to $\mathcal{O}$ because N is a unit and ω is integral. Each power of U preserves the lattice, so their $\mathcal{O}$ linear combination does as well. When N is not a unit, division by N is not an operation in $\mathcal{O}$, so the same argument is unavailable. □

At a prime dividing N, the factor $1/N$ no longer guarantees that the character average preserves the lattice. Cancellation may still make a particular projector integral. Coefficient residue selectors remain exact because they never divide by the group order.

9.1 The concrete rational sign matrix

Consider the integral involution

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \qquad A^2 = I.$$

Over a field of characteristic different from two, its sign projectors are

$$E_+ = \frac{I + A}{2} = \begin{pmatrix} 1 & 1/2 \\ 0 & 0 \end{pmatrix},$$

$$E_- = \frac{I - A}{2} = \begin{pmatrix} 0 & -1/2 \\ 0 & 1 \end{pmatrix}.$$

They are orthogonal idempotents and sum to I.

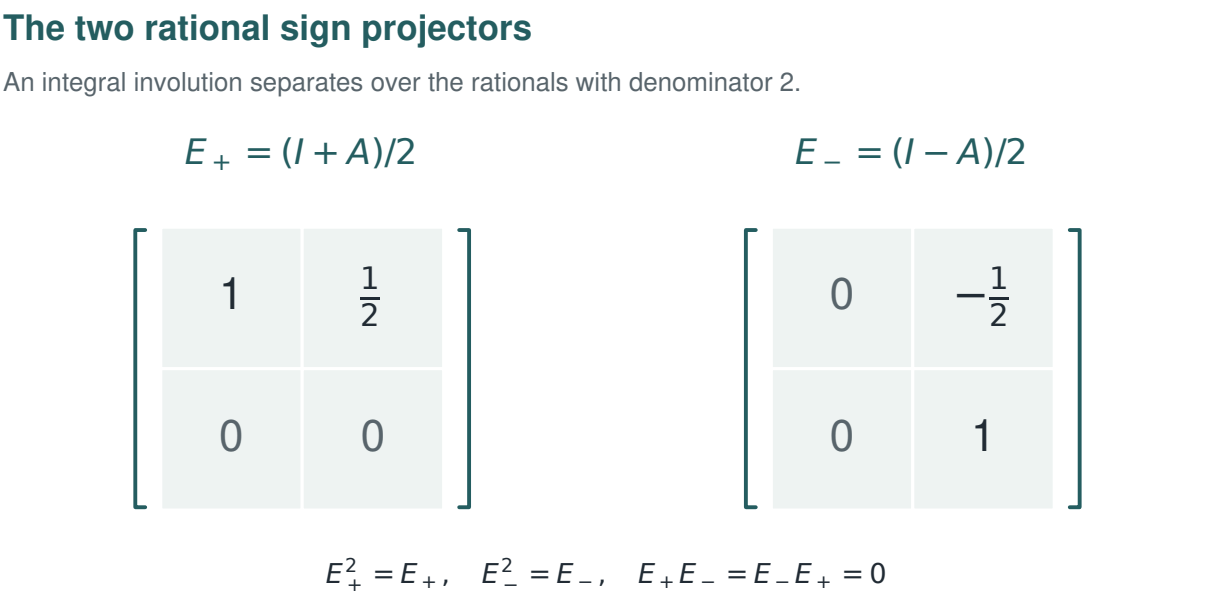


Figure 7. The rational sign projectors for an integral involution. Their denominator has 2-adic weight one.

Proposition 9.3 (Odd-adic versus dyadic behavior)**.** *For every prime $p \neq 2$, every entry of E_+ and E_- is p-integral, and each projector has exact local denominator weight zero. At $p = 2$, each projector has exact local denominator weight one. Multiplication by 2 makes it integral, while the unscaled matrix is not integral:*

$$s_{2,\mathbb{Z}_2^2}(E_+) = s_{2,\mathbb{Z}_2^2}(E_-) = 1.$$

The characteristic factors $x-1$ and $x+1$ have resultant 2. Its 2-adic valuation is one, matching both local projector weights above, and its p-adic valuation is zero at every prime $p \neq 2$.

Proof. The reduced denominators of the displayed entries are 1 or 2, which proves the local integrality and minimality statements directly. The resultant is

$$\operatorname{Res}(x-1, x+1) = 2.$$

Thus $v_2(2) = 1$ and $v_p(2) = 0$ for primes $p \neq 2$. □

The two characteristic factors $x-1$ and $x+1$ have the same reduction in $\mathbb{F}_2[x]$. The rational matrices E_+ and E_- themselves cannot be reduced modulo 2, because their entries contain $1/2$. Consequently the integral special fiber does not inherit their separate idempotent splitting, even though the two projectors remain distinct over $\mathbb{Q}_2$. This is the simplest model of delayed dyadic separation.

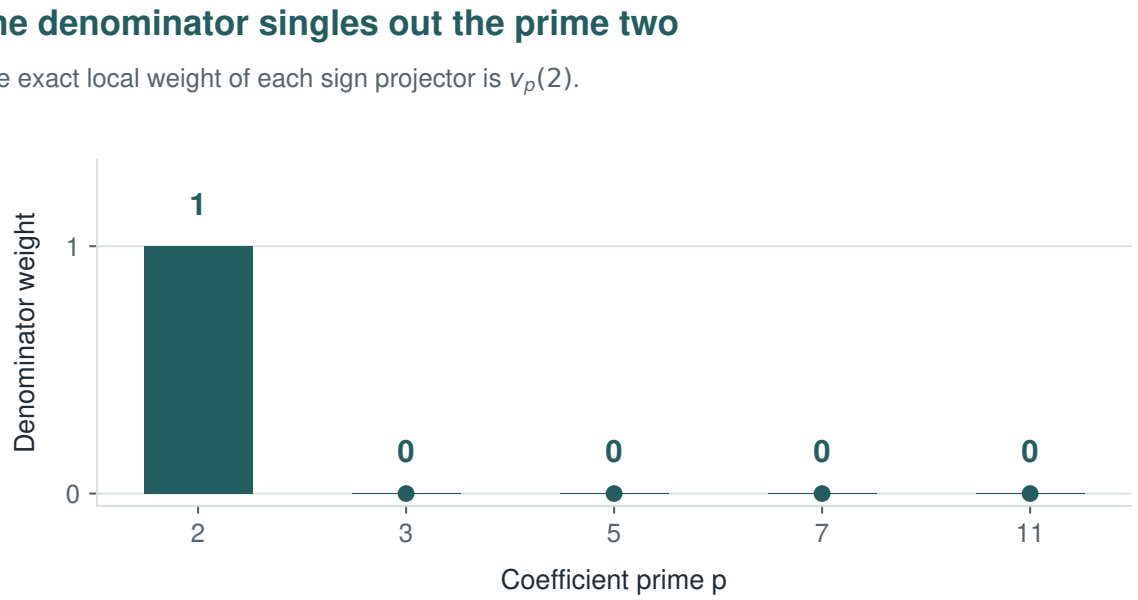


Figure 8. The denominator weight of the sign projector is nonzero only at the prime two.

9.1.1 Why coefficient packets remain available

The failure of the character formula

$$\frac{1}{2}(I \pm A)$$

to be integral does not invalidate coefficient selectors such as even and odd residue extraction. It shows that two distinct constructions must be distinguished:

integral address selector	$\neq$	character-average spectral projector

when the group order is not a unit. The first can remain exact over $\mathbb{Z}_2$. The second can acquire a denominator weight.

10 Smith normal form and congruence lifting

The gluing module gives the exact common denominator of packet projectors. Smith normal form gives the corresponding coordinate by coordinate solvability and branching rule.

Standard Fact 10.1 (Smith normal form over a discrete valuation ring)**.** Let $M \in M_{m,n}(\mathcal{O})$. There are invertible matrices

$$U \in \mathrm{GL}_m(\mathcal{O}), \qquad V \in \mathrm{GL}_n(\mathcal{O})$$

such that

$$UMV = \mathrm{diag}(\pi^{s_1}, \dots, \pi^{s_r}, 0, \dots, 0),$$

where

$$0 \le s_1 \le \cdots \le s_r$$

and r is the rank over K. The exponents are uniquely determined. See [6].

Proposition 10.2 (Computing a packet weight from an inclusion matrix)**.** *Choose an $\mathcal{O}$ basis of Λ and an $\mathcal{O}$ basis of Λ_{split}. Let B be the matrix of the inclusion*

$$\Lambda_{\mathrm{split}} \hookrightarrow \Lambda.$$

If the Smith form of B is

$$\mathrm{diag}(\pi^{a_1}, \dots, \pi^{a_n}), \qquad 0 \le a_1 \le \cdots \le a_n,$$

then

$$G_{\Lambda,\mathcal{P}} \cong \bigoplus_{j=1}^{n} \mathcal{O}/\pi^{a_j}\mathcal{O}, \qquad s_\Lambda(\mathcal{P}) = \max(\{0\} \cup \{a_j : 1 \le j \le n\}).$$

Proof. The gluing module is the cokernel of the inclusion matrix B. Smith normal form gives its invariant factors. The weight formula follows from theorem 8.2. □

Theorem 10.3 (Smith weighted congruence solve)**.** *Let $M \in M_{m,n}(\mathcal{O})$ have Smith exponents $s_1, \dots, s_r$. Assume the residue field is finite of cardinality q, fix $L \ge 1$, and consider*

$$Mx \equiv b \pmod{\pi^L}.$$

Put

$$\tilde{b} = Ub$$

for a Smith matrix U as in standard fact 10.1. The congruence is solvable if and only if

$$\tilde{b}_i \equiv 0 \pmod{\pi^{\min(s_i, L)}} \qquad (1 \le i \le r)$$

and

$$\tilde{b}_i \equiv 0 \pmod{\pi^L} \qquad (r < i \le m).$$

When it is solvable, the number of solutions in $(\mathcal{O}/\pi^L\mathcal{O})^n$ is

$$\boxed{q^{(n-r)L + \sum_{i=1}^{r} \min(s_i, L)}.}$$

Proof. Set $x = Vy$. Multiplying the congruence by U reduces it to independent equations

$$\pi^{s_i} y_i \equiv \tilde{b}_i \pmod{\pi^L} \qquad (1 \le i \le r)$$

and zero equations in the remaining rows. Multiplication by π^{s_i} on $\mathcal{O}/\pi^L\mathcal{O}$ has image equal to the elements divisible by $\pi^{\min(s_i,L)}$ and kernel of size

$$q^{\min(s_i,L)}.$$

Each of the $n - r$ unused variables is free and contributes q^L choices. Multiplying the independent counts gives the formula. □

Corollary 10.4 (Obstruction and branching data)**.** *The transformed entries of $\tilde{b}$ that fail the stated divisibility conditions are the complete finite congruence obstructions. When all obstructions vanish, the Smith exponents determine the number of solution branches at depth L.*

Proof. This is exactly the solvability and counting statement of theorem 10.3 written in invariant coordinates. □

Example 10.5 (Three weighted directions)**.** Take $p = 2$, $L = 6$, and the diagonal weights

$$(0, 1, 3).$$

The right side

$$(1, 2, 8)$$

is admissible. The three coordinate kernels have sizes

$$1, \quad 2, \quad 8,$$

so there are 16 solutions. Replacing the final entry 8 by 4 violates the required divisibility by 2^3 and makes the system unsolvable.

10.1 Concrete diagonal fibers over integer quotients

Let M be a matrix over a DVR. Smith normal form gives invertible matrices U, V such that

$$UMV = \operatorname{diag}(\pi^{s_1}u_1, \dots, \pi^{s_r}u_r, 0, \dots, 0),$$

where each u_i is a unit and

$$0 \le s_1 \le \cdots \le s_r.$$

The exponents s_i are directional lifting weights.

Theorem 10.6 (Concrete Smith-weighted congruence solve)**.** *Let $p > 1$ and consider the diagonal system over $(\mathbb{Z}/(p^L)\mathbb{Z})^d$*

$$\operatorname{diag}(p^{s_1}, \dots, p^{s_d})x = b \pmod{p^L}.$$

The system is solvable if and only if

$$p^{\min(s_i,L)} \mid (b_i)_{\mathrm{val}}$$

for every i, where $(b_i)_{\mathrm{val}}$ is the canonical natural representative. Conditional on this admissibility condition, the number of solutions is

$$\boxed{p^{\sum_{i=1}^{d} \min(s_i, L)}.}$$

If any coordinate is inadmissible, the solution set is empty and its cardinality is zero.

Proof. Lean proves the one-coordinate arithmetic directly over $\mathbb{Z}/p^L\mathbb{Z}$ for every $p > 1$:

$$(\exists x \in \mathbb{Z}/p^L\mathbb{Z} : \quad p^s x = b \pmod{p^L}) \quad \Longleftrightarrow \quad p^{\min(s,L)} \mid b_{\mathrm{val}},$$

where b_{val} is the canonical natural representative, and every admissible fiber has exactly $p^{\min(s,L)}$ elements. The coordinates are independent, so the product equivalence gives the displayed global count. If one coordinate fiber is empty, the global fiber is empty. Lean proves both the global solvability equivalence and the conditional count, including cardinality zero on the inadmissible branch. □

For an arbitrary DVR $\mathcal{O}$ with residue field cardinality q, the analogous formula

$$q^{\sum_i \min(s_i, L)}$$

follows from the solvability and fiber count for each coordinate. The formal correspondence records direct proofs of this DVR coordinate arithmetic and the rectangular transport as separate targets. The inherited abstract product interface remains available when the coordinate facts are supplied as certificates. A Smith form and its changes of basis are inputs to the transport statement. It does not provide an extracted diagonalization routine. See [8, 5].

If a Smith form is rank deficient, a literal zero diagonal factor requires the corresponding target coordinate to vanish modulo p^L and then contributes all p^L possible coordinate values. In the concrete quotient it can equivalently be represented by any weight $s_i \geq L$, since $p^{s_i} = 0$ in $\mathbb{Z}/(p^L)\mathbb{Z}$. The theorem then gives exactly that condition and factor. The numbered theorem takes the diagonal data as its input.

Example 10.7. At $p = 2$, depth $L = 6$, and weights $(0, 1, 3)$, the right side $(1, 2, 8)$ is admissible. The coordinate kernels have sizes

$$1, 2, 8,$$

so the system has 16 solutions. Replacing 8 by 4 in the final coordinate violates the divisibility condition and makes the system unsolvable.

How to read the result

The gluing calculation takes the largest Smith weight because every packet must be separated to the requested precision. The congruence calculation retains the individual directional weights. A coordinate with weight s_i contributes a solvability threshold and $\min(s_i, L)$ to the branching exponent. The independent coordinate counts multiply, so these truncated weights add in the exponent.

11 Linear refinement defects

Exact coordinate refinement describes two representations of the same object. In an application, the coarse operator may instead be chosen to model coarse dynamics. An intertwining defect measures the mismatch between applying the operator before or after passing to coarse coordinates.

Let X_f and X_c be vector spaces or Banach spaces over a common field, and let

$$C : X_f \longrightarrow X_c$$

be a linear map, and let

$$T_f \in \mathrm{End}(X_f), \qquad T_c \in \mathrm{End}(X_c).$$

Define

$$\Delta = CT_f - T_c C.$$

The central idea

There are two routes from a fine state to a coarse result. One applies the fine operator and then maps the result to the coarse space. The other maps the state first and then applies the coarse operator. The defect is the difference between these routes. Its vanishing says that the chosen operators are compatible with the comparison map. The norm estimates below combine the size of this defect with the norms of the surrounding operators.

Theorem 11.1 (Power defect identity)**.** *For every integer $n \geq 1$,*

$$CT_f^n - T_c^n C = \sum_{k=0}^{n-1} T_c^{n-1-k} \Delta T_f^k.$$

Proof. The statement is clear for $n = 1$. For the induction step, write

$$CT_f^{n+1} - T_c^{n+1} C = (CT_f^n - T_c^n C)T_f + T_c^n (CT_f - T_c C)$$

and apply the induction hypothesis. □

Corollary 11.2 (Power norm bound)**.** *For bounded operators between Banach spaces,*

$$\|CT_f^n - T_c^n C\| \leq \|\Delta\| \sum_{k=0}^{n-1} \|T_c\|^{n-1-k} \|T_f\|^k.$$

Proof. Apply the triangle inequality and submultiplicativity to theorem 11.1. □

Corollary 11.3 (Norm bound)**.** *For a submultiplicative operator norm,*

$$\|CT_f^n - T_c^n C\| \leq \|\Delta\| \sum_{k=0}^{n-1} \|T_c\|^{n-1-k} \|T_f\|^k.$$

If $\|T_f\|, \|T_c\| \leq \rho$, then

$$\boxed{\|CT_f^n - T_c^n C\| \leq n\rho^{n-1} \|\Delta\|.}$$

Corollary 11.4 (Polynomial defect)**.** *For*

$$p(z) = \sum_{n=0}^{N} a_n z^n,$$

we have

$$Cp(T_f) - p(T_c)C = \sum_{n=1}^{N} a_n \sum_{k=0}^{n-1} T_c^{n-1-k} \Delta T_f^k.$$

Proof. The constant terms cancel because $CI = IC$. Apply theorem 11.1 to every positive power and sum. □

Theorem 11.5 (Resolvent defect identity)**.** *If $z - T_f$ and $z - T_c$ are invertible, then*

$$C(z - T_f)^{-1} - (z - T_c)^{-1}C = (z - T_c)^{-1}\Delta(z - T_f)^{-1}.$$

Proof. Multiply the claimed identity on the left by $z - T_c$ and on the right by $z - T_f$. The left side becomes

$$(z - T_c)C - C(z - T_f) = CT_f - T_cC = \Delta.$$

Multiplying back by the two resolvents proves the formula. □

11.1 Inverse witnesses from one side and norm control

For continuous linear maps, the same identity gives the certified norm bound

$$\boxed{\|CR_f - R_cC\| \le \|R_c\|\,\|\Delta\|\,\|R_f\|,}$$

from a left inverse witness for the coarse scalar shift and a right inverse witness for the fine scalar shift. These witnesses suffice for the displayed identity. On their own, they need not define inverses on both sides and hence need not define resolvents. If $\Delta = 0$, Lean also proves, for every finite polynomial p, the exact compatibility

$$\boxed{C\,p(T_f) = p(T_c)\,C,}$$

in both the algebraic setting and the setting of continuous linear maps. Thus all powers and polynomial actions intertwine with coarse reconstruction. The stated inverse witnesses also intertwine. In particular, genuine resolvents do so when the scalar shifts are invertible. This is the linear prototype of the more general theory of refinement defects reserved for Chapter III.

11.2 Analytic functional calculus

Theorem 11.6 (Holomorphic functional calculus defect)**.** *Let X_f and X_c be complex Banach spaces, and let $C : X_f \to X_c$, T_f, and T_c be bounded complex-linear maps. Let h be holomorphic on an open set U containing both spectra. Let Γ be an admissible finite rectifiable Cauchy cycle in U and in both resolvent*

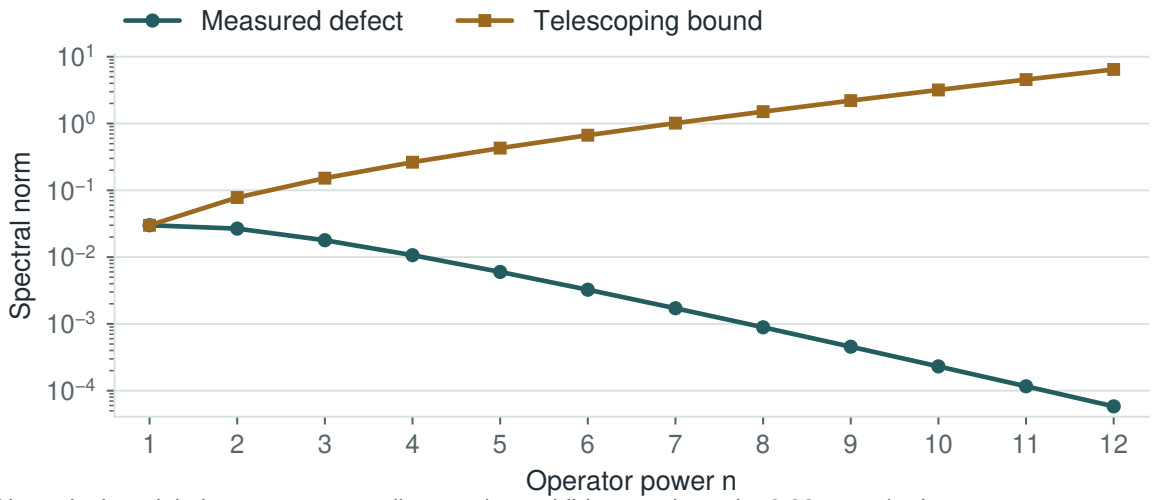


Figure 9. A controlled defect for one step and its telescoping norm bound under repeated operator application.

sets, with winding number one on the union of the spectra and zero outside U. The length of a cycle counts multiplicities. Then

$$Ch(T_f) - h(T_c)C = \frac{1}{2\pi i}\int_\Gamma h(z)(z - T_c)^{-1}\Delta(z - T_f)^{-1}\,dz.$$

Consequently,

$$\|Ch(T_f) - h(T_c)C\| \le \frac{\operatorname{length}(\Gamma)}{2\pi}\sup_{z\in\Gamma}|h(z)|\,\|\Delta\| \times \sup_{z\in\Gamma}\|(z - T_c)^{-1}\|\sup_{z\in\Gamma}\|(z - T_f)^{-1}\|.$$

Proof. The assertion is immediate if either space is zero. Otherwise, use the rectifiable-cycle form of the holomorphic functional calculus in [1, Chapter VII, §4, Theorem 4.1, formula (4.5), and Proposition 4.6]. For each operator T, compare Γ with a positive simple system around its spectrum. Their difference has zero winding outside $U \setminus \sigma(T)$, so Theorem 4.1 applied to the resolvent-valued integrand identifies the two integrals. Consequently,

$$h(T) = \frac{1}{2\pi i}\int_\Gamma h(z)(z - T)^{-1}\,dz.$$

Subtract the two formulas after composing with C, then use theorem 11.5 inside the integral. The norm estimate follows from the standard contour bound. □

Theorem 11.7 (Exponential defect identity). *Let $C : X_f \to X_c$, T_f, and T_c be bounded linear maps between Banach spaces over a common field. For real t, use real or complex spaces. For complex t, use complex spaces and complex-linear maps. Then*

$$Ce^{tT_f} - e^{tT_c}C = \int_0^t e^{(t-s)T_c}\Delta e^{sT_f}\,ds,$$

where the integral is taken along the straight segment from 0 to t when t is complex. For real $t \geq 0$,

$$\|Ce^{tT_f} - e^{tT_c}C\| \leq \|\Delta\| \int_0^t \|e^{(t-s)T_c}\|\|e^{sT_f}\|\,ds.$$

Proof. For real t, differentiate

$$F(s) = e^{(t-s)T_c}Ce^{sT_f}.$$

We obtain

$$F'(s) = e^{(t-s)T_c}\Delta e^{sT_f}.$$

Integration from 0 to t gives the identity. The complex statement follows by the same calculation along the straight segment. The norm bound follows from the integral formula. □

11.3 Several refinement levels

Let

$$C_\ell : X_{\ell+1} \longrightarrow X_\ell$$

be linear maps for $0 \leq \ell < L$, and let $T_\ell \in \mathrm{End}(X_\ell)$. Put

$$\Delta_\ell = C_\ell T_{\ell+1} - T_\ell C_\ell.$$

For $a < b$, write

$$C_{a,b} = C_a C_{a+1} \cdots C_{b-1},$$

with $C_{a,a} = I$.

Theorem 11.8 (Multilevel defect decomposition). *The total defect from level L to level zero is*

$$C_{0,L}T_L - T_0 C_{0,L} = \sum_{\ell=0}^{L-1} C_{0,\ell}\Delta_\ell C_{\ell+1,L}.$$

For bounded maps,

$$\|C_{0,L}T_L - T_0 C_{0,L}\| \leq \sum_{\ell=0}^{L-1} \|C_{0,\ell}\|\|\Delta_\ell\|\|C_{\ell+1,L}\|.$$

Proof. Insert the intermediate terms

$$C_{0,\ell}T_\ell C_{\ell,L}$$

for $0 \le \ell \le L$. Consecutive differences telescope. The difference between the terms indexed by $\ell + 1$ and ℓ is

$$C_{0,\ell}(C_\ell T_{\ell+1} - T_\ell C_\ell)C_{\ell+1,L}.$$

Summing gives the identity. The norm estimate follows from the triangle inequality. □

How to read the result

The multilevel formula keeps track of where each mismatch enters. A defect at one level is composed with the comparison maps needed to reach the input and output levels. The norm estimate therefore depends on both the local defects and the maps that carry them through the hierarchy. Controlling local mismatches alone does not control the total error if those maps amplify them.

Example 11.9 (A visible one step defect)**.** Let $X_f = \Bbbk^2$, let $X_c = \Bbbk$, and let

$$C(x_1, x_2) = x_1.$$

Take

$$T_f = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \qquad T_c = (a).$$

Then

$$\Delta = CT_f - T_c C = (0, b).$$

The coarse operator is exactly compatible with the fine operator precisely when the discarded coordinate does not feed into the retained coordinate. The entry c does not appear in the one step defect because it moves information in the opposite direction. It can still affect later powers through the terms in theorem 11.1.

Corollary 11.10 (Exact compatibility)**.** *If every one step defect Δ_ℓ vanishes, then*

$$C_{0,L}p(T_L) = p(T_0)C_{0,L}$$

for every polynomial p. The analogous resolvent, holomorphic, and exponential identities also hold whenever the relevant expressions are defined.

Proof. The total one step defect vanishes by theorem 11.8. Apply corollary 11.4 and theorems 11.5 to 11.7. □

12 Worked realizations in the common operator language

12.1 Signal polyphase coordinates

Let $x = (x_n)_{n\in\mathbb{Z}} \in c_{00}(\mathbb{Z}; R)$ and let the seed be δ_0 under the bilateral shifts $(S^r\delta_0)_n = \delta_{n,r}$. This makes the algebraic span and every displayed sum finite. The N packets are

$$x_j[m] = x[Nm + j].$$

Synthesis interleaves these sequences. A linear filter with a finite impulse response that is invariant under shifts becomes an $N \times N$ matrix of transfer blocks. Refining from N to M with $N \mid M$ splits each

coarse phase into its descendants without changing the reconstructed signal. The same formulas extend to standard sequence completions when the synthesis series converges. This is the usual polyphase viewpoint from multirate filter banks [10].

12.2 Sparsity of transfer blocks

A banded operator need not remain banded after residue reordering, but its nonzero couplings become a structured packet graph. With the convention $(S_r x)_n = x_{n-r}$, the only active transfer is

$$P_{N,j+r} S_r P_{N,j},$$

with indices taken modulo N. More general finite stencils produce a finite set of active phase differences.

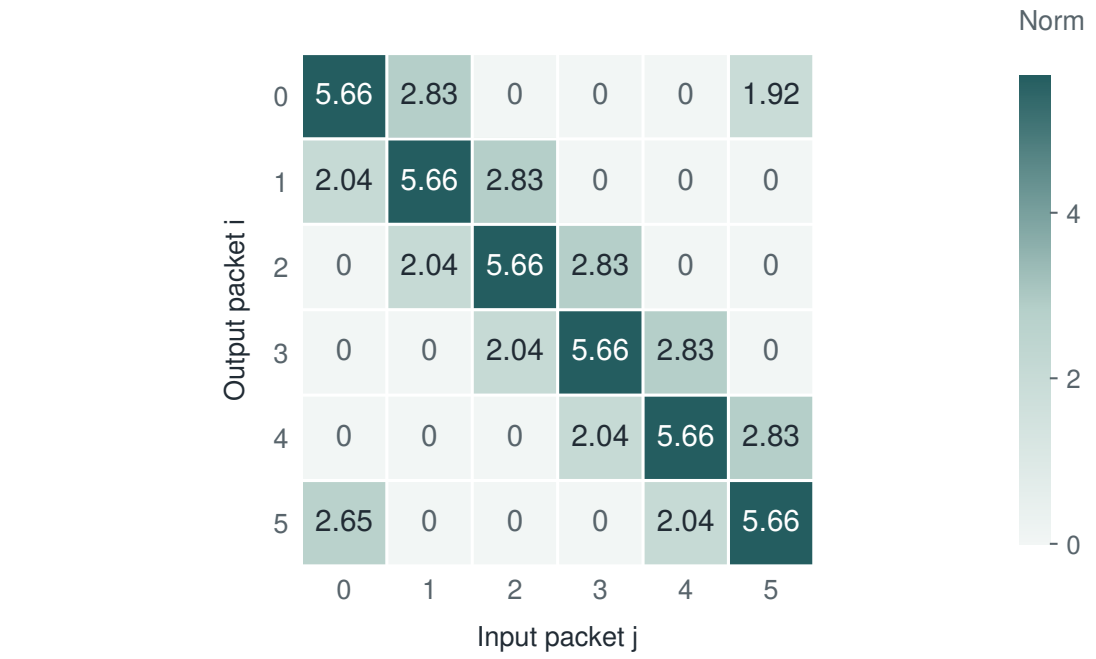


Figure 10. Norms of packet blocks for a finite stencil. The original operator is exactly reconstructed from a sparse transfer pattern.

12.3 A short realization in Smith chart coordinates

How to read the result

The Smith chart used here is a coordinate diagram for impedance and reflection. It is different from the Smith normal form used earlier to study integral congruences. The connection within this chapter is the way a change of coordinates makes an action easier to describe. Here a rotation in reflection coordinates becomes a fractional linear map in impedance coordinates. The sampled phase orbit is a geometric example and does not supply another linear projector decomposition.

Let Z and $Z_0 \neq 0$ be complex impedances, put $z = Z/Z_0$, and assume $z \neq -1$. Write

$$\Gamma = \frac{z-1}{z+1}.$$

On the open reflection disk $|\Gamma| < 1$, multiplication by any phase ω_N^k stays in the disk and has a finite inverse-chart image. The phase action is

$$\Gamma \mapsto \omega_N^k \Gamma.$$

Outside the open disk, the same chart formula requires $1 - \omega_N^k \Gamma \neq 0$. Pulling the phase action back on this domain gives

$$T_{N,k}(z) = \frac{(1+\omega_N^k)z + (1-\omega_N^k)}{(1-\omega_N^k)z + (1+\omega_N^k)}.$$

For $N = 2$ and $k = 1$,

$$T_{2,1}(z) = \frac{1}{z}, \qquad z \neq 0, -1.$$

Thus the nontrivial action at the first phase level exchanges normalized impedance and admittance. Calling these values "packets" is geometric terminology for the sampled orbit. It does not assert that the two values form linear idempotent ranges. Higher refinements at powers of 2 sample the same reflection circle at progressively finer electrical phases. Compare the standard Smith chart conventions in [7].

The symmetric weight

$$w_{\mathrm{Smith}}(\Gamma) = \frac{1}{|1-\Gamma|^2} + \frac{1}{|1+\Gamma|^2}$$

indicates where small errors in reflection coordinates produce large impedance or admittance errors.

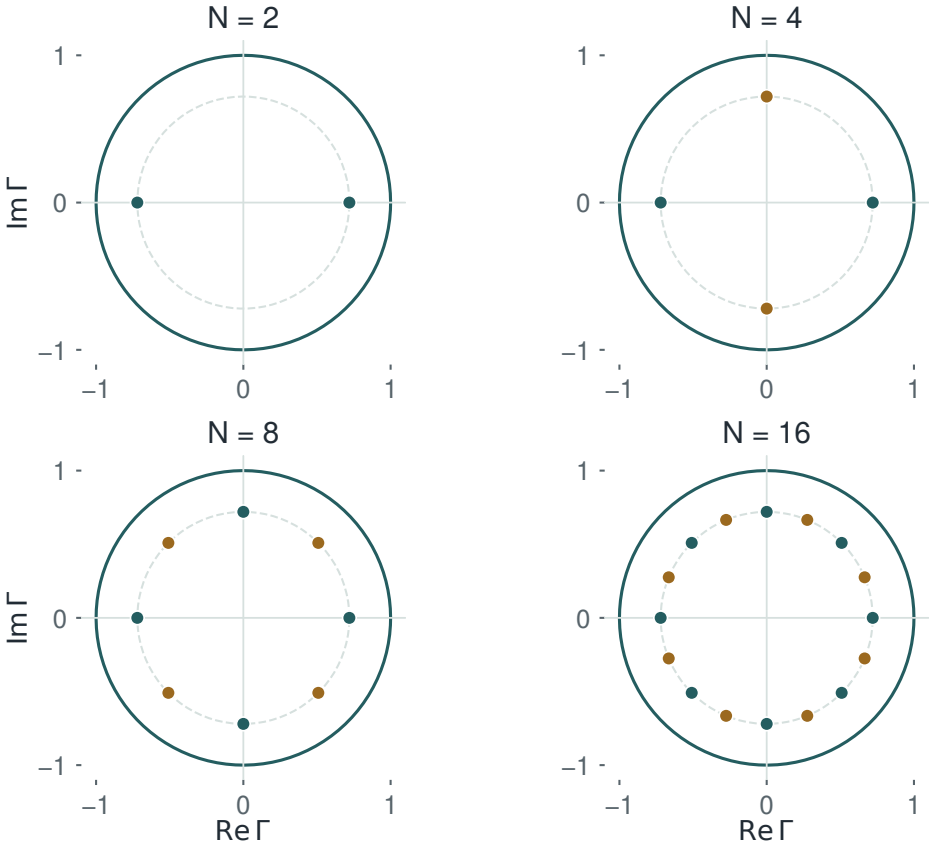
13 Finite computational certificates

The following results and tables record finite checks of the implementation. They do not certify the general lattice or analytic statements in Lean. The accompanying verification files record the execution evidence and tested ranges.

All numerical work in this chapter is finite and reproducible. It tests the Python implementation of identities proved above or checked against certificates. It does not replace those arguments or the Lean kernel checks. The recorded implementation run covers 48 tests, including contracts for invalid input, refinement transitivity, aggregation of transfer blocks, Fourier/DFT agreement, Smith fiber counts by exhaustive enumeration, and consistency of the stored JSON and CSV records.

Nested polyphase samples

Reflection coordinates retain their geometry at every dyadic refinement.



Every sample has |Γ| = 0.72. Gold marks points added at N = 4, 8, or 16.

Figure 11. Nested phase orbits at powers of 2 on the Smith reflection disk. The full treatment of the Smith chart is deferred to the signal chapter.

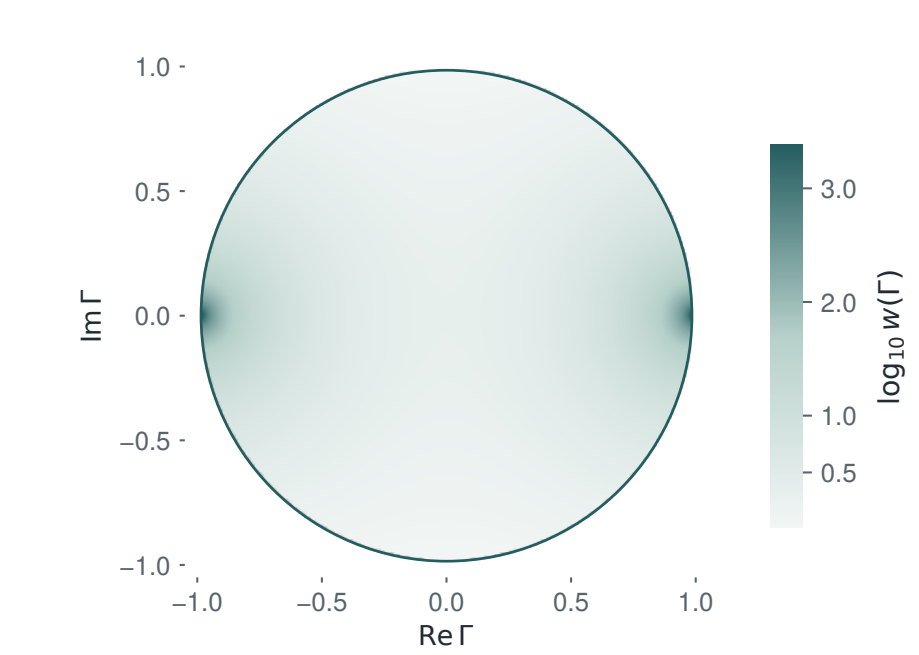


Figure 12. A symmetric refinement weight concentrated near the boundary points for open and short circuits.

Table 1. Representative numerical and symbolic checks.

Check	Result
Packet synthesis residuals for moduli $2, 3, 4, 6, 8, 12$	0
Nested refinement residuals for $2 \to 4$, $2 \to 8$, $3 \to 6$, $3 \to 12$, $4 \to 8$, $4 \to 12$, $6 \to 12$	0
CRT residuals for dyadic and odd moduli for 4×3, 8×3, 4×5, 8×5	0
Maximum finite Fourier projector residual	6.60×10^{-15}
Weighted inverse residual modulo 2^{10}	0
Weighted inverse residual modulo 3^8	0
Weighted inverse residual modulo 5^7	0
Difference between filtered and direct modular inverses	0
Transfer zeta coefficients versus direct walk enumeration through weight 30	exact match
Dyadic sign projector resultant weight	$v_2(2) = 1$
Sign projector resultant weight at the odd prime 3	$v_3(2) = 0$
Power defect telescoping identity residual	at most 6.94×10^{-18}
Resolvent defect identity residual	1.09×10^{-17}
Smith $N = 2$ phase versus $z \mapsto 1/z$	residual below 1.4×10^{-16}

Numerical Certificate 13.1 (Weighted inverse)**.** For the tested diagonal-plus-transfer matrices, the filtered inverse agrees exactly with direct modular inversion at all tested precisions:

$$2^{10}, \qquad 3^{8}, \qquad 5^{7}.$$

The maximum modular residual is zero in every case.

Numerical Certificate 13.2 (Transfer zeta)**.** The rational function

$$\frac{-t^3}{t^9 - t^7 - t^6 - t^5 + 2t^4 + t^3 + t^2 - 1}$$

was expanded through degree 30. Every coefficient agrees with direct weighted-walk enumeration.

Numerical Certificate 13.3 (Dyadic separation)**.** The code verifies exactly that the two rational sign projectors are idempotent, orthogonal, and complementary, that scaling by 2 makes both integral, and that the same projectors are already integral modulo every tested odd prime.

14 Scope and limits

The exact statements above have several deliberate boundaries.

First, refinement changes coordinates while preserving reconstruction. If a numerical method discards fine packets, the resulting error must be estimated separately.

Second, the weighted inverse is linear. A nonlinear map creates higher order interactions among packet coordinates. Those interactions are not contained in the path formula of theorem 5.2.

Third, a resultant is only a denominator bound. The exact weight depends on the lattice and is given by theorem 8.2. The two examples in examples 7.4 and 8.10 show that the same eigenvalue separation can give different integral packet weights.

Fourth, the dyadic sign example is a model rather than a classification of all phenomena at the prime two. Character averages, trace splittings, quadratic refinements, and deformation projectors must each be checked on their own lattices.

Finally, the defect formulas do not prove that a chosen coarse operator is a good approximation. They say exactly how a known one step defect propagates. A useful approximation theorem must also bound that defect in the problem under study.

A Notation table

Symbol	Meaning
$P_{\ell,\alpha}$	packet projector at refinement level ℓ
I_ℓ	packet address set at level ℓ
$\beta \succ \alpha$	fine packet β is a child of coarse packet α
$\mathcal{A}_\ell, \mathcal{S}_\ell$	packet analysis and synthesis maps
$J_{\ell+1,\ell}$	exact coordinate refinement map
$C_{\ell,\ell+1}$	exact coordinate regrouping from fine to coarse
$T^{(\ell)}_{\beta\alpha}$	transfer block $P_{\ell,\beta} T P_{\ell,\alpha}$

Symbol	Meaning
$P_{N,j}$	residue or phase selector for address $j \pmod N$
$w(e)$	positive filtration weight of a transfer edge
$W(\gamma)$	cumulative weight of a transfer word or graph walk
$A(t)$	polynomial weighted adjacency matrix
$Z_{s,t}(t)$	weighted transfer-zeta function from vertex s to vertex t
$s_{\pi,\Lambda}(E)$	separation weight of a rational projector on the chosen lattice, when a clearing exponent exists
Δ	linear refinement defect for one step $CT_f - T_cC$

B Algorithmic forms

B.1 Residue analysis and synthesis

```
def analyze_residues(x, N):
    require N > 0
    return [x[j::N] for j in range(N)]

def synthesize_residues(packets):
    N = len(packets)
    require packet lengths fit one N-way interleaving
    y = empty(sum(len(p) for p in packets))
    for j, packet in enumerate(packets):
        y[j::N] = packet
    return y
```

The verified contract is synthesize(analyze$(x, N)) = x$. The Lean counterpart uses packet coordinates restricted to projector ranges and proves both inverse laws. The finite Python routine checks the concrete array realization.

B.2 Weighted inverse at finite precision

```
Dinv = inverse(D)                       # certified on both sides
edge[e] = Dinv @ H_edge[e]
require Dinv @ H == sum(edge[e] for e in edges)
require edge[e] lies in filtration[weight[e]]
m = certified_lower_bound(weight)       # 1 <= m <= weight[e]
N = (L + m - 1) // m                    # so L <= m*N
S = 0
for n in range(N):
    for word in edge_tuples(edges, n):
        if sum(weight[e] for e in word) < L:
            S += (-1)**n * ordered_product(edge[e] for e in word)
return q_L(S @ Dinv)
```

Lean proves that the mapped pruned sum is exactly the mapped finite geometric inverse. The lower bound m gives $m|\gamma| \le W(\gamma)$, so $L \le mN$ derives $q_L(A^N) = 0$ from the filtration certificate and makes the returned value an inverse on both sides of $q_L(D + H)$. The Lean lemma $L \le m\lceil L/m \rceil$ justifies the displayed executable cutoff for $m > 0$. Taking $m = 1$ and $N = L$ gives the unconditional cutoff

for positive weights. The Python implementation checks the DVR matrix specialization against direct modular inversion.

B.3 Walk construction for transfer zeta

```
def coefficient(G, start, target, n):
    total = 0
    for length in range(n + 1):
        for word in edge_tuples(G, length):
            if is_walk(word, start, target) and weight(word) == n:
                total += 1
    return total

A[i,j] = sum(t**weight(e) for distinct edges e from i to j)
T = I - A
Q = det(T)
H = adjugate(T)
P = H[start, target]
assert Q.coeff(0) == 1
Z = coefficient_series(G, start, target)
prove Q * Z == P
```

Positive weights make the coefficient loop finite because every contributing word has length at most n. Lean proves the enumeration correct, derives its recurrence from the first edge, proves $(I - A)Z = I$, constructs the determinant/adjugate inverse, and identifies the two matrices by inverse uniqueness. Thus the normalized $QZ = P$ identity and its induced recurrence follow from finite graph data. The Python companion repeats the exact symbolic construction as an executable check. The Lean proof does not use it as an oracle.

C Proof dependency map

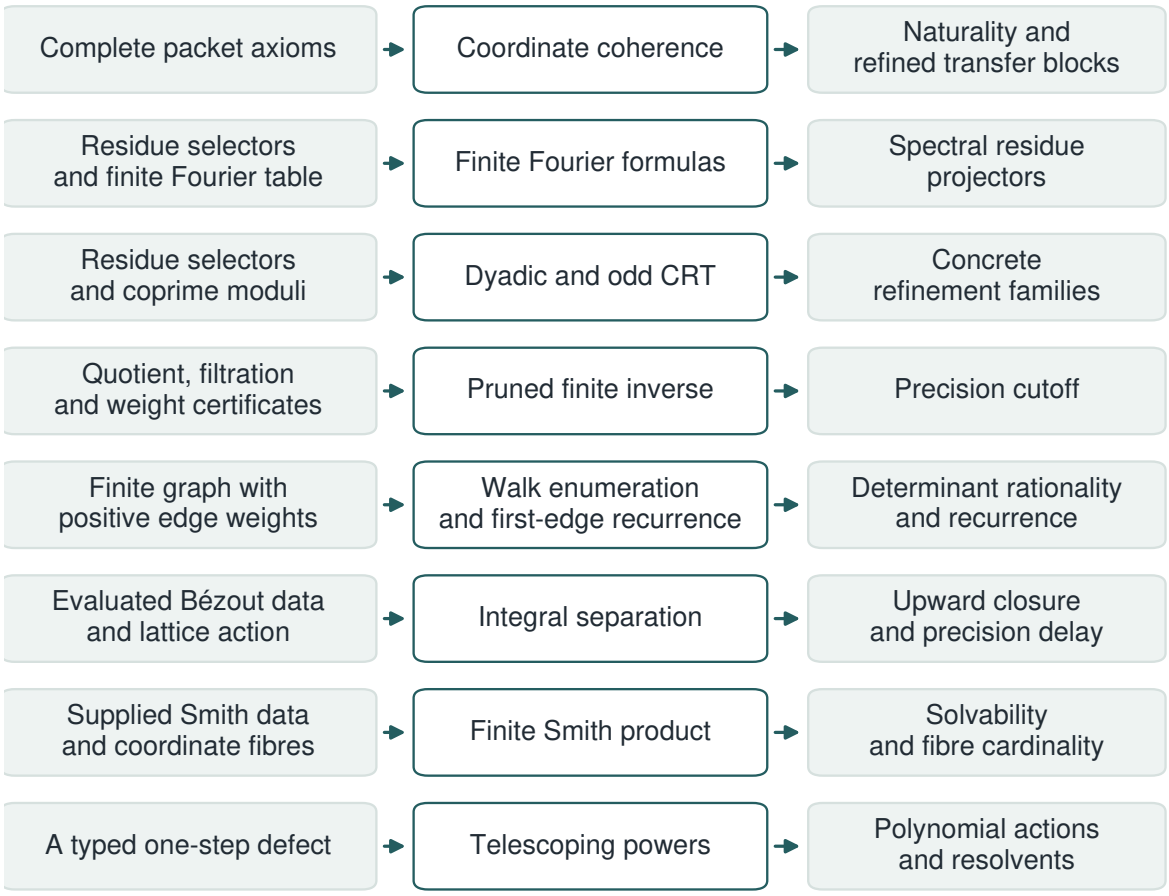
The proof order is:

complete packet axioms $\Rightarrow$ finite-packet uniqueness and coordinate coherence
$\Rightarrow$ coordinate naturality and refined transfer blocks,
residue selectors and a finite Fourier table $\Rightarrow$ Fourier formulas,
coprime moduli $\Rightarrow$ dyadic–odd formulas,
quotient, filtration, and edge-weight certificates $\Rightarrow$ explicit pruned inverse,
finite positive-weight graph $\Rightarrow$ semantic enumeration and first-edge recurrence
$\Rightarrow$ det / adj rationality and recurrence,
minimal evaluated Bézout data and inverse-unit lattice action
$\Rightarrow$ separation bound, upward closure, and precision delay,
concrete $\mathbb{Z}/p^L\mathbb{Z}$ coordinate fibres $\Rightarrow$ finite Smith product count,
one-step typed defect $\Rightarrow$ powers, polynomial actions, and witnessed resolvents.

The walk branch for transfer zeta is logically independent of operator inversion. Its rationality conclusion follows from finite graph data without a supplied $QZ = P$ premise. Likewise, CRT factorization follows directly from residue selectors and coprimality. It does not require the abstract refinement

Distinct proof branches

Each row identifies its own inputs and the conclusions they support.



The walk series is independent of operator inversion. General Smith and Bézout constructions remain upstream inputs to their certificate branches.

Figure 13. The dependency structure of Chapter II. Coordinate coherence, residue/CRT, weighted inversion, automatic finite-graph transfer rationality, and integral separation are distinct branches with explicit inputs.

theorem, although it supplies a concrete refinement family. General Smith diagonalization and the construction of polynomial Bézout/resultant data remain inputs to the formalized branches that use those certificates.

D Formal coverage and computational scope

This chapter develops common refinements, complete filtered inversion, lattice gluing, rectangular Smith congruences, and analytic and multilevel defects. It also includes the specializations, finite graph construction, worked realizations, and three algorithm specifications associated with the foundational formal contract.

Formal correspondence. The formal project is `formal/current/`. Its correspondence contains 81 source records, including 53 proof-level items. Chapter-wide targets and foundational records are kept separately, with exact source and proof hashes. The map and the Lean guide report each item's recorded status and verification checks.

Formal coverage. 53 of 53 proof-level source records have kernel checked, semantically accepted scope. No partial or pending proof records are listed in this correspondence. These are source-record counts under their stated hypotheses, not counts of independent discoveries.

The rectifiable analytic endpoints use a representation whose velocity length is bounded by the original geometric length. This inequality supplies the manuscript estimate. Equality of the two lengths is not required. The accepted correspondence identifies the exact canonical endpoints and their hypotheses.

The inherited Lean contract has 25 rows: 21 canonical numbered items, one Smith-phase identity, and three algorithm specifications. That foundational contract has its own source identity and scope. The chapter-wide correspondence separately records the exact statements, proof targets, source hashes, and execution evidence. A row may use several declarations, and several source labels may refer to one written theorem. These are distinct counting units.

The source archive preserves the foundational sources and their reports. Each verification record applies only to the source hashes and statement scopes it identifies. Preservation alone does not certify a changed statement.

The worked plates give exact models. The formal companion in the back matter explains the chapter-wide proof scope and the foundational declaration-level contract. Illustrations and finite checks support understanding and implementation. Universal claims depend on their written proofs and the exact formal statements where coverage is recorded.

The twenty-four worked plates follow this account of the proof scope. Their statement references identify the mathematical text of Chapter II.

E Visual atlas

Twenty-four worked plates follow the main constructions through explicit vectors, matrices, weighted walks, and congruence fibers. Their labels Plate II.1 to Plate II.24 stay fixed across the integrated atlas and the standalone guide. Each plate names the corresponding statement and its printed page in this monograph.

Three routes through the atlas

Exact coordinates. Read Plate II.1 to Plate II.8, then the finite filter in Plate II.20.
Weights and arithmetic. Read Plate II.9 to Plate II.19. The first group follows weighted operator words and graph walks. The second follows denominator clearing and Smith congruences.
Realizations and defects. Read Plate II.20 to Plate II.24. These plates separate the Smith impedance chart from Smith normal form, trace error propagation, and connect the examples to the formal contract.

The matrices and finite ranges shown here are explicit models. The general proofs remain in the preceding sections. A picture or finite computation does not extend the hypotheses of a theorem or the coverage of a Lean declaration.

How to use this volume

Twenty-four worked pages and twenty-two exact model figures accompany Chapter II and its formal companion. Integer, rational, modular and symbolic calculations determine the figures. The preserved Python model sources and their explicit finite verification ranges are included in the source package.

A route through the examples

The labels Plate II.1 through Plate II.24 identify worked pages independently of PDF pagination. Each page names its corresponding statement in the final mathematical text. The formal companion distinguishes chapter-wide coverage from the foundational contract.

Plate II.1 Three kinds of resolution
Plate II.2 An atlas of the refinement layer
Plate II.3 Follow one coefficient through the tower
Plate II.4 Regrouping is not averaging
Plate II.5 The operator changes coordinates too
Plate II.6 Selectors survive without division
Plate II.7 A Fourier certificate you can inspect
Plate II.8 One address, two independent residues
Plate II.9 Which words can survive precision five?
Plate II.10 An inverse computed in both orders
Plate II.11 The residual tells you when to stop
Plate II.12 Walks are not simple paths
Plate II.13 A finite graph produces its denominator
Plate II.14 Separate before dividing
Plate II.15 Where the dyadic digit is lost
Plate II.16 One extra digit is necessary
Plate II.17 See every Smith fiber modulo eight
Plate II.18 Transport the whole solution fiber
Plate II.19 Precision exposes directional freedom
Plate II.20 Filter first or split first: same output
Plate II.21 Two Smith constructions, different roles
Plate II.22 A small defect can grow under powers
Plate II.23 Resolvents amplify errors near poles
Plate II.24 Read the certificates with the pictures

Three reading routes

Exact coordinates: Plate II.3 to Plate II.8, followed by the finite filter in Plate II.20. **Precision and arithmetic:** Plate II.9 to Plate II.19. **Realizations and defects:** Plate II.20 to Plate II.24.

How to read an ordered product

In a written matrix word ab, the rightmost map b acts first. Directed walks may instead be listed in chronological edge order, as in the traditional text's block-path formula. Follow the convention stated with the example.

What the figures establish

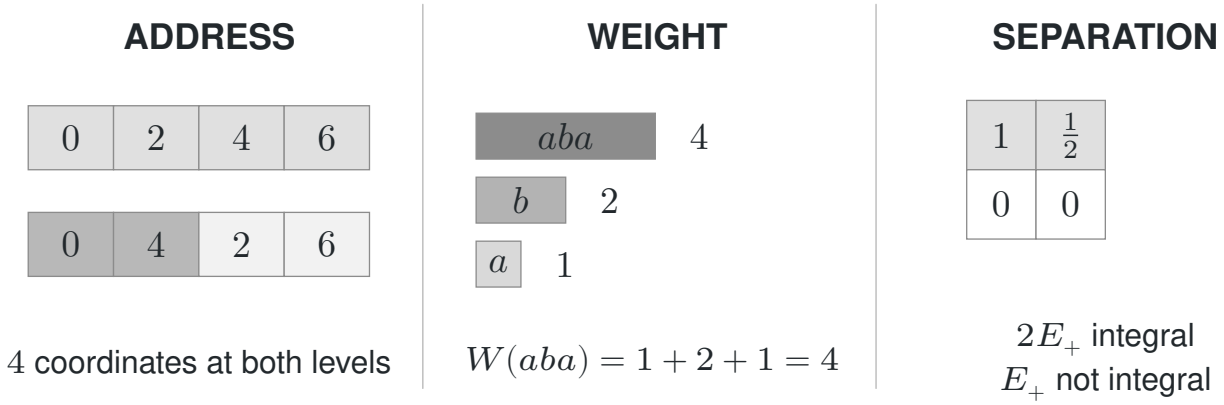
The models expose the mechanisms and their boundaries. Exact finite checks validate the displayed instances. Universal statements keep the hypotheses and proofs given in the mathematical text. Lean coverage is recorded by the chapter-wide correspondence and the foundational theorem map. Each compiler claim belongs to the source and logs identified by its verification record.

Plate II.1 / CHAPTER II / A WORKED VISUAL GUIDE

Three kinds of resolution

Chapter II, mathematical text: Definition 3.4, Theorem 5.10, Theorem 7.7

Chapter II asks three different questions that can all sound like "increase the resolution." This guide separates them, then follows small objects far enough that the distinctions can be computed rather than merely named.



Three different operations. On the left, all coefficients survive a regrouping. In the middle, an integer weight controls which operator words can matter modulo a chosen power. On the right, a rational projector needs a denominator cleared before integral precision can be recovered.

Address resolution

A coefficient such as x_6 can sit in the even packet, then the residue-2 packet modulo 4, then the residue-6 packet modulo 8. Its value does not become more accurate. Its *address becomes more specific*, while the full object remains exactly reconstructible.

Arithmetic precision

An operator word with total filtration weight at least L vanishes in the specified depth-L quotient. Increasing L can make additional words relevant. This is a change in the precision of the ring in which the answer is required, not a lossless rearrangement of one fixed list.

Separation precision

For the integral involution used later, $E_+ = (I + A)/2$ is an exact rational projector but not an integral one at 2. One extra input digit suffices to recover the projected difference to the requested precision. The packet is present over the fraction field even when its integral separation is delayed.

> **Read this as:** Keep three quantities distinct: the packet address, the precision cutoff, and the denominator weight. Finer addresses do not automatically improve arithmetic precision or remove a spectral denominator.

Plate II.2 / ORIENTATION / SYMBOLS WITH JOBS

An atlas of the refinement layer

Chapter II, mathematical text: Definition 3.2, Definition 3.4, Definition 7.6

The same letter can have different roles in different realizations. The table fixes the roles used here. Every small matrix in the guide is explicitly supplied. No diagonalization or classification theorem is inferred from a picture.

Object	Role	Worked instance
$P_{N,j}$	Retain coefficients with address j modulo N.	Even/odd masks, then mod-4 and mod-8 masks.
$\mathcal{A}_N, \mathcal{S}_N$	Split into range-restricted packet vectors, then sum them.	A permutation/interleaving of eight coordinates.
J, C	Exact refinement and regrouping. In the defect section, C may instead be any supplied coarsening map.	A permutation in Plate II.5 and a shear in Plate II.22.
T_{ba}	Input packet a, output packet b: $P_b T P_a$.	Blocks of a finite cyclic filter.
D, H, A	$T = D + H = D(I + A)$, with $A = D^{-1}H$.	Explicit integral 2×2 matrices.
$w(e), W(\gamma)$	Positive certified edge gain and its additive word weight.	$w(a) = 1$, $w(b) = 2$.
L	Precision depth in a quotient such as $\mathbb{Z}/2^L\mathbb{Z}$.	$L = 5$ gives arithmetic modulo 32.
$A(t), Z(t)$	Polynomial graph adjacency and unsigned walk-count series.	Two parallel edges remain distinct.
E, Λ, s	A rational projector, a specified lattice, and a clearing exponent.	E_+ has local weight 1 at 2.
(s_i)	Directional weights of a supplied Smith diagonal.	$(1, 2)$ modulo 8 or $(0, 1, 3)$ across depths.
Δ	The failure of an independently chosen operator to intertwine: $CT_f - T_c C$.	One nonzero entry $1/16$.

Suggested reading routes

For exact coordinates, read Plate II.3 through Plate II.8, then the finite filter on Plate II.20. For weighted inversion and counting, read Plate II.9 through Plate II.13. For arithmetic separation and congruence fibers, read Plate II.14 through Plate II.19. The final pages distinguish nonlinear chart geometry, error propagation, and the formal contract.

> **Read this as:** Operator words carry signed matrix values. Graph walks carry unsigned counts. Smith normal form concerns linear congruences. The Smith impedance chart on Plate II.21 is a different, nonlinear construction.

Plate II.3 / MODEL A / LOSSLESS ADDRESS REFINEMENT

Follow one coefficient through the tower

Chapter II, mathematical text: Theorem 3.3, Theorem 4.6

Take a vector $x = (x_0, \dots, x_{15})$. A packet is not a single summary number: it retains every coefficient at its selected addresses. For example,

$$P_{2,0}x = (x_0, 0, x_2, 0, \dots, x_{14}, 0).$$

The rail below lists the addresses in each packet. Its display order groups descendants under their parents. The matrix example on Plate II.5 instead uses increasing residue order.

	$r=0$								$r=1$							
$N=2$	0	2	4	6	8	10	12	14	1	3	5	7	9	11	13	15
					16											
	$r=0$				$r=2$				$r=1$				$r=3$			
$N=4$	0	4	8	12	2	6	10	14	1	5	9	13	3	7	11	15
					16											
	$r=0$		$r=4$		$r=2$		$r=6$		$r=1$		$r=5$		$r=3$		$r=7$	
$N=8$	0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15
					16											

Each cell is an address n; its coefficient travels with it.

Every row contains all sixteen addresses exactly once. A heavier boundary separates packet ranges. The numbers are coefficient addresses, not new coefficient values. There is no interpolation, averaging, or omitted high-frequency tail.

The address 6 is even, is 2 modulo 4, and is 6 modulo 8. Its coefficient therefore lies in the successive ranges

$$P_{8,6}X \subseteq P_{4,2}X \subseteq P_{2,0}X.$$

The corresponding identities hold as operators:

$$P_{2,0} = P_{4,0} + P_{4,2}, \qquad P_{4,2} = P_{8,2} + P_{8,6}.$$

Applying either composite parent selector to $P_{8,6}x$ leaves it unchanged. Applying the odd selector kills it. This is absorption between a parent and a child, rather than an approximate resemblance between nearby scales.

Why the reconstruction is unique

If a vector is a sum of components in disjoint packet ranges, apply the projector for one range. All other components vanish and the selected component remains. Thus every child component is recovered uniquely from the sum.

> **Read this as:** Refinement is a sharper address book for the same coefficients. The total number of free coordinates does not change.

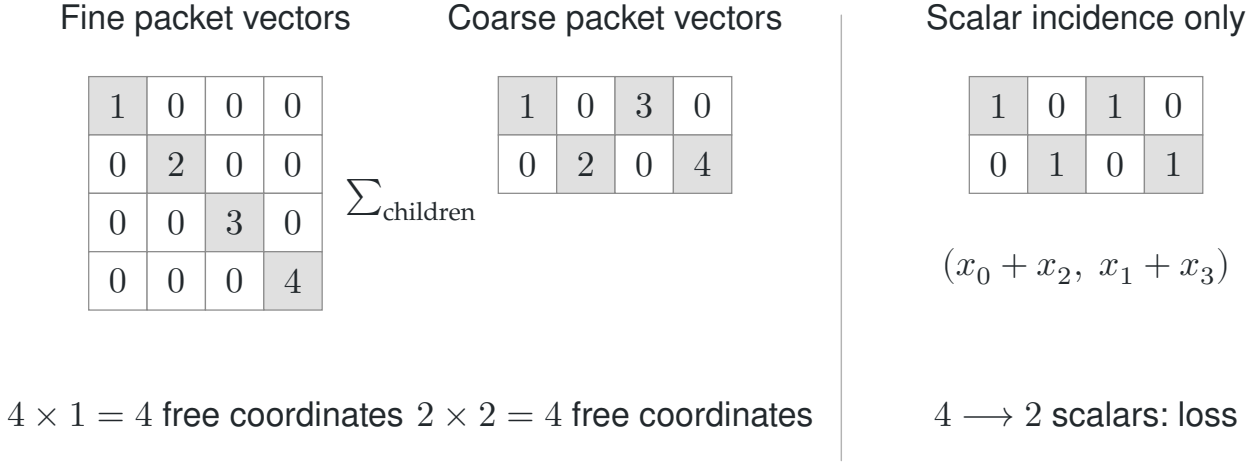
Plate II.4 / MODEL A / THE MEANING OF COARSENING

Regrouping is not averaging

Chapter II, mathematical text: Definition 3.4, Theorem 3.5

The word *coarsening* can suggest compression. In the exact coordinate theorem it means regrouping complete packet vectors. Replacing each packet by one scalar sum is a different operation and really does lose information.

For $x = (1, 2, 3, 4)$, the fine packets are the four coordinate-supported vectors. Regrouping gives $(1, 0, 3, 0)$ and $(0, 2, 0, 4)$. Every original coordinate remains available in one of these two vectors.



The left and middle panels store the same four independent coordinates. The right panel instead applies a 2×4 numerical incidence matrix to four scalar entries. That is not the range-restricted exact coordinate map.

Define

$$X_N^{\mathrm{pkt}} = \bigoplus_j P_{N,j} X, \qquad \mathcal{A}_N x = (P_{N,j} x)_j, \qquad \mathcal{S}_N (x_j)_j = \sum_j x_j.$$

Completeness and mutual annihilation give both inverse laws. Consequently,

$$J = \mathcal{A}_4 \mathcal{S}_2, \qquad C = \mathcal{A}_2 \mathcal{S}_4, \qquad CJ = I, \quad JC = I.$$

The larger list of fine packets has smaller individual packet spaces.

A kernel witnesses genuine information loss

For the scalar incidence matrix B drawn on the right,

$$B(1, 0, -1, 0)^{\mathsf{T}} = 0.$$

The zero vector and $(1, 0, -1, 0)$ therefore have the same two scalar summaries. No inverse can recover the original four scalars from that output alone.

> **Read this as:** Ask what each packet stores. A list of full packet vectors can be exactly invertible. A list of packet means or totals generally is not.

Plate II.5 / MODEL A / COHERENCE AND NATURALITY

The operator changes coordinates too

Chapter II, mathematical text: Theorem 3.5, Theorem 3.8

Use a finite periodic carrier $X = \mathbb{Q}^8$ for this example. Let $Se_j = e_{j+1 \bmod 8}$ and let

$$T = I + 2S + S^2.$$

This is a concrete cyclic model, not an unannounced boundary convention for the finitely supported signal on Plate II.20. Write R_N for the permutation that groups coordinates in increasing residue order modulo N.

$\widehat{T}_2$

1	·	·	1	·	·	·	2
1	1	·	·	2	·	·	·
·	1	1	·	·	2	·	·
·	·	1	1	·	·	2	·
2	·	·	·	1	·	·	1
·	2	·	·	1	1	·	·
·	·	2	·	·	1	1	·
·	·	·	2	·	·	1	1

$J_{4,2}$

1	·	·	·	·	·	·	·
·	·	1	·	·	·	·	·
·	·	·	·	1	·	·	·
·	·	·	·	·	·	1	·
·	1	·	·	·	·	·	·
·	·	·	1	·	·	·	·
·	·	·	·	·	1	·	·
·	·	·	·	·	·	·	1

$\widehat{T}_4$

1	·	·	·	·	1	·	2
·	1	·	·	1	·	2	·
2	·	1	·	·	·	·	1
·	2	·	1	·	·	1	·
1	·	2	·	1	·	·	·
·	1	·	2	·	1	·	·
·	·	1	·	2	·	1	·
·	·	·	1	·	2	·	1

$$J_{4,2}\widehat{T}_2 = \widehat{T}_4 J_{4,2}$$

Actual 8×8 matrices. Dots are zero entries. The other entries are exact integers. Heavy lines mark the packet blocks: size four at level two and size two at level four.

The reordered operators and change of coordinates are

$$\widehat{T}_N = R_N T R_N^{-1}, \qquad J_{4,2} = R_4 R_2^{-1}.$$

Multiplication cancels the intermediate coordinate maps:

$$J_{4,2}\widehat{T}_2 = \widehat{T}_4 J_{4,2}, \qquad J_{8,4}J_{4,2} = J_{8,2}.$$

Thus it makes no difference whether the operator is applied before or after a consistent coordinate change. One must transport the operator, not merely reuse the same numerical matrix under a new ordering.

Refined blocks still reconstruct the same map

A coarse block expands as

$$P_b T P_a = \sum_{\beta \succ b,\, \alpha \succ a} P_\beta T P_\alpha.$$

The terms are maps embedded in the original source and target spaces. Adding their norms, or adding their matrix entries after forgetting the embeddings, is not this operator identity.

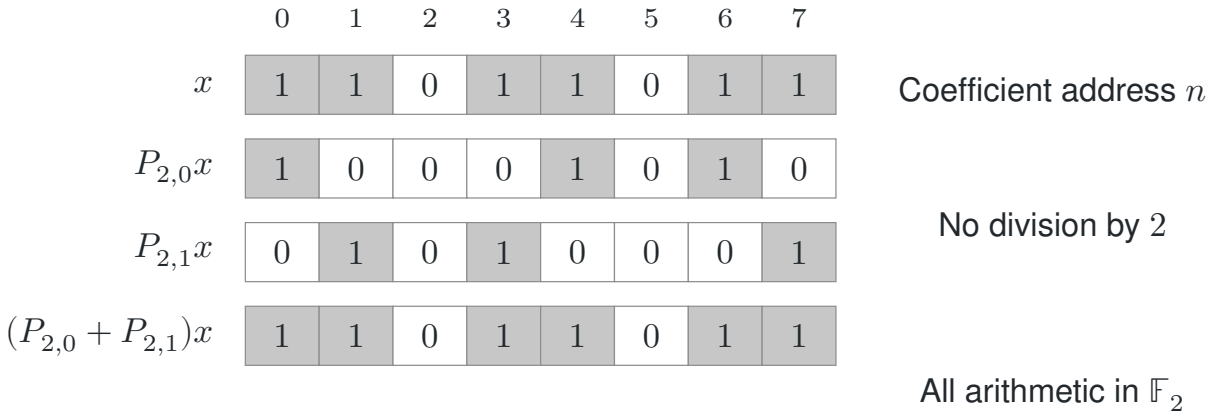
> **Read this as:** The fine matrix contains a different block organization of the same operator. An independently guessed coarse matrix is a separate approximation problem, measured later by a defect.

Plate II.6 / MODEL A / RESIDUE MASKS IN CHARACTERISTIC TWO

Selectors survive without division

Chapter II, mathematical text: Proposition 4.2

Work first in $\mathbb{F}_2$, where $1+1=0$. The eight coefficients below are just a finite display of a coefficient sequence. The even and odd projectors are diagonal masks with entries zero and one.



Each nonzero input coefficient is retained by exactly one mask. Reconstruction is exact over $\mathbb{F}_2$. No average and no root of unity is used.

For a general coefficient ring and a positive integer N, the definition is

$$(P_{N,j}x)_n = \begin{cases} x_n, & n \equiv j \pmod N, \\ 0, & \text{otherwise.} \end{cases}$$

An address belongs to one residue class, so applying two masks either keeps that same coefficient or kills it. This proves

$$P_{N,i}P_{N,j} = \delta_{ij}P_{N,j}, \qquad \sum_j P_{N,j} = I.$$

The argument is coordinatewise and does not require the coefficients to form a field.

Why the usual even/odd average is a different formula

Over a field of characteristic not two, parity extraction can be represented using $F(z)$ and $F(-z)$ and division by 2. In characteristic two, $-z = z$ and $2 = 0$, so that formula cannot be used. The coefficient mask, however, still distinguishes even from odd *exponents*.

This distinction also explains the later involution example. Failure to reduce a rational spectral projector containing $1/2$ does not make every integral coefficient-address selector disappear.

> **Read this as:** The basic object is the address selector. A Fourier average is one possible realization of it and has additional algebraic hypotheses.

Plate II.7 / MODEL A / AN EXACT NINE-ENTRY TABLE

A Fourier certificate you can inspect

Chapter II, mathematical text: Theorem 4.3

Take $N = 3$ and work in $\mathbb{F}_7$. Set

$$\omega = 2, \qquad \omega^{-1} = 4, \qquad \eta = 3^{-1} = 5.$$

Here $2^3 = 1$ and neither 2 nor 2^2 is one. The unnormalized geometric sums are the entries in the first matrix.

$\sum_k 4^{jk} 2^{rk} \pmod 7$ $\qquad$ $P_j(z^r) : \ 5 = 3^{-1} \pmod 7$

3	0	0
0	3	0
0	0	3

$\times 5$

1	0	0
0	1	0
0	0	1

$$\omega = 2, \quad \omega^3 = 1$$

$$\omega^{-1} = 4, \quad \eta = 5$$

$$j, r \in \{0, 1, 2\}$$

In $\mathbb{F}_3 \times \mathbb{F}_3$: order two does not imply orthogonality.

$$\omega = (-1, 1), \quad \eta = (2, 2), \quad \eta(1 + \omega) = (0, 1) \neq (0, 0)$$

All entries are exact residues modulo seven. Multiplying the diagonal value 3 by 5 produces 1. The lower calculation is a genuine failed Fourier certificate in a different coefficient ring.

On the monomial z^n, define $UF(z) = F(2z)$. The proposed projector acts by

$$5 \sum_{k=0}^{2} 4^{jk} 2^{nk}.$$

The identity $2^3 = 1$ makes this depend only on n mod 3. The nine-entry table therefore establishes the formula for every exponent once periodicity is used, rather than requiring a separate test at every coefficient.

Order alone is not sufficient over every ring

In $\mathbb{F}_3 \times \mathbb{F}_3$, the element $(-1, 1)$ has order two, but one component is a trivial character. Its nontrivial Fourier sum does not vanish in that component. Thus an element of order N in a ring with zero divisors is not, by itself, the full orthogonality certificate.

> **Read this as:** The finite table is a checkable sufficient input. Over a field, a primitive root supplies it. Over a more general ring, inspect the actual table instead of inferring orthogonality from order alone.

Plate II.8 / MODEL A / THE DYADIC AND ODD RECTANGLE

One address, two independent residues

Chapter II, mathematical text: Corollary 4.9

Because 4 and 3 are coprime, a residue modulo 12 is specified by a column $a = n \bmod 4$ and a row $b = n \bmod 3$. Each cell contains the unique combined address.

$$a = n \bmod 4$$

	$a=0$	$a=1$	$a=2$	$a=3$
$b=0$	0	9	6	3
$b=1$	4	1	10	7
$b=2$	8	5	2	11

$$b = n \bmod 3$$

$$(a,b) = (3,1) \iff n = 7 \pmod{12}$$

$$7,\ 19,\ 31, \ldots$$

$$P_{12,7} = P_{4,3}P_{3,1}$$

The twelve cells are a permutation of $0, \ldots, 11$, not twelve extra degrees of freedom. The selected cell contains every integer congruent to 7 modulo 12.

An explicit reconstruction formula is

$$n \equiv 9a + 4b \pmod{12},$$

because 9 is one modulo four and zero modulo three, while 4 has the opposite pair of residues. For $(a,b) = (3,1)$ this gives $31 \equiv 7$.

Applying the two masks in either order gives

$$P_{12,7} = P_{4,3}P_{3,1} = P_{3,1}P_{4,3}.$$

More generally, an address survives exactly when it satisfies both congruences. The coprime reconstruction identifies this intersection with one combined residue class.

The hypothesis can be seen failing

Replace 4 and 3 by 4 and 6. The requirements $n \equiv 0 \pmod 4$ and $n \equiv 1 \pmod 6$ are incompatible: the first makes n even and the second makes it odd. A full rectangular address product no longer exists. Compatible pairs must agree modulo the greatest common divisor.

> **Read this as:** CRT separates address coordinates. It does not claim that dyadic and odd spectral projectors have the same denominator weights or the same lifting behavior.

Plate II.9 / MODEL B / ORDERED TRANSFERS IN A FILTERED RING

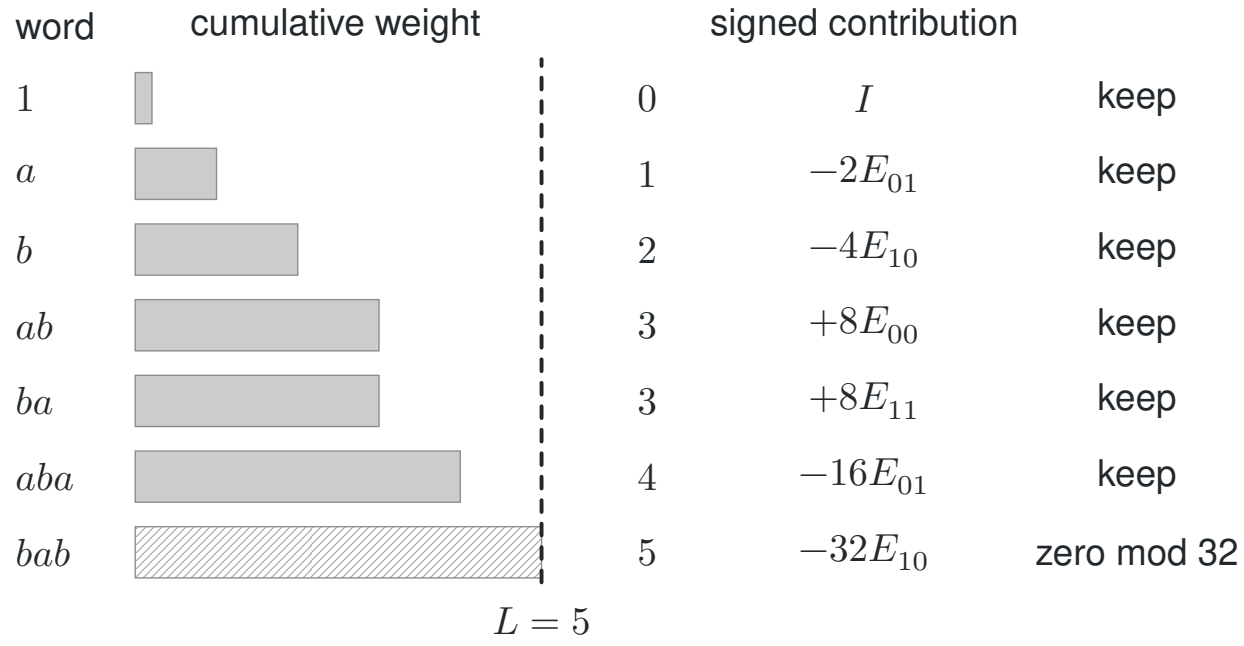
Which words can survive precision five?

Chapter II, mathematical text: Theorem 5.10, Corollary 5.11

Let E_{ij} send the jth basis vector to the ith. In integral 2×2 matrices set

$$a = 2E_{01}, \qquad b = 4E_{10}, \qquad A = a + b.$$

Their certified dyadic weights are 1 and 2. At precision $L = 5$, every word of weight at least five vanishes modulo 32.



All nonzero retained terms of the signed inverse sum, followed by the first discarded nonzero word. The empty word contributes I. Other words of weight below five can still be zero because $a^2 = b^2 = 0$.

The words are *written matrix products*: in ab, the map b acts first. Therefore

$$ab = 8E_{00}, \qquad ba = 8E_{11}, \qquad aba = 16E_{01}.$$

The two products ab and ba cannot be identified merely because they have equal weight. Both their order and their source/target information matter.

The pruning rule supplies a sufficient reason for a term to vanish. It is not an assertion that every word below the threshold contributes nontrivially. Packet incompatibility and algebraic relations can remove additional words.

> **Read this as:** Length controls the sign. Cumulative weight controls precision. The ordered matrix product controls the actual contribution. None of these three can replace the others.

Plate II.10 / MODEL B / A COMPLETE MODULO-32 CALCULATION

An inverse computed in both orders

Chapter II, mathematical text: Theorem 5.10, Corollary 5.11

Keep the matrices from Plate II.9 and choose

$$D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad D^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \quad T = D(I + A) = \begin{pmatrix} 5 & 3 \\ 4 & 1 \end{pmatrix}.$$

The six retained contributions sum to

$$S = I - a - b + ab + ba - aba = \begin{pmatrix} 9 & -18 \\ -4 & 9 \end{pmatrix}.$$

The required inverse is $K = SD^{-1}$ in the quotient, not $D^{-1}S$.

T		K		$TK \pmod{32}$		$KT \pmod{32}$	
5	3	9	5	1	0	1	0
4	1	28	13	0	1	0	1

$$S = \begin{pmatrix} 9 & -18 \\ -4 & 9 \end{pmatrix}, \qquad K = SD^{-1} \pmod{32}$$

The displayed K uses canonical residues from 0 to 31. Both left and right products are the identity modulo 32. These are exact modular equalities, not floating-point residuals.

One can verify the claim without trusting the word expansion:

$$TK = \begin{pmatrix} 129 & 64 \\ 64 & 33 \end{pmatrix} \equiv I, \qquad KT = \begin{pmatrix} 65 & 32 \\ 192 & 97 \end{pmatrix} \equiv I \pmod{32}.$$

The cancellation of the exterior factor D explains the order. Since $T = D(I + A)$, multiply its inverse by D^{-1} on the *right* of the inverse of $I + A$.

What the finite theorem does not require

There is no claim here that the same integer series converges in the ordinary real matrix norm. The quotient kills all sufficiently deep products, so a finite sum suffices. The complete normed-ring theorem is a separate result with its own strict contraction hypothesis.

> **Read this as:** A correct noncommutative formula must work on both sides and preserve the order of its factors. Here the entire result can be checked by two integer matrix products.

Plate II.11 / MODEL B / A CERTIFIED CUTOFF, NOT A HEURISTIC

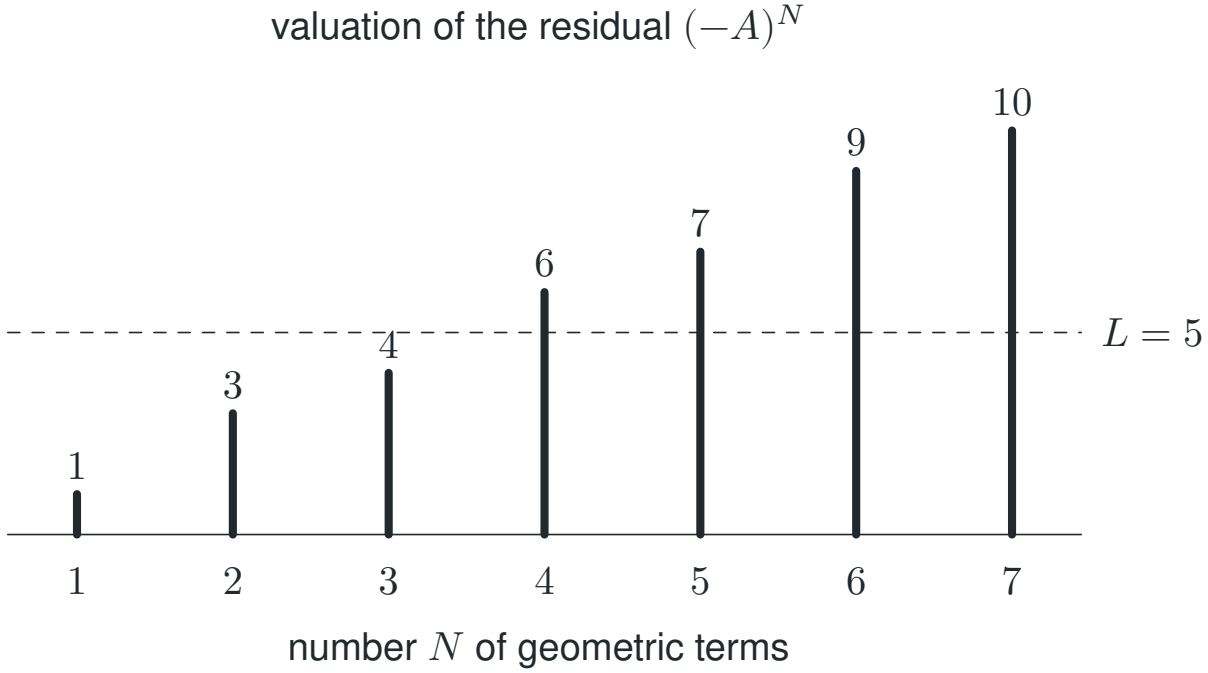
The residual tells you when to stop

Chapter II, mathematical text: Theorem 5.10, Corollary 5.11, Section B.2

For $S_N = \sum_{k=0}^{N-1} (-A)^k$, telescoping gives

$$(I + A)S_N = S_N(I + A) = I - (-A)^N.$$

Thus the exact residual is known. To certify precision L, it is enough to show that every residual entry is divisible by 2^L.



The plotted values are the minimum dyadic valuations of the nonzero entries of $(-A)^N$. The dashed line is the requested cutoff $L = 5$. This is an exact valuation profile, not a measured numerical convergence curve.

In this particular model, $A^2 = 8I$. Therefore

$$v_2(A^{2k}) = 3k, \qquad v_2(A^{2k+1}) = 3k + 1,$$

where the matrix valuation means the minimum entry valuation. Three geometric terms leave a residual of valuation four and do not suffice modulo 32. Four terms leave valuation six and do suffice.

The generic positive-weight estimate is more conservative. If every edge weight is at least $m > 0$, a word of length N has weight at least mN. Hence the cutoff

$$N = \left\lceil \frac{L}{m} \right\rceil$$

is sufficient. Here the certified minimum weight is $m = 1$, so the generic choice is five. The extra identity $A^2 = 8I$ justifies stopping at four in this example.

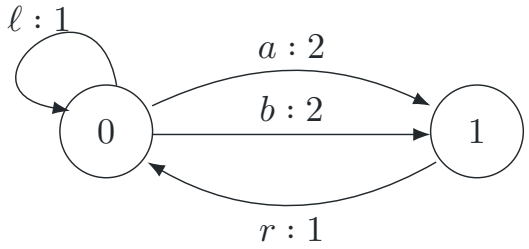
> **Read this as:** A structural identity may improve a cutoff. An observed small real norm or an arbitrarily larger claimed minimum edge weight does not supply that certificate.

Plate II.12 / MODEL C / A GRAPH WITH PARALLEL EDGES

Walks are not simple paths

Chapter II, mathematical text: Theorem 6.6

Consider the directed graph below. The loop ℓ has weight one, the two distinct edges a, b from 0 to 1 each have weight two, and the return edge r has weight one.



Edge labels carry positive integer weights.

All walks from 0 to 1 of weight 5

$\ell\ell\ell a$	$\ell\ell\ell b$
ara	arb
bra	brb

Every word on the right is read chronologically from left to right as an edge traversal. This is intentionally different from the right-to-left action of a written operator product in Plate II.9.

A walk may revisit a vertex and reuse an edge. At total weight five there are six walks from 0 to 1. Two use three loops followed by either a or b. Four travel to 1, return to 0, and travel to 1 again, with two independent choices of parallel edge.

Their multiplicities enter the polynomial adjacency matrix:

$$A(t) = \begin{pmatrix} t & 2t^2 \\ t & 0 \end{pmatrix}.$$

The coefficient 2 remembers the two distinct edges. Replacing it by 1 would count a different graph.

The first-edge decomposition

Let $a_n(s,t)$ count walks of total weight n from s to t. Removing the first edge gives

$$a_n(s,t) = \mathbf{1}_{s=t,\, n=0} + \sum_{\substack{e:\, \mathrm{source}(e)=s \\ w(e)\le n}} a_{n-w(e)}(\mathrm{target}(e), t).$$

The empty walk is allowed precisely on the diagonal at weight zero. Positive weights ensure that every contributing walk has length at most n, so enumeration at each fixed weight is finite.

> **Read this as:** The graph series counts unsigned walks, including repetitions and parallel-edge multiplicity. It does not count simple paths, and it does not know whether an associated operator product vanishes or cancels.

Plate II.13 / MODEL C / AUTOMATIC RATIONALITY AND RECURRENCE

A finite graph produces its denominator

Chapter II, mathematical text: Theorem 6.6, Corollary 6.7

For the graph on Plate II.12, compute the determinant and adjugate rather than guessing a recurrence:

$$I - A(t) = \begin{pmatrix} 1-t & -2t^2 \\ -t & 1 \end{pmatrix}, \quad Q(t) = 1 - t - 2t^3, \quad \operatorname{adj}(I-A)_{0,1} = 2t^2.$$

Since $Q(0) = 1$, its inverse exists as a formal power series with integer coefficients.

$$Z_{0,1}(t) = \frac{2t^2}{1 - t - 2t^3}$$

n	0	1	2	3	4	5	6	7	8	9	10	11	12
a_n	0	0	2	2	2	6	10	14	26	46	74	126	218

$$a_8 = a_7 + 2a_5 = 14 + 2 \cdot 6 = 26$$

The coefficients are independently computed by the first-edge recurrence and checked against the rational series through weight 24. Literal edge-walk enumeration supplies a third check through weight 10.

From $QZ_{0,1} = 2t^2$ one obtains

$$a_0 = a_1 = 0, \quad a_2 = 2, \qquad a_n = a_{n-1} + 2a_{n-3} \quad (n \geq 3).$$

The numerator is responsible for the initial exception. Omitting it would incorrectly force the entire series to vanish.

Three boundaries of this calculation

This is a *formal* series identity. No analytic convergence at a chosen real or complex value of t is needed. The canonical determinant need not be the minimal denominator after cancellation. Finally, positive weights prevent infinitely many walks at a fixed weight: a single weight-zero loop would already give the infinitely many distinct walks $1, \ell, \ell^2, \ldots$ at weight zero.

> **Read this as:** Finite positive-weight graph data determine a rational series and an eventual recurrence. The proof does not begin by assuming a certificate $QZ = P$ that already contains that conclusion.

Plate II.14 / MODEL D / AN EVALUATED BÉZOUT CERTIFICATE

Separate before dividing

Chapter II, mathematical text: Theorem 7.5, Definition 7.6

Use the integral involution

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \qquad A^2 = I.$$

With $F = A - I$ and $G = A + I$, the identities

$$-F + G = 2I, \qquad FG = 0$$

are an explicit evaluated Bézout certificate. Here all the required commutations follow because these operators are polynomials in the same matrix.

A

1	1
0	-1

$N = I + A$

2	1
0	0

$E_+ = N/2$

1	$\frac{1}{2}$
0	0

$$N^2 = 2N$$

$$E_+^2 = E_+$$

$$E_+(1,0) = (1,0), \qquad E_+(-1,2) = (0,0)$$

The denominator-free numerator is quasi-idempotent: $N^2 = 2N$. Division by the nonzero scalar 2 gives an idempotent over the fraction field. It does not yet make the projector integral on the selected lattice.

The numerator $N = G$ acts by 2 on $\ker(A - I)$ and by zero on $\ker(A + I)$. Thus

$$E_+ = \frac{1}{2}N = \begin{pmatrix} 1 & 1/2 \\ 0 & 0 \end{pmatrix}$$

is the required separator. The complementary projector is $E_- = I - E_+$.

A denominator bound need not be minimal

The same projector can be presented with the inflated certificate $-2F + 2G = 4I$. Its numerator is $2G$ and its normalization is still E_+. This supplies a two-digit clearing bound at 2, although one digit is enough. Thus the valuation of a supplied certificate gives an upper bound, not automatically the least possible weight.

For the original sign factors the concrete resultant is $\mathrm{Res}(x-1, x+1) = 2$ and the bound is sharp. That equality is special to this displayed calculation, not a universal minimality theorem for every resultant certificate.

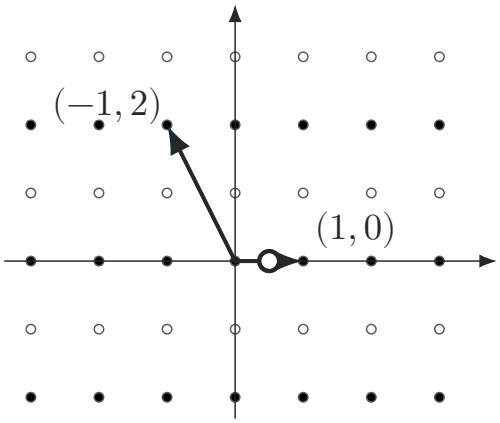
> **Read this as:** First check the evaluated identities and the numerator. Then normalize in the fraction field. Finally ask what denominator is needed on the particular lattice.

Plate II.15 / MODEL D / THE EIGENLATTICE HAS INDEX TWO

Where the dyadic digit is lost

Chapter II, mathematical text: Proposition 9.3

The eigenspaces are spanned by $(1,0)$ and $(-1,2)$. Their integral eigenvectors do not form an integral basis of $\mathbb{Z}^2$. The change-of-basis determinant is two.



$$E_+(0,1) = (\tfrac{1}{2}, 0)$$

$$B = \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix}, \quad \det B = 2$$

Filled: $B\mathbb{Z}^2 = \{(x,y) : y \text{ is even}\}$

Unfilled: the other lattice coset.

B^{-1} is integral at odd primes, not at 2.

A picture of the ordinary integer lattice, not a drawing of the topology of $\mathbb{Q}_2$. Filled dots have even second coordinate. Unfilled dots lie in the other coset. The hollow point at $(1/2, 0)$ records the image of an integral vector under E_+.

Write

$$B = \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix}, \qquad AB = B\,\mathrm{diag}(1,-1), \qquad E_+ = B\,\mathrm{diag}(1,0)B^{-1}.$$

The index-two eigenlattice explains the denominator in B^{-1} and in the projector. Over the localized lattice $\Lambda_p = \mathbb{Z}^2_{(p)}$, the number 2 is a unit for every odd prime p. Hence the local denominator weight is zero there and one at $p = 2$.

This is a *local* assertion. Odd localization does not change the fact that the displayed eigenvectors span an index-two sublattice inside the ordinary integer lattice.

The special fiber does not retain the rational splitting

Modulo two, $x - 1 = x + 1$ and A becomes a nontrivial unipotent matrix. The entries $1/2$ in $E_\pm$ cannot be reduced modulo two. By contrast, the coefficient selectors on Plate II.6 use only zero and one and remain available.

> **Read this as:** The obstruction is in integral separation, not in the existence of the two rational eigenspaces. The lattice is part of the question.

Plate II.16 / MODEL D / A SHARP EXAMPLE OF PRECISION DELAY

One extra digit is necessary

Chapter II, mathematical text: Theorem 7.7, Proposition 9.3

The projector acts by

$$E_+(x, y) = (x + y/2, 0).$$

Its factor $1/2$ can consume one dyadic digit. The claim concerns differences: if the inputs agree modulo $2^{m+1}\Lambda$, their projected differences lie in $2^m\Lambda$.

input pair	difference	same mod 8?	E_+ difference	same mod 8?
$0, (0, 8)$	$(0, 8)$	yes	$(4, 0)$	no
$0, (0, 16)$	$(0, 16)$	yes	$(8, 0)$	yes

$$m = 3: \quad 3 \text{ input digits fail; } 4 \text{ input digits suffice.}$$

For $m = 3$, inputs differing by $(0, 8)$ agree modulo eight, but their projected outputs differ by $(4, 0)$ and do not. Raising the input agreement to modulo sixteen makes the projected outputs agree modulo eight.

The example works at every positive depth:

$$E_+(0, 2^m) = (2^{m-1}, 0) \notin 2^m\Lambda, \qquad E_+(0, 2^{m+1}) = (2^m, 0) \in 2^m\Lambda.$$

Thus a zero-delay promise fails, while one extra input digit is sufficient. This witnesses sharpness for the specified projector and lattice.

The general calculation is just as short

If $\pi^s E\Lambda \subseteq \Lambda$ and $x - y = \pi^{m+s}v$, then

$$E(x - y) = \pi^m(\pi^s Ev) \in \pi^m\Lambda.$$

No rounding procedure is hiding in this identity. Nor does it authorize reducing E modulo π when its entries are nonintegral. Individual projected points may lie in a fractional lattice. The theorem controls their difference at the stated integral scale.

Accuracy and branching are different views

The later Smith example records how many inputs share one output at finite depth. Here we ask how accurately an input must be known to determine a projected output difference. Both involve directional divisibility, but they are different questions.

> **Read this as:** The phrase “one lost digit” has an exact meaning: both a sufficiency theorem and an explicit pair of inputs showing that a smaller delay fails.

Plate II.17 / MODEL E / TWO DIRECTIONAL WEIGHTS

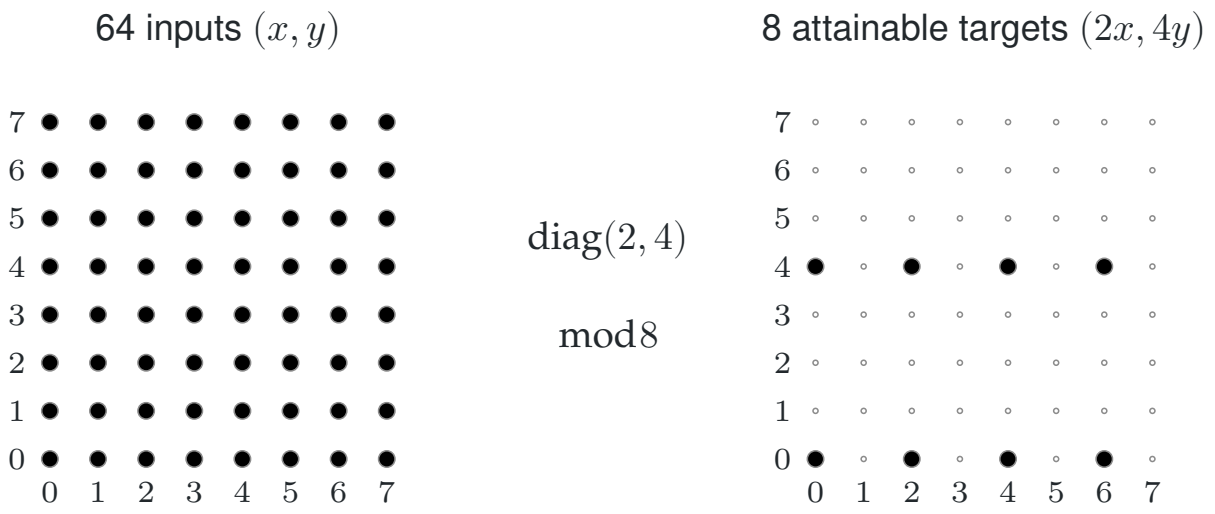
See every Smith fiber modulo eight

Chapter II, mathematical text: Theorem 10.6

Consider the actual map of finite sets

$$(\mathbb{Z}/8\mathbb{Z})^2 \longrightarrow (\mathbb{Z}/8\mathbb{Z})^2, \qquad (x, y) \longmapsto (2x, 4y).$$

The first coordinate of the output must be even. The second must be zero or four. Those are precisely the attainable targets in the right panel.



Every filled target has exactly $2 \cdot 4 = 8$ preimages.

All sixty-four input points and all sixty-four possible target addresses are shown. The eight filled target points are the image. Every attainable target has exactly eight preimages. The other fifty-six targets have none.

For a specific attainable target $(2, 4)$,

$$2x \equiv 2 \pmod 8 \iff x \in \{1, 5\}, \qquad 4y \equiv 4 \pmod 8 \iff y \in \{1, 3, 5, 7\}.$$

The coordinate choices are independent, giving $2 \cdot 4 = 8$ solutions. The entire carrier is partitioned consistently:

$$8 \text{ attainable targets} \times 8 \text{ inputs per target} = 64.$$

For an inadmissible target, such as $(1, 4)$ or $(2, 2)$, the fiber is empty. The same nonzero cardinality formula must not be applied to it.

The arithmetic behind the picture

Multiplication by p^s modulo p^L has kernel size $p^{\min(s,L)}$. Its image consists exactly of residues whose canonical representatives are divisible by that same power. The displayed model has $p = 2$, $L = 3$ and weights $(1, 2)$.

> **Read this as:** A directional weight has two consequences: a divisibility condition on the target and a conditional multiplicity of inputs. Solvability comes before counting.

Plate II.18 / MODEL E / EXPLICIT SMITH COORDINATES

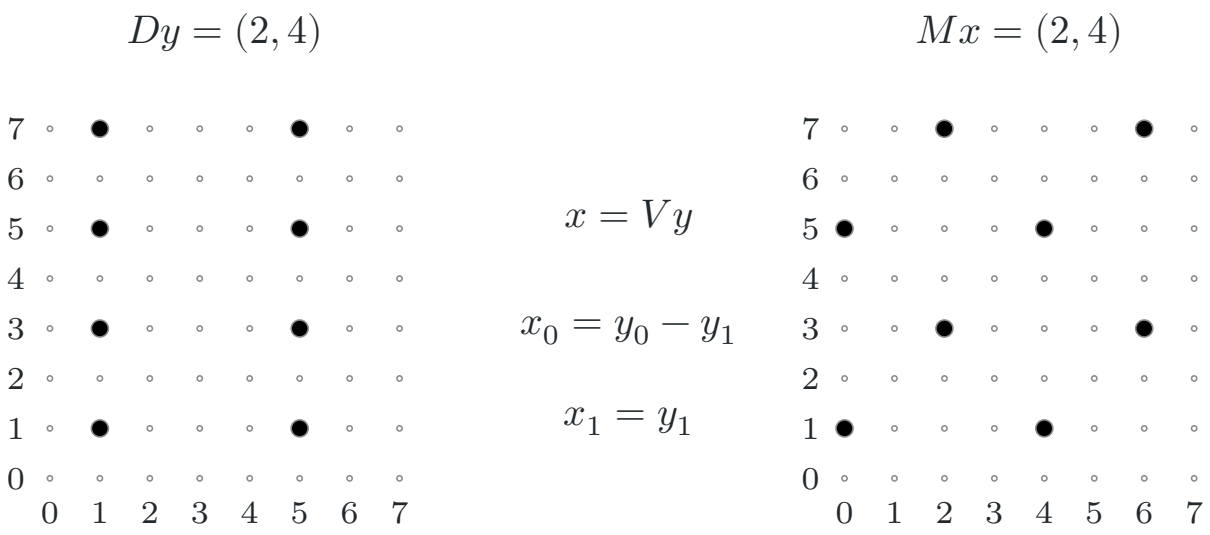
Transport the whole solution fiber

Chapter II, mathematical text: Theorem 10.6

The preceding diagonal system is connected to a nondiagonal one by the unimodular change of variables

$$M = \begin{pmatrix} 2 & 2 \\ 0 & 4 \end{pmatrix}, \quad V = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \quad MV = \operatorname{diag}(2,4) = D.$$

Since $\det V = 1$, $x = Vy$ is bijective modulo eight. Thus $Mx = b$ is equivalent to $Dy = b$.



The eight solutions are permuted, not created or discarded.

The eight solutions for the target $(2,4)$, before and after the coordinate shear. Coordinates are residues in $\{0,\dots,7\}$. No interpolation on this modular grid is intended.

For example, the diagonal solution $y = (1,3)$ corresponds to

$$x = Vy = (-2,3) \equiv (6,3) \pmod 8,$$

and $M(6,3)^{\mathsf{T}} = (18,12)^{\mathsf{T}} \equiv (2,4)^{\mathsf{T}}$. Applying the inverse shear recovers the original solution.

Two boundary cases worth keeping visible

A unit diagonal coordinate has exactly one solution for every target. A literal zero coordinate has all eight possible inputs when its target is zero, and no input when the target is nonzero. A zero coordinate is not inverted or silently omitted.

The displayed diagonalization is verified by multiplication. The theorem takes diagonal data as input. A Smith normal form for an arbitrary matrix or general DVR requires a separate structure theorem.

> **Read this as:** An invertible change of coordinates transports existence, every solution, and the fiber count. The target must be transformed as prescribed when checking admissibility.

Plate II.19 / MODEL E / A FIBER BUDGET AT EACH DEPTH

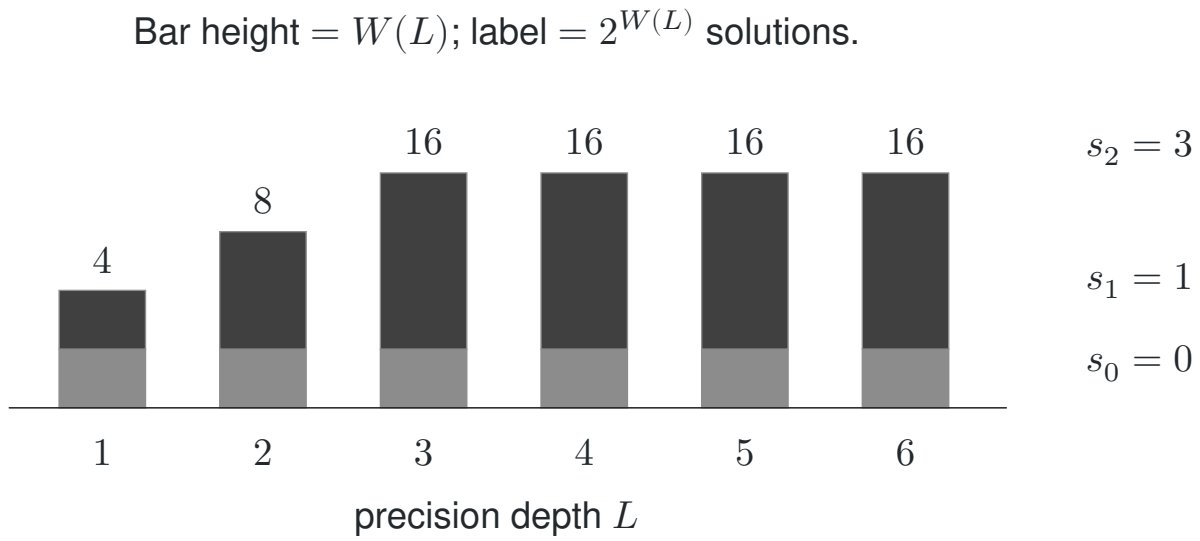
Precision exposes directional freedom

Chapter II, mathematical text: Theorem 10.6

Take $p = 2$ and the supplied diagonal weights $(s_0, s_1, s_2) = (0, 1, 3)$. At depth L, an admissible target has

$$2^{W(L)} \text{ solutions}, \qquad W(L) = \min(0, L) + \min(1, L) + \min(3, L).$$

Each direction contributes only the amount visible at the current depth.



The stacked height records the exponent $W(L)$. The label above each bar records the actual number of solutions. The zero-weight direction contributes no branching. This is a finite cardinality calculation, not an information-theoretic assumption about random inputs.

The values are

L	1	2	3	4	5	6
$W(L)$	2	3	4	4	4	4
# solutions	4	8	16	16	16	16.

Once $L \geq 3$, the nonzero weights have been fully exposed and the admissible fiber cardinality stabilizes at sixteen. The input space still grows as 2^{3L}. The number of attainable targets grows accordingly.

At depth six, $(1, 2, 8)$ is admissible and gives sixteen solutions. Replacing the last coordinate by four makes the target inadmissible and gives zero solutions, even though the weights and depth are unchanged.

What changes with a zero diagonal factor?

A literal zero factor contributes L, not a fixed eventual weight, and requires a zero target coordinate. Its fiber therefore keeps growing with the modulus. That case must not be mistaken for a finite nonzero Smith weight.

Read this as: The scalar sum of weights gives the count, but the separate weights give the admissibility conditions. Keeping only a determinant valuation can hide which target coordinate fails.

Plate II.20 / MODEL F / A FINITE POLYPHASE REALIZATION

Filter first or split first: same output

Chapter II, mathematical text: Section B.1

Use the finitely supported signal $x = (1, 2, 3, 4, 5, 6)$ and the filter $h = (1, 2, 1)$. Outside the displayed support, the input is zero. Let $e_m = x_{2m}$ and $o_m = x_{2m+1}$.

x	1	2	3	4	5	6		
$h * x$	1	4	8	12	16	20	17	6
y_{even}	1	8	16	17				
y_{odd}	4	12	20	6				

$h = (1, 2, 1)$

zero extension

no periodic wrap

Exact integer convolution, including both output tail coefficients. The two bottom rows are the even and odd output phases. Interleaving them reconstructs the full eight-term output.

If w denotes one delay on a phase sequence, the filter becomes

$$\begin{pmatrix} y_{\text{even}} \\ y_{\text{odd}} \end{pmatrix} = \underbrace{\begin{pmatrix} 1+w & 2w \\ 2 & 1+w \end{pmatrix}}_{H_{\text{phase}}(w)} \begin{pmatrix} e \\ o \end{pmatrix}.$$

The cross-phase delay is not optional. A unit shift sends an odd sample to the next even position, while it sends an even sample to the odd position in the same phase block.

Here $e = 1 + 3w + 5w^2$ and $o = 2 + 4w + 6w^2$, so direct polynomial multiplication gives

$$y_{\text{even}} = 1 + 8w + 16w^2 + 17w^3, \qquad y_{\text{odd}} = 4 + 12w + 20w^2 + 6w^3.$$

This is the same result as ordinary convolution followed by residue extraction. No numerical transform or limiting argument is needed because every sequence is finitely supported.

> **Read this as:** A phase matrix is not just a sparsity pattern. Its delay factors carry the information about quotient and remainder needed when a shift crosses a phase boundary.

Plate II.21 / MODEL G / A NONLINEAR PHASE ORBIT

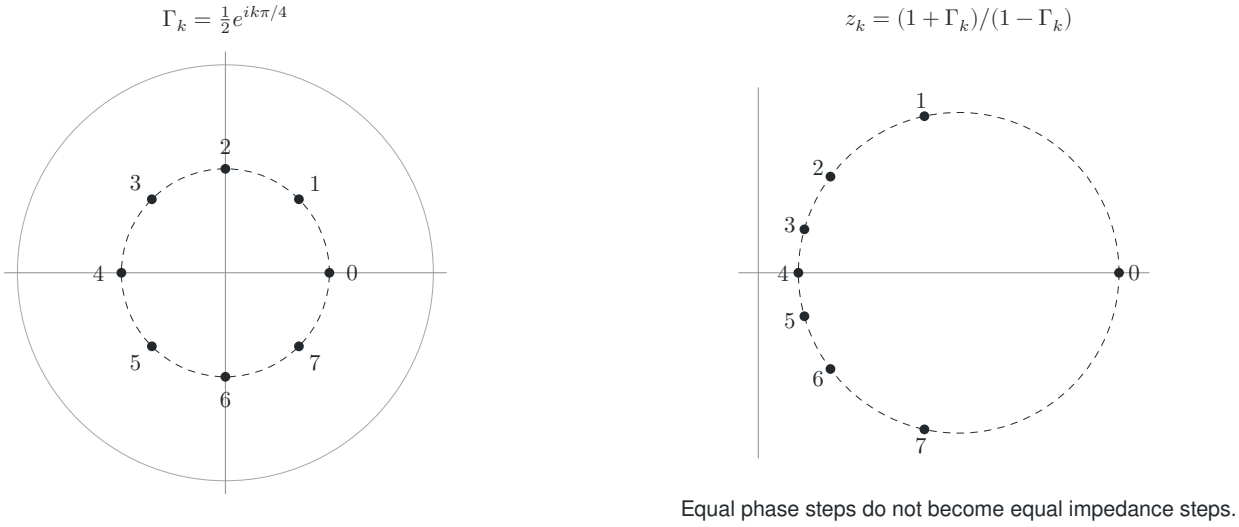
Two Smith constructions, different roles

Chapter II, mathematical text: Section 12.3, Smith-phase identity (p. 43)

Smith *normal form* concerns linear congruences. The Smith *impedance chart* uses the nonlinear coordinate change

$$\Gamma = \frac{z-1}{z+1}, \qquad z = \frac{1+\Gamma}{1-\Gamma}.$$

They share a name, not a mathematical definition. Sample the reflection circle at the exact algebraic points $\Gamma_k = \frac{1}{2}e^{ik\pi/4}$.



Equal phase steps do not become equal impedance steps.

Matching labels identify the same point in the two coordinates. All eight points are computed algebraically. Only their placement on the page is rendered numerically. The outer left circle is $|\Gamma| = 1$, not part of the sampled radius-$1/2$ orbit.

The half-turn $\Gamma \mapsto -\Gamma$ gives

$$z \longmapsto \frac{1}{z}, \qquad z \neq 0, -1.$$

Thus the labels 0 and 4 are 3 and $1/3$. Label 2 is $(3+4i)/5$, and the opposite point is its reciprocal. The domain restrictions come from the forward and inverse coordinate denominators.

These sampled phase points are not linear idempotent ranges. Refining from two to four to eight samples inserts more points on the same orbit. This does not imply a linear packet decomposition of impedance values.

> **Read this as:** The chart is a geometric realization of phase action. Its nonlinear coordinate sensitivity near $\Gamma = \pm 1$ is not the Smith congruence-fiber count and not a new linear projector theorem.

Plate II.22 / MODEL H / EXACT TELESCOPING, EXPLICIT GROWTH

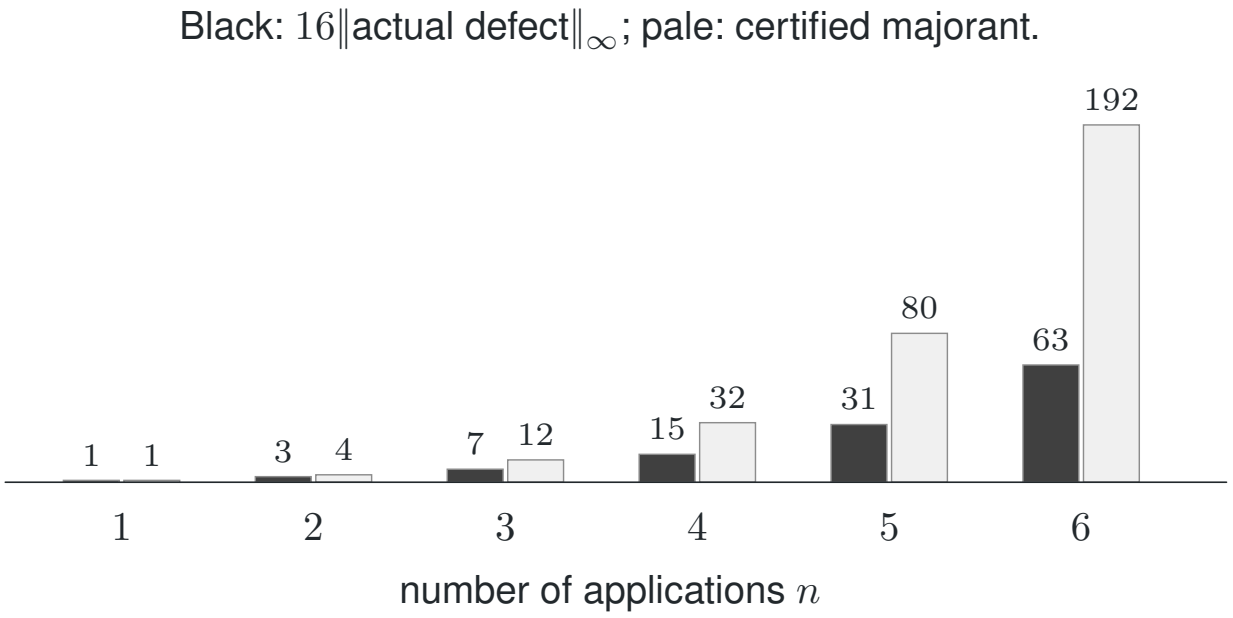
A small defect can grow under powers

Chapter II, mathematical text: Theorem 11.1, Corollary 11.3

Choose an invertible coordinate shear but independently keep the same numerical operator at both levels:

$$C = \begin{pmatrix} 1 & \varepsilon \\ 0 & 1 \end{pmatrix}, \quad T_f = T_c = \operatorname{diag}(1, 2), \quad \varepsilon = \frac{1}{16}.$$

Then $\Delta = CT_f - T_c C = \varepsilon E_{01}$. Exact coordinate invertibility has not forced these independently selected operators to intertwine.



Actual defects and certified upper bounds in the induced infinity norm, both divided by ε. All heights are exact integers. The upper estimate is valid but need not be tight.

Direct multiplication gives

$$CT_f^n - T_c^n C = \varepsilon(2^n - 1)E_{01}.$$

The general telescoping formula explains every contribution:

$$CT_f^n - T_c^n C = \sum_{k=0}^{n-1} T_c^{n-1-k} \Delta T_f^k.$$

Since both operator norms are two, the source theorem bounds the error by $n2^{n-1}\varepsilon$. It does not promise a small error uniformly in n merely because ε is small.

Had we chosen the transported operator $T_c' = CT_f C^{-1}$, the defect would be zero and every polynomial action would intertwine exactly. That is a different choice of coarse operator, not a stronger conclusion from the faulty one.

> **Read this as:** Separate coordinate coherence from operator consistency. A defect from one step is propagated by the dynamics on both sides. The telescoping identity tells you exactly how.

Plate II.23 / MODEL H / A DOMAIN-SENSITIVE COMPARISON

Resolvents amplify errors near poles

Chapter II, mathematical text: Theorem 11.5

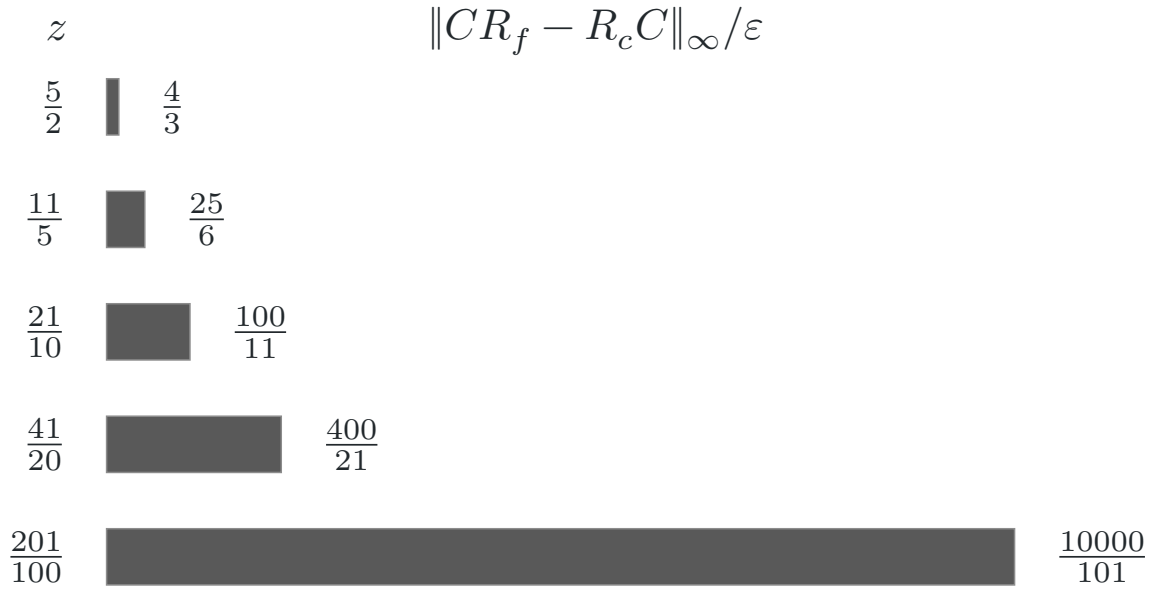
For the same model, take $z \notin \{1, 2\}$ and form the actual inverse matrices

$$R_f = (zI - T_f)^{-1}, \qquad R_c = (zI - T_c)^{-1}.$$

Their off-diagonal intertwining error is

$$CR_f - R_c C = R_c \Delta R_f = \frac{\varepsilon}{(z-1)(z-2)} E_{01}.$$

Both inverse hypotheses matter. Neither resolvent is asserted at a pole.



Exact rational samples; the point $z = 2$ itself is excluded.

The bars show exact amplification factors for five rational choices $z > 2$. Rows are separate samples, not equally spaced positions on the z axis. The coefficient $\varepsilon = 1/16$ has been divided out.

At $z = 3$ the amplification factor is only $1/2$. At $z = 201/100$, it is $10000/101$, so a small one-step defect can create a much larger resolvent defect. This follows from the exact formula rather than from unstable floating-point subtraction of two nearly singular inverses.

The certified general norm estimate is

$$\|CR_f - R_c C\| \le \|R_c\| \, \|\Delta\| \, \|R_f\|.$$

The chapter also proves a version using one-sided witnesses: a left inverse for the coarse shift and a right inverse for the fine shift suffice for the algebraic identity. That does not turn arbitrary one-sided witnesses into a claim of two-sided resolvent existence.

> **Read this as:** A small local defect is meaningful only together with control of the surrounding operators or inverses. Near a pole, the domain and amplification factors are part of the theorem, not technical decoration.

Plate II.24 / PROOF BOUNDARY / HOW TO USE THIS GUIDE

Read the certificates with the pictures

Formal evidence: chapter-wide correspondence and the 25-row foundational contract

The foundational formal contract has twenty-five items: twenty-one numbered definitions and results in the main text, one Smith-phase identity, and three algorithm specifications. The source archive preserves its evidence. The broader constructions have a separate correspondence and replay record. The table below describes the foundational interfaces. The chapter-wide map identifies the broader constructions. The models illustrate the mathematics, and their finite checks have recorded ranges.

Formal coverage. 53 of 53 proof-level source records have kernel checked, semantically accepted scope. No partial or pending proof records are listed in this correspondence. These are source-record counts under their stated hypotheses, not counts of independent discoveries.

Inherited interface	Inputs and checked scope	Boundary of that interface
Exact refinement	Complete projector families and a parent map when formulas relating parents to children are used.	A canonical refinement for every arbitrary pair of decompositions.
Fourier realization	A finite orthogonality table, or field/root hypotheses that provide it.	Orthogonality from multiplicative order alone over every ring.
Weighted inversion	A two-sided inverse of D, a multiplicative filtration, and positive certified gains.	Convergence in every possible complete filtered algebra.
Graph rationality	A finite multigraph with positive weights, together with its actual determinant and adjugate.	Simple-path counts, analytic convergence everywhere, or operator noncancellation.
Spectral separation	Evaluated Bézout identities and explicit lattice-preservation data.	Automatic construction or minimality of every resultant certificate.
Smith congruences	Concrete diagonal data modulo p^L, with $p > 1$.	An automatically constructed general-DVR Smith normal form.
Linear defects	Actual typed maps and the stated inverse or norm witnesses.	Small propagated errors independent of time or distance from a pole.

Reproduce the explanatory calculations

The preserved model source contains **118 named exact checks**. Each finite range is recorded in the file `EXACT_CHECKS.json`. They cover masks, CRT labels, modular inverse products, graph enumeration, rational recurrences, lattice denominators, complete finite fibers, phase identities, and defect formulas. Graphics are generated from those same exact model definitions. Numerical placement of geometric points is only a rendering step.

The computational report records 48 passing Python tests, 118 exact visual checks, 22 byte-matching regenerated vector plates, 13 regenerated PDF/PNG figure pairs, and nine matching numerical summary rows. Each record retains its tested range and source identity. Lean replay and source coverage are documented separately.

Read this as: Use the guide to understand what a theorem says and why its hypotheses matter. Use the theorem map and preserved proof sources to identify the precise formal result. Neither a picture nor a finite test expands that result's scope.

CHAPTER III

Projector Resolved Seed Calculus III

Typed transfer, nonlinear packets, and obstruction theory

The preceding constructions provide the linear setting. This chapter studies nonlinear coordinates and the limits of their extension.

Typed operations extend the linear language to nonlinear coordinates, analytic packet jets and obstruction theory. The analytic supplement develops the continuous packet-jet claims.

IN THIS CHAPTER

Mathematical text 178

Analytic supplement 248

Visual studies 254

Formal companion 1768

Sources and references 2670

Scope

Expanded ordinary proofs and the bounded Lean core have different coverage. The packet Taylor estimates, counting arguments and later extensions retain their stated ordinary-proof status. Arbitrary differentiability is not asserted.

The next chapter develops a cyclic analytic family through coefficient selection, transport and spectral data.

How to use this volume

Begin with typed edge words and exact coordinate transport. A chosen connection introduces defects, holonomy, and packet obstruction witnesses. Part II develops nonlinear packet jets, tree expansions, correction complexes, filtered lifting, elimination, and smooth transport.
Part I develops the finite algebraic core under its declaration-level formal contract. Part II gives the nonlinear and analytic development with ordinary proofs. The accompanying formal development verifies exact nonlinear coordinates, continuous packet jets on actual projector ranges, and the decorated composition and structural support laws in Lean. The integral Taylor remainder and the later analytic and obstruction results retain their ordinary proof status. Formal coverage stays attached to the identified declarations. The visual atlas at the end of this volume provides twenty-four worked plates. The separate Lean guide explains the formal contract and its evidence.

Overview

Finite projector packets turn linear transfer into typed walks and arbitrary maps into exact coordinate representations. Canonical coordinate transport satisfies an exact cocycle law. A chosen connection can fail to compose, producing holonomy and packet obstruction witnesses. Nonlinear packet jets lead to higher composition, convergent rooted tree expansions, graded operations, and nonlinear refinement defects. A specified correction map produces a canonical cokernel obstruction. Relative exact sequences, mapping cones, Hilbert splitting, nonlinear correction, and filtered lifting describe how these obstructions behave. Exact elimination produces Schur formulas, and finite transport reduces gluing to fundamental cycle holonomy. For smooth orthogonal packet families, direct packet rotations converge in operator norm to canonical unitary transport. The finite formal contract and the broader ordinary proof development retain their separate scopes throughout.

The chapter presents the mathematical development together with worked examples, a visual atlas, and a separate guide to the formal proof evidence. The packet norm estimate distinguishes the actual range jet from its ambient extension. The support statements keep external packet profiles and intermediate composition witnesses explicitly typed. The formal companion gives the exact source and scope of each formal verification.

Part I Finite typed transfer and the formal contract

The finite algebraic core supplies the algebraic foundation and a contract for its individual declarations. Here, "roadmap" and "outside the contract" describe the scope of the finite Lean formalization. Ordinary mathematical results in Part II may lie beyond that scope. The finite contract covers thirty result environments. The source contract has 41 mapping rows because it also includes definitions and other interfaces.

The formal companion records the exact Lean and Mathlib dependencies [3, 17], compilation, kernel replay, and axiom checks. References to declarations from Chapters I–II retain their explicitly inherited status. The companion distinguishes the finite algebraic contract from the additional packet-coordinate and analytic interfaces.

1 Purpose and formal status

Chapter I fixed typed seeds, operator grammars, exact projector packets, generated hulls, and reconstruction [13]. Chapter II added coherent refinement coordinates, weighted transfer, arithmetic separation, and linear defect identities [14]. The present chapter begins the next layer: explicit edge words, optional connection data, and obstructions within individual packets.

Canonical coordinate transport and a connection play different mathematical roles. Canonical transport compares two exact descriptions of the *same* ambient object and is forced by analysis and synthesis. A connection chooses transports that may carry extra modeling, discretization, or gauge information. Canonical transport obeys a cocycle theorem. A connection may fail to obey a cocycle, and that failure is meaningful data. Keeping separate notation lets us distinguish the automatic coherence of exact coordinates from the possible holonomy of a chosen connection.

1.1 Statement labels

Every substantial theorem, lemma, corollary, proposition, definition, API, and roadmap item carries one or more of the three labels below. When separate clauses use a Chapter III result and an inherited bridge, the statement carries both applicable labels. Illustrative examples do not require a label.

S1. Lean-backed in Chapter III: the displayed scope is no stronger than the named compiled declaration in the identified Chapter III module.

S2. Inherited Lean-backed: the displayed result is supplied by the stated formal contract of Chapter I or Chapter II.

S3. Roadmap only: the text states a future interface or theorem target and is excluded from the completion denominator.

Boundary 1.1 (Finite theorem denominator). The completion denominator of the finite algebraic contract consists of the thirty result environments in Part I. Of these, 26 resolve to compiled Chapter III declarations and four resolve to the stated Chapters I–II contracts. The authoritative theorem map names the exact declarations used by each statement. The kernel audit checks those targets and their supporting theorems. The namespace gate requires every declaration in the compiled Komorebi.PaperIII

namespace to pass its checks. Roadmap items are excluded from the denominator and remain visibly marked.

2 Finite typed packet graphs

2.1 Orientation and matrix convention

Let V be a finite packet type, E a finite edge type, and R a semiring. A directed multigraph is a pair of endpoint functions

$$\mathrm{src}, \mathrm{tgt} : E \longrightarrow V.$$

Parallel edges remain distinct elements of E. Rows are targets and columns are sources. Thus an edge $e : i \to j$ is represented by a matrix supported at row j, column i.

Definition 2.1 (Packet graph and scalar edge system)**.** For $a, b \in V$ and $r \in R$, define

$$\mathrm{packetMatrix}(a, b; r)_{yx} = \begin{cases} r, & y = b \text{ and } x = a, \\ 0, & \text{otherwise.} \end{cases}$$

An edge system is a coefficient function $c : E \to R$. Its elementary edge operator and total transfer operator are

$$L_e = \mathrm{packetMatrix}(\mathrm{src}(e), \mathrm{tgt}(e); c(e)), \qquad T = \sum_{e \in E} L_e.$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.EdgeOperators.PacketGraph;Komorebi.PaperIII.EdgeOperators.PacketGraph.packetMatrix;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp`

Words are written in execution order. In $L_f L_e$, the rightmost operator L_e acts first. For a written word $(e_1, \dots, e_n)$, e_1 acts first and the operator product is $L_{e_n} \cdots L_{e_1}$.

Reading the execution order

Read the edge list in the order of execution. The matrix product writes later actions farther to the left. The target packet of one edge must equal the source packet of the next before multiplication can contribute. Over a noncommutative semiring, this order also fixes the order of the scalar coefficients.

Lemma 2.2 (Declared support)**.** *For every edge e, $(L_e)_{yx} = 0$ unless $y = \mathrm{tgt}(e)$ and $x = \mathrm{src}(e)$.*

Formal contract: finite algebraic core.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_eq_zero_off_support`

Theorem 2.3 (Typed interface annihilation)**.** *If $\mathrm{tgt}(e) \neq \mathrm{src}(f)$, then*

$$L_f L_e = 0.$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_mul_edgeOp_eq_zero`

Proof. For a matrix entry (y, x),

$$(L_f L_e)_{yx} = \sum_{z \in V} (L_f)_{yz} (L_e)_{zx}.$$

A potentially nonzero summand would require simultaneously $z = \mathrm{src}(f)$ and $z = \mathrm{tgt}(e)$, contrary to the hypothesis. □

Theorem 2.4 (Collapse of a matched edge pair). *If* $\mathrm{tgt}(e) = \mathrm{src}(f)$, *then*

$$L_f L_e = \mathrm{packetMatrix}(\mathrm{src}(e), \mathrm{tgt}(f); c(f)c(e)).$$

Together with theorem 2.3, this gives both cases for the composition of two edge operators.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_mul_edgeOp_of_composable`

Proof. The sum defining the matrix product has exactly one potentially nonzero intermediate index, namely $\mathrm{tgt}(e) = \mathrm{src}(f)$. At that index the coefficients occur in operator order as $c(f)c(e)$, and the outer source and target are $\mathrm{src}(e)$ and $\mathrm{tgt}(f)$. □

Boundary 2.5 (Current fiber scope). The compiled Chapter III model above has one scalar coordinate per packet. For a family of modules $(X_i)_{i \in V}$, an intrinsic edge map would instead have type $\ell_e : X_{\mathrm{src}(e)} \to X_{\mathrm{tgt}(e)}$, and its totalization on $\bigoplus_i X_i$ would be $\iota_{\mathrm{tgt}(e)} \ell_e \pi_{\mathrm{src}(e)}$. The same support proof applies, but this generalization with dependent fibers is not attributed to the compiled Chapter III modules for scalar edge systems.

Figure 1 summarizes the support convention and shows how composable words differ from words with a broken interface. The next section uses this distinction.

3 Operator words and directed walks

3.1 Typed words

Definition 3.1 (Composable word and execution operator). A finite word $w = (e_1, \dots, e_n)$ is composable when

$$\mathrm{tgt}(e_r) = \mathrm{src}(e_{r+1}) \qquad (1 \le r < n).$$

Set

$$L_w = L_{e_n} \cdots L_{e_1}, \qquad L_\emptyset = I.$$

This definition records the written order of execution even though matrix composition is read right to left.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Composable;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp`

Theorem 3.2 (First broken interface). *If a word begins* $(e, f, \dots)$ *and* $\mathrm{tgt}(e) \neq \mathrm{src}(f)$, *then its word operator is zero.*

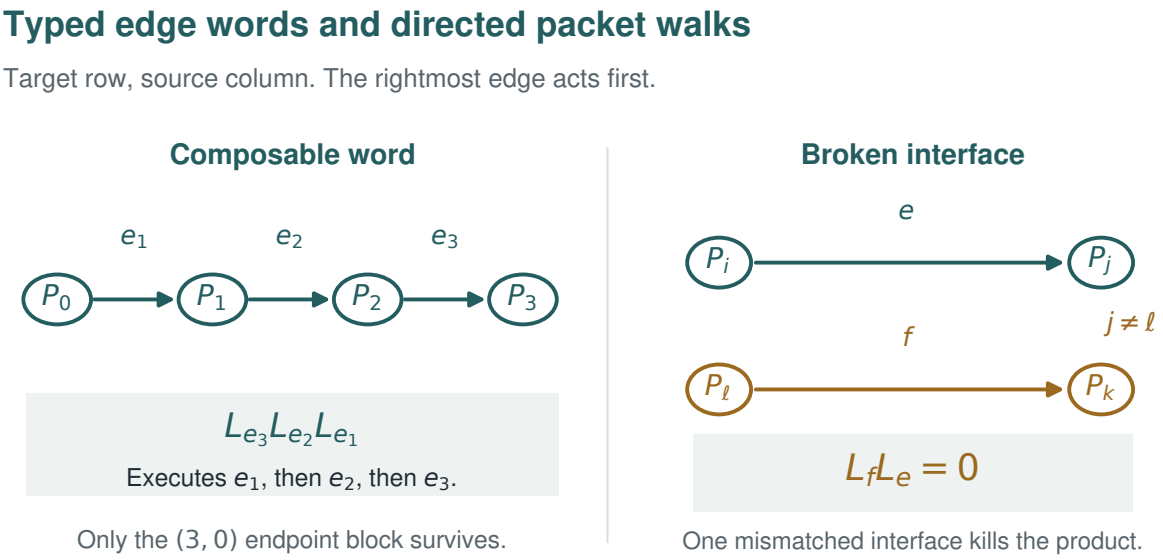


Figure 1. Rows record target packets and columns record source packets. This convention fixes execution order. Composable words collapse to endpoint blocks, while one broken packet interface annihilates the entire operator product.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_zero_of_first_break`

Theorem 3.3 (Noncomposable words vanish)**.** *If w is not composable, then $L_w = 0$.*

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_zero_of_not_composable`

Proof. Choose the first broken adjacent pair. The corresponding consecutive factor is zero by theorem 2.3, and associativity absorbs that zero into the complete product. The Lean proof implements the same argument by induction on the list. □

3.2 The checked recursive expansion

Definition 3.4 (Recursive word expansion)**.** Define

$$\mathcal{W}_0 = I, \qquad \mathcal{W}_{n+1} = T\mathcal{W}_n.$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordExpansion`

Theorem 3.5 (Transfer power equals recursive word expansion). *For every $n \in \mathbb{N}$,*

$$T^n = \mathcal{W}_n.$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_wordExpansion`

Proof. The zero case is the identity matrix. If $T^n = \mathcal{W}_n$, then the left-successor power convention gives $T^{n+1} = TT^n = T\mathcal{W}_n = \mathcal{W}_{n+1}$. □

Remark 3.6. The formal module also defines an explicit sum over tuples $w : \mathrm{Fin}(n) \to E$ and a second sum in which noncomposable tuples are replaced by zero. It proves that the recursive matrix, the explicit tuple sum, and the sum filtered by composability agree. The separate declarations make the finite enumeration claim directly checkable.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.explicitWordExpansion;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.walkExpansion;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordExpansion_eq_explicitWordExpansion;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.explicitWordExpansion_eq_walkExpansion`

3.3 Exact expansion from words to walks

For a nonempty composable word $w = (e_1, \dots, e_n)$, define its reverse coefficient product

$$c^{\leftarrow}(w) = c(e_n)c(e_{n-1}) \cdots c(e_1).$$

The reversal is essential over a noncommutative semiring.

Lemma 3.7 (One word is one endpoint block). *If $w = (e_1, \dots, e_n)$ is nonempty and composable, then*

$$L_w = \mathrm{packetMatrix}(\mathrm{src}(e_1), \mathrm{tgt}(e_n); c^{\leftarrow}(w)).$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.reverseCoefficient;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.terminalTarget;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_packetMatrix_of_composable`

Proof. Induct on n. The case $n = 1$ is the definition of L_e. Suppose the formula holds for the prefix $(e_1, \dots, e_n)$ and append f. Composability gives $\mathrm{tgt}(e_n) = \mathrm{src}(f)$. Multiplying the two packet matrices leaves one intermediate index and changes the coefficient to $c(f)c(e_n) \cdots c(e_1)$, with source $\mathrm{src}(e_1)$ and target $\mathrm{tgt}(f)$. □

Write $\mathrm{Walk}_n(i, j)$ for the finite set of composable edge tuples of length n whose first source is i and whose final target is j. For $n = 0$, it has one empty walk when $i = j$ and is empty otherwise. Assign the empty walk the coefficient 1.

The empty walk

The empty walk records doing nothing at a chosen packet. It contributes only to a diagonal entry. Thus the convention for length zero matches the identity matrix and lets the walk sum include every nonnegative power.

Theorem 3.8 (Finite operator expansion from words to walks). *For every $n \in \mathbb{N}$,*

$$\boxed{T^n = \sum_{w \in E^n} L_w = \sum_{w \in E^n} \begin{cases} L_w, & w \text{ composable}, \\ 0, & w \text{ not composable}. \end{cases}}$$

Parallel edges occur as distinct summands.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_explicitWordExpansion;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_walkExpansion`

Proof. Induction and finite distributivity give

$$\left(\sum_{e \in E} L_e\right)^n = \sum_{(e_1,\dots,e_n) \in E^n} L_{e_n} \cdots L_{e_1}.$$

Every noncomposable summand is zero by theorem 3.3, so inserting the composability filter does not change the sum. In Lean, tuple concatenation and the successor sum are handled by `tupleWordOp_snoc` and `explicitWordExpansion_succ`. The two bridge equalities then yield the displayed theorem. □

Corollary 3.9 (Endpoint coefficient walk sum). *For all $n \in \mathbb{N}$ and $i, j \in V$,*

$$\boxed{(T^n)_{ji} = \sum_{w \in \mathrm{Walk}_n(i,j)} c^{\leftarrow}(w).}$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Connects;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.endpointWalkCoefficientSum;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_apply_eq_sum_walks`

Proof. Take the (j, i) entry in theorem 3.8. By lemma 3.7, a surviving word contributes $c^{\leftarrow}(w)$ exactly when its endpoints are i and j, and contributes zero otherwise. At $n = 0$, the identity matrix gives the unique empty diagonal walk. □

Corollary 3.10 (Reachability obstruction at fixed length). *If* $\mathrm{Walk}_n(i, j) = \emptyset$, *then* $(T^n)_{ji} = 0$.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_apply_eq_zero_of_no_walk`

Proof. This is immediate from the empty sum in the preceding corollary. □

Corollary 3.11 (Orbit support obstruction). *Let $S \subseteq V$, and let $x \in R^V$ be supported on S. If j is not reachable by a directed path from any vertex of S, then*

$$(T^n x)_j = 0 \qquad \text{for every } n \in \mathbb{N}.$$

Hence the orbit under T of x cannot generate a vector with a nonzero coordinate at j.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Reachable;Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_mulVec_apply_eq_zero_of_unreachable_support`

Inherited formal contract: Chapters I–II.
`Komorebi.PacketNaturalityCompletion.Reachability.algebraicOrbitHull_le_reachable_support;Komorebi.PacketNaturalityCompletion.Reachability.unreachable_nonzero_packet_obstruction`

Proof. Expand $(T^n x)_j = \sum_i (T^n)_{ji} x_i$. Terms with $i \notin S$ vanish by the support of x, and terms with $i \in S$ vanish by corollary 3.10. The final claim follows because the j-coordinate also vanishes on every finite linear combination of orbit vectors. □

4 Canonical exact coordinate transport

4.1 The Chapter III frame model

Let every chart a carry a linear frame equivalence $F_a : M \xrightarrow{\sim} M$. The formal Chapter III module defines transport from a to b by

$$J_{b \leftarrow a} = F_b^{-1} F_a.$$

This frame formula means *synthesize in chart a, then analyze in chart b.*

Theorem 4.1 (Exact frame transport is coherent). *For every a, b, c,*

$$J_{a \leftarrow a} = \mathrm{id}, \qquad J_{a \leftarrow b} J_{b \leftarrow a} = \mathrm{id}, \qquad J_{c \leftarrow b} J_{b \leftarrow a} = J_{c \leftarrow a},$$

and therefore

$$J_{a \leftarrow c} J_{c \leftarrow b} J_{b \leftarrow a} = \mathrm{id}.$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.exactTransport_self;Komorebi.PaperIII.Transport.exactTransport_roundTrip;Komorebi.PaperIII.Transport.exactTransport_cocycle;Komorebi.PaperIII.Transport.exactTransport_triangleLoop`

Proof. Every identity is cancellation of adjacent $F_r F_r^{-1}$ factors. The Lean definitions use `LinearEquiv.trans`. The displayed formulas use the equivalent right-to-left notation for function composition fixed in this chapter. □

Theorem 4.2 (Every finite canonical path telescopes). *Let $p = (a_0, a_1, \dots, a_n)$ be any finite vertex list and put*

$$J_p = J_{a_n \leftarrow a_{n-1}} \cdots J_{a_1 \leftarrow a_0}, \qquad J_{(a_0)} = \mathrm{id}.$$

Then

$$J_p = J_{a_n \leftarrow a_0}.$$

In particular, if $a_n = a_0$, *then* $J_p = \mathrm{id}$.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.pathEndpoint;Komorebi.PaperIII.Transport.exactPathTransport;Komorebi.PaperIII.Transport.exactPathTransport_eq_endpoint;Komorebi.PaperIII.Transport.exactPathTransport_loop`

Proof. Induct on the list of successive vertices and apply the cocycle in theorem 4.1 at each step. If the endpoint is the start, the endpoint transport is the identity. □

4.2 Exact packet coordinates

The graph model above resolves a fixed scalar matrix. Coordinate transport is a separate construction. Let X be an R-module, and let

$$\mathcal{P}^a = (P_i^a)_{i \in I_a}$$

be a finite complete family of pairwise orthogonal linear idempotents:

$$P_i^a P_j^a = \delta_{ij} P_i^a, \qquad \sum_{i \in I_a} P_i^a = I_X.$$

Define the exact packet coordinate module

$$\mathrm{Coord}(\mathcal{P}^a) = \left\{ (x_i)_{i \in I_a} : P_i^a x_i = x_i \right\}.$$

Its analysis and synthesis maps are

$$\mathcal{A}_a(x) = (P_i^a x)_i, \qquad \mathcal{S}_a((x_i)_i) = \sum_i x_i.$$

Completeness and orthogonality give

$$\mathcal{S}_a \mathcal{A}_a = \mathrm{id}_X, \qquad \mathcal{A}_a \mathcal{S}_a = \mathrm{id}_{\mathrm{Coord}(\mathcal{P}^a)}.$$

What a packet coordinate contains

A packet coordinate is a whole vector fixed by its projector. Analysis splits an ambient vector into these components, and synthesis adds them back. The transport between exact systems keeps the ambient vector and changes its packet presentation.

Definition 4.3 (Canonical exact coordinate transport)**.** For two exact packet systems on the same ambient module, define

$$J^X_{b \leftarrow a} = \mathcal{A}_b \mathcal{S}_a : \mathrm{Coord}(\mathcal{P}^a) \longrightarrow \mathrm{Coord}(\mathcal{P}^b).$$

The arrow points from the source coordinate system a to the target coordinate system b.

Inherited formal contract: Chapters I–II.

`Komorebi.PaperII.NestedRefinement.refineCoordinates;Komorebi.PaperII.NestedRefinement.coordinateRefinementEquiv`

Theorem 4.4 (Identity, inverse, and cocycle for canonical transport)**.** *For any exact packet systems a, b, c on X,*

$$J^X_{a\leftarrow a} = \mathrm{id}, \qquad \left(J^X_{b\leftarrow a}\right)^{-1} = J^X_{a\leftarrow b}, \qquad J^X_{c\leftarrow b} J^X_{b\leftarrow a} = J^X_{c\leftarrow a}.$$

Inherited formal contract: Chapters I–II.

`Komorebi.PaperII.NestedRefinement.refineCoordinates_self;Komorebi.PaperII.NestedRefinement.coarsenCoordinates_refineCoordinates;Komorebi.PaperII.NestedRefinement.refineCoordinates_coarsenCoordinates;Komorebi.PaperII.NestedRefinement.refineCoordinates_comp;Komorebi.PaperII.NestedRefinement.coordinateRefinementEquiv_trans`

Proof. Insert $\mathcal{S}_b\mathcal{A}_b = \mathrm{id}_X$:

$$J^X_{c\leftarrow b} J^X_{b\leftarrow a} = \mathcal{A}_c \mathcal{S}_b \mathcal{A}_b \mathcal{S}_a = \mathcal{A}_c \mathcal{S}_a = J^X_{c\leftarrow a}.$$

The identity statement is $\mathcal{A}_a\mathcal{S}_a = \mathrm{id}$. Setting $c = a$ in the cocycle gives $J^X_{a\leftarrow b} J^X_{b\leftarrow a} = \mathrm{id}$. Interchanging a and b gives the other composition, so the two transports are inverses. □

If $F : X \to Y$ is linear and $\mathcal{P}^a, \mathcal{P}^b$ are source packet systems while $\mathcal{Q}^r, \mathcal{Q}^s$ are target packet systems, put

$$\widehat{F}_{r\leftarrow a} = \mathcal{A}^Y_r F \mathcal{S}^X_a.$$

Theorem 4.5 (Canonical coordinate naturality)**.** *The exact coordinate representation preserves typed composition and satisfies*

$$J^Y_{s\leftarrow r} \widehat{F}_{r\leftarrow a} = \widehat{F}_{s\leftarrow b} J^X_{b\leftarrow a}.$$

Inherited formal contract: Chapters I–II.

`Komorebi.PaperII.NestedRefinement.coordinateOperator_comp;Komorebi.PaperII.NestedRefinement.coordinateOperator_naturality`

Proof. Both sides reduce to $\mathcal{A}^Y_s F \mathcal{S}^X_a$ after inserting the appropriate synthesis–analysis identity. The composition law is the same calculation with the middle ambient module inserted. □

Corollary 4.6 (Canonical triangle holonomy is trivial)**.** *For every triple a, b, c,*

$$J^X_{a\leftarrow c} J^X_{c\leftarrow b} J^X_{b\leftarrow a} = \mathrm{id}.$$

Inherited formal contract: Chapters I–II.

`Komorebi.PaperII.NestedRefinement.refineCoordinates_comp;Komorebi.PaperII.NestedRefinement.refineCoordinates_self`

Boundary 4.7 (Refinement is not reduction). Although Chapter II uses the name *refineCoordinates*, J is an equivalence between two complete coordinate presentations. It neither truncates packets nor projects to a smaller state space. Any lossy restriction, interpolation, or learned identification is extra data and cannot inherit theorem 4.4 without proof.

5 A distinct noncanonical connection API

5.1 The minimal compiled API

The compiled Chapter III API assumes only normalization at each chart. All chart transports act on one module M, so chart labels distinguish routes but not Lean types.

Definition 5.1 (Noncanonical connection normalized at each chart)**.** A minimal connection is a family

$$\Gamma_{b\leftarrow a} \in \mathrm{Aut}_R(M) \qquad (a, b \in B)$$

with only

$$\Gamma_{a\leftarrow a} = \mathrm{id}_M.$$

Neither inversion on reversed edges nor the cocycle law is a field of the structure.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.Connection`

At a triple (a, b, c), the compiled cocycle predicate is

$$\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a} = \Gamma_{c\leftarrow a}.$$

Flatness means this equality holds for every triple. The formal multiplicative relative defect and triangle holonomy are

$$D^{\times}_{c,b,a} = (\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a})\Gamma^{-1}_{c\leftarrow a}, \qquad H_{c,b,a} = \Gamma_{a\leftarrow c}\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a}.$$

The defect is based at the target c. The holonomy is based at a.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.Connection.CocycleAt;Komorebi.PaperIII.Transport.Connection.Flat;Komorebi.PaperIII.Transport.Connection.cocycleDefect;Komorebi.PaperIII.Transport.Connection.triangleHolonomy`

Theorem 5.2 (Checked flatness consequences)**.** *For the minimal connection API, flatness implies*

$$D^{\times}_{c,b,a} = \mathrm{id}_M, \qquad H_{c,b,a} = \mathrm{id}_M$$

for every a, b, c.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.Connection.cocycleDefect_eq_refl_of_flat;Komorebi.PaperIII.Transport.Connection.triangleHolonomy_eq_refl_of_flat`

Proof. Flatness replaces the two-edge route in $D^{\times}$ by the direct route, which cancels its inverse. For H, flatness first collapses $a \to b \to c$ to $a \to c$, then collapses $a \to c \to a$ to the normalized transport $a \to a$. □

Proposition 5.3 (The minimal API admits nontrivial holonomy)**.** *There is a connection on three points over $\mathbb{Q}$ for which transport $0 \to 1$ is multiplication by 2, every other transport between distinct endpoints is the identity, and*

$$H_{2,1,0}(1) = 2.$$

In particular $H_{2,1,0} \neq \mathrm{id}$.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.twistedThreePointConnection;Komorebi.PaperIII.Transport.twisted_triangleHolonomy_apply_one;Komorebi.PaperIII.Transport.twisted_triangleHolonomy_ne_refl`

Boundary 5.4 (Why the minimal defect and loop are distinct). Without reverse-edge inversion, $\Gamma_{a\leftarrow c}$ need not equal $\Gamma^{-1}_{c\leftarrow a}$. Therefore the multiplicative relative defect $D^{\times}_{c,b,a}$ and triangle holonomy $H_{c,b,a}$ are different comparisons. The formal module proves that global flatness trivializes both. It does not identify them for an arbitrary connection.

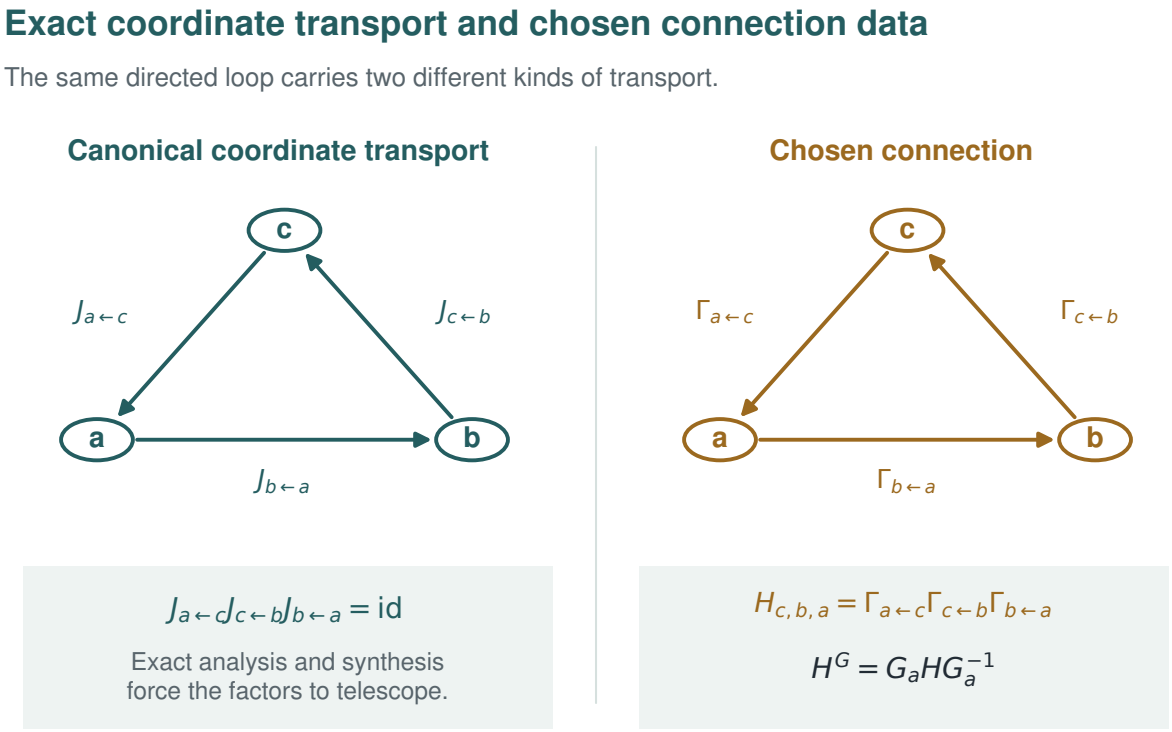


Figure 2. Exact coordinate transport and connection transport are distinct. The canonical loop telescopes to the identity. A chosen connection may carry nontrivial holonomy, which transforms by conjugation under gauge.

Figure 2 records this separation visually: the exact coordinate route telescopes, whereas the chosen connection route may carry a nonidentity loop.

Coordinate transport and connection choice

Canonical transport compares exact coordinate descriptions of the same object. Its loops telescope. A connection supplies a separate family of comparison maps, so its loops can carry information about that choice. The distinction keeps coordinate identities separate from connection assumptions.

5.2 A stronger reversible pair API

Let R be a commutative ring, let B be a type of charts or refinement states, and let each C_a be an R-module. The chart type is not assumed finite. The following definition is deliberately not derived from analysis and synthesis.

Definition 5.5 (Normalized reversible connection)**.** A normalized reversible connection is a family of linear equivalences

$$\Gamma_{b\leftarrow a} : C_a \xrightarrow{\sim} C_b \qquad (a, b \in B)$$

such that

$$\Gamma_{a\leftarrow a} = \mathrm{id}, \qquad \Gamma_{a\leftarrow b} = \Gamma_{b\leftarrow a}^{-1}.$$

No cocycle equation is included in the structure.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.ReversibleConnection`

This adds heterogeneous fibers and inversion on reversed edges to the homogeneous minimal connection. The connection is still a separate choice of maps, rather than a definition of canonical coordinate transport.

Definition 5.6 (Calibration and flatness)**.** When the C_a are exact packet coordinates of one ambient module, the connection is *canonically calibrated* if

$$\Gamma_{b\leftarrow a} = J_{b\leftarrow a} \quad \text{for every } a, b.$$

It is *flat* if

$$\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a} = \Gamma_{c\leftarrow a} \quad \text{for every } a, b, c.$$

The canonical connection is flat. A general flat connection is proved in theorem 6.3 to be related to it by a gauge change.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.ReversibleConnection.CocycleAt;Komorebi.PaperIII.Transport.ReversibleConnection.Flat;Komorebi.PaperIII.Transport.ReversibleConnection.canonicalConnection;Komorebi.PaperIII.Transport.ReversibleConnection.canonicalConnection_flat`

The words *calibrated* and *flat* have only the algebraic meanings displayed above.

Remark 5.7 (A connection on a chart graph). One may specify Γ_e only on the oriented edges of a chart graph, together with inverse maps on reversed edges. Path transport and loop holonomy below still make sense. A direct cocycle defect for (a, b, c), however, requires a declared direct transport from a to c. The pairwise API in definition 5.5 keeps this comparison total.

5.3 Cocycle defect and triangle holonomy

The commutative ring hypothesis makes subtraction of linear maps available from this point through section 7.

Definition 5.8 (Cocycle defect)**.** For $a, b, c \in B$, define the additive cocycle defect

$$\Delta_{c,b,a} = \Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a} - \Gamma_{c\leftarrow a} \in \mathrm{Hom}_R(C_a, C_c).$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.cocycleDefect`

The manuscript subscript order is target, intermediate, source. The Lean arguments are supplied in traversal order `(a,b,c)`.

Definition 5.9 (Triangle holonomy)**.** The triangle holonomy based at a is

$$H_{c,b,a} = \Gamma_{a\leftarrow c}\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a} \in \mathrm{Aut}_R(C_a).$$

It traverses $a \to b \to c \to a$.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleHolonomy;Komorebi.PaperIII.Transport.ReversibleConnection.triangleObstruction`

Holonomy is attached to the chosen connection Γ, never to an unspecified coordinate change.

Theorem 5.10 (Identity between defect and holonomy)**.** *For every triple* (a, b, c)*,*

$$\boxed{H_{c,b,a} - \mathrm{id}_{C_a} = \Gamma_{a\leftarrow c}\Delta_{c,b,a}.}$$

Consequently,

$$\Delta_{c,b,a} = 0 \quad \Longleftrightarrow \quad H_{c,b,a} = \mathrm{id}_{C_a}.$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleHolonomy_sub_id;Komorebi.PaperIII.Transport.ReversibleConnection.cocycleDefect_eq_zero_iff_triangleHolonomy_eq_refl`

Proof. Using reversibility,

$$\begin{aligned}\Gamma_{a\leftarrow c}\Delta_{c,b,a} &= \Gamma_{a\leftarrow c}\Gamma_{c\leftarrow b}\Gamma_{b\leftarrow a} - \Gamma_{a\leftarrow c}\Gamma_{c\leftarrow a}\\ &= H_{c,b,a} - \mathrm{id}_{C_a}.\end{aligned}$$

Left composition by the equivalence $\Gamma_{a\leftarrow c}$ is injective, which gives the equivalence of the two vanishing statements. □

Corollary 5.11 (Canonical transport has zero connection defect)**.** *If* $\Gamma_{b\leftarrow a} = J_{b\leftarrow a}$*, then*

$$\Delta_{c,b,a} = 0, \qquad H_{c,b,a} = \mathrm{id}$$

for all triples.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalTransport_zero_connectionDefect;Komorebi.PaperIII.Transport.ReversibleConnection.canonicalTransport_triangleHolonomy_eq_refl`

Proof. The defect vanishes by the canonical cocycle in theorem 4.4. Apply theorem 5.10. □

5.4 Paths and endpoint independence

Definition 5.12 (Connection transport along a path)**.** For a vertex sequence $p = (a_0, a_1, \dots, a_n)$, define

$$\Gamma_p = \Gamma_{a_n \leftarrow a_{n-1}} \cdots \Gamma_{a_1 \leftarrow a_0}, \qquad \Gamma_{(a_0)} = \mathrm{id}_{C_{a_0}}.$$

If $a_n = a_0$, then Γ_p is the holonomy of the loop p.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.pathTransport;Komorebi.PaperIII.Transport.ReversibleConnection.VertexPath;Komorebi.PaperIII.Transport.ReversibleConnection.indexedPathTransport`

Products follow path order: the first traversed edge acts on the right.

Proposition 5.13 (Flat pair transport collapses to endpoints)**.** *If Γ is flat, then every path satisfies*

$$\Gamma_p = \Gamma_{a_n \leftarrow a_0}.$$

In particular, every loop has identity holonomy.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.pathTransport_eq_direct_of_flat;Komorebi.PaperIII.Transport.ReversibleConnection.indexedPathTransport_eq_direct_of_flat;Komorebi.PaperIII.Transport.ReversibleConnection.indexedLoopTransport_eq_refl_of_flat`

Proof. Induct on the path length. Appending a_{n+1} left-multiplies the inductive expression by $\Gamma_{a_{n+1} \leftarrow a_n}$, and flatness replaces the two factors by $\Gamma_{a_{n+1} \leftarrow a_0}$. For a loop, normalization gives $\Gamma_{a_0 \leftarrow a_0} = \mathrm{id}$. □

Remark 5.14. The converse *trivial holonomy implies flatness* depends on which loops are present in an API defined on a chart graph. In the complete pair API, triviality of every triangle holonomy is equivalent to flatness by theorem 5.10. No theorem about topological simple connectivity is being invoked.

6 Gauge covariance and comparison with canonical transport

Definition 6.1 (Gauge change)**.** A gauge is a family of automorphisms $G_a \in \mathrm{Aut}_R(C_a)$. It acts on a connection by

$$\Gamma^G_{b \leftarrow a} = G_b \Gamma_{b \leftarrow a} G_a^{-1}.$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.ReversibleConnection.Gauge;Komorebi.PaperIII.Transport.ReversibleConnection.gauge`

A gauge changes coordinate representatives at each chart. The chart graph stays the same.

Theorem 6.2 (Gauge covariance)**.** *The defect and triangle holonomy transform as*

$$\Delta^G_{c,b,a} = G_c \Delta_{c,b,a} G_a^{-1}, \qquad H^G_{c,b,a} = G_a H_{c,b,a} G_a^{-1}.$$

Thus flatness and triviality of triangle holonomy are gauge invariant.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_cocycleDefect;Komorebi.PaperIII.Transport.ReversibleConnection.gauge_triangleHolonomy;Komorebi.PaperIII.Transport.ReversibleConnection.gauge_flat_iff;Komorebi.PaperIII.Transport.ReversibleConnection.gauge_triangleHolonomy_eq_refl_iff`

Proof. Substitute the definition of Γ^G. Adjacent factors $G_b^{-1}G_b$ and $G_c^{-1}G_c$ cancel. In the defect, both the composite and direct terms acquire the same outer factors G_c and G_a^{-1}. In the loop, all inner factors cancel and only conjugation at the base remains. □

Theorem 6.3 (Flat connections are gauges of the canonical pair transport)**.** *Suppose the C_a are exact packet coordinate modules for one ambient module, so the canonical $J_{b\leftarrow a}$ is available. If Γ is a flat normalized reversible pair connection, choose $o \in B$ and set*

$$G_a = \Gamma_{a\leftarrow o} J_{o\leftarrow a}.$$

Then every pair transport satisfies

$$\boxed{\Gamma_{b\leftarrow a} = G_b J_{b\leftarrow a} G_a^{-1}.}$$

Formal contract: finite algebraic core.
`Komorebi.PaperIII.Transport.ReversibleConnection.flatGauge;Komorebi.PaperIII.Transport.ReversibleConnection.flat_isGaugeOfCanonical`

Proof. Reversibility gives $G_a^{-1} = J_{a\leftarrow o}\Gamma_{o\leftarrow a}$. Therefore

$$\begin{aligned} G_b J_{b\leftarrow a} G_a^{-1} &= \Gamma_{b\leftarrow o} J_{o\leftarrow b} J_{b\leftarrow a} J_{a\leftarrow o} \Gamma_{o\leftarrow a} \\ &= \Gamma_{b\leftarrow o}\Gamma_{o\leftarrow a} \\ &= \Gamma_{b\leftarrow a}. \end{aligned}$$

The first reduction uses the canonical cocycle twice. The second uses flatness of Γ. □

Boundary 6.4 (Flat does not mean calibrated). The theorem produces a gauge relating Γ to J. It does not prove $\Gamma = J$ before that gauge is chosen. Conversely, nontrivial triangle holonomy cannot be removed by a gauge, because conjugation cannot turn a nonidentity automorphism into the identity.

7 Obstruction packets: local certificates and holonomy blocks

7.1 The generic compiled certificate

Let Candidate, Packet, and Residual be arbitrary types, with a distinguished $0 \in$ Residual, and let

$$r : \mathsf{Candidate} \longrightarrow \mathsf{Packet} \longrightarrow \mathsf{Residual}$$

be a residual function.

Definition 7.1 (Solution and obstruction packet for a candidate)**.** A candidate x is a solution when

$$\mathrm{IsSolution}_r(x) \quad \Longleftrightarrow \quad \forall p,\ r(x,p) = 0.$$

An obstruction packet for x is a pair consisting of a named packet p and a certificate $r(x, p) \neq 0$.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.NonlinearObstructions.IsSolution;Komorebi.PaperIII.NonlinearObstructions.ObstructionPacket`

Theorem 7.2 (Exact logical force of a local obstruction)**.** *Every obstruction packet for x proves that x is not a solution, and*

$$\neg\,\mathrm{IsSolution}_r(x) \quad \Longleftrightarrow \quad \textit{there exists an obstruction packet for } x.$$

The reverse implication from failure to a packet witness is classical and does not require the packet type to be finite.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.NonlinearObstructions.ObstructionPacket.not_isSolution;Komorebi.PaperIII.NonlinearObstructions.obstructionPacket_of_not_isSolution;Komorebi.PaperIII.NonlinearObstructions.not_isSolution_iff_nonempty_obstructionPacket`

Proof. A purported solution would make the residual vanish at the certified packet, contradicting its nonzero field. Conversely, if no packet carries a nonzero residual, then every residual is zero and the candidate satisfies the definition of solution. □

What one obstruction certifies

The certificate answers one question: does this specified candidate satisfy every declared packet condition? One nonzero residual settles that question for this candidate. A statement that no candidate can solve the problem would require an obstruction for every candidate.

Boundary 7.3 (Candidate-specific means candidate-specific). The theorem rules out the named x. It neither constructs a solution nor shows that every candidate is obstructed. It provides no minimality, stability, computability, or classification theorem for the packet witness.

7.2 Specialization to holonomy blocks

For a reversible connection, specialize the generic residual at a fixed triangle by taking a packet label to be a pair (μ, ν) and the residual to be the corresponding block of $H - \mathrm{id}$. A nonzero block, together with its label and a proof of nonvanishing, gives an obstruction packet certificate in the sense of definition 7.1. The reconstruction theorem below supplies the further step: vanishing of all these local residuals is equivalent to the cocycle equation at that triangle.

Fix a triangle (a, b, c) and resolve the base fiber C_a by a finite complete family of idempotents $(\Pi^a_\mu)_{\mu \in I_a}$. Orthogonality may also hold, as it does for the exact range coordinate projectors below, but the compiled reconstruction theorem does not require it. In exact range coordinates, Π^a_μ is simply the projector that retains component μ and zeros the other components.

Definition 7.4 (Holonomy obstruction and packet residuals)**.** Define

$$\mathrm{Ob}_{c,b,a} = H_{c,b,a} - \mathrm{id}_{C_a}, \qquad \mathrm{Ob}^{\mu\nu}_{c,b,a} = \Pi^a_\mu \,\mathrm{Ob}_{c,b,a}\, \Pi^a_\nu.$$

The map $\mathrm{Ob}^{\mu\nu}_{c,b,a}$ is the obstruction block, or packet residual, from input packet ν to output packet μ. If it is nonzero, then (μ, ν) and that nonvanishing proof form an obstruction packet certificate. Diagonal blocks detect holonomy within one packet. Off-diagonal blocks detect packet mixing induced by holonomy.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution;Komorebi.PaperIII.HolonomyBlocks.PacketResolution.block`

The certificate records a specified holonomy defect, the label of an input and output packet, and a proof that the corresponding block is nonzero.

Theorem 7.5 (Exact reconstruction and detection by obstruction blocks)**.** *The total obstruction has the exact finite resolution*

$$\boxed{\mathrm{Ob}_{c,b,a} = \sum_{\mu,\nu \in I_a} \mathrm{Ob}^{\mu\nu}_{c,b,a}\,.}$$

Moreover, the following are equivalent:

(i) $\Delta_{c,b,a} = 0$;

(ii) $H_{c,b,a} = \mathrm{id}$;

(iii) $\mathrm{Ob}_{c,b,a} = 0$;

(iv) $\mathrm{Ob}^{\mu\nu}_{c,b,a} = 0$ *for every* μ, ν.

Formal contract: finite algebraic core.

`Komorebi.PaperIII.HolonomyBlocks.obstruction_sum_blocks;Komorebi.PaperIII.HolonomyBlocks.obstructionBlocks_zero_iff;Komorebi.PaperIII.HolonomyBlocks.obstructionDetection_equivalences`

Proof. Insert completeness on both sides:

$$\mathrm{Ob} = \left(\sum_{\mu} \Pi^a_\mu\right) \mathrm{Ob} \left(\sum_{\nu} \Pi^a_\nu\right) = \sum_{\mu,\nu} \Pi^a_\mu \,\mathrm{Ob}\, \Pi^a_\nu.$$

Thus (iv) implies (iii), and (iii) implies (iv) by composition with the projectors. Conditions (ii) and (iii) are equivalent by the definition of Ob and additive cancellation. Conditions (i) and (ii) are equivalent by theorem 5.10. □

Why both packet indices matter

For the declared finite complete resolution, reconstruction uses both packet indices. Inspecting only diagonal blocks can miss transport from one packet to another. The full family records that mixing and reconstructs the total obstruction exactly.

Figure 3 depicts the finite block resolution on both sides of the obstruction and the off-diagonal blocks that witness packet mixing.

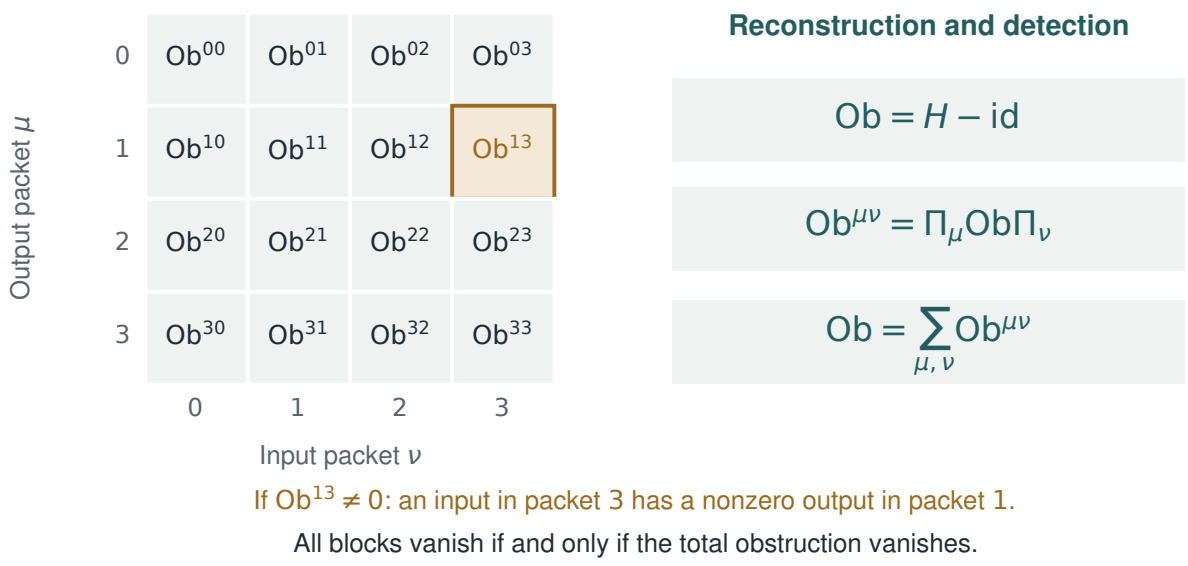


Figure 3. A finite complete packet resolution reconstructs the total holonomy obstruction from blocks indexed by input and output packets. A nonzero off-diagonal block is an exact witness of packet mixing.

Corollary 7.6 (Witness supported in one packet). *If* $\mathrm{Ob}^{\mu\nu}_{c,b,a} \neq 0$*, then there is a vector* x *supported in packet* ν *such that*

$$\Pi^a_\mu \,\mathrm{Ob}_{c,b,a}\, x \neq 0.$$

Such an x *is an exact finite witness of nonflatness.*

Formal contract: finite algebraic core.
`Komorebi.PaperIII.HolonomyBlocks.exists_packetWitness_of_obstructionBlock_ne_zero`

Proof. Classically choose y on which the nonzero linear map $\Pi^a_\mu \,\mathrm{Ob}\, \Pi^a_\nu$ is nonzero and set $x = \Pi^a_\nu y$. Idempotence gives $\Pi^a_\nu x = x$, and the required output is the chosen nonzero value. □

7.3 Gauge behavior of packet blocks

Vanishing or nonvanishing of the total triangle obstruction is gauge invariant by theorem 6.2, but a fixed packet pattern need not be. A general G_a can mix the projectors Π^a_μ.

Proposition 7.7 (Gauge covariance when packet projectors are preserved). *If* $G_a \Pi^a_\mu = \Pi^a_\mu G_a$ *for every* μ*, then*

$$\left(\mathrm{Ob}^G\right)^{\mu\nu} = G_a \,\mathrm{Ob}^{\mu\nu}\, G_a^{-1}.$$

Consequently the zero/nonzero status of each obstruction block is preserved by gauges that preserve the packet projectors.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.HolonomyBlocks.obstructionBlock_gauge_of_commutes;Komorebi.PaperIII.HolonomyBlocks.obstructionBlock_gauge_eq_zero_iff`

Proof. Use $\mathrm{Ob}^G = G_a \,\mathrm{Ob}\, G_a^{-1}$ and commute G_a and G_a^{-1} through the two packet projectors. Composition by an equivalence is injective in both directions. □

Boundary 7.8 (Packet patterns are not absolute). Without the commutation hypothesis in proposition 7.7, only the vanishing of the total obstruction is gauge invariant. The location of individual nonzero blocks may change under a gauge that mixes packets.

8 Products supported in packets and holonomy walks

The passage from words to walks also applies to products of different matrices. Let V be one finite packet type, let R be a semiring, and let $B_1, \dots, B_m \in R^{V\times V}$ be a finite sequence of possibly different packet matrices. Rows are targets and columns are sources as before.

Proposition 8.1 (Packet path expansion for different matrices)**.** *For $i, j \in V$ and $m \geq 1$, set $i_0 = i$ and $i_m = j$. Then*

$$(B_m \cdots B_1)_{ji} = \sum_{(i_1,\dots,i_{m-1})\in V^{m-1}} B_m[i_m, i_{m-1}] \cdots B_1[i_1, i_0].$$

When $m = 1$, the sum is over the unique empty tuple and its summand is $B_1[j, i]$. For $m = 0$, the product is the identity and the entry is the Kronecker delta δ_{ji}. If every $B_r[q, p]$ vanishes outside a declared relation $\mathcal{R}_r(p, q)$, then the endpoint entry is zero whenever there is no packet path compatible with the successive relations.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.EdgeOperators.heterogeneousProduct;Komorebi.PaperIII.EdgeOperators.heterogeneousPathSum;Komorebi.PaperIII.EdgeOperators.heterogeneousPacketPathExpansion;Komorebi.PaperIII.EdgeOperators.SupportedStep;Komorebi.PaperIII.EdgeOperators.SupportedPath;Komorebi.PaperIII.EdgeOperators.heterogeneousProduct_apply_eq_zero_of_no_supportedPath`

Proof. Expand each matrix product entry and distribute the finite sums. Induction on the factor list yields the displayed path sum, with the empty list evaluated as the identity matrix. If no supported intermediate packet exists, every summand contains an entry that is zero by its declared support relation. □

Boundary 8.2 (Changing packet types along a chain). The compiled theorems `heterogeneousPacketPathExpansion_two` and `heterogeneousPacketPathExpansion_three` allow different packet types at the successive stages of two- and three-factor products. The arbitrary-length theorem above permits different factors but uses one common finite packet type V. A fully dependent arbitrary-length chain of changing modules and packet types remains outside the finite algebraic contract.

Remark 8.3 (Checked specialization). When every B_r is one repeated endomorphism T, the recursive specialization to packet powers is inherited from Chapter I.

Inherited formal contract: Chapters I–II.
`Komorebi.PacketNaturalityCompletion.Packets.transferBlock_pow_eq_packetPathSum`

For a loop represented on one common finite scalar packet coordinate type, the proposition resolves its successive transport matrices into packet walks. If R is a ring, as in the reversible connection API, then $(H-I)_{ji} = H_{ji}-\delta_{ji}$. In particular, off-diagonal entries are unchanged. Thus a nonzero off-diagonal holonomy obstruction in this model with a common index type requires a transport word along the closed chart loop whose packet path runs from its input packet to its output packet. This conclusion requires declared block supports for the connection matrices. An arbitrary connection need not respect the transfer graph from section 2.

9 A minimal exact example

Example 9.1 (A scalar triangle)**.** Let $B = \{0, 1, 2\}$ and $C_0 = C_1 = C_2 = \Bbbk$ for a field $\Bbbk$. Choose $u, v, w \in \Bbbk^\times$ and set

$$\Gamma_{1\leftarrow 0}(x) = ux, \qquad \Gamma_{2\leftarrow 1}(x) = vx, \qquad \Gamma_{2\leftarrow 0}(x) = wx,$$

with reverse transports given by inverse scalars. Then

$$\Delta_{2,1,0}(x) = (vu - w)x,$$

and the triangle holonomy based at 0 is

$$H_{2,1,0}(x) = w^{-1}vux.$$

Therefore

$$H_{2,1,0} - \mathrm{id} = w^{-1}\Delta_{2,1,0}.$$

The connection is flat on this triangle exactly when $vu = w$. If the three coordinate systems are literally the same and canonical transport is used, then $u = v = w = 1$ and the obstruction vanishes. Choosing arbitrary nonzero u, v, w is a connection choice, not a consequence of changing exact packet coordinates.

Example 9.2 (A transfer walk with two steps)**.** Let a packet graph have edges $e : 0 \to 1$ and $f : 1 \to 2$, with no other edges. Then

$$T = L_e + L_f, \qquad (T^2)_{20} = c(f)c(e).$$

The products L_eL_e, L_eL_f, and L_fL_f vanish by typed mismatch. Only L_fL_e, corresponding to the walk $0 \to 1 \to 2$, contributes to the $(2, 0)$ entry. This example also fixes the noncommutative coefficient order.

10 Canonical coordinates for arbitrary maps

The analysis and synthesis identities are identities of functions, not only of linear maps. Consequently exact coordinate change does not become curved merely because the represented operation is nonlinear.

Proposition 10.1 (Exact naturality for arbitrary functions)**.** *Let $\mathcal{A}^X_a : X \xrightarrow{\sim} X_a$ and $\mathcal{A}^X_b : X \xrightarrow{\sim} X_b$ be exact coordinate equivalences for X, with inverses $\mathcal{S}^X_a$ and $\mathcal{S}^X_b$. Likewise, let $\mathcal{A}^Y_r : Y \xrightarrow{\sim} Y_r$ and $\mathcal{A}^Y_s : Y \xrightarrow{\sim} Y_s$ be*

exact coordinate equivalences for Y, with inverses $\mathcal{S}_r^Y$ and $\mathcal{S}_s^Y$. Let $F : X \to Y$ be any function, with no linearity or differentiability hypothesis. Set

$$J_{b\leftarrow a}^X = \mathcal{A}_b^X \circ \mathcal{S}_a^X, \qquad J_{s\leftarrow r}^Y = \mathcal{A}_s^Y \circ \mathcal{S}_r^Y,$$

and define

$$\widehat{F}_{r\leftarrow a} = \mathcal{A}_r^Y \circ F \circ \mathcal{S}_a^X.$$

Then

$$J_{s\leftarrow r}^Y \circ \widehat{F}_{r\leftarrow a} = \widehat{F}_{s\leftarrow b} \circ J_{b\leftarrow a}^X.$$

If $\mathcal{A}_t^Z : Z \xrightarrow{\sim} Z_t$ is another exact coordinate equivalence and $G : Y \to Z$ is any function, then exact coordinate representation also preserves composition:

$$\widehat{G \circ F}_{t\leftarrow a} = \widehat{G}_{t\leftarrow r} \circ \widehat{F}_{r\leftarrow a}.$$

Formal contract: finite algebraic core.

`Komorebi.PaperIII.Transport.exactCoordinateMap;Komorebi.PaperIII.Transport.exactCoordinateMap_comp;Komorebi.PaperIII.Transport.exactCoordinateMap_naturality`

Proof. Both sides of the naturality equation reduce to $\mathcal{A}_s^Y \circ F \circ \mathcal{S}_a^X$ using $\mathcal{S} \circ \mathcal{A} = \mathrm{id}$ in X and Y. For composition, the middle $\mathcal{S}^Y \circ \mathcal{A}^Y$ cancels in the same way. □

Boundary 10.2 (Where nonlinear defects can enter). A nonlinear *coordinate* defect cannot be blamed on the exact canonical J of definition 4.3. By proposition 10.1, that defect is identically zero. Nonzero defects require extra structure such as a noncanonical connection, truncation, approximation, restricted domains, or failure of an operation to respect a declared packet model.

11 Nonlinear operation trees: locality and the formalization frontier

11.1 The compiled foundation of unary and binary trees

Definition 11.1 (Finite grammar for operation trees)**.** For types of input labels I, constants K, unary symbols U, and binary symbols B, the finite formal grammar has four constructors:

$$\begin{aligned} t ::= \ & \mathrm{input}(i) \mid \mathrm{constant}(k) \mid \mathrm{unary}(u, t) \\ & \mid \mathrm{binary}(b, t_1, t_2). \end{aligned}$$

Given an arbitrary value type Y, evaluation takes completely explicit maps

$$I \to Y, \qquad K \to Y, \qquad U \to Y \to Y, \qquad B \to Y \to Y \to Y.$$

No algebraic, continuity, or differentiability law is imposed on these semantics.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.NonlinearObstructions.OperationTree;Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval`

When I has decidable equality, let $\mathrm{inputs}(t)$ be the finite set of input labels actually occurring in t.

Theorem 11.2 (Input locality for operation trees)**.** *Let $\eta_1, \eta_2 : I \to Y$ be two input environments. If*

$$\eta_1(i) = \eta_2(i) \qquad \textit{for every } i \in \mathrm{inputs}(t),$$

then the two evaluations of t are equal under the same constant, unary, and binary semantics. In particular, if $\mathrm{inputs}(t) = \emptyset$, evaluation is independent of the input environment.

Formal contract: finite algebraic core.
`Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval_congr_on_inputs;Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval_eq_of_inputs_eq_empty`

Proof. Structural induction on t. The input case is the agreement hypothesis. The constant case is reflexive. The unary case applies the induction hypothesis to its child. The binary case applies it independently to the left and right subtrees, whose input sets form a union. □

Boundary 11.3 (Exact force of locality). The theorem says evaluation reads only syntactically named inputs. It does not prove that the operations are linear or nonlinear in any analytic sense, that the tree is stable, that two different trees are equivalent, or that a solution to a tree equation exists.

11.2 Typed colored extension beyond the finite contract

Definition 11.4 (Finitary colored operation signature)**.** A finitary colored signature consists of:

- a finite type V of colors;
- a type Σ of operation symbols;
- an arity $r(\sigma) \in \mathbb{N}$ for every $\sigma \in \Sigma$;

- input colors $\mathrm{in}_\sigma : \mathrm{Fin}(r(\sigma)) \to V$;
- an output color $\mathrm{out}(\sigma) \in V$.

A realization assigns a space X_v to each color and a map

$$\Phi_\sigma : \prod_{q \in \mathrm{Fin}(r(\sigma))} X_{\mathrm{in}_\sigma(q)} \longrightarrow X_{\mathrm{out}(\sigma)}.$$

Outside the finite formal contract. The intended Lean target is a dependent inductive typed tree. No such declaration is part of the finite algebraic completion contract.

Definition 11.5 (Typed operation tree and evaluation)**.** A typed operation tree is recursively either a leaf of declared color or a node labeled by σ, with one correctly colored child for each input slot of σ. Evaluation is recursive:

$$\mathrm{ev}(\mathrm{node}(\sigma; t_1, \dots, t_{r(\sigma)})) = \Phi_\sigma(\mathrm{ev}(t_1), \dots, \mathrm{ev}(t_{r(\sigma)})).$$

Outside the finite formal contract. Candidate API: `Komorebi.PaperIII.NonlinearObstructions.ColoredOperationTree` and `ColoredOperationTree.eval`. This is stronger than the compiled uncolored unary–binary grammar.

11.3 Packet labelings and the multilinear first target

Suppose every X_v has a finite complete packet family $(P_i^v)_{i \in I_v}$. A packet labeling assigns an admissible packet index to each edge or node of a typed tree. For multilinear Φ_σ, inserting

$$\mathrm{id}_{X_v} = \sum_{i \in I_v} P_i^v$$

at every input and output yields a finite sum over trees with packet labels. This is the branching analogue of inserting intermediate projectors in proposition 8.1.

Formalization target 11.6 (Finite multilinear tree expansion). The nonlinear formalization target is an exact theorem saying that the evaluation of a finite tree of multilinear nodes equals the finite sum of its packet-resolved labeled evaluations. Required proof obligations include:

(a) a dependent well-typed tree syntax;

(b) finite packet-label types at every color;

(c) an explicit convention for nullary and unary nodes;

(d) multilinearity in every input slot;

(e) preservation of the order of input slots without assuming commutativity of operations.

Outside the finite formal contract. Candidate target: `Komorebi.PaperIII.NonlinearObstructions.ColoredOperationTree.eval_eq_sum_packetLabellings`.

11.4 Connection defects at nonlinear nodes

For coordinate models of the source and target colors, an additive target coordinate space, and a chosen noncanonical connection, a pointwise node defect can be defined by

$$\mathfrak{D}^{\Gamma}_{b,a}(\Phi) = \Gamma^{Y}_{b\leftarrow a} \circ \widehat{\Phi}_a - \widehat{\Phi}_b \circ \left(\prod_q \Gamma^{X_q}_{b\leftarrow a} \right).$$

This expression compares two *chosen* routes. With canonical exact transport in every slot it vanishes by proposition 10.1.

Formalization target 11.7 (Operation-tree defect propagation). For multilinear nodes, one expects an exact finite expansion obtained by replacing one or more nodes by their defects and transporting the remaining branches. For smooth nonlinear nodes, the corresponding chain rule is indexed by rooted partitions and Faà di Bruno trees. A formal theorem will require a fixed differentiability category, base points, derivative conventions, symmetry factors, and remainder policy.

Outside the finite formal contract. Tree-level defect identities, higher chain rules, and nonlinear norm bounds are outside the finite Lean contract. Ordinary mathematical results in these directions appear in Part II under their stated hypotheses.

Boundary 11.8 (Analytic and geometric exclusions). The finite formal contract does not include convergence of an infinite tree series, existence of derivatives, a Banach-space Faà di Bruno theorem, a Baker–Campbell–Hausdorff expansion, a continuum limit, or an identification of algebraic holonomy with curvature of a smooth bundle. Part II supplies ordinary proofs for its explicit nonlinear and analytic statements. The theory does not derive a connection from data, optimization, physics, or probability.

12 Finite algebraic formalization ledger

Topic	Status	Declaration or roadmap target	Exact boundary
Packet matrix and edge operator	Lean-backed III	`PacketGraph.packetMatrix`; `EdgeSystem.edgeOp`	Finite scalar matrices over a semiring, with one scalar coordinate per packet.
Edge support and pair dichotomy	Lean-backed III	`edgeOp_eq_zero_off_support`; `edgeOp_mul_edgeOp_eq_zero`; `edgeOp_mul_edgeOp_of_composable`	Broken interfaces vanish. Matched pairs collapse to the endpoint block with coefficient in operator order.
Typed word evaluation	Lean-backed III	`Composable`; `wordOp`; `wordOp_append`; `wordOp_eq_zero_of_not_composable`	List order is execution order. Concatenation reverses operator-product order. Nonwalks vanish.

Topic	Status	Declaration or roadmap target	Exact boundary
Finite word and walk expansions	Lean-backed III	`explicitWordExpansion`; `walkExpansion`; `wordExpansion_eq_explicitWordExpansion`; `transferOp_pow_eq_walkExpansion`	Actual finite tuple sum and zero-filtered composable-walk sum over finite graph and packet types.
Endpoint coefficient walk sum	Lean-backed III	`reverseCoefficient`; `Connects`; `transferOp_pow_apply_eq_sum_walks`	Exact scalar endpoint entry, including the empty-walk case and noncommutative coefficient order.
Fixed reachability and orbit support	Lean-backed III, with an inherited hull bridge	`transferOp_pow_apply_eq_zero_of_no_walk`; `transferOp_pow_mulVec_apply_eq_zero_of_unreachable_support`	Finite powers and supported seeds. The orbit-hull span conclusion uses the stated Chapter I contract.
Exact frame transport	Lean-backed III	`exactTransport_cocycle`; `exactPathTransport_eq_endpoint`; `exactPathTransport_loop`	One frame family on a single module. Arbitrary finite paths telescope and closed paths are trivial.
Exact coordinate cocycle	Inherited Lean-backed II	`NestedRefinement.refineCoordinates_comp`; `NestedRefinement.coordinateRefinementEquiv_trans`	Complete finite projector coordinates, not reduction.
Typed coordinate naturality	Inherited Lean-backed II	`NestedRefinement.coordinateOperator_comp`; `NestedRefinement.coordinateOperator_naturality`	Linear maps over semiring modules.
Minimal connection API	Lean-backed III	`Transport.Connection`; `cocycleDefect`; `triangleHolonomy`; `triangleHolonomy_eq_refl_of_flat`	Only self-normalization is assumed. Reverse inversion and the cocycle are not structure fields.
Twisted three-point loop	Lean-backed III	`twistedThreePointConnection`; `twisted_triangleHolonomy_ne_refl`	Concrete rational connection. The selected loop sends 1 to 2.
Reversible additive defect	Lean-backed III	`ReversibleConnection`; `triangleHolonomy_sub_id`; `cocycleDefect_eq_zero_iff_triangleHolonomy_eq_refl`	Heterogeneous module fibers, reverse-edge inversion, and an additive defect over a commutative ring.

Topic	Status	Declaration or roadmap target	Exact boundary
Reversible finite paths	Lean-backed III	`pathTransport_eq_direct_of_flat`; `indexedPathTransport_eq_direct_of_flat`; `indexedLoopTransport_eq_refl_of_flat`	Finite pair-connection paths collapse to endpoints under algebraic flatness.
Gauge covariance	Lean-backed III	`gauge_cocycleDefect`; `gauge_triangleHolonomy`; `gauge_flat_iff`	Fiberwise algebraic conjugation, without smooth gauge theory.
Flat versus canonical transport	Lean-backed III	`flatGauge`; `flat_isGaugeOfCanonical`	A chosen base chart and ambient frames exhibit every flat pair connection as a gauge transform of canonical transport.
Operation-tree locality	Lean-backed III	`OperationTree`; `OperationTree.eval`; `OperationTree.eval_congr_on_inputs`	Arbitrary unary and binary semantics. Only dependence on named inputs is proved.
Generic obstruction packet	Lean-backed III	`ObstructionPacket.not_isSolution`; `not_isSolution_iff_nonempty_obstructionPacket`	Exact candidate-specific logical certificate, without global classification.
Holonomy block obstruction	Lean-backed III	`obstruction_sum_blocks`; `obstructionBlocks_zero_iff`; `obstructionDetection_equivalences`	Finite complete idempotent packet family on the base fiber. Orthogonality is not required by the compiled reconstruction.
Packet witness and block gauge	Lean-backed III	`exists_packetWitness_of_obstructionBlock_ne_zero`; `obstructionBlock_gauge_of_commutes`; `obstructionBlock_gauge_eq_zero_iff`	Witness extraction is classical. Blockwise covariance requires the gauge to commute with every packet projector.
Heterogeneous matrix paths	Lean-backed III	`heterogeneousPacketPathExpansion`; `heterogeneousProduct_apply_eq_zero_of_no_supportedPath`	Arbitrarily many different matrices over one common finite packet type. Fully typed changing-index chains compile only at lengths two and three.

Topic	Status	Declaration or roadmap target	Exact boundary
Arbitrary-function coordinates	Lean-backed III	`exactCoordinateMap_comp`; `exactCoordinateMap_naturality`	Exact set-level equivalences and arbitrary functions, without a linearity or differentiability assumption.
Colored and smooth trees	Roadmap only	roadmap `ColoredOperationTree.eval_eq_sum_packetLabellings`	Dependent colors, multilinear packet expansion, derivatives, and Faà di Bruno identities remain outside the contract.

12.1 Kernel-backed Chapter III surface

The source register identifies the authoritative declaration-level contract. It groups the compiled surface as follows.

- **EdgeOperators and EdgeExtensions:** scalar packet matrices, typed edge and word support, finite word/walk expansions, endpoint coefficient sums, fixed-length reachability, orbit support, and common-index heterogeneous products with declared supports.
- **Transport and ReversibleConnections:** exact frame transport, the minimal connection, heterogeneous reversible pair transport, additive defect and triangle holonomy, finite paths, canonical connections, and the fiberwise gauge action.
- **CoordinateNaturality and HolonomyBlocks:** composition and naturality for arbitrary functions between exact coordinate systems, finite two-sided block reconstruction, equivalent obstruction criteria, packet-supported witnesses, and packet-preserving block gauges.
- **NonlinearObstructions:** finite unary–binary operation trees, input locality, and candidate-specific obstruction certificates.

The formal companion records the checker-generated declaration and audit totals. The parity gate requires every fully qualified Chapter III target printed beside a statement to occur in the theorem map, rejects stale or duplicate targets, and checks it against the compiled environment. `KernelAudit.lean` reports dependencies for the mapped and supporting theorems, and its fail-closed gate scans every compiled `Komorebi.PaperIII` declaration. The four inherited statement contracts remain identified separately by their inherited declaration names.

Boundary 12.1 (No scope promotion by prose). The displayed endpoint coefficient formula, reachability results, reversible connection theorems, additive defect–holonomy identity, gauge covariance, holonomy-block reconstruction, packet witnesses, common-index heterogeneous path expansion, and arbitrary-function coordinate naturality are all part of the finite algebraic kernel-backed surface. What remains outside is the dependent fiber-valued edge-system generalization, a fully dependent arbitrary-length chain whose module and packet type change at every stage, dependent colored operation trees, packet-labeled multilinear tree expansions, nonlinear tree-defect and Faà di Bruno identities, and the analytic and geometric extensions listed above.

13 The finite contract and the nonlinear transition

Typed packet support turns the expansion of a transfer power into a walk calculus. Incompatible words vanish because their interfaces do not match. Canonical packet coordinate transport is the exact equivalence obtained from analysis and synthesis. It satisfies the cocycle law, so every finite canonical loop telescopes to the identity. Holonomy enters after a separate connection is supplied. The compiled minimal API proves that flatness makes both the relative defect and holonomy trivial, and it exhibits a nontrivial rational loop. Under the compiled reversible API, a nonzero additive triangle defect is equivalent to nonidentity holonomy, transforms covariantly under gauge, and is detected exactly by a finite family of packet blocks. Finite flat paths collapse to their endpoints. Every flat reversible pair connection is explicitly related to canonical transport by a gauge change through a chosen base chart.

These facts define the boundary of the finite formal contract. At that boundary, exact coordinates preserve composition and satisfy naturality for arbitrary functions in Chapter III, while the typed linear packet coordinate counterpart remains inherited from Chapter II. Nontrivial nonlinear defects must therefore be attached to additional modeling choices or to packet compatibility of the particular operation. Colored operation trees, multilinear packet labelings, and Faà di Bruno expansions remain outside this finite Lean contract. Part II develops the corresponding ordinary mathematics, including packet jets, rooted tree expansions, and explicit nonlinear obstruction theorems.

Part II Nonlinear packet calculus, obstruction complexes, and holonomy

This part moves from exact coordinates for arbitrary functions to differentiable interactions, specified correction problems, and finite or smooth transport. Each ordinary proof retains its stated hypotheses. These results are not asserted to belong to the finite Lean contract of Part I.

The notational change from $\mathcal{A}, \mathcal{S}$ to $\mathcal{A}, \mathcal{S}$ is typographical: both denote analysis and synthesis for the declared packet system. A connection remains additional data. Smooth canonical transport is constructed from a moving projector family, whereas the algebraic exact coordinate transport in Part I compares fixed exact descriptions of one ambient object.

14 Introduction

A finite family of pairwise annihilating idempotents gives exact coordinates. If

$$P_\alpha P_\beta = 0 \quad \text{for } \alpha \neq \beta, \qquad \sum_\alpha P_\alpha = I,$$

then every vector is the sum of its packet components. Linear operators become block matrices in these coordinates. A nonlinear map cannot be reconstructed from its diagonal restrictions, since mixed interactions between packet components are usually present. It can still be represented exactly if its input is the complete packet tuple.

This part develops nonlinear packet representation and the obstruction theory that follows from it. Exact coordinates preserve a nonlinear problem, but they do not solve it. Once the equation is written in packet coordinates, its derivatives, compositions, correction maps, refinement failures, and gluing defects can be separated without discarding mixed terms.

The first stage of Part II treats the local nonlinear calculus. Packet analysis and synthesis conjugate arbitrary maps. Fréchet derivatives split into multilinear packet jets. The higher chain rule then becomes a Faà di Bruno formula whose terms carry intermediate packet labels. Standard background on Fréchet calculus and the higher chain rule may be found in [1, 2, 4].

The next stage studies nonlinear iteration. Repeated substitution is indexed by rooted trees rather than operator words. A scalar majorant gives an absolute convergence theorem for polynomial fixed-point equations. The quadratic case recovers the Catalan series. Rooted tree combinatorics are discussed in [16, 11, 6].

A linear correction map

$$D : U \longrightarrow V$$

places the obstruction to solving $Du = -r$ in the canonical quotient coker D. This quotient has a universal property and behaves naturally under commuting squares. Mapping cones extend the same construction to cochain complexes. In a Hilbert space with closed range, the quotient is represented by the orthogonal projection onto Ker D^*. The Moore–Penrose inverse then gives the least norm correction. Relevant background appears in [18, 15].

A vanishing linear obstruction is only the first step. We give a contraction theorem for nonlinear correction, followed by a local Lyapunov–Schmidt reduction. In a complete filtered problem, every lifting step has its own cokernel class. Vanishing at all depths and compatible choices of corrections produce an exact inverse limit solution. A filtered correction complex carries a spectral sequence. Its first page records graded obstruction candidates, while later differentials record interactions that can cancel them. We use the standard spectral sequence of a filtered complex from [18, 12].

The final stage studies elimination and gluing. The implicit function theorem gives an exact Schur derivative after hidden variables are removed. Finite transport systems reduce, after a spanning tree gauge, to one holonomy element for each fundamental cycle. The same idea extends to smooth projector families through Kato transport. We use standard material from [4, 5, 7, 8, 10].

The chapter does not claim that every nonlinear problem has a finite obstruction space. The cokernel is canonical only after a correction map has been specified. It may be infinite dimensional. A first-order obstruction may vanish while a higher-order obstruction remains. The purpose of the calculus is to identify the exact object in which each failure lives and to show how those objects change under packets, refinement, filtration, and transport.

15 A map of the nonlinear calculus

Four questions organize the nonlinear calculus: representation, interaction, correction, and gluing.

The first question is representation. Given a nonlinear map F, can its packet form be written without approximation? The answer is yes. Analysis and synthesis are inverse coordinate maps, so conjugation preserves the full equation.

The second question is interaction. Which input packets influence a given output packet, and at what derivative order? Packet jets answer this question. They form a directed hypergraph rather than an ordinary block matrix because one nonlinear interaction can have several inputs.

The third question is correction. Suppose an approximate solution leaves a defect r. A declared linearized correction map D removes the part of r lying in $\operatorname{im} D$. The quotient class

$$[r] \in \operatorname{coker} D$$

is the part that no allowed first-order correction can remove. This quotient class is the obstruction. Its vanishing must be supplemented by a nonlinear theorem.

The fourth question is gluing. Local coordinate changes can solve every edge problem and still fail around a loop. The remaining cycle product is holonomy. It measures the failure of a global frame even when each local equation has been solved.

These four layers fit into the sequence

$$\begin{aligned}\text{exact coordinates} \longrightarrow \text{packet interactions} \longrightarrow \text{correction classes}\\ \longrightarrow \text{global gluing.}\end{aligned}$$

Each arrow needs hypotheses. Exact coordinates need a complete packet system. Differential interactions need regularity. A cokernel obstruction needs a specified correction map. Holonomy needs invertible local transports.

A second distinction concerns scale. A refinement defect compares a fine nonlinear model with a coarse one. It is a point in the coarse output space. An obstruction class is the image of such a defect, or

of another residual, in a quotient by allowed corrections. The correction map must be declared before the defect determines an obstruction class. A small defect can have a nonzero obstruction, and a large defect can have zero obstruction if it lies entirely in the correction range.

The chapter uses finite packet systems in the coordinate and graph sections. Banach space hypotheses enter only where norms, contractions, or Fréchet derivatives are used. Module and complex language is used for cokernels, filtered lifting, and cohomological obstructions. This separation keeps each result at the level of generality supported by its proof.

Standing conventions

All packet systems are finite unless a statement says otherwise. In a normed setting every packet projector and every coarsening operator is bounded. Differentiability of operator-valued maps is taken in operator norm. A Banach space isomorphism is a bounded linear bijection with bounded inverse. For a decreasing filtration $(F^m X)$, completeness means that the canonical map

$$X \longrightarrow \varprojlim_m X/F^m X$$

is an isomorphism and separatedness means $\bigcap_m F^m X = 0$. Statements about congruence classes in a filtered nonlinear problem also assume that the nonlinear map preserves filtration congruences at the stated depth. These assumptions are repeated where they are essential to a proof.

16 Exact nonlinear packet coordinates

Let X be a module over a unital ring. A finite packet system on X is a family

$$\mathcal{P} = (P_\alpha)_{\alpha \in I} \subseteq \mathrm{End}(X)$$

of nonzero idempotents such that

$$P_\alpha P_\beta = 0 \quad \text{when } \alpha \neq \beta, \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

Write

$$X_\alpha = P_\alpha X, \qquad X^{\mathcal{P}} = \bigoplus_{\alpha \in I} X_\alpha.$$

The analysis and synthesis maps are

$$\mathcal{A}_{\mathcal{P}} x = (P_\alpha x)_\alpha, \qquad \mathcal{S}_{\mathcal{P}} (x_\alpha)_\alpha = \sum_\alpha x_\alpha.$$

They are inverse isomorphisms.

Definition 16.1 (Nonlinear packet coordinate map)**.** Let $\mathcal{P}$ be a packet system on X and let $\mathcal{Q}$ be a packet system on Y. For any map $F : X \to Y$, define

$$F_{\mathcal{Q} \leftarrow \mathcal{P}} = \mathcal{A}_{\mathcal{Q}} F \mathcal{S}_{\mathcal{P}}.$$

Thus

$$F_{\mathcal{Q}\leftarrow\mathcal{P}}((x_\alpha)_\alpha) = \left(Q_\gamma F\left(\sum_\alpha x_\alpha\right)\right)_\gamma.$$

Theorem 16.2 (Exact nonlinear coordinate theorem)**.** *For every map $F : X \to Y$,*

$$\mathcal{S}_{\mathcal{Q}} F_{\mathcal{Q}\leftarrow\mathcal{P}} \mathcal{A}_{\mathcal{P}} = F.$$

If $G : Y \to Z$ and $\mathcal{R}$ is a packet system on Z, then

$$(G \circ F)_{\mathcal{R}\leftarrow\mathcal{P}} = G_{\mathcal{R}\leftarrow\mathcal{Q}} \circ F_{\mathcal{Q}\leftarrow\mathcal{P}}.$$

No linearity, continuity, or differentiability assumption is needed.

Proof. The identities $\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}} = I_X$ and $\mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}} = I_Y$ give

$$\mathcal{S}_{\mathcal{Q}} F_{\mathcal{Q}\leftarrow\mathcal{P}} \mathcal{A}_{\mathcal{P}} = \mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}} F \mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}} = F.$$

For composition,

$$G_{\mathcal{R}\leftarrow\mathcal{Q}} F_{\mathcal{Q}\leftarrow\mathcal{P}} = \mathcal{A}_{\mathcal{R}} G \mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}} F \mathcal{S}_{\mathcal{P}} = \mathcal{A}_{\mathcal{R}}(G \circ F)\mathcal{S}_{\mathcal{P}}.$$

□

Corollary 16.3 (Dynamical equivalence)**.** *Suppose $X = Y$ and the same packet system is used on both sides. Packet coordinates preserve fixed points, periodic points, injectivity, surjectivity, and invertibility. In particular,*

$$F(x) = x \quad \Longleftrightarrow \quad F_{\mathcal{P}\leftarrow\mathcal{P}}(\mathcal{A}_{\mathcal{P}} x) = \mathcal{A}_{\mathcal{P}} x.$$

Proof. The packet map is conjugate to F through the isomorphism $\mathcal{A}_{\mathcal{P}}$. Every listed property is invariant under conjugacy. □

Proposition 16.4 (Change of packet coordinates)**.** *Let $\mathcal{P}$ and $\mathcal{P}'$ be packet systems on X. Let $\mathcal{Q}$ and $\mathcal{Q}'$ be packet systems on Y. For any two packet systems $\mathcal{U}$ and $\mathcal{V}$ on the same space, write*

$$J_{\mathcal{U}\leftarrow\mathcal{V}} = \mathcal{A}_{\mathcal{U}}\mathcal{S}_{\mathcal{V}}.$$

Then

$$F_{\mathcal{Q}'\leftarrow\mathcal{P}'} = J_{\mathcal{Q}'\leftarrow\mathcal{Q}} F_{\mathcal{Q}\leftarrow\mathcal{P}} J_{\mathcal{P}\leftarrow\mathcal{P}'}.$$

Proof. Insert $\mathcal{S}_{\mathcal{Q}}\mathcal{A}_{\mathcal{Q}} = I_Y$ and $\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}} = I_X$ into the definition of $F_{\mathcal{Q}'\leftarrow\mathcal{P}'}$. □

Exactness by cancellation

Analysis and synthesis undo each other on the complete packet tuple. In a composition of packet descriptions, the intermediate synthesis and analysis maps cancel. This compares exact descriptions of the same ambient object. A connection adds data for transport along parameters or paths, with its own hypotheses.

Example 16.5 (A mixed quadratic map)**.** Let $X = Y = \Bbbk^2$ with coordinate packets. For

$$F(x_1, x_2) = (x_1 x_2, x_1^2 + x_2),$$

the first output packet depends on both input packets. The second output packet contains a quadratic contribution from the first input packet and a linear contribution from the second. Packet coordinates record these interactions exactly. Replacing F by the two diagonal restrictions would lose the product $x_1 x_2$.

Remark 16.6. Exact packet coordinates do not imply packet independence. They retain the complete input tuple. Every output packet may depend on every input packet.

17 Packet jets and higher composition

Let X and Y be Banach spaces over the same field $\mathbb{R}$ or $\mathbb{C}$ and let the packet projectors on both spaces be bounded. In the complex case, derivatives are complex Fréchet derivatives and the Taylor integral below has a real parameter. Assume that $F : X \to Y$ is m times continuously Fréchet differentiable. The same local statements hold on open domains containing the points and segments in question. For $1 \le r \le m$, the derivative

$$D^r F(x) : X^r \longrightarrow Y$$

is a continuous symmetric multilinear map.

The range X_α is closed because $X_\alpha = \ker(I - P_\alpha)$. It is therefore Banach with its inherited norm. The same holds for every Y_γ. Give the finite product of ranges the maximum norm. Packet analysis and synthesis are bounded, since they are formed from finitely many bounded projectors and a finite sum. Their algebraic inverse identities from section 16 make them bounded linear isomorphisms. For a nonempty input family, $\|\mathcal{A}_\mathcal{P}\| \le \max_\alpha \|P_\alpha\|$ and $\|\mathcal{S}_\mathcal{P}\| \le |I|$. An empty complete family can occur only on the zero space. Its coordinate product is the zero vector space with norm zero, and the coordinate identities hold directly.

Definition 17.1 (Packet jet)**.** For $1 \le r \le m$, define

$$J^r_{\gamma|\alpha_1,\dots,\alpha_r} F(x) = Q_\gamma D^r F(x)\big|_{X_{\alpha_1} \times \cdots \times X_{\alpha_r}}.$$

This is a continuous multilinear map

$$X_{\alpha_1} \times \cdots \times X_{\alpha_r} \longrightarrow Y_\gamma.$$

Theorem 17.2 (Exact reconstruction of derivatives)**.** *For $h_1, \dots, h_r \in X$,*

$$D^r F(x)[h_1, \dots, h_r] = \sum_\gamma \sum_{\alpha_1,\dots,\alpha_r} J^r_{\gamma|\alpha_1,\dots,\alpha_r} F(x)[P_{\alpha_1} h_1, \dots, P_{\alpha_r} h_r].$$

Proof. Insert $I_Y = \sum_\gamma Q_\gamma$ on the output and $I_X = \sum_\alpha P_\alpha$ in every input slot. Multilinearity expands the finite sums. □

An exact jet model with two packets

$F(x,y) = (x + xy,\ y + x^2)$ at $(0,0)$. Coordinate packets in both input and output.

$$DF(0) = I: \quad DF(0)e_x = (1,0), \qquad DF(0)e_y = (0,1).$$

Output 1

H_1

	x	y
x	0	**1**
y	**1**	0

$D^2F_1(0)[e_x, e_y] = D^2F_1(0)[e_y, e_x] = 1$

Output 2

H_2

	x	y
x	**2**	0
y	0	0

$D^2F_2(0)[e_x, e_x] = 2$

$$D^2F(0)[h,k] = (h_xk_y + h_yk_x,\ 2h_xk_x)$$

$$\tfrac{1}{2}D^2F(0)[h,h] = (h_xh_y,\ h_x^2)$$

$F(h) = h + \frac{1}{2}D^2F(0)[h,h]$ exactly. All higher jets vanish.

Cells are ordered Hessian responses: the two mixed slots and the Taylor factor 1/2 are both retained.

Figure 4. Second packet jets for $F(x,y) = (x + xy, y + x^2)$ at the origin. Coordinate axis projectors give $DF(0) = I$. The first output has mixed second derivatives equal to one, while the second has a repeated first input derivative equal to two. Each grid records the ordered pair of input packets. The mixed entries retain an interaction that diagonal restrictions would lose.

Proposition 17.3 (Jet norm bound)**.** *Give every packet range its inherited subspace norm. The restricted jet satisfies the sharper estimate*

$$\left\|J^r_{\gamma|\alpha_1,\dots,\alpha_r}F(x)\right\| \le \|Q_\gamma\|\,\|D^rF(x)\|.$$

Its ambient extension inserts P_{α_j} in the jth slot and has norm at most $\|Q_\gamma\|\,\|D^rF(x)\|\prod_j\|P_{\alpha_j}\|$. Consequently the operator norm of the restricted packet jet also satisfies

$$\left\|J^r_{\gamma|\alpha_1,\dots,\alpha_r}F(x)\right\| \le \|Q_\gamma\|\,\|D^rF(x)\|\prod_{j=1}^{r}\|P_{\alpha_j}\|.$$

When all packets are orthogonal projections on Hilbert spaces, every nonzero projector has norm one and the bound reduces to the norm of the full derivative.

Proof. For $h_j \in X_{\alpha_j}$, the inherited norm gives

$$\|Q_\gamma D^rF(x)[h_1,\dots,h_r]\| \le \|Q_\gamma\|\,\|D^rF(x)\|\prod_j\|h_j\|.$$

Taking the multilinear operator norm proves the sharper restricted estimate. For ambient inputs, replace h_j by $P_{\alpha_j}h_j$ and apply the operator norm bound to each projector. Every nonzero idempotent has norm at least one, since it fixes a nonzero vector in its range. The displayed product bound therefore also follows for the restricted jet. If a zero packet is retained as a redundant index, multilinearity makes any jet with that input or output packet zero, so both estimates remain valid. □

Corollary 17.4 (Packet Taylor formula)**.** *Assume F is $m+1$ times continuously Fréchet differentiable on an open set containing the segment $x+th$ for $0\le t\le 1$. Then*

$$\begin{aligned}Q_\gamma F(x+h) = Q_\gamma F(x) &+ \sum_{r=1}^{m}\frac{1}{r!}\sum_{\alpha_1,\dots,\alpha_r} J^r_{\gamma|\alpha_1,\dots,\alpha_r}F(x)[P_{\alpha_1}h,\dots,P_{\alpha_r}h]\\ &+ Q_\gamma R_{m+1}(x,h),\end{aligned}$$

where the Bochner integral remainder is

$$R_{m+1}(x,h) = \frac{1}{m!}\int_0^1 (1-t)^m D^{m+1}F(x+th)[h,\dots,h]\,dt.$$

Proof. Set $g(t) = F(x+th)$ on the open preimage of the domain under $t\mapsto x+th$. It contains $[0,1]$, and repeated differentiation gives $g^{(r)}(t) = D^rF(x+th)[h,\dots,h]$. The derivatives are continuous on the compact interval, so their Bochner integrals exist in the Banach codomain. Put

$$T_j = \frac{1}{j!}\int_0^1 (1-t)^j g^{(j+1)}(t)\,dt \qquad (0\le j\le m).$$

The fundamental theorem of calculus gives $T_0 = g(1)-g(0)$. For $j\ge 1$, integration by parts gives $T_j = T_{j-1} - g^{(j)}(0)/j!$. Iterating this relation proves the Taylor identity with remainder $T_m = R_{m+1}(x,h)$ and its exact factorial normalization.

Apply Q_γ to the identity and use theorem 17.2. Pairwise annihilation and idempotence retain exactly the output packet γ. Boundedness of Q_γ allows it to pass through the Bochner integral. In particular, if $\|D^{m+1}F(x+th)\| \le M$ along the segment, then

$$\|Q_\gamma R_{m+1}(x,h)\| \le \frac{\|Q_\gamma\| M \|h\|^{m+1}}{(m+1)!}.$$

For $m = 0$ the Taylor sum is empty. For $h = 0$, multilinearity makes every positive-order term and the remainder zero. □

The relation to derivatives of the complete coordinate representation is also literal. For an open domain $U \subseteq X$, put $\widehat{F} = \mathcal{A}_{\mathcal{Q}} \circ F \circ \mathcal{S}_{\mathcal{P}}$ on the open coordinate set $\mathcal{A}_{\mathcal{P}}(U)$. Boundedness and linearity of analysis and synthesis imply, for $1 \le r \le m$,

$$D^r\widehat{F}(u)[v_1, \dots, v_r] = \mathcal{A}_{\mathcal{Q}}(D^r F(\mathcal{S}_{\mathcal{P}}u)[\mathcal{S}_{\mathcal{P}}v_1, \dots, \mathcal{S}_{\mathcal{P}}v_r]).$$

Indeed, the first derivative follows from the chain rule. Repeating it adds no higher derivatives of the fixed linear coordinate maps. If each v_j is supported in one coordinate slot α_j, synthesis includes that slot into X. Its γ output component is exactly the range jet defined above, including when a chosen direction is zero.

At a fixed base point, the nonzero packet jets form a directed hypergraph. An order r interaction has r input packet labels and one output packet label. The support therefore records interactions relative to the chosen packet system.

For a block $B = \{i_1 < \cdots < i_s\}$ and an intermediate packet γ, write

$$G_{\gamma,B}(x \mid h) = J^s_{\gamma|\alpha_{i_1},\dots,\alpha_{i_s}} G(x)[h_{i_1}, \dots, h_{i_s}].$$

Arity of packet jets

A packet jet has one output slot and as many input slots as its derivative order. In the composition formula, each partition block collects inputs for one derivative of the inner map. The number of blocks determines the arity of the derivative of the outer map. Intermediate packet labels record the outputs passed between them.

Theorem 17.5 (Packet Faà di Bruno formula)**.** *Let $n \ge 1$, let X, Y, Z be Banach spaces over the same real or complex field, and let $G : X \to Y$ and $F : Y \to Z$ be n times continuously Fréchet differentiable. Let $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ be packet systems on X, Y, and Z. Fix input packets $\alpha_1, \dots, \alpha_n$, an output packet δ, and directions $h_i \in X_{\alpha_i}$. Then*

$$\begin{aligned} &J^n_{\delta|\alpha_1,\dots,\alpha_n}(F \circ G)(x)[h_1, \dots, h_n] \\ &= \sum_{\pi=\{B_1,\dots,B_k\}} \sum_{\gamma_1,\dots,\gamma_k} J^k_{\delta|\gamma_1,\dots,\gamma_k} F(Gx)[G_{\gamma_1,B_1}(x \mid h), \dots, G_{\gamma_k,B_k}(x \mid h)]. \end{aligned}$$

The first sum runs over all set partitions of $\{1, \dots, n\}$.

Proof. First derive the unprojected partition formula. At order one it is the ordinary chain rule. Suppose it holds at order n and the maps are C^{n+1}. Differentiating a summand in direction h_{n+1} has two kinds

of contribution. A derivative of the outer map creates the new singleton block $\{n+1\}$. A derivative of an inner block jet adjoins $n+1$ to that block. Every partition of $\{1,\dots,n+1\}$ arises exactly once: if $\{n+1\}$ is a block, remove that block. Otherwise remove $n+1$ from the unique block that contains it. The chain and product rules for bounded multilinear evaluation justify the differentiation. This proves the unprojected formula by induction.

Insert $I_Y = \sum_\gamma Q_\gamma$ in each argument of the outer derivative and expand the finite sums by multilinearity. Apply R_δ to the output. The directions already lie in the declared input ranges, and each intermediate projection lands in its literal output range. The resulting summands are exactly the displayed packet jets. List directions within a block in increasing index order and blocks in increasing least-element order. Symmetry of the higher derivative makes the final value independent of the block ordering. □

Proposition 17.6 (Number of labeled composition terms)**.** *For $n \geq 1$, if Y has M packets, then the number of packet labeled terms at order n before zero jets are removed is*

$$N_{n,M} = \sum_{k=1}^{n} S(n,k) M^k,$$

where $S(n,k)$ is a Stirling number of the second kind. Thus $N_{n,M}$ is the nth Touchard polynomial evaluated at M.

Proof. A set partition with k blocks receives one independent intermediate packet label on each block. It therefore gives M^k labeled terms. There are $S(n,k)$ such partitions. □

Corollary 17.7 (Structural pruning)**.** *For a map H with fixed input and output packet systems, let $\operatorname{supp}_r H(x)$ be the set of external profiles $(\delta;\alpha_1,\dots,\alpha_r)$ for which the restricted jet $J^r_{\delta|\alpha_1,\dots,\alpha_r} H(x)$ is nonzero. A profile belongs to the admissible composition set if it admits a partition $\pi = \{B_1,\dots,B_k\}$ and intermediate labels $\gamma_1,\dots,\gamma_k$ such that the outer jet of F and every inner block jet of G in theorem 17.5 are nonzero. Then $\operatorname{supp}_n(F \circ G)(x)$ is contained in this admissible set. In particular, a summand vanishes whenever one of its constituent packet jets vanishes.*

Proof. If a profile has no admissible decorated partition, every summand in theorem 17.5 has a zero constituent and therefore vanishes. The composed jet is then the zero multilinear map. This proves the support inclusion by contraposition. The inclusion can be strict. Nonzero constituent maps may compose to zero, and nonzero summands may cancel. Thus the admissible set is a structural upper bound rather than a claim of equality. □

Example 17.8 (The second-order composition law)**.** At order two, the partition formula has two shapes. For packet directions h_1 and h_2,

$$\begin{aligned} R_\delta D^2(F\circ G)(x)[h_1,h_2] = &\sum_{\gamma_1,\gamma_2} J^2_{\delta|\gamma_1,\gamma_2} F(Gx)[J^1_{\gamma_1|\alpha_1} G(x)h_1, J^1_{\gamma_2|\alpha_2} G(x)h_2] \\ &+ \sum_{\gamma} J^1_{\delta|\gamma} F(Gx)[J^2_{\gamma|\alpha_1,\alpha_2} G(x)[h_1,h_2]]. \end{aligned}$$

The first line records two first-order changes entering a quadratic interaction of F. The second records a genuine second-order interaction created inside G before F is applied.

Remark 17.9. Jet sparsity is not invariant under an arbitrary change of coordinates. A useful packet system normally comes from symmetry, scale, support, grading, valuation, or another structure that is already present in the problem. Every sparsity claim should therefore name the packet system with respect to which it is made.

18 Rooted tree expansions

Repeated nonlinear substitution is indexed by rooted trees. We begin with a polynomial fixed point equation on a Banach space.

Let X be a Banach space. Fix $s \in X$ and continuous multilinear maps

$$B_r : X^r \longrightarrow X, \qquad 2 \le r \le d.$$

Set

$$a = \|s\|, \qquad b_r = \|B_r\|, \qquad \phi(t) = a + \sum_{r=2}^{d} b_r t^r.$$

An ordered rooted tree has leaves labeled by s and every internal vertex of arity r labeled by B_r. Its value is obtained by evaluating from the leaves toward the root.

Theorem 18.1 (Polynomial tree expansion). *Suppose there is $R > 0$ such that*

$$\phi(R) \le R$$

and

$$q = \sum_{r=2}^{d} r b_r R^{r-1} < 1.$$

Then the map

$$\Phi(x) = s + \sum_{r=2}^{d} B_r(x, \dots, x)$$

has a unique fixed point in the closed ball $\overline{B}_R(0)$. The sum of the values of all finite ordered rooted trees converges absolutely to that fixed point.

Proof. For $x, y \in \overline{B}_R(0)$, multilinearity and telescoping in the r input positions give

$$\|B_r(x, \dots, x) - B_r(y, \dots, y)\| \le r b_r R^{r-1} \|x - y\|.$$

Hence Φ is q-Lipschitz on the ball. The inequality $\phi(R) \le R$ shows that Φ preserves the ball. The contraction mapping theorem gives a unique fixed point u in that ball.

Let $\mathfrak{T}_m$ be the finite set of ordered rooted trees of height less than m. A leaf has value s. If the root of τ has arity r and ordered subtrees $\tau_1, \dots, \tau_r$, define

$$B_\tau(s) = B_r(B_{\tau_1}(s), \dots, B_{\tau_r}(s)).$$

Induction on m shows that the Picard iterate $u_0 = 0$ and $u_{m+1} = \Phi(u_m)$ satisfies

$$u_m = \sum_{\tau \in \mathfrak{T}_m} B_\tau(s).$$

Define a nonnegative scalar weight by assigning weight a to a leaf and applying the scalar map $(t_1, \dots, t_r) \mapsto b_r t_1 \cdots t_r$ at a vertex of arity r. A second induction gives

$$\sum_{\tau \in \mathfrak{T}_m} \|B_\tau(s)\| \le \sum_{\tau \in \mathfrak{T}_m} w(\tau) = t_m,$$

where $t_0 = 0$ and $t_{m+1} = \phi(t_m)$. The sequence (t_m) is increasing because ϕ has nonnegative coefficients. It is bounded by R because $t_0 \le R$ and $\phi(R) \le R$. The sum of the norms of all tree values is therefore finite. The tree series converges absolutely. Its partial sums are the Picard iterates, so its sum is u. □

Corollary 18.2 (Quadratic Catalan expansion)**.** *Let $B : X \times X \to X$ be bounded and consider*

$$u = s + B(u, u).$$

Put $a = \|s\|$ and $b = \|B\|$. If $4ab < 1$, then

$$u = \sum_{n \ge 1} \sum_{\substack{\tau \text{ full ordered binary tree} \\ \text{with } n \text{ leaves}}} B_\tau(s)$$

converges absolutely. It also satisfies

$$\|u\| \le \frac{1 - \sqrt{1 - 4ab}}{2b}$$

when $b > 0$.

Proof. If $b = 0$, the equation is $u = s$ and the tree series contains only its one leaf. If $a = 0$, every tree value is zero. Choose any R with $0 < R < 1/(2b)$ and apply theorem 18.1 to obtain uniqueness of the zero fixed point in that ball.

Assume now that $a > 0$ and $b > 0$. Take

$$R = \frac{1 - \sqrt{1 - 4ab}}{2b}.$$

Then $a + bR^2 = R$ and $2bR < 1$. Apply theorem 18.1. The number of full ordered binary trees with n leaves is the Catalan number C_{n-1}, which gives the displayed majorant. □

Theorem 18.3 (Packet labeled tree reconstruction)**.** *Let X carry a packet system $\mathcal{P}$. At each internal vertex of a tree, insert the output identity and one input identity for every child. Summing over all internal packet labels reconstructs the original tree value exactly.*

Proof. At a vertex labeled by an r multilinear map B_r, insert

$$I_X = \sum_{\beta} P_\beta$$

on the output and in every input slot. Multilinearity gives an exact finite sum of packet blocks at that vertex. Apply this identity successively from the leaves toward the root. Since the tree and the packet set are finite, all sums are finite and their total reconstructs the original tree value. □

For an infinite tree expansion, the finite sum of packet decorations for each underlying tree is kept together. The unlabeled tree majorant does not imply absolute convergence when all decorations are summed separately for arbitrary oblique projectors. Finite tree reconstruction and the grouped interpretation remain exact.

> **Arity in a tree**
>
> At each internal vertex, the number of children is the number of input slots of its multilinear map. Their order is retained because symmetry is not assumed. Packet labels resolve each finite tree into a finite sum. In an infinite expansion these sums stay grouped by tree, as required by the convergence statement above.

Corollary 18.4 (Tree pruning)**.** *A packet labeled tree vanishes if one internal vertex carries a zero packet block of the corresponding multilinear map. Removing all such zero terms leaves every finite tree sum unchanged and therefore leaves the grouped infinite series unchanged.*

Proof. A zero vertex map makes the value of the full tree zero. □

The tree theorem separates combinatorial growth from analytic convergence. The number of trees may grow rapidly, while multilinear bounds control their values. The scalar map ϕ controls the total weight of the expansion. Packet pruning changes the combinatorics but not the proof. It removes trees whose values are already zero before any norm estimate is used.

Example 18.5 (Catalan parity packets)**.** Over a field of characteristic zero, the formal equation

$$u = z + u^2$$

has the branch

$$u(z) = \frac{1 - \sqrt{1 - 4z}}{2}.$$

Write $u = u_0 + u_1$, where u_0 contains the even powers of z and u_1 contains the odd powers. Exact packet projection gives

$$u_0 = u_0^2 + u_1^2, \qquad u_1 = z + 2u_0u_1.$$

The two equations remain coupled. Their sum is the original fixed point equation.

19 Graded nonlinear operations

Let G be a finite abelian group and let

$$A = \bigoplus_{g \in G} A_g$$

be a G graded module.

Definition 19.1 (Grade respecting operation)**.** An r multilinear map $\mu : A^r \to A$ respects the grading when

$$\mu(A_{g_1}, \dots, A_{g_r}) \subseteq A_{g_1+\cdots+g_r}.$$

Theorem 19.2 (Graded packet convolution)**.** *If $a_j = \sum_{g\in G} a_{j,g}$ and μ respects the grading, then*

$$P_k\mu(a_1, \dots, a_r) = \sum_{g_1+\cdots+g_r=k} \mu(a_{1,g_1}, \dots, a_{r,g_r}).$$

Proof. Expand every input into its homogeneous components. Each term lands in the component whose label is the sum of its input labels. Projection onto A_k retains exactly the displayed terms. □

Corollary 19.3 (Cyclic product formula)**.** *For a cyclically graded algebra*

$$A = \bigoplus_{j\in\mathbb{Z}/N\mathbb{Z}} A_j, \qquad A_iA_j \subseteq A_{i+j},$$

one has

$$P_k(ab) = \sum_{i+j=k} a_ib_j.$$

Proof. Apply theorem 19.2 to multiplication. □

Theorem 19.4 (Packet Maurer Cartan equations)**.** *Let*

$$\mathcal{L} = \bigoplus_{g\in G} \mathcal{L}_g$$

be a differential graded Lie algebra over a ring in which two is invertible. Assume

$$d\mathcal{L}_g \subseteq \mathcal{L}_g, \qquad [\mathcal{L}_g, \mathcal{L}_h] \subseteq \mathcal{L}_{g+h}.$$

For a degree one element $\eta = \sum_g \eta_g$, the Maurer Cartan equation

$$d\eta + \frac{1}{2}[\eta, \eta] = 0$$

is equivalent to

$$d\eta_k + \frac{1}{2}\sum_{g+h=k} [\eta_g, \eta_h] = 0$$

for every $k \in G$.

Proof. The differential preserves the group label and the bracket adds labels. Apply the projector onto $\mathcal{L}_k$ and use exact reconstruction. □

The same argument applies to higher brackets.

Proposition 19.5 (Packet equations for L_∞ operations)**.** *Let $\mathcal{L}$ be complete and carry operations ℓ_r that respect a finite abelian group grading. Assume the packet projectors are continuous for the topology in which the series below converges. Assume the Maurer Cartan series*

$$\sum_{r\geq 1}\frac{1}{r!}\ell_r(\eta,\dots,\eta)$$

converges and the required factorials are invertible. Then its packet of label k is

$$\sum_{r\geq 1}\frac{1}{r!}\sum_{g_1+\cdots+g_r=k}\ell_r(\eta_{g_1},\dots,\eta_{g_r}).$$

The full Maurer Cartan equation holds if and only if every packet equation vanishes.

Proof. Apply theorem 19.2 to each operation ℓ_r. Continuity of the packet projectors permits the convergent sums to be projected term by term. □

Definition 19.6 (Divided square)**.** A divided square on degree one elements is a map q satisfying

$$q(x+y)=q(x)+q(y)+[x,y],\qquad q(ax)=a^2q(x),$$

whenever the expressions are defined.

Proposition 19.7 (Divided square packet equations)**.** *Assume $q(\mathcal{L}_g)\subseteq\mathcal{L}_{2g}$. Choose an ordering of the finite label set. The equation*

$$d\eta+q(\eta)=0$$

is equivalent to

$$d\eta_k+\sum_{2g=k}q(\eta_g)+\sum_{\substack{g<h\\ g+h=k}}[\eta_g,\eta_h]=0$$

for every k.

Proof. Repeated use of the polarization identity for q gives

$$q\left(\sum_g\eta_g\right)=\sum_g q(\eta_g)+\sum_{g<h}[\eta_g,\eta_h].$$

Project onto the component of label k. □

Example 19.8 (Matrix difference grading)**.** Let $A=M_N(\Bbbk)$ and assign the matrix unit E_{rs} the label $r-s$ modulo N. Since

$$E_{rs}E_{uv}=\delta_{su}E_{rv},$$

the output label is

$$(r-s)+(u-v)=r-v$$

whenever the product is nonzero. Matrix multiplication therefore obeys the cyclic convolution law exactly. The same grading resolves commutators and polynomial matrix equations.

Remark 19.9. These results begin with a graded differential or higher algebra. They do not assert that every deformation problem has such a controlling object. Differential graded and infinity algebra methods are developed in [11].

20 Nonlinear refinement defects

Let

$$F_f : X_f \longrightarrow Y_f, \qquad F_c : X_c \longrightarrow Y_c$$

be fine and coarse maps. Let

$$C_X : X_f \longrightarrow X_c, \qquad C_Y : Y_f \longrightarrow Y_c$$

be linear coarsening maps. Whenever a norm, a Lipschitz constant, or a Fréchet derivative is used, the spaces are Banach spaces and the coarsening maps are bounded.

Definition 20.1 (Nonlinear refinement defect)**.** The pointwise defect of F is

$$\Delta_F(x) = C_Y F_f(x) - F_c(C_X x).$$

The two models are exactly compatible when Δ_F vanishes everywhere.

The defect is a vector in Y_c. It determines an obstruction class after a correction map into Y_c has been chosen and the defect has been passed to the corresponding cokernel. Norm estimates measure the size of the defect. The quotient determines whether an allowed linear correction can remove it.

Theorem 20.2 (Exact defect under composition)**.** *Let $G_f : Y_f \to Z_f$ and $G_c : Y_c \to Z_c$, with coarsening $C_Z : Z_f \to Z_c$. Then*

$$\Delta_{G\circ F}(x) = \Delta_G(F_f(x)) + G_c(F_c(C_X x) + \Delta_F(x)) - G_c(F_c(C_X x)).$$

If G_c is L-Lipschitz on the relevant set, then

$$\|\Delta_{G\circ F}(x)\| \le \|\Delta_G(F_f(x))\| + L\|\Delta_F(x)\|.$$

Proof. Add and subtract $G_c(C_Y F_f(x))$. Use

$$C_Y F_f(x) = F_c(C_X x) + \Delta_F(x).$$

The norm estimate follows from the Lipschitz bound. □

Corollary 20.3 (Linear product rule)**.** *If all maps in theorem 20.2 are linear, then*

$$\Delta_{GF} = \Delta_G F_f + G_c \Delta_F.$$

Proof. Linearity turns the last difference in theorem 20.2 into $G_c\Delta_F(x)$. □

Theorem 20.4 (Iterated orbit defect). *Let $F_f : X_f \to X_f$ and $F_c : X_c \to X_c$, with coarsening $C : X_f \to X_c$. Suppose F_c is q-Lipschitz. For a fine orbit starting at x, assume*

$$\|\Delta_F(F_f^j x)\| \le \varepsilon_j.$$

Then

$$\left\|CF_f^n(x) - F_c^n(Cx)\right\| \le \sum_{j=0}^{n-1} q^{n-1-j}\varepsilon_j.$$

If $0 \le q < 1$ and every $\varepsilon_j \le \varepsilon$, then

$$\left\|CF_f^n(x) - F_c^n(Cx)\right\| \le \varepsilon\frac{1-q^n}{1-q}.$$

Proof. Let

$$e_n = \left\|CF_f^n(x) - F_c^n(Cx)\right\|.$$

The definition of the defect and the Lipschitz property give

$$e_{n+1} \le \varepsilon_n + qe_n.$$

Iterating this recurrence proves the first estimate. The second is the geometric sum. □

Theorem 20.5 (Fixed point stability). *Assume F_c is a contraction with constant $q < 1$. Let x_f and x_c be fixed points of F_f and F_c. If*

$$\|\Delta_F(x_f)\| \le \varepsilon,$$

then

$$\|Cx_f - x_c\| \le \frac{\varepsilon}{1-q}.$$

When the defect at x_f is zero, the fixed points are compatible.

Proof. Using the fixed-point equations,

$$\begin{aligned}\|Cx_f - x_c\| &\le \|CF_f(x_f) - F_c(Cx_f)\| + \|F_c(Cx_f) - F_c(x_c)\| \\ &\le \varepsilon + q\|Cx_f - x_c\|.\end{aligned}$$

Rearrange. □

Corollary 20.6 (Several refinement levels). *Let $C_\ell : X_{\ell+1} \to X_\ell$ for $0 \le \ell < m$. Suppose x_ℓ is a fixed point at level ℓ, the coarse map at level ℓ has contraction constant $q_\ell < 1$, and the adjacent defect at $x_{\ell+1}$ is at most ε_ℓ. Write*

$$C_{a,b} = C_a C_{a+1} \cdots C_{b-1} \quad \text{for } a < b, \qquad C_{a,a} = I.$$

Then

$$\|C_{0,m}x_m - x_0\| \le \sum_{\ell=0}^{m-1} \|C_{0,\ell}\|\frac{\varepsilon_\ell}{1-q_\ell}.$$

Proof. Insert the intermediate vectors $C_{0,\ell}x_\ell$ and telescope. Apply theorem 20.5 to every adjacent pair. □

Proposition 20.7 (Derivative defects). *Assume F_f and F_c are r times Fréchet differentiable and the coarsening maps are bounded and linear. Then*

$$D\Delta_F(x) = C_Y DF_f(x) - DF_c(C_X x)C_X.$$

For $r \geq 2$,

$$D^r\Delta_F(x)[h_1, \dots, h_r] = C_Y D^r F_f(x)[h_1, \dots, h_r] - D^r F_c(C_X x)[C_X h_1, \dots, C_X h_r].$$

Proof. Differentiate the defining identity for Δ_F. The coarsening maps are linear, so their higher derivatives vanish. □

Example 20.8 (Sharp affine bound). Let $F_c(y) = qy + b$ with $0 < q < 1$. Choose a fine fixed point whose coarsening has constant defect ε in the direction of the scalar state. The fixed point mismatch is exactly $\varepsilon/(1-q)$. Thus the constant in theorem 20.5 cannot be improved without more information.

21 Canonical obstruction objects

We begin by isolating the part of a defect that cannot be removed by an allowed linear correction.

Definition 21.1 (Linear correction datum). A linear correction datum consists of a homomorphism

$$D : U \longrightarrow V$$

and a defect $r \in V$. A correction is an element $u \in U$ satisfying

$$Du = -r.$$

The obstruction class is

$$\mathrm{Ob}_D(r) = [r] \in \operatorname{coker} D = V / \operatorname{im} D.$$

Theorem 21.2 (Cokernel obstruction criterion). *A linear correction exists if and only if*

$$\mathrm{Ob}_D(r) = 0.$$

The quotient map

$$q_D : V \longrightarrow \operatorname{coker} D$$

has the following universal property. If $\varphi : V \to W$ is linear and $\varphi D = 0$, then there is a unique linear map

$$\overline{\varphi} : \operatorname{coker} D \longrightarrow W$$

with $\varphi = \overline{\varphi} q_D$.

Proof. The class of r is zero exactly when $r \in \operatorname{im} D$, which is equivalent to solvability of $Du = -r$. If $\varphi D = 0$, then φ vanishes on $\operatorname{im} D$. It therefore descends uniquely to the quotient. □

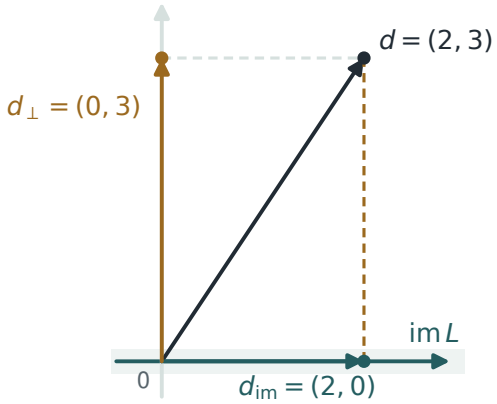
The universal property explains why the cokernel is the canonical obstruction object relative to D. Any linear obstruction test that vanishes on correctable defects factors through it.

Canonical obstruction data

The residual, the correction map, and the obstruction class are distinct data. The residual is a vector. The correction map specifies which changes are allowed, and its cokernel records the remaining class. Canonicality is relative to that declared correction map. A scalar or orthogonal representative of the class may require further choices or hypotheses.

Correctable defect and cokernel residual

$L : \mathbb{R}^2 \to \mathbb{R}^2$, $L(a, b) = (a, 0)$. Defect $d = (2, 3)$.



A chosen scalar detector

$$\ell(x, y) = y$$

$$\ell L = 0$$

$$\ell(d) = 3 \neq 0$$

$$u = (-2, 0): \quad d + Lu = (0, 3) \neq 0.$$

The class $[d] \in \mathbb{R}^2/\mathrm{im}\, L$ is represented by $(0, 3)$.

Standard Euclidean splitting is used. The scalar 3 depends on the displayed quotient identification.

Figure 5. A specified correction range in $\mathbb{R}^2$. For $L(a, b) = (a, 0)$, the image is the horizontal axis. The defect $(2, 3)$ splits into $(2, 0)$ in that image and the remaining representative $(0, 3)$. Subtracting the image component leaves the same cokernel class. The functional $(x, y) \mapsto y$ vanishes on the correction range and detects the value three. The correction equation $Lu = -(2, 3)$ therefore has no solution.

Theorem 21.3 (Naturality of obstruction classes)**.** *Suppose*

$$\begin{array}{ccc} U & \xrightarrow{D} & V \\ a\downarrow & & \downarrow b \\ U' & \xrightarrow{D'} & V' \end{array}$$

is a commuting square. Then there is a unique induced map

$$\overline{b} : \operatorname{coker} D \longrightarrow \operatorname{coker} D'$$

satisfying

$$\overline{b}\,\operatorname{Ob}_D(r) = \operatorname{Ob}_{D'}(br).$$

Proof. The relation $bD = D'a$ shows that the quotient map $q_{D'}b$ vanishes on $\operatorname{im} D$. Apply the universal property in theorem 21.2. □

Theorem 21.4 (Packet decomposition of an obstruction)**.** *Suppose*

$$U = \bigoplus_{\alpha\in I} U_\alpha, \qquad V = \bigoplus_{\alpha\in I} V_\alpha,$$

and $D(U_\alpha) \subseteq V_\alpha$ for every α. Let

$$D_\alpha : U_\alpha \longrightarrow V_\alpha$$

be the restriction. Then

$$\operatorname{coker} D \cong \bigoplus_{\alpha\in I} \operatorname{coker} D_\alpha.$$

Under this isomorphism, the obstruction class of $r = \sum_\alpha r_\alpha$ is the tuple

$$\big([r_\alpha]\big)_\alpha.$$

Hence r is correctable if and only if every packet obstruction vanishes.

Proof. The grading assumption gives

$$\operatorname{im} D = \bigoplus_{\alpha} \operatorname{im} D_\alpha.$$

Taking the quotient of the direct sum by this direct sum of images gives the displayed isomorphism. □

Remark 21.5. The conclusion of theorem 21.4 requires the correction map to preserve the packets. Off-diagonal correction blocks can couple packet defects. In that case the full cokernel remains canonical, but it need not split into the separate packet cokernels.

Proposition 21.6 (Projectors on the obstruction quotient)**.** *Let $(Q_\alpha)_{\alpha\in I}$ be a complete family of pairwise annihilating idempotents on V. Assume*

$$Q_\alpha(\operatorname{im} D) \subseteq \operatorname{im} D$$

for every α. Then each Q_α induces an idempotent $\overline{Q}_\alpha$ on coker D*, and*

$$\operatorname{coker} D = \bigoplus_{\alpha\in I} \overline{Q}_\alpha(\operatorname{coker} D).$$

For every defect $r \in V$,

$$\overline{Q}_\alpha \operatorname{Ob}_D(r) = [Q_\alpha r].$$

No packet decomposition of the correction space U is required.

Proof. If $v - v' \in \operatorname{im} D$, then

$$Q_\alpha(v - v') \in \operatorname{im} D.$$

Hence $[v] \mapsto [Q_\alpha v]$ is well defined on the quotient. The idempotent, annihilation, and reconstruction identities descend from V to $V/\operatorname{im} D$. The formula for an obstruction class follows from the definition. □

21.1 Relative obstruction under coarsening

Coarsening can erase an obstruction that remains present in the fine problem. The next exact sequence identifies the part removed by coarsening.

Theorem 21.7 (Relative obstruction exact sequence)**.** *Suppose the following diagram of modules commutes and has exact rows:*

$$\begin{array}{ccccccccc} 0 & \longrightarrow & U_0 & \xrightarrow{i_U} & U_f & \xrightarrow{p_U} & U_c & \longrightarrow & 0 \\ & & D_0 \downarrow & & D_f \downarrow & & D_c \downarrow & & \\ 0 & \longrightarrow & V_0 & \xrightarrow{i_V} & V_f & \xrightarrow{p_V} & V_c & \longrightarrow & 0. \end{array}$$

There is a natural exact sequence

$$\begin{aligned} 0 \longrightarrow \operatorname{Ker} D_0 \longrightarrow \operatorname{Ker} D_f \longrightarrow \operatorname{Ker} D_c \xrightarrow{\delta} \operatorname{coker} D_0 \\ \longrightarrow \operatorname{coker} D_f \longrightarrow \operatorname{coker} D_c \longrightarrow 0. \end{aligned}$$

For $u_c \in \operatorname{Ker} D_c$, choose $u_f \in U_f$ with $p_U u_f = u_c$. There is a unique $v_0 \in V_0$ such that

$$D_f u_f = i_V v_0,$$

and the connecting map is

$$\delta(u_c) = [v_0].$$

Proof. Since $D_c u_c = 0$, commutativity gives

$$p_V D_f u_f = D_c p_U u_f = 0.$$

Exactness of the lower row gives a unique v_0 with $D_f u_f = i_V v_0$. A different lift has the form $u_f + i_U u_0$. The corresponding element of V_0 changes by $D_0 u_0$, so $[v_0]$ is independent of the lift.

The map $\operatorname{Ker} D_0 \to \operatorname{Ker} D_f$ is injective because i_U is injective. If $u_f \in \operatorname{Ker} D_f$ maps to zero in U_c, then $u_f = i_U u_0$ for a unique u_0. The identity

$$i_V D_0 u_0 = D_f i_U u_0 = 0$$

and injectivity of i_V show that $u_0 \in \operatorname{Ker} D_0$. This proves exactness at $\operatorname{Ker} D_f$.

If u_c comes from an element of $\operatorname{Ker} D_f$, one may choose that element as its lift and obtain $\delta(u_c) = 0$. Conversely, if $\delta(u_c) = 0$, then $v_0 = D_0 u_0$ for some u_0. Replacing u_f by $u_f - i_U u_0$ gives an element of $\operatorname{Ker} D_f$ that maps to u_c. This proves exactness at $\operatorname{Ker} D_c$.

The class $[v_0] \in \operatorname{coker} D_0$ maps to zero in $\operatorname{coker} D_f$ whenever it equals $\delta(u_c)$, since the chosen lift satisfies $i_V v_0 = D_f u_f$. Conversely, if $i_V v_0 = D_f u_f$, then $u_c = p_U u_f$ lies in $\operatorname{Ker} D_c$ and $\delta(u_c) = [v_0]$. This proves exactness at $\operatorname{coker} D_0$.

If a class in $\operatorname{coker} D_f$ comes from $\operatorname{coker} D_0$, its image in $\operatorname{coker} D_c$ is zero. Conversely, suppose $[v_f]$ maps to zero in $\operatorname{coker} D_c$. Choose u_c with $D_c u_c = p_V v_f$ and lift it to $u_f \in U_f$. Then

$$p_V(v_f - D_f u_f) = 0,$$

so $v_f - D_f u_f = i_V v_0$ for some $v_0 \in V_0$. Thus $[v_f]$ comes from $[v_0]$. Surjectivity at the last term follows from surjectivity of p_V. □

Corollary 21.8 (Coarse vanishing and fine vanishing)**.** *Let $[r_f] \in \operatorname{coker} D_f$. If its image in $\operatorname{coker} D_c$ is zero, then $[r_f]$ comes from $\operatorname{coker} D_0$. Coarse vanishing therefore means that the fine obstruction is supported in information removed by coarsening. It does not imply $[r_f] = 0$.*

Proof. This is exactness of the sequence in theorem 21.7 at $\operatorname{coker} D_f$. □

Example 21.9 (A fine obstruction erased by coarsening)**.** Take all three correction spaces to be zero. Let

$$V_f = \Bbbk^2, \qquad V_c = \Bbbk, \qquad p_V(x, y) = x + y,$$

and let

$$V_0 = \operatorname{Ker} p_V = \operatorname{span}\{(1, -1)\}.$$

The fine defect $(1, -1)$ is a nonzero class in $\operatorname{coker} D_f = V_f$, while its coarse image is zero. The class comes from $\operatorname{coker} D_0 = V_0$, exactly as corollary 21.8 predicts.

The same construction has a cohomological form.

Definition 21.10 (Mapping cone)**.** Let $D : C^\bullet \to E^\bullet$ be a cochain map. Its mapping cone is

$$\operatorname{Cone}(D)^n = E^n \oplus C^{n+1}$$

with differential

$$d_{\operatorname{Cone}}(e, c) = (d_E e + Dc, -d_C c).$$

Theorem 21.11 (Cohomological obstruction through the cone)**.** *Let $e \in E^n$ satisfy $d_E e = 0$. The class of $(e, 0)$ in*

$$H^n(\operatorname{Cone}(D))$$

vanishes if and only if the class $[e] \in H^n(E)$ lies in the image of

$$H^n(D) : H^n(C) \longrightarrow H^n(E).$$

When C and E are concentrated in degree zero,

$$H^0(\mathrm{Cone}(D)) \cong \operatorname{coker} D.$$

Proof. The class of $(e, 0)$ vanishes exactly when there are $e' \in E^{n-1}$ and $c \in C^n$ with

$$(e, 0) = d_{\mathrm{Cone}}(e', c).$$

This means

$$e = d_E e' + Dc, \qquad d_C c = 0.$$

Equivalently, $[e] = H^n(D)[c]$. The degree zero statement follows directly from the definition of the cone differential. □

Corollary 21.12 (Packet obstruction complexes)**.** *If $C^\bullet$ and $E^\bullet$ split into finite packet subcomplexes and D preserves those packets, then* $\mathrm{Cone}(D)$ *splits into packet subcomplexes and*

$$H^n(\mathrm{Cone}(D)) \cong \bigoplus_\alpha H^n(\mathrm{Cone}(D_\alpha)).$$

Proof. Both the cone and its cohomology commute with finite direct sums. □

Example 21.13 (A two component obstruction)**.** Let

$$D : \Bbbk \longrightarrow \Bbbk^2, \qquad D(t) = (t, t).$$

The cokernel is one dimensional. The defect (a, b) is correctable exactly when $a = b$. The obstruction can be represented by $a - b$, but this scalar representative depends on a choice of isomorphism from the cokernel to $\Bbbk$. The quotient class itself is canonical.

22 Split Hilbert obstructions and nonlinear correction

The quotient obstruction has a concrete orthogonal representative when the correction map acts between Hilbert spaces and has closed range. The Hilbert and Banach space results below are analytic extensions of the finite calculus. They use the stated analytic hypotheses and lie outside the finite formal contract.

Theorem 22.1 (Orthogonal obstruction projector)**.** *Let $D : H_1 \to H_2$ be a bounded operator between Hilbert spaces with closed range. Let $D^\dagger$ be its Moore–Penrose inverse. Then*

$$P_{\mathrm{ob}} = I_{H_2} - DD^\dagger$$

is the orthogonal projector onto

$$(\operatorname{im} D)^\perp = \operatorname{Ker} D^*.$$

For $r \in H_2$, the following are equivalent:

1. *$Du = -r$ is solvable*
2. *$[r] = 0$ in* $\operatorname{coker} D$

3. $P_{\mathrm{ob}} r = 0$.

When these conditions hold, $-D^{\dagger} r$ is the least norm correction. For every r,

$$\operatorname{dist}(r, \operatorname{im} D) = \|P_{\mathrm{ob}} r\|.$$

Proof. Closed range gives the orthogonal decomposition

$$H_2 = \operatorname{im} D \oplus \operatorname{Ker} D^{*}.$$

The Moore–Penrose identities show that $DD^{\dagger}$ is the orthogonal projector onto $\operatorname{im} D$. The stated equivalences and distance formula follow from this decomposition. The least norm property is the standard characterization of $D^{\dagger} r$. □

Once the defect lies in the range, nonlinear correction still requires control of the nonlinear term. The next theorem gives that control through a bounded right inverse and a contraction estimate.

Theorem 22.2 (Right inverse correction theorem)**.** *Let $D : X \to Y$ be a bounded linear map between Banach spaces. Assume $Z = \operatorname{im} D$ is closed and there is a bounded linear right inverse*

$$R : Z \longrightarrow X, \qquad DR = I_Z.$$

Put $X_R = R(Z)$. Let

$$N : \overline{B}_{\rho}(0) \cap X_R \longrightarrow Z$$

be L-Lipschitz. Let $y \in Z$ and put

$$q = \|R\| L.$$

Assume $q < 1$ and

$$\|R(y - N(0))\| \leq (1 - q)\rho.$$

Then the equation

$$Dx + N(x) = y$$

has a unique solution in $\overline{B}_{\rho}(0) \cap X_R$.

Proof. The space X_R is closed. Indeed, if $Rz_n \to x$, then $z_n = DRz_n \to Dx$, so $x = R(Dx)$. Define

$$\Psi(x) = R(y - N(x)).$$

This map takes X_R to itself. For $x \in \overline{B}_{\rho}(0) \cap X_R$,

$$\begin{aligned} \|\Psi(x)\| &\leq \|R(y - N(0))\| + \|R\| \, \|N(x) - N(0)\| \\ &\leq (1 - q)\rho + q\rho = \rho. \end{aligned}$$

Also,

$$\|\Psi(x) - \Psi(x')\| \leq q \|x - x'\|.$$

The contraction mapping theorem gives a unique fixed point in the stated set. Applying D to the fixed point identity gives the equation. Conversely, a solution in X_R equals $R(Dx)$ and is therefore a fixed point of Ψ. □

The hypotheses behind correction

Closed range gives the obstruction an orthogonal representative in the Hilbert setting. A bounded right inverse chooses a correction slice. In the nonlinear theorem, the target and the nonlinear term both lie in the range, and the Lipschitz bound makes the correction map a contraction on the chosen ball. The uniqueness statement concerns the chosen ball within that correction slice.

Corollary 22.3 (Stability with respect to the target)**.** *Under the hypotheses of theorem 22.2, let x and x' be the fixed points associated with targets y and y'. If both fixed point maps preserve the same ball, then*

$$\|x - x'\| \le \frac{\|R\|}{1-q}\|y - y'\|.$$

Proof. Subtract the two fixed-point equations and use the Lipschitz bound for N:

$$\|x - x'\| \le \|R\|\|y - y'\| + q\|x - x'\|.$$

□

Lyapunov–Schmidt reduction separates the equation into two parts. The implicit function theorem solves the range component, leaving an equation in the obstruction space. Standard accounts of this reduction appear in [9, 19].

Theorem 22.4 (Local Lyapunov–Schmidt reduction)**.** *Let $\mathcal{U} \subseteq X$ be an open neighborhood of zero and let $F : \mathcal{U} \to Y$ be a C^r map between Banach spaces with $r \ge 1$, $F(0) = 0$, and $L = DF(0)$. Assume there are closed decompositions*

$$X = K \oplus X_1, \qquad Y = R \oplus O,$$

where $K = \operatorname{Ker} L$, $R = \operatorname{im} L$, and

$$L_1 = L|_{X_1} : X_1 \longrightarrow R$$

is a Banach space isomorphism. Let π_R and π_O be the two projections on Y. Then there are neighborhoods $U_K \subseteq K$ and $U_1 \subseteq X_1$ of zero and a unique C^r map

$$\psi : U_K \longrightarrow U_1$$

with $\psi(0) = 0$ such that

$$\pi_R F(k + \psi(k)) = 0$$

for every $k \in U_K$. After the neighborhoods are made smaller if needed, the solutions of $F(x) = 0$ near zero are exactly the points

$$x = k + \psi(k)$$

for which the reduced obstruction map

$$\kappa(k) = \pi_O F(k + \psi(k))$$

vanishes. At the base point,

$$D\psi(0) = 0, \qquad D\kappa(0) = 0.$$

Proof. Define

$$G(k, x_1) = \pi_R F(k + x_1).$$

The derivative of G in the X_1 variable at $(0, 0)$ is L_1, which is invertible. The implicit function theorem gives the unique map ψ. Every nearby x has a unique decomposition $x = k + x_1$. The range equation forces $x_1 = \psi(k)$, and the remaining equation is precisely $\kappa(k) = 0$.

Differentiate $G(k, \psi(k)) = 0$ at zero. Since L vanishes on K, one obtains $L_1 D\psi(0) = 0$, hence $D\psi(0) = 0$. Finally,

$$D\kappa(0) = \pi_O L|_K = 0.$$

□

Corollary 22.5 (Quadratic reduced obstruction)**.** *Assume the hypotheses of theorem 22.4 with $r \geq 2$. For $k_1, k_2 \in K$,*

$$D^2\kappa(0)[k_1, k_2] = \pi_O D^2 F(0)[k_1, k_2].$$

Proof. Differentiate $\kappa(k) = \pi_O F(k+\psi(k))$ twice. The terms containing $D\psi(0)$ vanish. The term containing $D^2\psi(0)$ is multiplied by $\pi_O L$, which is zero because $L(X) = R$. □

Example 22.6 (A purely quadratic obstruction)**.** Let

$$F(x, y) = (y, x^2)$$

on $\mathbb{R}^2$. At the origin,

$$L(h, k) = (k, 0).$$

The kernel is the x axis, the range is the first output axis, and the cokernel is the second output axis. The range equation forces $y = 0$. The reduced obstruction map is

$$\kappa(x) = x^2.$$

The linear obstruction vanishes for every kernel direction, but the nonlinear equation still has only the solution $x = 0$ near the origin.

Remark 22.7. The reduced map κ begins at quadratic order. This is why vanishing of the first-order cokernel class need not settle a nonlinear problem. Higher terms can produce a genuine obstruction on the kernel directions.

23 Filtered lifting

Let X and Y be modules with decreasing filtrations

$$X = F^0X \supseteq F^1X \supseteq \cdots, \qquad Y = F^0Y \supseteq F^1Y \supseteq \cdots.$$

Assume both are separated and complete for their filtration topologies. Let $\Phi : X \to Y$ be continuous and filtration congruence preserving in the sense that

$$h \in F^mX \quad \Longrightarrow \quad \Phi(x + h) - \Phi(x) \in F^mY$$

for every $x \in X$ and every m. This condition makes the statement $\Phi(x) \in F^m Y$ depend only on the class of x modulo $F^m X$.

Suppose x_m satisfies

$$\Phi(x_m) \in F^m Y.$$

Assume that corrections $h \in F^m X$ obey a first-order congruence

$$\Phi(x_m + h) \equiv \Phi(x_m) + L_m h \pmod{F^{m+1} Y},$$

where $L_m : F^m X \to F^m Y$ is a module homomorphism satisfying

$$L_m(F^{m+1} X) \subseteq F^{m+1} Y$$

and therefore induces

$$\overline{L}_m : F^m X / F^{m+1} X \longrightarrow F^m Y / F^{m+1} Y.$$

At depth m, only the associated graded correction is visible. The cokernel below has finite length whenever the target graded piece has finite length, even if the completed problem is infinite. Vanishing of the depth obstruction decides whether the current approximation can be improved by one level. When the graded pieces are finite, the branching count below applies. Finite length alone does not imply finite cardinality.

Definition 23.1 (Depth obstruction)**.** The depth m obstruction of x_m is

$$\mathrm{Ob}_m(x_m) = [\Phi(x_m)] \in \operatorname{coker} \overline{L}_m,$$

where $\Phi(x_m)$ is first viewed in $F^m Y / F^{m+1} Y$.

Theorem 23.2 (One-step lifting criterion)**.** *The approximation x_m has a correction $h_m \in F^m X$ such that*

$$\Phi(x_m + h_m) \in F^{m+1} Y$$

if and only if

$$\mathrm{Ob}_m(x_m) = 0.$$

When a correction exists, the set of its classes in

$$F^m X / F^{m+1} X$$

is a coset of $\operatorname{Ker} \overline{L}_m$*. If the graded pieces are vector spaces, this coset is an affine space.*

Proof. The next depth condition is equivalent to

$$\overline{L}_m[h_m] = -[\Phi(x_m)]$$

in $F^m Y / F^{m+1} Y$. This is solvable exactly when the displayed defect has zero cokernel class. If h_m and h'_m are two solutions, their difference lies in $\operatorname{Ker} \overline{L}_m$. Adding an element of the kernel to one solution gives another, so the solution set is a coset of the kernel. □

Corollary 23.3 (Finite branching count)**.** *Suppose the graded pieces are finite vector spaces over a finite field. If the depth obstruction vanishes, then the number of correction classes at depth m is*

$$|\operatorname{Ker}\overline{L}_m|.$$

If the obstruction does not vanish, the number is zero.

Proof. A nonempty affine space over a finite vector space has the same number of elements as that vector space. □

Lemma 23.4 (Well-defined lift classes)**.** *Under the filtration congruence assumption, the condition*

$$\Phi(x) \in F^m Y$$

depends only on the class of x in $X/F^m X$. Fix a representative x_m of a depth m class. The map

$$[h] \longmapsto [x_m + h]_{m+1}$$

identifies correction classes $[h] \in F^m X/F^{m+1}X$ with the classes at depth $m+1$ that reduce to $[x_m]_m$. Under this identification, the solutions of the associated graded correction equation in theorem 23.2 are exactly the children whose defect lies in $F^{m+1}Y$. A different representative of the same parent gives a translated parametrization of the same set of children.

Proof. If $x' - x \in F^m X$, then filtration congruence preservation gives

$$\Phi(x') - \Phi(x) \in F^m Y.$$

Thus one defect lies in $F^m Y$ exactly when the other does. If $h - h' \in F^{m+1}X$, then $x_m + h$ and $x_m + h'$ define the same depth $m+1$ class. Conversely, two depth $m+1$ classes reducing to $[x_m]_m$ differ from x_m by elements of $F^m X$, and they are equal precisely when those elements agree modulo $F^{m+1}X$. The first-order congruence shows that such a child has defect in $F^{m+1}Y$ exactly when its correction class solves the associated graded equation. Replacing x_m by another representative translates the correction coordinate by an element of $F^m X/F^{m+1}X$ and leaves the child classes themselves unchanged. □

Definition 23.5 (Lift tree)**.** Fix an initial approximation. A vertex at depth m is a class $[x]_m \in X/F^m X$ with $\Phi(x) \in F^m Y$ that reduces to the fixed initial class. A depth $m+1$ vertex is joined to its reduction in $X/F^m X$. By lemma 23.4, this rooted graph is independent of the chosen representatives. It is called the lift tree.

Theorem 23.6 (Finitely branching lift tree)**.** *Assume every level of the lift tree is nonempty and every vertex has only finitely many children. Then the tree contains an infinite branch. If X and Y are separated and complete and Φ is continuous, that branch determines an exact solution of $\Phi(x) = 0$.*

Proof. The first assertion is König's infinity argument. The root has finitely many children. Since vertices occur at arbitrarily large depths, at least one child has descendants at arbitrarily large depths. Choose such a child and repeat the argument at every chosen vertex. This gives one compatible vertex at each depth.

Choose representatives recursively. Once x_m represents the selected depth m vertex, choose a representative x_{m+1} of its selected child with

$$x_{m+1} - x_m \in F^m X.$$

The resulting sequence is Cauchy for the filtration topology. Each representative satisfies $\Phi(x_m) \in F^m Y$. The conclusion now follows from theorem 23.8. □

Corollary 23.7 (Unique lift branch)**.** *Suppose an initial approximation is fixed, every depth obstruction vanishes along its descendants, and every map $\overline{L}_m$ has zero kernel. Then at most one compatible lift branch extends the initial approximation.*

Proof. By theorem 23.2, the correction classes at each depth form a coset of $\operatorname{Ker} \overline{L}_m$. A zero kernel leaves at most one correction class, so induction gives at most one branch. □

Theorem 23.8 (Complete filtered lifting)**.** *Assume there is a compatible sequence*

$$x_{m+1} \equiv x_m \pmod{F^m X}$$

with

$$\Phi(x_m) \in F^m Y$$

for every m. Then x_m converges to an element $x_\infty \in X$ and

$$\Phi(x_\infty) = 0.$$

In particular, if the obstruction vanishes at every depth and compatible corrections are chosen, an exact solution exists.

Proof. For every fixed r, the congruences imply

$$x_n - x_m \in F^r X$$

whenever $n \geq m \geq r$. Thus (x_m) is Cauchy for the filtration topology. Completeness gives a limit x_∞.

Fix r. For every $m \geq r$, one has $\Phi(x_m) \in F^m Y \subseteq F^r Y$. The submodule $F^r Y$ is open and therefore closed in the filtration topology. Continuity of Φ gives $\Phi(x_m) \to \Phi(x_\infty)$, so $\Phi(x_\infty) \in F^r Y$. Since this holds for every r and Y is separated,

$$\Phi(x_\infty) \in \bigcap_r F^r Y = 0.$$

□

Theorem 23.9 (Naturality of filtered obstructions)**.** *Suppose a morphism of filtered lifting problems sends x_m to x'_m, intertwines L_m with L'_m, and sends $\Phi(x_m)$ to $\Phi'(x'_m)$ on the associated graded pieces. Then the induced cokernel map sends*

$$\mathrm{Ob}_m(x_m) \quad \text{to} \quad \mathrm{Ob}'_m(x'_m).$$

Proof. Apply theorem 21.3 to the induced commuting square on the depth m graded pieces. □

Example 23.10 (Three lifting outcomes)**.** The scalar equation $x^2-2=0$ at $x=3$ modulo 7 has invertible derivative. It has one correction modulo 49 and gives $x=10$ modulo 49.

For $x^2-3=0$ at $x=0$ modulo 3, the derivative is zero and the defect class is nonzero. No correction exists.

For $x^2=0$ at $x=0$ modulo 3, the derivative and defect both vanish. Every correction digit lies in the kernel, so the lift branches.

Proposition 23.11 (Maurer Cartan lifting obstruction)**.** *Let $\mathcal{L}$ be a complete filtered differential graded Lie algebra over a ring in which two is invertible. Assume*

$$dF^i\mathcal{L} \subseteq F^i\mathcal{L}, \qquad [F^i\mathcal{L}, F^j\mathcal{L}] \subseteq F^{i+j}\mathcal{L}.$$

For a degree one element η, put

$$\Omega(\eta) = d\eta + \frac{1}{2}[\eta,\eta].$$

Suppose $\Omega(\eta) \in F^m\mathcal{L}^2$ with $m \geq 1$. For every $h \in F^m\mathcal{L}^1$,

$$\Omega(\eta+h) \equiv \Omega(\eta) + d_\eta h \pmod{F^{m+1}\mathcal{L}^2}, \qquad d_\eta = d + [\eta,\cdot].$$

The obstruction to improving η by one depth is therefore the class of $\Omega(\eta)$ in the cokernel of

$$\mathrm{gr}^m(d_\eta) : \mathrm{gr}^m\,\mathcal{L}^1 \longrightarrow \mathrm{gr}^m\,\mathcal{L}^2.$$

Proof. Direct expansion gives

$$\Omega(\eta+h) = \Omega(\eta) + dh + [\eta,h] + \frac{1}{2}[h,h].$$

The filtration rule places $[h,h]$ in $F^{2m}\mathcal{L}^2$, which is contained in $F^{m+1}\mathcal{L}^2$ because $m \geq 1$. The congruence follows. The last statement is theorem 23.2 applied to the induced graded correction map. □

24 Filtered obstruction complexes

A direct packet decomposition splits a complex into independent pieces, so cohomology splits packet by packet. A filtration retains relations between depths. Its associated graded cohomology can contain classes that disappear on later pages of a spectral sequence.

There are therefore two ways for fine obstruction data to give zero total obstruction. Each fine class may already vanish, or later differentials may remove nonzero classes through compatibility between depths. Only the first mechanism occurs in a direct sum of subcomplexes. A filtration can exhibit both.

Standard Fact 24.1 (Spectral sequence of a finite filtered complex)**.** Let $C^\bullet$ be a cochain complex with a finite decreasing filtration by subcomplexes

$$C^\bullet = F^0C^\bullet \supseteq F^1C^\bullet \supseteq \cdots \supseteq F^{N+1}C^\bullet = 0.$$

There is a natural spectral sequence with

$$E_0^{p,q} = \mathrm{gr}^p\,C^{p+q}, \qquad E_1^{p,q} = H^{p+q}(\mathrm{gr}^p\,C^\bullet),$$

which converges to the associated graded object of $H^{p+q}(C^\bullet)$ with its induced filtration. A filtered cochain map induces a morphism of spectral sequences.

This is the standard construction for a filtered complex, in the form given in [18, 12]. When $C^\bullet$ is a correction or deformation complex, the groups on its pages can be read as graded obstruction candidates. For an arbitrary filtered complex, they are cohomology classes of the associated graded problem.

> **Lifting and survival answer different questions**
>
> A depth obstruction tests whether a chosen approximation admits one further correction. A spectral sequence follows cohomology classes of a filtered complex through later differentials. Turning successive corrections into an exact solution also uses compatible choices, completeness, continuity, and separatedness.

Proposition 24.2 (Fine obstruction survival)**.** *A class on the first page contributes to the total obstruction only if it is represented by a permanent cycle, so it is neither killed nor identified as a boundary on any later page. If the spectral sequence degenerates at the first page, then $H^n(C^\bullet) = 0$ exactly when $E_1^{p,n-p} = 0$ for every p.*

Proof. Convergence identifies the last page with the associated graded object of the total cohomology. A first page class contributes to that object exactly when it determines a nonzero class on the last page. If the sequence degenerates at the first page, the associated graded pieces of $H^n(C^\bullet)$ are the groups $E_1^{p,n-p}$. A finite filtered module is zero exactly when all its associated graded pieces are zero. □

Corollary 24.3 (Direct packet complexes)**.** *Suppose*

$$C^\bullet = \bigoplus_{\alpha \in I} C^\bullet_\alpha$$

is a finite direct sum of subcomplexes. Then

$$H^n(C^\bullet) \cong \bigoplus_{\alpha \in I} H^n(C^\bullet_\alpha).$$

A total cohomology class vanishes if and only if every packet class vanishes.

Proof. The differential acts independently on the summands. Kernels and images therefore split as finite direct sums, and so does their quotient. □

Example 24.4 (A fine class killed at the next page)**.** Let $C^0 = \Bbbk x$ and $C^1 = \Bbbk y$, with differential $d(x) = y$. Give C^0 the filtration

$$F^0C^0 = \Bbbk x, \qquad F^1C^0 = 0,$$

and give C^1 the filtration

$$F^0C^1 = F^1C^1 = \Bbbk y, \qquad F^2C^1 = 0.$$

The associated graded differential is zero in the relevant positions, so both x and y give nonzero classes on the next page. The next differential sends the class of x to the class of y. The total complex is acyclic. Thus nonzero fine obstruction classes can cancel through a later refinement differential.

Remark 24.5. The spectral sequence does not guarantee that a total obstruction vanishes. It records which fine classes survive and which are removed by compatibility across depths.

25 Nonlinear elimination

Let U, V, Y_P, Y_Q be Banach spaces. On an open neighborhood of $(u_0, v_0) \in U \times V$, split the variables and equations as

$$x = (u, v), \qquad F(u, v) = \begin{pmatrix} F_P(u, v) \\ F_Q(u, v) \end{pmatrix}.$$

Assume F takes values in $Y_P \times Y_Q$, is twice continuously Fréchet differentiable near (u_0, v_0), satisfies $F_Q(u_0, v_0) = 0$, and

$$A = D_v F_Q(u_0, v_0)$$

is a Banach space isomorphism from V to Y_Q. The implicit function theorem gives a local map $v = \varphi(u)$, with $\varphi(u_0) = v_0$, satisfying

$$F_Q(u, \varphi(u)) = 0.$$

Define the reduced equation

$$R(u) = F_P(u, \varphi(u)).$$

Theorem 25.1 (Exact local elimination). *After the neighborhoods are chosen small enough, the equation*

$$F(u, v) = 0$$

is equivalent to

$$v = \varphi(u) \qquad \textit{and} \qquad R(u) = 0.$$

Thus eliminating the hidden variable loses no local solutions.

Proof. The hidden equation $F_Q(u, v) = 0$ has the unique local solution $v = \varphi(u)$ by the implicit function theorem. After this identity is imposed, the remaining component of F is exactly $R(u)$. □

Theorem 25.2 (Nonlinear Schur derivative). *At (u_0, v_0),*

$$DR = D_u F_P - D_v F_P A^{-1} D_u F_Q.$$

Proof. Differentiate $F_Q(u, \varphi(u)) = 0$:

$$D_u F_Q + A D\varphi = 0.$$

Thus

$$D\varphi = -A^{-1} D_u F_Q.$$

Substitute this into

$$DR = D_u F_P + D_v F_P D\varphi.$$

□

The hidden variable still contributes

The reduced derivative includes the response of the hidden variable to a retained variation. That response travels through the inverse hidden derivative and then back into the retained equation.

The Schur term records this contribution. Its use depends on the hidden derivative being invertible, as required in the local elimination setup.

The second derivative records how this response enters the first nonlinear feedback from the hidden variables.

Theorem 25.3 (Second Schur derivative)**.** *For retained directions h, k, set*

$$\hat{h} = (h, D\varphi h), \qquad \hat{k} = (k, D\varphi k).$$

Then

$$D^2\varphi[h,k] = -A^{-1}D^2F_Q[\hat{h},\hat{k}]$$

and

$$D^2R[h,k] = D^2F_P[\hat{h},\hat{k}] - D_vF_PA^{-1}D^2F_Q[\hat{h},\hat{k}].$$

All derivatives on the right are evaluated at (u_0, v_0).

Proof. Differentiate the identity $F_Q(u, \varphi(u)) = 0$ twice. The chain rule gives

$$D^2F_Q[\hat{h},\hat{k}] + AD^2\varphi[h,k] = 0.$$

This proves the first formula. Differentiate $R(u) = F_P(u, \varphi(u))$ twice and substitute the first formula. □

Corollary 25.4 (Packet jets of the reduced equation)**.** *If the retained and hidden spaces carry packet systems, then the first and second packet jets of R are obtained by projecting the formulas in theorems 25.2 and 25.3. Hidden packet feedback is carried by the factors involving A^{-1}.*

Proof. Apply the packet projectors to the input and output slots of the two derivative formulas. □

Example 25.5 (Two retained variables and one hidden variable)**.** Suppose the derivative of F at a solution has block form

$$DF = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where D is invertible. Eliminating the hidden variable replaces the retained derivative A by

$$A - BD^{-1}C.$$

The feedback term is exact. Keeping only A would ignore the path from retained variables into the hidden variable and back.

Remark 25.6. Elimination and obstruction reduction address different situations. The Schur formula assumes that the hidden derivative is invertible. Lyapunov–Schmidt reduction allows a nontrivial kernel and cokernel, then leaves a reduced obstruction equation on the kernel directions.

26 Finite transport and holonomy

Let $\Gamma = (V, E)$ be a finite connected graph and fix a vector space X. At every vertex take one labeled copy of X. Every oriented edge $u \to v$ carries an invertible transport

$$T_{vu} \in \mathrm{GL}(X), \qquad T_{uv} = T_{vu}^{-1}.$$

Equivalently, one may begin with separate but isomorphic vertex fibers and choose one frame in each fiber. The matrices below are then the transports in those frames. For a path $\gamma = (v_0, \dots, v_r)$, set

$$T_\gamma = T_{v_r v_{r-1}} \cdots T_{v_1 v_0}.$$

For a loop, this product is its holonomy. In the based loop group, $\gamma_2\gamma_1$ means first traverse γ_1 and then γ_2, so $T_{\gamma_2\gamma_1} = T_{\gamma_2} T_{\gamma_1}$. Standard graph terminology may be found in [5].

An invertible map on each edge supplies the local transport data. A global frame must make these maps agree around every cycle. A spanning tree has no cycles, so a gauge can make every tree transport the identity. Each remaining edge closes a loop and retains that loop's holonomy.

Definition 26.1 (Gauge change)**.** A gauge is a family $G_v \in \mathrm{GL}(X)$. It changes the edge transports to

$$T'_{vu} = G_v T_{vu} G_u^{-1}.$$

Proposition 26.2 (Gauge covariance)**.** *For a loop γ based at v_0,*

$$\mathrm{Hol}'(\gamma) = G_{v_0} \mathrm{Hol}(\gamma) G_{v_0}^{-1}.$$

Thus the conjugacy class of each loop holonomy is gauge invariant.

Proof. Every intermediate gauge factor cancels with its inverse. The initial and terminal factors remain at the base vertex. □

Theorem 26.3 (Spanning tree normalization)**.** *Choose a root and a spanning tree $\mathcal{T}$ of Γ. There is a gauge in which every tree edge transport is the identity. In that gauge, the transport on each edge outside the tree is the holonomy of its fundamental cycle, with the orientation used to define that cycle.*

Proof. Set the gauge at the root to the identity. Propagate gauges along the unique tree paths so that every tree edge becomes the identity. An edge outside the tree closes one fundamental cycle. Since every tree factor is now the identity, the remaining edge transport is the cycle product. □

Theorem 26.4 (Cycle gluing criterion)**.** *There are vertex coordinates in which every edge transport is the identity if and only if every fundamental cycle holonomy is the identity.*

Proof. If all edge transports are identity maps, every loop product is the identity. Conversely, use theorem 26.3. Identity fundamental holonomy forces every edge outside the tree to become the identity as well. □

Corollary 26.5 (Number of independent cycle tests)**.** *For a connected graph with $|V|$ vertices and $|E|$ unoriented edges, a spanning tree leaves*

$$|E| - |V| + 1$$

fundamental cycle holonomies.

Proof. A spanning tree has $|V| - 1$ edges. Every remaining edge creates one fundamental cycle. □

26.1 Intertwining two transport systems

Let A be a transport system on copies of a fixed vector space V_0, and let B be a transport system on copies of a fixed vector space W_0. Fix a root r and an isomorphism

$$U_r : V_0 \longrightarrow W_0.$$

Theorem 26.6 (Fundamental holonomy intertwining criterion)**.** *The following conditions are equivalent.*

1. *There is a family of isomorphisms $U_v : V_0 \to W_0$ with the prescribed root value U_r and satisfying*

$$U_v A_{vu} = B_{vu} U_u$$

   for every oriented edge $u \to v$.

2. *For every fundamental loop γ based at r,*

$$U_r \operatorname{Hol}_A(\gamma) = \operatorname{Hol}_B(\gamma) U_r.$$

3. *The same identity holds for every loop based at r.*

When these conditions hold, the family (U_v) is uniquely determined by U_r.

Proof. If the edge intertwiners exist, multiply their equations around a loop. Every intermediate vertex map cancels. This proves that the first condition implies the third, and the third implies the second.

Conversely, choose a spanning tree. Starting from U_r, propagate U_v along the unique tree path by the rule

$$U_v = B_{vu} U_u A_{vu}^{-1}$$

on every tree edge. This is the only possible extension satisfying the tree edge equations. An edge outside the tree closes one fundamental loop. Its intertwining equation holds exactly when U_r intertwines the two corresponding fundamental holonomies. The second condition therefore supplies every remaining edge equation. Uniqueness follows from propagation along the tree. □

Corollary 26.7 (Scalar holonomy bridge)**.** *Suppose every vertex space is one dimensional. Two transport systems are gauge equivalent if and only if their holonomies agree on a fundamental cycle basis.*

Proof. Choose a nonzero root isomorphism. Scalar loop maps commute with it, so theorem 26.6 reduces to equality of the cycle scalars. □

Theorem 26.8 (Gauge classification of transports)**.** *Let $G \subseteq \mathrm{GL}(X)$ be the chosen structure group and fix a root r. Restrict both edge transports and gauge values to G. Gauge equivalence classes of G valued transport systems are naturally identified with*

$$\mathrm{Hom}(\pi_1(\Gamma, r), G)/G,$$

where G acts by conjugation. After a spanning tree and orientations of the remaining edges are chosen, this set is identified with simultaneous conjugacy classes of

$$b_1 = |E| - |V| + 1$$

tuples in G.

Proof. A transport system assigns a group element to every oriented edge, with inversion under reversal. The product along a based loop gives a homomorphism from the free group $\pi_1(\Gamma, r)$ to G. A gauge change conjugates this homomorphism by the gauge value at the root. Conversely, normalize a transport system on a spanning tree. The remaining edge values prescribe the images of a free basis of $\pi_1(\Gamma, r)$, and every homomorphism arises in this way. A further gauge preserving the tree normalization is constant at all vertices and acts by simultaneous conjugation. The number of non-tree edges is b_1. □

Corollary 26.9 (Abelian cohomology class)**.** *When the structure group A is abelian, gauge classes of transports are naturally*

$$H^1(\Gamma, A) \cong \mathrm{Hom}(H_1(\Gamma, \mathbb{Z}), A).$$

After a basis of $H_1(\Gamma, \mathbb{Z})$ is chosen, this group is identified with A^{b_1}.

Proof. An edge assignment is an A valued one cochain. A gauge change multiplies it by a coboundary. A graph has no two cells, so every one cochain is a cocycle. The usual computation of the first cohomology of a connected graph gives A^{b_1}. □

Corollary 26.10 (Projective transport)**.** *The same normalization and gluing statements hold for transports valued in a projective linear group. A projective transport is globally trivial exactly when all fundamental holonomies are identity elements in that projective group.*

Proof. Apply theorems 26.3 and 26.4 with the projective linear group as the structure group. □

Example 26.11 (A square loop)**.** A square graph has four vertices and four edges, so it has one fundamental cycle. After a tree gauge, three edges are identities and the fourth is the full loop holonomy. The system glues to one global frame exactly when that remaining matrix is the identity.

27 Smooth projector transport

Finite graph transport has a continuous counterpart. Let H be a Hilbert space and let

$$\mathcal{P}(t) = (P_\alpha(t))_{\alpha \in I}$$

be a complete orthogonal packet system with finitely many labels. Assume that it is continuously differentiable in operator norm for $t \in [0,T]$, where $T \geq 0$. Here every $P_\alpha(t)$ is a bounded self-adjoint projection, so $P_\alpha(t)^* = P_\alpha(t) = P_\alpha(t)^2$.

Zero projections are permitted in this section and contribute zero to all generator sums. When $T = 0$, every partition product is the empty product I. Completeness and orthogonality give

$$P_\alpha(t)P_\beta(t) = 0 \quad \text{for } \alpha \neq \beta, \qquad \sum_\alpha P_\alpha(t) = I.$$

Define the canonical generator

$$K(t) = \sum_\alpha \dot{P}_\alpha(t)P_\alpha(t).$$

Lemma 27.1 (Complete packet commutator identity)**.** *For every packet label β,*

$$\dot{P}_\beta(t) = [K(t), P_\beta(t)].$$

The generator is also skew-adjoint:

$$K(t)^* = -K(t).$$

Proof. Differentiating $P_\alpha^2 = P_\alpha$ gives

$$P_\alpha \dot{P}_\alpha P_\alpha = 0.$$

Differentiating $P_\beta P_\alpha = 0$ for $\alpha \neq \beta$ gives

$$P_\beta \dot{P}_\alpha = -\dot{P}_\beta P_\alpha.$$

Hence

$$KP_\beta = \dot{P}_\beta P_\beta$$

and

$$P_\beta K = -\dot{P}_\beta (I - P_\beta).$$

Their difference is $\dot{P}_\beta$. Also,

$$K + K^* = \sum_\alpha (\dot{P}_\alpha P_\alpha + P_\alpha \dot{P}_\alpha) = \sum_\alpha \dot{P}_\alpha = 0.$$

□

Theorem 27.2 (Canonical packet transport)**.** *Let $U(t)$ solve*

$$\dot{U}(t) = K(t)U(t), \qquad U(0) = I.$$

Then $U(t)$ is unitary and transports every packet exactly:

$$P_\alpha(t)U(t) = U(t)P_\alpha(0)$$

for all α.

Proof. The continuous bounded generator has an evolution inverse. Indeed, solve $\dot{V} = -VK$ with $V(0) = I$. Differentiation gives $(VU)' = 0$, hence $VU = I$. The product $W = UV$ satisfies $\dot{W} = KW - WK$ and $W(0) = I$. Uniqueness of this bounded linear equation gives $UV = I$. Thus U is invertible.

Skew-adjointness of $K(t)$ gives

$$\frac{d}{dt}(U(t)^*U(t)) = 0,$$

so $U(t)$ is unitary. By lemma 27.1,

$$\frac{d}{dt}(U(t)^{-1}P_\alpha(t)U(t)) = U(t)^{-1}(\dot{P}_\alpha - [K, P_\alpha])U(t) = 0.$$

The expression equals $P_\alpha(0)$ at time zero. □

Corollary 27.3 (One moving subspace)**.** *For one orthogonal projection $P(t)$ and its complement, the canonical generator is*

$$K(t) = [\dot{P}(t), P(t)].$$

The transport in theorem 27.2 maps $\operatorname{im} P(0)$ *unitarily onto* $\operatorname{im} P(t)$*.*

Proof. Apply the complete packet formula to $P(t)$ and $I - P(t)$, then simplify the sum. □

Definition 27.4 (Direct rotation of nearby packet systems)**.** Let $(P_\alpha)_{\alpha\in I}$ and $(Q_\alpha)_{\alpha\in I}$ be complete orthogonal packet systems on the same Hilbert space. Put

$$S(\mathcal{Q}, \mathcal{P}) = \sum_{\alpha\in I} Q_\alpha P_\alpha.$$

When $S(\mathcal{Q}, \mathcal{P})$ is invertible, define

$$\mathcal{U}(\mathcal{Q}, \mathcal{P}) = S(\mathcal{Q}, \mathcal{P})(S(\mathcal{Q}, \mathcal{P})^*S(\mathcal{Q}, \mathcal{P}))^{-1/2}.$$

Proposition 27.5 (Exact direct packet rotation)**.** *If*

$$\max(\{0\} \cup \{\|P_\alpha - Q_\alpha\| : \alpha \in I\}) < 1,$$

then $S(\mathcal{Q}, \mathcal{P})$ is invertible. The operator $\mathcal{U}(\mathcal{Q}, \mathcal{P})$ is unitary and satisfies

$$Q_\alpha\mathcal{U}(\mathcal{Q}, \mathcal{P}) = \mathcal{U}(\mathcal{Q}, \mathcal{P})P_\alpha$$

for every α.

Proof. The identities

$$Q_\alpha S(\mathcal{Q}, \mathcal{P}) = S(\mathcal{Q}, \mathcal{P})P_\alpha = Q_\alpha P_\alpha$$

follow from pairwise orthogonality. Also

$$S^*S = \sum_\alpha P_\alpha Q_\alpha P_\alpha, \qquad SS^* = \sum_\alpha Q_\alpha P_\alpha Q_\alpha.$$

Put $\delta = \max(\{0\} \cup \{\|P_\alpha - Q_\alpha\| : \alpha \in I\})$. On $P_\alpha H$,

$$P_\alpha Q_\alpha P_\alpha = P_\alpha - P_\alpha(P_\alpha - Q_\alpha)P_\alpha,$$

so this block is bounded below by $1-\delta > 0$. The same argument on $Q_\alpha H$ applies to $Q_\alpha P_\alpha Q_\alpha$. Thus both S^*S and SS^* are invertible. It follows that S is injective and surjective. The polar factor $\mathcal{U} = S(S^*S)^{-1/2}$ is unitary. Since S^*S commutes with every P_α, the intertwining identity for S passes to $\mathcal{U}$. □

Lemma 27.6 (One-step direct rotation estimate)**.** *Assume $\mathcal{P}(t)$ is twice continuously differentiable in operator norm on $[0,T]$. There are constants $h_0 > 0$ and $C > 0$ such that, whenever $t, t+h \in [0,T]$ and $|h| \le h_0$,*

$$\|\mathcal{U}(\mathcal{P}(t+h), \mathcal{P}(t)) - I - hK(t)\| \le Ch^2.$$

The constants can be chosen uniformly in t.

Proof. Taylor's theorem in the Banach space of bounded operators gives

$$P_\alpha(t+h) = P_\alpha(t) + h\dot{P}_\alpha(t) + E_\alpha(t,h), \qquad \|E_\alpha(t,h)\| \le C_1 h^2,$$

uniformly in t and α. Hence

$$S(\mathcal{P}(t+h), \mathcal{P}(t)) = I + hK(t) + E(t,h), \qquad \|E(t,h)\| \le C_2 h^2.$$

Since $K(t)^* = -K(t)$,

$$S^*S = I + E_2(t,h), \qquad \|E_2(t,h)\| \le C_3 h^2.$$

For small h, the spectrum of S^*S lies in a fixed compact subinterval of $(0,\infty)$. The functional calculus for $z^{-1/2}$ gives

$$(S^*S)^{-1/2} = I + E_3(t,h), \qquad \|E_3(t,h)\| \le C_4 h^2.$$

Multiplying the last two expansions proves the estimate. □

Theorem 27.7 (Direct rotations converge to packet transport)**.** *Assume the complete packet family $\mathcal{P}(t)$ is twice continuously differentiable in operator norm on $[0,T]$. For a partition*

$$0 = t_0 < t_1 < \cdots < t_N = T$$

with sufficiently small mesh, define

$$V_\pi = \mathcal{U}(\mathcal{P}(t_N), \mathcal{P}(t_{N-1})) \cdots \mathcal{U}(\mathcal{P}(t_1), \mathcal{P}(t_0)).$$

Then

$$V_\pi \longrightarrow U(T)$$

in operator norm, where U is the canonical transport from theorem 27.2. On a fixed compact interval the error is bounded by a constant times the mesh.

Proof. Let $U(t,s)$ be the exact propagator generated by K. Since K is continuously differentiable, the integral equation for $U(t+h,t)$ gives a uniform estimate

$$\|U(t+h,t) - I - hK(t)\| \le C_1 h^2.$$

By lemma 27.6, the direct rotation V_j from t_j to t_{j+1} therefore satisfies

$$\|V_j - U(t_{j+1}, t_j)\| \le C_2 h_j^2, \qquad h_j = t_{j+1} - t_j.$$

Both V_j and the exact propagator are unitary. A telescoping product comparison gives

$$\|V_\pi - U(T,0)\| \le \sum_j \|V_j - U(t_{j+1},t_j)\| \le C_2 \sum_j h_j^2 \le C_2 T|\pi|.$$

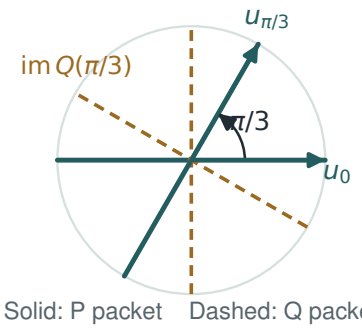
This proves norm convergence and the stated first-order global bound. □

Exact packet matching at each sample

Each direct rotation maps a sampled packet exactly to the corresponding packet at the next sample. The product therefore preserves packet labels throughout the partition. Convergence compares this discrete product with the canonical continuous transport. The uniform estimate uses twice continuous differentiability in operator norm, as stated in the theorem.

A rotating line and its complementary packet

$u_\theta = (\cos\theta, \sin\theta)$, $P(\theta) = u_\theta u_\theta^\mathsf{T}$, $Q(\theta) = I - P(\theta)$.



Canonical continuous transport

$$K = [P', P] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$U'(\theta) = KU(\theta), \quad U(0) = I$$

$$U(\theta) = R(\theta)$$

$$S = \cos\delta\, R(\delta), \qquad S^\mathsf{T}S = \cos^2\delta\, I$$

For $|\delta| < \pi/2$: polar$(S) = R(\delta)$ exactly.

Steps N	Each increment	Ordered product
2	$\pi/6$	$R(\pi/3)$
4	$\pi/12$	$R(\pi/3)$
8	$\pi/24$	$R(\pi/3)$

Every mesh product is exact in this finite model.
The general theorem concerns convergence for broader projector paths.

Figure 6. A rotating rank-one packet and its orthogonal complement. With $u(\theta) = (\cos\theta, \sin\theta)$ and $P(\theta) = u(\theta)u(\theta)^\mathsf{T}$, the canonical generator is $[P', P] = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. For a positive step $\delta < \pi/2$, the polar unitary is the usual rotation $R(\delta)$. On each displayed mesh, the direct rotation product equals $R(\pi/3)$ exactly. This special planar example illustrates the construction. The general theorem supplies convergence for families whose step products need not already equal the continuous transport.

Theorem 27.8 (Convergence of sampled transports)**.** *Assume K is Lipschitz on $[0,T]$. For a partition*

$$0 = t_0 < t_1 < \cdots < t_N = T,$$

define

$$U_\pi = \exp\big((t_N - t_{N-1})K(t_{N-1})\big)\cdots\exp\big((t_1 - t_0)K(t_0)\big).$$

As the mesh $|\pi|$ tends to zero,

$$U_\pi \longrightarrow U(T)$$

in operator norm. There is a constant C, depending only on T, the supremum norm of K, and its Lipschitz constant, such that

$$\|U_\pi - U(T)\| \le C|\pi|.$$

Proof. On each interval, compare the exact propagator with the frozen exponential. Variation of constants and the Lipschitz bound give a local error bounded by a constant times the square of the interval length. Every factor is unitary, so multiplication does not amplify norms. Summing the local errors gives a bound by a constant times

$$\sum_j (t_{j+1} - t_j)^2 \le T|\pi|.$$

□

Corollary 27.9 (Continuous holonomy from graph products)**.** *Assume the complete packet family is twice continuously differentiable in operator norm on $[0,T]$. If $P_\alpha(0) = P_\alpha(T)$ for every α, then $U(T)$ preserves every initial packet. Its restriction to each packet is the holonomy of the closed packet path. The products in theorems 27.7 and 27.8 converge to these holonomies. The direct rotation product has the added property that every factor maps each sampled packet exactly to its successor.*

Proof. The transport identity in theorem 27.2 gives packet preservation at the endpoint. The two convergence statements follow from theorems 27.7 and 27.8. □

Example 27.10 (A rotating line)**.** In $\mathbb{R}^2$, let

$$v(t) = (\cos t, \sin t)$$

and let $P(t)$ be the orthogonal projection onto the line spanned by $v(t)$. The canonical transport is the ordinary rotation matrix

$$U(t) = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}.$$

It carries the initial line and its orthogonal complement to the two moving packets. Products of short rotations converge to the same transport.

Standard Fact 27.11 (Curvature of a projected connection)**.** Let P be a smooth orthogonal projector-valued map on a parameter manifold. The connection

$$\nabla = P\,d$$

on the range bundle has curvature

$$\mathcal{F} = P(dP \wedge dP)P.$$

Proof. For a section s satisfying $Ps = s$,

$$\nabla^2 s = P\, d(P\, ds) = P(dP) \wedge ds.$$

Since $ds = (dP)s + Pds$ and $P(dP)P = 0$, the surviving term is

$$P(dP \wedge dP)Ps.$$ □

28 Realization data and limits of the calculus

The smooth transport theorem starts with a declared projector family. Applying it to a physical model requires a construction of that family and, where relevant, a separate continuum limit. Triviality of holonomy is a further geometric question. The curvature formula in standard fact 27.11 is classical. Relating it to global holonomy requires geometric information as in [7, 8, 10].

An application begins by specifying the following objects:

1. an ambient nonlinear equation or continuation problem
2. the allowed correction space
3. a linear correction map or a correction complex
4. the defect whose class is to be tested
5. a packet decomposition or filtration compatible with the correction data
6. transition maps between refinement levels
7. transport maps when local solutions must be glued.

Once these objects are fixed, the cokernel and cone constructions are functorial by theorems 21.3 and 21.11. Packet splitting and descent follow from theorem 21.4, proposition 21.6, and corollary 21.12. Relative loss under coarsening is measured by theorem 21.7.

Filtered lifting follows from theorems 23.2, 23.6 and 23.8. Gluing and comparison of transport systems follow from theorems 26.4 and 26.6. Smooth packet transport follows from theorems 27.2 and 27.7.

Exact packet coordinates give a representation of the data. Solving the associated problem still requires the correction and lifting hypotheses. The correction map belongs to the problem data, and its cokernel is canonical relative to that choice. The cokernel may be infinite dimensional. In the Hilbert setting, the orthogonal obstruction projector requires closed range.

Coarse vanishing can erase a nonzero fine obstruction, and higher nonlinear terms can obstruct a problem whose linear class vanishes. A spectral sequence records which classes survive the later differentials. An application of finite graph holonomy or smooth projector transport to a continuum field theory requires the additional realization data just described.

Whenever a nonlinear continuation problem supplies a correction map, its residual can be recorded in an exact quotient or cohomology object. Under the stated hypotheses, these objects admit packet decomposition, filtered refinement, natural transport, and holonomy tests. They identify the obstruction that a correction or gluing step must address.

REALIZATION CALCULUS

Projector Resolved Seed Calculus III

Analytic Packet Jets and Exact Composition

Mathematical and verification supplement

Komorebi P.

This supplement gives full ordinary mathematical arguments for the analytic packet claims of Section 17, identifies a stronger norm bound, and records how the calculation relates to the actual projector ranges. The accompanying formal development verifies multilinear reconstruction, continuous jets on the actual projector ranges, their norm bound, and the decorated composition and structural support laws. The ordinary Taylor and counting arguments retain their proof status as stated here. The companion Lean guide identifies the exact declarations and replay evidence.

1 Two distinct contracts

Let $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ be real Banach spaces. The same statements hold for complex Banach spaces with complex Fréchet derivatives. The integral below uses a real parameter. A *bounded packet system* on X is a finite family $P_i \in \mathcal{L}(\mathsf{X}, \mathsf{X})$ such that

$$\sum_{i \in I} P_i = \mathrm{id}_{\mathsf{X}}, \qquad P_i^2 = P_i, \qquad P_i P_j = 0 \quad (i \neq j). \tag{1}$$

Write $\mathsf{X}_i = \operatorname{ran} P_i$ with the subspace norm. Analogously, let $(Q_\gamma)_{\gamma \in J}$ and $(R_\delta)_{\delta \in K}$ be bounded packet systems on Y and Z. Zero projectors are allowed in this supplement. An empty family can satisfy (1) only on the zero space. Its coordinate product is the zero vector space with norm zero. For a continuous r-linear map, the operator norm is the supremum over tuples whose individual norms are at most one. Derivative orders are nonnegative integers, and a packet jet has positive order unless a nullary convention is explicitly invoked.

A weaker family of operators that only satisfies $\sum_i P_i = \mathrm{id}$ yields a useful algebraic expansion, but it need not produce independent coordinates. We keep these statements separate.

Lemma 1.1 (Closed ranges and bounded coordinates)**.** *Each X_i is closed. The maps*

$$A_{\mathsf{X}}(x) = (P_i x)_{i \in I} : \mathsf{X} \longrightarrow \prod_{i \in I} \mathsf{X}_i, \qquad S_{\mathsf{X}}((z_i)_{i \in I}) = \sum_{i \in I} z_i$$

are inverse bounded linear maps when the finite product carries the maximum norm.

Proof. The identity $\operatorname{ran} P_i = \ker(\mathrm{id} - P_i)$ follows from idempotence. The kernel of a continuous linear map is closed, so each range is itself Banach. Both coordinate maps are bounded because the family is finite. The identity $\sum_i P_i = \mathrm{id}_{\mathsf{X}}$ gives $S_{\mathsf{X}} A_{\mathsf{X}} = \mathrm{id}$. If $z_i \in \operatorname{ran} P_i$, then $P_i z_i = z_i$ and $P_j z_i = 0$ for $j \neq i$. Thus $A_{\mathsf{X}} S_{\mathsf{X}} = \mathrm{id}$. Moreover, $\|A_{\mathsf{X}}\| \leq \max_i \|P_i\|$ and $\|S_{\mathsf{X}}\| \leq |I|$ whenever I is nonempty. The case of the zero space follows from the identities directly. □

2 Multilinear reconstruction and packet jets

Theorem 2.1 (Completeness is enough for multilinear expansion)**.** *Let I, J be finite and $r \geq 0$. Suppose $\sum_i P_i = \mathrm{id}_{\mathsf{X}}$ and $\sum_\gamma Q_\gamma = \mathrm{id}_{\mathsf{Y}}$, without assuming idempotence or annihilation. Let $B : \mathsf{X}^r \to \mathsf{Y}$ be multilinear. Then, for all $h_1, \ldots, h_r \in \mathsf{X}$,*

$$B(h_1, \ldots, h_r) = \sum_{\gamma \in J} \sum_{\alpha \in I^r} Q_\gamma B(P_{\alpha_1} h_1, \ldots, P_{\alpha_r} h_r). \tag{2}$$

The statement includes $r = 0$ under the usual empty-product convention.

Proof. For every input, $h_a = \sum_{i \in I} P_i h_a$. Additivity in each slot expands B into the sum over all choices $\alpha \in I^r$. The finite output identity expresses each resulting value as its sum of Q_γ components. All exchanges of sums are finite. No property beyond completeness and multilinearity enters the calculation. For $r = 0$, the sum over input indices has exactly one term, and output completeness still gives the identity. □

Proposition 2.2 (Literal continuous range jet)**.** *Under (1) for the input and output families, let $B : \mathsf{X}^r \to \mathsf{Y}$ be a continuous multilinear map, $r \geq 1$. The formula*

$$J_{\gamma|\alpha} B(z_1, \ldots, z_r) = Q_\gamma B(z_1, \ldots, z_r), \qquad z_a \in \mathsf{X}_{\alpha_a},$$

defines a continuous multilinear map $\prod_a \mathsf{X}_{\alpha_a} \to \mathsf{Y}_\gamma$ with the actual range as codomain. For $z_a = P_{\alpha_a} h_a$, its image in Y is the corresponding summand in (2).

Proof. The output lies in $\operatorname{ran} Q_\gamma$ by construction. The subspace inclusions are continuous and isometric. The output projector Q_γ and B are continuous. Their composition therefore gives a continuous multilinear map into Y. Since Y_γ carries the induced norm and is closed, viewing this value in Y_γ does not change its norm or continuity. On the projected arguments $P_{\alpha_a} h_a$, the defining expression is precisely the summand in (2). □

Proposition 2.3 (A stronger range norm bound)**.** *The literal range jet satisfies*

$$\left\| J_{\gamma|\alpha} B \right\| \leq \|Q_\gamma\| \, \|B\| \, . \tag{3}$$

Its ambient extension $\widetilde{J} = Q_\gamma \circ B \circ (P_{\alpha_1}, \ldots, P_{\alpha_r})$ satisfies

$$\left\| \widetilde{J} \right\| \leq \|Q_\gamma\| \, \|B\| \prod_{a=1}^{r} \|P_{\alpha_a}\| \, . \tag{4}$$

In particular, the published bound for the range jet with the projector norm product on the right follows from (3).

Proof. For inputs $z_a \in \mathsf{X}_{\alpha_a}$, the induced norms yield $\|Q_\gamma B(z_1, \ldots, z_r)\| \leq \|Q_\gamma\| \, \|B\| \prod_a \|z_a\|$, proving (3). Apply additionally $\|P_{\alpha_a} h_a\| \leq \|P_{\alpha_a}\| \, \|h_a\|$ to obtain (4). If every selected input projector is nonzero, idempotence implies $\|P_{\alpha_a}\| \geq 1$ by evaluating at a unit vector in its range. Then (3) implies the weaker published inequality. If any selected input projector is zero, the corresponding input range is the zero space and the positive-order jet is zero. Both sides of the published inequality then vanish. This completes the comparison, including the degenerate case. □

3 Derivatives, coordinates, and Taylor remainder

Let $U \subset \mathsf{X}$ be open and $F : U \to \mathsf{Y}$ be C^m. For $1 \le r \le m$, the Fréchet derivative $\mathrm{D}^r F(x)$ is a continuous multilinear map. Apply Theorem 2.1 to $B = \mathrm{D}^r F(x)$. Thus the actual packet jets recover the entire derivative, without presupposing a separate packet derivative object.

Corollary 3.1 (Conjugated derivatives in packet coordinates)**.** *Let $\widehat{F} = A_{\mathsf{Y}} \circ F \circ S_{\mathsf{X}}$ on the open set $A_{\mathsf{X}}(U)$. Then $\widehat{F}$ is C^m and, for $1 \le r \le m$,*

$$\mathrm{D}^r \widehat{F}(u)[v_1, \dots, v_r] = A_{\mathsf{Y}}\big(\mathrm{D}^r F(S_{\mathsf{X}} u)[S_{\mathsf{X}} v_1, \dots, S_{\mathsf{X}} v_r]\big).$$

If v_a is supported only in packet slot α_a, let $z_a \in \mathsf{X}_{\alpha_a}$ denote its possibly zero coordinate in that slot. Then the output component γ is

$$J_{\gamma|\alpha}(\mathrm{D}^r F(S_{\mathsf{X}} u))[z_1, \dots, z_r].$$

Proof. Lemma 1.1 makes A_{X} a homeomorphism, so $A_{\mathsf{X}}(U)$ is open in the Banach coordinate product. The operators A_{Y} and S_{X} are bounded and linear. Their derivatives of order at least two vanish. Repeated application of the ordinary Fréchet chain rule therefore gives the stated formula. The last assertion follows because S_{X} includes a single supported slot into X and A_{Y} applies Q_γ in the chosen output slot. □

Theorem 3.2 (Packet Taylor identity with integral remainder)**.** *Let $F : U \to \mathsf{Y}$ be C^{m+1} on an open set containing the closed segment $[x, x+h]$. For $m \ge 0$,*

$$Q_\gamma F(x+h) = Q_\gamma F(x) + \sum_{r=1}^{m} \frac{1}{r!} \sum_{\alpha \in I^r} J_{\gamma|\alpha}(\mathrm{D}^r F(x))[P_{\alpha_1} h, \dots, P_{\alpha_r} h] + Q_\gamma R_{m+1},$$

where

$$R_{m+1} = \frac{1}{m!} \int_0^1 (1-t)^m \mathrm{D}^{m+1} F(x+th)[h^{m+1}]\, dt.$$

If $\|\mathrm{D}^{m+1} F(x+th)\| \le M$ on the segment, where $M \ge 0$, then

$$\|Q_\gamma R_{m+1}\| \le \frac{\|Q_\gamma\|\ M\, \|h\|^{m+1}}{(m+1)!}.$$

Proof. The preimage of U under $t \mapsto x + th$ is an open neighborhood of $[0,1]$, and $g(t) = F(x+th)$ is C^{m+1} there. Repeated chain differentiation gives $g^{(r)}(t) = \mathrm{D}^r F(x+th)[h^r]$. These derivatives are continuous on the compact interval, so the displayed Bochner integrals exist in the Banach codomain. To see the exact normalization, put

$$T_j = \frac{1}{j!} \int_0^1 (1-t)^j g^{(j+1)}(t)\, dt \qquad (0 \le j \le m).$$

The Banach-valued fundamental theorem of calculus gives $T_0 = g(1) - g(0)$. Integration by parts gives $T_j = T_{j-1} - g^{(j)}(0)/j!$ for $1 \le j \le m$. Iterating yields the Taylor identity with remainder $T_m = R_{m+1}$. Each Q_γ is bounded, so it commutes with the Bochner integral. Expand each derivative by Theorem 2.1 and use $Q_\gamma Q_\eta = 0$ for $\eta \ne \gamma$ and $Q_\gamma^2 = Q_\gamma$ to take its γ component. The norm bound follows from the derivative norm and $\int_0^1 (1-t)^m\, dt = 1/(m+1)$. When $m = 0$ the derivative sum is empty, and when $h = 0$ every positive-order term and the remainder vanish. □

4 The partition chain rule and packet decorations

Let $U \subseteq \mathsf{X}$ and $V \subseteq \mathsf{Y}$ be open, let $G : U \to \mathsf{Y}$ and $F : V \to \mathsf{Z}$ be C^n, and assume $G(U) \subseteq V$. Fix $x \in U$. Write Π_n for set partitions of $\{1, \dots, n\}$. Within each block the directions are listed by increasing index. Blocks may be listed by increasing least element to make every displayed tuple unambiguous.

Theorem 4.1 (Faà di Bruno in Banach spaces)**.** *For $n \geq 1$ and $h_1, \dots, h_n \in \mathsf{X}$,*

$$\mathrm{D}^n(F \circ G)(x)[h_1, \dots, h_n] = \sum_{\pi \in \Pi_n} \mathrm{D}^{|\pi|}F(Gx)\left[\mathrm{D}^{|B|}G(x)[h_i : i \in B] : B \in \pi\right].$$

The order of blocks is immaterial because Fréchet higher derivatives are symmetric.

Proof. For $n = 1$ this is the chain rule. Suppose the identity holds at order n and the maps are C^{n+1}. Differentiate a summand in direction h_{n+1}. Differentiating the outer $\mathrm{D}^{|\pi|}F(Gx)$ produces the same partition with the new singleton block $\{n+1\}$. Differentiating the argument $\mathrm{D}^{|B|}G(x)[h_i : i \in B]$ produces the partition obtained by adjoining $n+1$ to the existing block B. Every partition of $\{1, \dots, n+1\}$ arises in exactly one of these ways, depending on whether $\{n+1\}$ is a singleton. The product and chain rules apply to the bounded multilinear evaluations, and all sums are finite. Induction completes the proof. □

Corollary 4.2 (Decorated packet chain rule)**.** *Fix input labels α_i, directions $h_i \in \mathsf{X}_{\alpha_i}$, and an output label δ. For each partition π and intermediate label function $\gamma : \pi \to J$, replace the Bth argument of the outer derivative by $Q_{\gamma(B)}\mathrm{D}^{|B|}G(x)[h_i : i \in B]$. The sum over all such decorations, followed by R_δ, equals the δ output packet of $\mathrm{D}^n(F \circ G)(x)[h_1, \dots, h_n]$.*

Proof. In every outer argument of Theorem 4.1, insert the identity on Y as $\sum_{\gamma \in J} Q_\gamma$. Expand by multilinearity. The output projection R_δ distributes over the finite sum. Because the directions are already in their actual packet ranges, this is precisely the literal packet-jet expression. No statement assumes that mixed input labels disappear. □

Proposition 4.3 (Decorated term count and pruning)**.** *For $n \geq 1$, if the intermediate family contains $M \geq 0$ packet labels, the number of formal decorated terms before pruning is*

$$N_{n,M} = \sum_{k=1}^{n} S(n,k)M^k,$$

where $S(n,k)$ is a Stirling number of the second kind. A decorated summand vanishes if any of its constituent packet jets is the zero multilinear map.

More precisely, define the order-n support at x by

$$\mathcal{S}_n(F \circ G; x) = \{(\delta, \alpha) \in K \times I^n : J_{\delta|\alpha}(\mathrm{D}^n(F \circ G)(x)) \neq 0\}.$$

Let $\mathcal{A}_n(F, G; x) \subseteq K \times I^n$ consist of the profiles (δ, α) for which there exist $\pi \in \Pi_n$ and $\gamma : \pi \to J$ such that all the following restricted multilinear maps are nonzero:

$$J_{\delta|(\gamma(B))_{B\in\pi}}(\mathrm{D}^{|\pi|}F(Gx)),$$
$$J_{\gamma(B)|(\alpha_i)_{i\in B}}(\mathrm{D}^{|B|}G(x)) \qquad (B \in \pi).$$

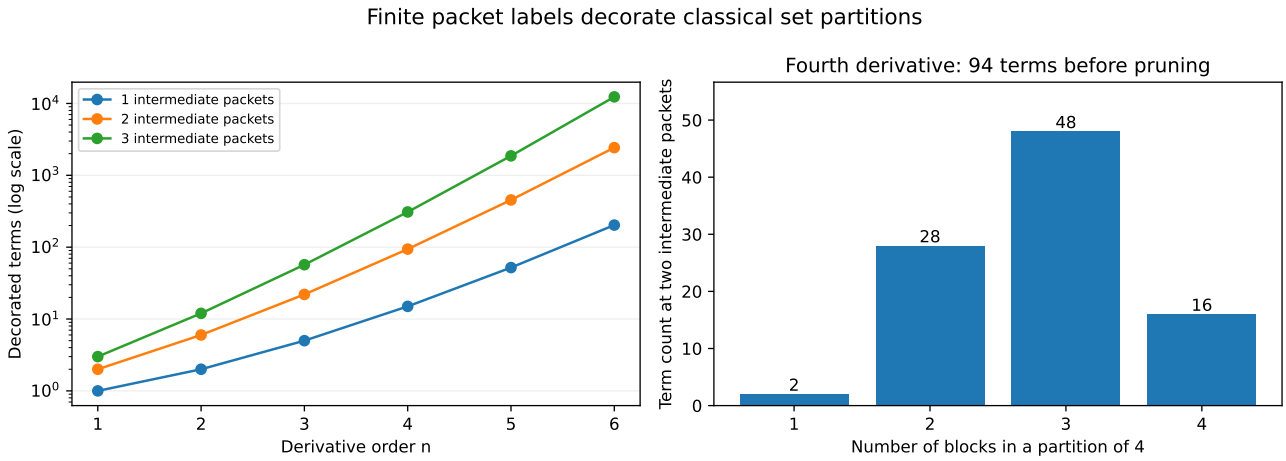


Figure 1. The growth of labeled partition terms. At fourth order with two intermediate packet labels, the contributions for the four block counts are 2, 28, 48, and 16, totaling 94. These are exact combinatorial counts, before structural pruning or cancellations.

Here the jet notation uses the relevant input and output packet families. Then

$$\mathcal{S}_n(F \circ G; x) \subseteq \mathcal{A}_n(F, G; x).$$

Thus the set of admissible decorated witnesses is projected to the same external packet profiles as the composed support. Equality need not hold.

Proof. There are $S(n,k)$ partitions with k blocks and M^k functions assigning labels to those blocks. If a profile is outside $\mathcal{A}_n(F, G; x)$, every decorated summand has a zero constituent jet. A zero inner jet supplies a zero argument to a positive-order multilinear outer jet. A zero outer jet gives zero directly. Every summand therefore vanishes for every tuple of directions, so the composed restricted jet is zero. This proves the support inclusion. Nonzero constituent maps can still compose to zero, and distinct nonzero summands can cancel. For example, take $\mathsf{X} = \mathsf{Z} = \mathbb{R}$ with one packet, $\mathsf{Y} = \mathbb{R}^2$ with its two coordinate packets, $G(t) = (t, t)$, and $F(y_1, y_2) = y_1 - y_2$. At order one both decorated terms are nonzero and their sum is zero. If $M = 0$, then $\mathsf{Y} = 0$. In that case, for $n \geq 1$ the term count and every derivative of $F \circ G$ are zero. These counts concern formal summands before structural pruning. □

5 Verification and reproduction

Layer	Verified scope
Algebraic reconstruction	The finite core and Section 16 source were compiled in the pinned Lean environment. Multilinear reconstruction retains every input label and the actual output packet.
Continuous packet jets	The accepted source uses literal projector ranges with inherited norms. Its reconstruction theorem specializes to mathlib's total `iteratedFDeriv`. Differentiability is asserted only when a theorem supplies the required hypotheses.

Layer	Verified scope
Composition and support	The accepted source expands the composite derivative over finite ordered partitions and all intermediate labels. A nonzero composite jet has a witness with nonzero constituent jets. This is a one-way support inclusion.
Taylor and counting	The integral remainder, its exact factorial normalization and bound, and the Stirling term count are proved mathematically in this supplement. They are not included in the accompanying Lean formalization.
Exact computation	Three suites passed 18,630 assertions on specified integer, rational, and polynomial examples. Repeated assertions and sampled norm inequalities are identified in the computation audit. These are finite checks.
Later monograph	The ordinary review covers all 59 later results and both classical standard facts under their stated hypotheses. Matching expanded Lean proofs were not recovered for those sections.

The source package contains the exact toolchain and dependency pins, compiler transcripts, ownership and axiom audit, kernel replay, adverse controls, mathematical reviews, and statement register. Run `python VERIFY_RELEASE.py` to check the extracted bytes. The separate Lean guide and `formal/README.md` give the formal replay commands.

Reading rule. An ordinary proof, an exact computation, a Lean theorem under explicit hypotheses, and an open research construction are separate kinds of evidence. The statement register records the scope of each claim.

How to use this volume

Twenty-four worked plates and twenty-two exact model figures accompany the monograph and its Lean guide. Integer, rational, and symbolic calculations determine the figures. The source package includes the model generators and their explicit finite verification ranges.

A route through the examples

The labels Plate III.1 through Plate III.24 identify worked pages independently of PDF pagination. Each page names its corresponding statement in the main text. The Lean guide provides the separate formal contract.

Plate III.1 Three objects, three different questions
Plate III.2 An atlas of the objects
Plate III.3 Read the product in execution order
Plate III.4 All sixteen words, only five walks
Plate III.5 Coefficients need not commute
Plate III.6 A reachable packet can receive zero
Plate III.7 Reachability is only an upper bound
Plate III.8 Paths through changing packet lists
Plate III.9 One vector, three exact frames
Plate III.10 Nonlinear charts still cancel exactly
Plate III.11 Reversibility is a real hypothesis
Plate III.12 Measure the mismatch at its base
Plate III.13 The identity can hide in the spectrum
Plate III.14 Gauge changes the picture, not flatness
Plate III.15 Flat is not the same as calibrated
Plate III.16 Resolve the entire obstruction
Plate III.17 Why diagonal checks are insufficient
Plate III.18 A gauge can move every packet label
Plate III.19 An obstruction rejects one candidate
Plate III.20 A tree reads its named leaves
Plate III.21 Where a nonlinear defect can enter
Plate III.22 Mixed terms are not optional
Plate III.23 From loop data to a packet witness
Plate III.24 Read the evidence with the pictures

Three reading routes

Typed transfer: Plate III.3 to Plate III.8. **Coordinates, connection, and gauge:** Plate III.9 to Plate III.18, then Plate III.23. **Candidate witnesses and trees:** Plate III.19 to Plate III.22, then Plate III.24.

From the finite models to the broader chapter

The monograph continues with rooted tree expansions, nonlinear refinement defects, obstruction objects, filtered lifting, elimination, and smooth projector transport. Read the finite examples as concrete preparation. The broader ordinary proofs keep their own hypotheses and are distinct from the finite Lean contract. Chapter III: rooted trees: Section 18. Nonlinear defects: Section 20. Obstruction objects: Section 21. Filtered lifting: Section 23. Smooth transport: Section 27.

Read products in traversal order

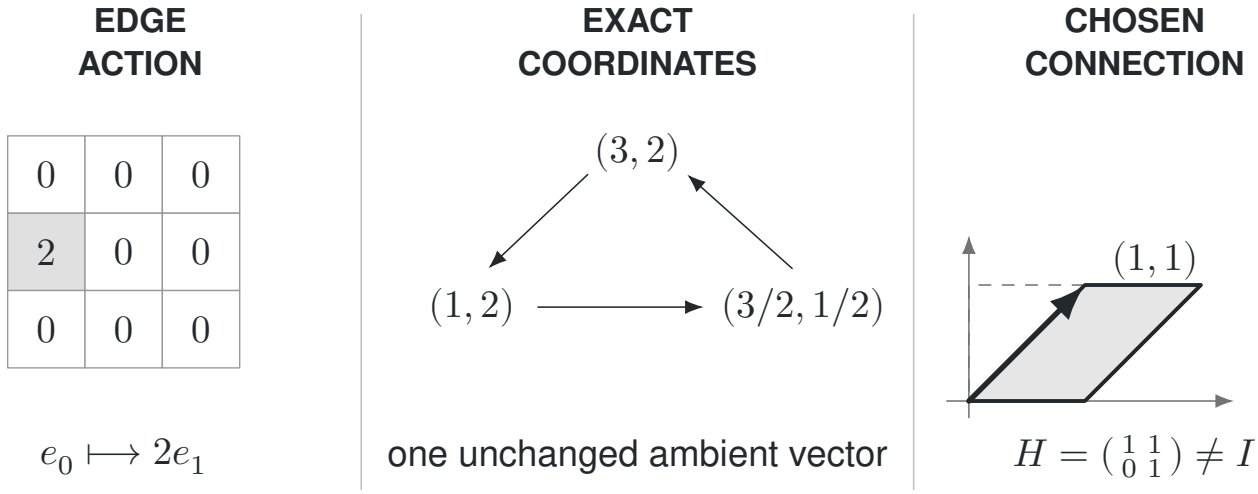
A path $e_1, e_2, \dots, e_n$ acts as $L_{e_n} \cdots L_{e_2} L_{e_1}$: the first traversed edge is the rightmost operator. Row indices are targets and column indices are sources throughout the guide.

Plate III.1 / CHAPTER III / A WORKED VISUAL COMPANION

Three objects, three different questions

Chapter III, mathematical text: Section 1, Section 4, Section 5

An edge operator moves data between named packets. An exact coordinate change describes the same data in another complete chart. A connection supplies an additional choice of transport. The three can use similar arrows while making different mathematical assertions.



Three actual models used below: a single supported matrix entry, one ambient vector in three coordinate systems, and a shear obtained as chosen loop transport. The coordinates in the middle are not three different ambient vectors.

One chapter, three tests

For an edge word, ask whether the target of each step matches the source of the next. For exact coordinates, ask whether analysis and synthesis are inverses. For a connection, ask whether its transports through two steps agree with its declared direct transports. None of these tests can replace another.

This guide follows explicit integer and rational matrices through these tests. Each example has an inspectable input, calculation, and conclusion. Vector drawings display those calculations rather than replace them. A graph with two paths can have zero net transfer. A loop with trace two and determinant one can still differ from the identity.

Scope of the worked models

The worked models illustrate the finite algebraic core. The main text also develops rooted tree expansions, nonlinear defects, obstruction objects, filtered lifting, and analytic transport under their stated hypotheses. Those ordinary mathematical developments have a wider scope than the finite Lean contract. The present models explain the finite mechanisms and provide a route into the broader chapter.

> **Read this as:** Follow the object, not the arrow shape. A typed transfer can vanish, a complete coordinate loop must telescope, and an independently chosen connection can have genuine algebraic holonomy.

Plate III.2 / ORIENTATION / KEEP THE TYPES VISIBLE

An atlas of the objects

Chapter III, mathematical text: Definition 2.1, Definition 5.5, Definition 7.4

Rows are targets and columns are sources throughout. If E_{ji} is a matrix unit, then $E_{ji}e_i = e_j$. The notation e_i means a basis vector. A named graph edge is written a, b, c, d or e, f when its role is explicit.

Object	What it records	First worked page
$\mathrm{src}, \mathrm{tgt} : E \to V$	The ordered endpoints of each distinct edge, including parallel edges.	Plate III.3 to Plate III.4
$L_e = c(e)E_{\mathrm{tgt}(e),\mathrm{src}(e)}$	One operator supported on a packet.	Plate III.3
$L_w = L_{e_n} \cdots L_{e_1}$	Execute the written word from left to right.	Plate III.4 to Plate III.5
$T = \sum_e L_e$	Add all edge operators before taking powers.	Plate III.4 to Plate III.7
F_a	A synthesis frame from chart a into the ambient space.	Plate III.9
$\mathcal{A}_a, \mathcal{S}_a$	Inverse analysis and synthesis maps, which may be maps of sets.	Plate III.10
$J_{b\leftarrow a}$	Canonical coordinate transport, forced by the exact charts.	Plate III.9 to Plate III.10
$\Gamma_{b\leftarrow a}$	Additional connection data, with reversibility stated explicitly.	Plate III.11 to Plate III.12
$\Delta_{c,b,a}$	The difference between transport through two steps and direct transport, a map from C_a to C_c.	Plate III.12
$H_{c,b,a}$	Loop transport based at a, an automorphism of C_a.	Plate III.13
G_a	An invertible gauge change at each chart.	Plate III.14 to Plate III.15
$\Pi_\mu \,\mathrm{Ob}\, \Pi_\nu$	Block from input to output of $\mathrm{Ob} = H - I$.	Plate III.16 to Plate III.18
$r(x, p)$	A residual at packet p for the named candidate x.	Plate III.19
$t, \mathrm{inputs}(t)$	A finite operation tree and its syntactically used input labels.	Plate III.20

Reading routes

For transfer, read Plate III.3 to Plate III.8. For the finite geometry models, read Plate III.9 to Plate III.18. For candidate certificates and nonlinear scope, read Plate III.19 to Plate III.22. Plate III.23 puts the matrix mechanisms together. Plate III.24 gives the formal boundary and replay guide.

> **Read this as:** A $C_a \to C_c$ defect and a $C_a \to C_a$ holonomy are not the same typed map. Returning the defect to the base fiber requires a specified inverse transport.

Plate III.3 / MODEL A / ONE EDGE IS ONE MATRIX ENTRY

Read the product in execution order

Chapter III, mathematical text: Lemma 2.2, Theorem 2.3, Theorem 2.4

Work over $\mathbb{Z}$ with packets $0, 1, 2$. Let $e : 0 \to 1$ have coefficient 2, and let $f : 1 \to 2$ have coefficient 3. The elementary operators are

$$L_e = 2E_{10}, \qquad L_f = 3E_{21}.$$

The location of the nonzero entry records the type of the transfer, not an arbitrary matrix convention.

L_e

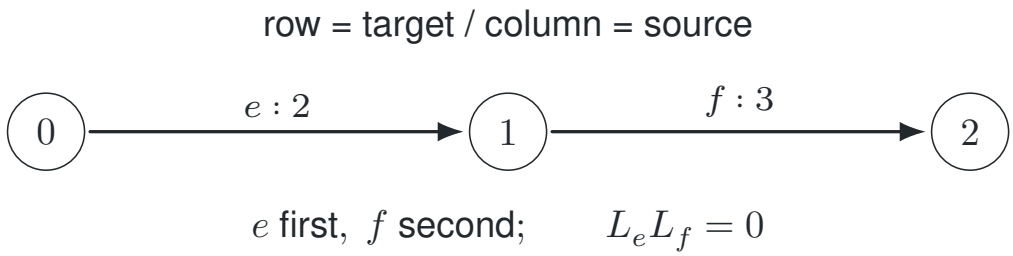
	0	1	2
0	0	0	0
1	2	0	0
2	0	0	0

L_f

	0	1	2
0	0	0	0
1	0	0	0
2	0	3	0

$L_f L_e$

	0	1	2
0	0	0	0
1	0	0	0
2	6	0	0



Each grid is an actual 3×3 matrix. The rightmost matrix factor acts first. In $L_f L_e$, packet 1 is the shared intermediate index. In $L_e L_f$ the interface is broken.

On a vector $(x_0, x_1, x_2)^{\mathsf{T}}$, the first step produces $(0, 2x_0, 0)^{\mathsf{T}}$, and the second produces $(0, 0, 6x_0)^{\mathsf{T}}$. Thus

$$L_f L_e = 6E_{20}, \qquad L_e L_f = 0.$$

For general edges, a summand in the entrywise product can survive only when the same intermediate packet is both $\mathrm{tgt}(e)$ and $\mathrm{src}(f)$:

$$(L_f L_e)_{yx} = \sum_z (L_f)_{yz} (L_e)_{zx}.$$

This is why a broken interface annihilates the entire product, even if both edge coefficients are nonzero.

What the zero says

It says that this ordered composition does not pass data through the declared packet types. It does not say that either edge is individually zero. For a longer word, finding one broken adjacent interface is sufficient to prove that the whole word vanishes.

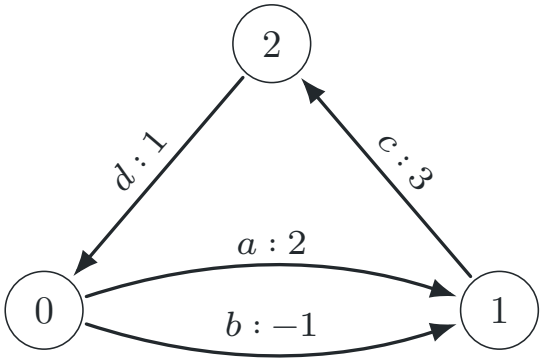
> **Read this as:** Write the traversed edges first. Then reverse their order when writing the operator product: e followed by f is $L_f L_e$.

Plate III.4 / MODEL A / PARALLEL EDGES REMAIN DISTINCT

All sixteen words, only five walks

Chapter III, mathematical text: Theorem 3.8, Corollary 3.9

Add four edges on three packets: $a : 0 \to 1$ with coefficient 2, $b : 0 \to 1$ with coefficient -1, $c : 1 \to 2$ with coefficient 3, and $d : 2 \to 0$ with coefficient 1. The parallel edges a and b are separate labels even though their source and target agree.



$$(T^2)_{20} = 6 - 3 = 3$$

first executed edge = column

	a	b	c	d
a	×	×	×	2
b	×	×	×	−1
c	6	−3	×	×
d	×	×	3	×

second executed edge = row

×: broken packet interface

The board tests all ordered pairs. Columns are the first executed edge and rows are the second. A shaded cell gives the ordered coefficient of a valid walk. A cross means the product is zero by type mismatch.

The surviving words are ac, bc, cd, da, db. Their coefficients are $6, -3, 3, 2, -1$, respectively. Their endpoints determine where these numbers enter the matrix:

$$T = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}, \qquad T^2 = \begin{pmatrix} 0 & 3 & 0 \\ 0 & 0 & 1 \\ 3 & 0 & 0 \end{pmatrix}.$$

At row 2, column 0, the two parallel choices give $6 - 3 = 3$. Deduplicating the edge labels before summing would alter the operator. On the other hand, adding the parallel coefficients in the already assembled matrix is legitimate: both act in the same supported matrix position.

The same calculation extends from pairs to all finite words by distributivity. The included check enumerates every word through length six and independently accumulates the endpoint coefficients. The empty word is treated separately as the identity.

> **Read this as:** The matrix power is a signed, ordered walk sum. It is not simply the number of paths, and a graph edge is not identified solely by its endpoints.

Plate III.5 / MODEL A / A MATRIX RING AS THE COEFFICIENT SEMIRING

Coefficients need not commute

Chapter III, mathematical text: Lemma 3.7, Section 9

The edge theorem allows a noncommutative coefficient semiring. Use $R = M_2(\mathbb{Q})$ and let the two coefficients themselves be matrices:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Attach A to $0 \to 1$ and B to $1 \to 2$. The single surviving endpoint coefficient is BA, not AB.

A

1	1
0	1

B

1	0
1	1

BA

1	1
1	2

AB

2	1
1	1

$$e : 0 \to 1 \text{ carries } A; \quad f : 1 \to 2 \text{ carries } B$$

$$BA(1,0)^{\mathsf{T}} = (1,1)^{\mathsf{T}}, \qquad AB(1,0)^{\mathsf{T}} = (2,1)^{\mathsf{T}}$$

All entries are exact integers. The two products differ on the diagonal and act differently on the same input vector. This example is within the scalar model with semiring coefficients by taking the scalar ring to be $M_2(\mathbb{Q})$.

The distinction is visible before any abstraction:

$$BA \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad AB \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

For a written walk $(e_1, \dots, e_n)$, its coefficient must therefore be

$$c^{\leftarrow}(w) = c(e_n)c(e_{n-1}) \cdots c(e_1).$$

A commutative example can hide an orientation mistake because reversing the coefficients changes nothing. This one cannot.

One ring is not a dependent family of fibers

The 3×3 matrix over $M_2(\mathbb{Q})$ can be displayed as an ordinary 6×6 matrix with 2×2 blocks. This is a concrete representation of the example over one ring. It does not establish an arbitrary family of differently typed edge fibers, which is outside the finite formal contract's scalar interface for edge systems.

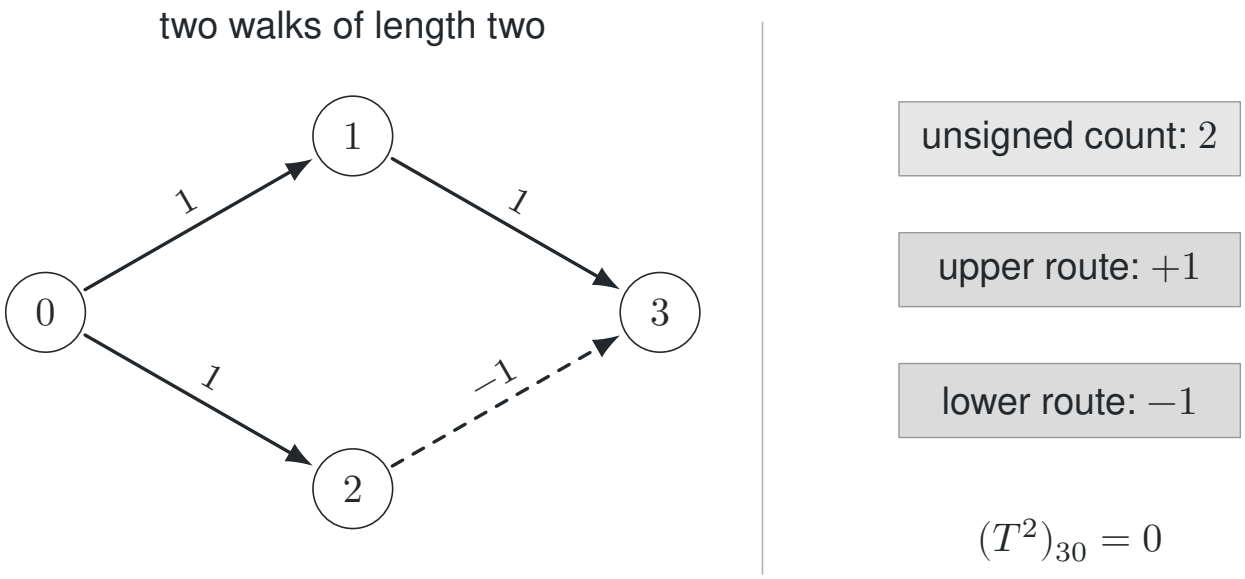
> **Read this as:** The order rule carries mathematical information. It survives exact tests over a noncommutative ring and must not be inferred from pictures drawn over scalar real numbers alone.

Plate III.6 / MODEL A / CANCELLATION IS NOT A BROKEN INTERFACE

A reachable packet can receive zero

Chapter III, mathematical text: Corollary 3.9, Corollary 3.10

Consider a diamond: two routes of length two from packet 0 to packet 3. Three edge coefficients are 1 and the lower final coefficient is -1. Both routes are correctly typed. Neither is annihilated by an interface mismatch.



The dashed edge carries the coefficient -1. It is an actual edge, not an uncertain or missing one. Both paths exist. The unsigned count is two, while their exact operator contributions sum to zero.

The endpoint identity gives

$$(T^2)_{30} = (1)(1) + (-1)(1) = 0.$$

Thus the chapter's reachability implication has a definite direction:

$$\text{no compatible walk} \implies (T^n)_{ji} = 0.$$

Its converse is false. A zero entry may come from absent paths, zero coefficients, zero divisors in the coefficient ring, or cancellation among nonzero terms. The diamond exhibits the last possibility over $\mathbb{Z}$, where no zero divisor is needed.

Three different support records

The declared graph records permitted routes. The support of an individual word records the endpoints of that word when its coefficient is nonzero. The support of the sum records what remains after all contributions have been added. Passing from either of the first two records to the last requires the coefficients, not just adjacency.

A deliberately false test in the companion script asserts that reachability forces nonzero transfer. This diamond makes the test fail. That is a check against overreading the theorem, not a defect in the theorem.

> **Read this as:** Do not interpret a visible route as an occupied output. Graph support is a useful upper bound. Signed operator support can be strictly smaller.

Plate III.7 / MODEL A / AN ORBIT THAT STAYS IN TWO DIMENSIONS

Reachability is only an upper bound

Chapter III, mathematical text: Corollary 3.11

Keep the diamond and add an isolated packet 4. Start with the single seed $e_0 \in \mathbb{Q}^5$. Direct multiplication gives

$$Te_0 = e_1 + e_2, \qquad T^2 e_0 = 0.$$

Every later orbit vector is zero. The orbit hull generated by T is therefore exactly the span of the first two displayed columns.

$[e_0 \;\; Te_0 \;\; T^2 e_0]$

0	1	0	0
1	0	1	0
2	0	1	0
3	0	0	0
4	0	0	0

rank $= 2$

a basis of the orbit hull

1	0
0	1
0	1
0	0
0	0

$\mathrm{Reach}(0) = \{0, 1, 2, 3\}$

$\mathrm{Hull}_T(e_0) = \{(a, b, b, 0, 0)\}$

e_3 : reachable but absent

e_4 : unreachable and absent

The first matrix contains the actual orbit columns. The second supplies a basis of their span. The orbit hull lies inside the reachable subspace on four coordinates, but it has dimension two, not four.

A vector belongs to this hull precisely when it has the form

$$(a, b, b, 0, 0)^{\mathsf{T}}, \qquad a, b \in \mathbb{Q}.$$

Packet 4 is excluded because it is unreachable. Packet 3 is also absent, but for a different reason: the two contributions cancel. Even the reachable packets 1 and 2 remain linked. Powers of the single total operator do not separate them.

The admissible operator grammar matters

Allowing each individual edge operator is a different generation problem. For instance, applying the edge $0 \to 1$ and then $1 \to 3$ individually produces e_3. It does not follow that e_3 belongs to the orbit of the summed operator T. The chapter's theorem on orbit support concerns the explicitly stated operator action.

> **Read this as:** Unreachable means impossible under the declared action. Reachable means only that graph geometry has not ruled it out. Linear dependence and cancellation still have to be checked.

Plate III.8 / MODEL B / RECTANGULAR PRODUCTS WITH TWO OR THREE STEPS

Paths through changing packet lists

Chapter III, mathematical text: Proposition 8.1

Here the packet lists have sizes $2, 3, 2, 1$ in succession. Let the actual maps be

$$\mathbb{Q}^2 \xrightarrow{A} \mathbb{Q}^3 \xrightarrow{B} \mathbb{Q}^2 \xrightarrow{C} \mathbb{Q}.$$

Unlike a power of one matrix, each step has its own entries and its own source and target list.

$A : \mathbb{Q}^2 \to \mathbb{Q}^3$

1	0
2	1
0	3

$B : \mathbb{Q}^3 \to \mathbb{Q}^2$

1	−1	2
0	1	1

$C : \mathbb{Q}^2 \to \mathbb{Q}$

2	−1

CBA

−4	6

	$(j,k)=(0,0)$	$(0,1)$	$(0,2)$	$(1,0)$	$(1,1)$	$(1,2)$	
$i=0$	2	−4	0	0	−2	0	$=-4$
$i=1$	0	−2	12	0	−1	−3	$=6$

The lower board expands one final row into all six choices of intermediate indices for each input packet. The entries are $C_{0j}B_{jk}A_{ki}$, with $k \in \{0,1,2\}$ and $j \in \{0,1\}$. Zero terms remain visible.

The intermediate and total products are

$$BA = \begin{pmatrix} -1 & 5 \\ 2 & 4 \end{pmatrix}, \qquad CBA = \begin{pmatrix} -4 & 6 \end{pmatrix}.$$

For input packet 0, the six contributions sum to $2 - 4 + 0 + 0 - 2 + 0 = -4$. For input packet 1, they sum to $0 - 2 + 12 + 0 - 1 - 3 = 6$. These are literal matrix entries, not probabilities or path counts.

The formal scope is deliberately precise

The finite formal contract's forms with two and three steps allow changing packet types. Its heterogeneous product theorem for arbitrary lengths allows changing *factors* over one common finite packet index. That formal statement does not assert a fully dependent chain of changing index types at arbitrary lengths. The finite rectangular illustration here uses exactly three factors.

For a single factor, the list of intermediate indices is empty and contributes that matrix entry once. For zero factors, the result is the identity on the common space. These conventions must not be replaced by an empty sum equal to zero.

> **Read this as:** Heterogeneous can mean different matrices or different spaces. State which one is allowed before using a path formula.

Plate III.9 / MODEL C / CANONICAL TRANSPORT HAS NO HOLONOMY

One vector, three exact frames

Chapter III, mathematical text: Theorem 4.1, Theorem 4.2

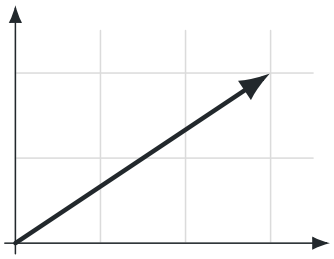
Let F_a synthesize chart coordinates into one ambient $\mathbb{Q}^2$. Choose

$$F_0 = I, \qquad F_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad F_2 = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}.$$

The same ambient vector $x = (3,2)^{\mathsf{T}}$ has chart coordinates $u_a = F_a^{-1}x$.

chart 0

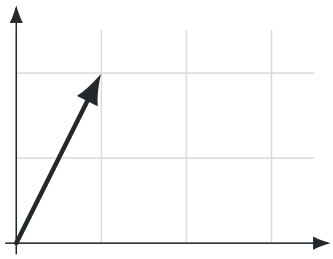
$(3,2)$



$F_0 u_0 = (3,2)^{\mathsf{T}}$

chart 1

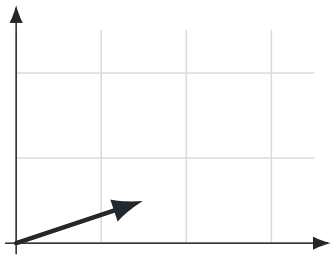
$(1,2)$



$F_1 u_1 = (3,2)^{\mathsf{T}}$

chart 2

$(\frac{3}{2}, \frac{1}{2})$



$F_2 u_2 = (3,2)^{\mathsf{T}}$

Each panel is a coordinate plane, not a different physical location. The numerical arrows differ because the frames differ. Multiplying each coordinate vector by its corresponding F_a recovers the same x.

Transport from chart a to chart b is forced:

$$J_{b \leftarrow a} = F_b^{-1} F_a.$$

The route $0 \to 1 \to 2$ agrees with the direct route because the middle frame cancels:

$$(F_2^{-1}F_1)(F_1^{-1}F_0) = F_2^{-1}F_0.$$

Returning to chart 0 cancels every remaining frame. The final coordinate vector is $(3,2)^{\mathsf{T}}$, not a sheared or rotated variant.

No metric is required

The frames need not be orthogonal. The second one is a shear, and the third has determinant two. Exact invertibility, rather than preservation of angles or lengths, gives the cocycle and loop identities. The same telescoping argument works for every finite chart path.

> **Read this as:** A canonical coordinate loop is a sequence of changes in description of one object. Nonidentity holonomy requires additional transport data, not merely nonorthogonal charts.

Plate III.10 / MODEL C / NATURALITY DOES NOT REQUIRE DERIVATIVES

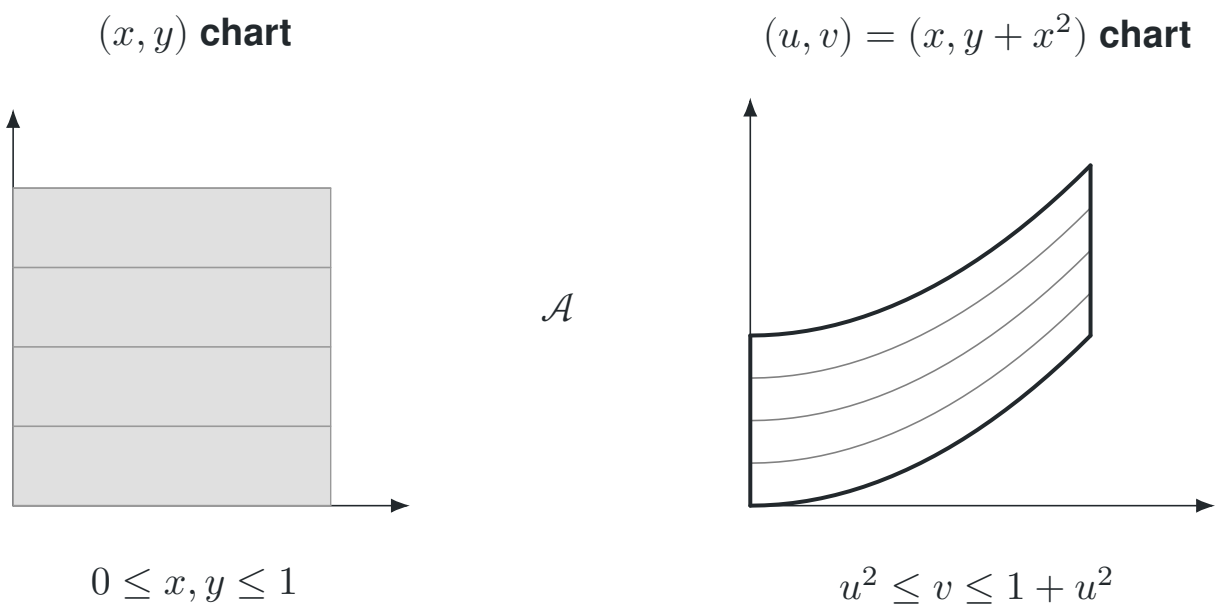
Nonlinear charts still cancel exactly

Chapter III, mathematical text: Proposition 10.1

Exact coordinate equivalences can be nonlinear. On $\mathbb{Q}^2$, use

$$\mathcal{A}(x,y) = (x, y + x^2), \qquad \mathcal{S}(u,v) = (u, v - u^2).$$

Both composites are the identity by direct substitution. The real drawing below is a geometric view of these same polynomial formulas.



Horizontal segments become parabolas, but no information is discarded. The drawing samples the displayed polynomial boundaries for rendering. Inverse identities and route equality are checked symbolically, not inferred from the curves.

Now represent the nonlinear operation $F(x,y) = (x + y, xy)$ in the new chart. Set $w = u + v - u^2$. Its correct coordinate formula is

$$\widehat{F}_b(u,v) = \mathcal{A}F\mathcal{S}(u,v) = (w,\ u(v - u^2) + w^2).$$

For $(x,y) = (1,2)$, applying F first gives $(3,2)$ and then $(3,11)$ in the new chart. Changing the input first gives $(u,v) = (1,3)$, and $\widehat{F}_b(1,3) = (3,11)$ as well.

The equality holds as a polynomial identity for all inputs. It is the naturality theorem for maps of sets: only the inverse laws for the coordinate maps are needed. No Taylor expansion, derivative, or higher chain rule is involved.

> **Read this as:** Nonlinearity alone does not create a coordinate defect. Reusing the old formula after changing coordinates would be an extra modeling choice, not exact conjugation.

Plate III.11 / MODEL D / THE MINIMAL AND REVERSIBLE INTERFACES

Reversibility is a real hypothesis

Chapter III, mathematical text: Definition 5.1, Definition 5.5

Choose the forward scalar transports $\Gamma_{1\leftarrow 0} = 2$, $\Gamma_{2\leftarrow 1} = 3$, and $\Gamma_{2\leftarrow 0} = 5$. Their multiplicative comparison between transport through two steps and direct transport is $D^{\times} = 6/5$. A loop also needs a declared return map $\Gamma_{0\leftarrow 2}$.

connection data	$\Gamma_{0\leftarrow 2}$	H	$D^{\times}$	$\Gamma_{0\leftarrow 2}\Gamma_{2\leftarrow 0}$
self normalization only	$\frac{1}{6}$	1	$6/5$	$\frac{5}{6}$
reversible	$\frac{1}{5}$	$\frac{6}{5}$	$6/5$	1

$$\Gamma_{1\leftarrow 0} = 2, \quad \Gamma_{2\leftarrow 1} = 3, \quad \Gamma_{2\leftarrow 0} = 5$$

The first row has H = identity at this triangle, but fails its cocycle.

All transports are invertible rational scalars and transport from each chart to itself is one. The first row is permitted by the minimal interface. The second uses the inverse of the direct forward edge and therefore satisfies that part of reversibility.

In the first row, choose the return scalar $1/6$. Then

$$H = (1/6) \cdot 3 \cdot 2 = 1, \qquad \Gamma_{2\leftarrow 1}\Gamma_{1\leftarrow 0} - \Gamma_{2\leftarrow 0} = 1.$$

This triangle loop is the identity even though its cocycle equation fails. The return is not the inverse of the direct transport: $(1/6)5 \neq 1$. Thus one cannot transfer the reversible equivalence between defect and holonomy to the minimal interface.

In the reversible row, the return is $1/5$. Now the loop is $6/5$, and the additive defect of one is converted into the loop obstruction $1/5$ by that inverse return map.

Two safeguards

Global flatness in the minimal interface implies trivial loops by using its cocycle on the whole route, including the return. Separately, in the reversible interface, triviality of each triangle loop is equivalent to that triangle's cocycle. Neither assertion says that one isolated loop certifies all triangles.

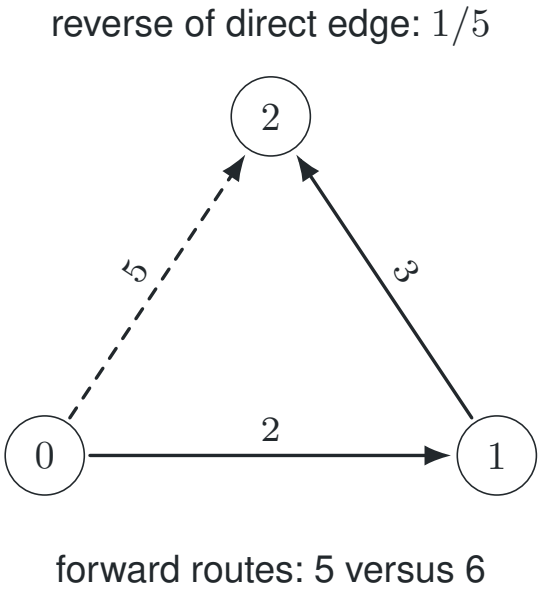
> **Read this as:** Self normalization means $\Gamma_{a\leftarrow a} = I$. Reversible additionally means $\Gamma_{a\leftarrow b} = \Gamma_{b\leftarrow a}^{-1}$. These are different assumptions.

Plate III.12 / MODEL D / AN EXACT REVERSIBLE TRIANGLE

Measure the mismatch at its base

Chapter III, mathematical text: Theorem 5.10, Section 9

Retain the reversible scalar model with forward values $2, 3, 5$ and all reverse values given by inverses. Two routes from chart 0 to chart 2 multiply the input by 6 and 5, respectively. The additive discrepancy equals the original input.



loop input 1 returns $6/5$

loops	returned value
0	1
1	$\frac{6}{5}$
2	$\frac{36}{25}$
3	$\frac{216}{125}$
4	$\frac{1296}{625}$

The dashed direct edge is a specified comparison map, not a missing edge. The table shows the exact effect of traversing the same loop repeatedly. These are rational values of this example, not an analytic convergence claim.

The defect lives at the target of the forward routes:

$$\Delta_{2,1,0} = 3 \cdot 2 - 5 = 1.$$

The holonomy lives back at chart 0:

$$H_{2,1,0} = 5^{-1} \cdot 3 \cdot 2 = \frac{6}{5}, \qquad H - I = \frac{1}{5} = \Gamma_{0 \leftarrow 2}\Delta.$$

For input 5, the direct output and the output after two steps are 25 and 30. Returning by $1/5$ gives 5 and 6. The target discrepancy of 5 has become a base discrepancy of 1. The types and locations of these differences matter even in an example in one dimension.

Because the return map is invertible, it cannot turn a nonzero defect into zero. This is the mechanism behind $\Delta = 0 \iff H = I$ in the reversible interface.

> **Read this as:** Holonomy compares paths after both have been returned to the same base fiber. It is the transported mismatch, not an untyped difference between outputs in different charts.

Plate III.13 / MODEL E / A SHEAR WITH NONTRIVIAL HOLONOMY

The identity can hide in the spectrum

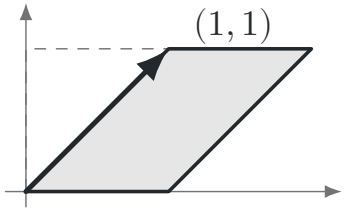
Chapter III, mathematical text: Definition 5.9, Definition 7.4

Put $\Gamma_{1\leftarrow 0} = H$, $\Gamma_{2\leftarrow 1} = I$, and $\Gamma_{2\leftarrow 0} = I$, with reverse maps given by inverses, where

$$H = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

The loop $0 \to 1 \to 2 \to 0$ then has exactly this shear as its holonomy.

The unit square under H



H

1	1
0	1

$H - I$

0	1
0	0

$$\operatorname{tr} H = 2, \quad \det H = 1, \quad (H - I)e_1 = e_0$$

The dashed square is the original set. The solid parallelogram is its exact image. The second coordinate basis vector moves to $(1, 1)^{\mathsf{T}}$. Its nonzero displacement is the first basis vector.

The usual scalar invariants can be misleading:

$$\operatorname{tr} H = 2, \qquad \det H = 1, \qquad \chi_H(t) = (t-1)^2.$$

These are also the invariants of the 2×2 identity matrix. Nevertheless,

$$\mathrm{Ob} = H - I = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \neq 0, \qquad \mathrm{Ob}\, e_1 = e_0.$$

A complete obstruction test must therefore retain the operator or a certificate strong enough to detect its nonzero action. Trace, determinant, and eigenvalue lists alone do not distinguish this loop from the identity.

What the picture certifies, and what it does not

The plotted corners are the images of the four corners of the unit square under the displayed matrix. The equation $\mathrm{Ob}\, e_1 = e_0$ is the exact witness. No smooth connection, curvature tensor, or continuum limit is used in this model. Its holonomy is the finite algebraic loop product. The mathematical text treats smooth transport separately under additional hypotheses.

> **Read this as:** Flatness is the equality $H = I$, not agreement of a short list of spectral summaries. A single explicit moved vector can disprove that equality.

Plate III.14 / MODEL E / CONJUGATION AND ITS LIMITS

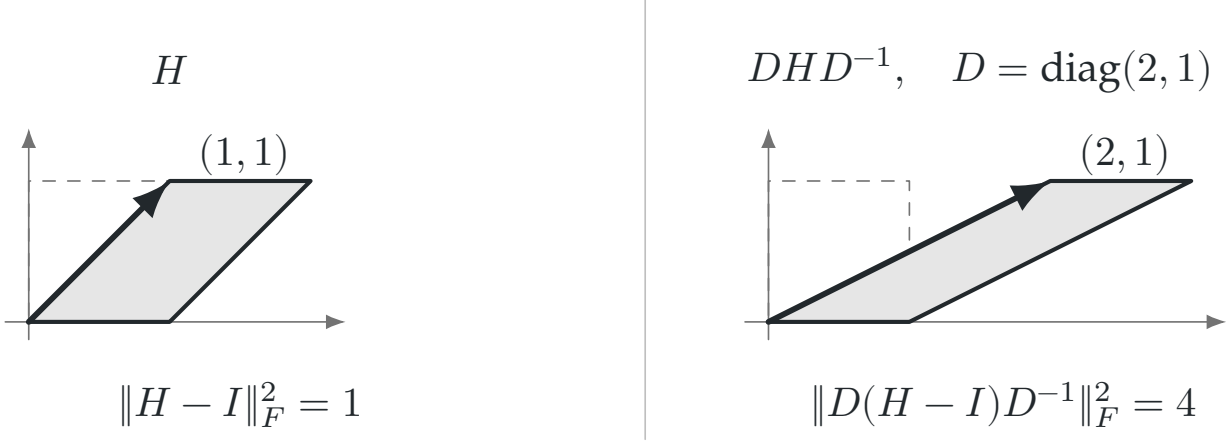
Gauge changes the picture, not flatness

Chapter III, mathematical text: Theorem 6.2

At the base chart, change coordinates by $D = \mathrm{diag}(2, 1)$. Transform every connection edge by the prescribed endpoint gauges. The loop transforms by conjugation, giving

$$H^D = DHD^{-1} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}.$$

The shear looks stronger in these fixed numerical axes, but the verdict on flatness is unchanged by gauge.



Both figures are exact images of a unit square under their displayed matrices. The Frobenius norms are computed in the indicated fixed coordinate bases. A nonorthogonal gauge need not preserve such a norm.

The rank of the obstruction is one on both sides. Trace, determinant, and characteristic polynomial are preserved under this conjugation. Conjugation also gives

$$D(H - I)D^{-1} = 0 \quad \Longleftrightarrow \quad H - I = 0.$$

An invertible coordinate change cannot turn a nonidentity loop into the identity. The fact that a particular residual norm changes does not contradict the invariance of exact vanishing.

The base chart determines the outer gauge

In a triangle product, adjacent intermediate gauges cancel:

$$(G_a \Gamma_{a \leftarrow c} G_c^{-1})(G_c \Gamma_{c \leftarrow b} G_b^{-1})(G_b \Gamma_{b \leftarrow a} G_a^{-1}) = G_a H G_a^{-1}.$$

Only the base gauge survives around the loop. For the additive defect, the surviving outer gauges instead belong to its different source and target fibers.

> **Read this as:** Exact nonvanishing is gauge invariant. The numerical size of a matrix entry or a chosen norm generally is not.

Plate III.15 / MODEL D / A FLAT CONNECTION WITH NONIDENTITY EDGES

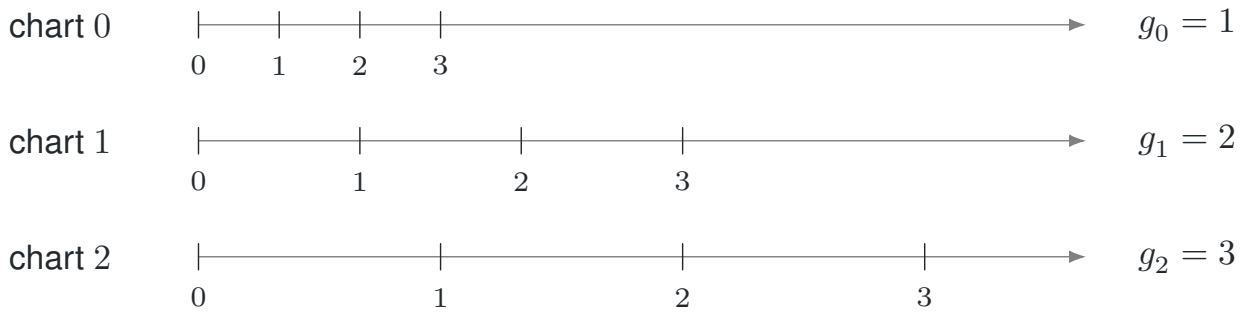
Flat is not the same as calibrated

Chapter III, mathematical text: Theorem 6.3

Make all three canonical charts literally identical, so $J_{b\leftarrow a} = 1$ over $\mathbb{Q}$. Choose scalar gauges $g_0 = 1$, $g_1 = 2$, and $g_2 = 3$, and define

$$\Gamma_{b\leftarrow a} = g_b g_a^{-1}.$$

The edge $0 \to 1$ is multiplication by two, although its canonical counterpart is the identity.



$$\Gamma_{1\leftarrow 0} = 2, \quad \Gamma_{2\leftarrow 1} = 3/2, \quad \Gamma_{2\leftarrow 0} = 3$$

The rulers display the chosen gauge scales: the marks are $g_a n$ for integer n. They visualize the supplied scalar factors, not an intrinsic metric. Relative transport is the ratio of the target and source scales.

The route through two steps $0 \to 1 \to 2$ has multiplier $(3/2)2 = 3$, equal to the direct route $0 \to 2$. More generally,

$$(g_c g_b^{-1})(g_b g_a^{-1}) = g_c g_a^{-1}$$

for every triple. Thus this connection is flat and reversible, but it is not calibrated to the original J, whose values are identities.

Recovering the gauge from one root

Choose a base chart o. The theorem constructs

$$G_a = \Gamma_{a\leftarrow o} J_{o\leftarrow a}, \qquad \Gamma_{b\leftarrow a} = G_b J_{b\leftarrow a} G_a^{-1}.$$

With $o = 0$ and $J = 1$, the recovered G_a are exactly $1, 2, 3$. The statement concerns the complete pairwise interface and its cocycle. It is not a topological claim about an arbitrary graph with missing direct comparisons.

> **Read this as:** Flatness makes the chosen transports independent of path. Calibration identifies them with one particular canonical transport before any gauge change.

Plate III.16 / MODEL F / PACKET BLOCKS FROM INPUT TO OUTPUT

Resolve the entire obstruction

Chapter III, mathematical text: Theorem 7.5, Corollary 7.6

Use three coordinate packets with projectors P_0, P_1, P_2, and set

$$H = I + 2E_{01} + 3E_{12}, \qquad \mathrm{Ob} = H - I.$$

Here H is invertible, so it can be the loop product of a reversible connection on three charts. Its entries outside the diagonal encode two effects that mix packets.

H

1	2	0
0	1	3
0	0	1

$\mathrm{Ob} = H - I$

0	2	0
0	0	3
0	0	0

	$\nu = 0$	$\nu = 1$	$\nu = 2$
$\mu = 0$	0	$2E_{01}$	0
$\mu = 1$	0	0	$3E_{12}$
$\mu = 2$	0	0	0

$$\mathrm{Ob}(4,5,6)^{\mathsf{T}} = (10,18,0)^{\mathsf{T}}$$

sum all nine blocks

The board on the right contains all nine actual embedded blocks $P_\mu \,\mathrm{Ob}\, P_\nu$. Row labels are output packets and column labels are input packets. Only the addresses $(0,1)$ and $(1,2)$ carry nonzero maps.

Completeness on both sides supplies the reconstruction:

$$\mathrm{Ob} = (P_0 + P_1 + P_2)\,\mathrm{Ob}\,(P_0 + P_1 + P_2) = \sum_{\mu,\nu} P_\mu \,\mathrm{Ob}\, P_\nu.$$

For $x = (4,5,6)^{\mathsf{T}}$, the input in packet 1 contributes $10e_0$ and the input in packet 2 contributes $18e_1$. Their sum is the full displacement $\mathrm{Ob}\, x$.

A block becomes a witness when it acts

The nonzero block $2E_{01}$ has the explicit source vector e_1, supported entirely in packet 1, and $P_0 \,\mathrm{Ob}\, e_1 = 2e_0 \neq 0$. Similarly, $P_1 \,\mathrm{Ob}\, e_2 = 3e_1 \neq 0$. The general theorem supplies such a witness classically when a nonzero block is known. The example displays it concretely.

> **Read this as:** An obstruction packet is not just a colored matrix cell. It consists of a specific label for a pair of packets and an exact nonvanishing statement about the corresponding map.

Plate III.17 / MODEL E / COMPLETENESS, TWO SIDES, AND OBLIQUE PACKETS

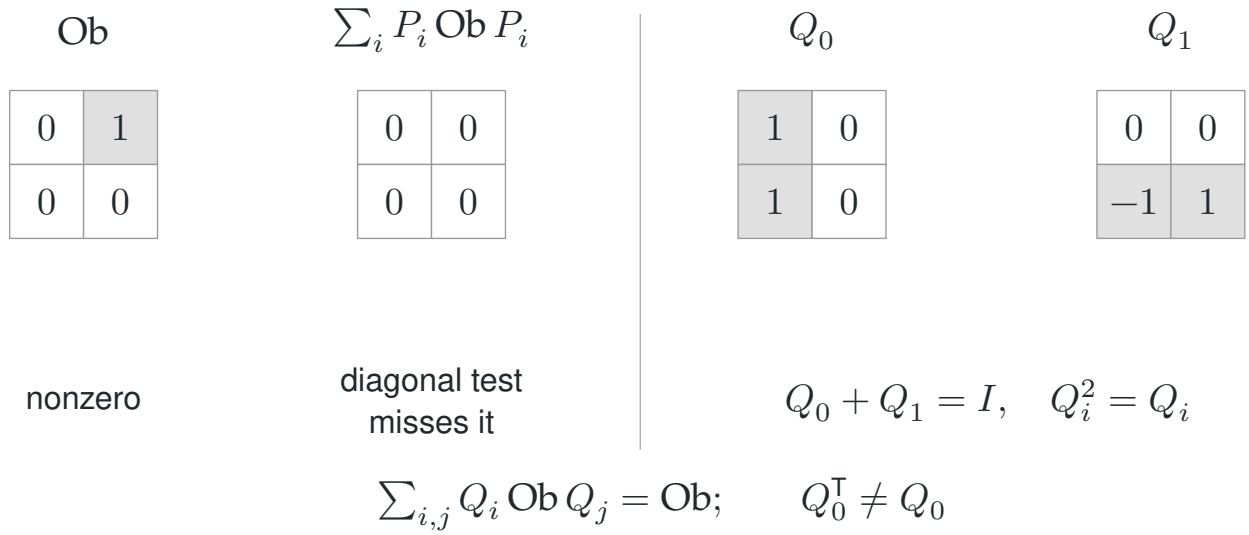
Why diagonal checks are insufficient

Chapter III, mathematical text: Theorem 7.5

Return to $\mathrm{Ob} = E_{01}$. With the two standard coordinate projectors, both diagonal blocks vanish:

$$P_0 \,\mathrm{Ob}\, P_0 = P_1 \,\mathrm{Ob}\, P_1 = 0.$$

The obstruction nevertheless survives in the block outside the diagonal $P_0 \,\mathrm{Ob}\, P_1 = E_{01}$.



The test of diagonal blocks alone on the left misses the whole obstruction. On the right, the displayed oblique projectors still form a complete exact resolution. They need not be orthogonal projections for the Euclidean inner product.

There is a second failure mode. Supplying only P_0 gives a family of idempotents but not a complete one. The single tested block is zero, while the omitted packet carries the missing input. The implication "all tested blocks vanish, therefore the operator vanishes" requires the completeness hypothesis.

Exact reconstruction does not require right angles

For the displayed Q_0, Q_1,

$$Q_0 + Q_1 = I, \qquad Q_i^2 = Q_i, \qquad Q_0 Q_1 = Q_1 Q_0 = 0,$$

but $Q_0^{\mathsf{T}} \neq Q_0$. Expanding both copies of I still proves

$$\sum_{i,j} Q_i \,\mathrm{Ob}\, Q_j = \mathrm{Ob}\,.$$

The block reconstruction identity itself uses the condition that the projectors sum to the identity. Idempotence is additionally used when constructing a witness that remains supported in one packet after projection. The general formal reconstruction statement does not require pairwise annihilation.

> **Read this as:** Test every block from input to output of a complete resolution. Testing only the diagonal, or only the packets supplied so far, answers a weaker question.

Plate III.18 / MODEL E / FIXED VERSUS TRANSPORTED PROJECTORS

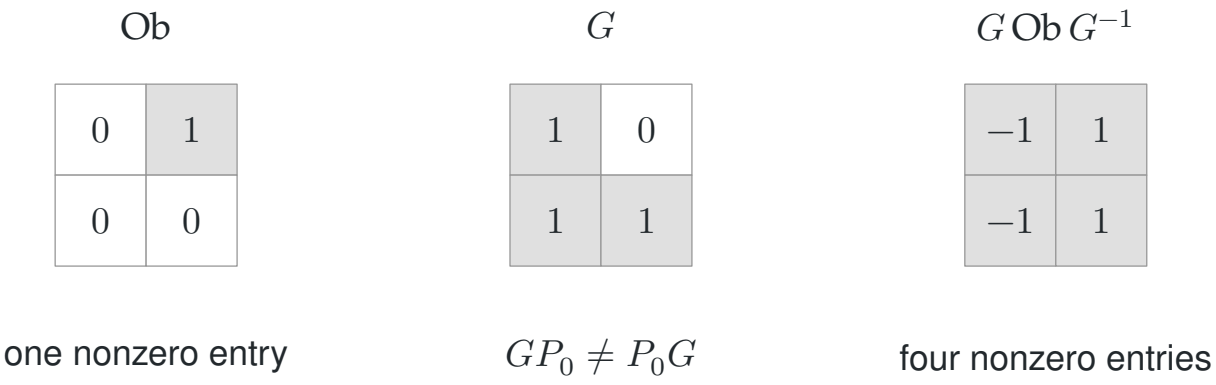
A gauge can move every packet label

Chapter III, mathematical text: Proposition 7.7

The diagonal gauge on Plate III.14 commuted with each standard projector. Now use the gauge that mixes packets

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad \mathrm{Ob} = E_{01}.$$

The total obstruction transforms by conjugation, but its pattern in the *fixed* standard packets changes.



The first obstruction has one nonzero entry. Its conjugate has four. Both have rank one and are nonzero. The change in fixed block pattern is not a change in whether the connection is flat.

Direct multiplication gives

$$G\,\mathrm{Ob}\,G^{-1} = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}.$$

For example, the original $(0,0)$ block vanishes, while the new $(0,0)$ block relative to the old coordinate projectors does not. This is why the theorem about preservation of each block's zero status requires $GP_i = P_iG$.

Transporting the packet frame as well

Define $Q_i = GP_iG^{-1}$. Then the identities cancel exactly:

$$Q_i(G\,\mathrm{Ob}\,G^{-1})Q_j = G(P_i\,\mathrm{Ob}\,P_j)G^{-1}.$$

Relative to these transported packets, the corresponding labeled blocks again have matching vanishing status. These are the same oblique projectors displayed on Plate III.17. One must distinguish moving the operator alone in a fixed resolution from moving the entire resolved description.

> **Read this as:** The total obstruction is intrinsic under gauge. Its location in a fixed packet grid is not, unless the gauge preserves that grid.

Plate III.19 / MODEL G / THE QUANTIFIERS ARE THE CONTENT

An obstruction rejects one candidate

Chapter III, mathematical text: Theorem 7.2

Let the candidates be rational numbers and the packet labels be $0, 1$. Define

$$r(x, 0) = x^2 - 1, \qquad r(x, 1) = x - 1.$$

A candidate is a solution when both residuals vanish. The five candidate columns below are a display of this explicit residual function, not a search over an unspecified class.

candidate x	-2	-1	0	1	2
$r(x, 0) = x^2 - 1$	3	0	-1	0	3
$r(x, 1) = x - 1$	-3	-2	-1	0	1
	rejected	rejected	rejected	solution	rejected

$$x = -1 : \quad r(-1, 0) = 0, \quad r(-1, 1) = -2$$

A shaded cell is a nonzero exact residual. At $x = -1$, the first equation passes but the second supplies an obstruction. At $x = 1$, both pass. The rejected candidate does not obstruct the successful one.

The statement proved by the generic certificate has the form

$$\neg(\forall p,\ r(x, p) = 0) \quad \Longleftrightarrow \quad \exists p,\ r(x, p) \neq 0.$$

The same candidate x appears on both sides. A certificate consists of the packet label and its nonvanishing proof. The general converse uses classical reasoning and does not itself provide an algorithm to find the label.

What would be required to rule out all solutions?

One would need the stronger assertion $\forall x \exists p,\ r(x, p) \neq 0$. The value $x = 1$ disproves that assertion in this example. A witness for $x = -1$ supplies no evidence for it. Likewise, finding an obstruction block for one triangle of one chosen connection does not classify every possible connection.

The term *obstruction* in this generic interface does not automatically mean a cokernel class, a cohomology group, a norm bound, or a minimal certificate. Those require additional constructions and theorems.

> **Read this as:** Keep the candidate attached to the witness. The theorem's strength is its exact logical scope, not a universal existence or nonexistence claim.

Plate III.20 / MODEL H / ACTUAL SYNTAX, ARBITRARY SEMANTICS

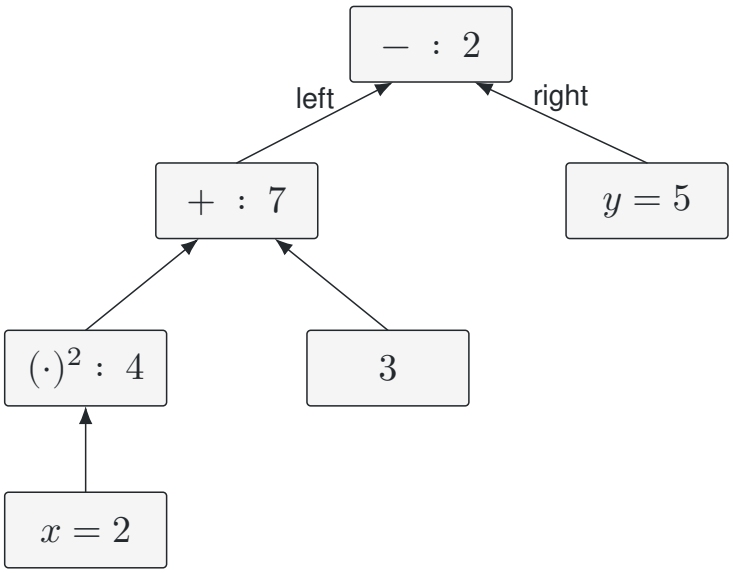
A tree reads its named leaves

Chapter III, mathematical text: Definition 11.1, Theorem 11.2

Use the grammar's constant, input, unary, and binary constructors to build

$$t = (x^2 + 3) - y.$$

Here "square," "add," and "subtract" are explicitly supplied meanings of operation symbols. At $x = 2, y = 5$, every node value can be evaluated exactly.



$\text{inputs}(t) = \{x, y\}$

unused input: -100 or 100

same value: 2

$x - x : \{x\}$ syntactically

but always zero

The rooted tree records the order of input slots. Arrows indicate evaluation from children to parent, not chart transport. An environment may contain additional variables, but this tree does not read them.

The input set is $\text{inputs}(t) = \{x, y\}$. If two environments agree at x and y, structural induction proves that the two evaluations agree: input leaves agree, constants agree, and every parent applies the same supplied operation to equal child values.

Syntactic locality is not minimal dependence

The tree $x - x$ has input set $\{x\}$, although its value is always zero under ordinary subtraction. Locality provides a sufficient agreement condition. It does not compute the smallest set of semantically relevant variables. Nor does it prove that two different trees are equivalent.

The semantics may act on arbitrary values. For example, a binary concatenation symbol applied to strings "ab" and "cd" gives "abcd." No linearity or differentiability is required. Conversely, swapping the ordered inputs of the subtraction root above changes the value from 2 to -2.

> **Read this as:** The proved tree theorem controls which input labels are read. It does not supply a nonlinear chain rule, a stability estimate, or a solution of a tree equation.

Plate III.21 / MODEL H / EXACT COORDINATES VERSUS A CHOSEN RULE

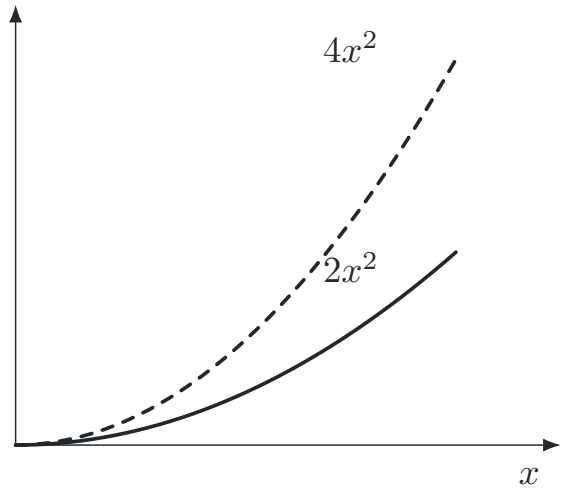
Where a nonlinear defect can enter

Chapter III, mathematical text: Proposition 10.1, Formalization target 11.7, Theorem 20.2

Take the unary operation $f(x) = x^2$ over $\mathbb{Q}$. Suppose both input and output transport are chosen as multiplication by two, while the same numerical formula f is reused at the target. The two routes give

$$2f(x) = 2x^2, \qquad f(2x) = 4x^2.$$

Their difference is $-2x^2$, which is generally nonzero.



x	0	1	2
$2f(x)$	0	2	8
$f(2x)$	0	4	16
$2f(x) - f(2x)$	0	-2	-8

correct chart formula: $f_b(v) = v^2/2$

Solid: $2x^2$. Dashed: $4x^2$. The curves are a real visualization of exact polynomial formulas. The table contains exact evaluations over the rationals. The mismatch is caused by the specified combination of transport and formula.

For a canonical coordinate change with input and output chart $v = 2x$, the target formula is not still v^2. It is

$$\hat{f}_b(v) = 2f(v/2) = \frac{v^2}{2}.$$

Consequently $\hat{f}_b(2x) = 2x^2$, and the exact coordinate defect vanishes. This is the same distinction illustrated in the example in two dimensions on Plate III.10.

An example is not a general propagation theorem

The polynomial calculation here is exact and checked symbolically. It explains a chosen node defect, while the mathematical text develops composition identities, iterated defects, fixed point stability, and derivative defects under explicit hypotheses. Those broader ordinary proofs must be read on their own terms. This one example and the finite Lean locality theorem do not certify them.

> **Read this as:** A nonzero nonlinear route difference must come from identifiable extra data: a changed transport, an approximation, a restricted model, or a reused formula that was not conjugated correctly.

Plate III.22 / FINITE EXAMPLES / WHY MULTILINEARITY MATTERS

Mixed terms are not optional

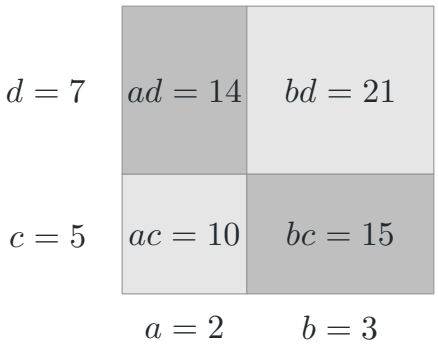
Chapter III, mathematical text: Formalization target 11.6, Theorem 18.3

A bilinear node distributes over two independently resolved inputs:

$$(a+b)(c+d) = ac + ad + bc + bd.$$

The four terms correspond to all ordered choices of one input packet from each slot. Keeping only matching packet labels loses the mixed terms.

$$(2+3)(5+7) = 60$$



Diagonal pieces are not the whole product

$$ac + bd = 10 + 21 = 31$$

$$ad + bc = 14 + 15 = 29$$

$$(2+3)^2 = 25 \neq 2^2 + 3^2 = 13$$

missing square term: $2 \cdot 2 \cdot 3 = 12$

The rectangle has width $2 + 3$ and height $5 + 7$, with differently scaled horizontal and vertical display units. Each cell is labeled by its actual product. The lighter diagonal pair contributes only 31 of the total 60.

For a unary square, inserting $a+b$ does not give a^2+b^2. The extra $2ab$ arises because the operation is nonlinear in its single argument. It can be analyzed by representing the square using a binary multiplication node with a repeated input, but that is an additional choice of syntax and semantics.

The broader tree theorem needs its hypotheses

General packet tree expansion needs correctly typed input slots, finite sets of packet labels at each node, and multilinearity in every slot used for distributive expansion. Empty inputs and unary nodes need explicit conventions. Slot order must survive even when values or operations do not commute. The mathematical text supplies rooted tree reconstruction and pruning results in its ordinary mathematical development.

This page supplies elementary exact examples of the issue. The finite algebraic Lean contract remains narrower: it contains syntactic input locality, not the general dependent tree expansion or the mathematical text's analytic convergence results.

Read this as: A branching picture does not justify throwing away mixed interactions. Exact packet expansion requires the appropriate multilinearity and typing, not merely a diagram that appears finite.

Plate III.23 / CAPSTONE / A COMPLETE FINITE DIAGNOSIS

From loop data to a packet witness

Chapter III, mathematical text: Theorem 5.10, Theorem 6.2, Corollary 7.6

Supply a reversible connection on three copies of $\mathbb{Q}^2$ by choosing

$$\Gamma_{1\leftarrow 0} = A, \qquad \Gamma_{2\leftarrow 1} = B, \qquad \Gamma_{2\leftarrow 0} = C = I,$$

and declaring every reverse edge to be the inverse. This time both nontrivial forward matrices are retained.

A

1	1
0	1

B

1	0
1	1

$H = BA$

1	1
1	2

$H - I$

0	1
1	1

$$e_0 = (1,0) \;\mapsto\; (1,0) \;\mapsto\; (1,1) \;\mapsto\; (1,1)$$

$$P_1(H - I)e_0 = e_1 \neq 0$$

---

replace direct edge $C = I$ by $C = BA$: then $C^{-1}BA = I$

The vector row starts with the source witness e_0 and follows it through A, then B, then C^{-1}. Returning to the base produces $(1,1)^{\mathsf{T}}$, whose displacement from the input is e_1.

The calculation gives

$$\Delta = BA - C = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \qquad H = C^{-1}BA = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Choose the standard packet projectors. The $(1,0)$ block is nonzero, and the explicit candidate vector satisfies $P_1(H - I)e_0 = e_1$. This proves nonflatness of the supplied connection at the supplied triangle.

Repairing data is not the same as changing gauge

Replacing the direct map $C = I$ by $C = BA$, and updating its reverse, makes the new loop equal to I. That changes the connection. It cannot be a gauge of the original nonflat connection, because gauge preserves whether holonomy is the identity. For this complete example on three charts all cocycles then hold. Changing one edge in a larger network would require checking the other affected loops as well.

> **Read this as:** Declare the data, multiply in traversal order, form the defect, resolve all blocks, and exhibit an input on which one block acts nontrivially. Each step has a checkable mathematical object.

Plate III.24 / EVIDENCE / EXAMPLES, PROOFS, AND FORMAL SCOPE

Read the evidence with the pictures

Chapter III, mathematical text: Section 12

The 24 worked plates and 22 exact figures connect finite models with the main text's broader nonlinear and analytic theory.

Read each evidence record in its scope

Evidence	What it supports
Formal development	The finite declarations and the additions in Sections 16 and 17 were compiled in the pinned environment. The Lean guide records the source scope, axiom audit, and kernel replay.
Ordinary mathematics	The mathematical text's expanded proofs, including nonlinear, obstruction, lifting, and smooth transport results. Inclusion in that text does not add these results to the finite Lean contract.
Exact model checks	The recorded run passed all 124 exact checks and rejected all four deliberately false controls. The scope is the displayed finite matrices, paths, polynomials, candidate tables, and trees.
Figure regeneration	All 22 editable TikZ figures are supplied with their exact model data and generators. The original figure sources and their earlier regeneration record remain preserved.

The computation record also contains 18,630 exact assertions for the examples in Sections 16 and 17. These finite calculations support the displayed mechanisms. General claims rest on their mathematical proofs and, where mapped, their Lean declarations.

Boundaries that remain in force

The formal edge system uses a single scalar coefficient semiring. The heterogeneous path theorem for arbitrary lengths uses a common packet index, with forms allowing changing types for two and three factors. General dependent tree expansion and analytic convergence lie outside this finite formal contract. The mathematical text states and proves its broader results with their own hypotheses.

The generic obstruction theorem rejects its named candidate. A nonzero holonomy block rejects flatness for its named loop data. Neither conclusion supplies a universal classification, a new arithmetic realization, or an algorithmic search guarantee.

Replay the worked layer

Check the exact examples with `python checks/verify_examples.py`.

Regenerate the figures with `python source/generate_graphics.py`.

Use a temporary copy to preserve the recorded evidence. The finite algebraic source distribution supplies the formal sources separately.

> **Read this as:** Use the figures to understand the mechanisms, the mathematical text to read the mathematical proofs, and the Lean guide to identify the exact formal contract. These are related records with distinct scopes.

CHAPTER IV

Projector Resolved Seed Calculus IV

Cyclic analytic functions, reciprocal families, and spectral zero data

The focus shifts from the general packet language to a particular analytic family.

A cyclic analytic family connects coefficient selection, differential transport, reciprocal polynomials and spectral zero data.

IN THIS CHAPTER

Mathematical text 280

Visual studies 355

Formal companion 1815

Sources and references 2672

Scope

The analytic statements retain their branch choices, multiplicities and integer-order hypotheses. Universal zero simplicity and a zeta functional equation remain outside these results.

The next chapter turns to signals, where filters, packet trees and reconstruction make changes of resolution concrete.

How to use this volume

Begin with cyclic projection and the signed differential cycle. The reciprocal sections then build parameter polynomials, Appell systems, and theta polynomials from a unit series. The second half follows the same denominator through large zero geometry, canonical products, and zero zeta functions.

The visual atlas at the end of this chapter supplies twenty-six worked plates. The companion Lean guide follows the formal declarations and their hypotheses. The main text contains sixty-eight theorem-like results and ninety-five numbered records.

Overview

Fix a positive integer N. Root of unity projection of the exponential produces residue-class sections that, for $\lambda \neq 0$, form a basis for $y^{(N)} = \lambda^N y$. The signed sections for $y^{(N)} = -y$ yield the denominator D_N and its reduced form Δ_N. We derive reciprocal coefficient polynomials, Bell and Newton recurrences, an additive parameter law, and residue section Appell and theta families. For $N \geq 2$, Δ_N has order $1/N$. We specialize classical two term zero theory for exponential sums to the regular exponent polygon. In this equal-weight family, conjugation symmetry combined with unique near-model localization places all sufficiently large zeros exactly on the symmetry rays, and direct expansion gives their leading displacement and derivative sign. The classical perimeter law yields zero-counting asymptotics with explicit leading constants and bounded error. These results give genus-zero products, the standard diagonal Fredholm representation, zero-power-sum formulas for reciprocal and logarithmic coefficients, nearest-zero coefficient asymptotics, and meromorphic continuation of the symmetry-orbit zero zeta by comparison with the Hurwitz zeta function. Complex powers are interpreted either formally or through a fixed logarithm on a zero-free domain. This chapter treats only this cyclic analytic family. Arithmetic interpolation, general nonlinear trees, general resolvent theory, and abstract completion are reserved for Chapters IX, III, VI, and X.

Formal scope and evidence

The recorded formal contract covers all sixty-eight theorem-like manuscript results under their stated hypotheses. The formalization ledger identifies the corresponding declarations.

Imported standard fact 11.1 is used in the ordinary manuscript growth argument. The Lean growth-order proofs establish that step separately. Open questions, branch choices, multiplicities, and the hypotheses of coefficient asymptotics remain explicit. Finite computations support the worked models. The companion Lean guide records the replay using pinned sources, declaration-kind checks, axiom audit, and their limits.

1 Introduction

For a power series

$$F(z) = \sum_{m \geq 0} a_m z^m$$

and a positive integer N, the coefficients may be separated according to the residue of m modulo N. Root of unity averaging gives exact projection operators for this separation. This elementary construction is usually called a multisection or a root filter. It appears throughout the classical theory of generating functions and special functions. Background on multisections, hypergeometric functions, and related coefficient systems may be found in [Rai60, AAR99, Com74].

Chapters I and II developed typed seed grammars, exact projector packets, and finite residue-projector algebra, while Chapter III treated typed edge operators and exact coordinate transport [P.26a, P.26b, P.26c]. The root filter used below is the finite cyclic scalar specialization of that earlier projector calculus. The signed companion matrix is likewise a concrete cyclic instance, not a replacement for Chapter III's general typed-edge theory.

This chapter follows the consequences of applying the cyclic projections to the exponential function. The projected exponentials form a finite differential system. After one sign is inserted around the derivative cycle, they form a basis for

$$y^{(N)} = -y.$$

The section with initial value one is

$$\mathsf{D}_N(z) = \sum_{n \geq 0} (-1)^n \frac{z^{Nn}}{(Nn)!}.$$

Its reciprocal powers produce polynomial coefficient families. Its logarithmic derivative produces a scalar higher Riccati identity. The same reciprocal construction applied to every exponential section gives normalized residue section functions and ordinary Appell polynomial sequences. These cyclic exponential sections and their matrix addition laws belong to the generalized hyperbolic function literature [Sch92, Mul05]. Appell and binomial-type sequences are discussed in [Rom84, Com74].

The second half of this chapter concerns the zeros of D_N. The finite exponential sum

$$\mathsf{D}_N(z) = \frac{1}{N} \sum_{\eta^N = -1} e^{\eta z}$$

has a regular polygon of exponents. The general relation between an exponent polygon and the zero strips of an exponential sum goes back to Pólya and Langer. Later developments include the work of Moreno and the treatment in Levin's monograph [Pól20, Lan31, Mor73, Lev80]. Bleher and Mallison give a particularly close general result: for convex-polygon exponential sums they locate all sufficiently large zeros in two term strips and prove exponential localization and simplicity [BM06]. The regular polygon and conjugation symmetry of the present family specialize that theory to N exact rays. We give a self-contained specialization because its notation is used throughout the product and zeta theory, and we compute the leading displacement term for this family. The resulting counting formulas have explicit leading constants and bounded error.

The rotational symmetry is more useful after the change of variable $w = z^N$. There is an entire function

$$\Delta_N(w) = \sum_{n\geq 0}(-1)^n \frac{w^n}{(Nn)!}$$

with

$$\mathsf{D}_N(z) = \Delta_N(z^N).$$

For $N \geq 2$, the order of Δ_N is $1/N$. Its Hadamard product therefore has genus zero. The logarithm of this product identifies the transfer tangent coefficients with power sums of the reduced zeros. The large zero formula also compares the reduced zero zeta with a Hurwitz zeta function. Standard material on entire functions, canonical products, and exponential sums appears in [Boa54, Lev80, Tit39]. We use singularity analysis in the form described in [FS09] and the classical continuation of the Hurwitz zeta function from [Apo76].

The cyclic exponential sections and their matrix addition laws have a classical literature. The tools used below include Appell and binomial-type sequences, canonical products, singularity analysis, and Fredholm determinants. These tools are also standard. Our aim is to organize this symmetric specialization and link its reciprocal coefficients to reduced-zero data. No historical priority is asserted for the component constructions.

2 A map of the construction

The central idea

The same object is read in different coordinates. Selecting residues makes its Taylor coefficients visible. A signed shift makes the differential system finite. Passing to $w = z^N$ removes rotational repetition from its zeros. The formulas below identify these descriptions under their stated hypotheses.

Two variables appear throughout this chapter. The variable z carries the differential equation and the rotational symmetry. The reduced variable

$$w = z^N$$

records one representative from each rotational orbit. Confusing these variables hides the main simplification. The denominator D_N has order one as a function of z, while Δ_N has order $1/N$ as a function of w.

The analytic objects are introduced in the following order. First,

$$\mathsf{E}_{N,j}(z) = \sum_{n\geq 0} \frac{z^{Nn+j}}{(Nn+j)!}$$

is the residue j part of the exponential. Second,

$$\mathsf{D}_{N,j}(z) = \sum_{n\geq 0}(-1)^n \frac{z^{Nn+j}}{(Nn+j)!}$$

is the signed basis for $y^{(N)} = -y$. The section $\mathsf{D}_N = \mathsf{D}_{N,0}$ is the denominator used in the reciprocal family

$$\mathsf{S}_N^{(r)} = \mathsf{D}_N^{-r}.$$

Set $\mathsf{T}_N = -\mathsf{D}_N'/\mathsf{D}_N$. Equivalently, T_N is the logarithmic derivative of $\mathsf{S}_N^{(1)}$.

A different reciprocal family comes from normalizing each exponential section at the origin. The function

$$\mathsf{C}_{N,j}(z) = \frac{z^j/j!}{\mathsf{E}_{N,j}(z)}$$

is a unit series in z^N. Its powers, after multiplication by e^{xz}, give Appell polynomials. The theta family couples a scaled argument with a reciprocal order.

The zero theory begins only after these algebraic relations have been established. The factorization

$$\mathsf{D}_N(z) = \Delta_N(z^N)$$

turns every orbit of N zeros in the z plane into one zero in the w plane. The large zeros in the z plane are located by two dominant exponentials. The zeros in the w plane then give a genus zero product. The logarithm of that product returns to the coefficient theory through

$$\text{zero power sums} = \text{logarithmic coefficients}.$$

This equality is the bridge between the first and second halves of this chapter.

All powers with a complex parameter are first defined as formal series at the origin. To represent this germ analytically, use a simply connected zero-free region containing the origin and the logarithm normalized to vanish there. On other regions, specify a branch separately. Integer reciprocal powers are globally meromorphic and do not require a branch choice.

The main statements fall into five groups. The finite algebra model in section 6 explains the derivative cycles and addition laws through one exponential identity. The coefficient theory in section 7 gives two complementary recurrences, one from the original denominator and one from its logarithm. The Appell theory in section 9 supplies both formal differential-operator formulas and a convolution law in the space and order parameters. The zero analysis in section 11 specializes classical two term exponential-sum theory, uses symmetry to place the uniquely localized large zeros on the rays, and computes a leading correction. The product theory in section 12 gives the standard diagonal Fredholm encoding of the genus zero product.

This chapter keeps formal and analytic statements separate. A formal identity needs only the unit constant term of the relevant series. A holomorphic complex power also needs a zero-free domain and a chosen logarithm. A meromorphic coefficient asymptotic is stated only for positive integral reciprocal order unless an additional branch domain is specified.

Part I Cyclic resolution and differential transport

3 Cyclic projection of analytic functions

A route through this section

The root filter asks which coefficient addresses belong to a packet. It is an exact selector. Multiplication then asks how two selected packets interact. Their residue labels add modulo N, so mixed inputs remain part of the product.

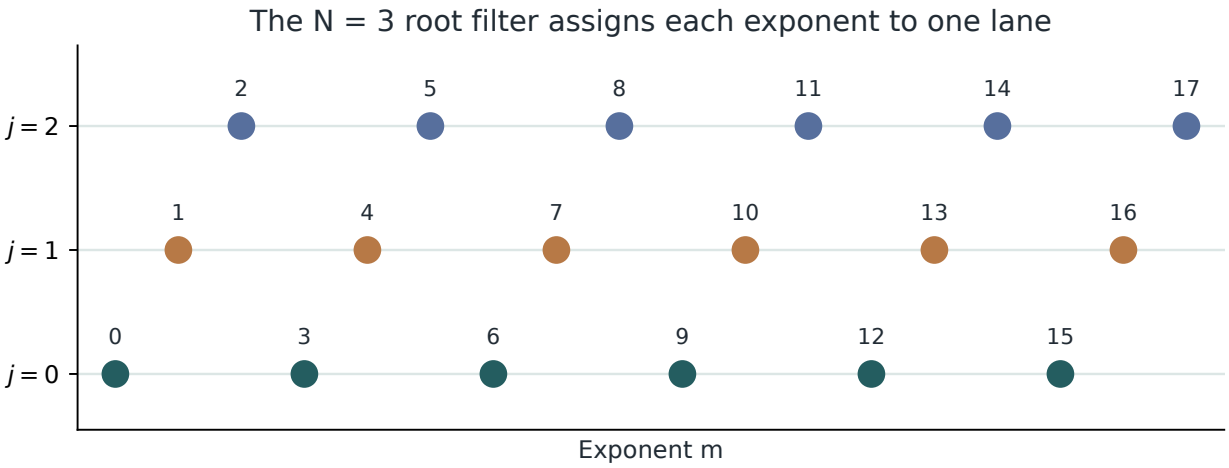


Figure 1. An exact address model for $N = 3$. Every displayed exponent belongs to one residue lane. The drawing records labels rather than coefficient magnitudes.

Fix $N \geq 1$ and write

$$\omega = e^{2\pi i/N}.$$

Let Ω be a disk centered at the origin, with the whole plane allowed, and assume that Ω is invariant under multiplication by ω. For a holomorphic function F on Ω, define

$$(\mathcal{B}_{N,j}F)(z) = \frac{1}{N}\sum_{\ell=0}^{N-1} \omega^{-j\ell} F(\omega^{\ell} z), \qquad 0 \leq j < N.$$

The same formula is valid for formal power series.

Theorem 3.1 (Root filter). *If*

$$F(z) = \sum_{m \geq 0} a_m z^m,$$

then

$$\mathcal{B}_{N,j}F(z) = \sum_{m \equiv j \bmod N} a_m z^m.$$

The operators satisfy

$$\mathcal{B}_{N,i}\mathcal{B}_{N,j} = \delta_{ij}\mathcal{B}_{N,j}$$

and

$$\sum_{j=0}^{N-1} \mathcal{B}_{N,j} = I.$$

Proof. Substitution of the power series gives

$$\mathcal{B}_{N,j}F(z) = \sum_{m \geq 0} a_m z^m \left(\frac{1}{N} \sum_{\ell=0}^{N-1} \omega^{(m-j)\ell} \right).$$

The inner average is one when $m \equiv j \bmod N$ and zero otherwise. The operator identities follow from the disjoint residue classes. □

Proposition 3.2 (Covariance characterization)**.** *A holomorphic function F belongs to the range of $\mathcal{B}_{N,j}$ if and only if*

$$F(\omega z) = \omega^j F(z).$$

Proof. Every monomial selected by $\mathcal{B}_{N,j}$ has the stated covariance. Conversely, write $F = \sum_k \mathcal{B}_{N,k}F$. Replacing z by ωz gives

$$\sum_k \omega^k \mathcal{B}_{N,k}F = \omega^j \sum_k \mathcal{B}_{N,k}F.$$

Application of $\mathcal{B}_{N,k}$ gives

$$(\omega^k - \omega^j)\mathcal{B}_{N,k}F = 0.$$

The powers $1, \omega, \ldots, \omega^{N-1}$ are distinct, so $\mathcal{B}_{N,k}F = 0$ for $k \neq j$. □

Multiplication is not diagonal in these coordinates. The residue labels add modulo N. This is the scalar cyclic instance of the packet product from Chapter I.

Theorem 3.3 (Cyclic product law)**.** *If F and G are holomorphic on the same rotation-invariant domain, then*

$$\mathcal{B}_{N,k}(FG) = \sum_{\substack{0 \leq i,j < N \\ i+j \equiv k \bmod N}} (\mathcal{B}_{N,i}F)(\mathcal{B}_{N,j}G).$$

Proof. Write $F = \sum_i F_i$ and $G = \sum_j G_j$, where $F_i = \mathcal{B}_{N,i}F$ and $G_j = \mathcal{B}_{N,j}G$. Their covariance gives

$$(F_i G_j)(\omega z) = \omega^{i+j} F_i(z) G_j(z),$$

so $F_i G_j$ belongs to the packet labeled by $i + j$ modulo N. Expanding the finite product and applying $\mathcal{B}_{N,k}$ gives the formula. □

Remark 3.4. Every identity in this section holds formally. In the holomorphic setting, the finite projector sums preserve local uniform convergence. No choice of branch is involved.

4 The transfer exponential basis

Definition 4.1. For $\lambda \in \mathbb{C}$ and $0 \le j < N$, define

$$\mathsf{E}_{N,j}(\lambda z) = \mathcal{B}_{N,j}(e^{\lambda z}) = \sum_{n \ge 0} \frac{(\lambda z)^{Nn+j}}{(Nn+j)!}.$$

Theorem 4.2 (Differential cycle and initial data)**.** *For $1 \le j < N$,*

$$\frac{d}{dz}\mathsf{E}_{N,j}(\lambda z) = \lambda \mathsf{E}_{N,j-1}(\lambda z),$$

while

$$\frac{d}{dz}\mathsf{E}_{N,0}(\lambda z) = \lambda \mathsf{E}_{N,N-1}(\lambda z).$$

Thus

$$\frac{d^N}{dz^N}\mathsf{E}_{N,j}(\lambda z) = \lambda^N \mathsf{E}_{N,j}(\lambda z).$$

For $0 \le k < N$,

$$\frac{d^k}{dz^k}\mathsf{E}_{N,j}(\lambda z)\Big|_{z=0} = \delta_{kj}\lambda^j.$$

If $\lambda \neq 0$, the N functions $\mathsf{E}_{N,j}(\lambda z)$ form a basis of the solution space of $y^{(N)} = \lambda^N y$.

Proof. Termwise differentiation gives the cyclic derivative formulas. One full cycle contributes λ^N. The first nonzero term of $\mathsf{E}_{N,j}(\lambda z)$ is $(\lambda z)^j/j!$, which gives the initial data. The initial value vectors are linearly independent when $\lambda \neq 0$. The solution space of an order N constant coefficient equation has dimension N, so these functions form a basis. □

Corollary 4.3 (Wronskian of the transfer basis)**.** *Assume $N \ge 2$. For $\lambda \neq 0$, the Wronskian of the ordered basis*

$$\mathsf{E}_{N,0}(\lambda z), \dots, \mathsf{E}_{N,N-1}(\lambda z)$$

is the constant

$$\lambda^{N(N-1)/2}.$$

Proof. At the origin, the derivative matrix is diagonal with entries

$$1, \lambda, \dots, \lambda^{N-1}.$$

For an explicit determinant proof, write each basis function as a finite sum of exponentials with exponents $a_\ell = \lambda\omega^\ell$. Its derivative matrix factors as

$$W(z) = V\ \mathrm{diag}(e^{a_\ell z})\, C, \qquad V_{k\ell} = a_\ell^k,$$

where C is the constant matrix of Fourier weights. Thus

$$\det W(z) = e^{z\sum_\ell a_\ell} \det W(0) = \det W(0),$$

because $N \geq 2$ implies $\sum_{\ell} \omega^{\ell} = 0$. The product of the diagonal initial values is the stated power of λ. □

Corollary 4.4 (Addition law)**.** *For* $0 \leq k < N$,

$$\mathsf{E}_{N,k}(\lambda(z+w)) = \sum_{\substack{0 \leq i,j < N \\ i+j \equiv k \bmod N}} \mathsf{E}_{N,i}(\lambda z)\mathsf{E}_{N,j}(\lambda w).$$

Proof. Expand $e^{\lambda(z+w)} = e^{\lambda z}e^{\lambda w}$ as a bivariate power series. A term $z^a w^b$ contributes to the kth section exactly when $a + b \equiv k \bmod N$. Grouping first by $i \equiv a \bmod N$ and $j \equiv b \bmod N$ gives the displayed sum. □

The basis also has a scalar hypergeometric form.

Theorem 4.5 (Hypergeometric form)**.** *Let*

$$A_{N,j} = \left\{ \frac{j+m}{N} : 1 \leq m \leq N \text{ and } j+m \neq N \right\}.$$

Then

$$\mathsf{E}_{N,j}(\lambda z) = \frac{(\lambda z)^j}{j!} {}_0F_{N-1}\left(\begin{matrix} - \\ A_{N,j} \end{matrix} , \frac{(\lambda z)^N}{N^N} \right).$$

Proof. The multiplication formula for the gamma function gives

$$(Nn+j)! = j! N^{Nn} n! \prod_{a \in A_{N,j}} (a)_n.$$

Indeed, the N Pochhammer factors supplied by the multiplication formula are $((j+m)/N)_n$ for $1 \leq m \leq N$. The excluded factor is $((j+N-j)/N)_n = (1)_n = n!$. Insertion into the defining series gives the stated hypergeometric expansion. □

Example 4.6. For $N = 1$, the only section is $e^{\lambda z}$. For $N = 2$,

$$\mathsf{E}_{2,0}(z) = \cosh z, \qquad \mathsf{E}_{2,1}(z) = \sinh z.$$

These are classical instances of the same cyclic basis.

5 Signed cyclic sections and the distinguished denominator

Definition 5.1. For $0 \leq j < N$, define

$$\mathsf{D}_{N,j}(z) = \sum_{n \geq 0} (-1)^n \frac{z^{Nn+j}}{(Nn+j)!}.$$

The distinguished signed denominator is

$$\mathsf{D}_N(z) = \mathsf{D}_{N,0}(z).$$

Fix any $\rho \in \mathbb{C}$ with $\rho^N = -1$.

Proposition 5.2 (Relation with the transfer exponential)**.** *For every j,*

$$\mathsf{D}_{N,j}(z) = \rho^{-j}\mathsf{E}_{N,j}(\rho z).$$

Proof. The coefficient of z^{Nn+j} on the right is

$$\rho^{-j}\rho^{Nn+j}\frac{1}{(Nn+j)!} = (-1)^n\frac{1}{(Nn+j)!}.$$

□

Theorem 5.3 (Signed derivative cycle)**.** *For $1 \le j < N$,*

$$\mathsf{D}'_{N,j} = \mathsf{D}_{N,j-1},$$

and

$$\mathsf{D}'_{N,0} = -\mathsf{D}_{N,N-1}.$$

Every signed section satisfies

$$\mathsf{D}^{(N)}_{N,j} = -\mathsf{D}_{N,j}.$$

The N signed sections form the basis of $y^{(N)} = -y$ determined by

$$\mathsf{D}^{(k)}_{N,j}(0) = \delta_{jk}$$

for $0 \le j, k < N$.

Proof. The derivative formulas follow directly from the series. The first nonzero term of $\mathsf{D}_{N,j}$ is $z^j/j!$, so the initial value matrix is the identity. The basis statement follows from the dimension of the solution space. □

Corollary 5.4 (Wronskian of the signed basis)**.** *Assume $N \ge 2$. The Wronskian of*

$$\mathsf{D}_{N,0}(z), \dots, \mathsf{D}_{N,N-1}(z)$$

is one.

Proof. The derivative matrix at the origin is the identity. The determinant factorization in the proof of corollary 4.3 applies with $a_\ell = \rho\omega^\ell$ and with the signed-section Fourier weights. Since $\sum_\ell a_\ell = 0$, the determinant is constant and equals one. □

Proposition 5.5 (Twisted addition law)**.** *For $0 \le k < N$,*

$$\mathsf{D}_{N,k}(z+w) = \sum_{\substack{0 \le i,j < N \\ i+j \equiv k \bmod N}} (-1)^{\lfloor (i+j)/N \rfloor}\mathsf{D}_{N,i}(z)\mathsf{D}_{N,j}(w).$$

Proof. Use proposition 5.2 and corollary 4.4. The factor introduced by a pair (i, j) is

$$\rho^{i+j-k}.$$

Since $0 \le i+j \le 2N-2$ and $i+j \equiv k \bmod N$, this factor is one when $i+j=k$ and -1 when $i+j=k+N$. □

Theorem 5.6 (Finite exponential representation)**.** *Let*

$$\mathcal{R}_N = \{\eta \in \mathbb{C} : \eta^N = -1\}.$$

Then

$$\mathsf{D}_N(z) = \frac{1}{N} \sum_{\eta \in \mathcal{R}_N} e^{\eta z}.$$

More generally,

$$\mathsf{D}_{N,j}(z) = \frac{\rho^{-j}}{N} \sum_{\ell=0}^{N-1} \omega^{-j\ell} e^{\rho\omega^\ell z}.$$

Proof. The second formula is the root filter expression for $\rho^{-j}\mathsf{E}_{N,j}(\rho z)$. The first is the case $j=0$. □

Definition 5.7. Define the reduced denominator

$$\Delta_N(w) = \sum_{n \ge 0} (-1)^n \frac{w^n}{(Nn)!}.$$

Then

$$\mathsf{D}_N(z) = \Delta_N(z^N).$$

Corollary 5.8 (Symmetries)**.** *The denominator satisfies*

$$\mathsf{D}_N(\omega z) = \mathsf{D}_N(z)$$

and

$$\mathsf{D}_N(\overline{z}) = \overline{\mathsf{D}_N(z)}.$$

Its nonzero zeros occur in orbits under multiplication by the Nth roots of unity and under complex conjugation.

Proof. The first identity follows from the powers z^{Nn}. The second follows from the real coefficients. Both symmetries preserve the zero set and multiplicity. □

Example 5.9. The first denominators are

$$\mathsf{D}_1(z) = e^{-z}, \qquad \mathsf{D}_2(z) = \cos z.$$

For $N=3$,

$$\mathsf{D}_3(z) = \frac{1}{3}\left(e^{-z} + 2e^{z/2}\cos\left(\frac{\sqrt{3}z}{2}\right)\right).$$

For $N=4$,

$$\mathsf{D}_4(z) = \cos\left(\frac{z}{\sqrt{2}}\right)\cosh\left(\frac{z}{\sqrt{2}}\right).$$

6 A finite algebra and matrix model

The central idea

The signed shift is a finite carrier for the differential cycle. At order four, three basis steps have coefficient one and the final step has coefficient minus one. Four steps return the vector with reversed sign. The matrix exponential packages every finite path length in one analytic expression.

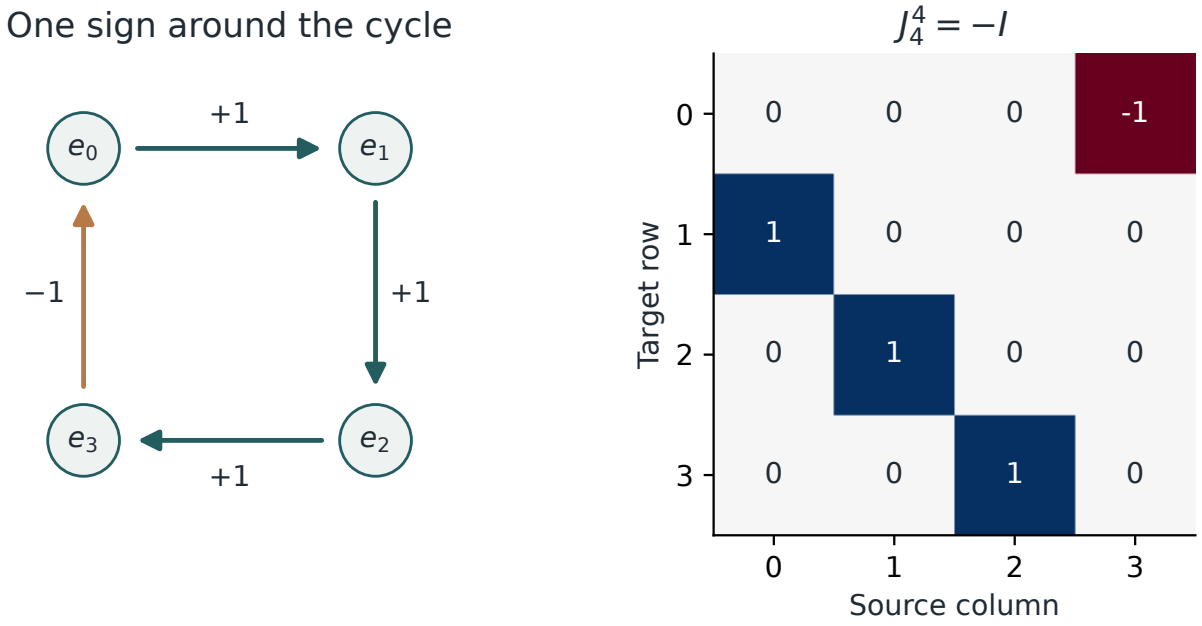


Figure 2. The exact signed shift J_4. Source columns and target rows describe the same action as the cycle on the left. Direct integer multiplication verifies $J_4^4 = -I$.

The derivative and addition laws can be read from one finite algebra identity. This formulation also identifies the exponent polygon as the spectrum of a signed cyclic shift.

Definition 6.1. Let

$$\mathfrak{A}_N^+ = \mathbb{C}[\xi]/(\xi^N - 1)$$

and

$$\mathfrak{A}_N^- = \mathbb{C}[\xi]/(\xi^N + 1).$$

Every element has a unique representative of degree below N.

Theorem 6.2 (Finite algebra exponential identities). *In $\mathfrak{A}_N^+$ one has*

$$e^{\lambda z \xi} = \sum_{j=0}^{N-1} \mathsf{E}_{N,j}(\lambda z)\xi^j.$$

In $\mathfrak{A}_N^-$ one has

$$e^{z\xi} = \sum_{j=0}^{N-1} \mathsf{D}_{N,j}(z)\xi^j.$$

Differentiation and multiplication of these identities recover the cyclic derivative systems and the two addition laws.

Proof. Expand the exponential and write every exponent uniquely as $Nn+j$. In $\mathfrak{A}_N^+$, the relation $\xi^{Nn+j} = \xi^j$ gives the transfer sections. In $\mathfrak{A}_N^-$, the relation $\xi^{Nn+j} = (-1)^n\xi^j$ gives the signed sections.

Differentiation multiplies by $\lambda\xi$ in the first algebra and by ξ in the second. Reduction of the product by the defining relation gives the derivative cycles. The identity

$$e^{(z+w)\xi} = e^{z\xi}e^{w\xi}$$

gives the addition laws after comparison of the coefficients of $1, \xi, \dots, \xi^{N-1}$. □

Definition 6.3. Let J_N be the signed cyclic shift on the standard basis $e_0, \dots, e_{N-1}$ of $\mathbb{C}^N$, defined by

$$J_N e_j = e_{j+1} \quad (0 \le j < N-1), \qquad J_N e_{N-1} = -e_0.$$

Then $J_N^N = -I$.

Theorem 6.4 (Companion matrix realization)**.** *Put*

$$\mathbf{D}_N(z) = \begin{pmatrix} \mathsf{D}_{N,0}(z) \\ \mathsf{D}_{N,1}(z) \\ \vdots \\ \mathsf{D}_{N,N-1}(z) \end{pmatrix}.$$

Then

$$\mathbf{D}_N(z) = e^{zJ_N}e_0$$

and

$$\mathsf{D}_N(z) = \frac{1}{N}\,\mathsf{tr}(e^{zJ_N}).$$

The eigenvalues of J_N are the N roots of -1, each with multiplicity one.

Proof. The vector $e^{zJ_N}e_0$ has derivative $J_N e^{zJ_N}e_0$ and initial value e_0. Its components therefore satisfy the same signed derivative cycle and initial data as $\mathbf{D}_N$. Uniqueness for the linear system gives the first identity.

The vectors

$$e_0, J_N e_0, \dots, J_N^{N-1}e_0$$

are the standard basis. Thus e_0 is cyclic for J_N, so the minimal polynomial of J_N has degree N. Since $J_N^N = -I$, that minimal polynomial divides $x^N + 1$ and therefore equals $x^N + 1$. The characteristic polynomial has the same degree and is divisible by the minimal polynomial, so it too equals $x^N + 1$.

This polynomial has distinct roots. Consequently J_N is diagonalizable and its spectrum consists of all N roots of -1, each once. Hence

$$\operatorname{tr}(e^{zJ_N}) = \sum_{\eta^N=-1} e^{\eta z} = N\mathsf{D}_N(z)$$

by theorem 5.6. □

Corollary 6.5 (Entries of the fundamental matrix)**.** *For $0 \le p, q < N$, let d_{pq} be the member of $\{0, \dots, N-1\}$ congruent to $p - q$ modulo N. Put*

$$\varepsilon_{pq} = \begin{cases} 1, & p \ge q, \\ -1, & p < q. \end{cases}$$

Then

$$\left(e^{zJ_N}\right)_{pq} = \varepsilon_{pq}\mathsf{D}_{N,d_{pq}}(z).$$

Thus every entry of the fundamental matrix is one of the signed sections, with the sign recording whether the cyclic path from q to p crosses the signed edge.

Proof. For every nonnegative integer n,

$$J_N^{Nn+d_{pq}} e_q = (-1)^n \varepsilon_{pq} e_p.$$

Only powers whose exponent is congruent to d_{pq} modulo N contribute to the (p, q) entry of the exponential. Therefore

$$\left(e^{zJ_N}\right)_{pq} = \varepsilon_{pq} \sum_{n \ge 0} (-1)^n \frac{z^{Nn+d_{pq}}}{(Nn + d_{pq})!}.$$

This is the stated formula. □

Corollary 6.6 (Addition and inversion laws for the companion matrix)**.** *Assume $N \ge 2$. The matrix*

$$U_N(z) = e^{zJ_N}$$

satisfies

$$U_N(z+w) = U_N(z)U_N(w), \qquad U_N(z)^{-1} = U_N(-z), \qquad \det U_N(z) = 1.$$

Its first column is $\mathbf{D}_N(z)$, and all of its entries are given by corollary 6.5.

Proof. The first two identities are the exponential group law. Since $\operatorname{tr} J_N = 0$,

$$\det(e^{zJ_N}) = e^{z \operatorname{tr} J_N} = 1.$$

The first column statement is theorem 6.4. □

Part II Reciprocal coefficients and polynomial families

7 Reciprocal powers and parameter polynomials

A route through this section

A unit constant term makes the formal reciprocal unambiguous. To compute one coefficient, only finitely many earlier coefficients are needed. The later analytic theory explains how those exact coefficients relate to the full zero set.

The factorial normalization carries the same coefficients in two forms

	n = 0	n = 1	n = 2	n = 3	n = 4
$P_{2,n}(1)$	1	1	5	61	1385
$[z^{2n}]\sec z$	1	1/2	5/24	61/720	1385/40320

Figure 3. At $N = 2$ and $r = 1$, the reciprocal is $\sec z$. The upper row gives block exponential coefficients and the lower row gives ordinary Taylor coefficients. Both rows encode the same five terms.

Since $\mathsf{D}_N(0) = 1$, its formal logarithm is defined at the origin. On a simply connected zero-free region containing the origin, there is a unique holomorphic branch normalized by $\log \mathsf{D}_N(0) = 0$, and its germ at the origin agrees with the formal logarithm.

Definition 7.1. For $r \in \mathbb{C}$, define

$$\mathsf{S}_N^{(r)}(z) = \mathsf{D}_N(z)^{-r} = \exp(-r \log \mathsf{D}_N(z)).$$

Write

$$\mathsf{S}_N^{(r)}(z) = \sum_{n \geq 0} P_{N,n}(r) \frac{z^{Nn}}{(Nn)!}.$$

Define the logarithmic coefficients $\kappa_{N,m}$ by

$$-\log \Delta_N(w) = \sum_{m \geq 1} \kappa_{N,m} \frac{w^m}{m}.$$

The polynomial dependence on r can be read directly from the formal logarithm. We use the complete exponential Bell polynomials B_n defined by

$$\exp\left(\sum_{m\geq 1} x_m \frac{t^m}{m!}\right) = \sum_{n\geq 0} B_n(x_1,\dots,x_n)\frac{t^n}{n!}.$$

Theorem 7.2 (Bell polynomial formula)**.** *For every $n \geq 0$,*

$$P_{N,n}(r) = \frac{(Nn)!}{n!} B_n(r\kappa_{N,1}, r\,1!\,\kappa_{N,2}, \dots, r\,(n-1)!\,\kappa_{N,n}).$$

For $n \geq 1$, this is a polynomial in r of exact degree n. Its leading coefficient is

$$\frac{(Nn)!}{n!(N!)^n}.$$

Proof. The identity

$$\Delta_N(w)^{-r} = \exp\left(r\sum_{m\geq 1}\kappa_{N,m}\frac{w^m}{m}\right)$$

fits the defining generating function for B_n with

$$x_m = r(m-1)!\kappa_{N,m}.$$

The coefficient of w^n is therefore the displayed Bell polynomial divided by $n!$. Multiplication by $(Nn)!$ gives $P_{N,n}(r)$. Since

$$\Delta_N(w) = 1 - \frac{w}{N!} + O(w^2),$$

we have $\kappa_{N,1} = 1/N!$. The term x_1^n gives the stated leading coefficient, which is nonzero. □

The coefficients also satisfy a triangular recurrence that does not require the logarithmic coefficients to be known first.

Theorem 7.3 (Parameter recurrence)**.** *We have $P_{N,0}(r) = 1$. For $n \geq 1$,*

$$\begin{aligned} P_{N,n}(r) = &-\sum_{k=1}^{n-1}(-1)^k\binom{Nn-1}{Nk}P_{N,n-k}(r) \\ &- r\sum_{k=1}^{n}(-1)^k\binom{Nn-1}{Nk-1}P_{N,n-k}(r). \end{aligned}$$

Proof. Set $S = \mathsf{D}_N^{-r}$. Logarithmic differentiation gives

$$\mathsf{D}_N S' + r\mathsf{D}_N' S = 0.$$

Taking the coefficient of $z^{Nn-1}/(Nn-1)!$ gives

$$\sum_{k=0}^{n-1}(-1)^k\binom{Nn-1}{Nk}P_{N,n-k}(r) + r\sum_{k=1}^{n}(-1)^k\binom{Nn-1}{Nk-1}P_{N,n-k}(r) = 0.$$

The term with $k = 0$ in the first sum is $P_{N,n}(r)$, and every other term has a smaller block index. Rearrangement gives the formula. □

Theorem 7.4 (Additive parameter law for reciprocal powers)**.** *For all $r, s \in \mathbb{C}$,*

$$\mathsf{S}_N^{(r)}(z)\mathsf{S}_N^{(s)}(z) = \mathsf{S}_N^{(r+s)}(z)$$

as a formal identity and on every common zero-free region with a fixed logarithm. Consequently,

$$P_{N,n}(r+s) = \sum_{q=0}^{n} \binom{Nn}{Nq} P_{N,q}(r) P_{N,n-q}(s).$$

Proof. Both statements follow by adding the exponents of the same logarithm and then multiplying the block exponential generating series. □

Corollary 7.5 (A sequence of binomial type)**.** *Define*

$$A_{N,n}(r) = \frac{n!}{(Nn)!} P_{N,n}(r).$$

Then

$$A_{N,n}(r+s) = \sum_{q=0}^{n} \binom{n}{q} A_{N,q}(r) A_{N,n-q}(s).$$

Thus $(A_{N,n})_{n\geq 0}$ is a polynomial sequence of binomial type.

Proof. Multiply the coefficient identity in theorem 7.4 by $n!/(Nn)!$ and simplify the factorials. □

The recurrence in theorem 7.3 uses the coefficients of the original denominator. A second recurrence uses only the logarithmic coefficients and is often shorter.

Theorem 7.6 (Logarithmic Newton recurrence)**.** *Set*

$$b_{N,n}(r) = \frac{P_{N,n}(r)}{(Nn)!}.$$

Then $b_{N,0}(r) = 1$ and for every $n \geq 1$,

$$n b_{N,n}(r) = r \sum_{k=1}^{n} \kappa_{N,k} b_{N,n-k}(r).$$

Equivalently,

$$P_{N,n}(r) = \frac{r}{n} \sum_{k=1}^{n} \frac{(Nn)!}{(N(n-k))!} \kappa_{N,k} P_{N,n-k}(r).$$

If $p_k = r\kappa_{N,k}$, then, for $n \geq 1$,

$$b_{N,n}(r) = \frac{1}{n!} \det \begin{pmatrix} p_1 & -1 & 0 & \cdots & 0 \\ p_2 & p_1 & -2 & \cdots & 0 \\ p_3 & p_2 & p_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -(n-1) \\ p_n & p_{n-1} & \cdots & p_2 & p_1 \end{pmatrix}.$$

Proof. Let

$$B_r(w) = \Delta_N(w)^{-r} = \sum_{n \geq 0} b_{N,n}(r) w^n.$$

Logarithmic differentiation gives

$$\frac{B_r'(w)}{B_r(w)} = r \sum_{k \geq 1} \kappa_{N,k} w^{k-1}.$$

Comparison of the coefficient of w^{n-1} proves the recurrence.

Let D_n denote the displayed determinant, with $D_0 = 1$. Expansion along the last row gives

$$D_n = \sum_{k=1}^{n} \frac{(n-1)!}{(n-k)!} p_k D_{n-k}.$$

Here the cofactor sign is canceled by the product of the negative superdiagonal entries $-(n-k+1), \dots, -(n-1)$. Thus $D_n/n!$ satisfies the same recurrence and initial value as $b_{N,n}(r)$. □

Corollary 7.7 (The first three reciprocal polynomials)**.** *For every $N \geq 1$,*

$$P_{N,1}(r) = r,$$

$$P_{N,2}(r) = r\left[\frac{1}{2}\binom{2N}{N}(r+1) - 1\right],$$

and

$$P_{N,3}(r) = r\left[\frac{(3N)!}{6(N!)^3}(r+1)(r+2) - \binom{3N}{N}(r+1) + 1\right].$$

Proof. The first logarithmic coefficients are

$$\kappa_{N,1} = \frac{1}{N!},$$

$$\kappa_{N,2} = \frac{1}{(N!)^2} - \frac{2}{(2N)!},$$

and

$$\kappa_{N,3} = \frac{1}{(N!)^3} - \frac{3}{N!(2N)!} + \frac{3}{(3N)!}.$$

Insert these values into theorem 7.2 and simplify. □

Corollary 7.8. *For $n \geq 1$, the polynomial $P_{N,n}(r)$ is divisible by r.*

Proof. At $r = 0$, the reciprocal family is the constant function one. Every positive-index block coefficient therefore vanishes. □

Example 7.9. For $N = 1$, one has $\mathsf{D}_1(z)^{-r} = e^{rz}$ and $P_{1,n}(r) = r^n$. For $N = 2$,

$$P_{2,1}(r) = r,$$

$$P_{2,2}(r) = r(3r + 2),$$

and

$$P_{2,3}(r) = r(15r^2 + 30r + 16).$$

At $r = 1$, these are the positive secant coefficients.

8 Logarithmic derivatives and Riccati equations

A coefficient calculation at order two

Write $\Delta_2(w)^{-r} = \sum_{n\geq 0} b_n(r)w^n$. Since

$$\Delta_2(w) = 1 - \frac{w}{2} + \frac{w^2}{24} + O(w^3), \qquad -\log \Delta_2(w) = \frac{w}{2} + \frac{w^2}{12} + O(w^3),$$

the first logarithmic coefficients are $\kappa_{2,1} = 1/2$ and $\kappa_{2,2} = 1/6$. Newton's recurrence gives

$$b_1(r) = \frac{r}{2}, \qquad 2b_2(r) = r\left(\frac{r}{4} + \frac{1}{6}\right).$$

Restoring the factorial normalization yields $P_{2,1}(r) = r$ and $P_{2,2}(r) = r(3r + 2)$. At $r = 1$, the second value is five. No zero-location assumption is needed for this finite calculation.

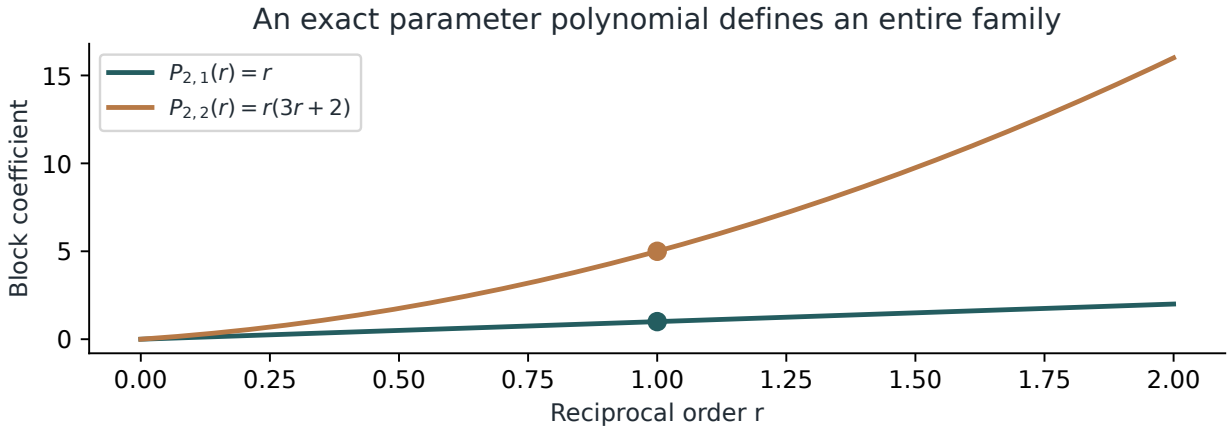


Figure 4. The two exact parameter polynomials from the worked calculation. Each plotted curve evaluates its displayed formula. The points at $r = 1$ recover the first two positive-index secant coefficients.

Definition 8.1. Define the transfer tangent

$$\mathsf{T}_N(z) = -\frac{\mathsf{D}_N'(z)}{\mathsf{D}_N(z)}.$$

Near the origin, its convergent block expansion has the form

$$\mathsf{T}_N(z) = \sum_{n\ge 0} T_N(n)\frac{z^{Nn+N-1}}{(Nn+N-1)!}.$$

Proposition 8.2 (Logarithmic coefficient formula)**.** *The transfer tangent satisfies*

$$\mathsf{T}_N(z) = N\sum_{m\ge 1}\kappa_{N,m}z^{Nm-1}.$$

Equivalently,

$$T_N(n) = N(Nn+N-1)!\kappa_{N,n+1}.$$

Proof. Since $\mathsf{D}_N(z) = \Delta_N(z^N)$,

$$-\frac{\mathsf{D}_N'(z)}{\mathsf{D}_N(z)} = -Nz^{N-1}\frac{\Delta_N'(z^N)}{\Delta_N(z^N)}.$$

Differentiate the defining series for $-\log\Delta_N$ and compare coefficients. □

Theorem 8.3 (Tangent recurrence)**.** *The coefficients satisfy $T_N(0) = 1$ and*

$$T_N(n) = (-1)^n - \sum_{k=1}^{n}(-1)^k\binom{Nn+N-1}{Nk}T_N(n-k).$$

Proof. The defining equation is

$$\mathsf{D}_N\mathsf{T}_N = -\mathsf{D}_N'.$$

Extracting the coefficient of $z^{Nn+N-1}/(Nn+N-1)!$ gives

$$\sum_{k=0}^{n}(-1)^k\binom{Nn+N-1}{Nk}T_N(n-k) = (-1)^n.$$

Isolating the $k=0$ term gives the recurrence. □

Corollary 8.4 (Derivative of the reciprocal family)**.** *For every $r\in\mathbb{C}$, the formal identity*

$$\frac{d}{dz}\mathsf{S}_N^{(r)}(z) = r\mathsf{T}_N(z)\mathsf{S}_N^{(r)}(z)$$

holds. The same identity holds analytically on every zero-free domain on which the logarithm used to define $\mathsf{S}_N^{(r)}$ is fixed.

Proof. Differentiate D_N^{-r} on a zero-free region. The formal identity follows from the same logarithmic calculation. □

For a function g, define the complete differential Bell polynomial Y_N by

$$\frac{d^N}{dz^N}e^{g(z)} = e^{g(z)}Y_N(g', g'', \dots, g^{(N)}).$$

Theorem 8.5 (Scalar higher Riccati identity). *At every point of the open set* $\{z : \mathsf{D}_N(z) \neq 0\}$,

$$Y_N(-\mathsf{T}_N, -\mathsf{T}_N', \dots, -\mathsf{T}_N^{(N-1)}) = -1.$$

Proof. Write $\mathsf{D}_N = e^g$ locally. Then $g' = -\mathsf{T}_N$. Divide the equation $\mathsf{D}_N^{(N)} = -\mathsf{D}_N$ by D_N and use the defining identity for Y_N. □

This scalar identity follows from one ordinary differential equation. It does not redevelop the general nonlinear operation-tree theory of Chapter III.

Example 8.6. For $N = 1$, $\mathsf{T}_1 = 1$. For $N = 2$, $\mathsf{D}_2 = \cos z$ and $\mathsf{T}_2 = \tan z$. The Riccati equation becomes

$$\mathsf{T}_2' = 1 + \mathsf{T}_2^2.$$

9 Residue-section reciprocals and Appell systems

The section $\mathsf{E}_{N,j}$ has a forced zero of order j at the origin. Removing it leaves a unit series in z^N.

Definition 9.1. For $0 \leq j < N$, define

$$\Phi_{N,j}(w) = j! \sum_{n \geq 0} \frac{w^n}{(Nn + j)!}.$$

Then

$$\mathsf{E}_{N,j}(z) = \frac{z^j}{j!}\Phi_{N,j}(z^N).$$

Define the residue-section reciprocal function

$$\mathsf{C}_{N,j}(z) = \Phi_{N,j}(z^N)^{-1} = \frac{z^j/j!}{\mathsf{E}_{N,j}(z)}.$$

For $j > 0$, the last quotient at $z = 0$ is interpreted by its removable continuation, namely the first expression in the display. Write

$$\mathsf{C}_{N,j}(z) = \sum_{n \geq 0} C_{N,j}(n)\frac{z^{Nn}}{(Nn)!}.$$

Theorem 9.2 (Triangular inverse system). *The coefficients are determined by $C_{N,j}(0) = 1$ and*

$$\sum_{q=0}^{n} \binom{Nn+j}{Nq} C_{N,j}(q) = \delta_{n0}.$$

For $n \geq 1$,

$$C_{N,j}(n) = -\frac{\displaystyle\sum_{q=0}^{n-1} \binom{Nn+j}{Nq} C_{N,j}(q)}{\binom{Nn+j}{Nn}}.$$

Proof. Multiply the defining series for $\mathsf{C}_{N,j}$ and $\mathsf{E}_{N,j}$. Their product is $z^j/j!$. At degree $Nn+j$, multiplication by $(Nn+j)!$ gives the stated binomial coefficient. The diagonal coefficient is nonzero, so the system is triangular. □

Definition 9.3. For $r \in \mathbb{C}$, define formal coefficients $C_{N,j,n}(r)$ by

$$\mathsf{C}_{N,j}(z)^r = \sum_{n \geq 0} C_{N,j,n}(r) \frac{z^{Nn}}{(Nn)!}.$$

On a simply connected region containing 0 and avoiding the zeros of $\Phi_{N,j}(z^N)$, interpret this power as e^{rL}, where L is the holomorphic logarithm of $\mathsf{C}_{N,j}$ normalized by $L(0) = 0$.

Theorem 9.4 (Parameter-addition law for residue-section reciprocals). *For each $n \geq 1$, the coefficient $C_{N,j,n}(r)$ is a polynomial in r of exact degree n. Its leading coefficient is*

$$\frac{(Nn)!}{n!} \left(-\frac{j!}{(N+j)!} \right)^n.$$

It satisfies

$$C_{N,j,n}(r+s) = \sum_{q=0}^{n} \binom{Nn}{Nq} C_{N,j,q}(r) C_{N,j,n-q}(s).$$

In particular,

$$C_{N,j,n}(1) = C_{N,j}(n).$$

Proof. The unit series $\Phi_{N,j}(w)^{-1}$ has a formal logarithm with zero constant term. Its linear term is

$$-\frac{j!}{(N+j)!} w.$$

Exponentiation after multiplication by r shows that the coefficient of w^n is a polynomial of degree at most n. The nth power of the linear logarithmic term gives the displayed nonzero leading coefficient,

so the degree is exactly n. Multiplication of the powers with parameters r and s gives the coefficient identity. More explicitly, before the factorial normalization the coefficient of $r^n w^n$ is

$$\frac{1}{n!}\left(-\frac{j!}{(N+j)!}\right)^n;$$

thus multiplication by $(Nn)!$ gives the displayed leading coefficient. □

Definition 9.5. Define polynomials $E_m^{(N,j,r)}(x)$ by

$$\mathsf{C}_{N,j}(z)^r e^{xz} = \sum_{m\geq 0} E_m^{(N,j,r)}(x)\frac{z^m}{m!}.$$

Theorem 9.6 (Appell laws)**.** *For $m \geq 1$, the polynomials satisfy*

$$\frac{d}{dx}E_m^{(N,j,r)}(x) = mE_{m-1}^{(N,j,r)}(x)$$

and, for $m \geq 0$,

$$E_m^{(N,j,r)}(x+y) = \sum_{q=0}^{m}\binom{m}{q}E_q^{(N,j,r)}(x)y^{m-q}.$$

Proof. Differentiate the generating function with respect to x for the first identity. Multiply it by e^{yz} and compare coefficients for the second. □

Proposition 9.7 (Explicit Appell polynomial)**.** *For every $m \geq 0$,*

$$E_m^{(N,j,r)}(x) = \sum_{0\leq n\leq m/N}\binom{m}{Nn}C_{N,j,n}(r)x^{m-Nn}.$$

Proof. Multiply the block series for $\mathsf{C}_{N,j}^r$ by the exponential series and collect the coefficient of total degree m. □

Theorem 9.8 (Joint Appell convolution in the space and order parameters)**.** *For all $r, s, x, y \in \mathbb{C}$ and $m \geq 0$,*

$$E_m^{(N,j,r+s)}(x+y) = \sum_{q=0}^{m}\binom{m}{q}E_q^{(N,j,r)}(x)E_{m-q}^{(N,j,s)}(y).$$

Proof. Multiply the generating functions

$$\mathsf{C}_{N,j}(z)^r e^{xz} \quad \text{and} \quad \mathsf{C}_{N,j}(z)^s e^{yz}.$$

Their product is the generating function with parameters $r+s$ and $x+y$. Coefficient comparison gives the formula. □

Proposition 9.9 (Formal differential-operator representation on polynomials)**.** *Let a formal series $A(z) = \sum_{q\ge 0} a_q z^q/q!$ act on polynomials by*

$$A(\partial_x) f(x) = \sum_{q\ge 0} \frac{a_q}{q!} \partial_x^q f(x).$$

The power r is applied to the formal symbol before the substitution $z = \partial_x$. Then

$$E_m^{(N,j,r)}(x) = \mathsf{C}_{N,j}(\partial_x)^r x^m.$$

The inverse relation is

$$x^m = \Phi_{N,j}(\partial_x^N)^r E_m^{(N,j,r)}(x).$$

Both sums are finite on every polynomial.

Proof. Applying $A(\partial_x)$ to e^{xz} multiplies it by $A(z)$. Taking $A = \mathsf{C}_{N,j}^r$ gives the first identity by the defining generating function. Constant coefficient operators on polynomials satisfy

$$(AB)(\partial_x) = A(\partial_x) B(\partial_x),$$

because all derivative sums are finite and powers of ∂_x commute. The two formal symbols

$$\mathsf{C}_{N,j}(z)^r = \Phi_{N,j}(z^N)^{-r} \quad \text{and} \quad \Phi_{N,j}(z^N)^r$$

multiply to one. Their substitutions at $z = \partial_x$ are therefore inverse constant coefficient operators on polynomials, which proves the second identity. □

Corollary 9.10 (Appell symmetry)**.** *Each polynomial $E_m^{(N,j,r)}$ is monic of degree m, and*

$$E_m^{(N,j,r)}(\omega x) = \omega^m E_m^{(N,j,r)}(x).$$

Proof. The term $n = 0$ in proposition 9.7 is x^m. Every other exponent is $m - Nn$, so replacement of x by ωx multiplies every term by ω^m. □

Example 9.11. For $(N, j) = (2, 0)$,

$$\mathsf{C}_{2,0}(z) = \operatorname{sech} z.$$

For $(N, j) = (2, 1)$,

$$\mathsf{C}_{2,1}(z) = \frac{z}{\sinh z}.$$

The first case is a scaled and translated Euler family. In fact, if $E_m(x)$ denotes the classical Euler polynomial, then

$$E_m^{(2,0,1)}(x) = 2^m E_m\left(\frac{x+1}{2}\right).$$

If $B_m(x)$ denotes the classical Bernoulli polynomial, the second case has the exact identification

$$E_m^{(2,1,1)}(x) = 2^m B_m\left(\frac{x+1}{2}\right),$$

obtained by substituting $t = 2z$ in $te^{yt}/(e^t - 1)$.

10 Scaled reciprocal coupling and theta polynomials

Definition 10.1. Set

$$H_N(z) = \mathsf{E}_{N,0}(z) = \sum_{n\geq 0} \frac{z^{Nn}}{(Nn)!}.$$

For $u \in \mathbb{C}$, define $\Theta_{N,n}(u)$ by

$$H_N(uz)^{-1} H_N(z)^{-u} = \sum_{n\geq 0} \Theta_{N,n}(u) \frac{z^{Nn}}{(Nn)!}.$$

This is first a formal identity at the origin. Analytically, it is interpreted on a common simply connected zero-free region with the branch of $\log H_N$ normalized by $\log H_N(0) = 0$.

Theorem 10.2 (Theta reconstruction)**.** *For every $n \geq 0$,*

$$\Theta_{N,n}(u) = (-1)^n \sum_{q=0}^{n} \binom{Nn}{Nq} P_{N,q}(1) u^{Nq} P_{N,n-q}(u).$$

Thus $\Theta_{N,n}$ is a polynomial of degree at most Nn.

Proof. Choose ρ with $\rho^N = -1$. Then

$$H_N(z) = \mathsf{D}_N(\rho z).$$

The coefficient of z^{Nn} in $H_N(z)^{-r}$ is therefore $(-1)^n P_{N,n}(r)/(Nn)!$. Apply this to $H_N(uz)^{-1}$ and $H_N(z)^{-u}$, then multiply the two block series. □

Proposition 10.3 (Trivial parameter factors)**.** *For $n \geq 1$, the polynomial $\Theta_{N,n}(u)$ is divisible by u. If N is even, it is also divisible by $u + 1$.*

Proof. At $u = 0$, the defining product is one. Hence every positive-index coefficient vanishes. If N is even, then $H_N(-z) = H_N(z)$. At $u = -1$, the product is again one, so every positive-index coefficient vanishes there as well. □

Proposition 10.4 (Theta Newton recurrence)**.** *Put*

$$\theta_{N,n}(u) = \frac{\Theta_{N,n}(u)}{(Nn)!}.$$

Then $\theta_{N,0}(u) = 1$ and

$$n\theta_{N,n}(u) = \sum_{m=1}^{n} (-1)^m \kappa_{N,m}(u^{Nm} + u)\theta_{N,n-m}(u).$$

In particular,

$$\Theta_{N,1}(u) = -(u^N + u).$$

Proof. Write $w = z^N$. Since $H_N(z) = \Delta_N(-w)$,

$$-\log H_N(z) = \sum_{m \geq 1} (-1)^m \kappa_{N,m} \frac{w^m}{m}.$$

The logarithm of the theta generating function is therefore

$$\sum_{m \geq 1} (-1)^m \kappa_{N,m} (u^{Nm} + u) \frac{w^m}{m}.$$

Logarithmic differentiation and coefficient comparison give the recurrence. The case $n = 1$ uses $\kappa_{N,1} = 1/N!$. □

Example 10.5. For $N = 2$,

$$\Theta_{2,1}(u) = -u(u+1)$$

and

$$\Theta_{2,2}(u) = u(u+1)(5u^2 + u + 2).$$

The two factors predicted by proposition 10.3 are visible.

Part III The geometry of zeros

11 Growth and the geometry of the large zeros

A route through this section

The finite exponential representation gives a polygon of exponents. Large zeros can occur where two dominant terms nearly cancel. The proof first localizes those zeros and then uses symmetry and uniqueness to place them on the rays. Finite exceptional zeros remain part of the later products and sums.

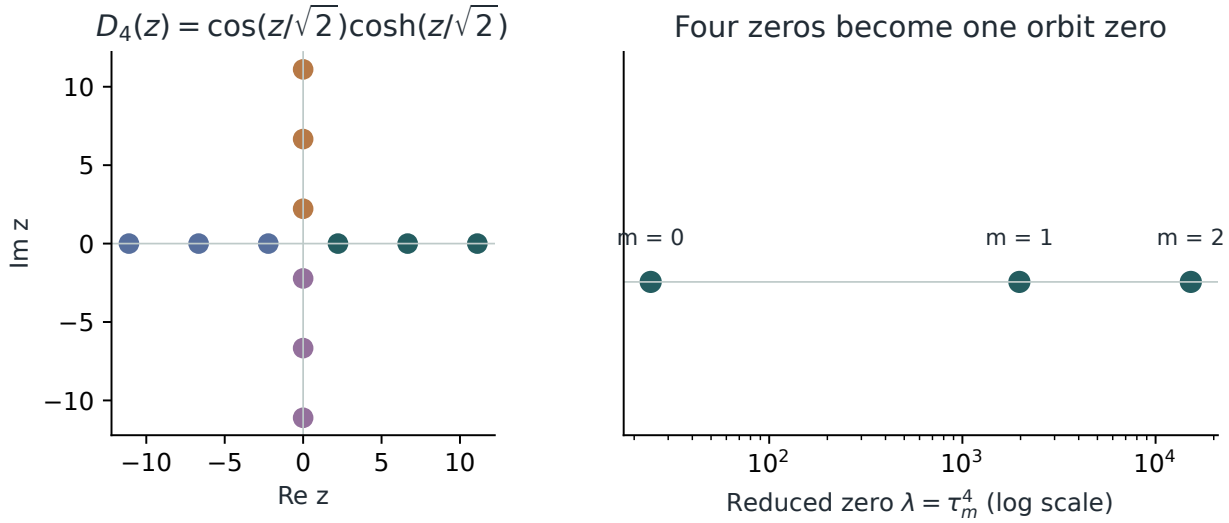


Figure 5. The exact order-four model. The first three zeros on each of four rays are mapped by $w = z^4$ to three positive reduced zeros. The reduced axis uses a logarithmic scale. This complete order-four example does not classify the finite exceptional set at other orders.

We now assume $N \geq 2$. Write

$$\eta_k = e^{(2k+1)\pi i/N}, \qquad 0 \leq k < N.$$

These are the vertices of a regular polygon and

$$\mathsf{D}_N(z) = \frac{1}{N} \sum_{k=0}^{N-1} e^{\eta_k z}.$$

Standard Fact 11.1 (Order from Taylor coefficients)**.** If $f(w) = \sum_{n\ge 0} a_n w^n$ is an entire function with infinitely many nonzero coefficients, then

$$\operatorname{ord} f = \limsup_{\substack{n\to\infty \\ a_n\neq 0}} \frac{n \log n}{\log(1/|a_n|)}.$$

Canonical products for entire functions of order below one have genus zero. We use these standard facts in the form given in [Boa54, Tit39].

Proposition 11.2 (Orders and exponential type)**.** *For $N \ge 2$,*

$$\operatorname{ord} \Delta_N = \frac{1}{N}.$$

The function D_N has order one and exponential type one.

Proof. The coefficient of w^n in Δ_N has modulus $1/(Nn)!$. Stirling's formula and standard fact 11.1 give order $1/N$. Since $M_{f(\cdot^N)}(r) = M_f(r^N)$ for the maximum modulus, composition with z^N multiplies the order by N and gives order one for D_N.

The finite exponential representation gives

$$|\mathsf{D}_N(z)| \le e^{|z|},$$

so the type is at most one. Choose a root η_0 and put $z = t\overline{\eta_0}$ with $t > 0$. The term $e^{\eta_0 z} = e^t$ has strictly larger real exponent than every other term. Hence

$$\mathsf{D}_N(t\overline{\eta_0}) = \frac{1}{N} e^t (1 + o(1)).$$

The type is therefore one. □

Definition 11.3. For $\theta \in \mathbb{R}$, define

$$h_N(\theta) = \max_{0\le k<N} \cos\left(\theta + \frac{(2k+1)\pi}{N}\right).$$

The tie directions are

$$\mathfrak{T}_N = \left\{ \frac{2\pi k}{N} : 0 \le k < N \right\} \mod 2\pi.$$

Theorem 11.4 (Indicator polygon and dominant sectors)**.** *The indicator of D_N equals h_N, with the limsup taken over radii where $\mathsf{D}_N(re^{i\theta}) \neq 0$:*

$$\limsup_{r\to\infty} \frac{1}{r} \log|\mathsf{D}_N(re^{i\theta})| = h_N(\theta).$$

If K is a compact set of directions disjoint from $\mathfrak{T}_N$, then there are $\delta > 0$ and $R > 0$ such that, for each $\theta \in K$, one root $\eta(\theta)$ of -1 satisfies

$$\mathsf{D}_N(re^{i\theta}) = \frac{1}{N}\exp(\eta(\theta)re^{i\theta})\left(1 + O(e^{-\delta r})\right)$$

uniformly for $r \geq R$. In particular, no sufficiently large zero can lie in those directions.

Proof. For $z = re^{i\theta}$, the real part of the exponent $\eta_k z$ is

$$r\cos\left(\theta + \frac{(2k+1)\pi}{N}\right).$$

The triangle inequality gives the upper indicator bound $h_N(\theta)$.

Away from a tie direction, the maximizing vertex is unique. It is constant on each connected component of the complement of the tie directions. A compact set avoiding all ties meets only finitely many such components, and compactness gives one uniform positive lower bound for every relevant projection gap. Division by the maximizing exponential gives the stated uniform asymptotic and the zero-free conclusion.

At a tie direction exactly two adjacent vertices attain the maximum. After rotation it is enough to take $\theta = 0$. With $c_N = \cos(\pi/N)$ and $s_N = \sin(\pi/N)$, the two dominant terms give

$$\frac{2}{N}e^{c_N r}\cos(s_N r)$$

and every other term is exponentially smaller. Along a sequence on which $|\cos(s_N r)| \geq 1/2$, the lower indicator bound is $c_N = h_N(0)$. Rotational symmetry treats every tie direction. □

The zero strings are governed by two adjacent exponents. Put

$$a_N = \frac{\pi}{N}, \qquad s_N = \sin a_N, \qquad c_N = \cos a_N,$$

and

$$A_N = \frac{\pi}{s_N}, \qquad t_m = A_N\left(m + \frac{1}{2}\right).$$

For $N \geq 3$, also put

$$\gamma_N = c_N - \cos(3a_N) > 0.$$

Lemma 11.5 (Two term dominance)**.** *Assume $N \geq 3$. There are positive constants Y, R, and C with the following properties.*

If $z = x + iy$ lies in the sector $|\arg z| \leq a_N$, has $x \geq R$, and is a zero of D_N, then $|y| \leq Y$.

On the strip $x \geq R$ and $|y| \leq Y$, define

$$R_N(z) = \sum_{\substack{\eta^N=-1 \\ \eta \neq e^{ia_N}, e^{-ia_N}}} \exp((\eta - e^{-ia_N})z).$$

Then

$$Ne^{-e^{-ia_N}z}\mathsf{D}_N(z) = 1 + e^{2is_N z} + R_N(z),$$

and, for $q = 0, 1, 2$,

$$|R_N^{(q)}(z)| \le Ce^{-\gamma_N x}.$$

Proof. A direct projection estimate gives the stronger bound

$$Y = \frac{\log(N-1)}{2s_N}$$

for every zero in the sector, without a large-radius assumption. Write $z = re^{i\theta}$ with $0 \le \theta \le a_N$ and $r \ge 0$. For $0 \le \ell \le N-2$, the angle $\theta + (2\ell+1)a_N$ lies between $a_N + \theta$ and $2\pi - a_N - \theta$. Monotonicity of cosine on $[0, \pi]$, followed by reflection about π, therefore gives

$$\Re(\eta_\ell z) \le r\cos(a_N + \theta).$$

The remaining vertex e^{-ia_N} has projection $r\cos(a_N - \theta)$. Their difference is at least $2s_N r\sin\theta = 2s_N y$. If $\mathsf{D}_N(z) = 0$, divide its finite exponential sum by the term with exponent $e^{-ia_N}z$ and apply the triangle inequality:

$$1 \le (N-1)e^{-2s_N y}.$$

Thus $y \le Y$. Conjugation gives the same bound for $-y$ on the lower half sector. Since $N \ge 3$ and $s_N > 0$, this Y is positive.

On the resulting strip, the roots e^{ia_N} and e^{-ia_N} have real part c_N, while each remaining root $\eta = u+iv$ satisfies $u \le \cos(3a_N) = c_N - \gamma_N$. Hence

$$\Re\big((\eta - e^{-ia_N})z\big) \le -\gamma_N x + 2Y.$$

Division by $e^{e^{-ia_N}z}$ gives the displayed normalization. Each differentiation multiplies a remainder term by $\eta - e^{-ia_N}$, whose modulus is at most two. Thus, for $q = 0, 1, 2$,

$$|R_N^{(q)}(z)| \le (N-2)2^q e^{2Y}e^{-\gamma_N x}.$$

One may take $R = 1$ and $C = 4(N-2)e^{2Y}$. □

Lemma 11.6 (Near cancellation of two exponentials)**.** *For every* $M > 0$*, there are constants* $\varepsilon_M > 0$ *and* $C_M > 0$ *such that the following holds. If*

$$|\Re w| \le M$$

and

$$|1 + e^w| \le \varepsilon_M,$$

then there is a unique integer m *with*

$$|w - (2m+1)\pi i| \le C_M|1 + e^w|.$$

Proof. In fact, the constants $\varepsilon_M = 1/2$ and $C_M = 3/2$ work for every complex w, without the strip restriction. Put $u = -(1 + e^w)$ and $L = \log(1+u) = \log(-e^w)$, using the principal logarithm. The elementary logarithm estimate on $|u| \le 1/2$ gives $|L| \le (3/2)|u|$. Since $e^{L+\pi i} = e^w$, there is an integer m such that

$$w = L + (2m+1)\pi i.$$

This gives the required distance bound. Distinct centers are separated by at least 2π, whereas two centers satisfying the bound would be separated by at most $3|1+e^{w}| \le 3/2$. Thus the integer is unique. The formal proof uses the convergent logarithm estimate and the exact periodicity theorem for the complex exponential. □

Theorem 11.7 (Large zero strings). *Assume $N \ge 3$. There are constants m_0, C, and $\gamma > 0$ and a sequence of positive real simple zeros $(\tau_m)_{m\ge m_0}$ of D_N such that*

$$\tau_m = t_m + O(e^{-\gamma t_m}).$$

Every zero of sufficiently large modulus is of the form

$$\omega^k \tau_m$$

for a unique $0 \le k < N$ and a unique $m \ge m_0$.
The derivatives satisfy

$$\mathsf{D}'_N(\omega^k \tau_m) = \omega^{-k} \mathsf{D}'_N(\tau_m)$$

and

$$|\mathsf{D}'_N(\tau_m)| = \frac{2s_N}{N} e^{c_N \tau_m} \left(1 + O(e^{-\gamma \tau_m})\right).$$

Proof. Work first in the sector $|\arg z| \le a_N$. By lemma 11.5, every sufficiently large zero lies in a fixed strip about the positive real axis. Put

$$G(z) = N e^{-e^{-i a_N} z} \mathsf{D}_N(z).$$

Then

$$G(z) = 1 + e^{2i s_N z} + R_N(z)$$

with an exponentially small remainder and derivative.

If $G(z) = 0$, then

$$|1 + e^{2i s_N z}| \le C e^{-\gamma_N \Re z}.$$

The real part of $2i s_N z$ is bounded in the strip. Apply lemma 11.6 with $w = 2i s_N z$. There is a unique integer m such that

$$|z - t_m| \le C_1 e^{-\gamma_N \Re z},$$

where

$$t_m = \frac{(2m+1)\pi}{2s_N}.$$

In particular, $|\Re z - t_m| = o(1)$. For all sufficiently large zeros we have $|\Re z - t_m| \le 1$, and hence

$$|z - t_m| \le C_2 e^{-\gamma_N t_m}$$

with a constant independent of m. Because $\Re z \to +\infty$ in this argument, the corresponding integers m are eventually nonnegative and tend to infinity.

Fix a large m, put $a = 2is_N$, and consider the closed disk $B_m = \{z : |z - t_m| \le 1/(2s_N)\}$. The remainder bounds hold uniformly on these fixed-radius disks. For sufficiently large m, $|R_N| \le 1/2$ and $|R_N'| \le s_N/2$ there. Define

$$\Phi_m(z) = t_m + \frac{\log(1 + R_N(z))}{2is_N},$$

using the principal logarithm. The estimate $|\log(1 + u)| \le (3/2)|u|$ shows that Φ_m maps B_m into itself. The derivative bound $|\Phi_m'| \le 1/2$ makes it a contraction. Banach's fixed-point theorem gives a unique fixed point. On B_m, $|a(z - t_m)| \le 1 < \pi$, so the principal logarithm inverts the exponential and the fixed-point equation is equivalent to $G(z) = 0$. Moreover,

$$|z - t_m| \le \frac{3}{4s_N}|R_N(z)| = O(e^{-\gamma_N t_m}).$$

At this zero,

$$|G'(z)| \ge |a|\,|1 + R_N(z)| - |R_N'(z)| \ge s_N/2 > 0,$$

so the zero is simple. The disks are mutually disjoint because their diameter $1/s_N$ is smaller than the center spacing π/s_N. Enlarging the strip if necessary, these disks lie inside the fundamental sector and the fixed strip for large m.

The localization estimate puts every sufficiently large zero in the fundamental sector into one of these disks. The zero sets of G and D_N coincide because their prefactor is nonzero. Each disk is invariant under complex conjugation, and D_N has real coefficients. The conjugate of its unique zero is therefore the same zero. It is real, positive for large m, and simple by the derivative bound above. Denote it by τ_m.

The parts of the fundamental sector away from the positive tie ray, including its boundary rays, were already excluded uniformly. Rotational invariance now gives every remaining large zero as $\omega^k\tau_m$. Distinct rays and the disjoint disks give uniqueness of k and m. Taking $\gamma = \gamma_N$ proves the zero location in the statement.

Differentiate the normalized expression. At $z = \tau_m$,

$$G'(\tau_m) = 2is_N e^{2is_N\tau_m} + R_N'(\tau_m) = -2is_N + O(e^{-\gamma_N\tau_m}).$$

Since $G(\tau_m) = 0$,

$$\mathsf{D}_N'(\tau_m) = \frac{1}{N}e^{e^{-ia_N\tau_m}}G'(\tau_m).$$

Taking absolute values gives the derivative asymptotic. Differentiating $\mathsf{D}_N(\omega z) = \mathsf{D}_N(z)$ gives the rotation formula. □

Theorem 11.8 (Leading displacement term and derivative sign)**.** *With the notation of lemma 11.5 and theorem 11.7,*

$$\tau_m = t_m + \frac{R_N(t_m)}{2is_N} + O(e^{-2\gamma_N t_m}).$$

The quotient $R_N(t_m)/(2is_N)$ is real. One also has

$$\mathsf{D}_N'(\tau_m) = (-1)^{m+1}\frac{2s_N}{N}e^{c_N\tau_m}\left(1 + O(e^{-\gamma_N\tau_m})\right).$$

Proof. Put $\delta_m = \tau_m - t_m$. The large zero theorem gives $\delta_m = O(e^{-\gamma_N t_m})$. Since $e^{2is_N t_m} = -1$,

$$1 + e^{2is_N(t_m+\delta_m)} = -2is_N\delta_m + O(\delta_m^2).$$

The remainder estimate through the first derivative gives

$$R_N(t_m + \delta_m) = R_N(t_m) + O(e^{-\gamma_N t_m}|\delta_m|).$$

Substitution in the zero equation gives the displacement formula.

At the real point t_m, the normalized value

$$Ne^{-e^{-ia_N}t_m}\mathsf{D}_N(t_m)$$

is purely imaginary. The two dominant terms cancel there, so $R_N(t_m)$ is purely imaginary and the displayed correction is real.

For the derivative, use the formula in the proof of theorem 11.7 and

$$e^{-is_N\tau_m} = (-1)^m(-i)\left(1 + O(e^{-\gamma_N\tau_m})\right).$$

The stated real sign follows. □

Corollary 11.9 (Leading correction for $N = 3$)**.** *For $N = 3$,*

$$t_m = \frac{2\pi}{\sqrt{3}}\left(m + \frac{1}{2}\right)$$

and

$$\tau_m = t_m + \frac{(-1)^m}{\sqrt{3}}e^{-3t_m/2} + O(e^{-3t_m}).$$

Proof. The only nondominant exponent is -1. Hence

$$R_3(t) = \exp\left(\left(-\frac{3}{2} + \frac{\sqrt{3}}{2}i\right)t\right).$$

At $t = t_m$, its phase is $i(-1)^m$. Insert this value into theorem 11.8. □

Corollary 11.10 (The case $N = 2$)**.** *For $N = 2$,*

$$\mathsf{D}_2(z) = \cos z.$$

Its zeros are

$$\pm\left(m + \frac{1}{2}\right)\pi, \qquad m \geq 0,$$

and all are simple. They agree with the formula $t_m = A_2(m + 1/2)$.

Proof. This is the standard zero set of the cosine function. □

Corollary 11.11 (Eventual simplicity and exact rays)**.** *For every $N \geq 2$, all but finitely many zeros of D_N are simple. Every sufficiently large zero lies on one of the rays*

$$\omega^k(0,\infty), \qquad 0 \leq k < N.$$

Proof. Use theorem 11.7 for $N \geq 3$ and corollary 11.10 for $N = 2$. □

Example 11.12 (Exact fourth-order strings)**.** For $N = 4$,

$$\mathsf{D}_4(z) = \cos\left(\frac{z}{\sqrt{2}}\right)\cosh\left(\frac{z}{\sqrt{2}}\right).$$

The positive zeros are exactly

$$\tau_m = \sqrt{2}\pi\left(m + \frac{1}{2}\right) = A_4\left(m + \frac{1}{2}\right).$$

Rotation by the fourth roots of unity gives the real and imaginary zero strings. In this case the error term in theorem 11.7 is identically zero.

Corollary 11.13 (Reduced zeros)**.** *Let Λ_N be the nonzero zero multiset of Δ_N. All but finitely many members of Λ_N are positive and simple. They can be indexed as*

$$\lambda_m = \tau_m^N$$

for all sufficiently large m. If $N \geq 3$, the notation may be chosen as in theorem 11.7, and

$$\lambda_m = t_m^N + O\left(t_m^{N-1} e^{-\gamma t_m}\right).$$

For $N = 2$, take $\tau_m = t_m = \pi(m + 1/2)$ for $m \geq 0$. Then $\lambda_m = t_m^2$ exactly.

Proof. The identity $\mathsf{D}_N(z) = \Delta_N(z^N)$ identifies each orbit of N nonzero zeros of D_N with one zero of Δ_N. For $N \geq 3$, the large orbit representatives are the positive zeros τ_m from theorem 11.7. For $N = 2$, use corollary 11.10. Simplicity is preserved because

$$\mathsf{D}_N'(z) = N z^{N-1} \Delta_N'(z^N)$$

and $z \neq 0$. The asymptotic follows by expanding τ_m^N about t_m. □

Theorem 11.14 (Zero counting asymptotics)**.** *Let $n_{\mathsf{D}_N}(R)$ count the zeros of D_N in $|z| \leq R$, with multiplicity. Let $n_{\Delta_N}(R)$ count the zeros of Δ_N in $|w| \leq R$. Then, as $R \to \infty$,*

$$n_{\mathsf{D}_N}(R) = \frac{N \sin(\pi/N)}{\pi} R + O(1)$$

and

$$n_{\Delta_N}(R) = \frac{\sin(\pi/N)}{\pi} R^{1/N} + O(1).$$

Proof. Each of the N rays contains one large zero near

$$t_m = \frac{\pi}{\sin(\pi/N)} \left(m + \frac{1}{2} \right).$$

Since $\tau_m - t_m = o(1)$ and the model spacing is the positive constant A_N, replacing $\tau_m \le R$ by $t_m \le R$ changes the count by at most a bounded amount. The number of such indices with $t_m \le R$ is

$$\frac{\sin(\pi/N)}{\pi} R + O(1).$$

Multiplication by N and addition of the finite exceptional set give the first formula. The second follows from

$$n_{\mathsf{D}_N}(R^{1/N}) = N n_{\Delta_N}(R).$$

□

Corollary 11.15 (Zero exponents)**.** *The exponent of convergence of the nonzero zeros of* D_N *is one. The exponent of convergence of the zeros of* Δ_N *is* $1/N$*.*

Proof. The asymptotic counting formulas in theorem 11.14 give the two exponents by partial summation. □

Part IV Products, moments, and zero zeta functions

12 Hadamard products and zero power sums

The central idea

The product uses every reduced zero with its multiplicity. Its logarithm turns multiplication into addition, so reciprocal-zero power sums become coefficients. This step depends on the global product theorem. A truncated zero list provides a numerical partial sum, while the identity uses the entire multiset.

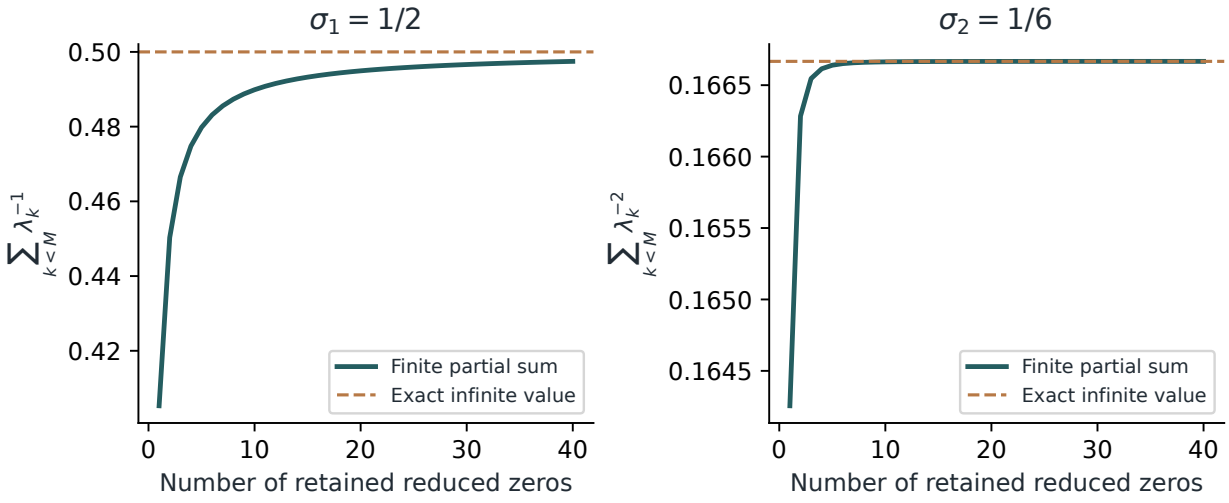


Figure 6. Order-two moment sums over the first forty reduced zeros $\lambda_k = \pi^2(k + 1/2)^2$. The curves are finite numerical partial sums. The dashed limits $1/2$ and $1/6$ follow from the exact logarithmic coefficients and the product identity.

The reduction from z to $w = z^N$ removes the rotational repetition of the zeros.

Theorem 12.1 (Genus zero product). *For $N \geq 2$, let Λ_N be the nonzero zero multiset of Δ_N. Then*

$$\Delta_N(w) = \prod_{\lambda \in \Lambda_N} \left(1 - \frac{w}{\lambda}\right),$$

where the product converges locally uniformly. Consequently,

$$\mathsf{D}_N(z) = \prod_{\lambda \in \Lambda_N} \left(1 - \frac{z^N}{\lambda}\right).$$

Proof. By proposition 11.2, the order of Δ_N is $1/N < 1$. By corollary 11.13, it has infinitely many zeros, and corollary 11.15 gives

$$\sum_{\lambda \in \Lambda_N} |\lambda|^{-1} < \infty.$$

Hadamard factorization therefore gives a canonical product of genus zero. The only possible exponential factor is a nonzero constant because the order is below one. Since $\Delta_N(0) = 1$, that constant is one. Substitution of $w = z^N$ gives the second formula. □

Theorem 12.2 (Diagonal Fredholm determinant representation)**.** *Let I_N be a multiplicity-sensitive countable index set for Λ_N and put $\mathcal{H}_N = \ell^2(I_N)$. On its standard orthonormal basis, define the diagonal operator*

$$K_N e_\lambda = \lambda^{-1} e_\lambda.$$

Then K_N is trace class and

$$\Delta_N(w) = \det_F(I - wK_N).$$

For every $m \geq 1$,

$$\kappa_{N,m} = \operatorname{tr}(K_N^m).$$

Proof. By corollary 11.15,

$$\sum_{\lambda \in \Lambda_N} |\lambda|^{-1} < \infty.$$

Thus the diagonal operator K_N is trace class. Its Fredholm determinant is the locally uniform product

$$\det_F(I - wK_N) = \prod_{\lambda \in \Lambda_N} \left(1 - \frac{w}{\lambda}\right),$$

which equals $\Delta_N(w)$ by theorem 12.1.

Taking the logarithm gives

$$-\log \Delta_N(w) = \sum_{m \geq 1} \frac{w^m}{m} \operatorname{tr}(K_N^m)$$

near the origin. Comparison with the definition of $\kappa_{N,m}$ proves the trace identity. □

This is the standard diagonal encoding of this family's genus zero product. It is not a general trace-class resolvent or Feshbach theorem.

Corollary 12.3 (Complete symmetric function form)**.** *For a positive integer r, let each eigenvalue λ^{-1} of K_N be repeated r times. If h_n denotes the complete homogeneous symmetric function of degree n in this absolutely summable multiset, then*

$$\frac{P_{N,n}(r)}{(Nn)!} = h_n.$$

For arbitrary complex r, the Bell polynomial formula is the polynomial continuation obtained by multiplying every power sum by r.

Proof. The generating function for the complete homogeneous symmetric functions is

$$\prod_{\lambda \in \Lambda_N} \left(1 - \frac{w}{\lambda}\right)^{-r} = \Delta_N(w)^{-r}.$$

Coefficient comparison gives the first identity. Power sums determine the complete homogeneous functions through the Newton recurrence. Replacing each power sum by r times itself gives theorems 7.2 and 7.6. □

Proposition 12.4 (Products for the normalized sections)**.** *For every $0 \le j < N$, the entire function*

$$\Phi_{N,j}(w) = j! \sum_{n \ge 0} \frac{w^n}{(Nn+j)!}$$

has order $1/N$. If $N \ge 2$ and $\Lambda_{N,j}$ is its nonzero zero multiset, then

$$\Phi_{N,j}(w) = \prod_{\mu \in \Lambda_{N,j}} \left(1 - \frac{w}{\mu}\right).$$

For $m \ge 1$, put

$$\sigma_{N,j,m} = \sum_{\mu \in \Lambda_{N,j}} \mu^{-m}.$$

Then

$$C_{N,j,n}(r) = \frac{(Nn)!}{n!} B_n(r\sigma_{N,j,1}, r1!\sigma_{N,j,2}, \dots, r(n-1)!\sigma_{N,j,n}).$$

Proof. The coefficient formula for the order gives $1/N$. The function is not a polynomial. An entire function of order below one with only finitely many zeros would be a polynomial times a constant by Hadamard factorization, which is impossible here. The genus zero product follows as in theorem 12.1.

Taking the logarithm gives

$$-\log \Phi_{N,j}(w) = \sum_{m \ge 1} \sigma_{N,j,m} \frac{w^m}{m}.$$

Since $\mathsf{C}_{N,j}(z)^r = \Phi_{N,j}(z^N)^{-r}$, the Bell polynomial calculation from theorem 7.2 gives the coefficient formula. □

Definition 12.5. For $m \ge 1$, define

$$\sigma_{N,m} = \sum_{\lambda \in \Lambda_N} \lambda^{-m},$$

with multiplicities.

The series converges absolutely because the exponent of convergence of Λ_N is $1/N < 1$.

Theorem 12.6 (Newton sums for the reduced zeros)**.** *For $|w|$ below the least zero modulus, with the analytic logarithm normalized to zero at the origin,*

$$-\log \Delta_N(w) = \sum_{m \ge 1} \sigma_{N,m} \frac{w^m}{m}.$$

Thus

$$\kappa_{N,m} = \sigma_{N,m}.$$

If

$$a_m = \frac{(-1)^m}{(Nm)!}, \qquad a_0 = 1,$$

then

$$\sigma_{N,m} = -m a_m - \sum_{k=1}^{m-1} a_k \sigma_{N,m-k}.$$

Proof. Take the logarithm of the product in theorem 12.1. Absolute convergence in a disk below the first zero permits termwise expansion

$$-\log\Big(1 - \frac{w}{\lambda}\Big) = \sum_{m \geq 1} \frac{w^m}{m \lambda^m}.$$

This gives the logarithmic formula and identifies $\kappa_{N,m}$ with $\sigma_{N,m}$.

Differentiate the product identity in logarithmic form to obtain

$$\Delta_N'(w) = -\Delta_N(w) \sum_{m \geq 1} \sigma_{N,m} w^{m-1}.$$

Comparison of the coefficient of w^{m-1} gives

$$m a_m = -\sum_{k=0}^{m-1} a_k \sigma_{N,m-k}.$$

Solving for $\sigma_{N,m}$ gives the recurrence. □

Corollary 12.7 (Convergent Mittag–Leffler partial-fraction form)**.** *Away from the zeros of* D_N*,*

$$\mathsf{T}_N(z) = N z^{N-1} \sum_{\lambda \in \Lambda_N} \frac{1}{\lambda - z^N}.$$

The series converges absolutely and locally uniformly on compact sets away from the poles. Thus no regularizing terms are required.

Proof. Logarithmic differentiation of the genus zero product gives

$$-\frac{\Delta_N'(w)}{\Delta_N(w)} = \sum_{\lambda \in \Lambda_N} \frac{1}{\lambda - w}.$$

The series is locally uniform because $\sum_\lambda |\lambda|^{-1} < \infty$. Substitute $w = z^N$ and use the chain rule. □

Proposition 12.8 (Theta Bell polynomial formula)**.** *Put*

$$\tilde{\sigma}_{N,m} = (-1)^m \sigma_{N,m}.$$

Then

$$\Theta_{N,n}(u) = \frac{(Nn)!}{n!} B_n(\tilde{\sigma}_{N,1}(u^N + u), 1!\tilde{\sigma}_{N,2}(u^{2N} + u), \dots, (n-1)!\tilde{\sigma}_{N,n}(u^{nN} + u)).$$

Proof. The reduced unsigned denominator is

$$\Psi_N(w) = \sum_{n\geq 0} \frac{w^n}{(Nn)!} = \Delta_N(-w).$$

By theorem 12.6,

$$-\log \Psi_N(w) = \sum_{m\geq 1} \tilde{\sigma}_{N,m} \frac{w^m}{m}.$$

With $w = z^N$, the logarithm of the theta generating function is

$$\sum_{m\geq 1} \tilde{\sigma}_{N,m}(u^{Nm} + u)\frac{w^m}{m}.$$

The complete Bell polynomial formula follows. □

Corollary 12.9 (Tangent coefficients as zero sums)**.** *For every $n \geq 0$,*

$$T_N(n) = N(Nn + N - 1)!\sigma_{N,n+1}.$$

Proof. Combine proposition 8.2 and theorem 12.6. □

Corollary 12.10 (Reciprocal coefficients from zero data)**.** *For every $n \geq 0$,*

$$P_{N,n}(r) = \frac{(Nn)!}{n!} B_n(r\sigma_{N,1}, r1!\sigma_{N,2}, \dots, r(n-1)!\sigma_{N,n}).$$

Proof. Use theorems 7.2 and 12.6. □

Example 12.11 (The cosine orbit sums)**.** For $N = 2$,

$$\Delta_2(w) = \cos\sqrt{w}.$$

Its zeros are

$$\lambda_m = \pi^2 \left(m + \frac{1}{2}\right)^2, \qquad m \geq 0.$$

Hence

$$\sigma_{2,q} = \frac{2^{2q} - 1}{\pi^{2q}} \zeta(2q).$$

The identity in corollary 12.9 recovers the classical tangent coefficients from these zero power sums.

13 Coefficient asymptotics from the nearest zeros

For an entire function f, let $\mathcal{Z}(f)$ denote its multiset of nonzero zeros. Let r be a positive integer. Then D_N^{-r} is meromorphic and a simple zero of D_N becomes a pole of order r.

Theorem 13.1 (Nearest-pole coefficient expansion). *Let $r \in \mathbb{Z}_{>0}$. Suppose the zeros of D_N of least modulus R are simple and form the finite set $\mathcal{Z}_R$. Suppose the next zero modulus is $R_2 > R$. At each $\rho \in \mathcal{Z}_R$, write the principal part*

$$\mathsf{D}_N(z)^{-r} = \sum_{\ell=1}^{r} a_{\rho,\ell} \left(1 - \frac{z}{\rho}\right)^{-\ell} + O(1).$$

Then, for every R_* with $R < R_* < R_2$ and $m = Nn$, as $n \to \infty$,

$$P_{N,n}(r) = m! \sum_{\rho \in \mathcal{Z}_R} \sum_{\ell=1}^{r} a_{\rho,\ell} \binom{m+\ell-1}{\ell-1} \rho^{-m} + O_{R_*}(m! R_*^{-m}).$$

The leading Laurent coefficient is

$$a_{\rho,r} = (-\rho \mathsf{D}_N'(\rho))^{-r}.$$

Proof. Subtract the complete principal parts at the zeros on $|z| = R$. The remainder is holomorphic on $|z| < R_2$, hence on a neighborhood of every closed disk $|z| \le R_* < R_2$. Cauchy's estimate gives an ordinary coefficient bound $O_{R_*}(R_*^{-m})$. The coefficient of z^m in

$$\left(1 - \frac{z}{\rho}\right)^{-\ell}$$

is

$$\binom{m+\ell-1}{\ell-1} \rho^{-m}.$$

Multiplication by $m!$ converts the ordinary coefficient to $P_{N,n}(r)$.

Near a simple zero,

$$\mathsf{D}_N(z) = -\rho \mathsf{D}_N'(\rho) \left(1 - \frac{z}{\rho}\right) + O\left(\left(1 - \frac{z}{\rho}\right)^2\right).$$

Raising to the power $-r$ gives the formula for $a_{\rho,r}$. □

Corollary 13.2 (Leading dominant zero term). *Under the hypotheses of theorem 13.1,*

$$P_{N,n}(r) = m! \binom{m+r-1}{r-1} \sum_{\rho \in \mathcal{Z}_R} \frac{\rho^{-m}}{(-\rho \mathsf{D}_N'(\rho))^r} + O\left(m! m^{r-2} R^{-m}\right) + O_{R_*}\left(m! R_*^{-m}\right).$$

For $r = 1$, the first error term is absent.

Proof. The terms with $\ell < r$ in theorem 13.1 are bounded by a constant times $m^{r-2} R^{-m}$. The term with $\ell = r$ gives the displayed main term. □

Corollary 13.3 (Orbit reduced nearest circle term)**.** *Assume the hypotheses of theorem 13.1. Choose one representative from each orbit of the nearest zeros under multiplication by ω, and call the resulting set $\mathcal{Z}_R^{\mathrm{orb}}$. With $m = Nn$,*

$$P_{N,n}(r) = Nm!\binom{m+r-1}{r-1}\sum_{\tau\in\mathcal{Z}_R^{\mathrm{orb}}}\frac{\tau^{-m}}{(-\tau\mathsf{D}_N'(\tau))^r} + O\left(m!m^{r-2}R^{-m}\right) + O_{R_*}\left(m!R_*^{-m}\right).$$

For $r = 1$, the first error term is absent.

Proof. If $\rho = \omega^k\tau$, then

$$\mathsf{D}_N'(\rho) = \omega^{-k}\mathsf{D}_N'(\tau).$$

Therefore

$$-\rho\mathsf{D}_N'(\rho) = -\tau\mathsf{D}_N'(\tau).$$

Since $m = Nn$, one also has $\rho^{-m} = \tau^{-m}$. Every orbit therefore contributes N equal terms in corollary 13.2. □

Corollary 13.4 (Secant asymptotic)**.** *For $N = 2$ and $r = 1$,*

$$P_{2,n}(1) \sim 2(2n)!\left(\frac{2}{\pi}\right)^{2n+1}.$$

Proof. The nearest zeros of $\cos z$ are $\rho = \pm\pi/2$, both simple. Their contributions in corollary 13.2 are equal. □

14 Zero zeta functions

The assumption $N \geq 2$ remains in force throughout this section. The zero counting theorem gives the exact abscissae of absolute convergence. The exponentially accurate zero locations give more. They compare the zero zeta with a Hurwitz zeta function.

Definition 14.1. Define

$$\mathfrak{Z}_N^{\mathrm{abs}}(s) = \sum_{\rho\in\mathcal{Z}(\mathsf{D}_N)}|\rho|^{-s}$$

and

$$\mathfrak{Z}_N^{\mathrm{red}}(s) = \sum_{\lambda\in\Lambda_N}|\lambda|^{-s},$$

with multiplicities. Each term means $e^{-s\log|\rho|}$ or $e^{-s\log|\lambda|}$.

Theorem 14.2 (Abscissae of absolute convergence)**.** *The series $\mathfrak{Z}_N^{\mathrm{abs}}$ converges absolutely and locally uniformly when*

$$\Re s > 1.$$

It fails to converge absolutely when $\Re s \le 1$. The series $\mathfrak{Z}_N^{\mathrm{red}}$ converges absolutely and locally uniformly when

$$\Re s > \frac{1}{N}.$$

It fails to converge absolutely when $\Re s \le 1/N$. Both functions are holomorphic on their open half-planes of absolute convergence.

Proof. By theorem 11.14, the zeros of D_N have linear counting density, while the reduced zeros have counting order $R^{1/N}$. Partial summation gives absolute convergence in the stated open half-planes. The positive terms obtained after taking absolute values diverge on the boundary and to its left. Local uniform convergence follows by the Weierstrass test on compact subsets. □

Theorem 14.3 (Meromorphic continuation of the symmetry-orbit zero zeta)**.** *Put*

$$A_N = \frac{\pi}{\sin(\pi/N)}.$$

There is an entire function Q_N^{red} such that

$$\mathfrak{Z}_N^{\mathrm{red}}(s) = A_N^{-Ns}\zeta\left(Ns, \frac{1}{2}\right) + Q_N^{\mathrm{red}}(s)$$

on $\Re s > 1/N$. The right-hand side gives a meromorphic continuation to $\mathbb{C}$ with one simple pole at $s = 1/N$. Its residue is

$$\frac{\sin(\pi/N)}{N\pi}.$$

There is an entire function Q_N^{abs} such that, initially on $\Re s > 1$,

$$\mathfrak{Z}_N^{\mathrm{abs}}(s) = NA_N^{-s}\zeta\left(s, \frac{1}{2}\right) + Q_N^{\mathrm{abs}}(s).$$

Thus $\mathfrak{Z}_N^{\mathrm{abs}}$ continues meromorphically to $\mathbb{C}$ with one simple pole at $s = 1$, of residue

$$\frac{N\sin(\pi/N)}{\pi}.$$

Proof. For $N = 2$, corollary 11.10 and $\mathsf{D}_2(z) = \Delta_2(z^2)$ give

$$\lambda_m = \pi^2\left(m + \frac{1}{2}\right)^2, \qquad m \ge 0.$$

Thus the first formula holds with $Q_2^{\mathrm{red}} = 0$.

For $N \ge 3$, corollary 11.13 writes all but finitely many reduced zeros as

$$\lambda_m = \left(A_N\left(m + \frac{1}{2}\right)\right)^N (1 + \varepsilon_m)$$

where $\lambda_m > 0$, $1 + \varepsilon_m > 0$, and

$$\varepsilon_m = O\left(t_m^{-1}e^{-\gamma t_m}\right).$$

On every compact set of s values,

$$(1+\varepsilon_m)^{-s}-1=O(|\varepsilon_m|)$$

uniformly for large m. The difference

$$\lambda_m^{-s}-A_N^{-Ns}\left(m+\frac{1}{2}\right)^{-Ns}$$

therefore has a normally convergent sum on every compact subset of $\mathbb{C}$. Indeed, the model power has at most polynomial growth in m on a compact set, while $|\varepsilon_m|$ decays exponentially. The finitely many exceptional terms $|\lambda|^{-s}$ and the finitely many omitted model terms are entire functions of s. Their addition to the normally convergent tail defines Q_N^{red} and proves the first formula. Thus that formula is established for every $N\geq 2$.

In either case, the Hurwitz zeta function has a meromorphic continuation with one simple pole at its first argument equal to one and residue one. Composition with Ns gives the stated pole and residue.

For every $N\geq 2$, each nonzero zero λ of Δ_N gives N zeros of D_N, each of modulus $|\lambda|^{1/N}$. Hence

$$\mathfrak{Z}_N^{\mathrm{abs}}(s)=N\mathfrak{Z}_N^{\mathrm{red}}(s/N).$$

Substitution gives the second formula with

$$Q_N^{\mathrm{abs}}(s)=NQ_N^{\mathrm{red}}(s/N),$$

which is entire. □

Corollary 14.4 (Exact zeta models in orders two and four)**.** *For $N=2$,*

$$\mathfrak{Z}_2^{\mathrm{red}}(s)=\pi^{-2s}\zeta\left(2s,\frac{1}{2}\right)$$

and

$$\mathfrak{Z}_2^{\mathrm{abs}}(s)=2\pi^{-s}\zeta\left(s,\frac{1}{2}\right).$$

For $N=4$, put $A_4=\sqrt{2}\pi$. Then

$$\mathfrak{Z}_4^{\mathrm{red}}(s)=A_4^{-4s}\zeta\left(4s,\frac{1}{2}\right)$$

and

$$\mathfrak{Z}_4^{\mathrm{abs}}(s)=4A_4^{-s}\zeta\left(s,\frac{1}{2}\right).$$

In these two cases the entire correction terms in theorem 14.3 vanish identically.

Proof. For $N=2$, the reduced zeros are

$$\pi^2\left(m+\frac{1}{2}\right)^2,\qquad m\geq 0.$$

For $N=4$, example 11.12 gives the reduced zeros

$$A_4^4\left(m+\frac{1}{2}\right)^4,\qquad m\geq 0.$$

Summing their negative powers proves the reduced formulas. The absolute formulas follow from

$$\mathfrak{Z}_N^{\mathrm{abs}}(s) = N\mathfrak{Z}_N^{\mathrm{red}}(s/N).$$

□

Definition 14.5. Choose a logarithm for each member of the finite exceptional part of Λ_N. Use the real logarithm for the eventual positive sequence. Define

$$\mathfrak{Z}_N^{\mathrm{orb}}(s) = \sum_{\lambda\in\Lambda_N} e^{-s\operatorname{Log}\lambda}.$$

This series is initially defined on $\Re s > 1/N$.

Corollary 14.6 (Branch-dependent symmetry-orbit continuation)**.** *The branch-dependent orbit zeta $\mathfrak{Z}_N^{\mathrm{orb}}$ continues meromorphically to $\mathbb{C}$. It has the same single pole and residue as $\mathfrak{Z}_N^{\mathrm{red}}$. Changing the logarithms of the finite exceptional zeros changes the continuation by an entire function.*

Proof. The tail consists of positive reduced zeros, so it has the same comparison with the Hurwitz zeta. Only finitely many terms depend on the branch choices. □

Definition 14.7. Let q be a positive integer. Choose logarithms of the simple zeros of D_N with arguments in a fixed bounded interval. Define

$$\mathfrak{Z}_{N,q}^{\mathrm{der}}(s) = \sum_{\substack{\mathsf{D}_N(\rho)=0\\ \mathsf{D}_N'(\rho)\neq 0}} \frac{e^{-s\operatorname{Log}\rho}}{\mathsf{D}_N'(\rho)^q}.$$

Theorem 14.8 (Derivative-weighted convergence)**.** *If $N \geq 3$ and $q \geq 1$, the series $\mathfrak{Z}_{N,q}^{\mathrm{der}}(s)$ converges locally uniformly for every $s \in \mathbb{C}$. It therefore defines an entire function.*
For $N = 2$, the series converges absolutely and locally uniformly on $\Re s > 1$, where it defines a holomorphic function.

Proof. For $N \geq 3$, theorem 11.7 gives

$$|\mathsf{D}_N'(\rho)|^{-1} \leq Ce^{-c_N|\rho|}$$

for every sufficiently large simple zero. Let L be a compact subset of the s plane. Because the chosen arguments of the zeros lie in a fixed bounded interval, there are constants C_L and M_L such that

$$\sup_{s\in L} |e^{-s\operatorname{Log}\rho}| \leq C_L|\rho|^{M_L}$$

for every sufficiently large zero. Hence the absolute value of the corresponding summand is at most

$$C_L'|\rho|^{M_L}e^{-qc_N|\rho|}.$$

The linear zero counting estimate makes this majorant summable. The Weierstrass test gives absolute local uniform convergence of the series of entire summands, so its sum is entire. The finite exceptional set causes no difficulty.

For $N = 2$, the derivative of the cosine has modulus one at every zero. The assertion then follows from the linear zero spacing. □

Remark 14.9. The derivative weighted sum omits the finitely many zeros that might be multiple. A treatment of those zeros requires the complete principal parts of D_N^{-q} rather than the single derivative weight.

Part V Scope and formal evidence

15 Scope boundaries and further questions

The large-zero theorem settles eventual simplicity but does not classify the finite exceptional set. A remaining question is whether every zero of D_N is simple and every reduced zero of Δ_N is positive. Both statements hold for $N = 2$ and $N = 4$. The argument here proves them only outside a compact set for general N.

The leading correction in theorem 11.8 may admit iteration to a full exponential expansion for τ_m. Such an expansion is an open problem, not a result of this chapter. It could refine the entire correction in the zero-zeta comparison and lead to explicit special values.

The symmetry-orbit zero zeta has a meromorphic continuation with one pole. No functional equation is obtained. A branch dependent sum over the N roots of each reduced zero must specify logarithms on the rays.

The coefficient families in sections 7, 9 and 10 have exact formal parameter-addition laws. Their arithmetic behavior, including odd-prime congruences, dyadic weighting, and p-adic or Amice–Mahler interpolation, belongs to Chapter IX. General bounded-operator semigroup, resolvent, and Feshbach theory belongs to Chapter VI. Chapter III covers general nonlinear operation trees, while abstract completion and occupancy questions belong to Chapter X. None of those theories is inferred from the scalar complex-analytic results proved here.

16 Formalization ledger

The source records specify the Lean and Mathlib environment used for the recorded verification [dMU21, The20]. Its 151 theorem-map rows resolve to 1164 unique declarations in Komorebi.PaperIV: 953 theorem declarations and 211 constructions. The namespace audit scanned 2234 declarations, including 1999 theorem declarations. The build and kernel audit, with warnings treated as errors, allowed only propext, Classical.choice, and Quot.sound as standard transitive axioms.

Manuscript records	Compiled Chapter IV scope
3.1–3.3	Actual holomorphic root filters, convergent coefficient selection, orthogonal resolution, the exact range–covariance characterization, and the complete pointwise cyclic product law. The formal versions hold as well.
4.1–4.5 and 5.1–5.8	Entire sections, finite exponential formulas, derivative cycles, both Wronskians, and unique entire ODE basis expansions. Both pointwise addition laws and denominator symmetries are checked. The exact ordinary hypergeometric formula follows from the omitted-parameter list, Gamma normalization, and Mathlib's everywhere convergent series.

Manuscript records	Compiled Chapter IV scope (continued)
6.1–6.6	Quotient-algebra power bases and actual exponential-series convergence in the finite-dimensional coefficient topology. Both pointwise section-coordinate identities. Signed-shift powers, all matrix-exponential entries, group laws and trace are checked. The characteristic polynomial $X^N + 1$, its distinct roots, exact spectrum and determinant one are proved.
7.1–7.8	Named reciprocal families, exact parameter recurrences, Bell formula, parameter degree and leading coefficient, and polynomial divisibility. Parameter addition holds with any single fixed holomorphic logarithm on the stated domain. Newton's recurrence gives the literal Hessenberg determinant. All three compact reciprocal-polynomial formulas are checked.
8.1–8.5	The full tangent coefficient series has the exact kappa and factorial normalization, including $N = 1$. Reciprocal differentiation holds on every stated fixed logarithm domain. The actual higher Riccati identity holds at every point where the denominator is nonzero.
9.1–9.10	Arbitrary-parameter residue reciprocals and their analytic germs, polynomial degree and leading term, sparse Appell laws, joint convolution, monicity and covariance. The Hasse and derivative-truncation encodings are proved equal. The named Appell polynomial satisfies both the forward action and the exact $\Phi_{N,j}(\partial_x^N)^r$ inverse formula.
10.1–10.4	Exact formal theta reconstruction, polynomial lift and degree, literal divisibility by u and, for even N, by $u + 1$, and the theta Newton recurrence with its first coefficient.
11.2, 11.4–11.11	Two-sided maximum-modulus bounds prove exact order and type. The indicator limsup formula holds in every direction, including ties. Dominant-exponential errors and zero freeness are uniform on compact sets avoiding ties. Near cancellation and the order-two zero set are checked. For $N \geq 3$, gap and remainder bounds, sector confinement, global exhaustion and multiplicity identities give the full large-zero-string theorem. The real leading correction, quadratic exponential error, signed derivative asymptotic, eventual alternating sign and cubic correction are checked.
11.13–11.15	The reduced series is entire, and $\mathsf{D}_N(z) = \Delta_N(z^N)$ preserves multiplicity away from zero. Finite exceptional sets, eventual positive simple reduced zeros, and their stated displacement are checked, including $N = 2$. Both full disk-counting laws retain all exceptional multiplicities. The full-zero reciprocal sums have exact thresholds $s > 1$ and $s > 1/N$, including boundary divergence.
12.1	For every $N \geq 2$, the complete multiplicity-indexed product converges uniformly on compact sets and equals Δ_N. Jensen and Poisson estimates prove rigidity of the actual quotient from derived growth bounds. No growth certificate remains. The original-variable product for D_N is also checked.
12.2–12.4	The actual diagonal operator is trace class, with a basis-independent trace and determinant given by limits of actual finite compressions. Infinite complete homogeneous coefficients are limits of finite monomial sums. Every normalized section has exact order $1/N$. For $N \geq 2$ its full zero product and zero-moment Bell formulas are proved. The actual complex-power germ at zero has the named residue Taylor coefficients.

Manuscript records	Compiled Chapter IV scope (continued)
12.6–12.10	The normalized logarithm on each zero-free disk has the convergent zero-moment series. Its coefficients equal $\kappa_{N,m}$ and satisfy Newton's recurrence. The Mittag–Leffler tangent expansion converges absolutely and locally uniformly away from poles. Tangent coefficients and the named reciprocal and theta families have their exact zero-moment Bell formulas. The full tangent series is identified with the analytic function.
13.1–13.4	Analytic division constructs each simple-zero principal part. Its leading coefficient is uniquely determined. Removing the principal parts gives a holomorphic remainder and the exact factorial-scaled Cauchy estimate. Both leading-term errors are bounded, with the first absent for $r = 1$. Rotation gives the orbit-reduced main term. The secant formula holds as a ratio limit, with an exponentially vanishing relative error.
14.2–14.4, 14.6	The exact all-zero sums retain analytic multiplicity. Their sharp absolute-convergence domains, boundary divergence and local uniform convergence are checked. Both Hurwitz comparisons, entire corrections, global meromorphic continuations, unique simple poles and exact residues are proved. The order-two/four models have identically zero corrections. Branch choices on the finite exceptional part change the continuation by an entire function and preserve its pole and residue.
14.8	The derivative-weighted series uses every actual simple zero and genuine logarithms with bounded arguments. Absolute and local uniform convergence and an entire sum are proved for $N \geq 3$. For $N = 2$, absolute/local uniform convergence and holomorphy hold on $\Re s > 1$.

The declaration-level theorem map is authoritative for the formal contract. It records full scoped coverage of all 68 theorem-like records: 68 full, zero partial, and zero unmapped. Rows 143–151 close the final fifteen partial records without altering their statements or assumptions. Imported standard fact 11.1 remains imported. Formal identities, normalized analytic germs, and fixed logarithm domain identities retain their respective stated meanings. Finite computations do not change these proof boundaries.

17 Conclusion

Cyclic projection turns the exponential into a finite differential basis, and the signed cycle produces the distinguished denominator D_N. Its reciprocal powers yield Bell and Newton recurrences, an additive parameter law, Appell systems, and theta polynomials. Reduction by $w = z^N$ then connects the reciprocal coefficients to a genus zero product and reduced-zero power sums.

Classical two term localization, specialized to the regular exponent polygon and combined with reflection symmetry, places all sufficiently large zeros on symmetry rays. The displacement, derivative and counting estimates yield the family-specific Hurwitz comparison for the orbit zero zeta.

A Visual atlas

The twenty-six plates retain the worked models of the separate visual guide. The stable labels for Plates IV.1–IV.26 name mathematical mechanisms and point back to the relevant statements. The plates distinguish exact identities, rational certificates, sampled curves, and inherited formal evidence.

Three routes through the atlas

For algebra, read Plates IV.3–IV.16. For zero geometry and its consequences, read Plates IV.17–IV.24. Plate IV.25 follows one coefficient through the complete calculation. Plate IV.26 explains the evidence attached to the models.

Plate IV.1 / READER'S GUIDE

A coefficient can become zero data

Chapter IV: Section 2, Section 7, Section 11, Section 12.

This guide follows a concrete analytic family through three descriptions: selected Taylor coefficients, a finite differential system, and the zeros of an entire function. The formulas are linked, but the evidence needed for each description is different.

COEFFICIENT LANES

z^0	z^3	z^6	z^9
z^1	z^4	z^7	z^{10}
z^2	z^5	z^8	z^{11}

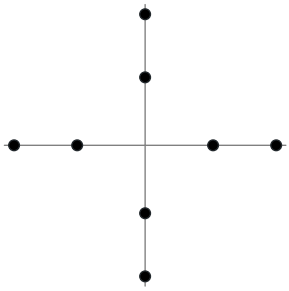
$m \equiv j \pmod 3$

RECIPROCAL COEFFICIENTS

1	0	0
$-\frac{1}{2}$	1	0
$\frac{1}{24}$	$-\frac{1}{2}$	1

$1, \quad \frac{1}{2}, \quad \frac{5}{24}, \ldots$

ZERO ORBITS



$\{\rho, i\rho, -\rho, -i\rho\}$

Keep three small models in view

For $N = 2$, the signed denominator is $D_2(z) = \cos z$. Its reciprocal is the familiar secant series. For $N = 3$, a third exponential perturbs the zeros of a dominant cosine. For $N = 4$, an exact factorization gives a complete zero picture:

$$D_4(z) = \cos(z/\sqrt{2})\cosh(z/\sqrt{2}).$$

The examples move between these orders deliberately. Order two makes the coefficient arithmetic transparent. Order three shows a genuine displacement. Order four makes orbit reduction completely explicit.

Three routes through the guide

Route	Pages and question
Algebra first	Plate IV.3 to Plate IV.16: how do selection, signs, reciprocals, and finite differential operators work?
Zeros first	Plate IV.17 to Plate IV.24: where can cancellation happen, and how does it determine coefficients?
One full calculation	Plate IV.25: derive the same number, 69, from denominator coefficients and from zero moments.

Read this as: The pictures carry computable objects. They do not replace the global analytic arguments or extend the inherited Lean contract.

Exact identities and rational certificates are distinguished from sampled curves. All plates are reproducible vector drawings. The source package preserves the original proofread manuscript.

Plate IV.2 / OBJECT ATLAS

Two variables, several normalizations

Chapter IV: Section 2, Definition 5.7, Definition 7.1, Definition 9.1.

The most important coordinate change is $w = z^N$. A term of degree n in w has degree Nn in z. Its coefficient does not change, but an exponential-generating-function label does.

$$\underbrace{b_{N,n}(r)\,w^n}_{\text{ordinary coefficient}} = \underbrace{P_{N,n}(r)\frac{z^{Nn}}{(Nn)!}}_{\text{block exponential coefficient}}, \qquad P_{N,n}(r) = (Nn)!\,b_{N,n}(r).$$

Object	Definition and role
$E_{N,j}(z)$	$\displaystyle\sum_{n\geq 0}\frac{z^{Nn+j}}{(Nn+j)!}$ selects one residue class of the exponential.
$D_{N,j}(z)$	The same series with $(-1)^n$. One full derivative cycle contributes a minus sign.
$D_N(z)$	$D_{N,0}(z)$, the denominator whose zeros control the signed reciprocal family.
$\Delta_N(w)$	$\displaystyle\sum_{n\geq 0}\frac{(-1)^n w^n}{(Nn)!}$. Thus $D_N(z) = \Delta_N(z^N)$.
$P_{N,n}(r)$	Block-EGF coefficients of D_N^{-r}. They are polynomials in r.
$\kappa_{N,m}$	$-\log \Delta_N(w) = \displaystyle\sum_{m\geq 1}\kappa_{N,m}w^m/m$.
$T_N(z)$	$-D_N'(z)/D_N(z)$, a function, not the coefficient $T_N(n)$.
$\Phi_{N,j}(w)$	$j!\displaystyle\sum_{n\geq 0}\frac{w^n}{(Nn+j)!}$ removes the forced zero of $E_{N,j}$.
$C_{N,j}(z)$	$1/\Phi_{N,j}(z^N)$. This is generally not $1/D_N(z)$.
Λ_N	Zeros of Δ_N, counted with multiplicity, one per rotational orbit of zeros of D_N.

A normalization test

The coefficient $5/24$ of w^2 in $\Delta_2(w)^{-1}$ is the same term as $5z^4/4!$. It is not $5w^2/2!$. Keeping the two factorials separate prevents errors in Bell formulas, convolution weights, and zero-moment reconstruction.

> **Read this as:** Use z for differentiation and rotational geometry. Use w for reciprocal recurrences, canonical products, and orbit-zero sums.

Plate IV.3 / CYCLIC PROJECTION

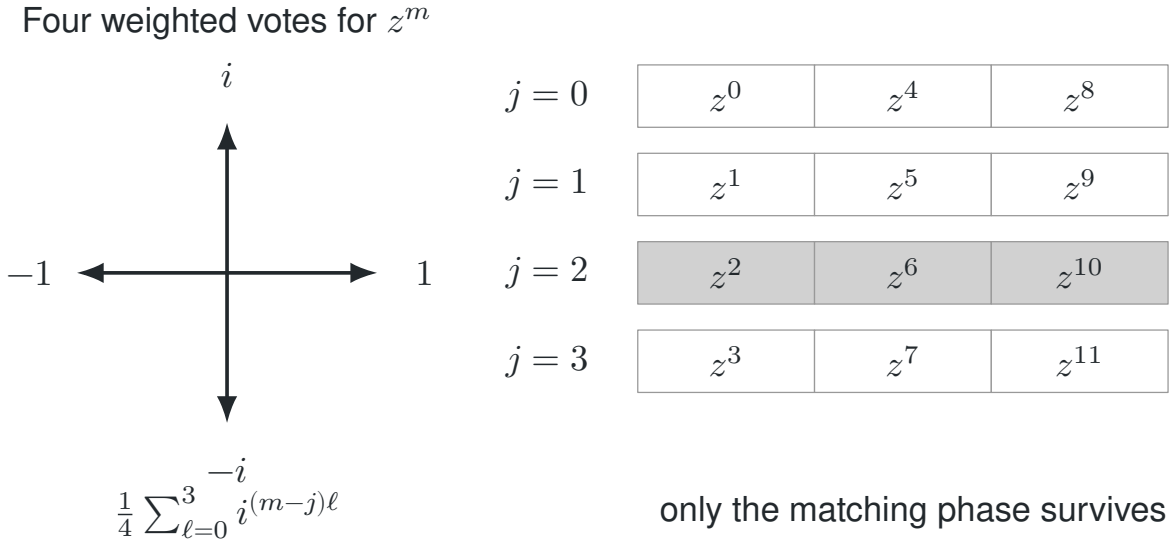
A root filter is an exact vote

Chapter IV: Theorem 3.1, Proposition 3.2.

For $N = 4$, write $\omega = i$. To extract residue j, evaluate the function at four rotated arguments and attach the phase $i^{-j\ell}$:

$$(\mathcal{B}_{4,j}F)(z) = \frac{1}{4}\sum_{\ell=0}^{3} i^{-j\ell}F(i^{\ell}z).$$

A monomial $a_m z^m$ is multiplied by the finite average of $i^{(m-j)\ell}$. Matching phases all vote 1. Nonmatching phases cancel exactly.



An example that can be checked coefficient by coefficient

Take $F(z) = \sum_{m=0}^{12}(m+1)z^m$. Then

$$\mathcal{B}_{4,2}F = 3z^2 + 7z^6 + 11z^{10}.$$

Applying the same selector again changes nothing. Applying a different selector gives zero. Summing the four selected polynomials returns the original F.

Selection is also a symmetry condition

For the selected polynomial F_2, rotation gives $F_2(iz) = -F_2(z)$. More generally, a residue-j section satisfies $F_j(\omega z) = \omega^j F_j(z)$. The phase label therefore describes both the coefficient addresses and the response to rotation.

> **Read this as:** This is not an approximate Fourier fit. The finite root sum is exactly 0 or 1 on every coefficient.

The checker compares the actual complex average with independent coefficient selection for this polynomial. The drawing shows the four complex phases, not a numerical convergence argument.

Plate IV.4 / CYCLIC CONVOLUTION

Products mix the residue lanes

Chapter IV: Theorem 3.3.

Selection separates a function additively. It does not make multiplication diagonal. A residue-i term multiplied by a residue-j term lands in residue $i + j \pmod N$. For $N = 3$, use

$$F = 1 + 2z + 3z^2 + z^3, \qquad G = 2 + z + z^2.$$

The table displays every product of their three selected sections.

	2	z	z^2
$z^3 + 1$	$2z^3 + 2$	$z^4 + z$	$z^5 + z^2$
$2z$	$4z$	$2z^2$	$2z^3$
$3z^2$	$6z^2$	$3z^3$	$3z^4$

$$\mathcal{B}_{3,0}(FG) = 2 + 7z^3$$

shaded cells: $(0,0), (1,2), (2,1)$ all land in residue 0

The shaded cells contribute to residue zero:

$$2(1 + z^3) + (2z)(z^2) + (3z^2)z = 2 + 7z^3.$$

Keeping only the product of the residue-zero inputs would give $2 + 2z^3$ and lose five units of the z^3 coefficient. The mixed cells are part of the exact result.

The general rule

$$\mathcal{B}_{N,k}(FG) = \sum_{i+j \equiv k \ (N)} (\mathcal{B}_{N,i}F)(\mathcal{B}_{N,j}G).$$

Every pair belongs to exactly one output residue. No term is discarded and no coefficient is estimated. This finite cyclic convolution is the algebra behind the later addition formulas and block-coefficient recurrences.

> **Read this as:** A packet label tells you where an interaction lands, not that interactions with other packets vanish.

All nine table entries and all three output sections are checked as exact polynomial identities. This is the scalar cyclic specialization used in this chapter.

Plate IV.5 / DIFFERENTIAL SYSTEM

One signed edge determines the cycle

Chapter IV: Theorem 5.3, Theorem 6.4, Corollary 6.5, Corollary 6.6.

Let $J_3e_0 = e_1$, $J_3e_1 = e_2$, and $J_3e_2 = -e_0$. A complete turn changes sign, so $J_3^3 = -I$. The signed sections are the coordinates of $e^{zJ_3}e_0$.

J_3

0	0	-1
1	0	0
0	1	0

e^{zJ_3}

D_0	$-D_2$	$-D_1$
D_1	D_0	$-D_2$
D_2	D_1	D_0

$$J_3^3 = -I$$

$$\mathbf{D}_3(0) = e_0$$

$$\det e^{zJ_3} = 1$$

$$D_0 \mapsto -D_2, \qquad D_1 \mapsto D_0, \qquad D_2 \mapsto D_1$$

Here D_j inside the matrix means $D_{3,j}(z)$. For example,

$$D_{3,0} = 1 - \frac{z^3}{3!} + \frac{z^6}{6!} - \cdots, \qquad D_{3,1} = z - \frac{z^4}{4!} + \cdots.$$

Rows are output coordinates and columns are input coordinates. The upper-triangular minus signs record crossing the signed edge while moving from the column to the row. They are not arbitrary orientation choices.

Two checks connect the matrix and the scalar equation

The initial derivative matrix is the identity. Each initial jet $(y(0), y'(0), y''(0))$ therefore determines a unique linear combination. Also,

$$\frac{1}{3}\operatorname{Tr}(e^{zJ_3}) = D_3(z), \qquad \det(XI - J_3) = X^3 + 1.$$

The roots of -1 appearing later in the exponential sum are already the spectrum of this concrete matrix.

> **Read this as:** The functions, their initial data, the addition law, and the exponent polygon come from one signed shift.

The exact tests check every matrix-entry Taylor coefficient through degree 14 for $N = 1, \ldots, 5$. The displayed determinant-one law is for $N \geq 2$. $N = 1$ is a separate boundary case.

Plate IV.6 / FINITE ALGEBRA

Addition is quotient multiplication

Chapter IV: Proposition 5.5, Theorem 6.2.

Work in $\mathbb{C}[\xi]/(\xi^3+1)$. Every element has coordinates in $1, \xi, \xi^2$. The entire exponential has the finite-coordinate form

$$e^{z\xi} = D_{3,0}(z) + D_{3,1}(z)\xi + D_{3,2}(z)\xi^2.$$

The multiplication table makes the carry sign unavoidable.

	1	ξ	ξ^2
1	1	ξ	ξ^2
ξ	ξ	ξ^2	-1
ξ^2	ξ^2	-1	$-\xi$

$$\xi^3 = -1$$

wrap once: change sign

$$\xi^2\xi^2 = -\xi$$

$$D_{3,0}(z+w) = D_{3,0}(z)D_{3,0}(w) - D_{3,1}(z)D_{3,2}(w) - D_{3,2}(z)D_{3,1}(w)$$

Multiply the coordinate forms of $e^{z\xi}$ and $e^{w\xi}$. Their product is $e^{(z+w)\xi}$. The coefficient of 1 receives one contribution without a wrap and two contributions with a wrap. Reading that coefficient gives the displayed addition law.

Compare the unsigned model

In $\mathbb{C}[\xi]/(\xi^3 - 1)$, the same table has no minus signs. Its exponential coordinates are the unsigned sections $E_{3,j}$. Thus the difference between the two differential systems is a single algebraic relation, not a change in how coefficients are grouped. For general N, the carry factor is

$$(-1)^{\lfloor (i+j)/N \rfloor}, \qquad 0 \le i, j < N.$$

Only zero or one wrap is possible in a product of two coordinate basis elements.

> **Read this as:** The twisted addition law is the ordinary exponential group law expressed in a signed cyclic algebra.

Bivariate Taylor identities are checked through total degree ten in every section for $N = 2, 3, 4$. Omitting the carry sign is one of the deliberately rejected controls.

Plate IV.7 / FACTORIAL NORMALIZATION

Why a hypergeometric name appears

Chapter IV: Theorem 4.5.

The hypergeometric form is another way to organize the same coefficients. It does not introduce an extra function into the construction. Split a factorial into its residue classes and divide each factor by N:

$$(Nn+j)! = j!N^{Nn}n! \prod_{a\in A_{N,j}} (a)_n.$$

The parameter equal to 1 is omitted from $A_{N,j}$ because the hypergeometric convention already supplies $(1)_n = n!$.

$$N=3,\ j=1,\ n=2: \quad 7! = 1!\,3^6\,2!\,(2/3)_2(4/3)_2$$

$\frac{2}{3}$	$\frac{5}{3}$
1	2
$\frac{4}{3}$	$\frac{7}{3}$

$(2/3)_2$

$(1)_2 = 2!$

$(4/3)_2$

$$A_{3,1} = \{2/3, 4/3\}$$

$$E_{3,1}(z) = z\,{}_0F_2\left(;\frac{2}{3},\frac{4}{3};\frac{z^3}{27}\right)$$

The omitted unit parameter is already the hypergeometric $n!$.

Check the displayed case

For $N=3, j=1, n=2$, the six factors after the initial $1!$ are $2,3,4,5,6,7$. Dividing by three and grouping them gives the three rows of the plate. The middle row supplies $2!$. The others supply $(2/3)_2$ and $(4/3)_2$. The right-hand side is exactly 5040. Consequently,

$$E_{3,1}(z) = z\left(1+\frac{z^3}{4!}+\frac{z^6}{7!}+\cdots\right).$$

The prefactor z retains the forced zero at the origin. The hypergeometric argument $z^3/27$ retains the scaling N^N.

> **Read this as:** Parameters, the omitted unit parameter, the prefactor, and the argument scaling all come from factorial bookkeeping.

Five choices of (N, j) are checked against the exact factorial identity for $n = 0, \dots, 7$. These finite checks support the example. The general analytic identification belongs to the inherited manuscript.

Plate IV.8 / THE REDUCED VARIABLE

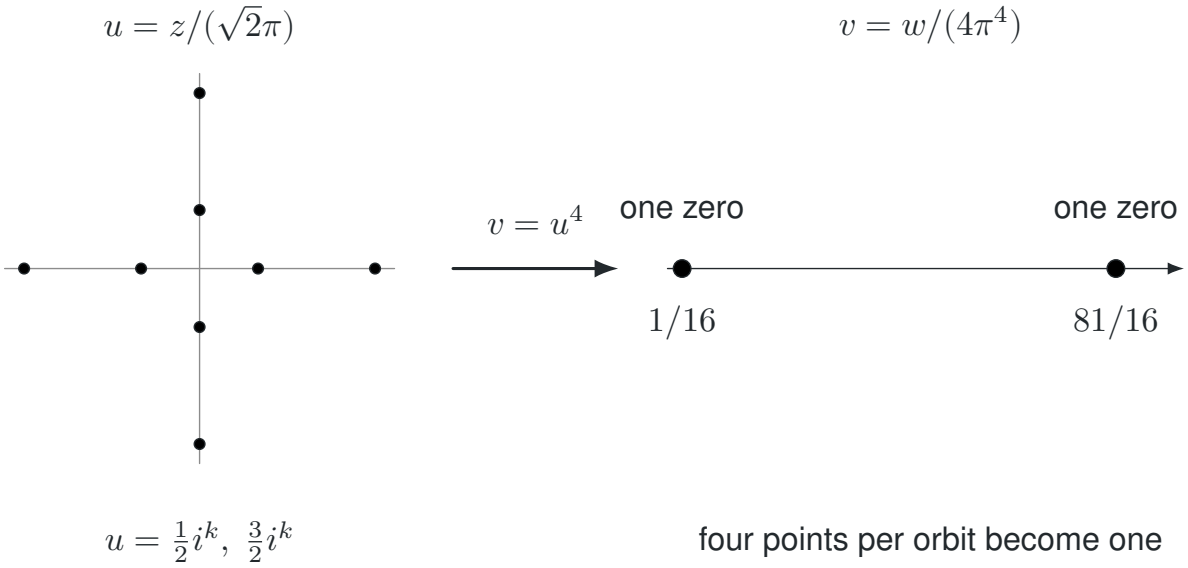
Four zeros become one orbit zero

Chapter IV: Definition 5.7, Example 11.12, Corollary 11.13.

For $N = 4$ the factorization is exact:

$$D_4(z) = \cos(z/\sqrt{2})\cosh(z/\sqrt{2}), \qquad D_4(z) = \Delta_4(z^4).$$

Set $A_4 = \sqrt{2}\pi$. All the zeros are $A_4(m + 1/2)i^k$, with $m \geq 0$ and $0 \leq k < 4$. Each zero is simple.



The drawing uses $u = z/A_4$ and $v = w/A_4^4$, so its marked coordinates are rational. The first four-point orbit has fourth power $1/16$. The second has fourth power $81/16$.

Multiplicity is not multiplied in the reduced function

At a nonzero point, $z \mapsto z^4$ has nonzero derivative. It preserves the local multiplicity of a zero. Four distinct simple zeros of D_4 therefore correspond to one simple zero of Δ_4, not one zero of multiplicity four. The reduced zeros are

$$\lambda_m = A_4^4(m + 1/2)^4 = 4\pi^4(m + 1/2)^4.$$

This is why an orbit factor in the z plane is $1 - z^4/\lambda_m$. It is the product of the four corresponding linear factors, with their phases included.

> **Read this as:** Orbit reduction removes repetition. It changes growth and counting exponents, while preserving each nonzero zero's local multiplicity.

The quartic factorization, its differential equation, and the displayed orbit powers are checked independently. Here the complete zero set is known exactly. A finite numerical root search is not being used to establish it.

Plate IV.9 / COEFFICIENT RECONSTRUCTION

A reciprocal is a triangular solve

Chapter IV: Theorem 7.3, Theorem 7.6.

Start in the reduced variable with

$$\Delta_2(w) = 1 - \frac{w}{2} + \frac{w^2}{24} - \frac{w^3}{720} + \cdots.$$

Write $1/\Delta_2(w) = \sum b_n w^n$. Matching the product with 1 gives a unit lower-triangular system. At each row there is exactly one new unknown.

denominator convolution				b		unit series
1	0	0	0	1	$=$	1
$-\frac{1}{2}$	1	0	0	$\frac{1}{2}$		0
$\frac{1}{24}$	$-\frac{1}{2}$	1	0	$\frac{5}{24}$		0
$-\frac{1}{720}$	$\frac{1}{24}$	$-\frac{1}{2}$	1	$\frac{61}{720}$		0

$$b_1 = \tfrac{1}{2}$$

$$b_2 = \tfrac{1}{4} - \tfrac{1}{24}$$

$$b_3 = \tfrac{61}{720}$$

$$P_{2,n}(1) = (2n)!b_n : \qquad 1,\ 1,\ 5,\ 61$$

The first three equations are

$$b_1 - \tfrac{1}{2} = 0, \qquad b_2 - \tfrac{1}{2}b_1 + \tfrac{1}{24} = 0, \qquad b_3 - \tfrac{1}{2}b_2 + \tfrac{1}{24}b_1 - \tfrac{1}{720} = 0.$$

They yield $b_1 = 1/2$, $b_2 = 5/24$, and $b_3 = 61/720$. Multiplication by $(2n)!$ produces the secant coefficients $1, 5, 61$.

Why this is an exact finite calculation

To compute the coefficient of w^n, only denominator coefficients through w^n are needed. Later terms of the entire series cannot affect it. A finite series inverse is therefore exact at its declared truncation order, even though evaluating that truncated series is only an approximation to the full analytic function away from the origin.

> **Read this as:** A unit constant term makes coefficient recovery triangular. It does not by itself specify the radius of analytic convergence.

The checks compare an independent binomial-series expansion, the parameter recurrence, and the logarithmic Newton recurrence through block index six. The table uses ordinary coefficients, with factorial conversion shown separately.

Plate IV.10 / BELL PARTITIONS

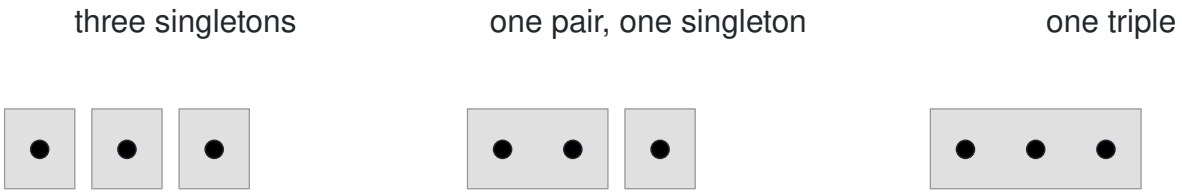
The reciprocal order is a polynomial

Chapter IV: Theorem 7.2, Corollary 7.7.

Let r vary instead of fixing the reciprocal power. Since

$$\Delta_N(w)^{-r} = \exp\left(r \sum_{m \geq 1} \kappa_{N,m} \frac{w^m}{m} \right),$$

the coefficient of w^n is a polynomial of degree n in r. Bell polynomials collect the possible ways logarithmic terms contribute total degree n.



$$r^3\kappa_1^3 \qquad 3r^2\kappa_1\kappa_2 \qquad 2r\kappa_3$$

$$15r^3 \qquad 30r^2 \qquad 16r$$

$$P_{2,3}(r) = \tfrac{6!}{3!}\, B_3(r\kappa_1, r\kappa_2, 2r\kappa_3)$$

For $N = 2$, the first three logarithmic coefficients are

$$\kappa_1 = \frac{1}{2}, \qquad \kappa_2 = \frac{1}{6}, \qquad \kappa_3 = \frac{1}{15}.$$

The third Bell polynomial is $B_3(x_1, x_2, x_3) = x_1^3 + 3x_1x_2 + x_3$. Substituting $x_1 = r\kappa_1$, $x_2 = r\kappa_2$, $x_3 = 2r\kappa_3$ and multiplying by $6!/3!$ gives

$$P_{2,3}(r) = 15r^3 + 30r^2 + 16r.$$

At $r = 1$ this is 61. At $r = 2$ it is 272. At $r = 0$ it vanishes, as every positive-index reciprocal coefficient must.

The leading term has a simple source

Only n copies of the linear logarithmic term can contribute $r^n w^n$. Therefore

$$[r^n]P_{N,n}(r) = \frac{(Nn)!}{n!(N!)^n}.$$

> **Read this as:** The polynomial parameter records how many logarithmic pieces are combined. It is not a numerical interpolation in the reciprocal order.

The partition plate is a finite combinatorial expansion. The exact tests verify the leading coefficient and parameter polynomial through block index six for five orders.

Plate IV.11 / NEWTON AND ADDITION

Two recurrences reach the same answer

Chapter IV: Theorem 7.3, Theorem 7.4, Theorem 7.6.

One recurrence uses the denominator coefficients. Another uses its logarithm. With $b_n(r) = P_{N,n}(r)/(Nn)!$, the shorter logarithmic form is

$$nb_n(r) = r\sum_{k=1}^{n} \kappa_{N,k} b_{n-k}(r), \qquad b_0 = 1.$$

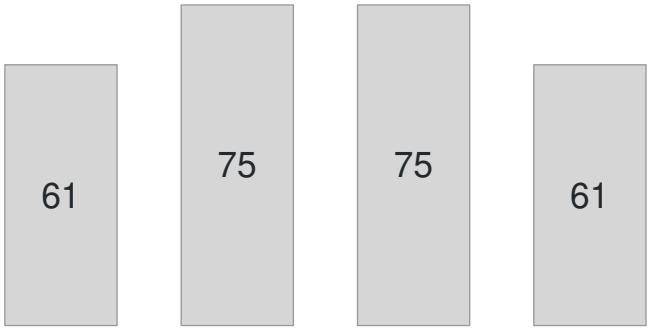
The negative superdiagonal in the Newton determinant is essential.

$n!b_n(r)$ for $n = 3$

$\frac{r}{2}$	-1	0
$\frac{r}{6}$	$\frac{r}{2}$	-2
$\frac{r}{15}$	$\frac{r}{6}$	$\frac{r}{2}$

$\det /6 = P_{2,3}(r)/720$

$P_{2,3}(2) = 61 + 75 + 75 + 61 = 272$



block weights: $1, 15, 15, 1$

The additive parameter law has block weights

Using the same logarithm, $D_N^{-r} D_N^{-s} = D_N^{-(r+s)}$. Thus

$$P_{N,n}(r+s) = \sum_{q=0}^{n} \binom{Nn}{Nq} P_{N,q}(r) P_{N,n-q}(s).$$

For $N = 2, n = 3, r = s = 1$, the four weights are $1, 15, 15, 1$. The two middle terms are $15 \cdot 1 \cdot 5$ and $15 \cdot 5 \cdot 1$. Together with the endpoints, they give 272. Using $\binom{n}{q}$ instead would be incorrect for these coefficients. That ordinary binomial law belongs to the separately normalized sequence $n!P_{N,n}/(Nn)!$.

> **Read this as:** Agreement of independent recurrences is a useful check precisely because their bookkeeping is different. Neither permits silently changing the factorial convention.

The checker compares both recurrence routes and the determinant, then checks the parameter-addition identity symbolically. A deliberately incorrect ordinary-binomial substitution is rejected.

Plate IV.12 / LOGARITHMIC DIFFERENTIATION

Compress the ODE into a tangent

Chapter IV: Proposition 8.2, Corollary 8.4, Theorem 8.5.

The transfer tangent is $T_N = -D_N'/D_N$. It compresses the linear differential equation into a nonlinear scalar identity. This division is allowed only on the zero-free set of the denominator.

LINEAR EQUATION	LOGARITHMIC DERIVATIVE
$D_2'' = -D_2$	$T_2' = 1 + T_2^2$
$D_3''' = -D_3$	$T_3'' - 3T_3T_3' + T_3^3 = 1$

$$T_N = -D_N'/D_N \quad \text{only where } D_N \neq 0$$

$$T_4(z) = \frac{z^3}{3!} + 34\frac{z^7}{7!} + \cdots$$

The coefficients retain the reduced logarithmic information:

$$T_N(z) = N\sum_{m\geq 1} \kappa_{N,m} z^{Nm-1}, \qquad T_N(n) = N(Nn+N-1)!\kappa_{N,n+1}.$$

For order four, $\kappa_{4,1} = 1/24$ and $\kappa_{4,2} = 17/10080$. These give $T_4(0) = 1$ and $T_4(1) = 34$, explaining the last line of the plate.

A nonlinear equation with a precise origin

If $D = e^g$ locally, then $g' = -T$. Differentiating e^g three times gives $D'''/D = -T'' + 3TT' - T^3$. Setting $D'''/D = -1$ yields the cubic Riccati identity shown above. No additional nonlinear model has been assumed. The reciprocal family also satisfies $(D_N^{-r})' = rT_N D_N^{-r}$ on a fixed-log domain. Behavior at zeros of D_N depends on the reciprocal order and the branch conventions stated on Plate IV.13.

> **Read this as:** The nonlinear tangent equation is exact where division is legal. It is not a global regular differential equation across the zeros.

The tests verify the tangent coefficient recurrence, the exact cubic ODE, and the symbolic Riccati reduction after clearing denominators. The order-one tangent is the separate constant function $T_1 = 1$.

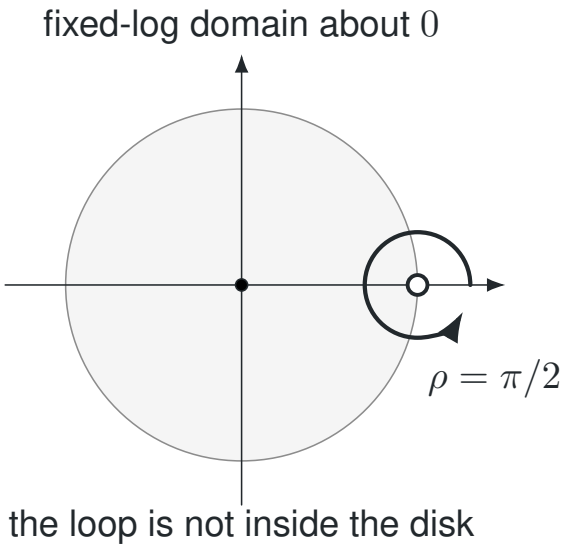
Plate IV.13 / BRANCH BOOKKEEPING

A complex power needs a domain

Chapter IV: Section 2, Definition 7.1, Corollary 8.4.

A formal unit series has a formal logarithm at the origin. To interpret that germ analytically, choose a simply connected zero-free region containing 0 and normalize the logarithm to vanish there.

$$D_N(z)^{-r} = \exp(-rL(z)), \qquad e^{L(z)} = D_N(z), \quad L(0) = 0.$$



$$D(z) \sim D'(\rho)(z - \rho)$$

$$\log D \longmapsto \log D + 2\pi i$$

$$D^{-r} \longmapsto e^{-2\pi i r} D^{-r}$$

$$r = 1/2 : \ -1 \qquad r \in \mathbb{Z} : \ 1$$

For $D_2 = \cos z$, the open disk $|z| < \pi/2$ is an admissible domain. A small loop around the simple zero $\rho = \pi/2$ is not contained in that disk. Analytic continuation around the zero changes the local logarithm by $2\pi i$, and the power by $e^{-2\pi i r}$.

Three different kinds of statement

A *formal* identity compares coefficients at the origin. A *holomorphic* identity for complex powers uses a common specified logarithm. A positive *integer* reciprocal power is globally meromorphic and needs no branch choice. These scopes cannot be exchanged without a further argument. For example, $D^{-r}D^{-s} = D^{-(r+s)}$ is immediate using one fixed logarithm. Combining values built from unrelated branches can introduce additional phase factors.

> **Read this as:** The branch choice is mathematical data. It is not a plotting convention that can be removed by drawing a cleaner curve.

The domain picture is schematic. The phase change comes from the local simple-zero model and winding number, not from a sampled loop. Later nearest-pole asymptotics retain positive integral reciprocal order.

Plate IV.14 / NORMALIZED SECTIONS

Three similar reciprocals differ

Chapter IV: Definition 9.1, Theorem 9.2, Example 9.11.

The signed denominator and the normalized unsigned sections produce different reciprocal families. In order two,

$$D_2^{-1} = \sec z, \qquad C_{2,0} = \operatorname{sech} z, \qquad C_{2,1} = z/\sinh z.$$

Their first block-EGF coefficients make the difference visible.

function	1	$z^2/2!$	$z^4/4!$	$z^6/6!$
$\sec z$	1	1	5	61
$\operatorname{sech} z$	1	-1	5	-61
$z/\sinh z$	1	$-\frac{1}{3}$	$\frac{7}{15}$	$-\frac{31}{21}$

$$E_{2,1}(z) = \sinh z = z\Phi_{2,1}(z^2), \qquad C_{2,1}(0) = 1$$

Remove the forced zero before taking a reciprocal

The section $E_{N,j}$ begins with $z^j/j!$. Define

$$E_{N,j}(z) = \frac{z^j}{j!}\Phi_{N,j}(z^N), \qquad \Phi_{N,j}(0) = 1.$$

Then $C_{N,j}(z) = 1/\Phi_{N,j}(z^N)$ has value 1 at the origin. For $j > 0$, the notation $(z^j/j!)/E_{N,j}(z)$ means this removable continuation. It does not mean evaluating an undefined $0/0$ expression. For $N = 2, j = 1$, the first nontrivial coefficient is $-1/3$ in front of $z^2/2!$, equivalently $-z^2/6$. Dropping the numerator z would instead produce the pole of $1/\sinh z$.

> **Read this as:** A minus sign in the denominator series and removal of an initial zero are different operations. Their reciprocal coefficients should not be interchanged.

All entries in the table are exact rationals computed from independent triangular inverses. The table explicitly uses block-EGF columns. Ordinary coefficients are obtained by dividing by the displayed factorials.

Plate IV.15 / APPELL POLYNOMIALS

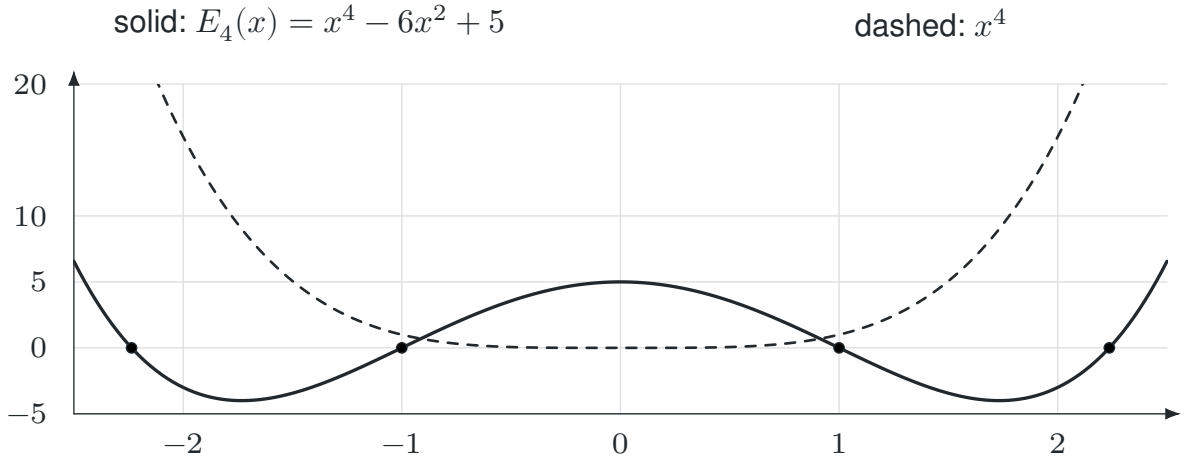
An infinite symbol acts finitely

Chapter IV: Theorem 9.6, Theorem 9.8, Proposition 9.9.

For $C_{2,0}(z) = \operatorname{sech} z$, define $E_m(x)$ by $\operatorname{sech} z\, e^{xz} = \sum E_m(x) z^m/m!$. The formal differential-operator formula is

$$E_m(x) = \operatorname{sech}(\partial_x) x^m.$$

Although the symbol is an infinite series, derivatives above degree m vanish.



$$\cosh(\partial_x) E_4 = E_4 + \tfrac{1}{2} E_4'' + \tfrac{1}{24} E_4^{(4)} = x^4$$

For $m = 4$, only three terms survive:

$$\left(1 - \frac{\partial_x^2}{2!} + 5\frac{\partial_x^4}{4!}\right) x^4 = x^4 - 6x^2 + 5.$$

The four real roots of this polynomial are $\pm 1, \pm\sqrt{5}$. They are not the zeros of the original analytic denominator. The plot distinguishes the input polynomial and its transformed output.

An inverse operation, not a truncation error

Applying $\cosh(\partial_x)$ returns x^4 exactly. The general inverse uses the reciprocal formal symbol. The lowering and translation identities are

$$E_m' = m E_{m-1}, \qquad E_m(x+y) = \sum_{q=0}^{m} \binom{m}{q} E_q(x) y^{m-q}.$$

Both follow from the generating function. No operator-series convergence assumption on an infinite-dimensional function space is needed for this polynomial calculation.

> **Read this as:** Formal substitution into the derivative is safe here because every polynomial sees only finitely many derivatives. That is a domain statement.

The polynomial, its inverse image, and the Appell identities are checked symbolically through degree eight in four normalized-section families. The displayed curves are samples of exact polynomials.

Plate IV.16 / THETA COUPLING

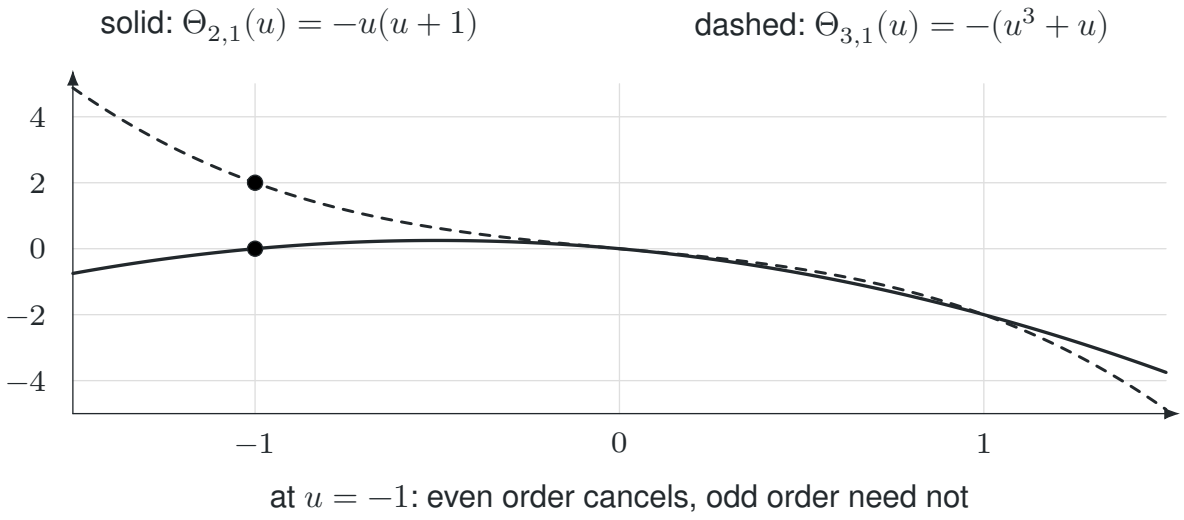
Scale and order enter differently

Chapter IV: Theorem 10.2, Proposition 10.3, Proposition 10.4.

Let $H_N(z) = E_{N,0}(z)$. The theta family uses the same parameter u twice, but in two different roles:

$$H_N(uz)^{-1} H_N(z)^{-u} = \sum_{n \geq 0} \Theta_{N,n}(u) \frac{z^{Nn}}{(Nn)!}.$$

Argument scaling contributes u^{Nm} at logarithmic block m. Reciprocal order contributes u.



Their combined logarithm has coefficient

$$(-1)^m \kappa_{N,m} \frac{u^{Nm} + u}{m}.$$

At the first block this gives $\Theta_{N,1}(u) = -(u^N + u)$. Every positive-index theta polynomial vanishes at $u = 0$. For even N, $H_N(-z) = H_N(z)$, so the defining product is also 1 at $u = -1$.

The odd-order boundary is visible

For $N = 3$, $\Theta_{3,1}(-1) = 2$, not zero. The factor $u + 1$ is not a uniform law across orders. In contrast,

$$\Theta_{2,2}(u) = u(u+1)(5u^2 + u + 2).$$

The two guaranteed factors follow from the whole generating function, not from guessing the first few polynomial roots.

> **Read this as:** A common parameter does not make argument scaling and reciprocal order interchangeable. Their powers must be tracked separately.

The exact checker compares theta convolution with its logarithmic Newton recurrence through block index four. The odd-order factor claim is explicitly rejected as an invalid control.

Plate IV.17 / DOMINANT EXPONENTS

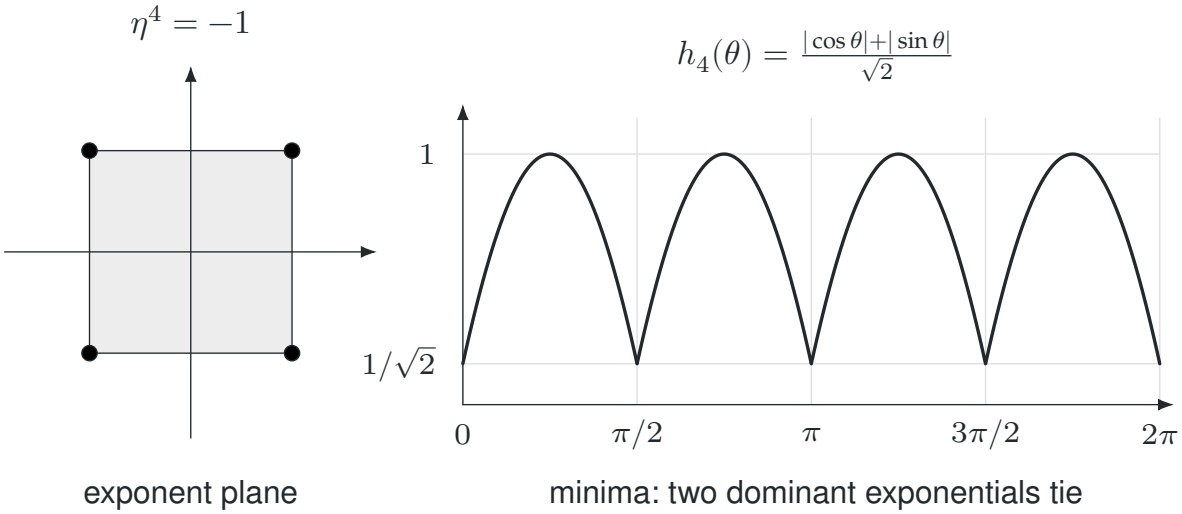
Growth comes from a polygon

Chapter IV: Definition 11.3, Theorem 11.4.

Write the signed denominator as a finite exponential sum:

$$D_N(z) = \frac{1}{N} \sum_{\eta^N=-1} e^{\eta z}.$$

For $z = re^{i\theta}$, the size of one exponential is $e^{r\Re(\eta e^{i\theta})}$. The maximal projection determines the directional exponential growth.



The indicator is

$$h_N(\theta) = \max_{0\le k<N} \cos\left(\theta + \frac{(2k+1)\pi}{N}\right).$$

Away from a tie direction, one exponential dominates all the others by a uniform positive projection gap on compact angular sets. For sufficiently large radius, smaller terms cannot cancel it. Those sectors are eventually zero-free.

Why zeros remain possible on a growing ray

At a tie, two adjacent exponents have the same real projection. Their complex phases can be opposite. Their sum can therefore vanish even though each term is exponentially large. The minima in the order-four indicator plot occur precisely at the tie directions $\theta = k\pi/2$. The indicator uses a limsup in radius. It does not assert that $r^{-1}\log|D_N(re^{i\theta})|$ has an ordinary limit through the zeros on a tie ray.

> **Read this as:** The polygon predicts where large cancellation is possible. It does not, by itself, enumerate zeros or prove their simplicity.

The order-four polygon and support-function formula are explicit. The curve is a sampled display of that formula. Eventual localization and uniqueness require the separate zero-analysis arguments.

Plate IV.18 / THE CUBIC MODEL

Two large terms nearly cancel

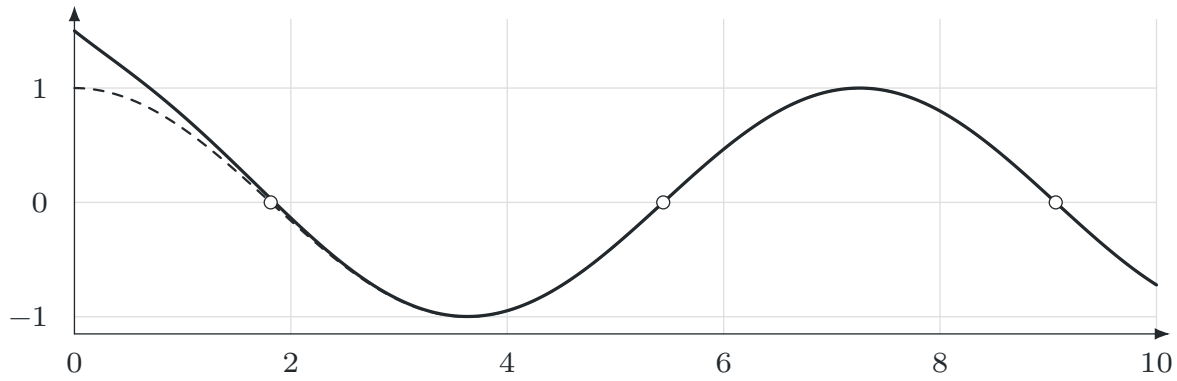
Chapter IV: Lemma 11.5, Theorem 11.7, Corollary 11.9.

For real t, order three has the exact formula

$$D_3(t) = \frac{e^{-t} + 2e^{t/2}\cos(\sqrt{3}t/2)}{3}.$$

Divide by the nonzero factor $2e^{t/2}/3$. The resulting equation separates the dominant cosine from the small third exponential.

$$\tfrac{3}{2}e^{-t/2}D_3(t) = \cos(\sqrt{3}t/2) + \tfrac{1}{2}e^{-3t/2}$$



solid: full expression; dashed: two-term model; circles: model zeros

Ignoring the last term gives model zeros

$$t_m = \frac{2\pi}{\sqrt{3}}(m + 1/2).$$

At t_m, the full normalized expression equals $\frac{1}{2}e^{-3t_m/2} > 0$, so t_m is not an exact zero. Linearizing the cosine at that point predicts

$$\delta_m^{(1)} = \frac{(-1)^m}{\sqrt{3}}e^{-3t_m/2}.$$

The alternating slope explains why the true zero is to the right of the first model zero, then to the left of the second, and so on in the eventual regime.

Symmetry needs uniqueness

Real coefficients alone allow nonreal conjugate pairs. The manuscript first proves that a near-model disk contains exactly one zero. Conjugation preserves that disk, so the unique zero must equal its conjugate. This additional uniqueness step, not the symmetry alone, places it exactly on the ray.

> **Read this as:** The correction is caused by a real, nonzero remainder. A good cosine approximation does not make its model zeros exact.

The curves are sampled analytic expressions. The next page independently certifies four narrow real-zero intervals rather than trusting the plotted crossing or the floating-point root finder.

Plate IV.19 / EXACT RATIONAL BRACKETS

A root estimate becomes a certificate

Chapter IV: Theorem 11.7, Theorem 11.8, Corollary 11.9.

The root finder is used only to suggest intervals. Each displayed interval is accepted by exact rational arithmetic: opposite endpoint signs prove existence, and an interval bound for D_3' excludes zero throughout the interval, proving uniqueness inside it.

	certified lower endpoint	$\tau_m - t_m$	$\delta_m/\delta_m^{(1)} - 1$
$m = 0$	1.84981279919014347629	$+3.60134 \times 10^{-2}$	-5.243×10^{-2}
$m = 1$	$5.441233355023663355 40$	-1.64738×10^{-4}	$+2.471 \times 10^{-4}$
$m = 2$	$9.068997534871577658 05$	$+7.13700 \times 10^{-7}$	-1.071×10^{-6}
$m = 3$	$12.696595546546757294 40$	-3.09277×10^{-9}	$+4.639 \times 10^{-9}$

Each upper endpoint is the displayed lower endpoint plus 10^{-20}.

Root intervals: exact rational certificates. Displacement columns: numerical diagnostics.

How the certificate works

Evaluate the Taylor polynomial

$$S_{90}(t) = \sum_{n=0}^{90} \frac{(-1)^n t^{3n}}{(3n)!}$$

at the two rational endpoints. From the next term onward, magnitudes decrease, because their successive ratio is

$$\frac{t^3}{(3n+1)(3n+2)(3n+3)} < 1$$

there. The alternating-series remainder is bounded by the first omitted term. A separate outward interval evaluation of the differentiated polynomial, plus its tail bound, certifies the derivative sign. The first bracket is

$$1.84981279919014347629 < \tau_0 < 1.84981279919014347630.$$

Its displacement differs from the leading asymptotic prediction by about five percent. The later examples have much smaller relative discrepancies, as the plate records.

> **Read this as:** A small residual is not a root certificate. Endpoint signs plus a derivative bound give a checkable local existence-and-uniqueness statement.

Four certified intervals do not establish that no other real or complex zeros exist. The displacement columns are high-precision diagnostics, not a finite verification of a global big-O theorem. All rational enclosures are included in the checks file.

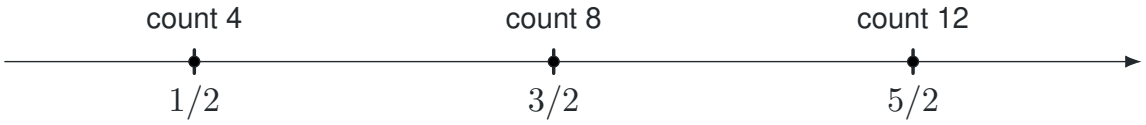
Plate IV.20 / EXACT FOURTH-ORDER STRINGS

Counting changes under the power map

Chapter IV: Example 11.12, Theorem 11.14.

Order four gives a case with no asymptotic displacement at all. In normalized coordinates the positive radial locations are $u = m + 1/2$, while the reduced-zero locations are $v = (m + 1/2)^4$.

$$u = |z|/(\sqrt{2}\pi)$$



$$v = |w|/(4\pi^4)$$



$$n_{D_4}(\sqrt{2}\pi u) = 4\lfloor u + 1/2 \rfloor, \quad n_{\Delta_4}(4\pi^4 v) = \lfloor v^{1/4} + 1/2 \rfloor$$

The two rows describe the same zero orbits. The upper row counts four zeros each time a new radius is reached. The lower row counts one reduced zero each time its fourth-power location is reached. Both formulas include zeros on the boundary of the disk.

Separate radius conversion from orbit multiplicity

In unscaled variables the general relation is

$$n_{D_N}(R^{1/N}) = N n_{\Delta_N}(R).$$

There are two operations here: replace the radius by its Nth root and divide the count by N. Forgetting either operation changes the leading constant. For this family the asymptotics are

$$n_{D_N}(R) = \frac{N \sin(\pi/N)}{\pi} R + O(1), \qquad n_{\Delta_N}(R) = \frac{\sin(\pi/N)}{\pi} R^{1/N} + O(1).$$

The different powers agree with the entire-function orders 1 and $1/N$.

> **Read this as:** The reduced function has fewer repeated zeros and a different radius scale. Its counting law is not obtained by dividing a same-radius count alone.

The exact order-four staircase formulas are tested at rational radii including jump points. The large-zero theorem for general N permits finitely many exceptions. These exact order-four identities do not justify discarding that boundary elsewhere.

Plate IV.21 / CANONICAL PRODUCT

A product and an operator share zeros

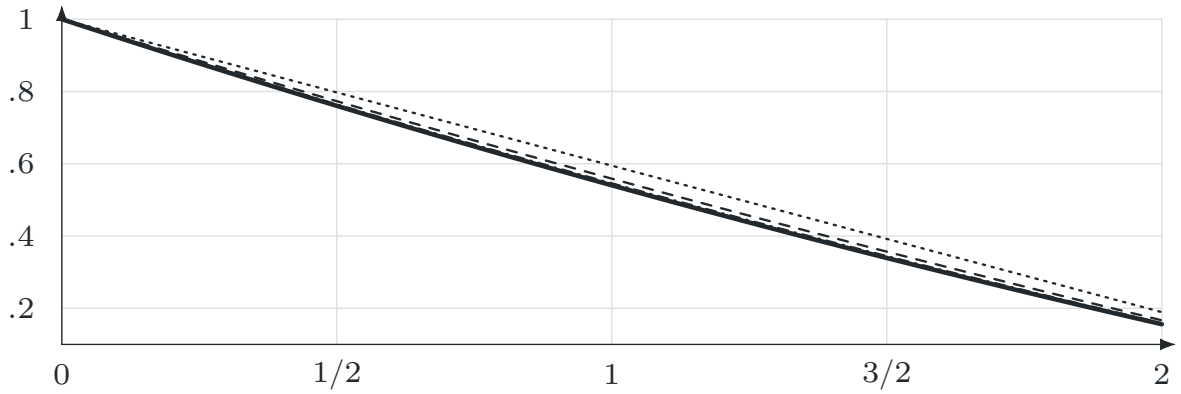
Chapter IV: Theorem 12.1, Theorem 12.2.

For $N \geq 2$, the reduced entire function has order $1/N < 1$. Its multiplicity-sensitive genus-zero product is

$$\Delta_N(w) = \prod_{\lambda \in \Lambda_N} (1 - w/\lambda).$$

Putting $K_N e_\lambda = \lambda^{-1} e_\lambda$ on $\ell^2(\Lambda_N)$ gives the corresponding diagonal Fredholm determinant. Repeated zeros require repeated diagonal entries.

$$\Delta_2(w) = \cos\sqrt{w}, \qquad P_L(w) = \prod_{m=0}^{L-1}(1 - w/\lambda_m)$$



thick: entire function; dotted $L = 1$; dashed $L = 3$; dash-dot $L = 10$

Finite compression is an approximation

For $N = 2$, $\lambda_m = \pi^2(m + 1/2)^2$. The finite determinant P_L is exactly the displayed finite product, but it is not exactly Δ_2. For $0 \leq w \leq 1$ and $L \geq 1$, an elementary tail estimate gives

$$0 \leq 1 - \frac{\Delta_2(w)}{P_L(w)} \leq w \sum_{m \geq L} \lambda_m^{-1} \leq \frac{1}{9(L - 1/2)}.$$

The last bound uses $\pi > 3$ and integral comparison. Thus truncation has an explicit error mechanism.

> **Read this as:** The infinite product is justified by the analytic theorem. Finite determinants provide controlled approximations, not substitutes for that theorem.

The exact finite determinant test includes a repeated eigenvalue and rejects deduplication. The curves use known cosine zeros. Order one is excluded: $\Delta_1(w) = e^{-w}$ has no zeros, so its empty product would be 1, not the function.

Plate IV.22 / ZERO POWER SUMS

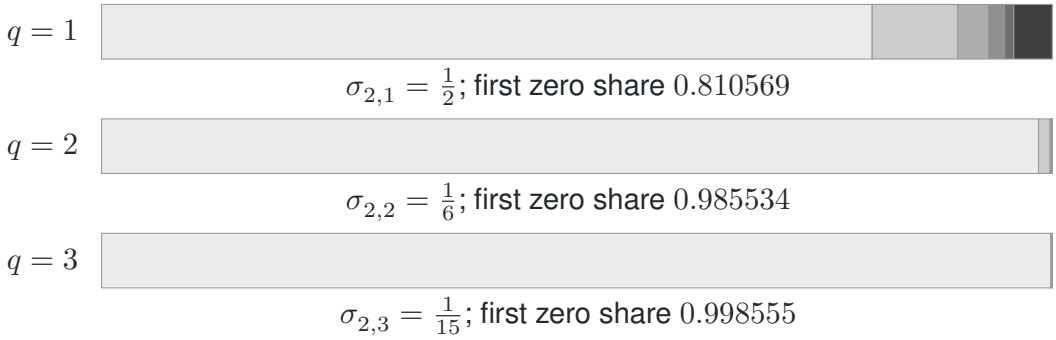
Finite coefficients give infinite moments

Chapter IV: Theorem 12.6, Corollary 12.7, Corollary 12.9, Corollary 12.10.

Taking the logarithm of the canonical product identifies

$$\kappa_{N,q} = \sigma_{N,q} := \sum_{\lambda \in \Lambda_N} \lambda^{-q} = \mathrm{Tr}(K_N^q), \qquad q \geq 1.$$

The same exact numbers can be computed from finitely many denominator coefficients by Newton's recurrence. No numerical zero list is needed for that calculation.



five finite contributions plus the remaining tail; higher moments emphasize the nearest zero

For order two,

$$\sigma_{2,1} = \frac{1}{2}, \quad \sigma_{2,2} = \frac{1}{6}, \quad \sigma_{2,3} = \frac{1}{15}, \quad \sigma_{2,4} = \frac{17}{630}.$$

The plate shows how successive inverse powers emphasize the nearest zero. Its shaded pieces are contributions from the first five positive reduced zeros, followed by the remaining tail. The total in each row is the exact rational moment.

The tangent is a sum of poles

Logarithmic differentiation also gives

$$T_N(z) = N z^{N-1} \sum_{\lambda \in \Lambda_N} \frac{1}{\lambda - z^N}$$

away from poles. Summability of $|\lambda|^{-1}$ permits the locally uniform partial-fraction expansion without added regularizing terms.

> **Read this as:** A global analytic product theorem connects the infinite zero sum to a finite algebraic recurrence. The recurrence alone does not prove the product theorem.

Displayed moments are checked against formal logarithm coefficients and exact order-two/order-four zeta values. Bar widths are numerical shares. General power sums use λ^{-q}, not $|\lambda|^{-q}$ when exceptional nonpositive zeros are present.

Plate IV.23 / SINGULARITY ANALYSIS

Nearest poles set coefficient growth

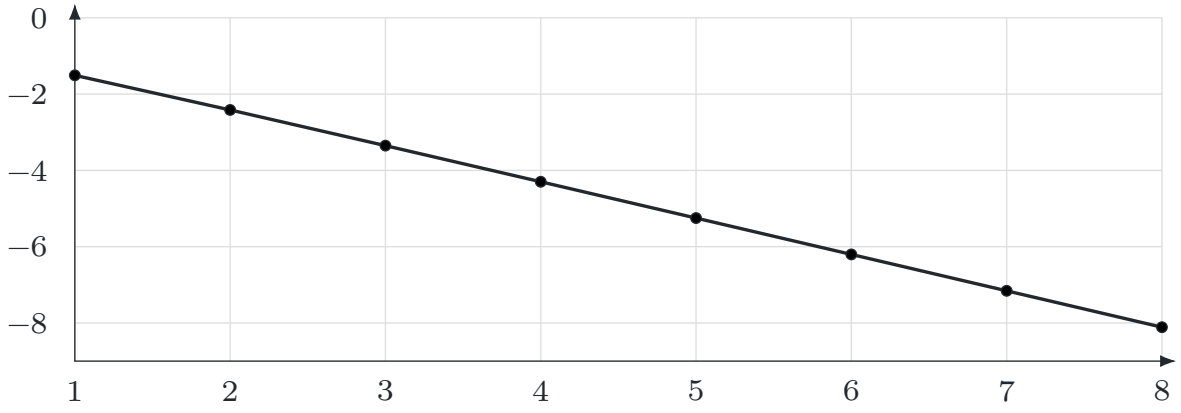
Chapter IV: Theorem 13.1, Corollary 13.2, Corollary 13.3, Corollary 13.4.

For $1/\cos z$, the nearest poles are $\rho = \pm\pi/2$. They contribute equally to even-degree coefficients. Including both poles gives

$$P_{2,n}(1) \sim 2(2n)! \left(\frac{2}{\pi}\right)^{2n+1}.$$

The factor two is orbit multiplicity, not an adjustable fitted constant.

$$\log_{10}(1 - R_n), \quad R_n = \frac{P_{2,n}(1)}{2(2n)!(2/\pi)^{2n+1}}$$



Each ratio is enclosed with a rational Machin-formula bound for π.

A finite quantitative check

For $n = 1, \dots, 8$, the exact integer coefficients and a rational enclosure for π bound the ratio R_n strictly between zero and one. In this tested range, $1 - R_n < 3^{-(2n+1)}$. The plot displays the logarithm of the enclosed error, not an independently asserted all-n bound.

Higher reciprocal powers need more Laurent data

For positive integral order r, a simple denominator zero becomes a pole of order r. The highest Laurent term contributes a factor $\binom{m+r-1}{r-1}$, but lower Laurent terms still contribute on the same exponential scale. Subtracting *all* principal parts is what leaves a holomorphic remainder controlled by the next zero radius.

> **Read this as:** Nearest-zero geometry explains factorial-times-exponential growth. Multiplicity, reciprocal order, and lower principal parts determine its precise normalization.

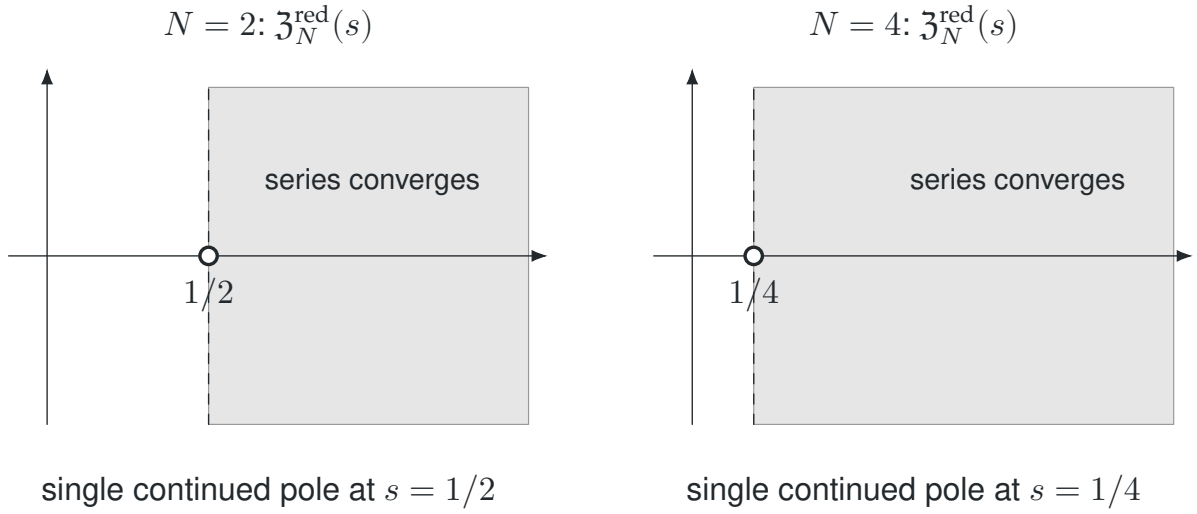
The ratio certificates use a rational Machin-formula enclosure for π. The inherited general theorem retains simple nearest zeros, a gap to the next modulus, and positive integral reciprocal order. The plot does not remove those hypotheses.

Plate IV.24 / ZERO ZETA FUNCTIONS

Convergence is not continuation

Chapter IV: Theorem 14.2, Theorem 14.3, Corollary 14.4.

Reduced zeros grow like m^N. Their absolute zero-zeta series therefore has half-plane of absolute convergence $\Re s > 1/N$. The absolute zero zeta in the original variable instead requires $\Re s > 1$.



For $N = 2$ and $N = 4$, the lattices are exact:

$$\mathfrak{Z}_2^{\mathrm{red}}(s) = \pi^{-2s}\zeta(2s, 1/2), \qquad \mathfrak{Z}_4^{\mathrm{red}}(s) = (\sqrt{2}\pi)^{-4s}\zeta(4s, 1/2).$$

Each identity first holds where its defining series converges. The right side supplies its meromorphic continuation. There is a single pole, at $1/2$ or $1/4$, with residue $1/(2\pi)$ or $1/(4\sqrt{2}\pi)$ respectively.

The factor N matters twice

For every order in this chapter,

$$\mathfrak{Z}_N^{\mathrm{abs}}(s) = N\mathfrak{Z}_N^{\mathrm{red}}(s/N).$$

For general N, exponentially accurate zero locations leave an entire correction to the Hurwitz model. A finite exceptional zero set also contributes entire terms. This reasoning establishes continuation, not a zeta functional equation.

> **Read this as:** An analytic continuation can exist at points where the original infinite sum diverges. It must not be evaluated there by naively summing that series.

The shaded regions show convergence, and the open circles mark poles of the continued functions. They are not zeros of the zeta functions. Branch-dependent orbit zetas and derivative-weighted zetas retain their separately stated conventions.

Plate IV.25 / COEFFICIENT AND ZERO WORKED MODEL

One closed calculation: the number 69

Chapter IV: Corollary 7.7, Example 11.12, Theorem 12.6.

Use $N = 4$ throughout this page. Its denominator, its exact zero lattice, and the reciprocal recurrences provide two routes to the same coefficient. All symbols in the plate refer to this one order.

denominator	$1 - w/24 + w^2/40320 + \cdots$
negative logarithm	$w/24 + (17/20160)w^2 + \cdots$
zero moments	$\sigma_1 = 1/24, \quad \sigma_2 = 17/10080$
reciprocal	$1 + w/24 + 69w^2/40320 + \cdots$
tangent	$z^3/3! + 34z^7/7! + \cdots$

$$69 = 8!\,\tfrac{\sigma_1^2+\sigma_2}{2}, \qquad 34 = 4 \cdot 7!\,\sigma_2$$

Route one: invert the denominator

The w^2 coefficient of Δ_4^{-1} is

$$\frac{1}{24^2} - \frac{1}{8!} = \frac{69}{8!}.$$

Thus $P_{4,2}(1) = 69$. Keeping r variable gives $P_{4,2}(r) = 35r^2 + 34r$.

Route two: use the zero moments

The exact reduced zeros are $4\pi^4(m + 1/2)^4$. Their first two power sums are $\sigma_1 = 1/24$ and $\sigma_2 = 17/10080$. The Bell/Newton identity gives the same reciprocal coefficient $(\sigma_1^2 + \sigma_2)/2$. The tangent independently reads off the second moment as $4 \cdot 7!\sigma_2 = 34$. This links coefficients, geometry, and derivatives without a fitted parameter or an inferred zero list.

> **Read this as:** The content is the agreement of precisely normalized descriptions. A matching small example illustrates the general bridges. It is not a proof of all their analytic hypotheses.

Every rational value and polynomial identity on this page is checked exactly. The infinite zero-sum interpretation uses the inherited canonical-product theorem and the exact order-four zero formula.

Plate IV.26 / FORMAL AND COMPUTATIONAL BOUNDARIES

What is evidence, and for what?

Chapter IV: Section 15, Section 16.

These twenty-six worked plates connect explicit mathematical models with the main text and its formal companion. Each evidence record states the scope of the model, computation, or formal result.

Evidence layer	What it supports
Formal record	The sources, theorem map, dependency pins, saved axiom reports, and mathematical review retain their own scope. The 68-result contract and its replay using pinned sources are described in the Lean guide.
Exact and symbolic suite	The recorded isolated execution passed all 147 finite, symbolic, and enclosure checks and rejected five deliberately false controls. The unchanged checker and models were run with SymPy 1.14.0 and mpmath 1.3.0.
Certified zero brackets	Four rational intervals of width 10^{-20} establish local existence and uniqueness only. The full suite reproduced all four rational brackets, opposite endpoint signs, and derivative exclusion throughout each interval. These are local certificates, not a global zero census.
Numerical visualization	Sampled analytic curves, finite products, moment shares, and displacement diagnostics have the roles stated on their pages. They are not exact global zero censuses.
Figure provenance	Regeneration from the models reproduced all twenty-four editable TikZ files. The mathematical labels and objects are preserved.
Editorial preservation	The original worked formulas, model code, checker, and vector figure sources are preserved. Locators refer to the main text.

Boundaries that remain in force

The general family has eventual positivity and simplicity of its reduced zeros, not an assertion that every reduced zero is positive and simple. This chapter leaves universal simplicity, positivity of every reduced zero, full displacement expansions, and a zeta functional equation open. The exact order-two and order-four models do not settle those questions for every order. Imported standard fact 11.1 stays outside the 68-result contract. Fixed logarithms, zero-free domains, multiplicities, and the hypotheses of nearest-pole asymptotics remain explicit. No historical priority is inferred from successful computation or formal verification.

Replay the worked layer

The model checker is `checks/verify_examples.py`. The plate generator is `source/generate_graphics.py`. They require the listed Python dependencies. Run those mathematical tools in a temporary copy to preserve the delivered evidence.

> **Read this as:** A proof, a finite exact test, a root enclosure, and a sampled curve answer different questions. Keeping their scopes explicit makes the pictures useful mathematical evidence.

Exact certificates and formal verification records accompany the source. Each retains its stated mathematical scope.

How to use this volume

Twenty-six worked pages and twenty-four editable vector figures connect finite coefficient calculations with differential equations, zero orbits, canonical products, and spectral moments. Exact identities and rational certificates are distinguished from sampled analytic displays throughout.

A route through the examples

Plates IV.1–IV.26 identify worked pages independently of PDF pagination. Each page names the corresponding statements in Chapter IV. The formal companion records the separate formal contract.

IV.1 A coefficient can become zero data
IV.2 Two variables, several normalizations
IV.3 A root filter is an exact vote
IV.4 Products mix the residue lanes
IV.5 One signed edge determines the cycle
IV.6 Addition is quotient multiplication
IV.7 Why a hypergeometric name appears
IV.8 Four zeros become one orbit zero
IV.9 A reciprocal is a triangular solve
IV.10 The reciprocal order is a polynomial
IV.11 Two recurrences reach the same answer
IV.12 Compress the ODE into a tangent
IV.13 A complex power needs a domain
IV.14 Three similar reciprocals differ
IV.15 An infinite symbol acts finitely
IV.16 Scale and order enter differently
IV.17 Growth comes from a polygon
IV.18 Two large terms nearly cancel
IV.19 A root estimate becomes a certificate
IV.20 Counting changes under the power map
IV.21 A product and an operator share zeros
IV.22 Finite coefficients give infinite moments
IV.23 Nearest poles set coefficient growth
IV.24 Convergence is not continuation
IV.25 One closed calculation: the number 69
IV.26 What is evidence, and for what?

Three reading routes

Algebra first: Plates IV.3–IV.16. Zeros first: Plates IV.17–IV.24. Full calculation: Plate IV.25. Evidence: Plate IV.26.

Keep the two variables distinct

Use z for differentiation and rotational geometry. Use $w = z^N$ for reciprocal recurrences, canonical products, and orbit-zero sums. An ordinary coefficient in w becomes a block exponential coefficient in z only after multiplication by $(Nn)!$.

Exact models and analytic conclusions

The order-four factorization gives its complete zero set. The four rational cubic brackets give local existence and uniqueness in their stated intervals. General zero localization, infinite products, and zeta continuation require the traditional paper's analytic hypotheses and arguments.

PLATE IV.1 / READER'S GUIDE

A coefficient can become zero data

Chapter IV: Section 2, Section 7, Section 11, Section 12.

This guide follows a concrete analytic family through three descriptions: selected Taylor coefficients, a finite differential system, and the zeros of an entire function. The formulas are linked, but the evidence needed for each description is different.

COEFFICIENT LANES

z^0	z^3	z^6	z^9
z^1	z^4	z^7	z^{10}
z^2	z^5	z^8	z^{11}

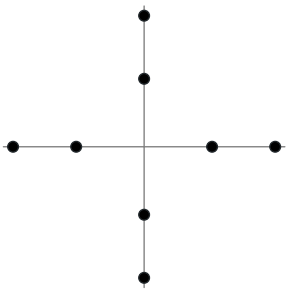
$$m \equiv j \pmod 3$$

RECIPROCAL COEFFICIENTS

1	0	0
$-\frac{1}{2}$	1	0
$\frac{1}{24}$	$-\frac{1}{2}$	1

$$1, \quad \tfrac{1}{2}, \quad \tfrac{5}{24}, \ldots$$

ZERO ORBITS



$$\{\rho, i\rho, -\rho, -i\rho\}$$

Keep three small models in view

For $N = 2$, the signed denominator is $D_2(z) = \cos z$. Its reciprocal is the familiar secant series. For $N = 3$, a third exponential perturbs the zeros of a dominant cosine. For $N = 4$, an exact factorization gives a complete zero picture:

$$D_4(z) = \cos(z/\sqrt{2})\cosh(z/\sqrt{2}).$$

The examples move between these orders deliberately. Order two makes the coefficient arithmetic transparent. Order three shows a genuine displacement. Order four makes orbit reduction completely explicit.

Three routes through the guide

Route	Pages and question
Algebra first	Plates IV.3–IV.16: selection, signs, reciprocals, and finite differential operators.
Zeros first	Plates IV.17–IV.24: cancellation and its consequences for coefficients.
One full calculation	Plate IV.25: derive the integer 69 from denominator coefficients and from zero moments.

Read this as: The pictures carry computable objects. They do not replace the global analytic arguments or extend the inherited Lean contract.

Exact identities and rational certificates are distinguished from sampled curves. All plates are reproducible vector drawings. The source package preserves the original proofread manuscript.

PLATE IV.2 / OBJECT ATLAS

Two variables, several normalizations

Chapter IV: Section 2, Definition 5.7, Definition 7.1, Definition 9.1.

The most important coordinate change is $w = z^N$. A term of degree n in w has degree Nn in z. Its coefficient does not change, but an exponential-generating-function label does.

$$\underbrace{b_{N,n}(r)\,w^n}_{\text{ordinary coefficient}} = \underbrace{P_{N,n}(r)\frac{z^{Nn}}{(Nn)!}}_{\text{block exponential coefficient}}, \qquad P_{N,n}(r) = (Nn)!\,b_{N,n}(r).$$

Object	Definition and role
$E_{N,j}(z)$	$\displaystyle\sum_{n\geq 0}\frac{z^{Nn+j}}{(Nn+j)!}$ selects one residue class of the exponential.
$D_{N,j}(z)$	The same series with $(-1)^n$. One full derivative cycle contributes a minus sign.
$D_N(z)$	$D_{N,0}(z)$, the denominator whose zeros control the signed reciprocal family.
$\Delta_N(w)$	$\displaystyle\sum_{n\geq 0}\frac{(-1)^n w^n}{(Nn)!}$. Thus $D_N(z) = \Delta_N(z^N)$.
$P_{N,n}(r)$	Block-EGF coefficients of D_N^{-r}. They are polynomials in r.
$\kappa_{N,m}$	$-\log \Delta_N(w) = \displaystyle\sum_{m\geq 1}\kappa_{N,m}w^m/m$.
$T_N(z)$	$-D_N'(z)/D_N(z)$, a function, not the coefficient $T_N(n)$.
$\Phi_{N,j}(w)$	$j!\displaystyle\sum_{n\geq 0}\frac{w^n}{(Nn+j)!}$ removes the forced zero of $E_{N,j}$.
$C_{N,j}(z)$	$1/\Phi_{N,j}(z^N)$. This is generally not $1/D_N(z)$.
Λ_N	Zeros of Δ_N, counted with multiplicity, one per rotational orbit of zeros of D_N.

A normalization test

The coefficient $5/24$ of w^2 in $\Delta_2(w)^{-1}$ is the same term as $5z^4/4!$. It is not $5w^2/2!$. Keeping the two factorials separate prevents errors in Bell formulas, convolution weights, and zero-moment reconstruction.

> **Read this as:** Use z for differentiation and rotational geometry. Use w for reciprocal recurrences, canonical products, and orbit-zero sums.

PLATE IV.3 / CYCLIC PROJECTION

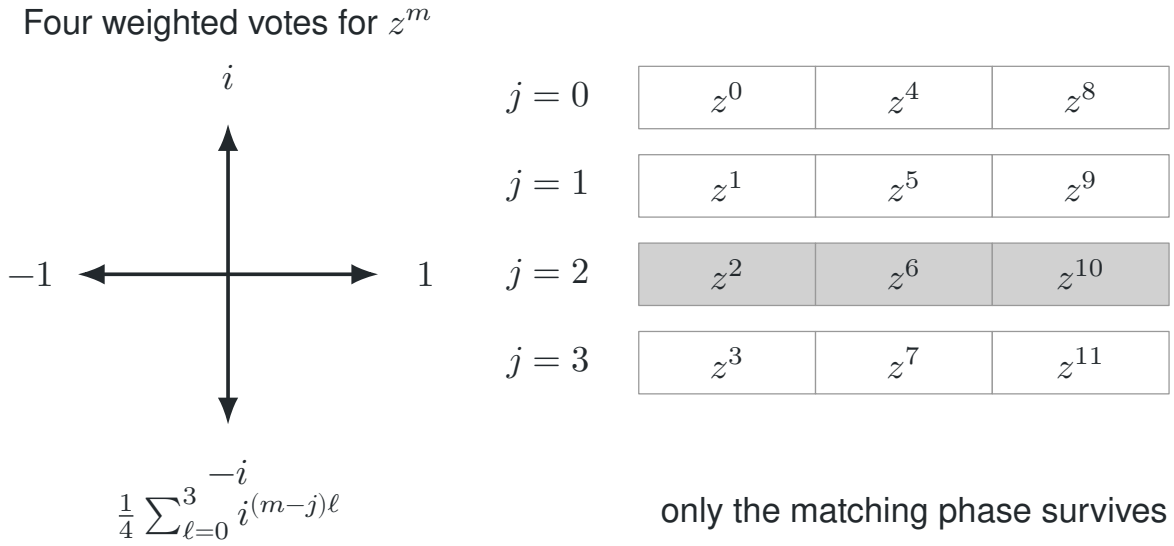
A root filter is an exact vote

Chapter IV: Theorem 3.1, Proposition 3.2.

For $N = 4$, write $\omega = i$. To extract residue j, evaluate the function at four rotated arguments and attach the phase $i^{-j\ell}$:

$$(\mathcal{B}_{4,j}F)(z) = \frac{1}{4}\sum_{\ell=0}^{3} i^{-j\ell}F(i^{\ell}z).$$

A monomial $a_m z^m$ is multiplied by the finite average of $i^{(m-j)\ell}$. Matching phases all vote 1. Nonmatching phases cancel exactly.



An example that can be checked coefficient by coefficient

Take $F(z) = \sum_{m=0}^{12}(m+1)z^m$. Then

$$\mathcal{B}_{4,2}F = 3z^2 + 7z^6 + 11z^{10}.$$

Applying the same selector again changes nothing. Applying a different selector gives zero. Summing the four selected polynomials returns the original F.

Selection is also a symmetry condition

For the selected polynomial F_2, rotation gives $F_2(iz) = -F_2(z)$. More generally, a residue-j section satisfies $F_j(\omega z) = \omega^j F_j(z)$. The phase label therefore describes both the coefficient addresses and the response to rotation.

> **Read this as:** This is not an approximate Fourier fit. The finite root sum is exactly 0 or 1 on every coefficient.

The checker compares the actual complex average with independent coefficient selection for this polynomial. The drawing shows the four complex phases, not a numerical convergence argument.

PLATE IV.4 / CYCLIC CONVOLUTION

Products mix the residue lanes

Chapter IV: Theorem 3.3.

Selection separates a function additively. It does not make multiplication diagonal. A residue-i term multiplied by a residue-j term lands in residue $i + j \pmod N$. For $N = 3$, use

$$F = 1 + 2z + 3z^2 + z^3, \qquad G = 2 + z + z^2.$$

The table displays every product of their three selected sections.

	2	z	z^2
$z^3 + 1$	$2z^3 + 2$	$z^4 + z$	$z^5 + z^2$
$2z$	$4z$	$2z^2$	$2z^3$
$3z^2$	$6z^2$	$3z^3$	$3z^4$

$$\mathcal{B}_{3,0}(FG) = 2 + 7z^3$$

shaded cells: $(0,0), (1,2), (2,1)$ all land in residue 0

The shaded cells contribute to residue zero:

$$2(1 + z^3) + (2z)(z^2) + (3z^2)z = 2 + 7z^3.$$

Keeping only the product of the residue-zero inputs would give $2 + 2z^3$ and lose five units of the z^3 coefficient. The mixed cells are part of the exact result.

The general rule

$$\mathcal{B}_{N,k}(FG) = \sum_{i+j \equiv k \ (N)} (\mathcal{B}_{N,i}F)(\mathcal{B}_{N,j}G).$$

Every pair belongs to exactly one output residue. No term is discarded and no coefficient is estimated. This finite cyclic convolution is the algebra behind the later addition formulas and block-coefficient recurrences.

> **Read this as:** A packet label tells you where an interaction lands, not that interactions with other packets vanish.

All nine table entries and all three output sections are checked as exact polynomial identities. This is the scalar cyclic specialization used by the paper.

PLATE IV.5 / DIFFERENTIAL SYSTEM

One signed edge determines the cycle

Chapter IV: Theorem 5.3, Theorem 6.4, Corollary 6.5, Corollary 6.6.

Let $J_3e_0 = e_1$, $J_3e_1 = e_2$, and $J_3e_2 = -e_0$. A complete turn changes sign, so $J_3^3 = -I$. The signed sections are the coordinates of $e^{zJ_3}e_0$.

J_3

0	0	-1
1	0	0
0	1	0

e^{zJ_3}

D_0	$-D_2$	$-D_1$
D_1	D_0	$-D_2$
D_2	D_1	D_0

$$J_3^3 = -I$$

$$\mathbf{D}_3(0) = e_0$$

$$\det e^{zJ_3} = 1$$

$$D_0 \mapsto -D_2, \qquad D_1 \mapsto D_0, \qquad D_2 \mapsto D_1$$

Here D_j inside the matrix means $D_{3,j}(z)$. For example,

$$D_{3,0} = 1 - \frac{z^3}{3!} + \frac{z^6}{6!} - \cdots, \qquad D_{3,1} = z - \frac{z^4}{4!} + \cdots.$$

Rows are output coordinates and columns are input coordinates. The upper-triangular minus signs record crossing the signed edge while moving from the column to the row. They are not arbitrary orientation choices.

Two checks connect the matrix and the scalar equation

The initial derivative matrix is the identity. Each initial jet $(y(0), y'(0), y''(0))$ therefore determines a unique linear combination. Also,

$$\frac{1}{3}\operatorname{Tr}(e^{zJ_3}) = D_3(z), \qquad \det(XI - J_3) = X^3 + 1.$$

The roots of -1 appearing later in the exponential sum are already the spectrum of this concrete matrix.

> **Read this as:** The functions, their initial data, the addition law, and the exponent polygon come from one signed shift.

The exact tests check every matrix-entry Taylor coefficient through degree 14 for $N = 1, \dots, 5$. The displayed determinant-one law is for $N \geq 2$. $N = 1$ is a separate boundary case.

PLATE IV.6 / FINITE ALGEBRA

Addition is quotient multiplication

Chapter IV: Proposition 5.5, Theorem 6.2.

Work in $\mathbb{C}[\xi]/(\xi^3+1)$. Every element has coordinates in $1, \xi, \xi^2$. The entire exponential has the finite-coordinate form

$$e^{z\xi} = D_{3,0}(z) + D_{3,1}(z)\xi + D_{3,2}(z)\xi^2.$$

The multiplication table makes the carry sign unavoidable.

	1	ξ	ξ^2
1	1	ξ	ξ^2
ξ	ξ	ξ^2	-1
ξ^2	ξ^2	-1	$-\xi$

$$\xi^3 = -1$$

wrap once: change sign

$$\xi^2\xi^2 = -\xi$$

$$D_{3,0}(z+w) = D_{3,0}(z)D_{3,0}(w) - D_{3,1}(z)D_{3,2}(w) - D_{3,2}(z)D_{3,1}(w)$$

Multiply the coordinate forms of $e^{z\xi}$ and $e^{w\xi}$. Their product is $e^{(z+w)\xi}$. The coefficient of 1 receives one contribution without a wrap and two contributions with a wrap. Reading that coefficient gives the displayed addition law.

Compare the unsigned model

In $\mathbb{C}[\xi]/(\xi^3-1)$, the same table has no minus signs. Its exponential coordinates are the unsigned sections $E_{3,j}$. Thus the difference between the two differential systems is a single algebraic relation, not a change in how coefficients are grouped. For general N, the carry factor is

$$(-1)^{\lfloor (i+j)/N \rfloor}, \qquad 0 \le i, j < N.$$

Only zero or one wrap is possible in a product of two coordinate basis elements.

> **Read this as:** The twisted addition law is the ordinary exponential group law expressed in a signed cyclic algebra.

Bivariate Taylor identities are checked through total degree ten in every section for $N = 2, 3, 4$. Omitting the carry sign is one of the deliberately rejected controls.

PLATE IV.7 / FACTORIAL NORMALIZATION

Why a hypergeometric name appears

Chapter IV: Theorem 4.5.

The hypergeometric form is another way to organize the same coefficients. It does not introduce an extra function into the construction. Split a factorial into its residue classes and divide each factor by N:

$$(Nn+j)! = j!N^{Nn}n! \prod_{a\in A_{N,j}} (a)_n.$$

The parameter equal to 1 is omitted from $A_{N,j}$ because the hypergeometric convention already supplies $(1)_n = n!$.

$$N=3,\ j=1,\ n=2: \quad 7! = 1!\,3^6\,2!\,(2/3)_2(4/3)_2$$

$\frac{2}{3}$	$\frac{5}{3}$
1	2
$\frac{4}{3}$	$\frac{7}{3}$

$(2/3)_2$

$(1)_2 = 2!$

$(4/3)_2$

$$A_{3,1} = \{2/3, 4/3\}$$

$$E_{3,1}(z) = z\,{}_0F_2\left(;\frac{2}{3},\frac{4}{3};\frac{z^3}{27}\right)$$

The omitted unit parameter is already the hypergeometric $n!$.

Check the displayed case

For $N=3, j=1, n=2$, the six factors after the initial $1!$ are $2,3,4,5,6,7$. Dividing by three and grouping them gives the three rows of the plate. The middle row supplies $2!$. The others supply $(2/3)_2$ and $(4/3)_2$. The right-hand side is exactly 5040. Consequently,

$$E_{3,1}(z) = z\left(1+\frac{z^3}{4!}+\frac{z^6}{7!}+\cdots\right).$$

The prefactor z retains the forced zero at the origin. The hypergeometric argument $z^3/27$ retains the scaling N^N.

> **Read this as:** Parameters, the omitted unit parameter, the prefactor, and the argument scaling all come from factorial bookkeeping.

Five choices of (N, j) are checked against the exact factorial identity for $n = 0, \dots, 7$. These finite checks support the example. The general analytic identification belongs to the inherited manuscript.

PLATE IV.8 / THE REDUCED VARIABLE

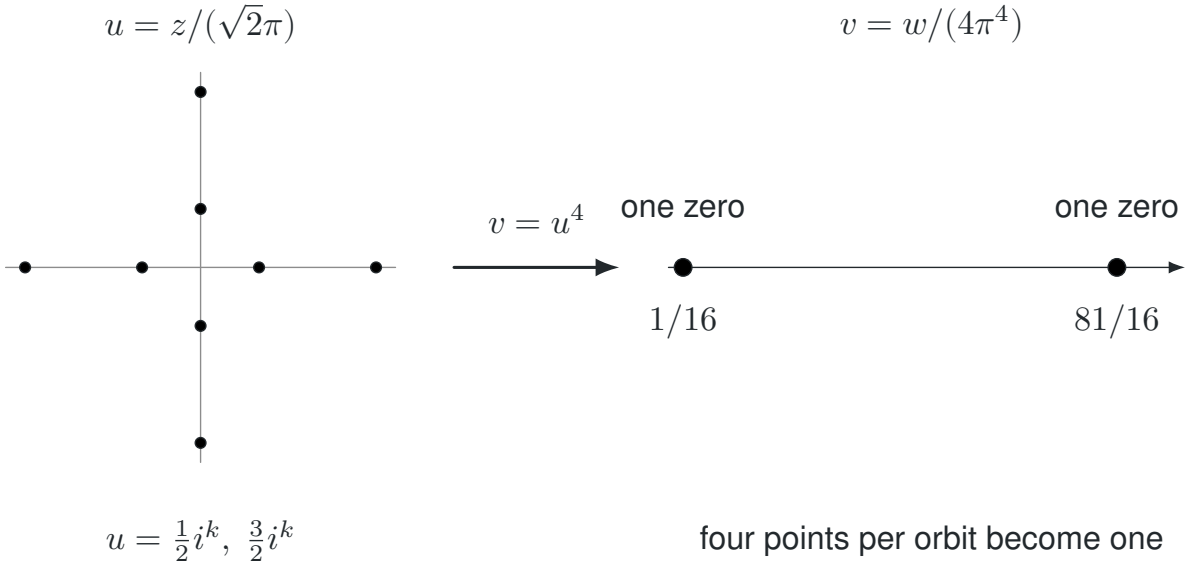
Four zeros become one orbit zero

Chapter IV: Definition 5.7, Example 11.12, Corollary 11.13.

For $N = 4$ the factorization is exact:

$$D_4(z) = \cos(z/\sqrt{2})\cosh(z/\sqrt{2}), \qquad D_4(z) = \Delta_4(z^4).$$

Set $A_4 = \sqrt{2}\pi$. All the zeros are $A_4(m + 1/2)i^k$, with $m \geq 0$ and $0 \leq k < 4$. Each zero is simple.



The drawing uses $u = z/A_4$ and $v = w/A_4^4$, so its marked coordinates are rational. The first four-point orbit has fourth power $1/16$. The second has fourth power $81/16$.

Multiplicity is not multiplied in the reduced function

At a nonzero point, $z \mapsto z^4$ has nonzero derivative. It preserves the local multiplicity of a zero. Four distinct simple zeros of D_4 therefore correspond to one simple zero of Δ_4, not one zero of multiplicity four. The reduced zeros are

$$\lambda_m = A_4^4(m + 1/2)^4 = 4\pi^4(m + 1/2)^4.$$

This is why an orbit factor in the z plane is $1 - z^4/\lambda_m$. It is the product of the four corresponding linear factors, with their phases included.

> **Read this as:** Orbit reduction removes repetition. It changes growth and counting exponents, while preserving each nonzero zero's local multiplicity.

The quartic factorization, its differential equation, and the displayed orbit powers are checked independently. Here the complete zero set is known exactly. A finite numerical root search is not being used to establish it.

PLATE IV.9 / COEFFICIENT RECONSTRUCTION

A reciprocal is a triangular solve

Chapter IV: Theorem 7.3, Theorem 7.6.

Start in the reduced variable with

$$\Delta_2(w) = 1 - \frac{w}{2} + \frac{w^2}{24} - \frac{w^3}{720} + \cdots.$$

Write $1/\Delta_2(w) = \sum b_n w^n$. Matching the product with 1 gives a unit lower-triangular system. At each row there is exactly one new unknown.

denominator convolution				b		unit series
1	0	0	0	1	$=$	1
$-\frac{1}{2}$	1	0	0	$\frac{1}{2}$		0
$\frac{1}{24}$	$-\frac{1}{2}$	1	0	$\frac{5}{24}$		0
$-\frac{1}{720}$	$\frac{1}{24}$	$-\frac{1}{2}$	1	$\frac{61}{720}$		0

$$b_1 = \tfrac{1}{2}$$

$$b_2 = \tfrac{1}{4} - \tfrac{1}{24}$$

$$b_3 = \tfrac{61}{720}$$

$$P_{2,n}(1) = (2n)!b_n : \qquad 1,\ 1,\ 5,\ 61$$

The first three equations are

$$b_1 - \tfrac{1}{2} = 0, \qquad b_2 - \tfrac{1}{2}b_1 + \tfrac{1}{24} = 0, \qquad b_3 - \tfrac{1}{2}b_2 + \tfrac{1}{24}b_1 - \tfrac{1}{720} = 0.$$

They yield $b_1 = 1/2$, $b_2 = 5/24$, and $b_3 = 61/720$. Multiplication by $(2n)!$ produces the secant coefficients $1, 5, 61$.

Why this is an exact finite calculation

To compute the coefficient of w^n, only denominator coefficients through w^n are needed. Later terms of the entire series cannot affect it. A finite series inverse is therefore exact at its declared truncation order, even though evaluating that truncated series is only an approximation to the full analytic function away from the origin.

> **Read this as:** A unit constant term makes coefficient recovery triangular. It does not by itself specify the radius of analytic convergence.

The checks compare an independent binomial-series expansion, the parameter recurrence, and the logarithmic Newton recurrence through block index six. The table uses ordinary coefficients, with factorial conversion shown separately.

PLATE IV.10 / BELL PARTITIONS

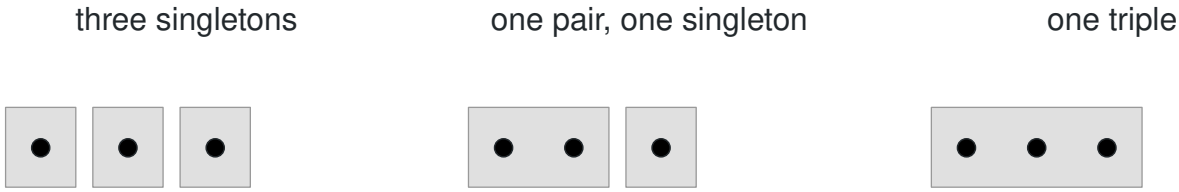
The reciprocal order is a polynomial

Chapter IV: Theorem 7.2, Corollary 7.7.

Let r vary instead of fixing the reciprocal power. Since

$$\Delta_N(w)^{-r} = \exp\left(r \sum_{m\geq 1} \kappa_{N,m} \frac{w^m}{m} \right),$$

the coefficient of w^n is a polynomial of degree n in r. Bell polynomials collect the possible ways logarithmic terms contribute total degree n.



$r^3\kappa_1^3$	$3r^2\kappa_1\kappa_2$	$2r\kappa_3$
$15r^3$	$30r^2$	$16r$

$$P_{2,3}(r) = \tfrac{6!}{3!}\, B_3(r\kappa_1, r\kappa_2, 2r\kappa_3)$$

For $N = 2$, the first three logarithmic coefficients are

$$\kappa_1 = \frac{1}{2}, \qquad \kappa_2 = \frac{1}{6}, \qquad \kappa_3 = \frac{1}{15}.$$

The third Bell polynomial is $B_3(x_1, x_2, x_3) = x_1^3 + 3x_1x_2 + x_3$. Substituting $x_1 = r\kappa_1$, $x_2 = r\kappa_2$, $x_3 = 2r\kappa_3$ and multiplying by $6!/3!$ gives

$$P_{2,3}(r) = 15r^3 + 30r^2 + 16r.$$

At $r = 1$ this is 61. At $r = 2$ it is 272. At $r = 0$ it vanishes, as every positive-index reciprocal coefficient must.

The leading term has a simple source

Only n copies of the linear logarithmic term can contribute $r^n w^n$. Therefore

$$[r^n]P_{N,n}(r) = \frac{(Nn)!}{n!(N!)^n}.$$

> **Read this as:** The polynomial parameter records how many logarithmic pieces are combined. It is not a numerical interpolation in the reciprocal order.

The partition plate is a finite combinatorial expansion. The exact tests verify the leading coefficient and parameter polynomial through block index six for five orders.

PLATE IV.11 / NEWTON AND ADDITION

Two recurrences reach the same answer

Chapter IV: Theorem 7.3, Theorem 7.4, Theorem 7.6.

One recurrence uses the denominator coefficients. Another uses its logarithm. With $b_n(r) = P_{N,n}(r)/(Nn)!$, the shorter logarithmic form is

$$nb_n(r) = r\sum_{k=1}^{n} \kappa_{N,k} b_{n-k}(r), \qquad b_0 = 1.$$

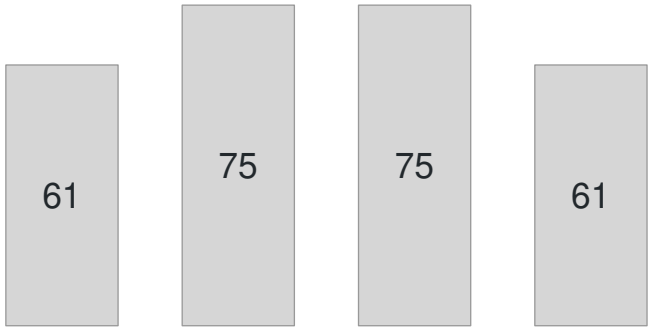
The negative superdiagonal in the Newton determinant is essential.

$n!b_n(r)$ for $n = 3$

$\frac{r}{2}$	-1	0
$\frac{r}{6}$	$\frac{r}{2}$	-2
$\frac{r}{15}$	$\frac{r}{6}$	$\frac{r}{2}$

$\det /6 = P_{2,3}(r)/720$

$P_{2,3}(2) = 61 + 75 + 75 + 61 = 272$



block weights: $1, 15, 15, 1$

The additive parameter law has block weights

Using the same logarithm, $D_N^{-r} D_N^{-s} = D_N^{-(r+s)}$. Thus

$$P_{N,n}(r+s) = \sum_{q=0}^{n} \binom{Nn}{Nq} P_{N,q}(r) P_{N,n-q}(s).$$

For $N = 2, n = 3, r = s = 1$, the four weights are $1, 15, 15, 1$. The two middle terms are $15 \cdot 1 \cdot 5$ and $15 \cdot 5 \cdot 1$. Together with the endpoints, they give 272. Using $\binom{n}{q}$ instead would be incorrect for these coefficients. That ordinary binomial law belongs to the separately normalized sequence $n!P_{N,n}/(Nn)!$.

> **Read this as:** Agreement of independent recurrences is a useful check precisely because their bookkeeping is different. Neither permits silently changing the factorial convention.

The checker compares both recurrence routes and the determinant, then checks the parameter-addition identity symbolically. A deliberately incorrect ordinary-binomial substitution is rejected.

PLATE IV.12 / LOGARITHMIC DIFFERENTIATION

Compress the ODE into a tangent

Chapter IV: Proposition 8.2, Corollary 8.4, Theorem 8.5.

The transfer tangent is $T_N = -D_N'/D_N$. It compresses the linear differential equation into a nonlinear scalar identity. This division is allowed only on the zero-free set of the denominator.

LINEAR EQUATION	LOGARITHMIC DERIVATIVE
$D_2'' = -D_2$	$T_2' = 1 + T_2^2$
$D_3''' = -D_3$	$T_3'' - 3T_3T_3' + T_3^3 = 1$

$$T_N = -D_N'/D_N \quad \text{only where } D_N \neq 0$$

$$T_4(z) = \frac{z^3}{3!} + 34\frac{z^7}{7!} + \cdots$$

The coefficients retain the reduced logarithmic information:

$$T_N(z) = N\sum_{m\geq 1} \kappa_{N,m} z^{Nm-1}, \qquad T_N(n) = N(Nn+N-1)!\kappa_{N,n+1}.$$

For order four, $\kappa_{4,1} = 1/24$ and $\kappa_{4,2} = 17/10080$. These give $T_4(0) = 1$ and $T_4(1) = 34$, explaining the last line of the plate.

A nonlinear equation with a precise origin

If $D = e^g$ locally, then $g' = -T$. Differentiating e^g three times gives $D'''/D = -T'' + 3TT' - T^3$. Setting $D'''/D = -1$ yields the cubic Riccati identity shown above. No additional nonlinear model has been assumed. The reciprocal family also satisfies $(D_N^{-r})' = rT_N D_N^{-r}$ on a fixed-log domain. The tangent has a pole at each zero of D_N. Positive integer reciprocal powers also have poles there. General complex powers retain the stated branch convention.

> **Read this as:** The nonlinear tangent equation is exact where division is legal. It is not a global regular differential equation across the zeros.

The tests verify the tangent coefficient recurrence, the exact cubic ODE, and the symbolic Riccati reduction after clearing denominators. The order-one tangent is the separate constant function $T_1 = 1$.

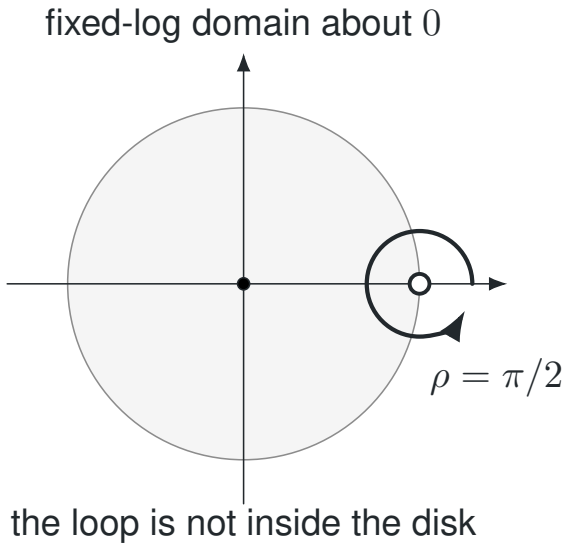
PLATE IV.13 / BRANCH BOOKKEEPING

A complex power needs a domain

Chapter IV: Section 2, Definition 7.1, Corollary 8.4.

A formal unit series has a formal logarithm at the origin. To interpret that germ analytically, choose a simply connected zero-free region containing 0 and normalize the logarithm to vanish there.

$$D_N(z)^{-r} = \exp(-rL(z)), \qquad e^{L(z)} = D_N(z), \quad L(0) = 0.$$



$$D(z) \sim D'(\rho)(z-\rho)$$

$$\log D \longmapsto \log D + 2\pi i$$

$$D^{-r} \longmapsto e^{-2\pi i r} D^{-r}$$

$$r = 1/2 : \ -1 \qquad r \in \mathbb{Z} : \ 1$$

For $D_2 = \cos z$, the open disk $|z| < \pi/2$ is an admissible domain. A small loop around the simple zero $\rho = \pi/2$ is not contained in that disk. Analytic continuation around the zero changes the local logarithm by $2\pi i$, and the power by $e^{-2\pi i r}$.

Three different kinds of statement

A *formal* identity compares coefficients at the origin. A *holomorphic* identity for complex powers uses a common specified logarithm. A positive *integer* reciprocal power is globally meromorphic and needs no branch choice. These scopes cannot be exchanged without a further argument. For example, $D^{-r}D^{-s} = D^{-(r+s)}$ is immediate using one fixed logarithm. Combining values built from unrelated branches can introduce additional phase factors.

> **Read this as:** The branch choice is mathematical data. It is not a plotting convention that can be removed by drawing a cleaner curve.

The domain picture is schematic. The phase change comes from the local simple-zero model and winding number, not from a sampled loop. Later nearest-pole asymptotics retain positive integral reciprocal order.

PLATE IV.14 / NORMALIZED SECTIONS

Three similar reciprocals differ

Chapter IV: Definition 9.1, Theorem 9.2, Example 9.11.

The signed denominator and the normalized unsigned sections produce different reciprocal families. In order two,

$$D_2^{-1} = \sec z, \qquad C_{2,0} = \operatorname{sech} z, \qquad C_{2,1} = z/\sinh z.$$

Their first block-EGF coefficients make the difference visible.

function	1	$z^2/2!$	$z^4/4!$	$z^6/6!$
$\sec z$	1	1	5	61
$\operatorname{sech} z$	1	-1	5	-61
$z/\sinh z$	1	$-\frac{1}{3}$	$\frac{7}{15}$	$-\frac{31}{21}$

$$E_{2,1}(z) = \sinh z = z\Phi_{2,1}(z^2), \qquad C_{2,1}(0) = 1$$

Remove the forced zero before taking a reciprocal

The section $E_{N,j}$ begins with $z^j/j!$. Define

$$E_{N,j}(z) = \frac{z^j}{j!}\Phi_{N,j}(z^N), \qquad \Phi_{N,j}(0) = 1.$$

Then $C_{N,j}(z) = 1/\Phi_{N,j}(z^N)$ has value 1 at the origin. For $j > 0$, the notation $(z^j/j!)/E_{N,j}(z)$ means this removable continuation. It does not mean evaluating an undefined $0/0$ expression. For $N = 2, j = 1$, the first nontrivial coefficient is $-1/3$ in front of $z^2/2!$, equivalently $-z^2/6$. Dropping the numerator z would instead produce the pole of $1/\sinh z$.

> **Read this as:** A minus sign in the denominator series and removal of an initial zero are different operations. Their reciprocal coefficients should not be interchanged.

All entries in the table are exact rationals computed from independent triangular inverses. The table explicitly uses block-EGF columns. Ordinary coefficients are obtained by dividing by the displayed factorials.

PLATE IV.15 / APPELL POLYNOMIALS

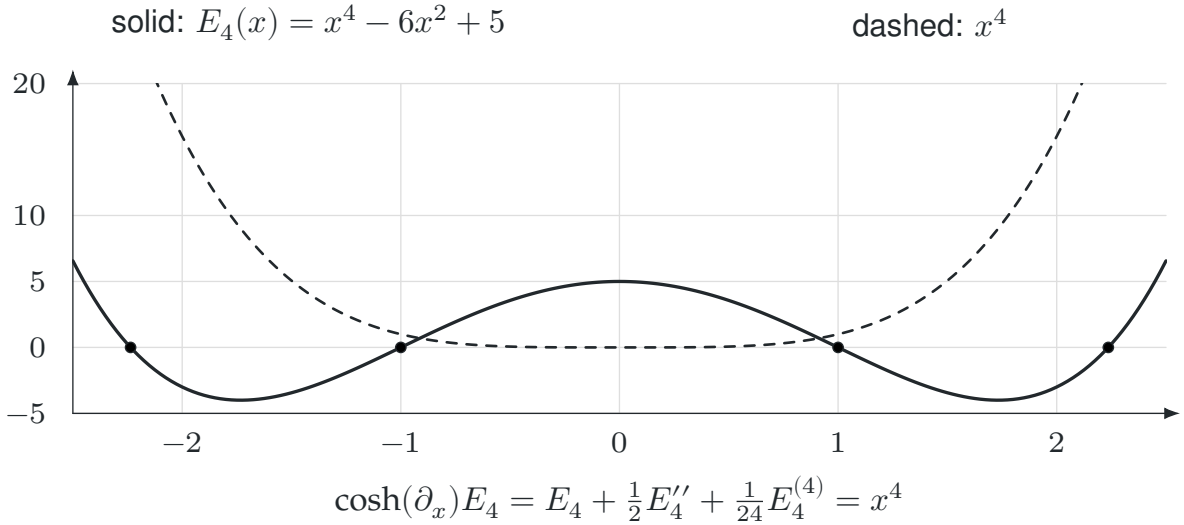
An infinite symbol acts finitely

Chapter IV: Theorem 9.6, Theorem 9.8, Proposition 9.9.

For $C_{2,0}(z) = \operatorname{sech} z$, define $E_m(x)$ by $\operatorname{sech} z\, e^{xz} = \sum E_m(x) z^m/m!$. The formal differential-operator formula is

$$E_m(x) = \operatorname{sech}(\partial_x) x^m.$$

Although the symbol is an infinite series, derivatives above degree m vanish.



$$\cosh(\partial_x) E_4 = E_4 + \tfrac{1}{2} E_4'' + \tfrac{1}{24} E_4^{(4)} = x^4$$

For $m = 4$, only three terms survive:

$$\left(1 - \frac{\partial_x^2}{2!} + 5\frac{\partial_x^4}{4!}\right) x^4 = x^4 - 6x^2 + 5.$$

The four real roots of this polynomial are $\pm 1, \pm\sqrt{5}$. They are not the zeros of the original analytic denominator. The plot distinguishes the input polynomial and its transformed output.

An inverse operation, not a truncation error

Applying $\cosh(\partial_x)$ returns x^4 exactly. The general inverse uses the reciprocal formal symbol. The lowering and translation identities are

$$E_m' = m E_{m-1}, \qquad E_m(x+y) = \sum_{q=0}^{m} \binom{m}{q} E_q(x) y^{m-q}.$$

Both follow from the generating function. No operator-series convergence assumption on an infinite-dimensional function space is needed for this polynomial calculation.

> **Read this as:** Formal substitution into the derivative is safe here because every polynomial sees only finitely many derivatives. That is a domain statement.

The polynomial, its inverse image, and the Appell identities are checked symbolically through degree eight in four normalized-section families. The displayed curves are samples of exact polynomials.

PLATE IV.16 / THETA COUPLING

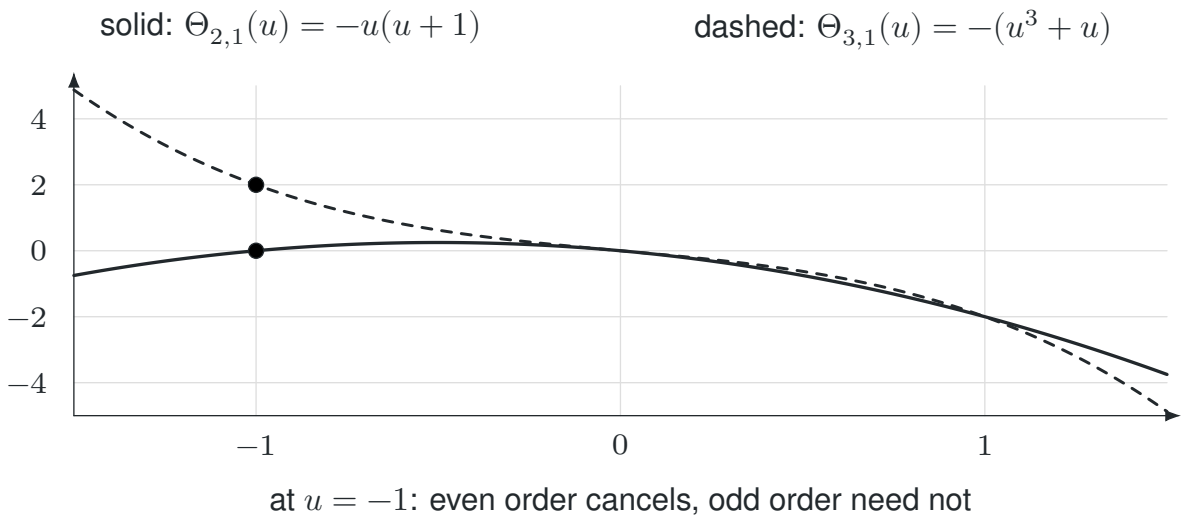
Scale and order enter differently

Chapter IV: Theorem 10.2, Proposition 10.3, Proposition 10.4.

Let $H_N(z) = E_{N,0}(z)$. The theta family uses the same parameter u twice, but in two different roles:

$$H_N(uz)^{-1} H_N(z)^{-u} = \sum_{n \geq 0} \Theta_{N,n}(u) \frac{z^{Nn}}{(Nn)!}.$$

Argument scaling contributes u^{Nm} at logarithmic block m. Reciprocal order contributes u.



Their combined logarithm has coefficient

$$(-1)^m \kappa_{N,m} \frac{u^{Nm} + u}{m}.$$

At the first block this gives $\Theta_{N,1}(u) = -(u^N + u)$. Every positive-index theta polynomial vanishes at $u = 0$. For even N, $H_N(-z) = H_N(z)$, so the defining product is also 1 at $u = -1$.

The odd-order boundary is visible

For $N = 3$, $\Theta_{3,1}(-1) = 2$, not zero. The factor $u + 1$ is not a uniform law across orders. In contrast,

$$\Theta_{2,2}(u) = u(u+1)(5u^2 + u + 2).$$

The two guaranteed factors follow from the whole generating function, not from guessing the first few polynomial roots.

> **Read this as:** A common parameter does not make argument scaling and reciprocal order interchangeable. Their powers must be tracked separately.

The exact checker compares theta convolution with its logarithmic Newton recurrence through block index four. The odd-order factor claim is explicitly rejected as an invalid control.

PLATE IV.17 / DOMINANT EXPONENTS

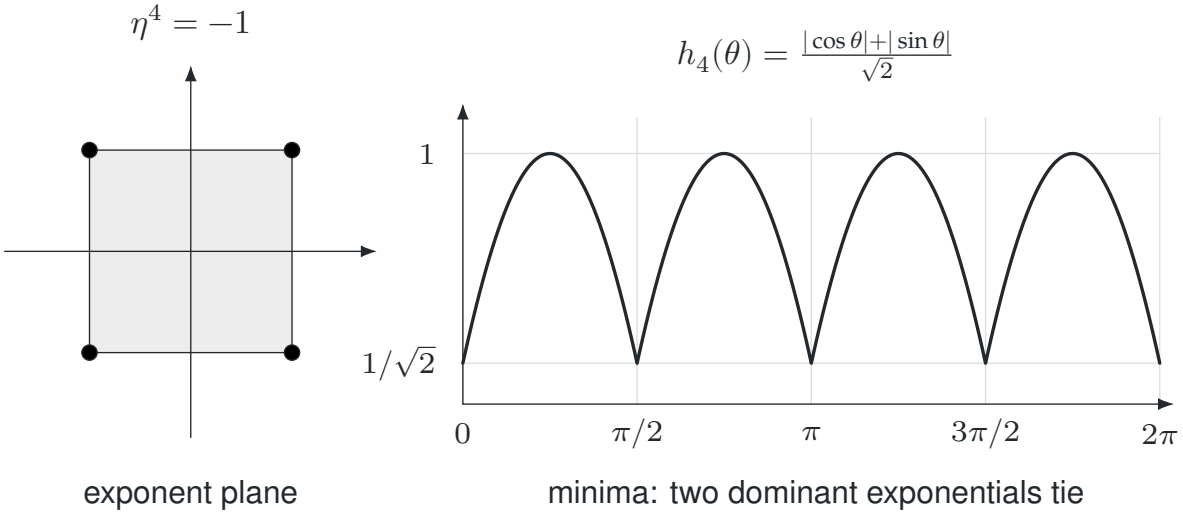
Growth comes from a polygon

Chapter IV: Definition 11.3, Theorem 11.4.

Write the signed denominator as a finite exponential sum:

$$D_N(z) = \frac{1}{N} \sum_{\eta^N=-1} e^{\eta z}.$$

For $z = re^{i\theta}$, the size of one exponential is $e^{r\Re(\eta e^{i\theta})}$. The maximal projection determines the directional exponential growth.



The indicator is

$$h_N(\theta) = \max_{0 \le k < N} \cos\left(\theta + \frac{(2k+1)\pi}{N}\right).$$

Away from a tie direction, one exponential dominates all the others by a uniform positive projection gap on compact angular sets. For sufficiently large radius, smaller terms cannot cancel it. Those sectors are eventually zero-free.

Why zeros remain possible on a growing ray

At a tie, two adjacent exponents have the same real projection. Their complex phases can be opposite. Their sum can therefore vanish even though each term is exponentially large. The minima in the order-four indicator plot occur precisely at the tie directions $\theta = k\pi/2$. The indicator uses a limsup in radius. It does not assert that $r^{-1}\log|D_N(re^{i\theta})|$ has an ordinary limit through the zeros on a tie ray.

> **Read this as:** The polygon predicts where large cancellation is possible. It does not, by itself, enumerate zeros or prove their simplicity.

The order-four polygon and support-function formula are explicit. The curve is a sampled display of that formula. Eventual localization and uniqueness require the separate zero-analysis arguments.

PLATE IV.18 / THE CUBIC MODEL

Two large terms nearly cancel

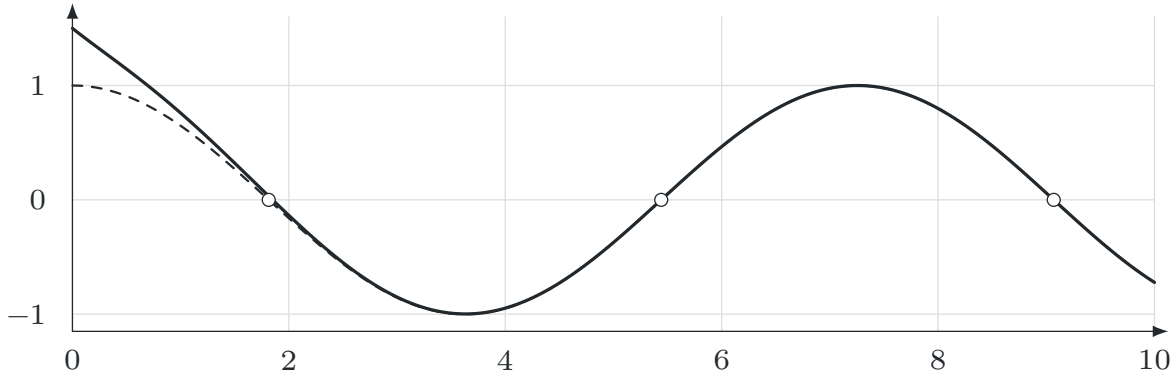
Chapter IV: Lemma 11.5, Theorem 11.7, Corollary 11.9.

For real t, order three has the exact formula

$$D_3(t) = \frac{e^{-t} + 2e^{t/2}\cos(\sqrt{3}t/2)}{3}.$$

Divide by the nonzero factor $2e^{t/2}/3$. The resulting equation separates the dominant cosine from the small third exponential.

$$\tfrac{3}{2}e^{-t/2}D_3(t) = \cos(\sqrt{3}t/2) + \tfrac{1}{2}e^{-3t/2}$$



solid: full expression; dashed: two-term model; circles: model zeros

Ignoring the last term gives model zeros

$$t_m = \frac{2\pi}{\sqrt{3}}(m + 1/2).$$

At t_m, the full normalized expression equals $\frac{1}{2}e^{-3t_m/2} > 0$, so t_m is not an exact zero. Linearizing the cosine at that point predicts

$$\delta_m^{(1)} = \frac{(-1)^m}{\sqrt{3}}e^{-3t_m/2}.$$

The alternating slope explains why the true zero is to the right of the first model zero, then to the left of the second, and so on in the eventual regime.

Symmetry needs uniqueness

Real coefficients alone allow nonreal conjugate pairs. The manuscript first proves that a near-model disk contains exactly one zero. Conjugation preserves that disk, so the unique zero must equal its conjugate. This additional uniqueness step, not the symmetry alone, places it exactly on the ray.

> **Read this as:** The correction is caused by a real, nonzero remainder. A good cosine approximation does not make its model zeros exact.

The curves are sampled analytic expressions. The next page independently certifies four narrow real-zero intervals rather than trusting the plotted crossing or the floating-point root finder.

PLATE IV.19 / EXACT RATIONAL BRACKETS

A root estimate becomes a certificate

Chapter IV: Theorem 11.7, Theorem 11.8, Corollary 11.9.

The root finder is used only to suggest intervals. Each displayed interval is accepted by exact rational arithmetic: opposite endpoint signs prove existence, and an interval bound for D_3' excludes zero throughout the interval, proving uniqueness inside it.

	certified lower endpoint	$\tau_m - t_m$	$\delta_m/\delta_m^{(1)} - 1$
$m = 0$	1.84981279919014347629	$+3.60134 \times 10^{-2}$	-5.243×10^{-2}
$m = 1$	5.44123335502366335540	-1.64738×10^{-4}	$+2.471 \times 10^{-4}$
$m = 2$	9.06899753487157765805	$+7.13700 \times 10^{-7}$	-1.071×10^{-6}
$m = 3$	12.69659554654675729440	-3.09277×10^{-9}	$+4.639 \times 10^{-9}$

Each upper endpoint is the displayed lower endpoint plus 10^{-20}.

Root intervals: exact rational certificates. Displacement columns: numerical diagnostics.

How the certificate works

Evaluate the Taylor polynomial

$$S_{90}(t) = \sum_{n=0}^{90} \frac{(-1)^n t^{3n}}{(3n)!}$$

at the two rational endpoints. From the next term onward, magnitudes decrease, because their successive ratio is

$$\frac{t^3}{(3n+1)(3n+2)(3n+3)} < 1$$

there. The alternating-series remainder is bounded by the first omitted term. A separate outward interval evaluation of the differentiated polynomial, plus its tail bound, certifies the derivative sign. The first bracket is

$$1.84981279919014347629 < \tau_0 < 1.84981279919014347630.$$

Its displacement differs from the leading asymptotic prediction by about five percent. The later examples have much smaller relative discrepancies, as the plate records.

> **Read this as:** A small residual is not a root certificate. Endpoint signs plus a derivative bound give a checkable local existence-and-uniqueness statement.

Four certified intervals do not establish that no other real or complex zeros exist. The displacement columns are high-precision diagnostics, not a finite verification of a global big-O theorem. All rational enclosures are included in the checks file.

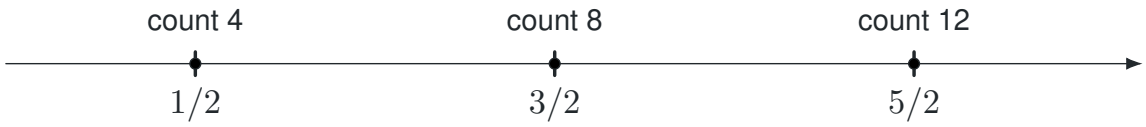
PLATE IV.20 / EXACT FOURTH-ORDER STRINGS

Counting changes under the power map

Chapter IV: Example 11.12, Theorem 11.14.

Order four gives a case with no asymptotic displacement at all. In normalized coordinates the positive radial locations are $u = m + 1/2$, while the reduced-zero locations are $v = (m + 1/2)^4$.

$$u = |z|/(\sqrt{2}\pi)$$



$$v = |w|/(4\pi^4)$$



$$n_{D_4}(\sqrt{2}\pi u) = 4\lfloor u + 1/2 \rfloor, \quad n_{\Delta_4}(4\pi^4 v) = \lfloor v^{1/4} + 1/2 \rfloor$$

The two rows describe the same zero orbits. The upper row counts four zeros each time a new radius is reached. The lower row counts one reduced zero each time its fourth-power location is reached. Both formulas include zeros on the boundary of the disk.

Separate radius conversion from orbit multiplicity

In unscaled variables the general relation is

$$n_{D_N}(R^{1/N}) = N n_{\Delta_N}(R).$$

There are two operations here: replace the radius by its Nth root and divide the count by N. Forgetting either operation changes the leading constant. For this family the asymptotics are

$$n_{D_N}(R) = \frac{N \sin(\pi/N)}{\pi} R + O(1), \qquad n_{\Delta_N}(R) = \frac{\sin(\pi/N)}{\pi} R^{1/N} + O(1).$$

The different powers agree with the entire-function orders 1 and $1/N$.

> **Read this as:** The reduced function has fewer repeated zeros and a different radius scale. Its counting law is not obtained by dividing a same-radius count alone.

The exact order-four staircase formulas are tested at rational radii including jump points. The large-zero theorem for general N permits finitely many exceptions. These exact order-four identities do not justify discarding that boundary elsewhere.

PLATE IV.21 / CANONICAL PRODUCT

A product and an operator share zeros

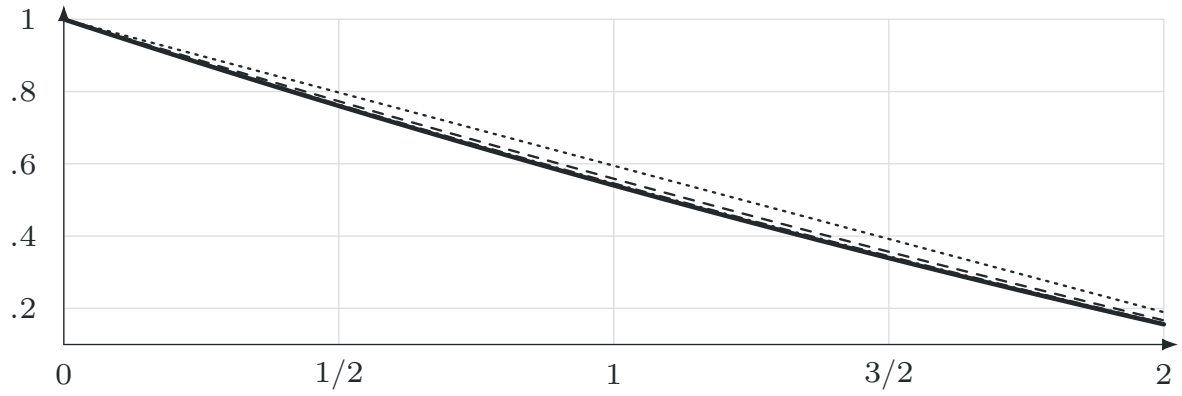
Chapter IV: Theorem 12.1, Theorem 12.2.

For $N \geq 2$, the reduced entire function has order $1/N < 1$. Its multiplicity-sensitive genus-zero product is

$$\Delta_N(w) = \prod_{\lambda \in \Lambda_N} (1 - w/\lambda).$$

Putting $K_N e_\lambda = \lambda^{-1} e_\lambda$ on $\ell^2(\Lambda_N)$ gives the corresponding diagonal Fredholm determinant. Repeated zeros require repeated diagonal entries.

$$\Delta_2(w) = \cos\sqrt{w}, \qquad P_L(w) = \textstyle\prod_{m=0}^{L-1}(1 - w/\lambda_m)$$



thick: entire function; dotted $L = 1$; dashed $L = 3$; dash-dot $L = 10$

Finite compression is an approximation

For $N = 2$, $\lambda_m = \pi^2(m + 1/2)^2$. The finite determinant P_L is exactly the displayed finite product, but it is not exactly Δ_2. For $0 \leq w \leq 1$ and $L \geq 1$, an elementary tail estimate gives

$$0 \leq 1 - \frac{\Delta_2(w)}{P_L(w)} \leq w \sum_{m \geq L} \lambda_m^{-1} \leq \frac{1}{9(L - 1/2)}.$$

The last bound uses $\pi > 3$ and integral comparison. Thus truncation has an explicit error mechanism.

> **Read this as:** The infinite product is justified by the analytic theorem. Finite determinants provide controlled approximations, not substitutes for that theorem.

The exact finite determinant test includes a repeated eigenvalue and rejects deduplication. The curves use known cosine zeros. Order one is excluded: $\Delta_1(w) = e^{-w}$ has no zeros, so its empty product would be 1, not the function.

PLATE IV.22 / ZERO POWER SUMS

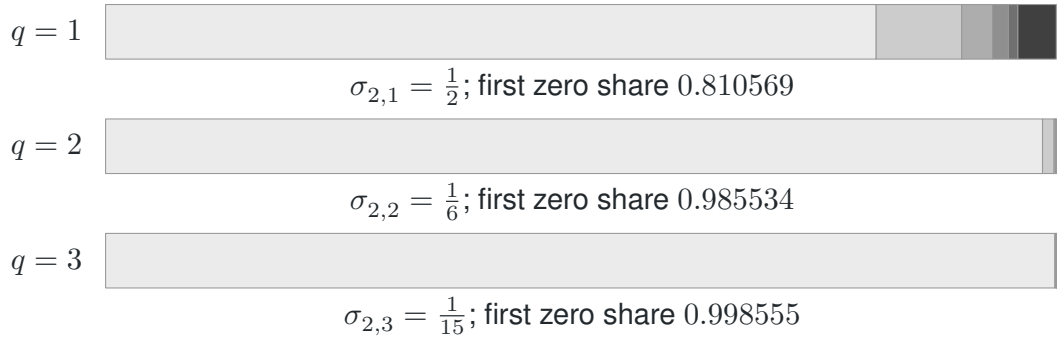
Finite coefficients give infinite moments

Chapter IV: Theorem 12.6, Corollary 12.7, Corollary 12.9, Corollary 12.10.

Taking the logarithm of the canonical product identifies

$$\kappa_{N,q} = \sigma_{N,q} := \sum_{\lambda \in \Lambda_N} \lambda^{-q} = \operatorname{Tr}(K_N^q), \qquad q \geq 1.$$

The same exact numbers can be computed from finitely many denominator coefficients by Newton's recurrence. No numerical zero list is needed for that calculation.



five finite contributions plus the remaining tail. Higher moments emphasize the nearest zero

For order two,

$$\sigma_{2,1} = \frac{1}{2}, \quad \sigma_{2,2} = \frac{1}{6}, \quad \sigma_{2,3} = \frac{1}{15}, \quad \sigma_{2,4} = \frac{17}{630}.$$

The plate shows how successive inverse powers emphasize the nearest zero. Its shaded pieces are contributions from the first five positive reduced zeros, followed by the remaining tail. The total in each row is the exact rational moment.

The tangent is a sum of poles

Logarithmic differentiation also gives

$$T_N(z) = N z^{N-1} \sum_{\lambda \in \Lambda_N} \frac{1}{\lambda - z^N}$$

away from poles. Summability of $|\lambda|^{-1}$ permits the locally uniform partial-fraction expansion without added regularizing terms.

> **Read this as:** A global analytic product theorem connects the infinite zero sum to a finite algebraic recurrence. The recurrence alone does not prove the product theorem.

Displayed moments are checked against formal logarithm coefficients and exact order-two/order-four zeta values. Bar widths are numerical shares. General power sums use λ^{-q}, not $|\lambda|^{-q}$ when exceptional nonpositive zeros are present.

PLATE IV.23 / SINGULARITY ANALYSIS

Nearest poles set coefficient growth

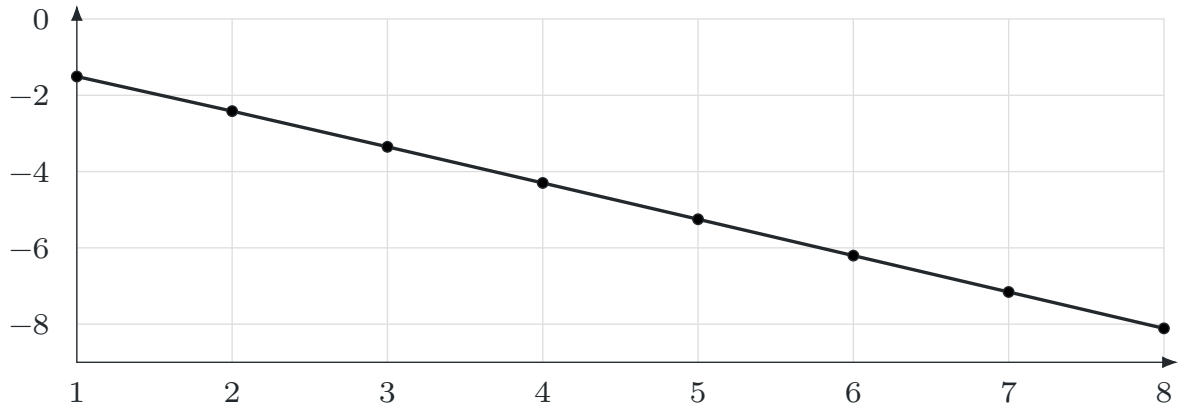
Chapter IV: Theorem 13.1, Corollary 13.2, Corollary 13.3, Corollary 13.4.

For $1/\cos z$, the nearest poles are $\rho = \pm\pi/2$. They contribute equally to even-degree coefficients. Including both poles gives

$$P_{2,n}(1) \sim 2(2n)! \left(\frac{2}{\pi}\right)^{2n+1}.$$

The factor two is orbit multiplicity, not an adjustable fitted constant.

$$\log_{10}(1 - R_n), \quad R_n = \frac{P_{2,n}(1)}{2(2n)!(2/\pi)^{2n+1}}$$



Each ratio is enclosed with a rational Machin-formula bound for π.

A finite quantitative check

For $n = 1, \ldots, 8$, the exact integer coefficients and a rational enclosure for π bound the ratio R_n strictly between zero and one. In this tested range, $1 - R_n < 3^{-(2n+1)}$. The plot displays the logarithm of the enclosed error, not an independently asserted all-n bound.

Higher reciprocal powers need more Laurent data

For positive integral order r, a simple denominator zero becomes a pole of order r. The highest Laurent term contributes a factor $\binom{m+r-1}{r-1}$, but lower Laurent terms still contribute on the same exponential scale. Subtracting *all* principal parts is what leaves a holomorphic remainder controlled by the next zero radius.

> **Read this as:** Nearest-zero geometry explains factorial-times-exponential growth. Multiplicity, reciprocal order, and lower principal parts determine its precise normalization.

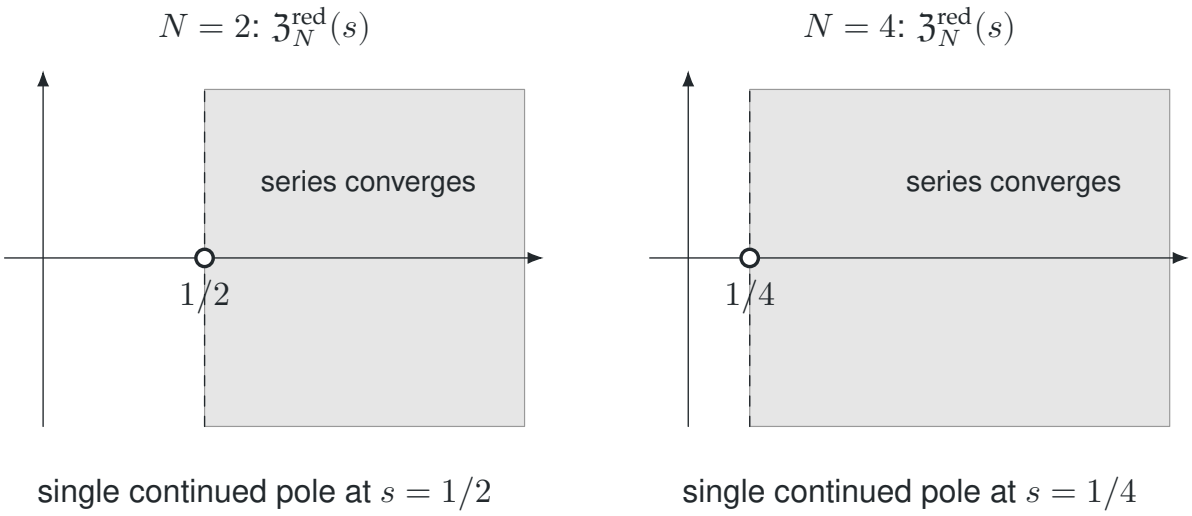
The ratio certificates use a rational Machin-formula enclosure for π. The inherited general theorem retains simple nearest zeros, a gap to the next modulus, and positive integral reciprocal order. The plot does not remove those hypotheses.

PLATE IV.24 / ZERO ZETA FUNCTIONS

Convergence is not continuation

Chapter IV: Theorem 14.2, Theorem 14.3, Corollary 14.4.

Reduced zeros grow like m^N. Their absolute zero-zeta series therefore has a half-plane of absolute convergence $\Re s > 1/N$. The absolute zero zeta in the original variable instead requires $\Re s > 1$.



For $N = 2$ and $N = 4$, the lattices are exact:

$$\mathfrak{Z}_2^{\mathrm{red}}(s) = \pi^{-2s}\zeta(2s, 1/2), \qquad \mathfrak{Z}_4^{\mathrm{red}}(s) = (\sqrt{2}\pi)^{-4s}\zeta(4s, 1/2).$$

Each identity first holds where its defining series converges. The right side supplies its meromorphic continuation. There is a single pole, at $1/2$ or $1/4$, with residue $1/(2\pi)$ or $1/(4\sqrt{2}\pi)$ respectively.

The factor N matters twice

For every order in the paper,

$$\mathfrak{Z}_N^{\mathrm{abs}}(s) = N\mathfrak{Z}_N^{\mathrm{red}}(s/N).$$

For general N, exponentially accurate zero locations leave an entire correction to the Hurwitz model. A finite exceptional zero set also contributes entire terms. This reasoning establishes continuation, not a zeta functional equation.

> **Read this as:** An analytic continuation can exist at points where the original infinite sum diverges. It must not be evaluated there by naively summing that series.

The shaded regions show convergence, and the open circles mark poles of the continued functions. They are not zeros of the zeta functions. Branch-dependent orbit zetas and derivative-weighted zetas retain their separately stated conventions.

PLATE IV.25 / COEFFICIENT AND ZERO WORKED MODEL

One closed calculation: the number 69

Chapter IV: Corollary 7.7, Example 11.12, Theorem 12.6.

Use $N = 4$ throughout this page. Its denominator, its exact zero lattice, and the reciprocal recurrences provide two routes to the same coefficient. All symbols in the plate refer to this one order.

denominator	$1 - w/24 + w^2/40320 + \cdots$
negative logarithm	$w/24 + (17/20160)w^2 + \cdots$
zero moments	$\sigma_1 = 1/24, \quad \sigma_2 = 17/10080$
reciprocal	$1 + w/24 + 69w^2/40320 + \cdots$
tangent	$z^3/3! + 34z^7/7! + \cdots$

$$69 = 8!\,\frac{\sigma_1^2+\sigma_2}{2}, \qquad 34 = 4 \cdot 7!\,\sigma_2$$

Route one: invert the denominator

The w^2 coefficient of Δ_4^{-1} is

$$\frac{1}{24^2} - \frac{1}{8!} = \frac{69}{8!}.$$

Thus $P_{4,2}(1) = 69$. Keeping r variable gives $P_{4,2}(r) = 35r^2 + 34r$.

Route two: use the zero moments

The exact reduced zeros are $4\pi^4(m + 1/2)^4$. Their first two power sums are $\sigma_1 = 1/24$ and $\sigma_2 = 17/10080$. The Bell/Newton identity gives the same reciprocal coefficient $(\sigma_1^2 + \sigma_2)/2$. The tangent independently reads off the second moment as $4 \cdot 7!\sigma_2 = 34$. This links coefficients, geometry, and derivatives without a fitted parameter or an inferred zero list.

> **Read this as:** The content is the agreement of precisely normalized descriptions. A matching small example illustrates the general bridges. It is not a proof of all their analytic hypotheses.

Every rational value and polynomial identity on this page is checked exactly. The infinite zero-sum interpretation uses the inherited canonical-product theorem and the exact order-four zero formula.

PLATE IV.26 / FORMAL AND COMPUTATIONAL BOUNDARIES

What is evidence, and for what?

Chapter IV: Section 15, Section 16.

These twenty-six plates develop the mathematical models accompanying Chapter IV. Their references connect each example to the chapter's statements. The formal companion records the separate verification contracts and evidence scope.

Evidence layer	What it supports
Recorded formal audit	All 118 production modules compiled in the recorded audit. Separate kernel replay and manuscript correspondence checks cover the 68 scoped results. Exact counts, hypotheses, and controls appear in the Lean guide.
Recorded exact checks	All 1,008 rational algebra assertions and fifteen finite transfer-derivative polynomial identities passed. Fifteen deliberate corruptions were rejected. These are bounded checks.
Exact certificates	The retained visual run records 147 checks, five rejected controls, and four rational zero brackets of width 10^{-20}. The brackets certify local existence and uniqueness. The symbolic suites were hash-validated, not rerun in the recorded audit.
Numerical visualization	Sampled analytic curves, finite products, moment shares, and displacement diagnostics have the roles stated on their pages. They are not exact global zero censuses.
Figure provenance	The source records preserve all twenty-four editable TikZ figures. Plate IV.13 retains its reviewed label adjustment.
Editorial preservation	Assembly does not regenerate figures. The original worked formulas, model code, checker, and vector sources are preserved. Locators refer to Chapter IV.

Boundaries that remain in force

The general family has eventual positivity and simplicity of its reduced zeros, not an assertion that every reduced zero is positive and simple. The paper leaves universal simplicity, positivity of every reduced zero, full displacement expansions, and a zeta functional equation open. The exact order-two and order-four models do not settle those questions for every order. Imported standard fact 11.1 stays outside the 68-result contract. Fixed logarithms, zero-free domains, multiplicities, and the hypotheses of nearest-pole asymptotics remain explicit. No historical priority is inferred from successful computation or formal verification.

Replay the worked layer

Run `python build.py` in this editable guide folder to compile the PDF in an isolated temporary directory. The separate model checker is `checks/verify_examples.py`. The plate generator is `source/generate_graphics.py`. They require the listed Python dependencies. Run those mathematical tools in a temporary copy to preserve the delivered evidence.

> **Read this as:** A proof, a finite exact test, a root enclosure, and a sampled curve answer different questions. Keeping their scopes explicit makes the pictures useful mathematical evidence.

The source package preserves the formal sources, proofreading record, verification audit, exact certificates, and rebuild instructions. PDF assembly remains separate from mathematical verification.

CHAPTER V

Multiresolution Signal Calculus

Exact coordinates, singular packets, and stable reconstruction

The analytic constructions give way to signal representations and reconstruction.

Signal models move through filters, packet trees and reconstruction maps. Exact coordinates and singular packets distinguish representation from approximation and recovery.

IN THIS CHAPTER

Mathematical text 383

Visual studies 467

Formal companion 1936

Sources and references 2675

Scope

Separated-jump results require the stated model, gap and grid assumptions. Stable edge-packet recovery and the broader approximation class remain open.

The next chapter asks how transfers evolve in time and how eliminated states contribute memory.

How to use this volume

Part I. The twenty-result formal core develops exact polyphase reconstruction, filter transfers, Hilbert packet operators, finite adaptive trees, edge error bounds, and Smith coordinates.

Part II. The complete extended development treats lattice coordinates, stable banks, singular packets, separated jump recovery, geometric error bounds, and multiscale localization.

Visual atlas. Forty worked plates close the volume. The separate visual guide provides the same examples, and the Lean guide gives the precise formal contract.

Reading this chapter

This chapter develops signal calculus in two complementary parts. Part I gives twenty results on exact coordinates, filtering, adaptive selection and packet error. Part II develops lattice coordinates, stable banks, singular packets and separated jump recovery, with E-prefixed section numbers. The two parts share constructions but use them under different hypotheses.

For a first reading, follow Part I Sections 1–7. These sections establish the coordinate maps, filter transfers, exact reconstruction, and the separation of packet error from smooth approximation error. Section 11 develops finite adaptive selection. Sections 13–14 give the cyclic Smith realization. Continue with Part II Sections E2–E6 for lattice and stability generalizations, E7–E11 for singular packets and recovery, and E12–E14 for nonlinear and Cayley coordinates. Each visual plate identifies the corresponding manuscript result.

The formal companion maps all twenty Part I results to 225 supporting Lean theorem targets across 23 proof modules. Part II has ordinary written proofs and is outside that twenty-result formal map. Exact examples and numerical studies have their own stated scopes. The formal companion and source records distinguish kernel verification, finite checks and numerical replay.

The two broad edge recovery conjectures remain open. Part II proves recovery for a declared one-dimensional separated jump class under its sampling, contrast, and noise assumptions. These results do not supply universal image segmentation or infinite-tree consistency. Numerical illustrations keep the finite interpretation stated in their captions.

Part I: The formal core

Coordinates first, decisions second

A residue decomposition changes the coordinates of a signal without changing its information. Filtering then acts between those coordinates. Approximation enters when an operator compresses a component or when a decision discards coefficients. The formal core follows these operations in that order.

Overview

This chapter develops the signal realization of the projector-resolved seed calculus. The impulse δ_0 is taken as a primitive signal seed and its shift orbit supplies the address system. Residue projectors split a sequence into exact polyphase channels, nested divisibility gives exact refinement, and every finite impulse-response operator becomes a matrix of packet transfers. The standard perfect-reconstruction condition for a filter bank is therefore an analysis–synthesis identity in the common operator language of the series. Haar analysis and every finite-depth wavelet-packet transform are treated on complex ℓ^2, together with concrete finite-signal models and a predict–update lifting step. The edge problem is separated into two statements. When a discontinuity packet is known and retained exactly, approximation error is exactly the approximation error of the remaining smooth packet. No spectral approximation of the jump is performed. When the packet is estimated, a symmetric-difference bound isolates mask error, amplitude error, and smooth-background error. Numerical studies compare global Fourier truncation, exact known-packet reconstruction, and a deliberately narrow positive-contrast estimator in one and two dimensions, including curved boundaries, corners, and multiple amplitudes. The same cyclic phase action is then placed on the Smith reflection disk. Its pullback through the Cayley transform is a Möbius action on impedance, and the two-phase case is precisely impedance–admittance inversion. Nonlinear coefficient products obey cyclic packet convolution. The accompanying formal development gives a Lean formalization of all twenty theorem-like statements under the hypotheses stated below: 225 supporting theorem targets in 23 modules. Eighteen regenerated figures, machine-readable numerical results, the 33-test original companion suite, and 36 additional exact-model and verifier tests accompany the chapter. Exact signal theorems, limited numerical estimators, classical constructions, and open recovery problems are kept distinct throughout.

Scope and claim boundary

The chapter proves exact polyphase reconstruction, exact finite filter-transfer identities, perfect reconstruction for the stated filter banks, the conditional known-edge theorem, mask and amplitude perturbation bounds, nonlinear residue-convolution laws, and the Smith-coordinate phase action. It does not claim that every discontinuity can be detected automatically, that every form of Gibbs ringing has been eliminated, or that the supplied two-class estimator is a universal segmentation method. The estimator is included as one reproducible finite example. For every finite packet refinement tree, the chapter proves existence of a globally optimal complete pruning, an exact dynamic-programming selector for additive costs, and a deterministic margin-stability certificate. Wavelets, filter banks, the Cayley transform, and the classical Smith grid are treated as classical inputs organized in the common seed–packet–refinement language.

Verification convention. "Lean-checked" refers to the explicitly scoped theorem statements and their supporting declarations, accepted by the pinned Lean kernel. It does not certify the English text automatically, the floating-point Python implementation, physical modeling choices, or the two conjectures. The manuscript-to-declaration correspondence has been reviewed manually. Section D records it result by result. Numerical replay has a separate record and does not enlarge the formal theorem scope.

1 The signal realization of the seed calculus

Let $\ell^2(\mathbb{Z};\mathbb{C})$, abbreviated $\ell^2(\mathbb{Z})$, be the Hilbert space of square-summable complex discrete signals. The inner product is conjugate-linear in its first argument. A superscript $*$ on a bounded operator denotes its Hilbert adjoint. Write δ_n for the unit impulse at address n and let the bilateral shift be

$$(Sx)[n] = x[n-1].$$

Then

$$S^n\delta_0 = \delta_n.$$

The primitive signal seed is therefore $s = \delta_0$, the address operators are the shift words S^n, and the completion is the ℓ^2 norm.

Theorem 1.1 (Impulse seed theorem). *The closed shift-generated hull of δ_0 is all of $\ell^2(\mathbb{Z})$:*

$$\boxed{\overline{\operatorname{span}\{S^n\delta_0 : n \in \mathbb{Z}\}}^{\ell^2} = \ell^2(\mathbb{Z}).}$$

Every finitely supported signal has the exact finite expression

$$x = \sum_{n\in\mathbb{Z}} x[n]S^n\delta_0.$$

Proof. The shift orbit is the standard orthonormal basis $\{\delta_n\}_{n\in\mathbb{Z}}$. For $x \in \ell^2$, the finite truncations

$$x_F = \sum_{n\in F} x[n]\delta_n, \qquad F \subset \mathbb{Z} \text{ finite},$$

satisfy $\|x - x_F\|_2^2 = \sum_{n\notin F} |x[n]|^2 \to 0$ as the finite sets exhaust $\mathbb{Z}$. Thus the unordered Hilbert-space sum $\sum_n x[n]\delta_n$ converges to x, and the finite-support statement is exact. The shift is a surjective isometry because $n \mapsto n-1$ is a bijection of $\mathbb{Z}$, with inverse shift $n \mapsto n+1$. This proves the closed-span assertion. □

This realization fixes the language used below:

$$\text{impulse seed} \longrightarrow \text{shift address} \longrightarrow \text{packet projection} \longrightarrow \text{filter transfer}$$
$$\longrightarrow \text{refinement} \longrightarrow \text{reconstruction}.$$

Remark 1.2 (Several completions). The same algebraic seed description applies to finite sequences, ℓ^1, ℓ^2, periodic sequences, tempered distributions, and multidimensional lattices. The topology decides which infinite reconstructions are legitimate. Finite-dimensional experiments in this chapter use ordinary Euclidean norms. Theorem statements specify the ambient norm when it matters.

2 Residue projectors and exact polyphase coordinates

A packet has two useful representations

The masked packet keeps its original sample addresses and sets the other phases to zero. The decimated channel stores only the retained addresses, now indexed by the coarse coordinate. They carry the same values, but their domains differ. Refinement sums belong on the original lattice. Interleaving converts the channels back to that lattice.

Fix an integer $N \geq 1$. For $0 \leq r < N$, define

$$(P_{N,r}x)[n] = \mathbf{1}_{n \equiv r \,(\mathrm{mod}\, N)} x[n].$$

The corresponding decimated channel is

$$x_r[q] = x[Nq + r].$$

Theorem 2.1 (Polyphase projector theorem)**.** *The maps $P_{N,r}$ are bounded orthogonal projectors on ℓ^2, with operator norm at most one, and satisfy*

$$P_{N,r}P_{N,s} = \delta_{rs}P_{N,r}, \qquad \sum_{r=0}^{N-1} P_{N,r} = I.$$

Consequently,

$$\boxed{x = \sum_{r=0}^{N-1} P_{N,r}x}$$

with no approximation.

Proof. Each integer belongs to exactly one residue class modulo N. Multiplying the two indicator functions gives zero for distinct classes and the same indicator for equal classes. Summing all residue indicators gives one at every address. Moreover,

$$\|P_{N,r}x\|_2^2 = \sum_{n \equiv r \,(\mathrm{mod}\, N)} |x[n]|^2 \leq \|x\|_2^2, \qquad \langle P_{N,r}x, y\rangle = \langle x, P_{N,r}y\rangle.$$

The mask is real-valued, so the inner-product identity follows term by term. Hence each projector is self-adjoint, and the ranges of different residue projectors are orthogonal. □

Define the analysis and synthesis maps

$$\mathcal{A}_N x = (x_0, \dots, x_{N-1}), \qquad \mathcal{S}_N(x_0, \dots, x_{N-1})[Nq + r] = x_r[q].$$

Corollary 2.2 (Analysis–synthesis identity)**.**

$$\boxed{\mathcal{S}_N\mathcal{A}_N = I.}$$

On the product of the channel spaces one also has $\mathcal{A}_N\mathcal{S}_N = I$. Equip this finite product with the Hilbert norm

$$\|(x_0, \dots, x_{N-1})\|_{\mathrm{ch}}^2 = \sum_{r=0}^{N-1} \|x_r\|_2^2.$$

Then $\mathcal{A}_N$ is a linear isometric equivalence and

$$\|x\|_2^2 = \sum_{r=0}^{N-1} \|x_r\|_2^2.$$

Proof. Euclidean division gives a unique pair (r, q) with $0 \le r < N$ and $n = Nq + r$, including negative n. The address maps $n \mapsto (n \bmod N, n \operatorname{div} N)$ and $(r, q) \mapsto Nq + r$ are inverse. Analysis and synthesis are reindexing by this bijection. Reindexing the nonnegative sum $\sum_n |x[n]|^2$ proves the norm identity and shows that both maps preserve square summability. In Lean the target is $\ell^2(\{0, \dots, N-1\} \times \mathbb{Z})$, which realizes exactly this Hilbert norm. □

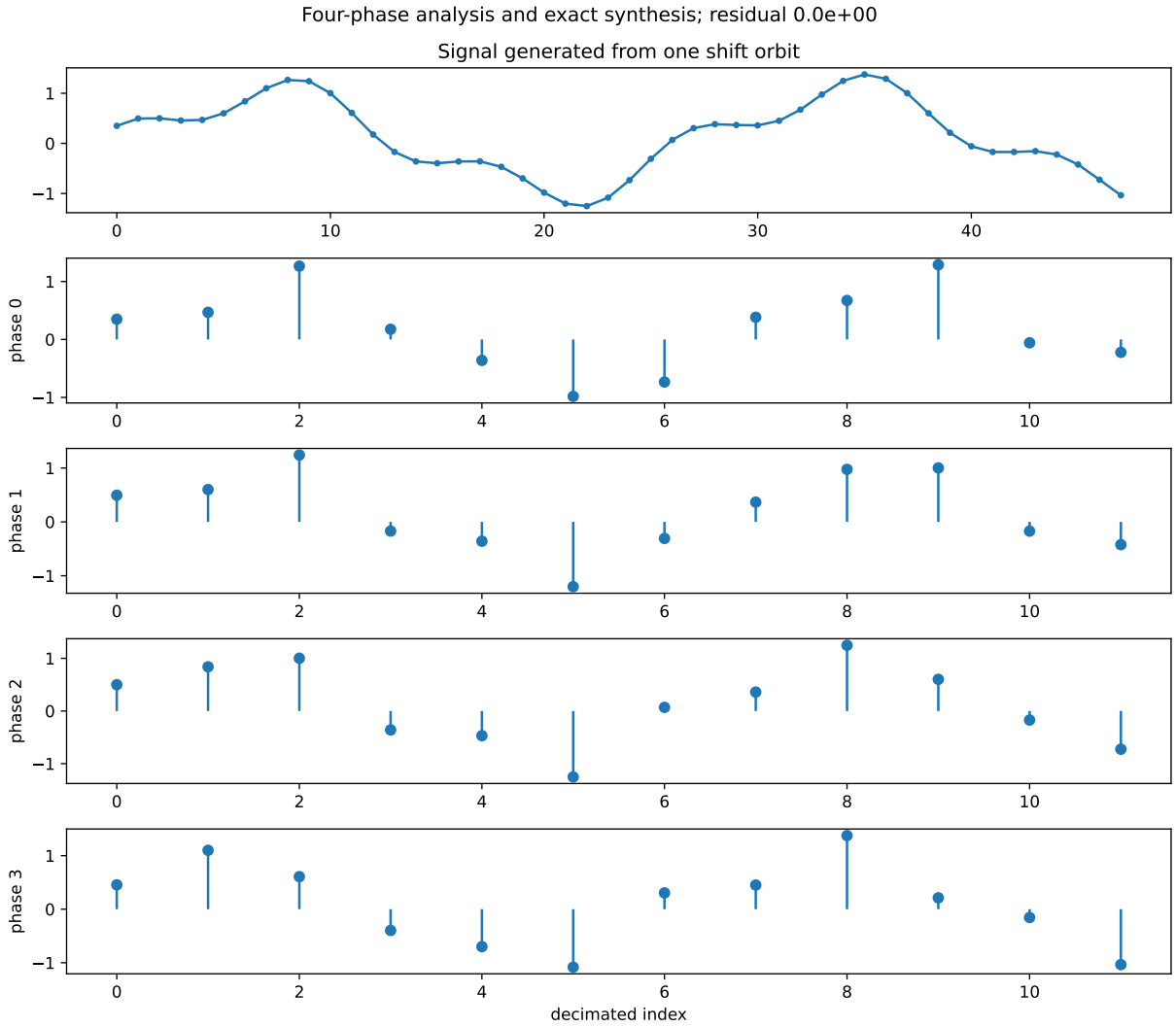


Figure 1. Four-phase analysis of a finite signal. The channels are decimated coordinate packets. Interleaving them gives the original signal exactly.

2.1 Nested refinement

Suppose $N, M \geq 1$ and $N \mid M$. Every coarse phase has a disjoint family of fine descendants:

$$\mathrm{Child}_M(N, r) = \{s : 0 \leq s < M,\ s \equiv r \pmod{N}\}.$$

Theorem 2.3 (Exact nested refinement). *For $N, M \geq 1$ and $N \mid M$,*

$$\boxed{P_{N,r} = \sum_{\substack{0 \leq s < M \\ s \equiv r \,(\mathrm{mod}\, N)}} P_{M,s}.}$$

Proof. An integer n satisfies $n \equiv r \pmod{N}$ exactly when its unique residue $s = n \bmod M$ lies among the listed descendants. The fine indicators are disjoint, so their sum is the coarse indicator. □

This identity is the signal form of the refinement law introduced in Chapter II. It is not merely a denser sampling rule. The coarse masked packet, represented on the original integer lattice, is exactly the sum of its fine masked descendants. Decimated channels at different moduli must first be placed in these common coordinates before being added.

3 Linear time-invariant operators as transfer matrices

Let h be a finite impulse response and

$$y = h * x, \qquad y[n] = \sum_{q \in \mathbb{Z}} h[q]x[n-q].$$

Resolve both input and output into N phases. For output phase k and input phase s, define the transfer kernel

$$\boxed{h_{ks}[m] = h[Nm + k - s].}$$

Out-of-range taps are understood to be zero.

Theorem 3.1 (Exact FIR polyphase transfer law). *The output channels satisfy*

$$\boxed{y_k[n] = \sum_{s=0}^{N-1} \sum_{m \in \mathbb{Z}} h_{ks}[m] x_s[n-m].}$$

For the numerical channel transform, take $z \in \mathbb{C} \setminus \{0\}$ and assume

$$\sum_{n \in \mathbb{Z}} |x_s[n] z^{-n}| < \infty \quad (0 \leq s < N).$$

Define $X_s(z) = \sum_n x_s[n] z^{-n}$. The output series then also converge absolutely, and

$$\boldsymbol{Y}(z) = \boldsymbol{H}^{(N)}(z)\boldsymbol{X}(z), \qquad H^{(N)}_{ks}(z) = \sum_m h[Nm + k - s] z^{-m}.$$

Proof. Evaluate y at address $Nn + k$ and partition every input index by its residue s:

$$\begin{aligned} y[Nn + k] &= \sum_{q} h[q]x[Nn + k - q] \\ &= \sum_{s=0}^{N-1} \sum_{m} h[Nm + k - s]x[N(n - m) + s]. \end{aligned}$$

For a fixed output phase k, the change of tap address $q = Nm + k - s$ is a bijection between $\mathbb{Z}$ and the pairs (s, m) with $0 \le s < N$: choose $s \equiv k - q \pmod N$ and then $m = (q - k + s)/N$. Thus this is a reindexing of the finite filter support, valid for every bilateral input sequence, without a convergence assumption on x.

The last factor is $x_s[n-m]$, giving the time-domain formula. Under the additional transform hypothesis, each delayed channel has transform

$$\sum_{n} x_s[n - m]z^{-n} = z^{-m} \sum_{j} x_s[j]z^{-j}.$$

Absolute summability permits this bijective reindexing, and only finitely many delayed channels are combined. Taking their finite linear combination gives the matrix formula. The time-domain theorem therefore remains valid even when no numerical bilateral transform exists. □

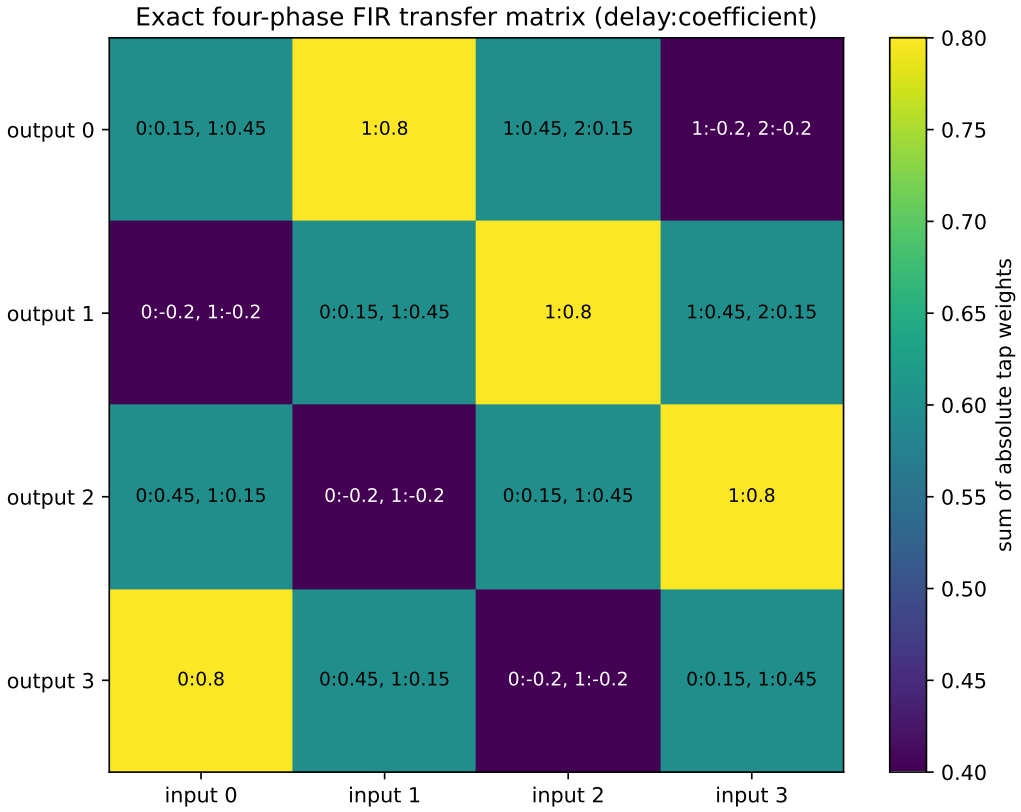


Figure 2. The exact transfer matrix for a seven-tap FIR filter in four phases. Every cell records the delays and coefficients carrying one input packet into one output packet.

Remark 3.2 (Operator-block form). If H denotes convolution by h, the same identity is

$$H = \sum_{k,s} P_{N,k} H P_{N,s}.$$

Thus the familiar polyphase matrix is exactly the transfer-block matrix of the seed calculus.

4 Analysis banks, synthesis banks, and perfect reconstruction

Read the delay in the coordinate system used

A step in one decimated channel crosses a full block of original samples. A one-sample shift also permutes the residue labels. At two phases, these operations have different matrices. The following calculation keeps the original sample delay visible instead of hiding it in a channel convention.

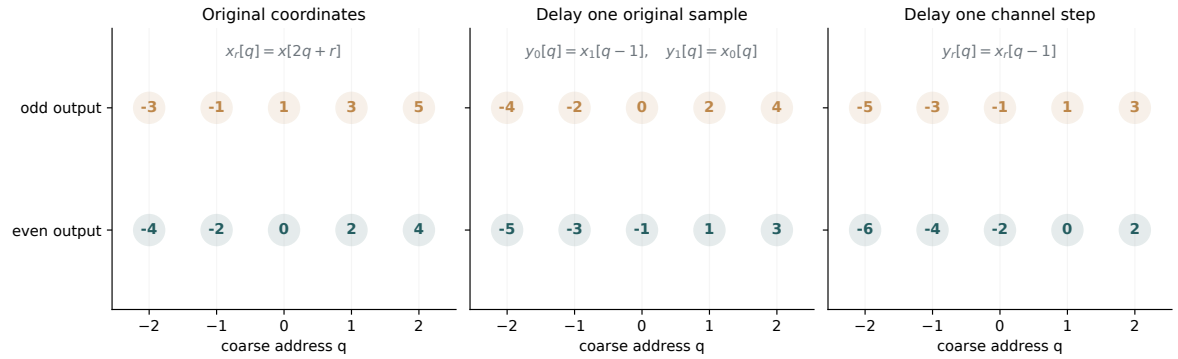


Figure 3. Exact address comparison for a two-phase split. A delay of one original sample sends the odd channel into the even channel with a coarse delay, and the even channel into the odd channel without a coarse delay. A one-step channel delay moves both original residues by two samples. All plotted values are integer sample addresses.

Work with finite impulse-response banks. Let $\mathcal{L} = \mathbb{C}[z, z^{-1}]$ be the Laurent-polynomial algebra, with the monomial z^{-m} acting on a bilateral sequence by $x[n] \mapsto x[n-m]$. Let $\boldsymbol{E} \in \mathrm{Mat}_{N\times N}(\mathcal{L})$ and $\boldsymbol{R} \in \mathrm{Mat}_{N\times N}(\mathcal{L})$ be analysis and synthesis matrices. Their composition acts on all bilateral channel sequences by finite convolutions. At a nonzero z where the input transforms are absolutely convergent, this gives

$$\widehat{\boldsymbol{X}}(z) = \boldsymbol{R}(z)\boldsymbol{E}(z)\boldsymbol{X}(z).$$

Theorem 4.1 (Perfect-reconstruction criterion). *For $d \in \mathbb{Z}$, such a bank reconstructs every input up to a delay of d* channel indices *exactly when the following Laurent-polynomial identity holds:*

$$\boxed{\boldsymbol{R}(z)\boldsymbol{E}(z) = z^{-d} I_N.}$$

If the right side is $cz^{-d}I_N$ with $c \neq 0$, reconstruction holds up to gain c and channel delay d. After interleaving, this is a delay of Nd original samples. A negative delay denotes an advance.

Proof. Finite convolution represents Laurent-polynomial multiplication: composing shifts adds their delays, and distributivity handles finite sums. Thus the matrix identity implies the asserted action.

Conversely, let $\boldsymbol{B} = \boldsymbol{R}\boldsymbol{E} - cz^{-d}I_N$ and suppose its action vanishes on every channel sequence. Insert an impulse at channel s, address zero, with every other channel zero. Output channel k reads the coefficients of the Laurent polynomial B_{ks} at all addresses. All of these coefficients vanish, so $B_{ks} = 0$ for every k, s. The representation is faithful, proving necessity as a polynomial identity rather than merely equality at one evaluation point. The displayed unit-gain case is $c = 1$. □

Original-sample delay. For an arbitrary original-sample delay $D \in \mathbb{Z}$, the output phase k has

$$y_k[n] = x_{r_k}[n + q_k], \qquad r_k = (k - D) \bmod N, \qquad q_k = (k - D) \operatorname{div} N.$$

Its channel matrix has the sole nonzero entry z^{q_k} in row k, column r_k. In particular, a one-sample delay at $N = 2$ has matrix

$$\begin{pmatrix} 0 & z^{-1} \\ 1 & 0 \end{pmatrix},$$

whereas $z^{-1}I_2$ delays the original signal by two samples. The general address identity is valid for negative sample addresses as well.

4.1 The Haar bank

The Haar analysis matrix is

$$\boldsymbol{E}_{\mathrm{H}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

It is orthogonal, so $\boldsymbol{R}_{\mathrm{H}} = \boldsymbol{E}_{\mathrm{H}}^{\mathsf{T}}$ and

$$\boldsymbol{R}_{\mathrm{H}}\boldsymbol{E}_{\mathrm{H}} = I_2.$$

The channels are

$$a[n] = \frac{x[2n] + x[2n+1]}{\sqrt{2}}, \qquad d[n] = \frac{x[2n] - x[2n+1]}{\sqrt{2}}.$$

Corollary 4.2 (Haar energy identity)**.**

$$\boxed{\|x\|_2^2 = \|a\|_2^2 + \|d\|_2^2.}$$

Proof. For any $u, v \in \mathbb{C}$, the parallelogram identity gives

$$\left|\frac{u+v}{\sqrt{2}}\right|^2 + \left|\frac{u-v}{\sqrt{2}}\right|^2 = |u|^2 + |v|^2.$$

Apply this to $u = x[2n]$, $v = x[2n+1]$ and sum over $n \in \mathbb{Z}$. Each summand is nonnegative. The right side is $\|x\|_2^2$ by the two-channel address equivalence. This also proves that the literal low and detail channels belong to ℓ^2. The real Haar matrix is unitary over $\mathbb{C}$, so the finite-signal statement is the same calculation with a finite sum. □

4.2 Predict–update lifting

Split x into even and odd channels e, o. Define

$$d = o - e, \qquad s = e + \frac{1}{2}d.$$

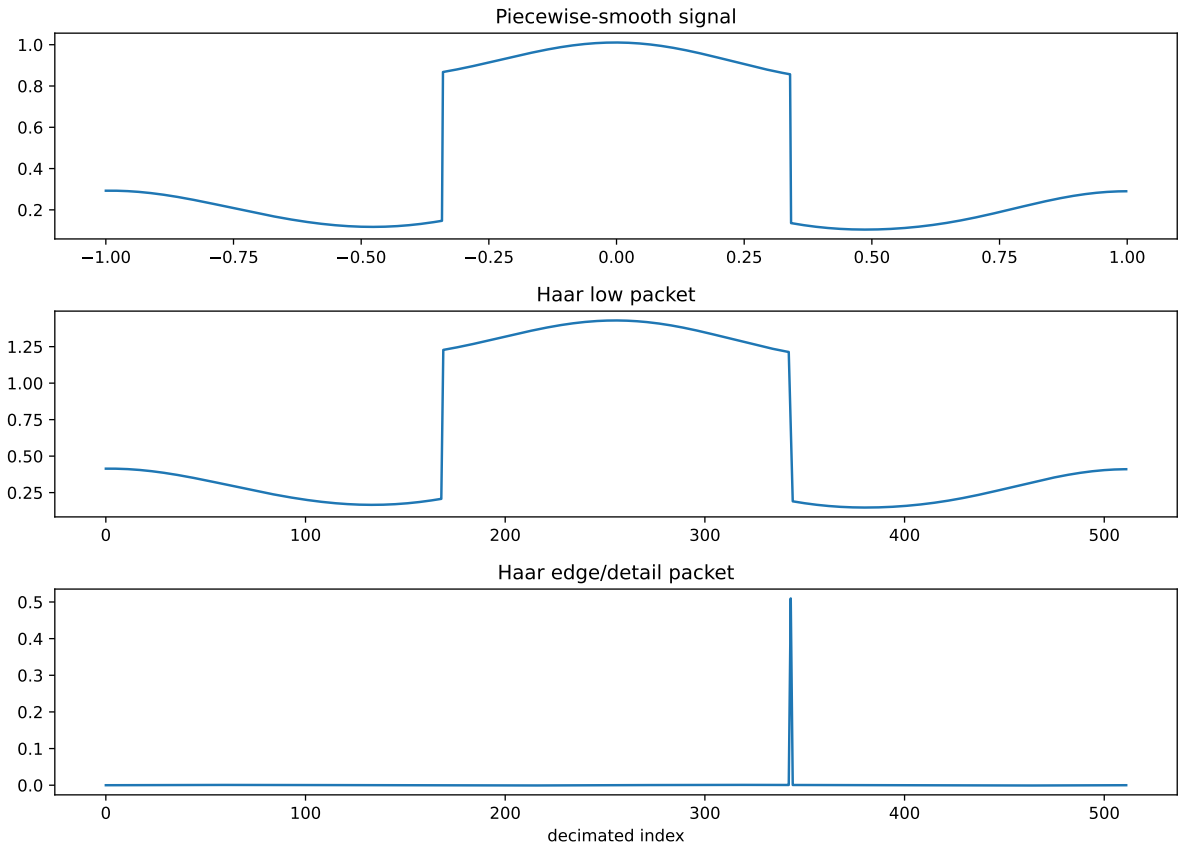


Figure 4. A piecewise-smooth signal and its first Haar low/detail packets. The low channel records local averages. The detail channel records adjacent-sample differences, including pairs that cross a jump.

The inverse is

$$e = s - \frac{1}{2}d, \qquad o = d + e.$$

Theorem 4.3 (Exact lifting reconstruction). *The predict–update map $(e, o) \mapsto (s, d)$ is invertible with determinant one and the displayed inverse. Consequently the associated finite transform reconstructs every even-length signal exactly.*

Proof. The forward matrix is

$$\begin{pmatrix} s \\ d \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} e \\ o \end{pmatrix},$$

whose determinant is one. Direct substitution gives the inverse formulas. □

5 Wavelet packets as addressed refinements

On $\ell^2(\mathbb{Z}; \mathbb{C})$, apply the Haar analysis to both children at each stage. For a finite input, assume its length is divisible by 2^ℓ when a depth-ℓ transform is used. A binary word $\alpha \in \{0, 1\}^\ell$ addresses a depth-ℓ packet, denoted $W_{\ell,\alpha}x$. Appending 0 or 1 selects the low or high child.

Theorem 5.1 (Wavelet-packet reconstruction and energy). *At every finite depth ℓ,*

$$\boxed{x = \sum_{\alpha \in \{0,1\}^\ell} W_{\ell,\alpha}^* W_{\ell,\alpha} x,}$$

and

$$\boxed{\|x\|_2^2 = \sum_{\alpha \in \{0,1\}^\ell} \|W_{\ell,\alpha} x\|_2^2.}$$

Proof. Write H_0, H_1 for the literal low and detail operators. The preceding energy identity implies $\|H_b x\|_2 \le \|x\|_2$, so both have bounded Hilbert adjoints. At depth zero the only packet operator is the identity. If the energy identity holds at depth ℓ, apply the one-level identity to each $W_{\ell,\alpha}x$ and sum. This proves it at depth $\ell + 1$.

Set $A = \sum_\alpha W_{\ell,\alpha}^* W_{\ell,\alpha}$. This is a finite sum of bounded operators. The energy result says $\langle x, Ax\rangle = \langle x, x\rangle$ for every x. Complex polarization, applied also to $x + y$ and $x + \mathrm{i}y$, gives $\langle y, (A - I)x\rangle = 0$ for all x, y, hence $A = I$. Applying the operator identity to x proves synthesis. This argument uses the actual Hilbert adjoints of the decimation operators, not a separately postulated inverse symbol. □

Classical Input 5.2 (Wavelet boundary). The existence and perfect-reconstruction theory of wavelets and filter banks are classical. Their role here is structural: they provide a fully developed realization in which seed generation, packet addressing, refinement, transfer blocks, and exact synthesis can all be read literally in the common operator language.

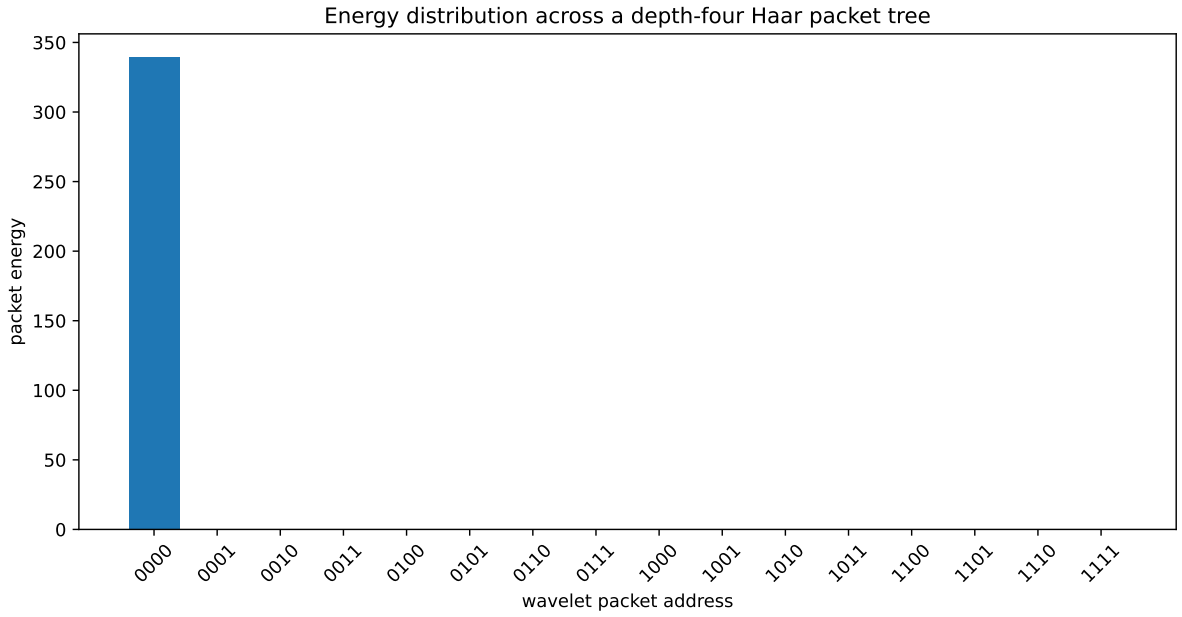


Figure 5. Energy of the sixteen depth-four Haar packet addresses for the test signal. The packet address is an operator word in the common seed grammar.

6 Known edge packets and spectral reconstruction

Retention changes the approximation target

The retained packet is supplied as part of the model. It passes through synthesis exactly. The approximation operator acts on the smooth remainder, so its error can be studied without forcing a global smooth basis to reproduce the jump. Finding that packet from observations is a separate recovery problem.

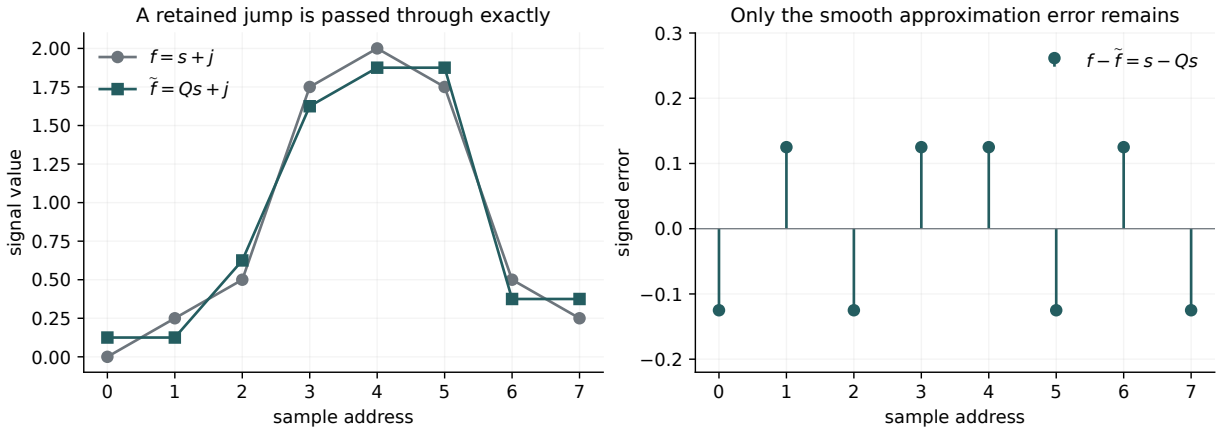


Figure 6. An exact rational eight-coordinate example of retained packet cancellation. The supplied jump packet is kept, while the linear operator averages adjacent pairs in the smooth component. The two error vectors agree coordinate by coordinate. This model illustrates cancellation and does not test packet detection or a Fourier convergence rate.

The central distinction is between approximating a jump globally and retaining it as its own packet. Let

$$f = s + j$$

where s belongs to a smooth approximation class and j is a known discontinuity packet. Let Q_m be any approximation operator applied only to s, and define

$$\tilde{f}_m = Q_m s + j.$$

Theorem 6.1 (Exact known-edge theorem)**.** *For every norm for which the expressions are defined,*

$$\boxed{f - \tilde{f}_m = s - Q_m s, \qquad \|f - \tilde{f}_m\| = \|s - Q_m s\|.}$$

No approximation of j occurs.

Proof. Subtract:

$$f - \tilde{f}_m = (s + j) - (Q_m s + j) = s - Q_m s.$$

Taking the norm gives the second identity. □

Corollary 6.2 (Several exact edge packets)**.** *If*

$$f = s + \sum_{q=1}^{M} j_q$$

and every j_q is retained exactly, then

$$\left\| f - \left(Q_m s + \sum_{q} j_q \right) \right\| = \|s - Q_m s\|.$$

Curved edges, corners, and different jump amplitudes do not alter the identity.

Proof. The finite sum $j = \sum_{q=1}^{M} j_q$ is a single retained packet. Applying the preceding cancellation identity gives the result. The empty family $M = 0$ is included with sum zero. □

Boundary 6.3 (What the theorem does and does not say). The theorem explains why global ringing caused by spectrally approximating a known jump is absent from the reconstructed jump packet. It does not discover the packet. If the packet is wrong, its error is reconstructed exactly as well. Nor does the theorem prevent unrelated truncation errors in the smooth packet, domain-boundary errors, noise, or model mismatch.

7 Mask, amplitude, and smooth-background error

Let (X, Σ, μ) be a measure space, with measurable masks $\Omega, \widehat{\Omega} \subseteq X$ and scalar amplitudes $a, \hat{a} \in \mathbb{C}$ (real amplitudes are a special case). Suppose a scalar jump packet is

$$j = a\mathbf{1}_{\Omega},$$

and an estimated packet is

$$\hat{j} = \hat{a}\mathbf{1}_{\widehat{\Omega}}.$$

Theorem 7.1 (Packet perturbation bound). *For $1 \le p < \infty$, interpreted first as an inequality of extended L^p seminorms,*

$$\boxed{\|j - \hat{j}\|_{L^p} \le |a - \hat{a}|\,\mu(\Omega)^{1/p} + |\hat{a}|\,\mu(\Omega \triangle \widehat{\Omega})^{1/p}.}$$

If $\mu(\Omega)$ and $\mu(\Omega \triangle \widehat{\Omega})$ are finite, all terms in this displayed jump bound are finite and it is an ordinary real-valued L^p inequality. For the pointwise supremum, and consequently also the essential L^∞ seminorm,

$$\|j - \hat{j}\|_\infty \le |a - \hat{a}| + |\hat{a}|\,\mathbf{1}_{\Omega \triangle \widehat{\Omega} \neq \emptyset}.$$

Proof. Write

$$a\mathbf{1}_\Omega - \hat{a}\mathbf{1}_{\widehat{\Omega}} = (a - \hat{a})\mathbf{1}_\Omega + \hat{a}(\mathbf{1}_\Omega - \mathbf{1}_{\widehat{\Omega}}).$$

Apply the triangle inequality and use

$$\|\mathbf{1}_\Omega - \mathbf{1}_{\widehat{\Omega}}\|_{L^p} = \mu(\Omega \triangle \widehat{\Omega})^{1/p}.$$

For the supremum estimate, $|\mathbf{1}_\Omega - \mathbf{1}_{\widehat{\Omega}}| \le 1$ pointwise, and this function vanishes everywhere when the symmetric difference is empty. The same pointwise bound also bounds the essential supremum. Nonempty null symmetric differences need not contribute to the essential norm. The stated bound remains valid in that case and need not be sharp. Measurability justifies the L^p triangle inequality and the indicator norm formula. □

If $s, \hat{s}$ are measurable smooth-background models, the same triangle inequality gives, for finite p,

$$\boxed{\|f - (\hat{s} + \hat{j})\|_p \le \|s - \hat{s}\|_p + |a - \hat{a}|\mu(\Omega)^{1/p} + |\hat{a}|\mu(\Omega \triangle \widehat{\Omega})^{1/p}.}$$

This is the correct separation of errors. Detection error is not hidden inside a generic reconstruction score.

Conjecture 7.2 (Stable edge-packet recovery)**.** For specified piecewise-smooth classes, contrast bounds, separation conditions, and noise models, there should exist a multiscale packet estimator whose symmetric-difference error and amplitude error converge at rates controlled independently of the smooth approximation rate. A theorem of this kind would turn the conditional known-edge result into a stable data-driven reconstruction theorem. No universal version is asserted here.

8 One-dimensional numerical study

The one-dimensional signal contains a smooth non-bandlimited background and a positive rectangular packet. Three reconstructions are compared:

(i) global Fourier truncation of the complete signal;

(ii) Fourier truncation of the known smooth packet plus the exact jump packet;

(iii) a two-class estimated packet followed by harmonic background filling, spectral smoothing, and jump-amplitude estimation.

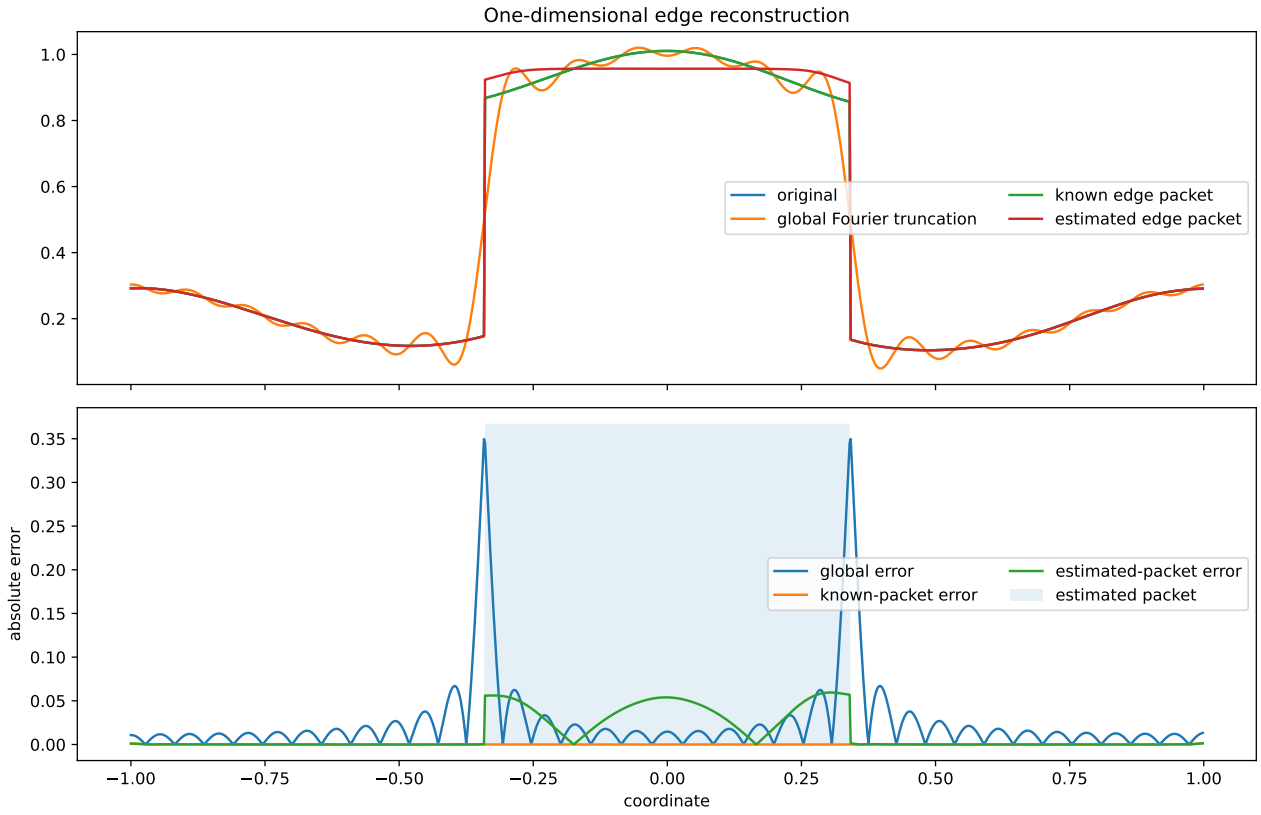


Figure 7. One-dimensional reconstruction and absolute error. The known packet removes the edge from the global spectral approximation. The estimated packet is visibly better than the global reconstruction in this limited test but retains smooth-background estimation error.

Numerical Certificate 8.1 (One-dimensional errors). For 1024 samples on $[-1, 1)$ and Fourier cutoff $m = 18$, the retained frequency indices are $-18, \dots, 18$: 37 bins including the zero-frequency bin. The measured values were

method	RMSE	maximum absolute error
global Fourier truncation	5.4023×10^{-2}	3.4968×10^{-1}
known edge packet	1.3692×10^{-4}	1.4503×10^{-3}
estimated edge packet	2.4029×10^{-2}	5.9530×10^{-2}

The simple estimator recovered the binary mask exactly in this deliberately separated two-class example. That result is a property of this test, not a universal guarantee.

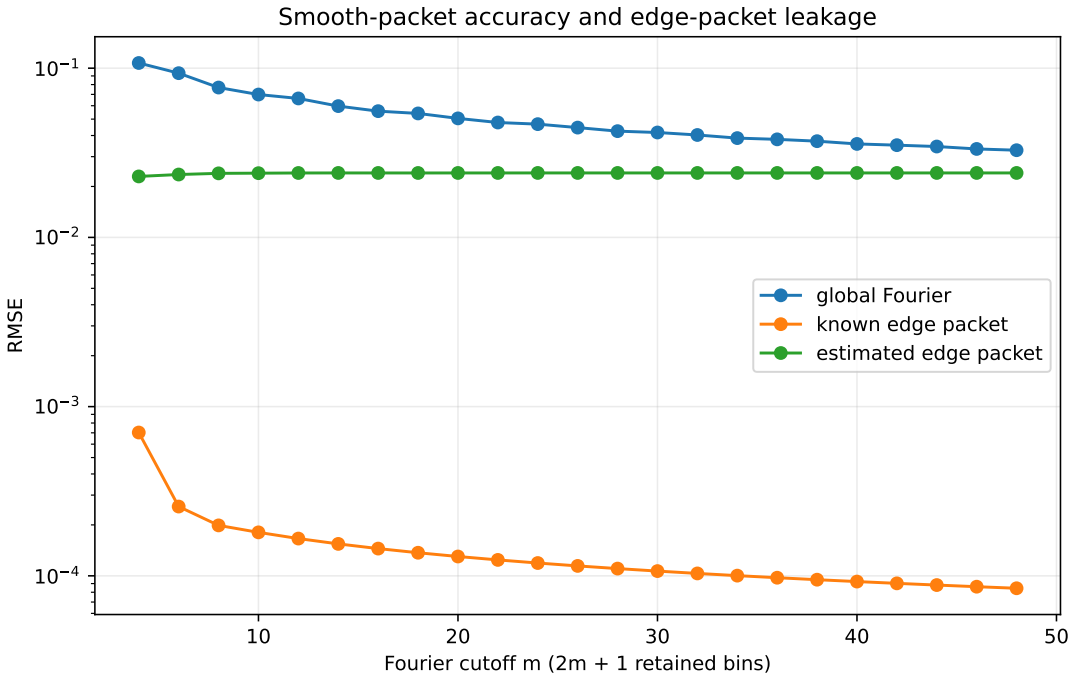


Figure 8. Error as spectral resolution increases. The known packet follows only the smooth approximation error. The estimated reconstruction reaches a floor determined by background and amplitude estimation.

9 Two-dimensional edge packets

The first image consists of a smooth background and one sharp disk. The same three logical levels must remain separate:

(a) the exact theorem using the known decomposition;

(b) a limited estimator designed for a positive-contrast two-class packet;

(c) the open problem of recovering arbitrary edge packets under noise and model ambiguity.

Numerical Certificate 9.1 (Two-dimensional errors). For the 192×192 disk experiment, a square Fourier mask retains indices $|k_x|, |k_y| \leq 16$, or $33^2 = 1089$ bins. The reported errors are

method	full-image RMSE	boundary-band RMSE
global Fourier truncation	3.7643×10^{-2}	1.8541×10^{-1}
known edge packet	6.4752×10^{-5}	1.2709×10^{-5}
estimated edge packet	3.4043×10^{-3}	8.8834×10^{-3}

The mask intersection-over-union was one in this positive-contrast synthetic case. The amplitude was estimated rather than inserted from the generating model.

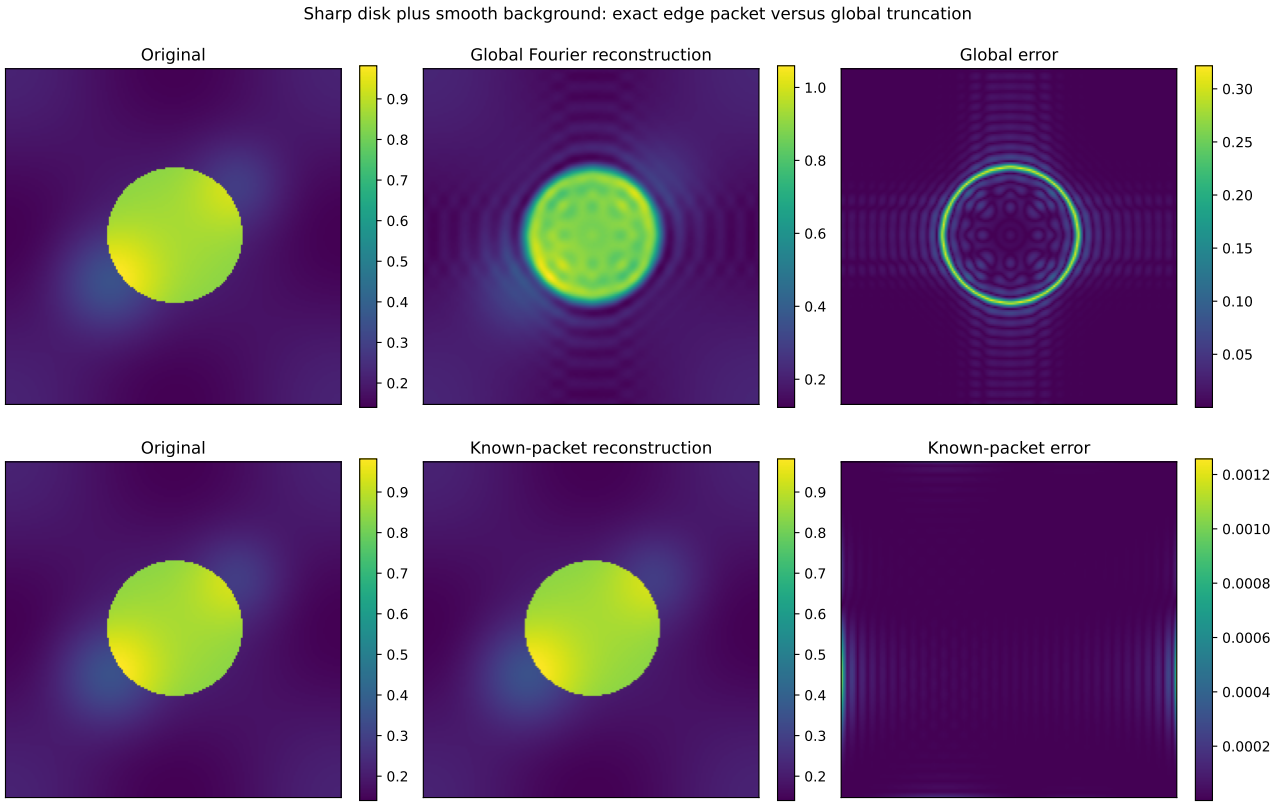


Figure 9. A sharp disk on a smooth background. Global Fourier truncation produces a bright boundary halo and oscillatory interior and exterior error. Retaining the exact edge packet leaves only the smooth-background approximation error.

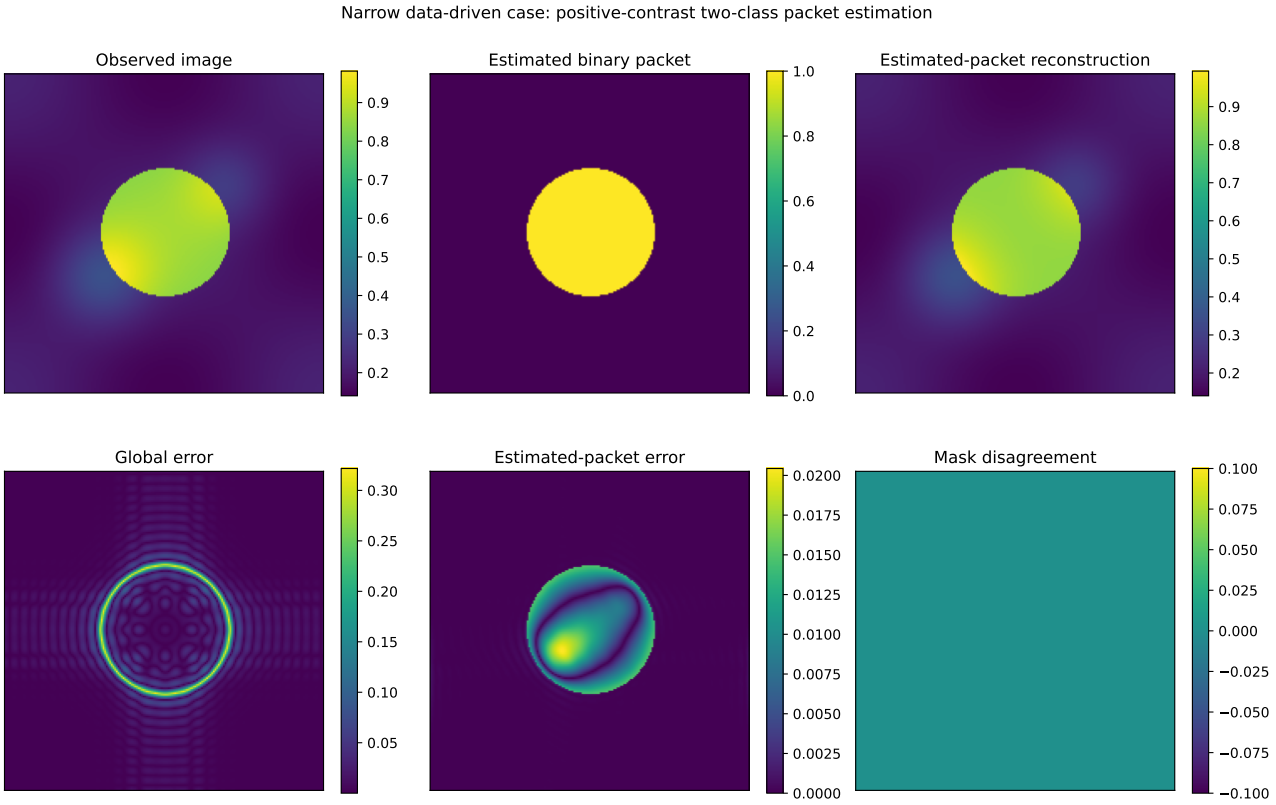


Figure 10. The limited data-driven experiment. The observed image alone is thresholded into two intensity classes, cleaned topologically, harmonically filled, spectrally smoothed, and recombined with an estimated jump.

9.1 Curves, corners, and multiple amplitudes

The exact theorem does not depend on the shape of the known edge packet. The next experiment contains a disk, a square, and a diamond with different jump amplitudes.

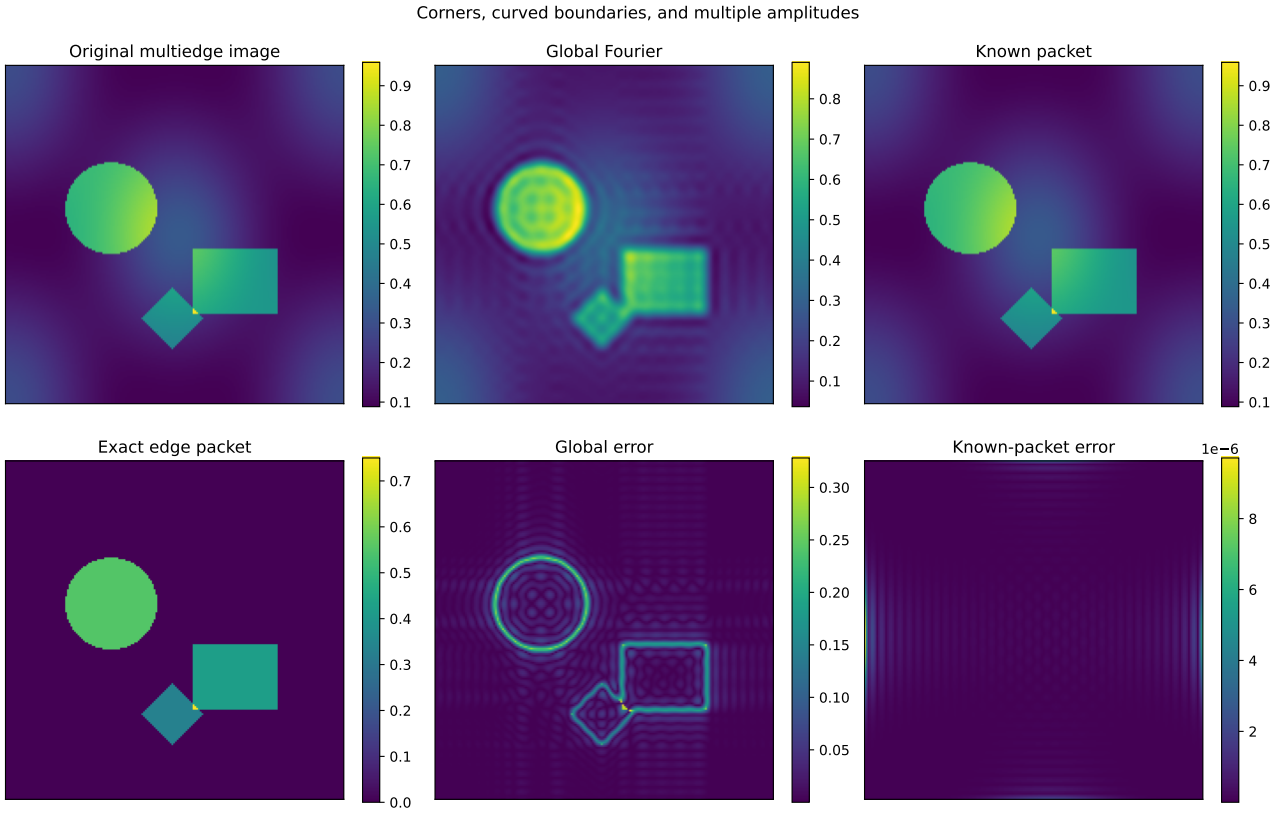


Figure 11. Known-packet stress test with curved boundaries, corners, overlap, and different jump amplitudes. The exact packet is not a binary mask multiplied by one amplitude. It is the complete piecewise packet.

Numerical Certificate 9.2 (Multiedge errors). The global reconstruction had RMSE 3.3740×10^{-2} and maximum error 3.2807×10^{-1}. The known-packet reconstruction had only the remaining smooth-background error. The experiment checks the finite implementation of the theorem for a composite packet. It is not a data-driven multiregion segmentation result.

10 Sparse wavelet packets and global Fourier modes

A sharp edge is spatially local but spectrally global. Wavelets often provide a more economical coordinate system once enough localized coefficients are retained. To keep the comparison explicit, the numerical study subtracts the signal mean, keeps the largest coefficients in a fixed depth-five Haar packet basis, and compares them with the same number of largest-magnitude discrete Fourier coefficients. These are coefficient-count budgets, not equal bit-storage budgets. The Fourier reconstruction uses the real part of the inverse transform. For the five reported budgets, the selected nonzero-frequency bins occur in conjugate pairs, so this operation removes only floating-point imaginary residuals.

Boundary 10.1 (Basis selection). This table compares two fixed finite coordinate systems for one signal. It does not prove that Haar packets are optimal, that Fourier is inferior on smooth or oscillatory data, or

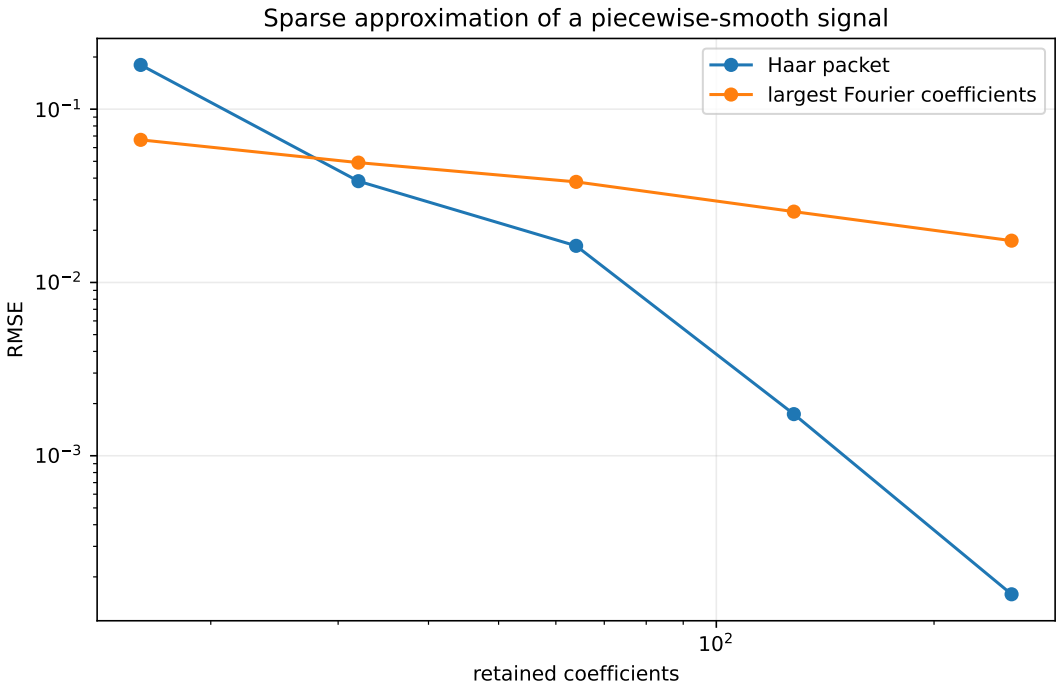


Figure 12. Sparse approximation of the test signal. With very few coefficients the global Fourier basis is initially better. After the edge packets are resolved, the Haar packet error decays much faster in this example.

Table 1. Sparse approximation errors for the one-dimensional test signal.

retained coefficients	Haar packet RMSE	Fourier RMSE
16	1.7999×10^{-1}	6.6522×10^{-2}
32	3.8414×10^{-2}	4.9153×10^{-2}
64	1.6267×10^{-2}	3.8056×10^{-2}
128	1.7387×10^{-3}	2.5631×10^{-2}
256	1.5870×10^{-4}	1.7424×10^{-2}

that a single packet tree is best for every edge geometry. The seed calculus records both systems with the same analysis, transfer, refinement, and synthesis maps.

11 Finite adaptive packet trees

Three levels of a finite tree decision

First choose a complete pruning, so every terminal path is represented once and synthesis remains exact. Then assign a finite objective to those feasible choices. Additivity permits the local dynamic recursion. Finally, a positive cost gap protects the selected pruning against a declared perturbation envelope. Each step uses a different hypothesis.

The preceding sparse comparison used one fixed depth-five packet basis. A refinement tree permits a stronger choice: different branches may stop at different depths while the selected packets still form one complete coordinate system.

Definition 11.1 (Complete packet pruning)**.** Let $\mathcal{T}$ be a finite rooted refinement tree. Every node v carries a projector P_v, the root satisfies $P_\emptyset = I$, and every internal node satisfies

$$P_v = \sum_{u \in \mathrm{ch}(v)} P_u, \qquad P_u P_w = 0 \quad (u \neq w \text{ in } \mathrm{ch}(v)).$$

A set Λ of nodes is a complete pruning if it is an antichain and every root-to-terminal path meets Λ exactly once.

Theorem 11.2 (Exact synthesis from every complete pruning)**.** *For every complete pruning Λ,*

$$\boxed{\sum_{v \in \Lambda} P_v = I,} \qquad \boxed{x = \sum_{v \in \Lambda} P_v x.}$$

Thus changing the selected tree changes the coordinates but does not introduce reconstruction error.

Proof. Starting from the root identity $P_\emptyset = I$, replace every unselected internal node by the exact sum of its children. If a complete pruning contains the root it contains no other node. Otherwise its restriction to each child subtree is itself a complete pruning: every terminal path through that child must be met exactly once. Induction on the finite tree therefore justifies the substitution and its termination precisely at Λ. Child subtrees have distinct node addresses, so their selected-node sets form a disjoint union. The sum identity uses only root identity and parent-to-child conservation. Sibling orthogonality gives the additional interpretation as nonoverlapping projector packets. Applying the resulting identity to x gives the synthesis formula. □

Let y denote the observed data. For a complete pruning Λ, consider any finite criterion

$$J_y(\Lambda) = R_y(\Lambda) + \eta L_y(\Lambda) + \tau K(\Lambda),$$

where R_y is a representation cost, L_y is a cross-packet leakage cost, and K is a tree-complexity penalty. Here R_y, L_y, K take finite real values, and η, τ are fixed real weights (usually nonnegative in applications). No additivity is needed for the first theorem.

Theorem 11.3 (Finite globally optimal stable selector)**.** *Let $\mathfrak{P}(\mathcal{T})$ be the finite set of complete prunings of $\mathcal{T}$.*

(i) A global minimizer

$$\Lambda_y^* \in \operatorname*{argmin}_{\Lambda \in \mathfrak{P}(\mathcal{T})} J_y(\Lambda)$$

always exists.

(ii) If the minimizer is unique and at least one competing pruning exists, its gap

$$\Gamma_y = \min_{\Lambda \neq \Lambda_y^*} \left(J_y(\Lambda) - J_y(\Lambda_y^*)\right)$$

is attained and strictly positive. If the pruning set is a singleton, its only element minimizes every criterion. No real-valued minimum over an empty competitor set is introduced.

(iii) In the case with competitors, if new data y' satisfy

$$\sup_{\Lambda \in \mathfrak{P}(\mathcal{T})} |J_{y'}(\Lambda) - J_y(\Lambda)| < \frac{\Gamma_y}{2},$$

then $\Lambda_{y'}^* = \Lambda_y^*$.

(iv) Every selected minimizer preserves the exact synthesis identity of the preceding theorem.

Proof. The root-only pruning exists, and every pruning is a subset of the finite node set, so the feasible set is nonempty and finite. Thus the minimum is attained. If competitors exist, their finitely many differences from a unique optimum are strictly positive, so their minimum is positive and is achieved by a competitor. In the singleton case there is no alternative selector under any objective change. In the remaining case, for any competing pruning $\Lambda \neq \Lambda_y^*$,

$$\begin{aligned} J_{y'}(\Lambda) - J_{y'}(\Lambda_y^*) &\geq J_y(\Lambda) - J_y(\Lambda_y^*) - 2 \sup_{\Xi} |J_{y'}(\Xi) - J_y(\Xi)| \\ &> \Gamma_y - \Gamma_y = 0. \end{aligned}$$

Hence the same pruning remains uniquely optimal. Exact synthesis depends only on completeness of the pruning, not on the cost used to select it. □

For tree-local costs, the optimum can be found without enumerating all prunings.

Theorem 11.4 (Dynamic-programming best packet basis)**.** *Suppose*

$$J_y(\Lambda) = \sum_{v \in \Lambda} c_v(y).$$

For a node v, define recursively

$$V_y(v) = \begin{cases} c_v(y), & v \text{ terminal}, \\ \min\left\{ c_v(y), \sum_{u \in \mathrm{ch}(v)} V_y(u) \right\}, & v \text{ internal}. \end{cases}$$

Stop at v when $c_v(y) \le \sum_u V_y(u)$, including equality, and refine otherwise. This deterministic rule produces a globally optimal complete pruning. With the node costs supplied and child values cached, the evaluator uses one comparison per internal node, one visit per node, and at most $|\mathcal{T}| - 1$ additions. Its arithmetic-and-visit count is therefore at most $3|\mathcal{T}|$. This is a unit-cost statement about the specified evaluator. It excludes evaluation of the local costs, bit complexity of scalar arithmetic, and materialization or sorting of packet-address strings.

Proof. Proceed upward from the terminal nodes. Every complete pruning below an internal node v has exactly two possible forms: it either stops at v, with cost $c_v(y)$, or it refines and is the union of complete child prunings, whose minimum total cost is the sum of the already computed child minima. The recursion therefore gives the exact minimum on every subtree, in particular at the root. This characterization also proves both directions of the correspondence between recursive cuts and complete covering antichains, for arbitrary positive finite branching, including unary branches. Summing cached child values charges at most one addition per child edge. A finite rooted tree has $|\mathcal{T}| - 1$ such edges. The number of internal nodes is at most $|\mathcal{T}|$, which gives the stated work bound. Ties do not affect optimality, but the stop-on-ties rule makes the selected pruning unambiguous. □

Corollary 11.5 (Local perturbation certificate)**.** *Assume*

$$|c_v(y') - c_v(y)| \le \varepsilon_v.$$

Define

$$E(v) = \begin{cases} \varepsilon_v, & v \text{ terminal}, \\ \max\left\{ \varepsilon_v, \sum_{u \in \mathrm{ch}(v)} E(u) \right\}, & v \text{ internal}. \end{cases}$$

For the complete prunings $\mathfrak{P}_v$ below v, the envelope is exactly

$$E(v) = \max_{\Lambda \in \mathfrak{P}_v} \sum_{a \in \Lambda} \varepsilon_a.$$

The maximum is attained. Every complete-pruning cost changes by at most $E(\emptyset)$, and the two globally optimal values differ by at most this quantity. Consequently, for a unique optimum with competitors,

$$\boxed{\Gamma_y > 2E(\emptyset)}$$

is a deterministic certificate that the globally optimal pruning is unchanged. A singleton pruning set is stable without a gap condition. Assume the random node costs and the joint local-error event are measurable. If a noise model supplies all *local bounds simultaneously on one event of probability at least $1 - \delta$, and the displayed strict-gap condition holds, the same selector is recovered with probability at least $1 - \delta$. Individual nodewise probabilities alone do not give this joint confidence without an additional argument, such as a union bound.*

Proof. For any pruning of the subtree rooted at v, either v itself is selected, producing perturbation at most ε_v, or the pruning is the disjoint union of child prunings, producing perturbation at most the sum of the child envelopes. The same decomposition applied to the nonnegative weights ε_a proves equality with the maximum weight: at each node choose either the stop option or a maximizing pruning in each child, whichever gives the larger total. Finiteness ensures these choices attain their maxima. Evaluating each cost at a minimizer of the other cost bounds the difference of the two optimal values by the uniform envelope. The selector statement follows from the strict-gap theorem. The finite deterministic comparisons defining the selector make its preservation event measurable when the random node costs are measurable. The joint local-error event is contained in the selector-preservation event, so monotonicity of probability proves the probabilistic assertion. □

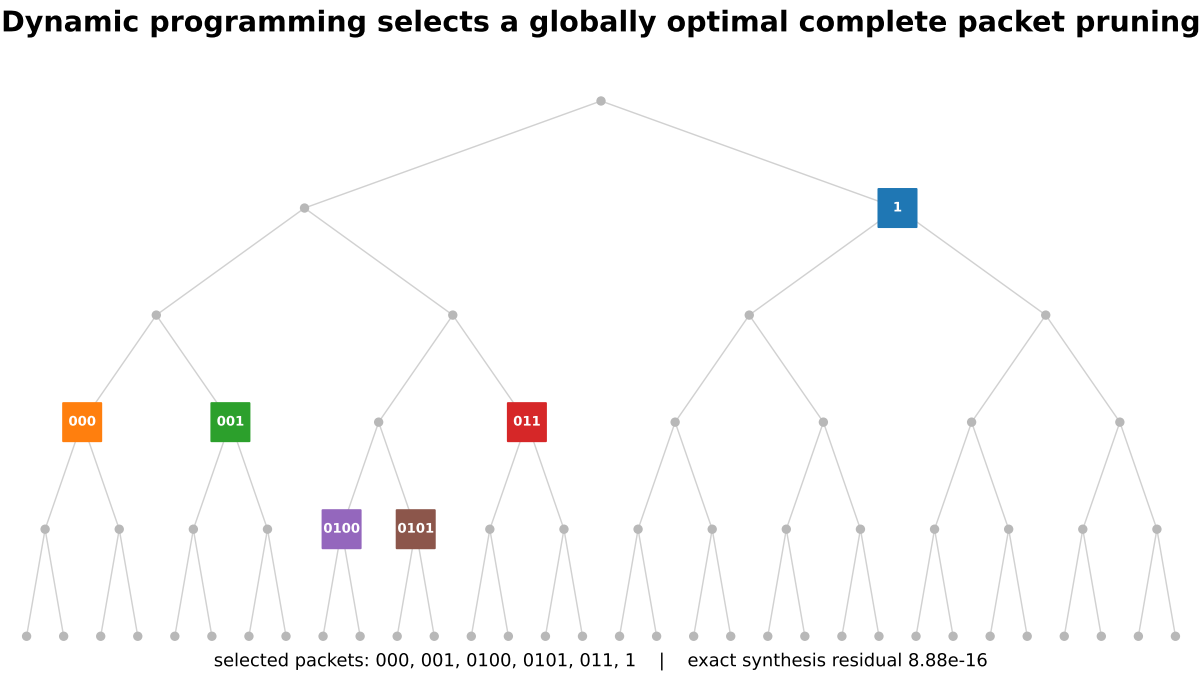


Figure 13. A globally optimal mixed-depth Haar packet pruning. The selected antichain covers the full tree, so its synthesis is exact even though different branches stop at different depths.

Numerical Certificate 11.6 (Adaptive-tree optimum). For the depth-five example, the complete binary tree has $458{,}330$ admissible prunings. Dynamic programming and exhaustive enumeration returned the same minimum with residual 0. The selected pruning was

$$\{000, 001, 0100, 0101, 011, 1\},$$

and mixed-depth synthesis reconstructed the signal with maximum residual 8.88×10^{-16}. The optimum gap was 3.1909×10^{-4}, while twice the declared perturbation envelope was 6.4×10^{-5}, so the finite selector is certified stable.

Boundary 11.7 (Finite versus asymptotic selection). The theorem resolves the finite adaptive-tree existence, exact-synthesis, global-optimality, and margin-stability problem. It does not by itself prove minimax rates, statistical consistency on an infinite tree, or a universal choice of representation and leakage costs for every signal class.

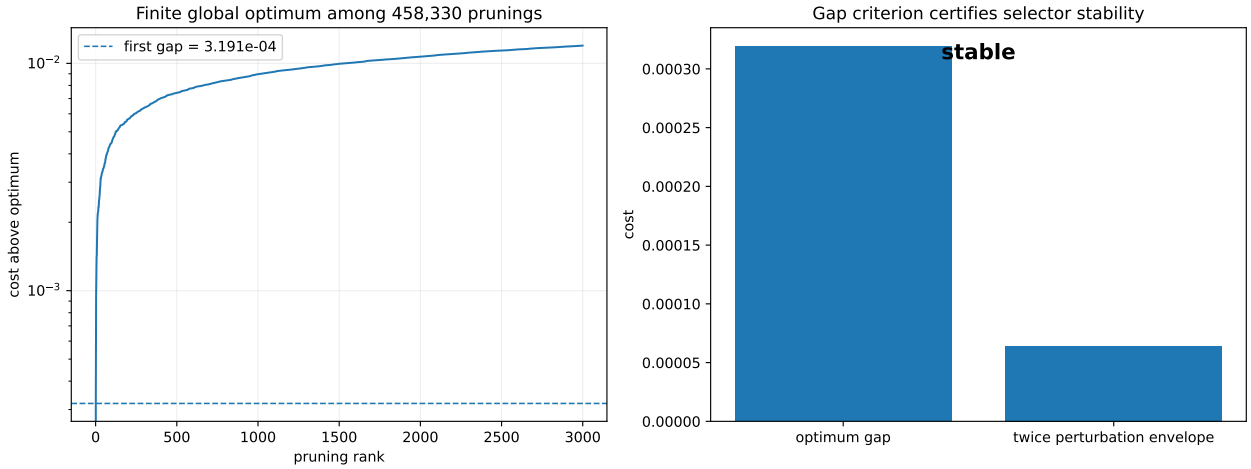


Figure 14. The dynamic-programming optimum agrees with exhaustive enumeration. A positive gap larger than twice the declared perturbation envelope certifies that the selected tree is stable.

12 Nonlinear signal packets

Linear time invariance is not required for exact packet coordinates. Let A and B be formal power series over a commutative coefficient ring (in particular $\mathbb{C}$). Set

$$A(z) = \sum_{n \geq 0} a_n z^n, \qquad A_i = P_{N,i} A.$$

All phase indices range from 0 to $N - 1$. For another representation $B(z)$, multiplication gives a nonlinear or bilinear interaction.

Theorem 12.1 (Cyclic product law)**.**

$$\boxed{P_{N,k}(AB) = \sum_{i+j \equiv k \,(\mathrm{mod}\, N)} A_i B_j.}$$

For $q \in \mathbb{N}$, the qth-power identity is

$$P_{N,k}(A^q) = \sum_{i_1 + \cdots + i_q \equiv k} A_{i_1} \cdots A_{i_q}.$$

Proof. Every coefficient in $A_i B_j$ has degree congruent to $i + j$ modulo N. At any fixed degree n, the Cauchy coefficient sum is finite. Sorting these terms by the residues of their two indices gives the first identity coefficient by coefficient. For the power identity, use induction on q and the finite product law. At $q = 0$, the index set contains the single empty tuple, its product is 1, and its residue sum is 0. Hence the right side is 1 for $k = 0$ and 0 otherwise, as required. No convergence is needed for these formal identities. For analytic power series, the same identities hold wherever the series are absolutely convergent and their Cauchy products may be evaluated. □

For a finite nonlinear shift operator

$$F(x)[n] = x[n] + \alpha x[n] x[n-1],$$

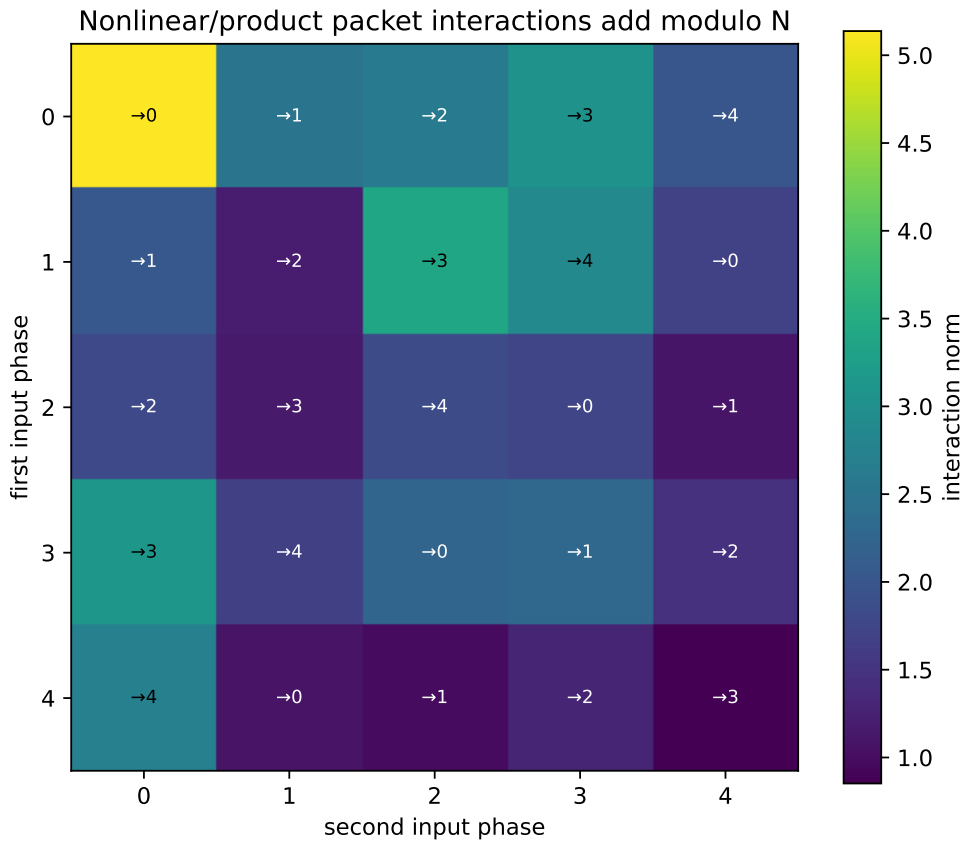


Figure 15. Norms of pairwise product interactions for five residue phases. The label in each cell gives the output phase $(i + j) \bmod 5$.

the exact packet coordinates are simply

$$F_r((x_s)_s) = P_{N,r} F\left(\sum_s P_{N,s} x\right).$$

Analysis followed by this packet system and synthesis reproduces the direct nonlinear output exactly. The numerical residual in the companion implementation was zero to machine representation.

13 Smith coordinates as cyclic signal coordinates

Let $Z_0 > 0$ be a reference impedance and

$$z = \frac{Z}{Z_0} = r + \mathrm{i}x.$$

The reflection coefficient is

$$\boxed{\Gamma = \frac{z-1}{z+1}, \qquad z = \frac{1+\Gamma}{1-\Gamma}.}$$

Classical Input 13.1 (Cayley geometry). The map sends the finite passive half-plane $r \geq 0$ onto the closed unit disk with $\Gamma = 1$ omitted. Adjoining the point at infinity in impedance adds that missing boundary point. The open right half-plane maps conformally onto the open disk. Constant resistance and reactance become the familiar Smith circles. This conformal geometry is classical and is not replaced by polyphase refinement.

Writing $\Gamma = u + \mathrm{i}v$, the grid equations are

$$\left(u - \frac{r}{1+r}\right)^2 + v^2 = \frac{1}{(1+r)^2},$$

and, for $x \neq 0$,

$$(u-1)^2 + \left(v - \frac{1}{x}\right)^2 = \frac{1}{x^2}.$$

Fix

$$\omega_N = e^{2\pi \mathrm{i}/N}$$

and rotate the reflection coefficient by

$$R_{N,k}(\Gamma) = \omega_N^k \Gamma.$$

Theorem 13.2 (Polyphase Möbius action). *Regard normalized impedance as the projective line $\mathbb{P}^1(\mathbb{C}) = \mathbb{C} \cup \{\infty\}$. For $N \geq 1$, the pullback is a globally defined projective action. In the finite chart, wherever the denominator is nonzero, it is*

$$\boxed{T_{N,k}(z) = \frac{(1+\omega_N^k)z + (1-\omega_N^k)}{(1-\omega_N^k)z + (1+\omega_N^k)}.}$$

As maps on $\mathbb{P}^1(\mathbb{C})$ it satisfies

$$T_{N,k} \circ T_{N,\ell} = T_{N,k+\ell \,(\mathrm{mod}\, N)}.$$

Proof. Let $C(z) = (z-1)/(z+1)$. Then

$$T_{N,k} = C^{-1} \circ R_{N,k} \circ C.$$

For a division-free construction, represent a point by a nonzero pair $[u : v]$, identifying $[cu : cv] = [u : v]$ for $c \neq 0$. With $w = \omega_N^k$, use the matrix

$$A(w) = \frac{1}{2}\begin{pmatrix} 1+w & 1-w \\ 1-w & 1+w \end{pmatrix}.$$

It has determinant $w \neq 0$, preserves nonzero pairs, and commutes with rescaling, so it induces a projective map. Direct multiplication gives $A(w)A(w') = A(ww')$ and $A(1) = I$. Since $\omega_N^N = 1$, this proves the cyclic composition law modulo N. The Cayley coordinate of $[u : v]$ is $[u - v : u + v]$, and applying $A(w)$ changes this pair to $[w(u-v) : u+v]$. This proves the conjugacy also at poles. For $[z : 1]$, taking the coordinate ratio gives the displayed finite-chart formula. □

Corollary 13.3 (Impedance–admittance is the two-phase split)**.** *For $N = 2$ and finite $z \neq 0$,*

$$\boxed{T_{2,1}(z) = z^{-1}.}$$

On the full projective line it exchanges 0 and ∞. Thus a half-turn of the reflection disk exchanges normalized impedance and admittance.

Proof. $\omega_2 = \exp(\pi \mathrm{i}) = -1$, so $A(\omega_2)[u : v] = [v : u]$. The finite ratio is $1/z$ when $z \neq 0$. The coordinate swap also gives $[0 : 1] \leftrightarrow [1 : 0]$. No ordinary division by zero is used to represent infinity. □

13.1 Radial seeds and physical phase

For $0 \leq \rho < 1$ and angle θ,

$$\Gamma = \rho e^{\mathrm{i}\theta}$$

gives

$$z(\rho, \theta) = \frac{1 + \rho e^{\mathrm{i}\theta}}{1 - \rho e^{\mathrm{i}\theta}},$$

with

$$r(\rho, \theta) = \frac{1 - \rho^2}{1 - 2\rho\cos\theta + \rho^2}, \qquad x(\rho, \theta) = \frac{2\rho \sin\theta}{1 - 2\rho\cos\theta + \rho^2}.$$

A fixed ρ is a constant-mismatch circle. Refining N increases angular resolution without changing ρ.

For a lossless transmission line,

$$\Gamma_{\text{in}} = \Gamma_L e^{-2\mathrm{i}\beta\ell}.$$

One N-phase step corresponds to a line-length increment $\lambda/(2N)$, up to the chosen orientation convention. Hence the phase address has a direct electrical interpretation.

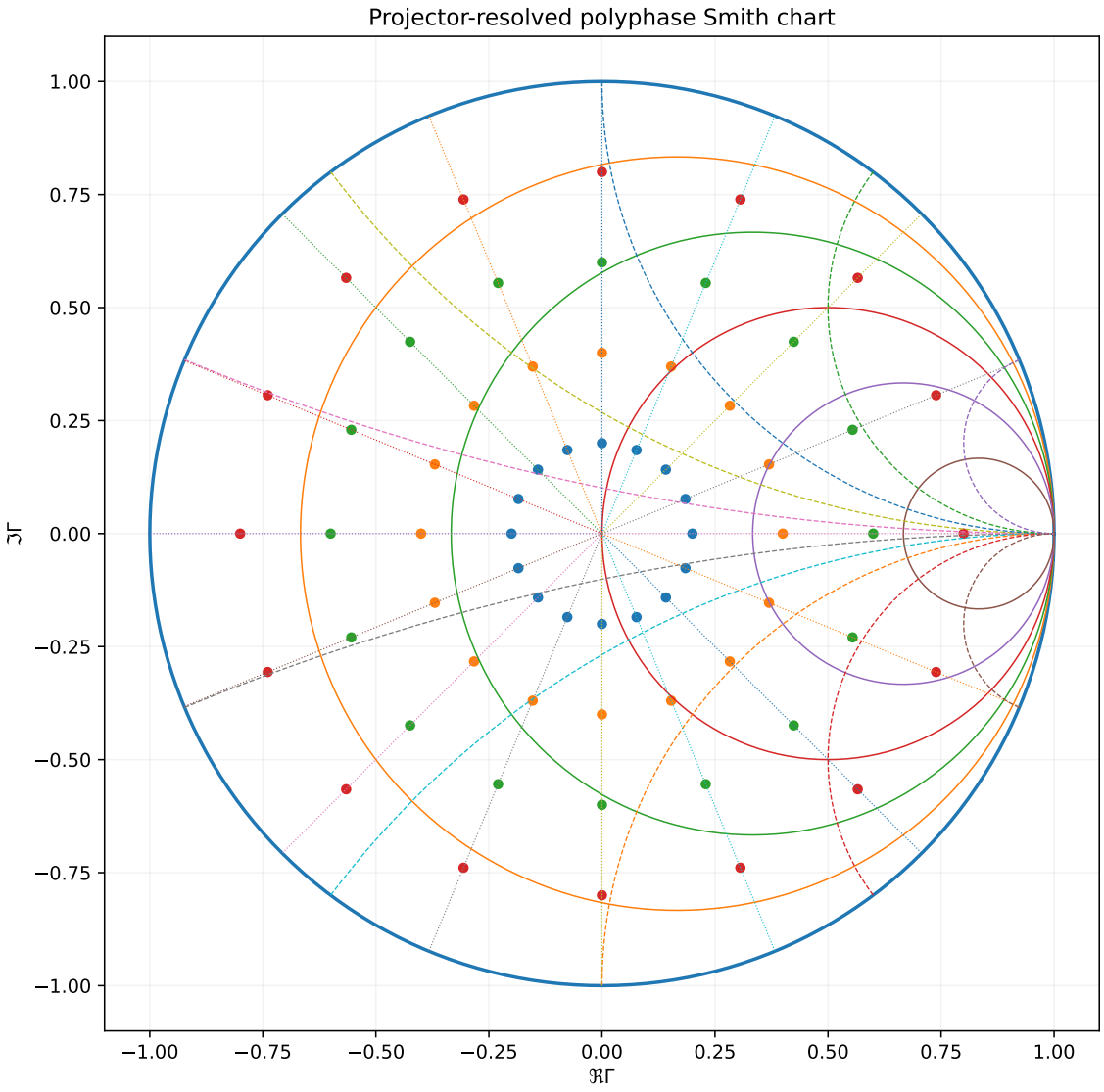


Figure 16. The classical Smith grid equipped with sixteen exact phase rays and four radial seed orbits. The Cayley transform creates the chart. The projector calculus supplies the nested cyclic coordinate system.

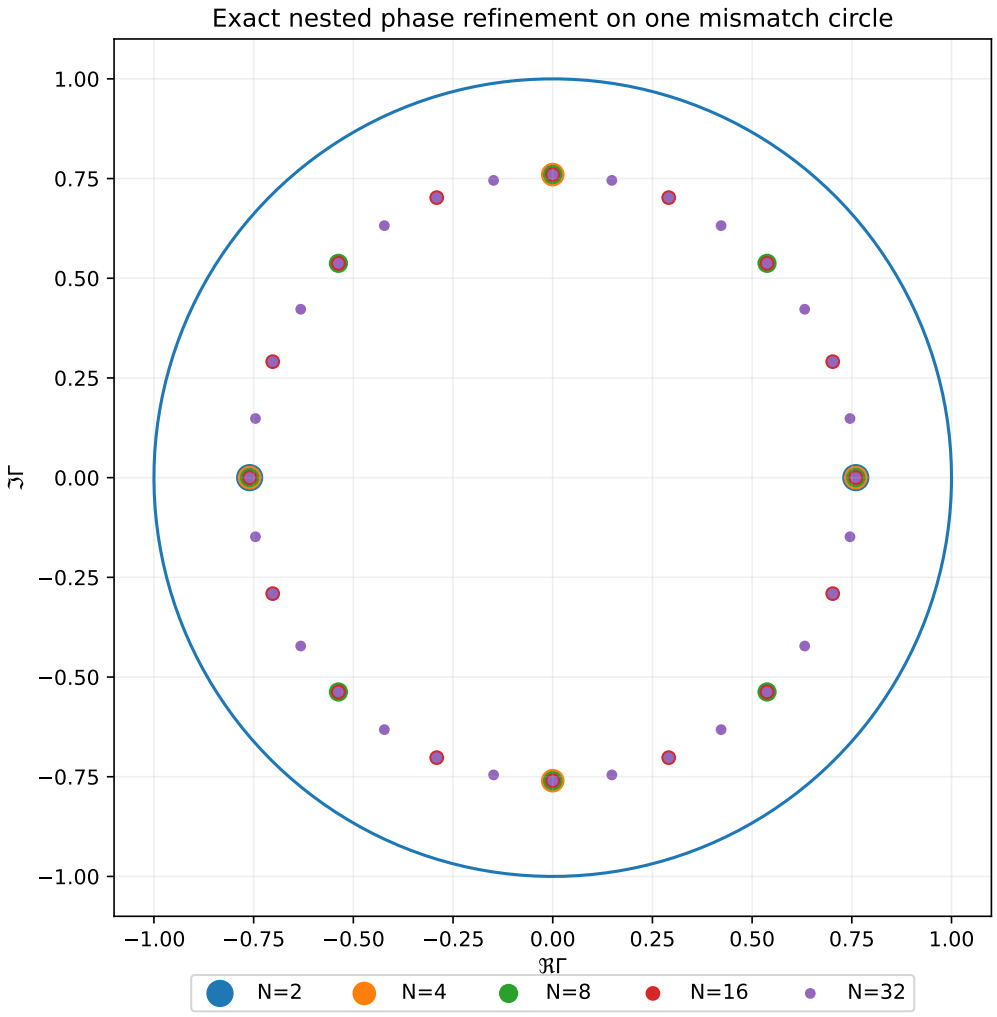


Figure 17. The same reflection circle at phase orders $2, 4, 8, 16, 32$. Each coarse phase ray appears in every finer grid whose order is a multiple of its order. The exact residue-projector refinement law concerns coefficient packets, not sums of impedance points.

14 Dyadic–odd phase factorization and boundary weighting

Write

$$N = 2^s M, \qquad M \text{ odd}.$$

The cyclic index splits by the Chinese remainder theorem:

$$\mathbb{Z}/N\mathbb{Z} \cong \mathbb{Z}/2^s\mathbb{Z} \times \mathbb{Z}/M\mathbb{Z}.$$

Proposition 14.1 (Dyadic–odd packet factorization)**.** *After identifying the residue labels by the Chinese remainder map, the phase projector factors as*

$$\boxed{P_{N,j} = P_{2^s,j_2} P_{M,j_o}.}$$

Proof. The two congruence conditions are independent because 2^s and M are coprime. Their indicator functions multiply to the indicator of the unique residue class modulo N satisfying both. □

The first dyadic step is exceptional in Smith coordinates because it is the inversion $z \mapsto z^{-1}$. Higher dyadic levels refine the open/short duality into quarter-, eighth-, and finer phase sectors.

Away from its pole $\Gamma = 1$, the inverse Cayley map has derivative

$$\frac{dz}{d\Gamma} = \frac{2}{(1-\Gamma)^2},$$

while normalized admittance has derivative proportional to $(1+\Gamma)^{-2}$. Away from $\Gamma = \pm 1$, this motivates the symmetric sensitivity

$$\boxed{w_{\text{Smith}}(\Gamma) = \frac{1}{|1-\Gamma|^2} + \frac{1}{|1+\Gamma|^2}.}$$

Boundary 14.2 (Weighted refinement). The sensitivity formula follows directly from the exact coordinate derivatives. Using it to choose an optimal adaptive grid, interpolation rule, or measurement design is a separate numerical problem. The illustration displays the weight but does not prove optimality.

15 Boundary-localized error and claim control

The strongest visual distinction between global Fourier reconstruction and exact packet retention is seen in a band around the discontinuity.

Conjecture 15.1 (Edge-packet approximation class)**.** There should exist natural function spaces in which a signal is represented as

$$f = s + \sum_q a_q \mathbf{1}_{\Omega_q},$$

with separate norms for smooth approximation, boundary geometry, packet amplitude, and packet identification. A projector-resolved approximation theory on such spaces should yield rates that do not mix smooth truncation error with discontinuity-location error.

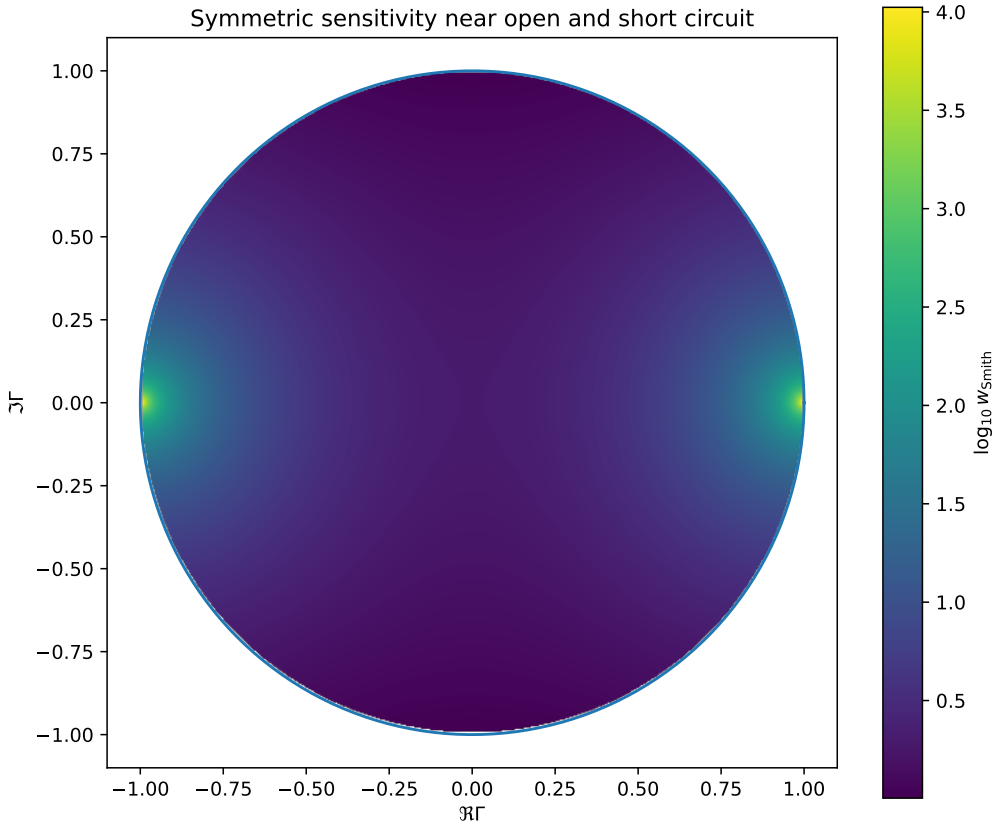


Figure 18. The proposed symmetric refinement weight becomes large near the open- and short-circuit boundary points. It is a condition-based adaptive coordinate, not an asserted optimal sampling law.

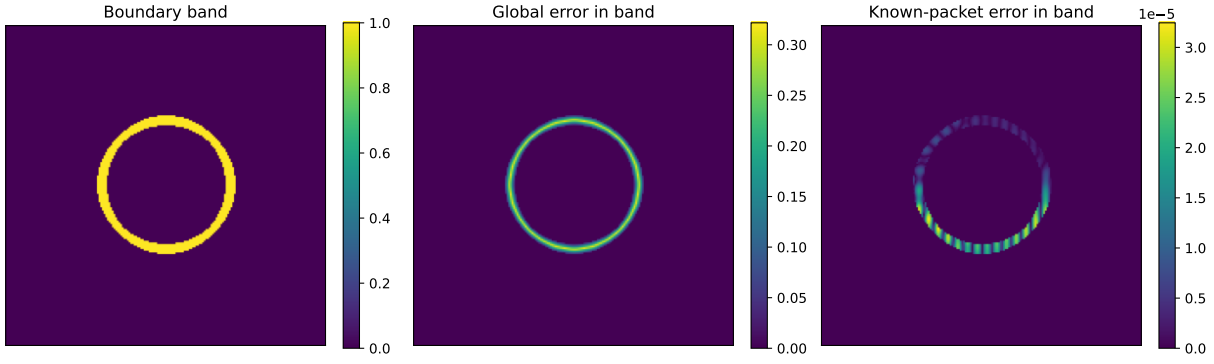


Figure 19. Error restricted to a narrow boundary band. Global truncation concentrates large error on the jump. The known-packet reconstruction has only the small smooth-packet error in the same band.

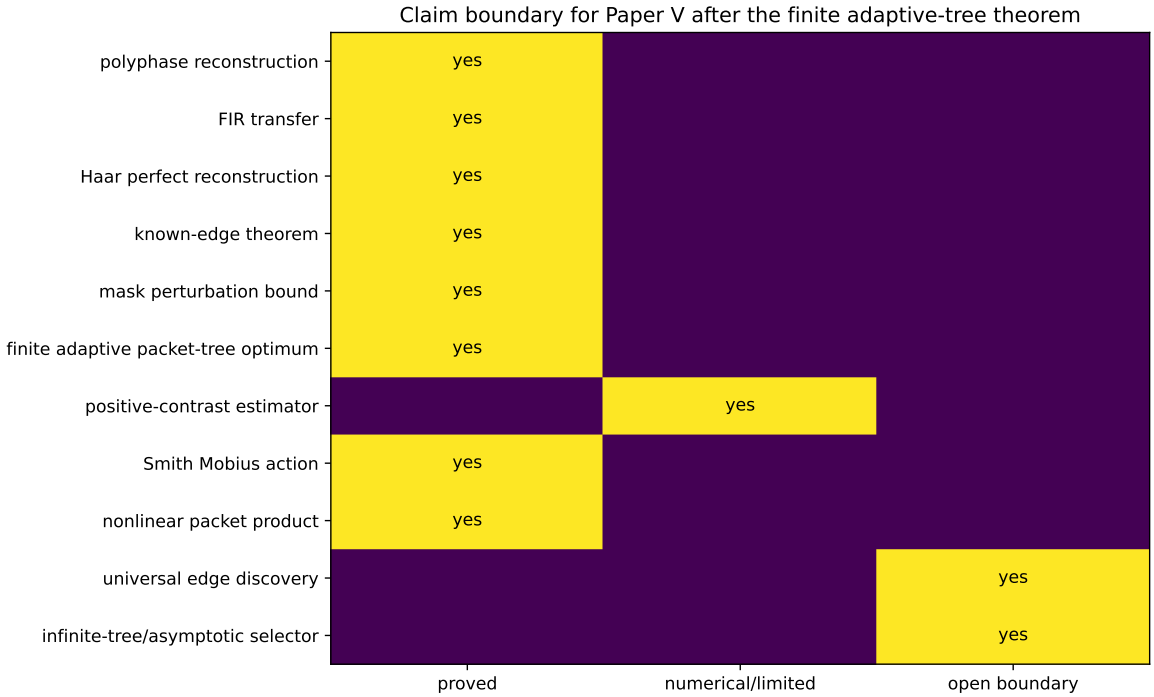


Figure 20. Proof and claim boundary. Finite adaptive-tree selection is proved. The remaining open rows concern universal recovery and infinite-tree or asymptotic selection.

16 Implementation and reproducibility

The companion package is organized by mathematical layer:

Table 2. Numerical companion modules.

module	implemented content
`polyphase.py`	residue projectors, analysis/synthesis, refinement, exact FIR transfer kernels, polyphase convolution
`filterbanks.py`	Haar analysis/synthesis, complete Haar packet trees, energy accounting, predict–update lifting
`adaptive.py`	complete-pruning enumeration, exact mixed-depth synthesis, dynamic-programming optimization, and stability certificates
`edges.py`	one- and two-dimensional Fourier truncation, known-packet reconstruction, narrow two-class estimators, harmonic filling, error metrics
`smith.py`	Cayley coordinates, cyclic Möbius action, Smith circles, sensitivity weight, coefficient projectors
`nonlinear.py`	coefficient packets, cyclic product convolution, nonlinear shifted packet reconstruction
`experiments.py`	all machine-readable numerical certificates and tabular summaries
`visuals.py`	the eighteen figure pairs used in the chapter

The recorded companion run reports that its 33 tests passed, and all 18 figure pairs and numerical tables were regenerated. The Lean package contributes 20 independent exact-model tests and 16 tests of the verifier itself. These account for another

36 passing tests. The exact-model suites count 305,933 assertions. These counts are separate from the 225 Lean theorem targets.

All numerical computations are CPU-based. Floating-point residuals measure finite implementations and may vary at rounding level with the scientific libraries. The universal mathematical statements are supported by the Lean proofs. The Python modules are not compiled or verified by Lean. The adaptive-tree certificate uses floating-point costs and an exhaustive finite comparison, so its displayed gap is numerical evidence rather than an interval-arithmetic proof about those computed costs. Appendix D explains the verification boundary.

Numerical Certificate 16.1 (Finite numerical residuals). The largest recorded residuals were

identity	residual
polyphase reconstruction	0
nested refinement	0
polyphase convolution	4.44×10^{-16}
Haar reconstruction	8.88×10^{-16}
wavelet-packet reconstruction	2.22×10^{-15}
adaptive-tree optimum versus exhaustive search	0
adaptive mixed-depth synthesis	8.88×10^{-16}
lifting reconstruction	4.44×10^{-16}
Smith rotation conjugacy	3.34×10^{-16}
Smith impedance–admittance duality	0
nonlinear product-packet reconstruction	1.33×10^{-15}
nonlinear shifted-filter reconstruction	0

The Haar and wavelet energy residuals were 5.68×10^{-14} and 2.27×10^{-13}, respectively.

17 Conclusions

The signal realization makes the seed, address, packet, transfer, refinement, and synthesis operations concrete. The objects and laws can be read directly:

$$\begin{aligned}
\text{seed} &= \delta_0, \\
\text{address} &= S^n, \\
\text{packet} &= P_{N,r}x, \\
\text{transfer} &= P_{N,k}HP_{N,s}, \\
\text{refinement} &= P_{N,r} = \sum_{s \succ r} P_{M,s}, \\
\text{synthesis} &= \sum_{r} P_{N,r}x, \\
\text{nonlinearity} &= P_{N,r}F\left(\sum_{s} P_{N,s}x\right).
\end{aligned}$$

Filter banks, wavelet packets, adaptive complete prunings, edge packets, and Smith coordinates inhabit the same operator grammar, but their claims differ. Polyphase and filter-bank identities are exact.

Wavelet perfect reconstruction is exact. The known-edge theorem is exact conditional on the decomposition. Estimated-edge performance depends on a detection model. The Smith chart remains the classical Cayley geometry equipped with an exact cyclic refinement algebra. Keeping those distinctions explicit is part of the calculus, not an editorial afterthought.

Chapter VI turns from discrete signal transfer to semigroups, resolvents, conditioning, stochastic evolution, fractional powers, and PDE kernels.

A Algorithm statements

A.1 Exact polyphase filtering

Given x, h, and N:

1. Form $x_s[n] = x[Nn + s]$ for $0 \le s < N$.
2. Form every transfer kernel $h_{ks}[m] = h[Nm + k - s]$.
3. Compute
$$y_k[n] = \sum_s \sum_m h_{ks}[m] x_s[n - m].$$
4. Interleave the y_k channels.

The algorithm is algebraically identical to direct convolution. Its purpose is to expose independent and interacting packets for block computation and refinement.

A.2 Conditional known-edge reconstruction

Given a decomposition $f = s + j$:

1. Apply the selected smooth approximation Q_m to s only.
2. Retain j exactly.
3. Return $Q_m s + j$.

The error certificate is $\|s - Q_m s\|$.

A.3 Limited positive-contrast estimator

The included finite estimator does the following:

1. Split the observed values into two classes by an Otsu threshold.
2. Keep coherent connected components and fill holes.
3. Harmonically fill the selected packet to estimate the missing background.
4. Smooth the filled background spectrally.
5. Estimate one packet amplitude by a robust median residual.

6. Recombine the estimated smooth and edge packets.

This procedure is useful for the supplied separated synthetic case. It assumes a positive-contrast two-class packet and is not promoted to a general edge theorem.

B Numerical record

Table 3. Main edge-study measurements.

case	global RMSE	known-packet RMSE	estimated RMSE
1D rectangular packet	5.4023×10^{-2}	1.3692×10^{-4}	2.4029×10^{-2}
2D disk packet	3.7643×10^{-2}	6.4752×10^{-5}	3.4043×10^{-3}
2D multiedge packet	3.3740×10^{-2}	known composite packet only	

Table 4. Main software artifacts.

artifact	purpose
`results/numerical_results.json`	complete machine-readable certificate record
`tables/numerical_summary.csv`	flat summary of errors and residuals
`tables/sparse_representation.csv`	wavelet/Fourier coefficient study
`tests/`	automated finite identity and estimator tests
`figures/`	PNG and vector-PDF figure pairs
`VERIFY_PAPER.py`	combined manuscript, numerical, and Lean checks

C Notation crosswalk

Table 5. Notation crosswalk.

common symbol	**signal realization**	**interpretation**
s	δ_0	primitive signal seed
A_n	S^n	address operator
$P_{N,r}$	residue mask	phase packet projector
T_{ks}	$P_{N,k} H P_{N,s}$	input-to-output phase transfer
E, R	analysis and synthesis matrices	coordinate conversion and reconstruction
$W(\gamma)$	filter or path weight	refinement or implementation cost
$\Delta(F)$	coarse/fine signal defect	aliasing, leakage, or estimator inconsistency
$\mathrm{Hol}(\gamma)$	closed phase transfer	cycle consistency in a network
Γ	reflection coefficient	Smith-disk coordinate

D Formalization and reproducibility record

D.1 What has been checked

The preserved Part I manuscript has twenty theorem-like statements: Theorems, Propositions, and Corollaries, counted together. The Lean project contains 225 explicit supporting theorem targets in 23 proof modules. All twenty statements have an explicit scope row and principal declaration targets. The retained compiler record documents a successful build and axiom audit with warnings treated as errors. The formal companion identifies its source, toolchain, and reproduction procedure. The audit inspected all 827 declarations in the project namespace, including definitions and automatically generated declarations. The latter is an implementation census, not a count of 827 mathematical theorems.

The accepted axiom dependencies are `propext`, `Classical.choice`, and `Quot.sound`. No admitted proof or additional project axiom is permitted. The verifier checks the imported modules, registered theorem inventory, dependency pins, source hashes, and coverage of the twenty-row scope contract. Its successful status is `LEAN_CHECKS_PASS`. The correspondence between a mathematical sentence in this PDF and its Lean type is a manual mathematical review, not an automated semantic-equivalence test.

The mathematical text integrates the checked hypotheses and conclusions. Editorial changes do not increase the Lean theorem count. Eleven foundational formal-power-series proofs in `CyclicSeries` are adaptations of Chapter IV and are identified as such in the provenance record.

D.2 Result-by-result correspondence

The following table summarizes the reviewed statement boundary. Full qualified declaration names, source locations, and supplementary targets appear in `MANUSCRIPT_TARGET_MAP.md` and the machine-readable scope contract. Every row below has supporting kernel-checked declarations. Numerical certificates, classical-input paragraphs, remarks, and conjectures are not additional theorem rows.

Table 6. The twenty manuscript results and their Lean statement scopes.

Result	Principal modules	Checked mathematical content
1.1	HilbertPackets	Complex ℓ^2 impulse expansion, finite-support identity, unitary shifts, and closed shift-generated span.
2.1	HilbertPackets	Bounded self-adjoint residue projectors, pairwise annihilation, completeness, and orthogonal ranges.
2.2	IntegerPolyphase; HilbertPackets	Signed quotient/remainder coordinates, both inverse identities, and the joint Hilbert norm of the channels.
2.3	HilbertPackets	Literal bounded-projector refinement for positive divisible moduli.
3.1	FIRTransfer; ZTransform	Finite-support tap reindexing for all bilateral inputs; numerical channel transforms under weighted absolute summability.
4.1	FilterBank; FiniteTransforms	Finite Laurent/FIR bank composition, faithful action on sequences, gain-delay reconstruction criterion, and the Nd original-sample delay.
4.2	HaarChannels; HaarHilbert	Literal even/odd input samples, normalized complex low/detail channels, and ℓ^2 energy equality.
4.3	Lifting; FiniteTransforms	Predict/update inverse maps, determinant one, and both flat finite-signal inverse identities.

Result	Principal modules	Checked mathematical content
5.1	WaveletPackets	Every finite depth of literal Haar packet operators, energy equality, and reconstruction using actual Hilbert adjoints.
6.1	KnownEdge	Exact cancellation of one retained packet, and equality of approximation errors.
6.2	KnownEdge	Finite families of retained packets, with no geometry assumption on the packets.
7.1	MaskPerturbation	Measurable-mask symmetric-difference norm, extended finite-p error bound, pointwise and essential supremum bounds, and smooth-background error.
11.2	CompletePrunings; PruningSums	Recursive cuts correspond to complete covering antichains; literal finite node sums preserve synthesis.
11.3	PruningSums	Finite arbitrary-objective minimization, attained positive gap with competitors, strict stability, and the singleton case.
11.4	FiniteAdaptive; PruningAlgorithm	Deterministic stop-on-ties selector, global optimum, arbitrary positive finite branching, and the stated unit-operation count.
11.5	PruningSums; PruningAlgorithm	Attained maximum error envelope, optimal-value perturbation, strict-gap selector stability, and probability transfer from a joint event.
12.1	CyclicSeries	Formal Cauchy-product residue convolution and the finite-tuple power identity, including the empty product at exponent zero.
13.2	SmithAction; SmithCyclic	Nonzero homogeneous pairs modulo scaling, Cayley conjugacy, the actual exponential phase, and cyclic composition on the projective line.
13.3	SmithCyclic	Two-phase coordinate swap, $0 \leftrightarrow \infty$, and finite impedance inversion when $z \neq 0$.
14.1	CRT	Signed integer residue-mask factorization for coprime moduli, including the dyadic–odd case.

The remaining foundational modules `Packets`, `FiniteSelection`, and `BinaryAdaptive` provide generic coordinate, finite-minimum, and binary-tree lemmas used or retained alongside the concrete constructions. The Hilbert target is the square-summable space on a product of indices. It is not the supremum-norm function space that Lean would otherwise assign to an unqualified finite family of channels.

D.3 Three bridges that matter for the statement

Finite filters and numerical transforms. The filter-bank proof is an algebraic theorem about finite Laurent-polynomial kernels acting faithfully on bilateral sequences. Equality at a single complex value of z would be insufficient to prove a polynomial identity. The impulse argument avoids that error. The separate transform theorem proves the finite-filter action agrees with evaluation wherever the weighted input series converge absolutely. An arbitrary bilateral ℓ^2 sequence need not have an absolutely convergent numerical z-transform. The all-sequence time-domain theorem makes no such assumption.

Recursive trees and set-valued prunings. The formal tree permits any positive finite arity at an internal node, including arity one. A node has a position in the tree, so equal data labels on distinct nodes do not identify the nodes. A recursive cut is either a stop or a collection of child cuts. Its equivalence

with a covering antichain is proved in both directions. The cost, envelope, and operator sums are then shown to equal actual finite sums over the selected node set. This connects the recursive optimizer to the manuscript's set-valued definition instead of assuming that connection.

Hilbert operators and finite experiments. The coordinate maps, literal Haar channels, and packet operators are constructed on complex ℓ^2. Square summability and boundedness are established before adjoints are used. The finite Python experiments separately check indexing, reconstruction, and energy at finite lengths. They illustrate these operators but are not the objects executed by the Lean kernel. Likewise, the Python Smith routine evaluates only finite charts. The Lean projective construction supplies the pole-safe universal statement.

D.4 Recorded checks and numerical conventions

Check family	Accepted count
Original numerical companion tests	33
Independent exact-model tests	20
Verifier positive and negative controls	16
Total tests in these three families	69
Exact assertions inside the independent model suites	305,933
Registered Lean theorem targets	225
Lean proof modules	23
Regenerated figure pairs	18

The recorded numerical comparison states that all scalar entries in the recovered numerical-results JSON reproduced exactly in its environment. This includes the 458,330-pruning depth-five comparison, its selected six-node pruning, the optimum gap, and the reconstruction residuals. The five Fourier sparseapproximation budgets were also checked for conjugate-pair closure. The figures and tables use the regenerated data. This exact replay of the stored floating-point numbers is stronger than a visual similarity check, but it is not a promise of bitwise equality on every platform.

For the one-dimensional low-pass experiment, "cutoff 18" means 37 signed frequency bins, not 18 retained coefficients. The disk experiment uses a square 33×33 frequency mask, not a radial cutoff. The sparse comparison uses largest-coefficient budgets instead of either low-pass convention. The full-image and boundary-band RMSEs are normalized by the number of samples in the respective comparison set. The supplied inpainting routines perform a fixed finite number of periodic neighbor-averaging iterations. The term "harmonic filling" describes that iteration, not a separately proved exact continuum Dirichlet solver.

The independent exact-model suite checks signed polyphase/FIR addresses, exact lifting and normalized Haar arithmetic, mask bounds, and exhaustive finite-tree objectives. Its tree census includes all 626 ordered rooted tree shapes with one through eight nodes and 129,958 node subsets. These finite checks can catch implementation or convention errors. They are not substitutes for the universal Lean proofs. The verifier controls exercise failure paths, including incomplete scope contracts and unregistered proof targets. Mandatory verification conditions use explicit exceptions rather than Python assertions, so optimized Python cannot silently remove them.





D.6 What remains open

All twenty scoped theorem-like statements have reviewed, accepted formal counterparts. The two edge-recovery conjectures remain open research directions. No universal segmentation theorem, infinite-tree consistency result, minimax rate, or optimal Smith sampling theorem is inferred from the finite results. Classical Smith-grid and transmission-line formulas provide application context. Those contextual assertions are not additional Lean-certified theorem rows.

> The formal scope consists of the twenty Part I results. Supporting declarations, reused results and proofs in the broader written development have distinct roles.

Part II: Extended mathematical development

Multiresolution Signal Calculus, Singular Packets, and Stable Reconstruction

This part preserves the complete standalone mathematical branch, with the explicit audit repairs recorded in the accompanying source ledger. Its sixty-three proofs are prose proofs. No additional Lean coverage is asserted. Section and result numbers carry the prefix E to distinguish this scope from Part I.

Overview

Signals on a lattice admit exact coordinates indexed by the residue classes of every finite index sublattice. Convolution becomes a matrix of packet transfers, while filter banks and packet trees become exact changes of coordinates. We develop this structure on sequence spaces, prove quantitative stability for imperfect and perturbed banks, and give an exact dynamic recursion for adaptive packet trees. A singular signal is written as a smooth component plus step or truncated power packets that record jumps in the signal and its derivatives. Retaining the correct packets leaves only the smooth approximation error. Estimated packets are governed by exact formulas involving value error, region overlap, boundary displacement, and the norm of the smooth approximation operator. The main recovery result treats separated jumps sampled on a uniform grid. An explicit contrast gap gives exact recovery of every jump cell, grid scale location accuracy, amplitude control, and finite norm consistency. Probability bounds cover sub-Gaussian noise and repeated measurements. Haar coefficients, two shifted grids, and annihilating differences give complementary localization results. The final sections treat Gibbs oscillation, nonlinear residue interactions, numerical illustrations, and cyclic Cayley coordinates.

E1 Introduction

Polyphase decompositions, filter banks, lifting schemes, and wavelet packets are standard parts of multirate signal processing and harmonic analysis. Their classical development is described in [CR83, Vai93, VK95, SN96, Dau92, Mal09]. These theories are often introduced through several parallel languages. Residue classes appear in polyphase analysis. Matrix symbols appear in filter banks. Trees appear in wavelet packets. Singularities are handled through a different approximation or detection method. The first aim of this part is to place these constructions in one exact coordinate system.

The second aim is to separate representation from recovery. If a signal has already been written as

$$f = s + j$$

with a smooth part s and a singular packet j, then

$$\tilde{f} = Qs + j$$

satisfies

$$f - \tilde{f} = s - Qs.$$

This identity is exact, but it does not discover j. A useful signal theory must therefore distinguish three errors. The first is the approximation error of the smooth component. The second is the error in the estimated singular packet. The third is the amplification caused by applying a reconstruction operator to that packet error. The middle sections of this part isolate all three terms.

The singular packets considered here range from indicator functions of regions to truncated powers

$$\frac{(t-\tau)_+^m}{m!}.$$

The latter encode jumps in derivatives of order m. They permit a piecewise smooth function to be reduced to a globally Sobolev remainder under explicit trace assumptions. This gives a precise version of the informal statement that a discontinuity should be represented directly rather than approximated by a global smooth basis.

The recovery theory begins with one-dimensional separated jumps. For bounded deterministic noise, thresholding adjacent sample contrasts recovers the exact grid cells containing the jumps whenever the jump size exceeds the combined smooth variation and noise by an explicit margin. The same proof gives location and amplitude bounds. A sub-Gaussian version follows from a union bound. Averaging repeated measurements reduces the required threshold in the expected way. Haar coefficients provide a second detection mechanism. Two grids shifted by half a cell guarantee that every jump is well placed in at least one grid, which removes the most elementary dyadic alignment failure.

The literature on singularity detection is much broader than the results proved here. Wavelet modulus maxima are developed in [MH92]. Best basis selection in packet libraries was introduced in [CW92]. Nonlinear approximation is surveyed in [DeV98]. Geometric image models, wedgelets, and curvelets appear in [Don99, CD04]. The present part does not replace those theories. Its narrower purpose is to give a common exact calculus in which coordinate reconstruction, adaptive basis selection, singular packet estimation, and smooth approximation remain logically separate.

This part is organized as follows. Appendices E2 and E3 develop lattice addresses and exact residue coordinates in any dimension. Appendices E4 and E5 express convolution and filter banks in those coordinates, including stability under imperfect reconstruction and filter perturbation. Appendix E6 treats arbitrary pruned packet trees and stability of the best basis decision. Appendices E7 and E8 introduce step and higher order singular packets, exact reconstruction identities, and error formulas. Appendices E9 and E10 prove deterministic and probabilistic recovery results. Appendices E11 and E12 treat Gibbs oscillation and nonlinear interactions. The final sections record finite numerical illustrations and a secondary cyclic Cayley realization.

Several conclusions are deliberately excluded. The chapter does not prove a universal image segmentation theorem. Exact recovery of separated jumps in one dimension does not imply stable recovery of arbitrary boundaries in higher dimensions. Retaining a correct singular packet does not remove noise or model error. An adaptive tree selected by an additive objective need not remain optimal when global hardware costs or cross-packet interactions are introduced. The Cayley coordinate section is a secondary conformal realization and is not used in the recovery theorems.

E1.1 Main results

The first result is an exact coordinate theorem. Let L be a finite index sublattice of $\mathbb{Z}^d$, and let R contain one representative from every coset. The analysis map

$$x \longmapsto (x_r)_{r\in R}, \qquad x_r(\ell) = x(r+\ell),$$

is an isometric isomorphism on every ℓ^p space. This statement includes the usual one-dimensional polyphase decomposition, but it also applies to arbitrary finite index lattice packets in several dimensions. If $L' \leq L$, the coarse packets split exactly into their fine descendants. A convolution operator becomes a matrix of convolution operators on L, with entries

$$h_{rs}(m) = h(r - s + m).$$

The matrix is therefore a coordinate representation of the original operator, not a reduced model.

The filter bank results distinguish exact reconstruction from quantitative near reconstruction. Exact recovery with delay d is the pointwise matrix identity

$$R(z)E(z) = z^{-d}I.$$

When this identity fails, the reconstruction error is controlled by the essential supremum of the delay compensated multiplier defect. If the filters themselves are perturbed, theorem E5.4 separates the analysis error, the synthesis error, and their product. The frame bound in theorem E5.7 shows that a lower singular value gap survives every sufficiently small perturbation. This makes the distinction between algebraic invertibility and stable numerical reconstruction explicit.

The packet tree results treat every admissible pruning, not only complete trees of a fixed depth. Unitary local splits give a unitary global transform. Local frame bounds propagate recursively in the nonunitary case. If the cost of a pruning is additive over retained nodes, then

$$F(v) = \min\left\{ c(v), \sum_{w \in \mathrm{child}(v)} F(w) \right\}$$

gives the exact optimal tree. The cost and the selected basis are stable when the node costs change by less than the gap separating the optimal pruning from its competitors.

The singular packet results begin with trace data. If a piecewise smooth function has a jump in its derivative of order m at τ_q, then the packet

$$\left[f^{(m)}\right]_{\tau_q} \frac{(t - \tau_q)_+^m}{m!}$$

removes exactly that trace jump. Subtracting all packets through order $r - 1$ leaves a global $W^{r,p}$ remainder. The packet coefficients are uniquely determined by the one-sided trace jumps. This gives a finite and explicit singular part for the class covered by theorem E7.2.

The reconstruction formula with an estimated packet is

$$f - \hat{f}_m = (I - Q_m)s + (I - Q_m)(j - \hat{j}).$$

This identity identifies the two terms that must be controlled. The exact packet formulas in appendix E8 then evaluate or bound $\|j - \hat{j}\|$ from amplitude error, region overlap, boundary displacement, and the order of a derivative jump. A value jump has location modulus $|\tau - \hat{\tau}|^{1/p}$ in finite L^p, while every positive order truncated power packet is Lipschitz in its location.

The principal recovery theorem uses adjacent sample contrasts. Let

$$b_h = Mh + 2\varepsilon$$

be the largest contrast that can be produced by a Lipschitz background and bounded noise in one grid cell. If

$$a_{\min} > 2b_h,$$

then every threshold between b_h and $a_{\min} - b_h$ detects exactly the cells containing jumps. The jump locations are recovered within $h/2$, and each amplitude error is at most b_h. The probability theorem replaces the deterministic noise term by

$$2\sigma\sqrt{\log(2n/\delta)}.$$

Two half shifted Haar grids provide a second recovery mechanism. Every point is at least one quarter of a cell away from the boundary of one of the grids, so the Haar coefficient gap can be made independent of dyadic alignment.

E1.2 Notation

The following symbols recur throughout this part.

symbol	meaning
$L \le \mathbb{Z}^d$	a finite index sublattice
R	a set of representatives for $\mathbb{Z}^d/L$
P_r	the coordinate projector onto the coset $r + L$
$\mathcal{A}_L, \mathcal{S}_L$	lattice analysis and interleaving synthesis
E, R	filter bank analysis and synthesis symbols
Q_m	a bounded linear approximation operator acting on the smooth part
$j, \hat{j}$	the true and estimated singular packets
H_τ	the step packet $\mathbf{1}_{[\tau,1]}$
$\phi_{m,\tau}$	the truncated power packet $(t-\tau)_+^m/m!$
h	the grid spacing in the recovery theorems

E2 Signal seeds and lattice addresses

Let $1 \le p \le \infty$. For a countable set G, write $\ell^p(G)$ for the usual complex sequence space. For an integer $d \ge 1$, when $G = \mathbb{Z}^d$, translation by $a \in \mathbb{Z}^d$ is

$$(T_a x)(n) = x(n-a).$$

Let δ_0 be the impulse at the origin.

Theorem E2.1 (Impulse generation). *For $1 \le p < \infty$,*

$$\overline{\operatorname{span}\{T_a\delta_0 : a \in \mathbb{Z}^d\}}^{\ell^p} = \ell^p(\mathbb{Z}^d).$$

The same statement holds in $c_0(\mathbb{Z}^d)$ with the supremum norm. In $\ell^\infty(\mathbb{Z}^d)$, the closed span of the translation orbit is exactly $c_0(\mathbb{Z}^d)$.

Proof. The translation orbit is the standard coordinate family $(\delta_a)_{a\in\mathbb{Z}^d}$. Its finite span is the space of finitely supported sequences. Finitely supported sequences are dense in ℓ^p for finite p and in c_0. Their closure in the supremum norm is c_0, which is a proper closed subspace of ℓ^∞. □

Remark E2.2. The distinction at $p = \infty$ is structural. The impulse generates every finitely supported signal algebraically, but it is not a topological cyclic seed for all bounded signals.

Proposition E2.3 (Finite periodic signals)**.** *Let $G = \mathbb{Z}^d/L$, where $L \leq \mathbb{Z}^d$ has finite index. The translation orbit of the impulse at the identity is a basis of $\mathbb{C}^G$. In particular, one impulse generates every signal on the finite periodic lattice.*

Proof. Translation sends the impulse at the identity to the impulse at each element of G. These impulses form the standard basis of $\mathbb{C}^G$. □

E3 Polyphase coordinates on finite index sublattices

Let $L \leq \mathbb{Z}^d$ be a sublattice of finite index q. Choose a set $R \subset \mathbb{Z}^d$ containing one representative from every coset in $\mathbb{Z}^d/L$. Thus

$$\mathbb{Z}^d = \bigsqcup_{r\in R} (r + L).$$

For $x : \mathbb{Z}^d \to \mathbb{C}$, its channel at r is the sequence on L

$$x_r(\ell) = x(r + \ell).$$

Definition E3.1 (Lattice packet)**.** For $r \in R$, define the coordinate projector

$$(P_r x)(n) = \mathbf{1}_{r+L}(n)x(n).$$

The family $(P_r)_{r\in R}$ is called the packet system associated with L and R.

For finite p, equip $\bigoplus_{r\in R} \ell^p(L)$ with

$$\|(u_r)\|^p = \sum_{r\in R} \|u_r\|_p^p.$$

For $p = \infty$, use the maximum norm.

Theorem E3.2 (Exact lattice polyphase decomposition)**.** *For every $1 \leq p \leq \infty$, the analysis map*

$$\mathcal{A}_L : \ell^p(\mathbb{Z}^d) \longrightarrow \bigoplus_{r\in R} \ell^p(L), \qquad \mathcal{A}_L x = (x_r)_{r\in R},$$

is an isometric isomorphism. Its inverse is the interleaving map

$$\mathcal{S}_L(u)(r + \ell) = u_r(\ell).$$

The projectors satisfy

$$P_r P_s = \delta_{rs} P_r, \qquad \sum_{r \in R} P_r = I.$$

Proof. Every address has a unique representation $r + \ell$. For finite p,

$$\|x\|_p^p = \sum_{r \in R} \sum_{\ell \in L} |x(r+\ell)|^p = \sum_{r \in R} \|x_r\|_p^p.$$

The corresponding supremum identity is immediate when $p = \infty$. Unique address representation proves that analysis and interleaving are inverse maps. The projector identities follow from the disjoint coset partition. □

Corollary E3.3 (One dimensional residue coordinates)**.** *For $L = N\mathbb{Z}$ and $R = \{0, \dots, N-1\}$,*

$$x_r(k) = x(Nk + r)$$

and

$$\ell^p(\mathbb{Z}) \cong \bigoplus_{r=0}^{N-1} \ell^p(\mathbb{Z})$$

isometrically.

Proof. Apply theorem E3.2 to the standard residue representatives. □

Proposition E3.4 (Translation covariance)**.** *Fix $a \in \mathbb{Z}^d$. For every $r \in R$, write uniquely*

$$r - a = r_a + \lambda_{a,r}, \qquad r_a \in R, \quad \lambda_{a,r} \in L.$$

Then

$$(\mathcal{A}_L T_a x)_r(\ell) = x_{r_a}(\ell + \lambda_{a,r}).$$

Thus a fine translation permutes the lattice packets and applies an internal translation on L.

Proof. Evaluate at $r + \ell$:

$$(T_a x)(r + \ell) = x(r + \ell - a) = x(r_a + \ell + \lambda_{a,r}).$$

This is the stated channel formula. □

E3.1 Refinement of sublattices

Let $L' \le L \le \mathbb{Z}^d$, with both indices finite. Choose representative systems R for $\mathbb{Z}^d/L$ and R' for $\mathbb{Z}^d/L'$. For $r \in R$, define

$$\mathrm{Child}(r) = \{s \in R' : s + L' \subset r + L\}.$$

Theorem E3.5 (Exact lattice refinement)**.** *For every coarse packet,*

$$P_r^L = \sum_{s \in \mathrm{Child}(r)} P_s^{L'}.$$

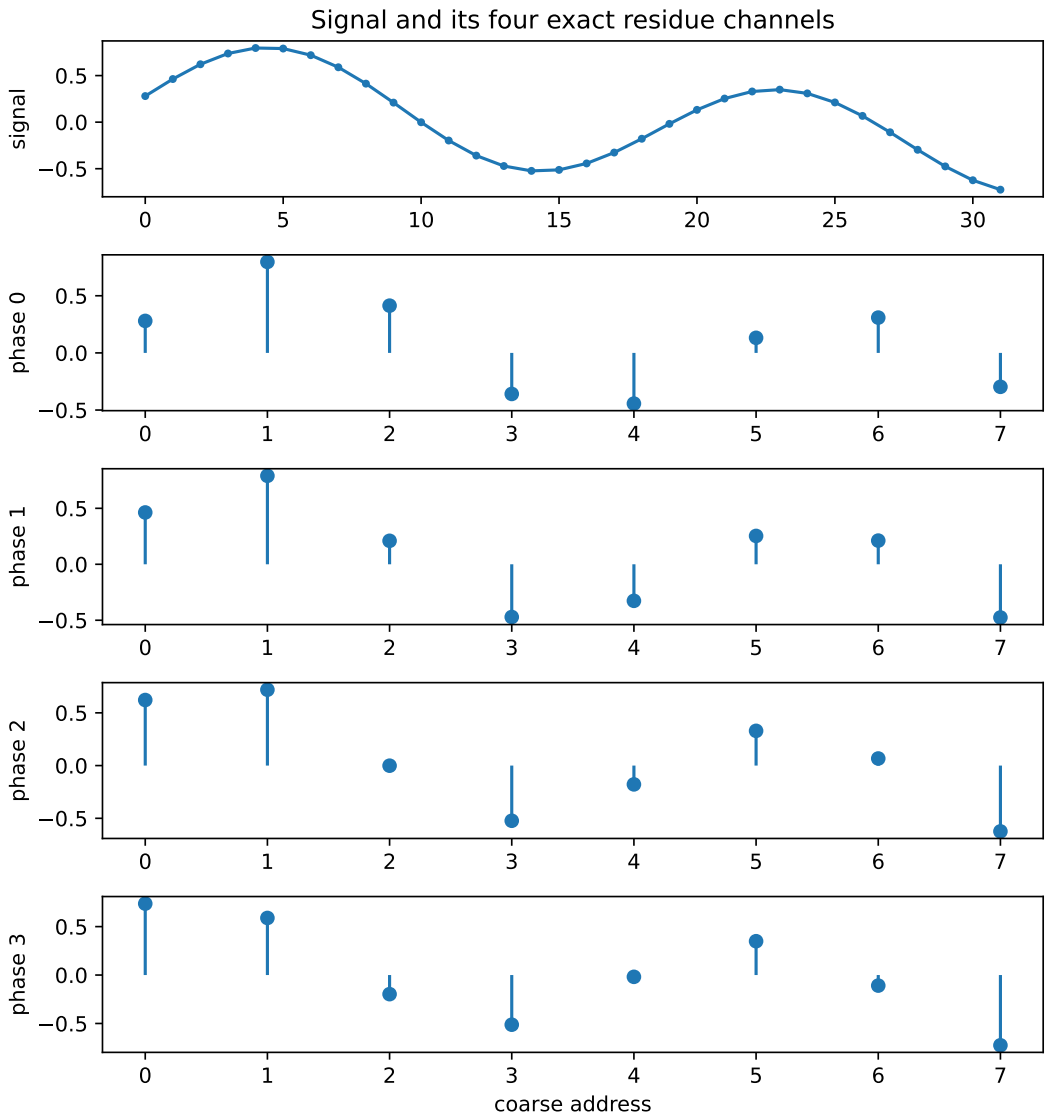


Figure E1. A signal and its four exact residue channels. Interleaving the channels returns the original samples without approximation.

The fine packets in the sum are pairwise disjoint. Refinement through several nested sublattices is transitive.

Proof. The coset $r + L$ is the disjoint union of the L' cosets contained in it. Multiplying by the corresponding indicator functions gives the projector identity. A nested partition of a coset is independent of whether the intermediate level is inserted, which proves transitivity. □

Example E3.6 (Rectangular image phases)**.** For

$$L = M\mathbb{Z} \times N\mathbb{Z} \subset \mathbb{Z}^2,$$

one may take

$$R = \{0, \dots, M-1\} \times \{0, \dots, N-1\}.$$

The analysis map separates an image into MN exact phases. Refining (M, N) to (M', N'), where $M \mid M'$ and $N \mid N'$, splits every coarse phase into

$$\frac{M'}{M}\,\frac{N'}{N}$$

fine phases.

E4 Convolution as a packet transfer matrix

Let $h \in \ell^1(\mathbb{Z}^d)$ and $x \in \ell^p(\mathbb{Z}^d)$, with $1 \le p \le \infty$, and define

$$(Hx)(n) = (h * x)(n) = \sum_{q \in \mathbb{Z}^d} h(q)x(n-q).$$

Young's inequality gives

$$\|Hx\|_p \le \|h\|_1 \|x\|_p$$

for every $1 \le p \le \infty$.

Definition E4.1 (Packet transfer kernel)**.** For output residue $r \in R$ and input residue $s \in R$, define a kernel on L by

$$h_{rs}(m) = h(r - s + m), \qquad m \in L.$$

Theorem E4.2 (Exact convolution transfer law)**.** *Let $y = h * x$. Then*

$$y_r(\ell) = \sum_{s \in R} \sum_{m \in L} h_{rs}(m)x_s(\ell - m).$$

Equivalently,

$$\mathcal{A}_L H \mathcal{A}_L^{-1} = (H_{rs})_{r,s \in R},$$

where H_{rs} is convolution on L by h_{rs}.

Proof. Evaluate the output at $r + \ell$. Write every input address as $s + \ell'$. The convolution offset is

$$(r + \ell) - (s + \ell') = (r - s) + (\ell - \ell').$$

Setting $m = \ell - \ell'$ gives the formula. □

Proposition E4.3 (Transfer norm control)**.** *For every output residue r,*

$$\sum_{s \in R} \|h_{rs}\|_{\ell^1(L)} = \|h\|_{\ell^1(\mathbb{Z}^d)}.$$

The packet transfer matrix acts on the polyphase direct sum with norm at most $\|h\|_1$ on every ℓ^p space.

Proof. For fixed r, every $q \in \mathbb{Z}^d$ has a unique representation $q = r - s + m$ with $s \in R$ and $m \in L$. This proves the equality of the absolute sums. The operator norm bound also follows by conjugating Young's inequality through the isometric analysis map. □

Proposition E4.4 (Composition of transfer matrices)**.** *If G and H are convolution operators with packet matrices (G_{rs}) and (H_{rs}), then the packet matrix of GH is*

$$(GH)_{rt} = \sum_{s \in R} G_{rs} H_{st}.$$

In one-dimensional transform notation, the corresponding matrix symbols multiply pointwise.

Proof. Conjugation by $\mathcal{A}_L$ preserves composition:

$$\mathcal{A}_L GH \mathcal{A}_L^{-1} = (\mathcal{A}_L G \mathcal{A}_L^{-1})(\mathcal{A}_L H \mathcal{A}_L^{-1}).$$

Reading the (r, t) block gives the formula. □

E4.1 The one-dimensional symbol

For $L = N\mathbb{Z}$ with $N \geq 1$, first take $|z| = 1$. The assumption $h \in \ell^1(\mathbb{Z})$ makes the following symbol series absolutely convergent. Define

$$H_{ks}(z) = \sum_{m \in \mathbb{Z}} h(Nm + k - s) z^{-m}.$$

For $x \in \ell^1(\mathbb{Z})$, the channel transforms converge absolutely on the unit circle and satisfy

$$\boldsymbol{Y}(z) = \boldsymbol{H}_N(z) \boldsymbol{X}(z).$$

For $x \in \ell^2(\mathbb{Z})$, the same identity holds almost everywhere in the L^2 Fourier sense. For finite impulse-response kernels it also holds at nonzero z wherever the input channel transforms are weighted absolutely summable. This is the classical polyphase matrix. It is the exact block matrix of the original convolution operator.

Corollary E4.5 (Finite impulse response closure)**.** *If h has finite support, every packet kernel h_{ks} has finite support. Matrix multiplication of polyphase symbols agrees with composition of the original finite impulse response operators.*

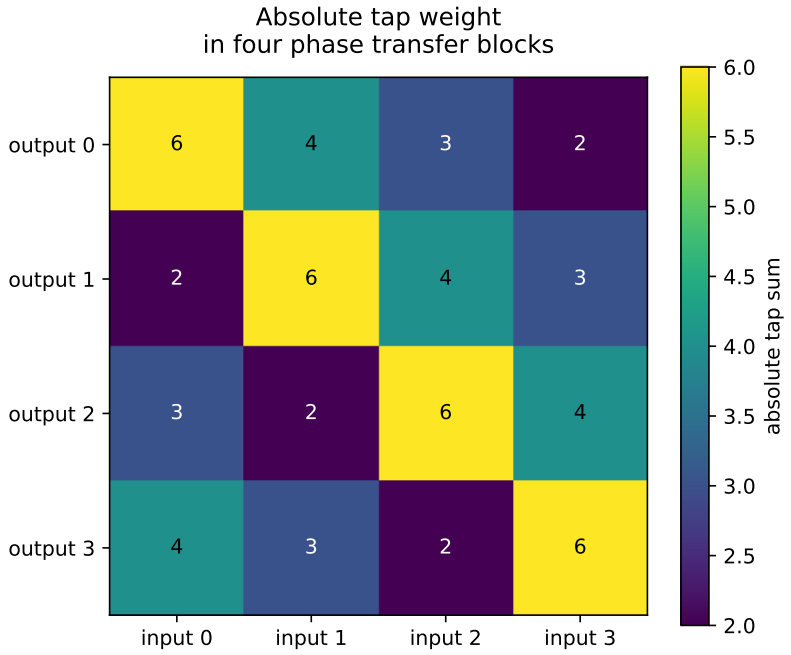


Figure E2. Absolute tap weight in the transfer blocks of the four phase operator in example E4.6.

Proof. Finite support is preserved by taking residue subsequences. The composition statement is proposition E4.4. □

Example E4.6 (A four phase finite impulse response operator)**.** Let

$$h(0) = 1, \quad h(1) = 2, \quad h(2) = 3, \quad h(3) = 4, \quad h(4) = 5,$$

with all other taps zero. For four residue channels, the transfer symbol is

$$\boldsymbol{H}_4(z) = \begin{pmatrix} 1 + 5z^{-1} & 4z^{-1} & 3z^{-1} & 2z^{-1} \\ 2 & 1 + 5z^{-1} & 4z^{-1} & 3z^{-1} \\ 3 & 2 & 1 + 5z^{-1} & 4z^{-1} \\ 4 & 3 & 2 & 1 + 5z^{-1} \end{pmatrix}.$$

The delay term records the tap that crosses a coarse cell boundary. No approximation is introduced by the matrix representation.

E5 Filter banks, frame bounds, and lifting

Throughout this section the banks act boundedly on the relevant polyphase ℓ^2 spaces. A finite channel analysis bank acts in Fourier coordinates by a matrix $E(z)$. A synthesis bank acts by a matrix $R(z)$. The matrices may be square or rectangular. Their entries are Laurent polynomials for finite filters and bounded measurable functions for general stable filters.

Theorem E5.1 (Perfect reconstruction criterion)**.** *Let $E(z)$ be an M by N analysis matrix and let $R(z)$ be an N by M synthesis matrix. The bank reconstructs every input with a delay of d coarse channel samples, equivalent to Nd original samples in the one-dimensional N-phase convention, exactly if and only if*

$$R(z)E(z) = z^{-d} I_N$$

for almost every point on the unit circle. A nonzero scalar factor gives exact reconstruction up to gain and delay.

Proof. The channel Fourier transform turns analysis followed by synthesis into multiplication by $R(z)E(z)$. If the matrix identity holds, inverse transformation gives the delayed input. Conversely, if every input is reconstructed with that delay, the difference matrix annihilates every vector valued Fourier datum and must vanish almost everywhere. □

Theorem E5.2 (Quantitative imperfect reconstruction)**.** *Define the delay compensated defect*

$$D(\mathrm{e}^{\mathrm{i}\omega}) = \mathrm{e}^{\mathrm{i}d\omega} R(\mathrm{e}^{\mathrm{i}\omega}) E(\mathrm{e}^{\mathrm{i}\omega}) - I_N.$$

If

$$\varepsilon_{\mathrm{rec}} = \operatorname*{ess\,sup}_{\omega} \|D(\mathrm{e}^{\mathrm{i}\omega})\|_{\mathrm{op}} < \infty,$$

then the delay compensated reconstruction $\tilde{x}$ satisfies

$$\|\widetilde{x} - x\|_2 \le \varepsilon_{\mathrm{rec}} \|x\|_2.$$

The constant is sharp as an operator norm.

Proof. In polyphase Fourier coordinates, the error is $D(\mathrm{e}^{\mathrm{i}\omega})\boldsymbol{X}(\mathrm{e}^{\mathrm{i}\omega})$. Pointwise multiplication and Plancherel's theorem give the upper bound. The operator norm of a measurable matrix multiplier is its essential supremum, which proves sharpness. □

Example E5.3 (A reconstruction certificate)**.** Suppose a two channel bank has delay compensated defect

$$D(\mathrm{e}^{\mathrm{i}\omega}) = \begin{pmatrix} 0.01\cos\omega & 0 \\ 0 & 0.02\sin\omega \end{pmatrix}.$$

Then

$$\operatorname*{ess\,sup}_{\omega} \|D(\mathrm{e}^{\mathrm{i}\omega})\|_{\mathrm{op}} = 0.02.$$

Every input therefore has relative reconstruction error at most two percent in ℓ^2. The estimate is independent of the input length.

Theorem E5.4 (Stability under filter perturbation)**.** *Assume $RE = z^{-d} I_N$. Let*

$$\widetilde{E} = E + \Delta E, \qquad \widetilde{R} = R + \Delta R.$$

Set

$$e_0 = \operatorname{ess\,sup} \|E\|_{\mathrm{op}}, \qquad r_0 = \operatorname{ess\,sup} \|R\|_{\mathrm{op}},$$
$$\delta_E = \operatorname{ess\,sup} \|\Delta E\|_{\mathrm{op}}, \quad \delta_R = \operatorname{ess\,sup} \|\Delta R\|_{\mathrm{op}}.$$

Then the delay compensated reconstruction defect obeys

$$\varepsilon_{\mathrm{rec}} \le r_0\delta_E + e_0\delta_R + \delta_R\delta_E.$$

Proof. Expand

$$\widetilde{R}\widetilde{E} - RE = R\Delta E + \Delta R E + \Delta R \Delta E.$$

Take pointwise operator norms, then the essential supremum. Multiplication by the delay factor has unit modulus. □

Theorem E5.5 (Frame bounds from the polyphase symbol)**.** *Assume that*

$$AI_N \le E(\mathrm{e}^{\mathrm{i}\omega})^* E(\mathrm{e}^{\mathrm{i}\omega}) \le BI_N$$

for almost every ω, where $0 < A \le B < \infty$. Then the analysis bank satisfies

$$A\|x\|_2^2 \le \|\mathcal{A}_E x\|_2^2 \le B\|x\|_2^2.$$

The pointwise matrix

$$R_{\mathrm{can}}(\mathrm{e}^{\mathrm{i}\omega}) = (E^*E)^{-1}E^*$$

is a bounded left inverse and gives the canonical dual synthesis bank.

Proof. Apply the matrix inequality to the polyphase Fourier vector at every frequency and integrate. Plancherel's theorem gives the frame inequalities. The lower bound makes E^*E invertible almost everywhere, and direct multiplication gives $R_{\mathrm{can}}E = I_N$. The same spectral bounds show that the canonical dual is bounded. □

Corollary E5.6 (Coefficient noise under canonical synthesis)**.** *Under the hypotheses of theorem E5.5, suppose the observed coefficient vector is*

$$c = \mathcal{A}_E x + \nu.$$

Canonical synthesis gives

$$\hat{x} = R_{\mathrm{can}} c$$

and

$$\|\hat{x} - x\|_2 \le \frac{1}{\sqrt{A}} \|\nu\|_2.$$

Proof. The smallest singular value of E is at least $\sqrt{A}$, so the pointwise norm of its pseudoinverse is at most $1/\sqrt{A}$. Apply Plancherel's theorem to $R_{\mathrm{can}}\nu$. □

Theorem E5.7 (Perturbation of frame bounds)**.** *Under the hypotheses of theorem E5.5, let $\widetilde{E} = E + \Delta E$ and suppose*

$$\delta = \operatorname*{ess\,sup}_{\omega} \|\Delta E(\mathrm{e}^{\mathrm{i}\omega})\|_{\mathrm{op}} < \sqrt{A}.$$

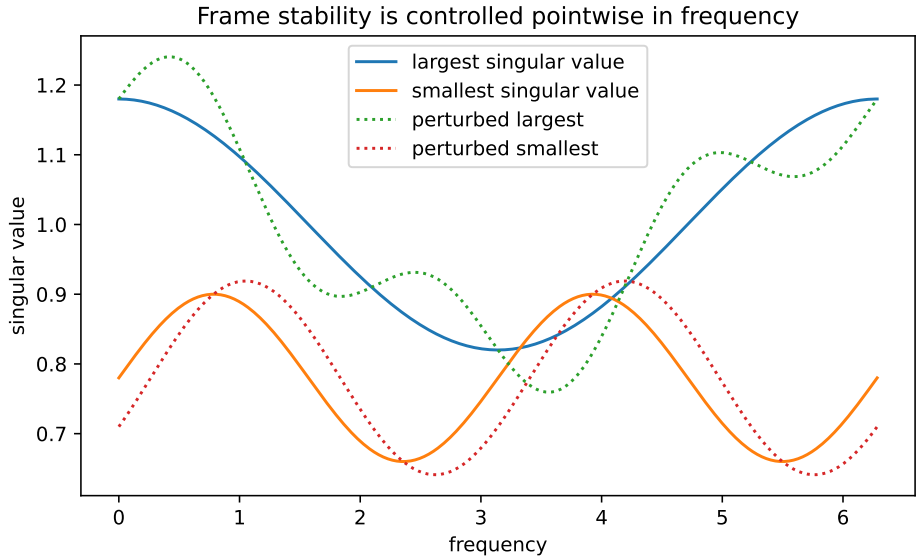


Figure E3. Pointwise singular value bounds govern both the original frame and its perturbed analysis bank.

Then $\widetilde{E}$ is a frame analysis bank with bounds

$$(\sqrt{A}-\delta)^2 \quad \textit{and} \quad (\sqrt{B}+\delta)^2.$$

Proof. For every vector v,

$$\|\widetilde{E}v\| \geq \|Ev\| - \|\Delta Ev\| \geq (\sqrt{A}-\delta)\|v\|.$$

The upper bound follows from

$$\|\widetilde{E}v\| \leq (\sqrt{B}+\delta)\|v\|.$$

Square the inequalities and integrate in frequency. □

Remark E5.8. Even when E is a Laurent polynomial matrix, the canonical dual need not have finite impulse response. Finite support of both analysis and synthesis is an additional algebraic condition.

Corollary E5.9 (Paraunitary bank)**.** *If $E(z)^*E(z) = I_N$ on the unit circle, then analysis is an isometry and synthesis may be chosen as $R(z) = E(z)^*$.*

Proof. Use theorem E5.5 with $A = B = 1$, then multiply the matrices. □

E5.1 Haar analysis

The Haar matrix is

$$E_H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

For even and odd samples $e(n) = x(2n)$ and $o(n) = x(2n+1)$, the low and high coordinates are

$$a(n) = \frac{e(n)+o(n)}{\sqrt{2}}, \qquad d(n) = \frac{e(n)-o(n)}{\sqrt{2}}.$$

Corollary E5.10 (Haar reconstruction and energy)**.** *The Haar bank reconstructs exactly and*

$$\|x\|_2^2 = \|a\|_2^2 + \|d\|_2^2.$$

Proof. The matrix E_H is unitary. □

E5.2 Lifting steps

Let $P(z)$ and $Q(z)$ be Laurent polynomials. Define

$$L_P(z) = \begin{pmatrix} 1 & 0 \\ P(z) & 1 \end{pmatrix}, \qquad U_Q(z) = \begin{pmatrix} 1 & Q(z) \\ 0 & 1 \end{pmatrix}.$$

Theorem E5.11 (Exact lifting update)**.** *The lifting matrices are invertible with*

$$L_P^{-1} = L_{-P}, \qquad U_Q^{-1} = U_{-Q}.$$

If $R_0E_0 = I$, then every finite product V of lifting matrices gives another exact bank by

$$E = VE_0, \qquad R = R_0V^{-1}.$$

Every lifting step has determinant one.

Proof. Direct multiplication proves the inverse formulas and the determinant statement. The reconstructed product is

$$RE = R_0V^{-1}VE_0 = R_0E_0 = I.$$

□

Example E5.12 (A two neighbor predict and update transform)**.** Split a sequence into even and odd samples. Define

$$d(n) = o(n) - \frac{e(n) + e(n+1)}{2}$$

and

$$s(n) = e(n) + \frac{d(n-1) + d(n)}{4}.$$

The inverse is obtained in reverse order:

$$e(n) = s(n) - \frac{d(n-1) + d(n)}{4},$$

then

$$o(n) = d(n) + \frac{e(n) + e(n+1)}{2}.$$

Thus reconstruction is exact over every coefficient field in which two is invertible. This is a standard compactly supported biorthogonal lifting pattern [CDF92, Swe96, DS98].

E6 Wavelet packet trees and admissible bases

A packet transform is built by deciding at each node whether to retain the current coordinate space or to split it into children. The proof of reconstruction is local and does not depend on a particular wavelet family.

Definition E6.1 (Unitary packet tree)**.** A finite unitary packet tree consists of Hilbert spaces $\mathcal{H}_v$ at its vertices and, at every internal vertex v, a unitary map

$$U_v : \mathcal{H}_v \longrightarrow \bigoplus_{w \in \mathrm{child}(v)} \mathcal{H}_w.$$

An admissible pruning is a set of vertices that meets every root to terminal path exactly once.

Theorem E6.2 (Reconstruction on a pruned packet tree)**.** *Every admissible pruning $\mathcal{L}$ defines a unitary analysis map*

$$\mathcal{A}_{\mathcal{L}} : \mathcal{H}_{\mathrm{root}} \longrightarrow \bigoplus_{v \in \mathcal{L}} \mathcal{H}_v.$$

Its inverse is obtained by applying the adjoints of the local splits from the leaves toward the root. In particular,

$$\|x\|^2 = \sum_{v \in \mathcal{L}} \|x_v\|^2.$$

Proof. Proceed by induction on the depth of the retained subtree. If the root is retained, the statement is immediate. If the root is split, apply the unitary map U_{root}, then apply the induction hypothesis independently in each child subtree. A direct sum of unitary maps is unitary. Reversing the maps with adjoints gives synthesis. □

Corollary E6.3 (Coefficient perturbations in a unitary tree)**.** *Let $c = \mathcal{A}_{\mathcal{L}} x$ be the leaf coefficients of an admissible unitary pruning, and let $\hat{c} = c + e$. If $\hat{x}$ is synthesized from $\hat{c}$, then*

$$\|\hat{x} - x\| = \|e\|.$$

Thus coefficient quantization and coefficient noise are neither amplified nor attenuated by exact unitary synthesis.

Proof. The synthesis map is unitary by theorem E6.2. Hence

$$\hat{x} - x = \mathcal{A}_{\mathcal{L}}^{-1} e$$

has the same norm as e. □

Proposition E6.4 (Bounds for nonunitary local splits)**.** *Suppose every internal node has an analysis map U_v and constants $0 \le A_v \le B_v < \infty$ satisfying*

$$A_v \|x\|^2 \le \sum_{w \in \mathrm{child}(v)} \|(U_v x)_w\|^2 \le B_v \|x\|^2.$$

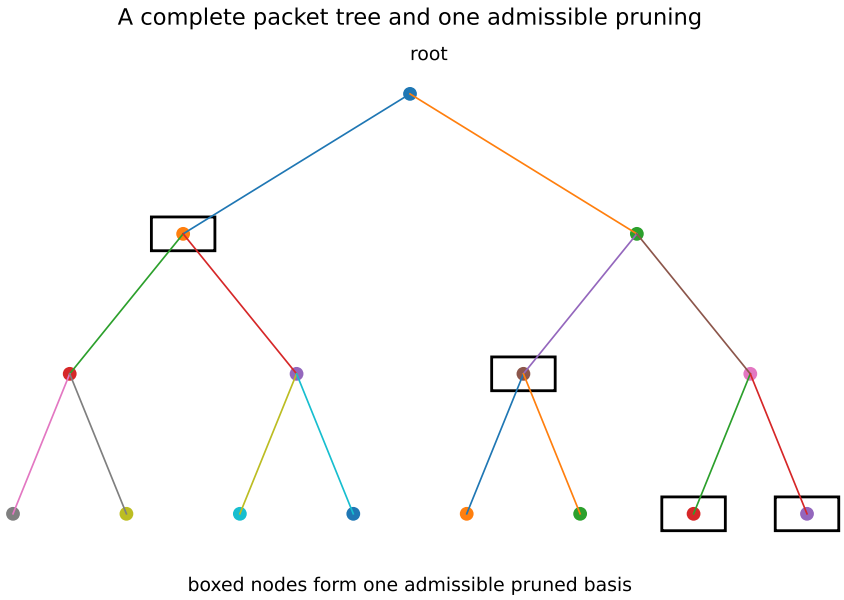


Figure E4. A complete packet tree and one admissible pruning. Retained nodes may occur at different depths.

Define recursively

$$a(v) = \begin{cases} 1, & v \text{ retained}, \\ A_v \min_w a(w), & v \text{ split}, \end{cases}$$

and

$$b(v) = \begin{cases} 1, & v \text{ retained}, \\ B_v \max_w b(w), & v \text{ split}. \end{cases}$$

Then the leaf analysis of the pruned tree has bounds $a(\mathrm{root})$ *and* $b(\mathrm{root})$.

Proof. Assume the child subtree bounds have been proved. Then

$$\sum_w \sum_{\ell \in \mathcal{L}_w} \|x_\ell\|^2 \geq \min_w a(w) \sum_w \|(U_v x)_w\|^2 \geq A_v \min_w a(w) \|x\|^2.$$

The upper bound follows in the same way with the maximum and B_v. Induction completes the proof. □

E6.1 Best basis recursion

> **The minimum is taken over feasible prunings**
>
> The value at a node is the least cost among all admissible prunings of the subtree rooted there. It is one optimal value. Individual competing prunings may have larger costs. The recursion compares retaining the node with the sum of the already optimal child values, with retention chosen on a tie.

Assign a real cost $c(v)$ to retaining each node. For an admissible pruning $\mathcal{L}$, let

$$C(\mathcal{L}) = \sum_{v \in \mathcal{L}} c(v).$$

The cost may measure entropy, coefficient count, an approximation penalty, or any other additive local objective.

Theorem E6.5 (Exact additive best basis recursion)**.** *For each vertex define*

$$F(v) = \min \left\{ c(v), \sum_{w \in \text{child}(v)} F(w) \right\},$$

with $F(v) = c(v)$ at terminal vertices. Then $F(v)$ is the minimum cost among all admissible prunings of the subtree rooted at v. A minimizing pruning is recovered by retaining v when the first term is no larger than the second and splitting it otherwise.

Proof. Every admissible pruning of the subtree has exactly two forms. It either retains v, with cost $c(v)$, or splits v and independently chooses an admissible pruning in every child subtree. The minimum in the second case is the sum of the child minima. This gives the recursion and the reconstruction rule. In an equality case both choices minimize, and the stated rule retains the node. □

Theorem E6.6 (Stability of the optimal cost)**.** *Let c and $\tilde{c}$ be two node cost functions. For an admissible pruning $\mathcal{L}$, set*

$$\varepsilon(\mathcal{L}) = \sum_{v \in \mathcal{L}} |c(v) - \tilde{c}(v)|$$

and let

$$\varepsilon_* = \max_{\mathcal{L}} \varepsilon(\mathcal{L}).$$

If F_* and $\widetilde{F}_*$ are the two optimal root costs, then

$$|F_* - \widetilde{F}_*| \le \varepsilon_*.$$

Proof. For every admissible pruning,

$$|C(\mathcal{L}) - \widetilde{C}(\mathcal{L})| \le \varepsilon_*.$$

Evaluate $\widetilde{C}$ on a minimizer for C to obtain $\widetilde{F}_* \le F_* + \varepsilon_*$. Reverse the roles to obtain $F_* \le \widetilde{F}_* + \varepsilon_*$. □

Corollary E6.7 (Stability of the selected basis)**.** *Assume the optimal pruning $\mathcal{L}_*$ for c is unique and at least one competing pruning exists. Define $g = \min_{\mathcal{L} \ne \mathcal{L}_*} (C(\mathcal{L}) - C(\mathcal{L}_*)) > 0$. If*

$$2\varepsilon_* < g,$$

then the optimal pruning for $\tilde{c}$ is the same. If only one admissible pruning exists, it remains selected under every cost change without a gap condition.

Proof. The cost of the original minimizer can increase by at most ε_*, while the cost of every competitor can decrease by at most ε_*. The original gap therefore remains positive. □

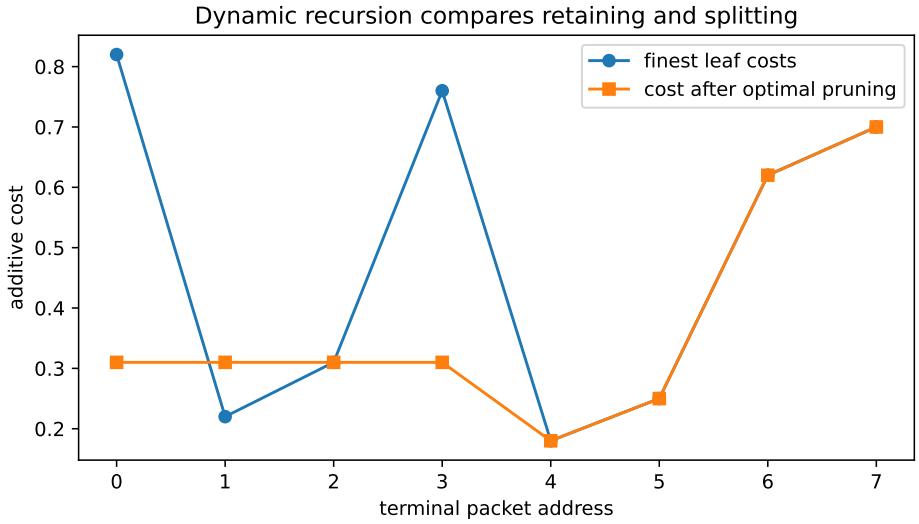


Figure E5. An additive best basis recursion compares the cost of retaining a packet with the total cost of splitting it.

E7 Singular packets and exact reconstruction

Trace data identify the singular part

A value jump needs a step packet. A jump in a derivative needs a truncated power of the corresponding order. Subtracting these finite packets matches the one-sided traces and leaves the regular remainder covered by the theorem's Sobolev assumptions. The packet construction and the approximation operator play separate roles.

A packet decomposition becomes useful for approximation when it separates a component that should be retained from a component that should be compressed. The simplest singular packets are steps. Higher order packets record jumps in derivatives.

For $m \geq 0$ and $0 < \tau < 1$, define

$$\phi_{m,\tau}(t) = \frac{(t-\tau)_+^m}{m!}.$$

Thus $\phi_{0,\tau} = \mathbf{1}_{[\tau,1]}$. On each side of τ, the packet is a polynomial. Its derivative of order m has a unit jump at τ.

Definition E7.1 (Sobolev step packet class)**.** Fix $r \geq 1$, $1 \leq p \leq \infty$, and an integer $Q \geq 0$. The class $\mathcal{F}_{r,p}(Q)$ consists of functions

$$f = s + \sum_{q=1}^{Q} a_q \phi_{0,\tau_q}$$

with $s \in W^{r,p}(0,1)$ and

$$0 < \tau_1 < \cdots < \tau_Q < 1.$$

No lower bound on the jump amplitudes is required for representation. Recovery results later impose such a bound.

Theorem E7.2 (Higher order singular packet decomposition). *Let*

$$0 = \tau_0 < \tau_1 < \cdots < \tau_Q < \tau_{Q+1} = 1.$$

Suppose the restriction of f to every interval (τ_q, τ_{q+1}) belongs to $W^{r,p}$, and choose the one-dimensional Sobolev representatives whose derivatives through order $r-1$ have one-sided traces at the endpoints. Define

$$a_{q,m} = \left[f^{(m)}\right]_{\tau_q} = f^{(m)}(\tau_q+) - f^{(m)}(\tau_q-)$$

and

$$j_f(t) = \sum_{q=1}^{Q} \sum_{m=0}^{r-1} a_{q,m} \phi_{m,\tau_q}(t).$$

Then

$$s_f = f - j_f$$

has matching one-sided derivatives through order $r-1$ at every breakpoint and belongs to $W^{r,p}(0,1)$. Its weak derivative of order r equals the piecewise classical derivative $f^{(r)}$ almost everywhere.

Proof. For $0 \le \ell \le m$, the classical derivative of $\phi_{m,\tau}$ on the two open sides is

$$\phi_{m,\tau}^{(\ell)}(t) = \frac{(t-\tau)_+^{m-\ell}}{(m-\ell)!}.$$

This derivative has zero jump when $\ell < m$ and a unit jump when $\ell = m$. Derivatives of order larger than m vanish on both open sides. It follows that subtracting the term $a_{q,m}\phi_{m,\tau_q}$ removes exactly the jump in the derivative of order m without changing the other classical trace jumps. Hence s_f has matching traces through order $r-1$.

On each partition interval, j_f is a polynomial of degree at most $r-1$, so its weak derivative of order r is zero there. A piecewise $W^{r,p}$ function whose traces through order $r-1$ match across every breakpoint belongs to $W^{r,p}(0,1)$. Repeated integration by parts on the partition intervals shows that all boundary terms cancel. Thus the weak derivative of order r is the piecewise weak derivative of f. □

Corollary E7.3 (Uniqueness of the trace packet). *Fix the breakpoints τ_q. If*

$$f = s + \sum_{q=1}^{Q} \sum_{m=0}^{r-1} b_{q,m} \phi_{m,\tau_q}$$

with $s \in W^{r,p}(0,1)$, then

$$b_{q,m} = \left[f^{(m)}\right]_{\tau_q}.$$

The finite truncated power packet is therefore the unique packet supported at the declared breakpoints that removes all trace jumps through order $r-1$.

Proof. Every representative in $W^{r,p}(0,1)$ has derivatives through order $r-1$ that admit absolutely continuous representatives on compact subintervals, with matching one-sided traces at each interior point. Taking the jump of the derivative of order m therefore removes the contribution of s. The packet ϕ_{m,τ_q}

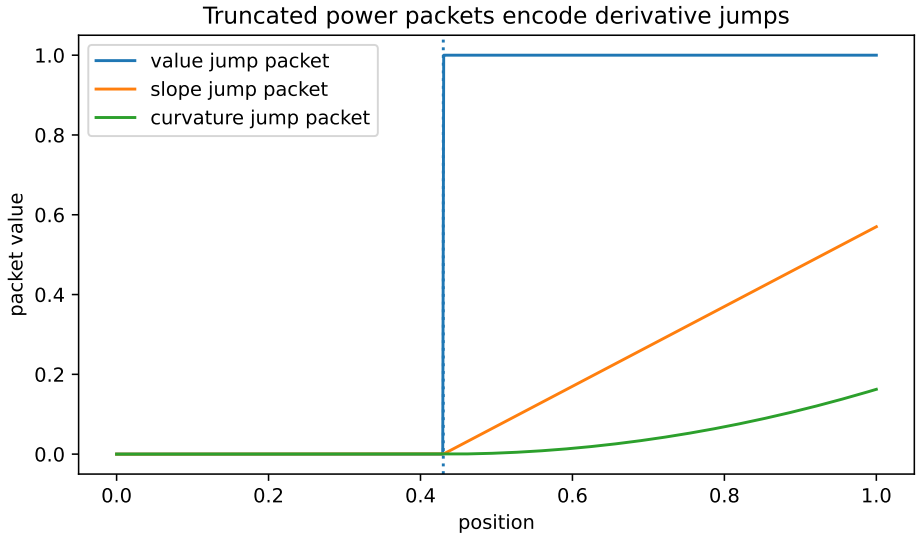


Figure E6. Truncated power packets encode jumps in a function and in its successive derivatives.

has unit jump in its derivative of order m, while all other declared packets have zero jump at that derivative order and breakpoint. Hence the remaining jump is exactly $b_{q,m}$. □

Remark E7.4. The theorem isolates only finitely many trace jumps. More general free discontinuity models may contain jump sets, Cantor parts, or singular derivatives that are not represented by a finite truncated power sum. The natural ambient theory is related to bounded variation and free discontinuity spaces [AFP00].

Let X be a normed linear space and let $Q : X \to X$ be bounded and linear. The next identity is elementary, but it is the central separation principle.

Theorem E7.5 (Exact retained packet identity). *Let $f = s + j$ in X. Define*

$$\tilde{f} = Qs + j.$$

Then

$$\boxed{f - \tilde{f} = (I - Q)s.}$$

Consequently,

$$\|f - \tilde{f}\| = \|s - Qs\|.$$

Proof. Subtract:

$$f - \tilde{f} = s + j - Qs - j = (I - Q)s.$$

□

Remark E7.6. The theorem assumes that the packet j is known. It does not supply a detector. It also does not say that every representation of f as $s + j$ is useful. Once a meaningful packet has been identified, its approximation error can be removed from the smooth approximation problem exactly.

Corollary E7.7 (Sobolev rate after packet separation). *Suppose Q_m satisfies*

$$\|s - Q_m s\|_{L^p} \le C 2^{-mr} \|s\|_{W^{r,p}}$$

for every $s \in W^{r,p}(0,1)$. If $f = s + j$ and j is retained, then

$$\|f - (Q_m s + j)\|_{L^p} \le C 2^{-mr} \|s\|_{W^{r,p}}.$$

The same conclusion applies to the higher order decomposition in theorem E7.2.

Proof. Use theorem E7.5 and the assumed approximation estimate. □

Theorem E7.8 (Reconstruction with an estimated packet). *Let $Q_m : X \to X$ be linear, let $f = s + j$, let $\hat{j}$ be an estimated packet, and define*

$$\hat{f}_m = Q_m(f - \hat{j}) + \hat{j}.$$

Then

$$\boxed{f - \hat{f}_m = (I - Q_m)s + (I - Q_m)(j - \hat{j}).}$$

If Q_m is bounded, then

$$\|f - \hat{f}_m\| \le \|s - Q_m s\| + (1 + \|Q_m\|)\|j - \hat{j}\|.$$

Proof. Since $f - \hat{j} = s + (j - \hat{j})$,

$$\hat{f}_m = Q_m s + Q_m(j - \hat{j}) + \hat{j}.$$

Subtract this expression from $s + j$. The norm bound follows from

$$\|(I - Q_m)u\| \le (1 + \|Q_m\|)\|u\|.$$

□

E8 Exact error formulas for estimated packets

The error of an estimated packet is often described by a segmentation score. For approximation, an exact norm formula is more useful because it separates value errors from geometric disagreement.

Theorem E8.1 (Exact scalar region formula). *Let E, F be measurable subsets of a measure space, let $a, b \in \mathbb{C}$, and let $1 \le p < \infty$. Then*

$$\|a\mathbf{1}_E - b\mathbf{1}_F\|_p^p = |a-b|^p \mu(E \cap F) + |a|^p \mu(E \setminus F) + |b|^p \mu(F \setminus E).$$

For equal amplitudes,

$$\|a\mathbf{1}_E - a\mathbf{1}_F\|_p = |a| \mu(E \triangle F)^{1/p}.$$

Proof. The sets $E \cap F$, $E \setminus F$, and $F \setminus E$ are disjoint. The difference takes the values $a - b$, a, and $-b$ on these three sets and is zero elsewhere. Integrating its pth power gives the formula. □

Corollary E8.2 (Location error for one step). *Let $0 \le \tau, \hat{\tau} \le 1$ and $H_\tau = \mathbf{1}_{[\tau,1]}$. For $1 \le p < \infty$,*

$$\|aH_\tau - aH_{\hat{\tau}}\|_p = |a| |\tau - \hat{\tau}|^{1/p}.$$

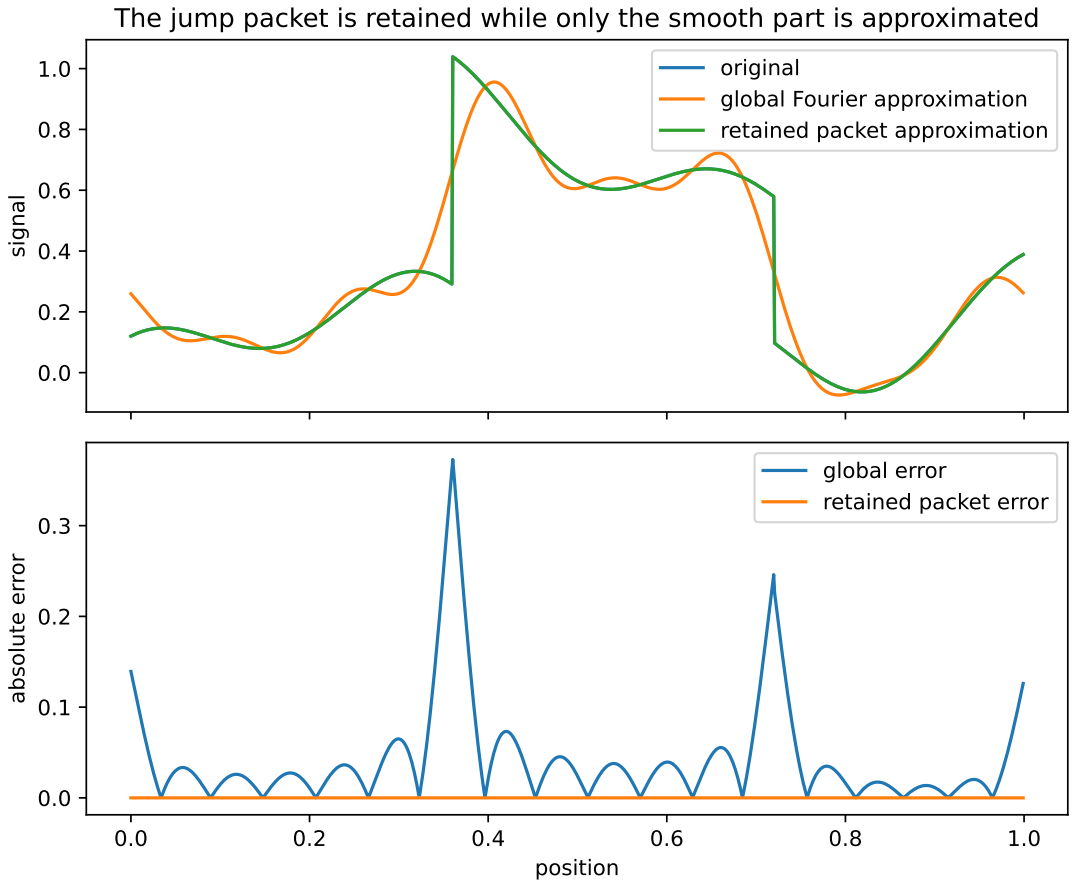


Figure E7. A global Fourier approximation must represent the jumps. A retained packet approximation applies the Fourier truncation only to the smooth component.

In the supremum norm, the error equals $|a|$ whenever $\tau \neq \hat{\tau}$.

Proof. The symmetric difference of the two intervals has length $|\tau - \hat{\tau}|$. The supremum statement follows because the two steps differ by a on a nonempty interval. □

Theorem E8.3 (Location stability for higher order packets). *Let $m \geq 1$, $1 \leq p < \infty$, and $0 \leq \tau, \hat{\tau} \leq 1$. Then*

$$\|\phi_{m,\tau} - \phi_{m,\hat{\tau}}\|_{L^p(0,1)} \leq \frac{|\tau - \hat{\tau}|}{(m-1)!((m-1)p+1)^{1/p}}.$$

For $p = \infty$,

$$\|\phi_{m,\tau} - \phi_{m,\hat{\tau}}\|_{\infty} \leq \frac{|\tau - \hat{\tau}|}{(m-1)!}.$$

Thus location error is Lipschitz for every derivative jump packet of positive order, while a value jump has only the finite L^p modulus in corollary E8.2.

Proof. Assume $\tau < \hat{\tau}$. Differentiation with respect to the location parameter gives

$$\frac{\partial}{\partial u}\phi_{m,u}(t) = -\phi_{m-1,u}(t)$$

for almost every t. Hence

$$\phi_{m,\tau} - \phi_{m,\hat{\tau}} = \int_{\tau}^{\hat{\tau}} \phi_{m-1,u}\,\mathrm{d}u.$$

Minkowski's inequality gives

$$\|\phi_{m,\tau} - \phi_{m,\hat{\tau}}\|_p \leq |\tau - \hat{\tau}| \sup_u \|\phi_{m-1,u}\|_p.$$

The supremum is bounded by the norm of $t^{m-1}/(m-1)!$ on $[0,1]$, which is the stated constant. The supremum norm argument is identical. □

E8.1 Several regions

Let $(E_i)_{i=0}^{I}$ and $(F_j)_{j=0}^{J}$ be measurable partitions of the same domain, up to null sets. Let a_i and b_j be the corresponding values. The index zero may be used for a background region.

Theorem E8.4 (Overlap matrix formula). *Define*

$$M_{ij} = \mu(E_i \cap F_j).$$

For $1 \leq p < \infty$,

$$\left\| \sum_{i=0}^{I} a_i \mathbf{1}_{E_i} - \sum_{j=0}^{J} b_j \mathbf{1}_{F_j} \right\|_p^p = \sum_{i=0}^{I} \sum_{j=0}^{J} |a_i - b_j|^p M_{ij}.$$

The row sums of M are $\mu(E_i)$, and the column sums are $\mu(F_j)$.

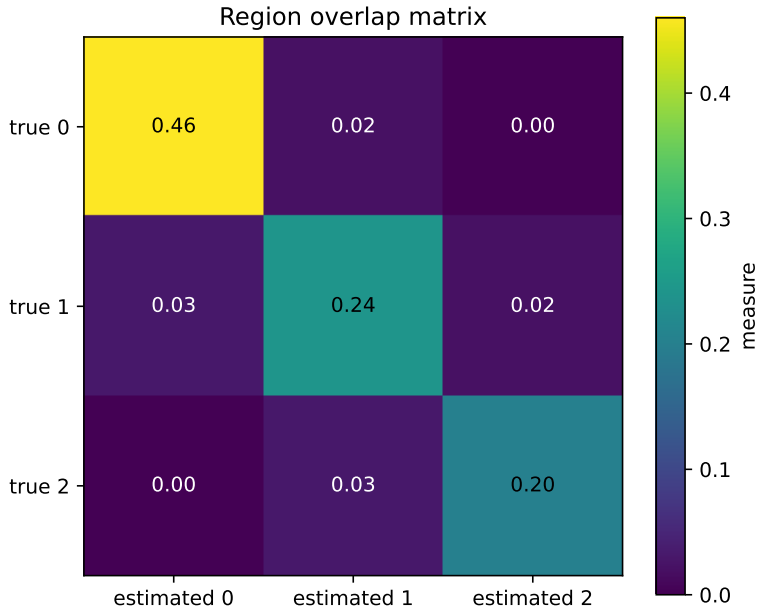


Figure E8. An overlap matrix records both correct regions and every type of misclassification. The exact packet norm weights each entry by the corresponding value difference.

Proof. The intersections $E_i \cap F_j$ form a measurable partition. On such an intersection, the two piecewise constant functions have values a_i and b_j. Integrate the constant error over every intersection. The marginal identities follow by summing the partition pieces in a fixed row or column. □

Corollary E8.5 (Background and packet error)**.** *Suppose the true and estimated packets use the same amplitude list $(a_i)_{i=0}^{I}$, but possibly different regions. Then*

$$\|j - \hat{j}\|_p^p = \sum_{i,k} |a_i - a_k|^p \mu(E_i \cap F_k).$$

Only off-diagonal overlaps contribute.

Proof. Apply theorem E8.4 with $b_k = a_k$. The diagonal terms vanish. □

E8.2 Boundary tube control

For a closed set Γ, write

$$N_\delta(\Gamma) = \{x : \operatorname{dist}(x, \Gamma) < \delta\}.$$

Definition E8.6 (Linear tube bound)**.** A boundary Γ has a linear tube bound if there are constants C_Γ and $\delta_0 > 0$ such that

$$\mu(N_\delta(\Gamma)) \le C_\Gamma \delta$$

for $0 < \delta < \delta_0$.

Bounded Lipschitz domains and many finite perimeter sets have estimates of this form, with constants controlled by geometric size. The next theorem only requires the displayed bound.

Theorem E8.7 (Geometric packet bound)**.** *Let E be a region with boundary Γ, and suppose*

$$E \triangle F \subseteq N_\delta(\Gamma).$$

If Γ has a linear tube bound and $0 < \delta < \delta_0$ lies in its stated range, then for $1 \le p < \infty$,

$$\|a\mathbf{1}_E - a\mathbf{1}_F\|_p \le |a|(C_\Gamma \delta)^{1/p}.$$

If amplitudes a and b differ, then

$$\|a\mathbf{1}_E - b\mathbf{1}_F\|_p \le |a-b|\mu(E \cap F)^{1/p} + (|a| + |b|)(C_\Gamma \delta)^{1/p}.$$

Proof. The equal amplitude case follows from theorem E8.1 and the tube inclusion. For unequal amplitudes, split the error into its value mismatch on $E \cap F$ and the two disagreement pieces. Apply the triangle inequality and the same tube estimate. □

Example E8.8 (A perturbed disk)**.** Let E and F be concentric disks of radii r and $r+\delta$ in $\mathbb{R}^2$, with $\delta > 0$. Then

$$\mu(E \triangle F) = \pi\big((r+\delta)^2 - r^2\big) = 2\pi r\delta + \pi\delta^2.$$

For equal amplitude a, the exact L^2 packet error is

$$|a|\sqrt{2\pi r\delta + \pi\delta^2}.$$

This gives the expected square root dependence on boundary displacement.

E9 Stable recovery of separated jumps

The detector needs a visible interval of thresholds

A smooth cell and bounded sample noise give one upper contrast bound. A jump cell gives a lower contrast bound after the same uncertainty is subtracted. Recovery is possible when the lower jump bound exceeds the upper smooth bound. The gap is a condition on the observation model, rather than an automatic consequence of packet coordinates.

The previous sections describe the error after a packet has been estimated. We now give a direct recovery theorem for one-dimensional separated jumps. The detector is intentionally simple because the aim is to expose the exact contrast gap.

Let

$$t_k = kh, \qquad k = 0, \dots, n, \qquad h = \frac{1}{n}.$$

Assume

$$f(t) = s(t) + \sum_{q=1}^{Q} a_q H_{\tau_q}(t), \qquad H_\tau(t) = \mathbf{1}_{[\tau,1]}(t),$$

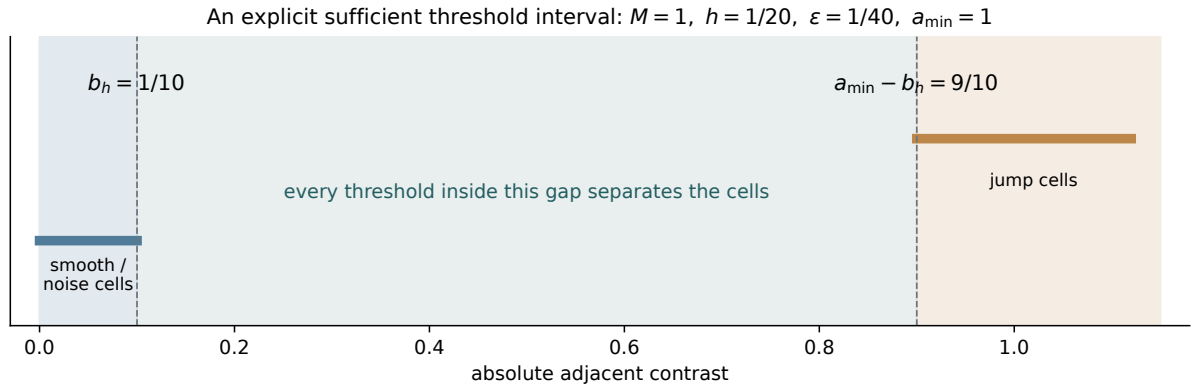


Figure E9. A sufficient threshold interval from exact rational contrast bounds. With the displayed model parameters, smooth and noise contrasts are at most $1/10$ and jump contrasts are at least $9/10$. Every threshold strictly between those values separates the cells under the sampling assumptions stated below. The figure does not assert an optimal threshold for other observation models.

where s is Lipschitz with constant M, and $Q \geq 1$ is finite with nonzero jump amplitudes. The zero-jump case requires no minimum-amplitude condition. Suppose no jump lies at a sample point and no grid cell contains more than one jump. Observe

$$y_k = f(t_k) + \eta_k.$$

Define the adjacent contrast

$$D_k = y_{k+1} - y_k.$$

Lemma E9.1 (Contrast gap)**.** *Assume* $|\eta_k| \leq \varepsilon$ *and set*

$$b_h = Mh + 2\varepsilon.$$

If the cell (t_k, t_{k+1}) *contains no jump, then*

$$|D_k| \leq b_h.$$

If it contains one jump of amplitude a_q*, then*

$$|D_k - a_q| \leq b_h$$

and therefore

$$|D_k| \geq |a_q| - b_h.$$

Proof. In a cell without a jump,

$$D_k = s(t_{k+1}) - s(t_k) + \eta_{k+1} - \eta_k.$$

The Lipschitz and noise bounds give $|D_k| \leq Mh + 2\varepsilon$. In a cell with one jump, the same formula has the additional term a_q, which proves the second estimate. □

Theorem E9.2 (Deterministic jump cell recovery)**.** *Let*

$$a_{\min} = \min_q |a_q|$$

and assume

$$a_{\min} > 2b_h.$$

Choose a threshold θ satisfying

$$b_h < \theta < a_{\min} - b_h.$$

Then

$$\widehat{\mathcal{K}} = \{k : |D_k| > \theta\}$$

is exactly the set of grid cells containing jumps. If $\hat{\tau}_q$ is the midpoint of the detected cell and $\hat{a}_q = D_k$, then after matching by order,

$$|\hat{\tau}_q - \tau_q| \le \frac{h}{2}$$

and

$$|\hat{a}_q - a_q| \le b_h.$$

Proof. By lemma E9.1, every cell without a jump has contrast at most $b_h < \theta$. Every cell with a jump has contrast at least $a_{\min} - b_h > \theta$. Hence the detected set is exact. The midpoint of a cell is within $h/2$ of every point in that cell. The amplitude estimate is the second inequality in lemma E9.1. □

Corollary E9.3 (Jump count and sign recovery)**.** *Under the hypotheses of theorem E9.2, the number of detected cells equals Q, and*

$$\operatorname{sgn}(\hat{a}_q) = \operatorname{sgn}(a_q)$$

for every real valued jump.

Proof. Exact cell recovery gives the jump count. Since

$$|\hat{a}_q - a_q| \le b_h < |a_q|,$$

the amplitude estimate cannot cross zero. □

Corollary E9.4 (Packet reconstruction after detection)**.** *Let*

$$j = \sum_{q=1}^{Q} a_q H_{\tau_q}, \qquad \hat{j} = \sum_{q=1}^{Q} \hat{a}_q H_{\hat{\tau}_q}.$$

Under the hypotheses of theorem E9.2, for $1 \le p < \infty$,

$$\|j - \hat{j}\|_p \le Q b_h + \left(\frac{h}{2} \sum_{q=1}^{Q} |a_q|^p \right)^{1/p}.$$

If Q_m is bounded and linear and

$$\hat{f}_m = Q_m(f - \hat{j}) + \hat{j},$$

then

$$\|f-\hat{f}_m\|_p \le \|s-Q_m s\|_p + (1+\|Q_m\|)\left[Qb_h+\left(\frac{h}{2}\sum_q |a_q|^p\right)^{1/p}\right].$$

Proof. Write

$$j-\hat{j}=\sum_q (a_q-\hat{a}_q)H_{\hat{\tau}_q}+\sum_q a_q(H_{\tau_q}-H_{\hat{\tau}_q}).$$

The first sum has norm at most $\sum_q |a_q-\hat{a}_q| \le Qb_h$. The location error interval for the qth jump is contained in its detected grid cell and has length at most $h/2$. Distinct detected cells are disjoint, so the location errors have disjoint supports. Their pth powers therefore add exactly, which gives the displayed location term. The reconstruction estimate follows from theorem E7.8. □

Remark E9.5 (Why finite L^p matters). A shifted step does not converge in the supremum norm until the location is exact. The finite L^p error does converge because the disagreement is confined to an interval whose length tends to zero.

Corollary E9.6 (Deterministic consistency)**.** *Assume $h \to 0$ along grids satisfying the preceding sampling assumptions, $\varepsilon_h \to 0$, the number of jumps is fixed, and the nonzero jump amplitudes and positive minimum separation remain fixed. Then the detected packet converges to the true packet in every finite L^p norm. If Q_m has a vanishing smooth approximation error and uniformly bounded norm, the complete reconstruction also converges in L^p.*

Proof. The gap condition holds for sufficiently small h. Then $b_h \to 0$, and the bound in corollary E9.4 tends to zero. □

Example E9.7 (A numerical threshold certificate)**.** Take $h=1/256$, $M=0.5$, and $\varepsilon=0.002$. Then

$$b_h=\frac{0.5}{256}+0.004<0.00596.$$

If every jump has magnitude at least 0.08, then $0.08>2b_h$. Any threshold between 0.00596 and 0.07404 recovers the exact jump cells. The location error is at most

$$\frac{1}{512}$$

and every amplitude error is less than 0.00596.

E9.1 Sub-Gaussian noise

Assume the noise variables are independent, centered, and σ sub-Gaussian:

$$\mathbb{E}\mathrm{e}^{\lambda\eta_k} \le \mathrm{e}^{\sigma^2\lambda^2/2}.$$

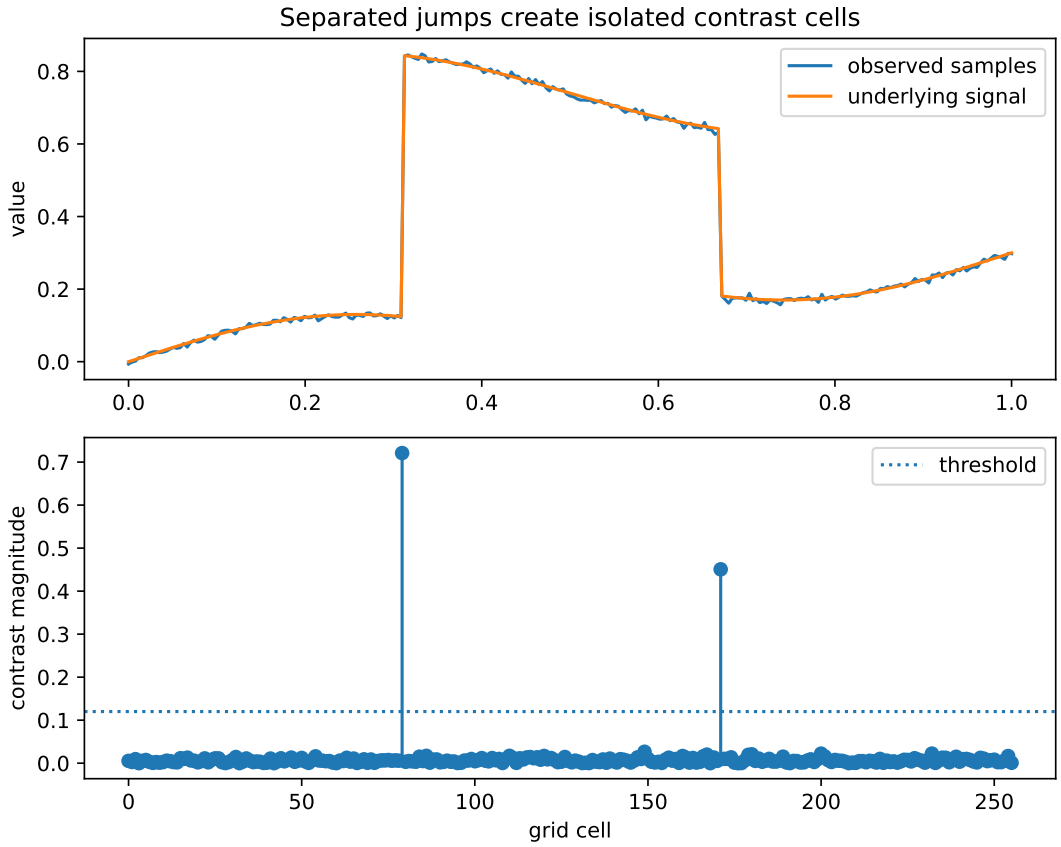


Figure E10. Separated jumps create isolated adjacent sample contrasts above the smooth and noise background.

Theorem E9.8 (Probability recovery bound)**.** *Fix $0 < \delta < 1$ and define*

$$t_{n,\delta} = 2\sigma\sqrt{\log\left(\frac{2n}{\delta}\right)}$$

and

$$b_{h,\delta} = Mh + t_{n,\delta}.$$

With probability at least $1 - \delta$, every contrast satisfies the conclusions of lemma E9.1 with b_h replaced by $b_{h,\delta}$.
If

$$a_{\min} > 2b_{h,\delta},$$

then thresholding between $b_{h,\delta}$ and $a_{\min} - b_{h,\delta}$ recovers every jump cell with probability at least $1 - \delta$. The same location and amplitude bounds hold with $b_{h,\delta}$.

Proof. If $\sigma = 0$, every centered zero-sub-Gaussian noise variable vanishes almost surely, and the noise-free deterministic conclusion applies. For $\sigma > 0$, the difference $\eta_{k+1} - \eta_k$ is sub-Gaussian with variance proxy $2\sigma^2$, so

$$\mathbb{P}(|\eta_{k+1} - \eta_k| > t) \le 2\exp\left(-\frac{t^2}{4\sigma^2}\right).$$

A union bound over the n cells gives

$$\mathbb{P}\left(\max_k |\eta_{k+1} - \eta_k| > t_{n,\delta}\right) \le \delta.$$

On the complementary event, repeat the deterministic proof. □

Corollary E9.9 (Repeated measurements)**.** *Suppose R observations are averaged at every sample location, with all noise variables independent across sites and repetitions and each centered and σ sub-Gaussian. Then theorem E9.8 holds with σ replaced by $\sigma/\sqrt{R}$. In particular, a sufficient recovery condition is*

$$a_{\min} > 2Mh + \frac{4\sigma}{\sqrt{R}}\sqrt{\log\left(\frac{2n}{\delta}\right)}.$$

Proof. The average of R independent σ sub-Gaussian variables is $\sigma/\sqrt{R}$ sub-Gaussian. Substitute this parameter in theorem E9.8. □

Remark E9.10. The displayed uniform noise certificate scales as $\sigma\sqrt{\log n}$. At fixed positive σ, it does not imply consistency of this detector. Decreasing the noise scale or averaging repeated measurements gives sufficient regimes. Other averaging schemes require their own analysis.

Corollary E9.11 (Probability consistency)**.** *Let $\sigma_n\sqrt{\log n} \to 0$ and $h = 1/n$. For every fixed $0 < \delta < 1$, along grids satisfying the preceding sampling assumptions, all jump cells are recovered with probability at least $1 - \delta$ for sufficiently large n. For the fixed finite jump model above, along grids satisfying the preceding sampling assumptions, the packet error converges to zero in probability in every finite L^p norm.*

Proof. For each fixed $\delta > 0$, $b_{h,\delta} \to 0$. Apply theorem E9.8 and corollary E9.4. For any error tolerance $\epsilon > 0$, the high-probability bound is eventually smaller than ϵ, so the limit superior of the probability that the packet error exceeds ϵ is at most δ. Since δ is arbitrary, convergence is in probability. □

E9.2 Signals with values in a normed space

The recovery proof does not depend on an ordering of the signal values.

Proposition E9.12 (Vector jump recovery). *Let the values lie in a normed space V. Assume*

$$\|s(t) - s(u)\|_V \le M|t-u|, \qquad \|\eta_k\|_V \le \varepsilon,$$

and let the jump vectors satisfy $\|a_q\|_V \ge a_{\min}$. Thresholding $\|D_k\|_V$ gives the same recovery theorem under the condition

$$a_{\min} > 2(Mh + 2\varepsilon).$$

Proof. Only the triangle inequality was used in lemma E9.1 and theorem E9.2. □

E10 Haar localization and annihilating differences

The adjacent difference detector is local in the sample domain. Wavelet coefficients and higher differences give related multiscale packets. In the Haar subsections the signal is defined on $\mathbb{R}$. Here the step convention is $H_\tau(t) = \mathbf{1}_{[\tau,\infty)}(t)$, whose restriction to $(0,1)$ agrees with the preceding interval step. A chosen extension beyond the observation interval must use this half-line step convention, and the smooth extension must satisfy the stated Lipschitz hypothesis on every displayed wavelet support.

E10.1 Haar coefficient separation

Let

$$I_{j,k} = [k2^{-j}, (k+1)2^{-j})$$

and let $\psi_{j,k}$ be the normalized Haar wavelet supported on $I_{j,k}$, positive on the left half and negative on the right half.

Theorem E10.1 (Smooth Haar coefficient bound). *If s is Lipschitz with constant M, then*

$$|\langle s, \psi_{j,k}\rangle| \le \frac{M}{4}\, 2^{-3j/2}.$$

Proof. Let $L = 2^{-j}$ and let a be the left endpoint of the support. Pair every point $t \in [a, a + L/2]$ with $t + L/2$. Then

$$\langle s, \psi_{j,k}\rangle = L^{-1/2} \int_a^{a+L/2} \overline{s(t) - s(t + L/2)}\ \mathrm{d}t.$$

The integrand has magnitude at most $ML/2$, and the integration interval has length $L/2$. The result is $ML^{3/2}/4$. □

Theorem E10.2 (Exact Haar coefficient of a step). *Let $\tau \in I_{j,k}$, let $L = 2^{-j}$, and write*

$$u = \frac{\tau - kL}{L} \in [0,1].$$

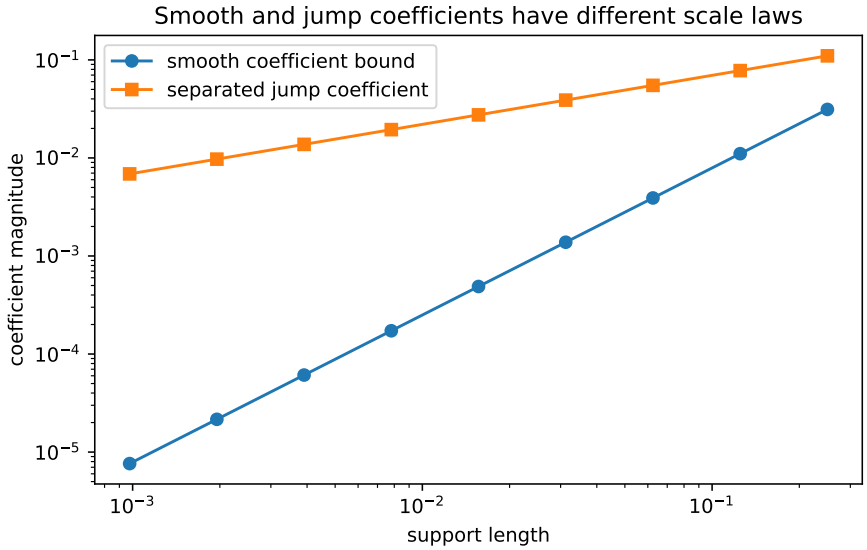


Figure E11. The smooth contribution decays like $L^{3/2}$, while a separated jump contribution decays like $L^{1/2}$.

Then

$$\left|\langle aH_\tau, \psi_{j,k}\rangle\right| = |a| L^{1/2} \min\{u, 1-u\}.$$

Every Haar coefficient whose support does not cross τ is zero.

Proof. Away from the jump, the step is constant on the support and the Haar wavelet has zero mean. When the jump lies in the support, integrate the positive and negative halves directly. If $u \le 1/2$, the magnitude is $|a| u L^{1/2}$. If $u \ge 1/2$, it is $|a|(1-u)L^{1/2}$. □

Corollary E10.3 (Scale separation). *Suppose $f = s + aH_\tau + \eta$, where s is M Lipschitz and $\|\eta\|_\infty \le \varepsilon$. If the jump lies at relative distance at least ρ from both ends of $I_{j,k}$, then*

$$|\langle f, \psi_{j,k}\rangle| \ge \left(|a|\rho - \varepsilon - \frac{M}{4} L\right) L^{1/2}.$$

A wavelet support without a jump satisfies

$$|\langle f, \psi_{j,k}\rangle| \le \left(\varepsilon + \frac{M}{4} L\right) L^{1/2}.$$

Thus a threshold gap exists whenever

$$|a|\rho > 2\varepsilon + \frac{M}{2} L.$$

Proof. Combine theorems E10.1 and E10.2. The noise coefficient is at most

$$\|\eta\|_\infty \|\psi_{j,k}\|_1 = \varepsilon L^{1/2}.$$

The final inequality makes the lower bound on a jump support larger than the upper bound on a smooth support. □

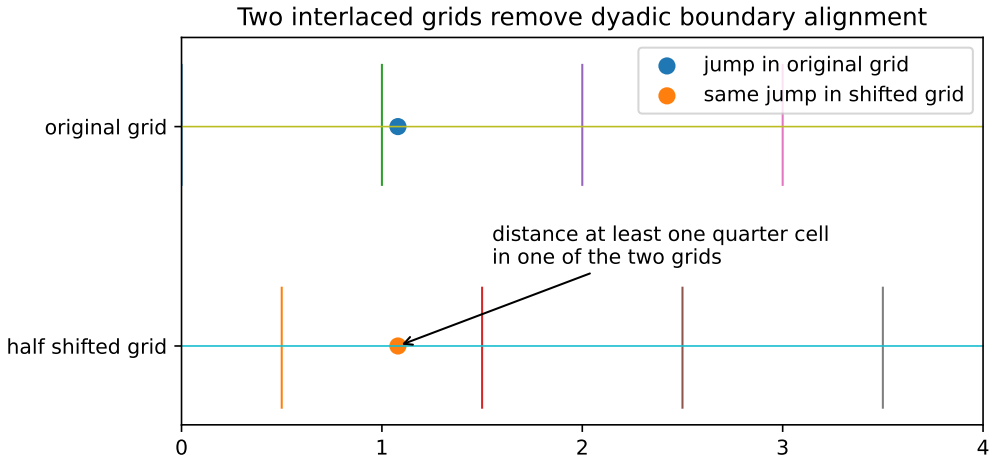


Figure E12. The boundaries of the two grids interlace. Every jump is well inside a cell in at least one grid.

E10.2 Two shifted grids

A jump on a dyadic boundary can be invisible to a wavelet at that scale. One additional grid removes this exact alignment.

Theorem E10.4 (Half cell shift guarantee). *Fix a cell length $L > 0$. Consider the grid with boundaries kL and the grid with boundaries $(k + 1/2)L$. Every point $\tau \in \mathbb{R}$ lies in a cell of at least one grid at distance at least $L/4$ from both cell boundaries.*

Proof. Reduce τ modulo L and write its representative as $x \in [0, L)$. If

$$L/4 \le x \le 3L/4,$$

then the distance from x to both boundaries of its cell in the original grid is at least $L/4$. If $x < L/4$ or $x > 3L/4$, then x lies in the cell of the shifted grid bounded by $-L/2$ and $L/2$, or in its translate, at distance at least $L/4$ from both shifted boundaries. One of the two grids therefore has the required property. □

Corollary E10.5 (Alignment free two grid Haar detection). *Assume the support length L is smaller than the minimum jump separation. Analyze the signal with Haar wavelets on the original grid and on the half shifted grid. Every jump appears in a support with $\rho \ge 1/4$ in one of the two systems. Therefore a threshold gap exists independently of alignment whenever*

$$\frac{|a|}{4} > 2\varepsilon + \frac{M}{2}L.$$

Proof. Apply theorem E10.4 and then corollary E10.3 with $\rho = 1/4$. The separation assumption prevents a selected support from containing two jumps. □

E10.3 Higher differences

Let

$$(\Delta x)(n) = x(n + 1) - x(n).$$

Then

$$\Delta^m x(n) = \sum_{r=0}^{m} (-1)^{m-r} \binom{m}{r} x(n+r).$$

Theorem E10.6 (Polynomial annihilation and boundary support)**.** *Let x be a sequence that agrees with a polynomial of degree at most d on each interval of a finite partition of $\mathbb{Z}$. Then*

$$\Delta^{d+1} x(n) = 0$$

whenever the complete stencil

$$\{n, n+1, \dots, n+d+1\}$$

lies inside one polynomial piece. Hence $\Delta^{d+1} x$ is supported in a band of at most $d+1$ addresses before each breakpoint.

Proof. The $(d+1)$st forward difference of a polynomial of degree at most d is zero. The only stencils for which this argument does not apply are those crossing a breakpoint. □

Proposition E10.7 (Noise amplification by differences)**.** *If $\|\eta\|_\infty \le \varepsilon$, then*

$$\|\Delta^m \eta\|_\infty \le 2^m \varepsilon.$$

Therefore no false response above $2^{d+1}\varepsilon$ can occur on a stencil lying inside one polynomial piece.

Proof. The absolute values of the binomial coefficients in Δ^m sum to 2^m. □

Corollary E10.8 (Robust breakpoint bands)**.** *Assume every true breakpoint produces at least one response of magnitude ρ in its $(d+1)$st difference band. If*

$$\rho > 2^{d+2}\varepsilon,$$

then thresholding at any level between $2^{d+1}\varepsilon$ and $\rho - 2^{d+1}\varepsilon$ detects a nonempty cluster near every breakpoint and creates no cluster away from the breakpoint bands.

Proof. Use theorem E10.6 and proposition E10.7. Noise cannot cross the threshold away from a breakpoint. At a response of magnitude ρ, the perturbed magnitude remains at least $\rho - 2^{d+1}\varepsilon$, which is above the threshold. □

E11 Packet separation and the Gibbs phenomenon

A Fourier partial sum is global. When the target has a jump, the partial sum must approximate that jump using continuous trigonometric polynomials. This creates the classical Gibbs oscillation. Standard accounts and recovery methods appear in [Zyg02, GS97].

Standard Fact E11.1 (Classical square-wave Gibbs behavior). Let $q(x) = \frac{1}{2}\operatorname{sign}(\sin x)$, whose jump at zero has size one. For $n \ge 1$, its partial sum through the first n positive odd frequencies is

$$Q_n(x) = \frac{2}{\pi} \sum_{k=0}^{n-1} \frac{\sin((2k+1)x)}{2k+1}.$$

At $x_n = \pi/(2n)$,

$$x_n \longrightarrow 0, \qquad Q_n(x_n) \longrightarrow \frac{\mathrm{Si}(\pi)}{\pi} = \frac{1}{2} + G, \qquad G \approx 0.089489872236,$$

where $\mathrm{Si}(z) = \int_0^z \sin(t)/t\, dt$ and $G = \mathrm{Si}(\pi)/\pi - 1/2 > 0$. Thus a nonzero overshoot persists along points approaching the jump. Scaling by a real amplitude a gives an overshoot or undershoot of magnitude $|a|G$.

For the limit, put $u_k = (k + \frac{1}{2})/n$. The identity

$$Q_n(x_n) = \frac{1}{\pi n} \sum_{k=0}^{n-1} \frac{\sin(\pi u_k)}{u_k} \longrightarrow \frac{1}{\pi} \int_0^1 \frac{\sin(\pi u)}{u}\, du = \frac{\mathrm{Si}(\pi)}{\pi}$$

is a midpoint Riemann sum, since the integrand has continuous value π at zero. This is the normalized classical example in [Nat26, Section 6.16(i), equations 6.16.1–6.16.4].

The square-wave example supplies a classical comparison. The following retained-packet identities do not require a general overshoot theorem. They change the approximation problem because the discontinuity is no longer represented by the Fourier partial sum.

Theorem E11.2 (Uniform convergence with exact jump packets)**.** *Let $f = s + j$ on a compact domain, where j is any bounded singular packet. Suppose $Q_m s \to s$ uniformly. Then*

$$\tilde{f}_m = Q_m s + j$$

converges uniformly to f, even when f is discontinuous:

$$\|f - \tilde{f}_m\|_\infty = \|s - Q_m s\|_\infty \longrightarrow 0.$$

There is no error proportional to the jump size because the jump is shared exactly by f and $\tilde{f}_m$.

Proof. Apply theorem E7.5 in the supremum norm. □

Theorem E11.3 (Estimated packet Gibbs bound). *Let Q_m be linear and uniformly bounded on L^p, with*

$$\sup_m \|Q_m\| \le C_Q.$$

For $f = s + j$ and an estimated packet $\hat{j}_m$, define

$$\hat{f}_m = Q_m(f - \hat{j}_m) + \hat{j}_m.$$

Then

$$\|f - \hat{f}_m\|_p \le \|s - Q_m s\|_p + (1 + C_Q)\|j - \hat{j}_m\|_p.$$

Consequently, smooth approximation and packet recovery may be improved independently. No claim of Gibbs removal is justified unless both terms are controlled in the norm under consideration.

Proof. This is theorem E7.8 with the uniform operator bound. □

Corollary E11.4 (Boundary band error). *Let Γ be the singular boundary and assume*

$$\mu(N_\delta(\Gamma)) \le C_\Gamma \delta$$

for small δ. For finite p,

$$\|f - \tilde{f}_m\|_{L^p(N_\delta(\Gamma))} \le (C_\Gamma \delta)^{1/p} \|s - Q_m s\|_\infty.$$

Proof. The error equals $s - Q_m s$. Bound it by its supremum and multiply by the measure of the boundary tube. □

Example E11.5 (Square wave Fourier tail). Let

$$j(t) = \mathbf{1}_{[-a,a]}(t)$$

on the circle and let s be smooth. If Q_m is a Fourier partial sum, then $Q_m(s + j)$ has the ordinary oscillatory boundary layer. In contrast,

$$Q_m s + j$$

has error exactly $s - Q_m s$. The visual comparison in figure E13 illustrates this change in approximation target. It is not a new Fourier summability theorem.

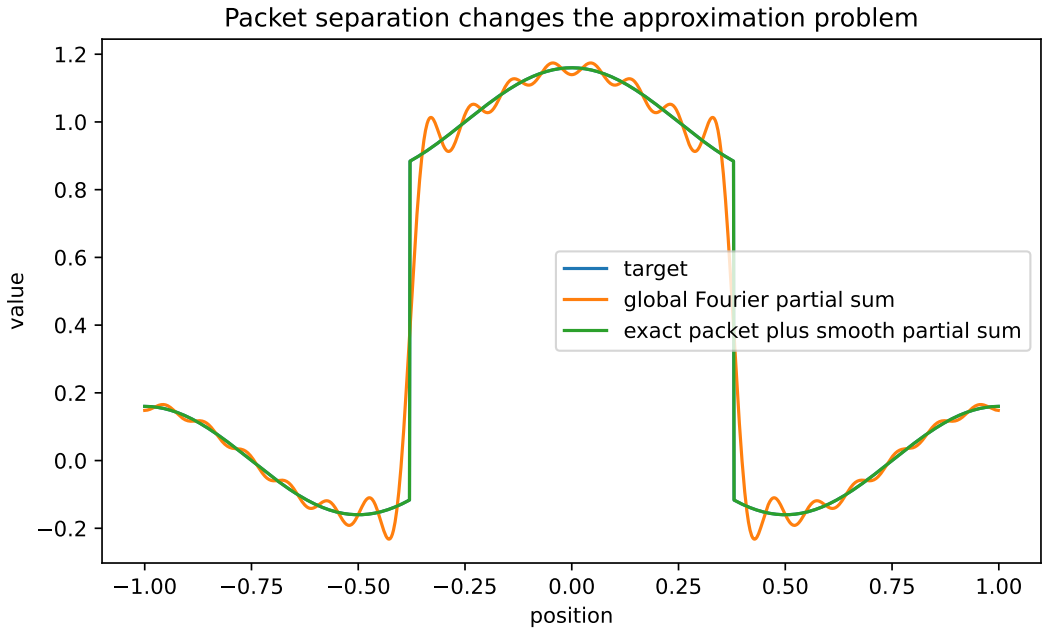


Figure E13. Global Fourier truncation represents the jump and exhibits an oscillatory boundary layer. Packet separation retains the jump and approximates only the smooth component.

E12 Nonlinear interactions in residue coordinates

Linear transfer matrices do not directly describe nonlinear products. Residue coordinates still obey exact conservation laws because integer addresses add modulo the packet modulus.

Theorem E12.1 (Residue law for Cauchy products)**.** *Let*

$$F(z) = \sum_{n \geq 0} f_n z^n, \qquad G(z) = \sum_{n \geq 0} g_n z^n$$

be formal power series. The same statement holds for power series that converge on a common disk. Decompose both series into residue packets modulo N:

$$F = \sum_{r=0}^{N-1} F_r, \qquad G = \sum_{s=0}^{N-1} G_s.$$

If $H = FG$, then its residue packet H_t is

$$\boxed{H_t = \sum_{r+s \equiv t \pmod{N}} F_r G_s.}$$

Proof. Every monomial in $F_r G_s$ has degree congruent to $r + s$ modulo N. Conversely, every coefficient of degree congruent to t is formed from pairs whose residues sum to t. Grouping the Cauchy product by this residue gives the formula. □

Theorem E12.2 (Shifted pointwise monomial law)**.** *Let*

$$y(n) = \prod_{j=1}^{m} x(n - a_j)$$

and write

$$a_j = Nc_j + b_j, \qquad 0 \le b_j < N.$$

For output residue r, define

$$r_j \equiv r - b_j \pmod{N}, \qquad 0 \le r_j < N,$$

and let

$$\kappa_{r,b_j} = \begin{cases} 1, & r < b_j, \\ 0, & r \ge b_j. \end{cases}$$

Then the output channel is

$$\boxed{y_r(q) = \prod_{j=1}^{m} x_{r_j}(q - c_j - \kappa_{r,b_j}).}$$

Proof. For $n = Nq + r$,

$$n - a_j = N(q - c_j) + r - b_j.$$

If $r \ge b_j$, this address has residue $r - b_j$ and coarse index $q - c_j$. If $r < b_j$, add N to the residue and subtract one from the coarse index. Multiplying the resulting channel values gives the formula. □

Corollary E12.3 (Polynomial shift filters)**.** *Every finite polynomial in shifted samples has an exact finite expression in residue coordinates. Each monomial selects a deterministic list of input residues and coarse shifts, and the complete output is their finite sum.*

Proof. Apply theorem E12.2 to each monomial and add the results. □

Example E12.4 (Quadratic nearest neighbor interaction)**.** For

$$y(n) = x(n)x(n-1)$$

and modulus N,

$$y_r(q) = \begin{cases} x_r(q)x_{r-1}(q), & r \ge 1, \\ x_0(q)x_{N-1}(q-1), & r = 0. \end{cases}$$

The carry at residue zero is the nonlinear analogue of a transfer crossing a coarse cell boundary.

E13 Finite numerical illustrations

The figures and tables in this section are finite checks and examples. They are not substitutes for the preceding proofs. Their purpose is to show the scale and shape of the terms appearing in the exact estimates.

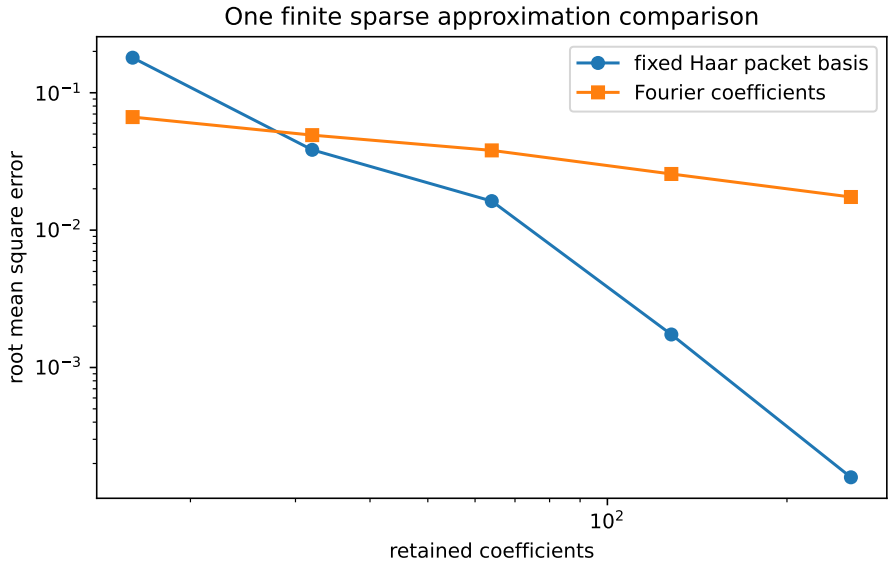


Figure E14. One finite sparse approximation comparison. Fourier coordinates are better at the smallest budget, while the fixed Haar packet basis becomes better after localized edge coordinates are resolved.

Table E1. Finite sparse approximation record for one test signal.

retained	Haar packet RMSE	Fourier RMSE
16	1.7999×10^{-1}	6.6522×10^{-2}
32	3.8414×10^{-2}	4.9153×10^{-2}
64	1.6267×10^{-2}	3.8056×10^{-2}
128	1.7387×10^{-3}	2.5631×10^{-2}
256	1.5870×10^{-4}	1.7424×10^{-2}

A fixed depth Haar packet transform and a Fourier basis were applied to the same piecewise smooth test signal. The largest coefficients were retained in each representation.

The table does not prove that Haar packets dominate Fourier coordinates for every signal. Smooth periodic oscillations may favor Fourier coordinates. Curved edges in two dimensions may favor directional systems. The geometric approximation results in [Don99, CD04] give much deeper statements for specific image classes.

The estimated packet rows used a narrow positive contrast synthetic model. They are not evidence for a general segmentation theorem. The exact packet rows verify theorem E7.5 numerically after the packet is supplied.

Table E2. Finite edge reconstruction record from the original computational study.

case	global	exact packet	estimated packet
one-dimensional step	5.4023×10^{-2}	1.3692×10^{-4}	2.4029×10^{-2}
two-dimensional disk	3.7643×10^{-2}	6.4752×10^{-5}	3.4043×10^{-3}
multiple regions	3.3740×10^{-2}	5.2118×10^{-7}	not evaluated

E14 Cyclic Cayley coordinates

Keep the rotation and its chart separate

For the phase factorization below, M and N are positive phase orders. The reflection coordinate rotates in the unit disk. The Cayley map expresses that same motion as a fractional linear impedance action. A pole of the chart does not mean that the projective action fails. This realization is independent of the separated jump detector.

This section records a secondary realization from the original signal manuscript. It is logically independent of the recovery theorems. The classical Smith chart uses the Cayley transform between normalized impedance and a reflection coordinate [Smi69].

Let

$$\mathbb{H} = \{z \in \mathbb{C} : \Re z > 0\}$$

and define

$$C(z) = \frac{z-1}{z+1}.$$

It maps $\mathbb{H}$ conformally onto the unit disk $\mathbb{D}$, with inverse

$$C^{-1}(w) = \frac{1+w}{1-w}.$$

For $\theta \in \mathbb{R}$, let

$$R_\theta(w) = \mathrm{e}^{\mathrm{i}\theta} w.$$

Theorem E14.1 (Cyclic Möbius action)**.** *The pullback*

$$\Phi_\theta = C^{-1} R_\theta C$$

is the fractional linear map

$$\boxed{\Phi_\theta(z) = \frac{(1+\mathrm{e}^{\mathrm{i}\theta})z + (1-\mathrm{e}^{\mathrm{i}\theta})}{(1-\mathrm{e}^{\mathrm{i}\theta})z + (1+\mathrm{e}^{\mathrm{i}\theta})}.}$$

It preserves $\mathbb{H}$ and satisfies

$$\Phi_\theta \Phi_\varphi = \Phi_{\theta+\varphi}.$$

For an integer $N \geq 1$ and $\theta = 2\pi/N$, the map has order N.

Proof. Substitute $\mathrm{e}^{\mathrm{i}\theta}(z-1)/(z+1)$ into C^{-1} and simplify. Preservation of $\mathbb{H}$ follows because C maps it to $\mathbb{D}$, rotation preserves $\mathbb{D}$, and C^{-1} maps back. The group law follows by conjugating the rotation group law. When $\theta = 2\pi/N$, the Nth power is the identity. Conversely, if its kth power is the identity, conjugation by C and evaluation at $w = 1/2$ give $\mathrm{e}^{2\pi \mathrm{i} k/N} = 1$, so N divides k. Hence the order is exactly N. □

Corollary E14.2 (Impedance and admittance)**.** *For $N = 2$, $\theta = \pi$ and*

$$\Phi_\pi(z) = \frac{1}{z}.$$

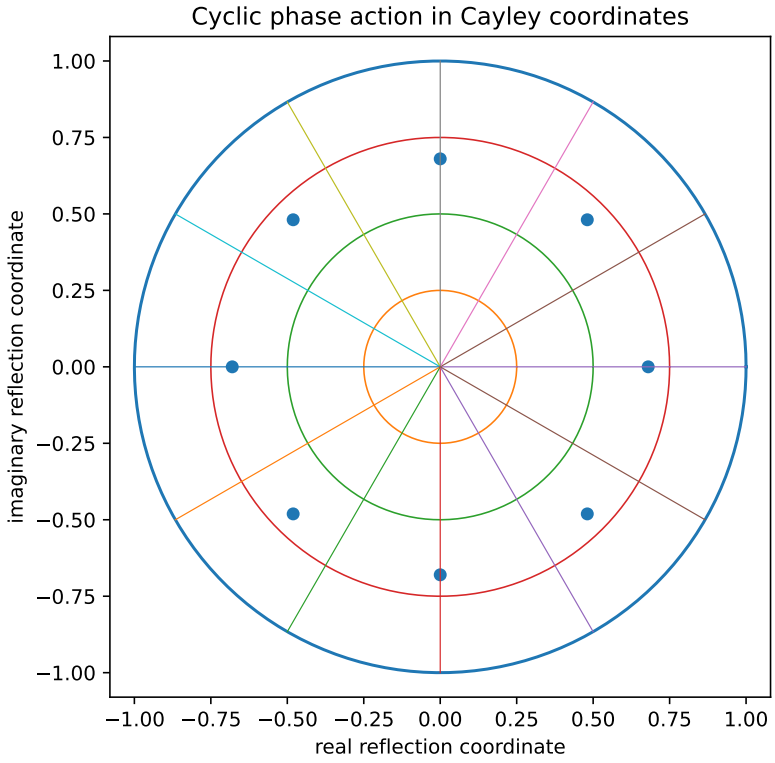


Figure E15. The cyclic action is an ordinary phase rotation in the reflection disk and a fractional linear action in the impedance half plane.

Thus the two phase action is normalized impedance and admittance inversion.

Proof. Insert $e^{i\pi} = -1$ in the formula of theorem E14.1. □

Proposition E14.3 (Radial and phase coordinates)**.** *Write*

$$C(z) = \rho e^{i\varphi}, \qquad 0 \le \rho < 1.$$

Then

$$C(\Phi_\theta(z)) = \rho e^{i(\varphi+\theta)}.$$

The radial mismatch ρ is invariant and the phase coordinate advances by θ.

Proof. This is the defining conjugacy $C\Phi_\theta = R_\theta C$. □

Proposition E14.4 (Coprime phase factorization)**.** *Let M and N be coprime. The cyclic phase labels modulo MN are equivalent to pairs of labels modulo M and modulo N:*

$$\mathbb{Z}/(MN)\mathbb{Z} \cong \mathbb{Z}/M\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}.$$

Under this identification, the rotation by $2\pi/(MN)$ is determined by its two projected cyclic actions.

Proof. This is the Chinese remainder theorem applied to the additive phase label. A label modulo MN is uniquely determined by its two residues, and addition of labels is preserved. □

E15 Limits and further directions

The one-dimensional recovery theorem identifies an explicit admissible threshold interval for an adjacent contrast detector. More general recovery requires a broader observation model. A multiscale average can reduce fixed sample noise, but its bias depends on the smoothness class and the separation of nearby jumps. A useful next theorem should select the averaging scale from the data while retaining a finite probability certificate.

Higher dimensional recovery requires geometry. The overlap and tube formulas in appendix E8 become quantitative once a boundary estimate is known. What remains is a stable method that recovers topology, orientation, curvature, and region values under a declared noise model. Wedgelets and curvelets show why directional geometry is essential for broad image classes [Don99, CD04].

The best basis recursion is exact for additive objectives. Cross packet leakage, shared computation, and global hardware constraints introduce couplings between subtrees. Those problems require a larger compiler or an approximation theorem. The cost stability results in theorem E6.6 and corollary E6.7 identify one condition under which the additive decision survives perturbation.

The higher order singular packet theorem removes finitely many trace jumps. A more complete approximation theory should connect packet norms to geometric measure quantities in BV, piecewise Sobolev spaces, and free discontinuity models. It should also determine when a redundant packet dictionary can identify the derivative order of a singularity rather than merely its location.

Finally, the two shifted Haar grids remove one exact alignment failure, but they are redundant. A complete redundant theory should provide stable synthesis, noise aware coefficient aggregation, and a best basis rule that does not count the same singularity several times.

E16 Conclusion

Finite index sublattices give exact coordinates for lattice signals. In those coordinates, convolution is a matrix of transfer kernels. Perfect reconstruction is a matrix inverse identity. Approximate reconstruction is controlled by a multiplier norm. Frame bounds are pointwise singular value bounds. Lifting steps are exact triangular coordinate changes. Every admissible packet tree gives a complete coordinate system, and every additive tree objective has an exact dynamic recursion.

The singular packet theory separates representation from recovery. A finite family of truncated powers records jumps in a function and its derivatives, leaving a globally Sobolev remainder under explicit trace assumptions. If the packet is known and retained, the error is exactly the smooth approximation error. If the packet is estimated, exact region overlap and location formulas quantify the remaining defect, while the reconstruction operator contributes a transparent stability factor.

For separated one-dimensional jumps, a deterministic contrast gap gives exact cell recovery, midpoint location accuracy, and amplitude control. A sub-Gaussian version follows from one union bound, and repeated measurements reduce the effective noise scale. Haar coefficients provide a second separation law because smooth coefficients decay one full power of scale faster than jump coefficients. Two shifted grids remove boundary alignment at a fixed scale. Higher differences annihilate piecewise polynomial interiors and confine their response to breakpoint bands.

The resulting chain is

$$\begin{aligned}&\text{lattice coordinates} \longrightarrow \text{transfer blocks} \longrightarrow \text{stable filter banks} \longrightarrow \text{packet trees}\\ &\longrightarrow \text{singular packet model} \longrightarrow \text{local recovery} \longrightarrow \text{smooth approximation rate.}\end{aligned}$$

Every arrow is exact or quantitatively controlled under the hypotheses stated in the corresponding result.

Part III: Forty worked visual plates

A second route through the mathematics

These forty plates develop the finite models used in the separate visual guide. Plates V.1–V.40 stay fixed across both documents. They lead from exact signal coordinates through filter transfers, adaptive decisions, packet errors, stable banks, and separated jump recovery. Every page gives the relevant manuscript locator. Finite arithmetic and diagrams explain the stated hypotheses and do not extend the formal contract.

How to use this volume

Forty worked pages and thirty-seven editable vector figures follow exact signal coordinates, a fully enumerated packet tree, and a reflection orbit, then extend the route to stable banks, singular trace packets, and conditional recovery of separated jumps.

Part I / A route through the examples

Plates V.1 through V.40 identify worked pages independently of PDF pagination. Each page names its corresponding statements in the mathematical text. The Lean guide records the separate formal contract.

Plate V.1 Three objects, three different questions

Plate V.2 An atlas of the objects

Plate V.3 Decimation is a permutation, not a loss

Plate V.4 Refine packets on a common lattice

Plate V.5 A filter becomes a matrix of actual delays

Plate V.6 One sample is not one channel step

Plate V.7 Perfect reconstruction is an operator identity

Plate V.8 A jump can hide from the first Haar split

Plate V.9 Lifting preserves information, not energy

Plate V.10 The second level resolves the hidden change

Plate V.11 Deleting coefficients is the approximation step

Plate V.12 A mixed-depth basis must cover every branch

Plate V.13 Solve every alternative without searching every alternative

Plate V.14 How far can the costs move before the tree changes?

Plate V.15 Retain the jump and approximate the other packet

Plate V.16 Why a Fourier cutoff cannot reproduce this jump pointwise

Plate V.17 Small location errors, different norms

Plate V.18 Separate amplitude error from mask error

Plate V.19 Geometry localizes error while data may leave it ambiguous

Plate V.20 Two products, two packet laws

Plate V.21 Nonlinear neighbors cross channel boundaries

Plate V.22 One reflection circle, four very different impedances

Plate V.23 A pole is a chart boundary, not a broken group action

Plate V.24 Residue labels are not impedance averages

Plate V.25 Sensitivity grows before the pole is reached

Plate V.26 What the reader can now verify

Three reading routes

Signal coordinates: Plates V.3–V.11, then Plate V.20 and Plate V.21. **Selection and error:** Plates V.12–V.19. **Nonlinear geometry:** Plates V.22–V.25.

One model per question

The sixteen-sample signal gives exact rational values. The depth-three tree has twenty-six complete prunings. The reflection orbit has four exact quarter-turn points. These bounded examples expose the chapter's mechanisms without replacing its general statements.

Read the hypotheses with the picture

Zero extension, periodic indexing, and infinite sequences describe different objects. Known jumps differ from estimated jumps. A finite optimum depends on a specified cost, and a coordinate transformation does not establish an optimal sampling rule. Plate V.26 records the evidence behind this guide.

Part II / From coordinates to recovery

The expanded traditional development supplies additional theorems beyond the inherited twenty-result formal contract. These fourteen new worked pages follow its hypotheses through exact examples and quantitative bounds.

Plate V.27 What the extended developments add

Plate V.28 Six image phases are six exact address classes

Plate V.29 A positive singular-value gap survives perturbation

Plate V.30 Nonunitary tree bounds propagate along branches

Plate V.31 Derivative jumps have their own packets

Plate V.32 The approximation operator also acts on packet error

Plate V.33 An overlap matrix is an exact error ledger

Plate V.34 A shifted step and a shifted ramp have different moduli

Plate V.35 A strict contrast gap locates every separated jump cell

Plate V.36 Repeated observations shrink the noise side of the gap

Plate V.37 Haar sees a jump only through its position in the support

Plate V.38 Two half-shifted grids remove exact alignment failure

Plate V.39 Higher differences annihilate polynomial interiors

Plate V.40 Conditional Gibbs control closes the reconstruction estimate

A connected route

The lattice example extends exact addressing. Frame bounds then quantify the stability of analysis and synthesis. Trace packets remove specified singularities, while overlap and location formulas measure errors in their estimates. Three local detectors address separated jumps under contrast hypotheses. The final page combines the recovered packet with a chosen approximation operator.

Keep the evidence attached to its claim

The twenty-result core carries its preserved Lean map. Part II carries traditional proofs and the recorded mathematical review, with a separate exact checker for the worked examples. The symbolic visual suite, arithmetic runs, saved compiler transcripts, and rendered figures each have a stated scope. Neither a figure nor a finite check expands the formal theorem map.

What the example numbers mean

The additional checker uses exact rational arithmetic. Its logarithmic sampling budget is enclosed using a positive Taylor partial sum. Curve coordinates are converted to decimal positions for drawing. Displayed formulas retain exact values. All plates drawn from the source are editable TikZ, and the final comparison figure is preserved from the expanded manuscript's vector source.

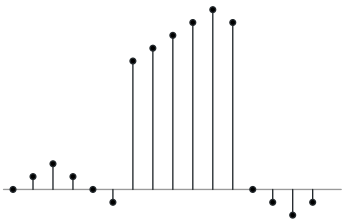
Plate V.1 / PAPER V / A WORKED VISUAL COMPANION

Three objects, three different questions

Chapter V: Section 1, Section 6, Section 11, Section 13.

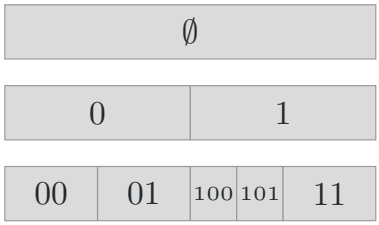
A packet calculus can reorganize a signal exactly, choose among finitely many representations, or describe a nonlinear coordinate geometry. These are not the same task. This guide keeps one concrete object in view for each, so that the reader can compute rather than only follow arrows.

SIGNAL COORDINATES



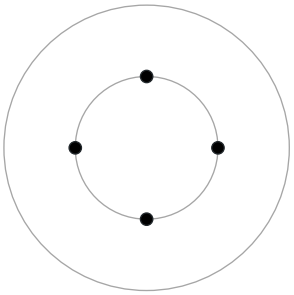
$f = s + j \quad \text{in } \mathbb{Q}^{16}$

FINITE TREE CHOICES



26 complete prunings

REFLECTION ORBIT



$\Gamma = \frac{1}{2} i^k$

A signal that can be checked by hand

Our main finite signal is $f = s + j \in \mathbb{Q}^{16}$. The background repeats

$$\tfrac{1}{4}(0, 1, 2, 1, 0, -1, -2, -1),$$

and $j[n] = 3$ at $6 \le n < 12$, with $j[n] = 0$ elsewhere. This is a discrete background model, not an assertion of continuum smoothness. Its energy is exactly 57. We use it for residue coordinates, Haar transforms, coefficient deletion, and known-packet cancellation.

A tree whose alternatives can all be seen

The full binary tree of depth three has only 26 complete prunings. We enumerate all of them, compute a unique mixed-depth optimum, and certify its stability with rational costs. The displayed objective is deliberately explicit, not claimed to be a universal score for signal quality.

A coordinate orbit with no hidden numerical poles

The four reflection points $\Gamma = i^k/2$ map to four exact impedances. A second orbit passes through infinity. Homogeneous coordinates make that point an ordinary part of the construction.

> **Read this as:** Exact coordinates preserve information. An optimizer chooses coordinates. An estimator must still identify information not already supplied.

Plate V.2 / READING CONVENTIONS / KEEP THE TYPES VISIBLE

An atlas of the objects

Chapter V: Section 20 (notation crosswalk), Section 21 (formal correspondence).

Several symbols in the chapter describe different realizations of the same operator language. The following distinctions are used throughout this guide.

Object	Literal meaning	What it is not
S, δ_0	Sample shift and unit impulse	A finite truncation of every ℓ^2 signal
$P_{N,r} f$	A masked signal on the original lattice	The shorter decimated channel
$f_r[q]$	The value $f[Nq + r]$	An average of neighboring samples
D	One channel-index delay	Necessarily one original-sample delay
$E(D), R(D)$	Finite Laurent analysis and synthesis matrices	Values at one test frequency
W_α	The addressed Haar analysis operator	An edge detector by itself
$W_\alpha^* W_\alpha$	A projector on the original signal space	A scalar packet energy
Λ	A complete covering antichain of a finite tree	An arbitrary collection of nodes
$j, \hat{j}$	Known and estimated discontinuity packets	Interchangeable hypotheses
Γ, z	Reflection coefficient and normalized impedance	Coefficients that can be linearly interchanged

Three arithmetic conventions

FIR examples use bilateral sequences with finite support and *zero extension*. Haar examples use vectors of length 16, with lengths divisible by the required powers of two. The three-tap smoother and nonlinear neighbor example use *periodic* indexing. These boundary conventions are stated where they enter. None is silently substituted for another. The polynomial variable t on Plate V.20 indexes coefficients. The delay symbol D acts on channels. The Smith variable z is impedance. Keeping them separate prevents a visual resemblance from becoming an invalid identity.

> **Read this as:** A correct formula needs the correct object, product, norm, and boundary convention. A familiar matrix does not supply them automatically.

Plate V.3 / EXACT ADDRESSES / THE RUNNING SIGNAL

Decimation is a permutation, not a loss

Chapter V: Theorem 2.1, Corollary 2.2.

For four phases, write every sample address uniquely as $n = 4q + r$ with $0 \le r < 4$. Analysis stores $f[n]$ at channel address (r, q). Synthesis puts that exact value back at $4q + r$.

FOUR PHASES: EVERY ADDRESS APPEARS ONCE

	$q=0$	$q=1$	$q=2$	$q=3$	
$r=0$	$n=0:\ 0$	$n=4:\ 0$	$n=8:\ 3$	$n=12:\ 0$	$n = 4q + r$
$r=1$	$n=1:\ \frac{1}{4}$	$n=5:\ -\frac{1}{4}$	$n=9:\ \frac{13}{4}$	$n=13:\ -\frac{1}{4}$	$-1 = 4(-1) + 3$
$r=2$	$n=2:\ \frac{1}{2}$	$n=6:\ \frac{5}{2}$	$n=10:\ \frac{7}{2}$	$n=14:\ -\frac{1}{2}$	
$r=3$	$n=3:\ \frac{1}{4}$	$n=7:\ \frac{11}{4}$	$n=11:\ \frac{13}{4}$	$n=15:\ -\frac{1}{4}$	$-5 = 4(-2) + 3$

Every cell contains an original address and its rational sample value. Shading marks the known jump packet, not an estimated feature or a frequency band.

What has changed? Only where the values are stored

The complete tuple of channels contains sixteen values, just as the original vector does. A single decimated channel is shorter, but keeping all four loses no samples. The residue projector $P_{4,r}f$ is different: it keeps the original lattice and writes zeros outside phase r.

$$\mathcal{S}_4\mathcal{A}_4 f = f, \qquad \|f\|_2^2 = \sum_{r=0}^{3} \|f_r\|_2^2 = 57.$$

The norm on the channel tuple is the sum-of-squares Hilbert norm, not the maximum of the channel norms. This distinction matters in the infinite-dimensional statement.

Negative addresses are ordinary addresses

Euclidean division, rather than truncation toward zero, gives $-1 = 4(-1) + 3$ and $-5 = 4(-2) + 3$. The finite tests exercise both positive and negative sample addresses. The algebraic identity applies before any plotting window or finite buffer is chosen.

> **Read this as:** Downsampling one channel discards information. Splitting into all residue channels is an invertible address change.

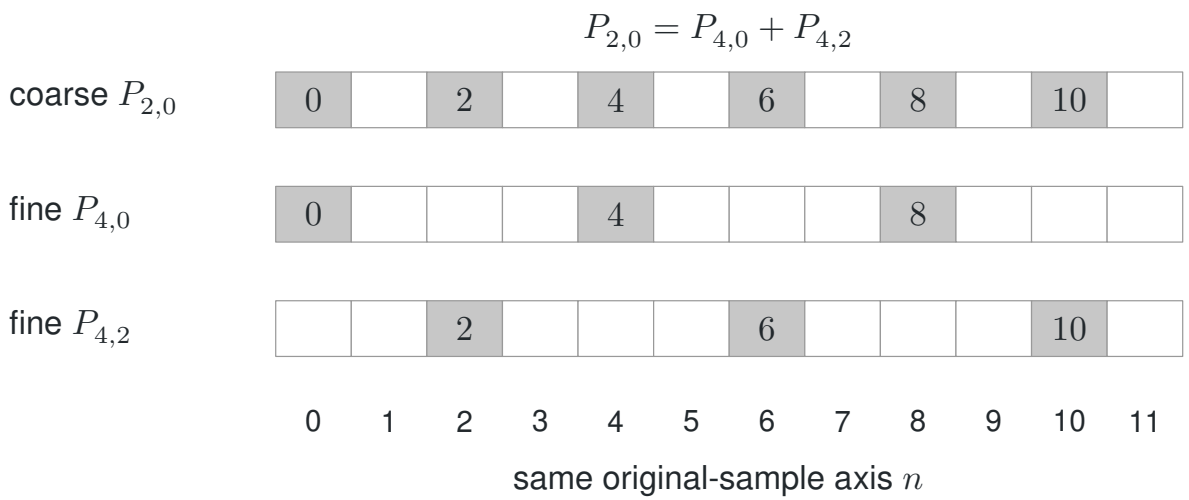
Plate V.4 / EXACT REFINEMENT / WHAT CAN BE ADDED

Refine packets on a common lattice

Chapter V: Theorem 2.3.

An even sample belongs to exactly one of residues 0 and 2 modulo four. Therefore the coarse even mask is the sum of two disjoint fine masks:

$$P_{2,0} = P_{4,0} + P_{4,2}.$$



The common horizontal axis is the original sample index. Empty cells denote masked-out addresses. There are no interpolated samples between the displayed cells.

Why the coordinate convention matters

The decimated channel $f_0[q] = f[2q]$ has its own index q. At modulus four the descendants are $f[4q]$ and $f[4q+2]$. These shorter arrays must be interleaved to recover the coarse channel. Adding them entrywise instead produces sums of different original samples. For example, at the first descendant address the values $f[0] = 0$ and $f[2] = 1/2$ are two separate values. Their sum $1/2$ cannot tell us both of them. Exact refinement is a disjoint partition, not an averaging rule or a new measurement.

The general formula

When N divides M,

$$P_{N,r} = \sum_{\substack{0 \le a < M \\ a \equiv r \pmod{N}}} P_{M,a}.$$

The divisibility hypothesis ensures that every fine residue class lies entirely inside one coarse class. Without it, this particular descendant relation is not the right refinement rule.

> **Read this as:** Finer coordinates can preserve the object exactly. Compressing their values to averages is an additional operation, with an additional information-loss question.

Plate V.5 / FIR TRANSFER / A FOUR-TAP EXAMPLE

A filter becomes a matrix of actual delays

Chapter V: Theorem 3.1.

Take the causal filter $h = (1, 2, -1, 1)$, with taps at $0, 1, 2, 3$, acting by $y[n] = \sum_q h[q] f[n-q]$. Resolve the input into even and odd channels. Let D denote one delay in the *channel* index.

$$h = (1, 2, -1, 1), \qquad D : \; u[n] \mapsto u[n-1]$$

	input 0	input 1
output 0	$1 - D$	$D^2 + 2D$
output 1	$D + 2$	$1 - D$

$$y_0 = (1 - D)x_0 + (2D + D^2)x_1$$

Reading one matrix entry

The odd-input contribution to the even output is $(2D + D^2)f_1$. The tap at original delay one reaches the preceding odd channel sample. The tap at delay three reaches the odd channel two positions back. These are different delays despite coming from the same input phase. The two channel equations are

$$y_0[n] = f_0[n] - f_0[n-1] + 2f_1[n-1] + f_1[n-2],$$
$$y_1[n] = 2f_0[n] + f_0[n-1] + f_1[n] - f_1[n-1].$$

Interleaving gives exactly the direct convolution, including its boundary samples under zero extension.

The general address calculation

For N phases, the block from input s to output k has kernel $h_{ks}[m] = h[Nm + k - s]$. This comes from the exact identity $Nn + k - q = N(n - m) + s$. It is a finite reindexing of taps, so no numerical transform of the input is needed. Evaluation of a bilateral transform is a separate step with the summability hypotheses stated in the manuscript.

> **Read this as:** The matrix does not approximate the filter. Its entries record which input phase is used, how far back to read, and what coefficient to multiply.

Plate V.6 / DELAY UNITS / AN IMPULSE MAKES THE DIFFERENCE VISIBLE

One sample is not one channel step

Chapter V: Theorem 4.1.

A one-sample delay swaps the two phases and wraps one of them into the preceding channel index. Its matrix is $J(D)$, not DI_2.

one original sample two original samples

$$J(D)^2 = \begin{bmatrix} 0 & D \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$$

$\delta_0 \mapsto \delta_1$

$\delta_0 \mapsto \delta_2$

$$y_0[n] = x_1[n-1], \quad y_1[n] = x_0[n]$$

channel delay 1 at $N = 2$ is original-sample delay 2

Trace one impulse

The input δ_0 has a single nonzero value in even channel address zero. Under $J(D)$ it appears in odd channel address zero: this is the original sample δ_1. Under DI_2 it stays even but moves to channel address one: this is δ_2. Thus

$$J(D)^2 = DI_2, \qquad H(D) = I_2 + 2J(D) - J(D)^2 + J(D)^3$$

recovers the previous page's four-tap transfer matrix. The same delay algebra explains both the off-diagonal phase swap and the channel powers.

Signed delays need the same care

For an original-sample delay d at modulus N,

$$y_k[n] = f_{(k-d) \bmod N}\left[n + \left\lfloor \frac{k-d}{N} \right\rfloor\right].$$

This includes advances when $d < 0$. The finite verification checks the formula on signed sample addresses. A bank with product $D^q I_N$ delays the original signal by Nq samples, not q samples.

> **Read this as:** Always state the unit of a delay. A phase permutation and a channel delay together can represent one original sample.

Plate V.7 / FILTER BANKS / WHY A FREQUENCY SPOT-CHECK IS INSUFFICIENT

Perfect reconstruction is an operator identity

Chapter V: Theorem 4.1.

Consider the finite Laurent bank

$$E(D) = \begin{pmatrix} 1 & D \\ 0 & 1 \end{pmatrix}, \qquad R(D) = \begin{pmatrix} 1 & -D \\ 0 & 1 \end{pmatrix}.$$

Its synthesis cancels the exact delayed cross-channel contribution introduced by analysis.

$E(D)$

1	D
0	1

$R(D)$

1	$-D$
0	1

$$R(D)E(D) = I$$

every impulse is recovered

$$B(D) = \operatorname{diag}(D, 1): \quad B(1) = I, \quad B(D) \neq I$$

testing one frequency cannot detect a channel delay

Two inverses, all inputs

Direct polynomial multiplication gives $RE = ER = I_2$. Since D acts as a shift, these identities imply perfect reconstruction on every bilateral channel sequence. With finite kernels, composition requires only finitely many operations at each address. The converse uses impulses. Insert an impulse in one input channel. Each output channel lists every coefficient of the corresponding transfer polynomial. If all such responses vanish for $RE - I$, every coefficient is zero. Hence the polynomial matrix itself is zero.

A deliberately misleading test

The matrix $B(D) = \operatorname{diag}(D, 1)$ satisfies $B(1) = I_2$. At that one transform value it appears perfect. In the time domain, however, it delays only the first channel. An impulse in that channel immediately distinguishes it from the identity. This is why a small set of successful frequency evaluations is not automatically a proof of the full Laurent identity. A separate degree bound and a sufficient interpolation argument would be needed to turn finitely many evaluations into such a certificate.

> **Read this as:** Exact reconstruction concerns every coefficient of the transfer operator, not just an attractive response curve or one evaluation point.

Plate V.8 / HAAR ENERGY / EDGE ALIGNMENT

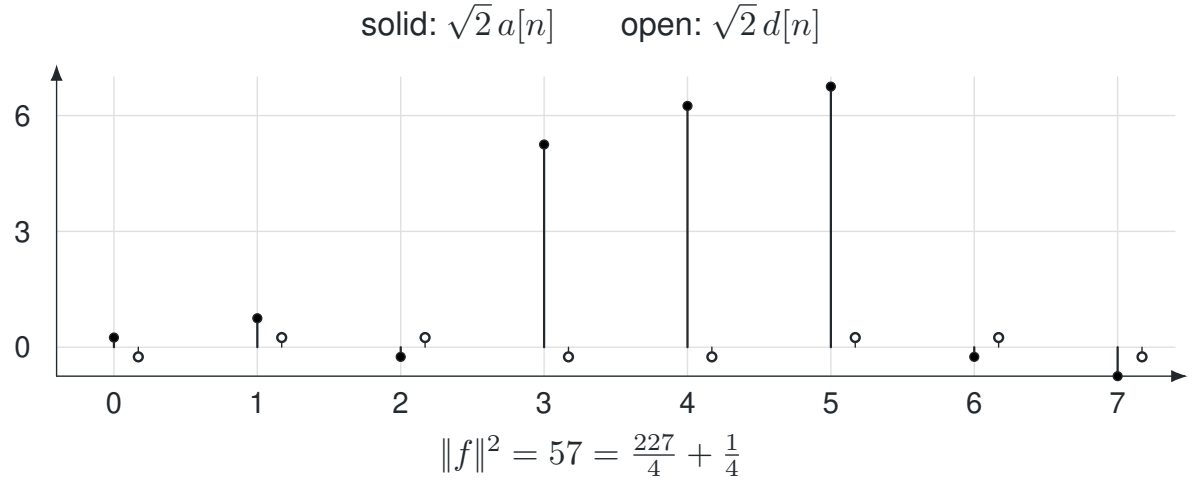
A jump can hide from the first Haar split

Chapter V: Corollary 4.2.

For the running signal, Haar analysis gives

$$a[n] = \frac{f[2n] + f[2n+1]}{\sqrt{2}}, \qquad d[n] = \frac{f[2n] - f[2n+1]}{\sqrt{2}}.$$

The plot scales both channels by $\sqrt{2}$ so that all displayed ordinates are rational.



The exact energy calculation

The low channel has energy $227/4$, and the detail channel has energy $1/4$. Their sum is 57, exactly the input energy. The normalization is essential: unnormalized sums and differences would multiply the total energy by two.

Why is the first detail channel so small?

The jump starts at sample 6 and ends just before sample 12. Both lie *between* the pairs $(0,1), (2,3), \dots, (14,15)$. Inside every pair the jump contributes either $(0,0)$ or $(3,3)$, so its first-level difference is zero. The small detail values come entirely from the discrete background. This does not mean the edge is absent from the transform. The next split of the low channel compares neighboring pairs, and those comparisons do cross the jump. The next pages expose that contribution explicitly.

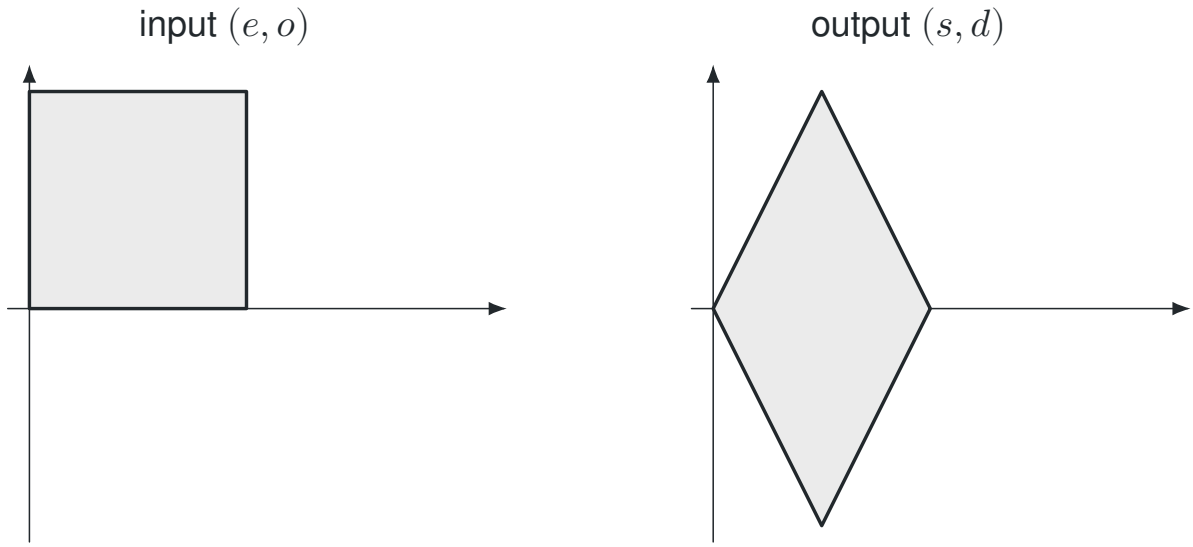
> **Read this as:** Haar energy is exact, but a particular detail level is not a universal edge detector. Edge visibility depends on alignment, scale, and the chosen packets.

Plate V.9 / PREDICT–UPDATE / INVERTIBILITY VERSUS ISOMETRY

Lifting preserves information, not energy

Chapter V: Theorem 4.3.

For an even/odd pair (e, o), form the prediction residual $d = o - e$, then update $s = e + d/2$. The inverse reads $e = s - d/2$, followed by $o = d + e$.



The square $[0, 4]^2$ and its exact image under the lifting matrix have the same area. The separate pair $(2, 6)$ illustrates that Euclidean energy need not be preserved.

$$\begin{pmatrix} s \\ d \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} e \\ o \end{pmatrix}, \qquad \begin{pmatrix} e \\ o \end{pmatrix} = \begin{pmatrix} 1 & -1/2 \\ 1 & 1/2 \end{pmatrix} \begin{pmatrix} s \\ d \end{pmatrix}.$$

What determinant one does establish

The matrix is invertible, and in these real coordinates it preserves oriented area. The predict and update operations can be undone in reverse order. Applied to each pair, the construction gives exact finite-signal reconstruction.

What determinant one does not establish

The pair $(2, 6)$ becomes $(4, 4)$. Its squared Euclidean norm changes from 40 to 32. The lifting matrix is not unitary. Energy accounting must therefore use its actual coordinate metric or a separately normalized transform. Invertibility alone cannot replace the Haar energy identity. All values in this example are rational. Integer-to-integer lifting with a rounding rule is a different algorithm. Its inverse and error conventions require their own specification and are not silently inferred from the displayed rational matrix.

> **Read this as:** Perfect reconstruction, determinant one, and norm preservation are different properties. This transform has the first two, not the third.

Plate V.10 / ACTUAL WAVELET PACKETS / FOUR RATIONAL CHANNELS

The second level resolves the hidden change

Chapter V: Theorem 5.1.

Apply Haar analysis to *both* children. A binary address records the chronological low/detail choices, with 0 for low and 1 for detail. Each depth-two channel contains four coefficients.

	coefficient positions within each packet				energy
00	$\frac{1}{2}$	$\frac{5}{2}$	$\frac{13}{2}$	$-\frac{1}{2}$	49
01	$-\frac{1}{4}$	$-\frac{11}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{31}{4}$
10	0	0	0	0	0
11	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

$$\sum_{\alpha\in\{0,1\}^2} W_\alpha^* W_\alpha = I_{16}$$

The edge that first-level detail missed

Packet 01 is the detail child of the first low channel. It contains $-11/4$, the large change between neighboring pair averages. Packet 10 vanishes for this particular signal because the first-level differences alternate in canceling pairs. Neither feature is a universal identity for all inputs. The packet energies are

$$49, \qquad 31/4, \qquad 0, \qquad 1/4,$$

which again sum to 57. Two factors of $1/\sqrt{2}$ explain why the depth-two coefficients are rational halves and quarters.

Coefficients and projectors are different objects

$W_\alpha f$ is the short coefficient vector. The original-space component is $W_\alpha^* W_\alpha f$. Summing these synthesized components, not the short arrays entrywise, reconstructs f:

$$f = \sum_{|\alpha|=2} W_\alpha^* W_\alpha f.$$

The finite verification constructs the actual analysis matrices and checks their product with their adjoints. The chapter's ℓ^2 statement additionally establishes boundedness and the existence of the literal Hilbert adjoints.

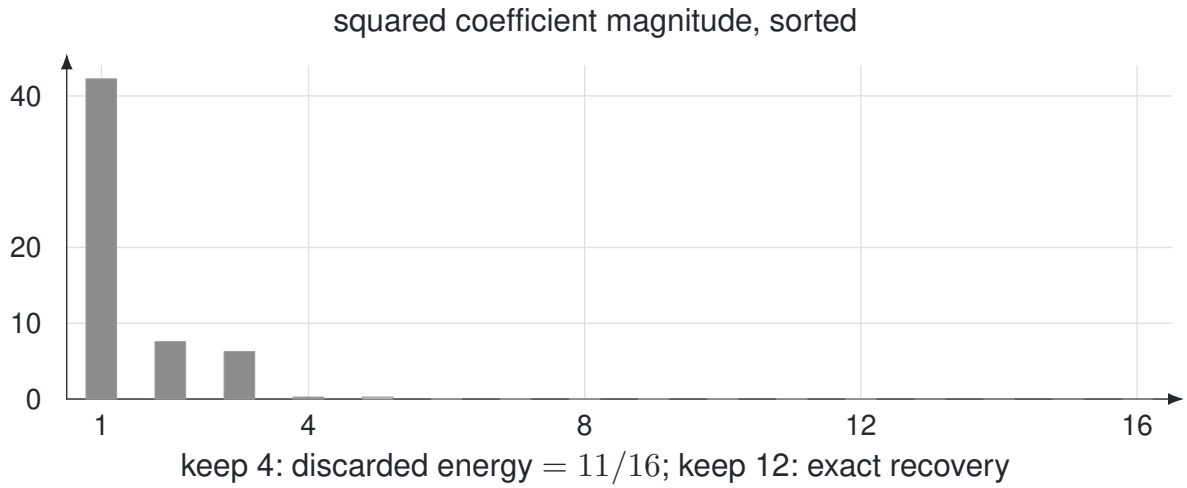
> **Read this as:** A packet address specifies an operator word. To add different packets as signals, synthesize them back into the common signal space first.

Plate V.11 / SPARSITY / AN EXACT ERROR LEDGER

Deleting coefficients is the approximation step

Chapter V: Section 10.

The complete depth-two transform is exact. Approximation begins only when we discard some of its sixteen coefficients. Order them by squared magnitude, with a deterministic address-and-position rule for ties.



Every discarded square has a meaning

In this fixed orthonormal basis, the squared reconstruction error equals the sum of the squared discarded coefficients. For our running signal:

kept coefficients k	1	2	4	8	12	16
squared error	$59/4$	$115/16$	$11/16$	$1/4$	0	0

The four zero coefficients belong to packet 10, so twelve coefficients suffice for exact reconstruction here. This is a property of the selected signal and basis, not a guarantee that all edge signals need only twelve coefficients.

Do not compare incompatible budgets

A cutoff m in a one-dimensional signed Fourier spectrum means $2m + 1$ bins, including zero. A budget of k largest coefficients is a different rule. Real Fourier synthesis may also require conjugate-pair closure. Neither count alone equals a bit-storage budget once amplitudes, phases, indices, or a selected tree must be encoded. The manuscript's Fourier-versus-Haar experiment is a finite comparison, not a proof that one basis dominates every signal class. Here we isolate the part that can be checked exactly: reconstruction error within one fixed finite basis.

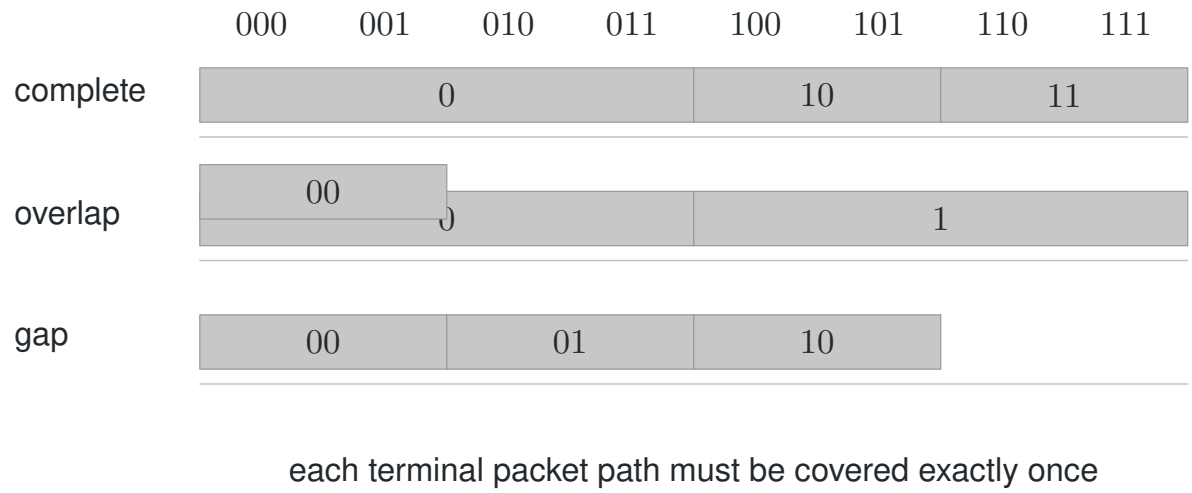
> **Read this as:** Changing coordinates creates no error. Throwing away coefficients does, and orthogonality makes that error exactly accountable.

Plate V.12 / COMPLETE PRUNINGS / NO GAPS AND NO DOUBLE COUNTING

A mixed-depth basis must cover every branch

Chapter V: Definition 11.1, Theorem 11.2.

A complete pruning chooses nodes of a finite refinement tree so that every root-to-terminal path meets exactly one selected node. The horizontal positions below are *terminal packet addresses*, not time intervals in the original signal.



Three candidate selections

The complete selection $\{0, 10, 11\}$ covers every terminal branch once. The selection $\{0, 00, 1\}$ covers the 00 branch twice: it is not an antichain. The selection $\{00, 01, 10\}$ misses the 11 branch entirely. For actual Haar packets, set $P_v = W_v^* W_v$. Parent-to-child conservation gives

$$P_v = P_{v0} + P_{v1}.$$

Replacing unselected parents by their children until the complete pruning is reached proves $\sum_{v \in \Lambda} P_v = I$. This is why the coefficients at different depths still form an exact coordinate system.

An exhaustive finite check

At depth three the pruning count obeys $a_0 = 1$ and $a_{d+1} = 1 + a_d^2$, giving $1, 2, 5, 26$. The supplied suite checks all 26 covering conditions and all 26 actual 16×16 projector-sum identities. These finite checks illustrate, but do not replace, the universal tree theorem.

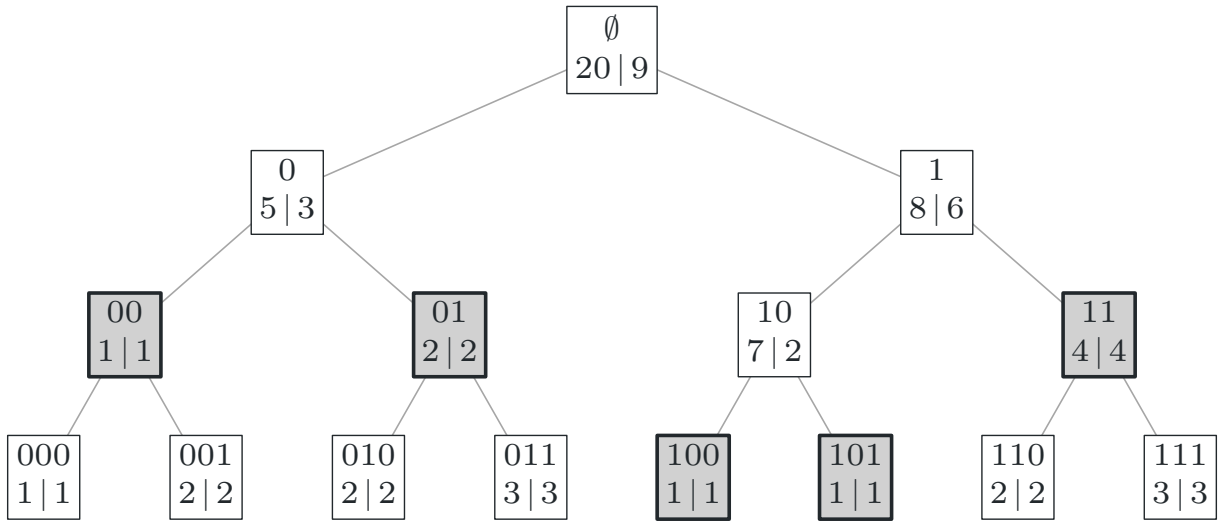
> **Read this as:** Mixed depth is allowed. Missing a terminal path or selecting an ancestor and its descendant is not.

Plate V.13 / ADDITIVE COSTS / EXACT DYNAMIC PROGRAMMING

Solve every alternative without searching every alternative

Chapter V: Theorem 11.3, Theorem 11.4.

Give every node the rational stop cost shown below. A node may either stop and pay its own cost, or refine and pay the already optimized costs of its children. The shaded nodes form the selected complete pruning.



node labels: stop cost c_v | optimal subtree cost $V(v)$

$$V(v) = \min\{c_v,\ V(v0) + V(v1)\}, \qquad V(v) = c_v \text{ at a terminal node.}$$

Read the calculation from the bottom upward

At 10, the children cost $1 + 1 = 2$, better than stopping for 7. At 11, stopping costs 4, better than $2 + 3 = 5$. The right subtree therefore costs 6, better than its stop cost 8. The left subtree costs $1 + 2 = 3$, better than 5. At the root, $3 + 6 = 9$ beats 20. The unique optimum is

$$\Lambda_* = \{00, 01, 100, 101, 11\}, \qquad J(\Lambda_*) = 9.$$

All 26 complete prunings are separately enumerated and give the same minimum. At equality, the rule stops, making the selector deterministic. A singleton tree needs no comparison. The costs here are supplied numbers. The theorem optimizes that stated additive objective. It does not prove those numbers encode the best denoising criterion. Coupled costs that cannot be written as a sum of selected-node costs need a different optimizer.

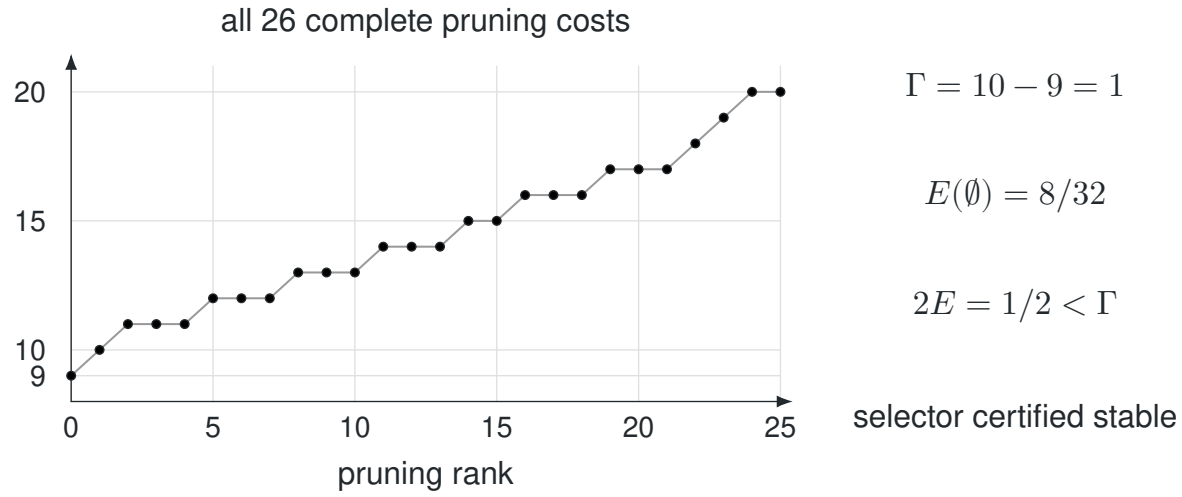
> **Read this as:** Global optimality follows from the exact stop-or-refine decomposition of a complete pruning, not from a greedy assumption.

Plate V.14 / A GLOBAL MARGIN / NOT A LOCAL HEURISTIC

How far can the costs move before the tree changes?

Chapter V: Theorem 11.3, Corollary 11.5.

The best pruning on the previous page costs 9. The cheapest competitor costs 10. Its exact global gap is therefore $\Gamma = 1$. The plate lists every complete-pruning cost in sorted order, giving each pruning its own horizontal rank.



Propagate an error budget as carefully as a cost

Suppose every node cost changes by at most $\varepsilon_v = 1/32$. A pruning may select as many as eight leaves, so its total cost can change by at most $8/32 = 1/4$. The recursive envelope proves this without guessing which pruning will be selected:

$$E(v) = \max\{\varepsilon_v,\ E(v0) + E(v1)\}, \qquad E(\emptyset) = \tfrac{1}{4}.$$

For any competitor Λ, the changed cost difference is bounded below by

$$J'(\Lambda) - J'(\Lambda_*) \geq \Gamma - 2E(\emptyset) = \tfrac{1}{2} > 0.$$

Thus the optimum is unchanged for *every* permitted simultaneous perturbation, not just a sampled noise realization. The envelope is over all feasible prunings, not just the five nodes in the current optimum.

Strict inequality has content

If two competing costs are 9 and 10, moving them to 9.5 and 9.5 removes uniqueness. Equality in $2E = \Gamma$ is therefore not a unique-selector certificate. For ordinary probability, the random node costs and the joint good event must be measurable. Separate nodewise probability bounds do not automatically give the probability of that joint event.

> **Read this as:** The exact selector is stable because its global gap exceeds the total worst-case movement of both competing costs.

Plate V.15 / AN EXACT IDENTITY / A CONDITIONAL INPUT

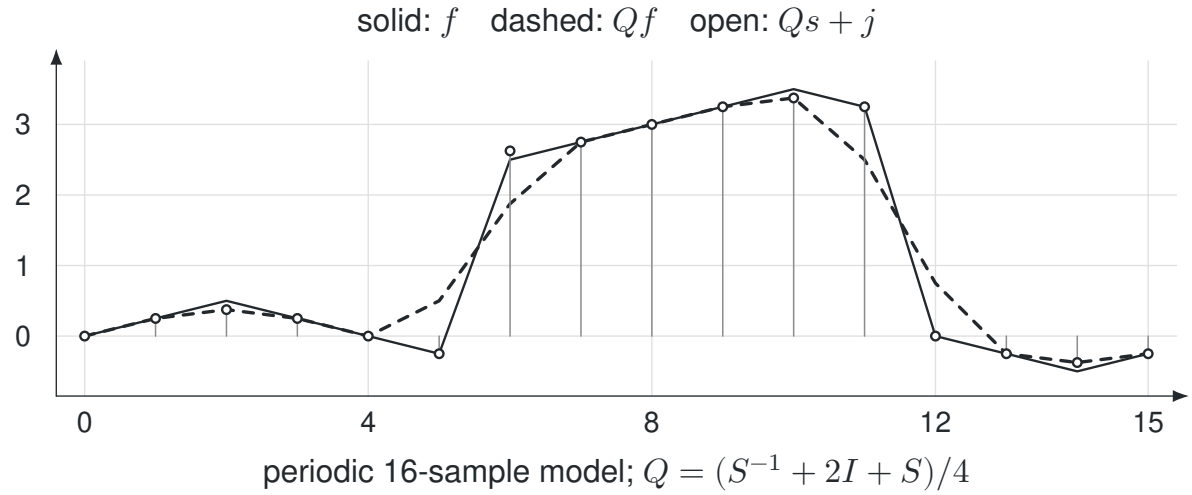
Retain the jump and approximate the other packet

Chapter V: Theorem 6.1, Corollary 6.2.

Return to the running signal $f = s + j$. For a concrete finite calculation, use the periodic three-tap smoother

$$(Qx)[n] = \tfrac{1}{4}x[n-1] + \tfrac{1}{2}x[n] + \tfrac{1}{4}x[n+1], \qquad n \in \mathbb{Z}/16\mathbb{Z}.$$

The plots compare smoothing the full signal with retaining its supplied jump packet. Every plotted sample is a rational number computed from the displayed formula.



The cancellation is the theorem

Define $\tilde{f} = Qs + j$. Then, sample by sample,

$$\boxed{f - \tilde{f} = s - Qs.}$$

By contrast, linearity gives

$$f - Qf = (s - Qs) + (j - Qj).$$

The second error includes a new term caused by smoothing the discontinuity. In this finite example that term is concentrated near the two jump boundaries. The identity itself requires no linearity of the approximation applied to s. Linearity is used only in the comparison with Qf.

The input is not discovered by subtraction

The data here include the correct j. Replacing it with a wrong $\hat{j}$ retains the error $j - \hat{j}$, even when s is approximated perfectly. Multiple amplitudes and overlapping shapes are allowed if their full sum is supplied, but the identity does not estimate that sum from observations. Our finite periodic smoother is an illustration, not a new continuum approximation theorem. Its chosen boundary convention is part of the experiment.

> **Read this as:** Known-edge reconstruction removes the jump from the approximation problem. It does not remove the need to identify the jump.

Plate V.16 / A CONTINUOUS PULSE / EXACT ENDPOINT VALUES

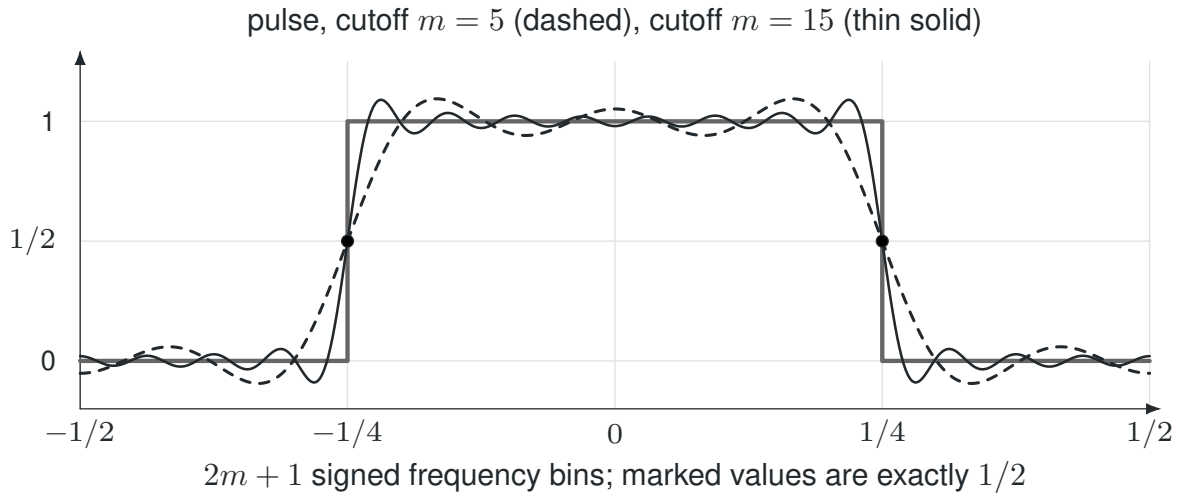
Why a Fourier cutoff cannot reproduce this jump pointwise

Chapter V: Section 6, Section 8.

Use a separate period-one example on $[-1/2, 1/2)$:

$$j(t) = \mathbf{1}_{\{|t|<1/4\}}, \qquad S_m j(t) = \tfrac{1}{2} + \frac{2}{\pi}\sum_{n=1}^{m} \frac{\sin(\pi n/2)}{n}\cos(2\pi n t).$$

This is the exact finite Fourier formula. The drawn curves sample it numerically. The endpoint calculation below is symbolic.



One value is fixed at every cutoff

At $t = \pm 1/4$, every summand vanishes because $\sin(\pi n/2)\cos(\pi n/2) = 0$ for integer n. Hence

$$S_m j(\pm 1/4) = \tfrac{1}{2} \qquad \text{for every } m \geq 0.$$

A larger cutoff resolves a narrower transition, but it does not make the finite trigonometric polynomial discontinuous. The two displayed cutoffs retain 11 and 31 signed frequency bins, respectively, not five and fifteen coefficients.

An exact known-packet comparison

Take $s(t) = \frac{1}{4}\cos(2\pi t) + \frac{1}{10}\cos(6\pi t)$. For $m \geq 3$ this smooth packet is already in the retained Fourier span. Thus $S_m s + j = s + j$ exactly in this special model, whereas $S_m(s + j)$ still replaces the pulse by $S_m j$. The values assigned at the two endpoints matter for pointwise error, but do not affect an L^p norm for finite p. The illustrated ringing is not a root enclosure or an asymptotic error certificate.

> **Read this as:** Increasing spectral resolution and retaining a known discontinuity are different operations, with different pointwise consequences.

Plate V.17 / MASK GEOMETRY / TRANSLATION BY ONE SIXTEENTH

Small location errors, different norms

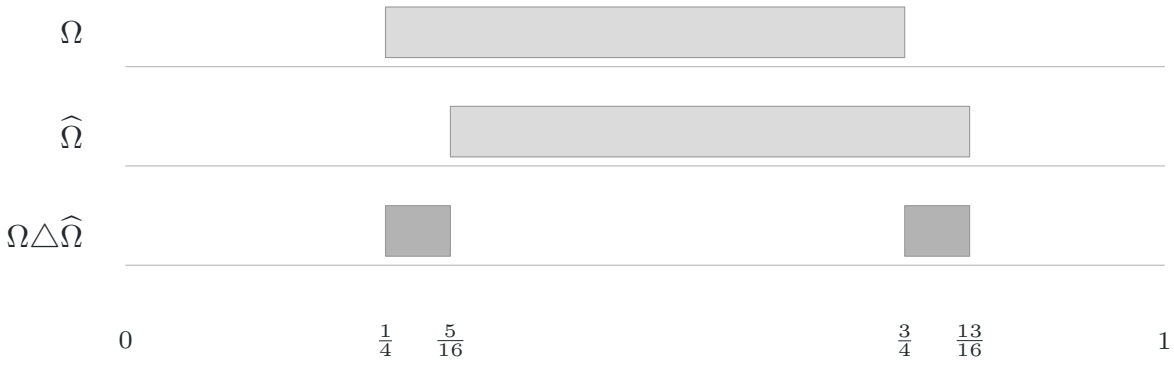
Chapter V: Theorem 7.1.

In Lebesgue measure on the line, let

$$\Omega = [\tfrac{1}{4}, \tfrac{3}{4}], \qquad \widehat{\Omega} = [\tfrac{5}{16}, \tfrac{13}{16}], \qquad a = \hat{a} = 3.$$

The two intervals have the same length. Their disagreement consists of one missed strip and one extra strip, each of width $1/16$.

$$\delta = 1/16, \qquad \mu(\Omega \triangle \widehat{\Omega}) = 2\delta = 1/8$$



$$\|j - \hat{j}\|_1 = 3/8, \qquad \|j - \hat{j}\|_2^2 = 9/8, \qquad \|j - \hat{j}\|_\infty = 3$$

$$|\Omega \triangle \widehat{\Omega}| = \tfrac{1}{8}, \qquad j - \hat{j} = 3(\mathbf{1}_\Omega - \mathbf{1}_{\widehat{\Omega}}).$$

error measure	L^1 norm	squared L^2 norm	L^∞ norm
exact value	$3/8$	$9/8$	3

Area shrinks while the missed jump keeps its height

For a translation by δ with $0 < |\delta| < 1/2$, the symmetric difference has measure $2|\delta|$. At finite p,

$$\|j - \hat{j}\|_p = 3(2|\delta|)^{1/p}.$$

It tends to zero as the displacement tends to zero. The essential supremum stays 3 for every nonzero displacement, because the error reaches height 3 on a set of positive measure. At exactly zero displacement all norms vanish. A nonempty disagreement set of measure zero would be different: it can affect the pointwise supremum while leaving the essential norm unchanged. This is why the chapter distinguishes the two statements.

> **Read this as:** A shrinking boundary strip can give small integrated error and still contain a full-height jump error.

Plate V.18 / COUNTING MEASURE / AN EXACT FINITE BUDGET

Separate amplitude error from mask error

Chapter V: Theorem 7.1.

On the 16 sample addresses, let $\Omega = \{6, \dots, 11\}$ and $\widehat{\Omega} = \{7, \dots, 12\}$. Use true amplitude $a = 3$ and estimate $\hat{a} = 5/2$. There are six true samples, five overlapping samples, and two disagreement samples.

$$a = 3, \quad \hat{a} = 5/2, \quad |\Omega| = 6, \quad |\Omega \triangle \widehat{\Omega}| = 2$$

	4	5	6	7	8	9	10	11	12	13	14
j	0	0	3	3	3	3	3	3	0	0	0
$\hat{j}$	0	0	0	$\frac{5}{2}$	$\frac{5}{2}$	$\frac{5}{2}$	$\frac{5}{2}$	$\frac{5}{2}$	$\frac{5}{2}$	0	0
$j - \hat{j}$	0	0	3	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{5}{2}$	0	0

$$\|j - \hat{j}\|_1 = 3 + 5(1/2) + 5/2 = 8$$

The residual splits exactly as

$$3\mathbf{1}_{\Omega} - \tfrac{5}{2}\mathbf{1}_{\widehat{\Omega}} = \tfrac{1}{2}\mathbf{1}_{\Omega} + \tfrac{5}{2}(\mathbf{1}_{\Omega} - \mathbf{1}_{\widehat{\Omega}}).$$

The missed sample has error 3, the five shared samples have error $1/2$, and the extra sample has error $-5/2$. With *counting measure*,

$$\|j - \hat{j}\|_1 = 8, \qquad \|j - \hat{j}\|_2^2 = \tfrac{33}{2}, \qquad \|j - \hat{j}\|_\infty = 3.$$

Read the theorem as a budget, not an equality claim

Here the L^1 bound is attained:

$$|3 - \tfrac{5}{2}|\,|\Omega| + \tfrac{5}{2}|\Omega \triangle \widehat{\Omega}| = \tfrac{1}{2} \cdot 6 + \tfrac{5}{2} \cdot 2 = 8.$$

The L^2 triangle bound is $\frac{1}{2}\sqrt{6} + \frac{5}{2}\sqrt{2}$, and need not be sharp. A background estimate adds another term $\|s - \hat{s}\|_p$. Cancellation between terms can make the total smaller than the bound. Normalized RMSE divides the L^2 norm by $\sqrt{16}$. A boundary-band RMSE uses the band size instead. Changing that denominator changes the reported quantity.

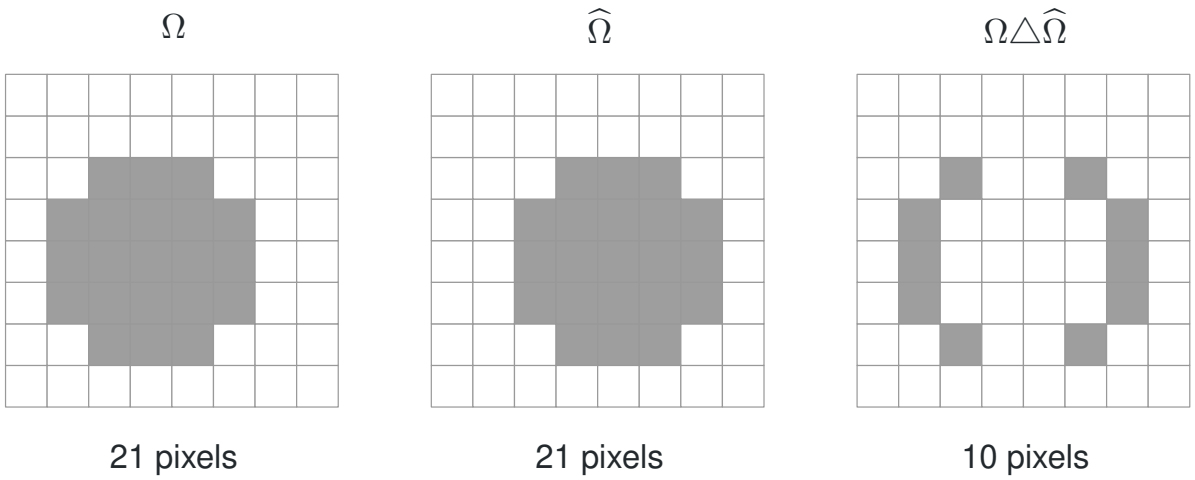
> **Read this as:** Mask error, amplitude error, and background error are separate mathematical inputs. A single reconstruction score conceals that separation.

Plate V.19 / A FINITE IMAGE / AN AMBIGUOUS OBSERVATION

Geometry localizes error while data may leave it ambiguous

Chapter V: Section 7, Section 9.

On the integer 8×8 grid, choose the disk-shaped mask $\Omega = \{(i,j) : (i-3)^2 + (j-3)^2 \le 5\}$ and translate it one pixel to the right. Each mask contains 21 pixels. Their symmetric difference contains exactly 10.



A geometric error calculation is not a segmentation algorithm

For a common amplitude a, the squared counting-measure error is $10a^2$, entirely on the disagreement pixels. The full-image mean squared error is $10a^2/64$. Both are exact, but answer different normalization questions. Now consider the four-sample observation

$$y = (0, \tfrac{3}{2}, 3, 3).$$

It is consistent with either true signal

$$u = (0, 0, 3, 3), \qquad v = (0, 3, 3, 3)$$

under deterministic sup-norm noise at most $3/2$. Indeed $y = (u+v)/2$, and both error norms equal $3/2$. A rule receiving only y and that noise bound cannot be guaranteed to select both distinct true masks correctly.

What additional structure would have to do

A lower noise level, separation, contrast, geometry, or a specified prior can change the identification problem. The example rules out a guarantee at the *stated* noise radius and signal pair, not all recovery results. The manuscript's positive-contrast estimator is deliberately narrower than a universal segmentation method.

> **Read this as:** Exact synthesis can preserve an identified boundary. It cannot decide between two boundaries that the observation model makes indistinguishable.

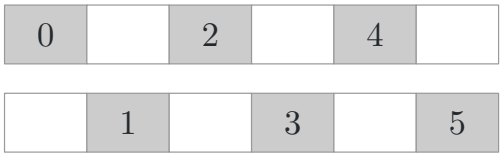
Plate V.20 / POINTWISE SIGNALS VERSUS CAUCHY COEFFICIENTS

Two products, two packet laws

Chapter V: Theorem 12.1.

The word “product” must be typed. Multiplying two signal values at the same time address is not the same operation as multiplying generating series. The distinction changes which packets can interact.

POINTWISE SIGNAL PRODUCT



$$(P_{2,0}x)\,(P_{2,1}y) = 0$$

CAUCHY COEFFICIENT PRODUCT

0	1	2
1	2	0
2	0	1

output label $(i + j) \bmod 3$

$$A = 1 + 2t + 3t^2,\ B = 2 + t^2 : \qquad P_{3,0}(AB) = 2 + 2t^3$$

Pointwise multiplication preserves the same address

For time-residue masks, $(P_{N,i}x)[n](P_{N,j}y)[n] = 0$ when $i \neq j$. Therefore

$$P_{N,k}(x \cdot y) = (P_{N,k}x) \cdot (P_{N,k}y).$$

Here the dot means multiplication at the same sample, not convolution.

Cauchy multiplication adds degrees

For formal series $A(t), B(t)$, the coefficient of t^n is a sum over $a + b = n$. Consequently the output residue is $i + j \pmod N$:

$$P_{N,k}(AB) = \sum_{i+j \equiv k\ (N)} (P_{N,i}A)(P_{N,j}B).$$

For $N = 3$, take $A = 1 + 2t + 3t^2$ and $B = 2 + t^2$. Direct multiplication gives

$$AB = 2 + 4t + 7t^2 + 2t^3 + 3t^4, \qquad P_{3,0}(AB) = 2 + 2t^3.$$

The $2t^3$ term comes from residues $1+2$, not $0+0$. At power zero, the empty product contributes 1 only in residue zero. These are formal coefficient identities and require no convergence assumption.

> **Read this as:** Before applying a packet multiplication rule, specify whether multiplication acts on sample values or on series coefficients.

Plate V.21 / FULL INPUT TUPLES / EXACT CYCLIC INDEXING

Nonlinear neighbors cross channel boundaries

Chapter V: Section 12.

For the periodic running signal define

$$F(x)[n] = x[n] + \tfrac{1}{2}x[n]x[n-1].$$

This is nonlinear and not a convolution operator. Nevertheless its two-channel representation is exact, provided the neighbor comes from the correct channel and index.

$$F(x)[n] = x[n] + \tfrac{1}{2}x[n]x[n-1]$$

$$F_0[n] = e[n] + \tfrac{1}{2}e[n]o[n-1] \qquad F_1[n] = o[n] + \tfrac{1}{2}o[n]e[n]$$

$$e[1] = 1/2, \quad o[0] = 1/4 \qquad o[1] = 1/4, \quad e[1] = 1/2$$

$$F(x)[2] = 9/16 \qquad F(x)[3] = 5/16$$

phase wrap changes the neighbor address; it does not remove the mixed term

Put $e[q] = x[2q]$ and $o[q] = x[2q+1]$, with channel indices taken modulo eight. Then

$$F(x)[2q] = e[q] + \tfrac{1}{2}e[q]o[q-1],$$
$$F(x)[2q+1] = o[q] + \tfrac{1}{2}o[q]e[q].$$

The even output needs the *previous* odd sample. The odd output needs the current even sample. This is the same carry in the address system that appeared in the one-sample delay matrix.

Check two actual outputs

The running values are $x[1] = 1/4$, $x[2] = 1/2$, and $x[3] = 1/4$, so

$$F(x)[2] = \tfrac{1}{2} + \tfrac{1}{2}\tfrac{1}{2}\tfrac{1}{4} = \tfrac{9}{16}, \qquad F(x)[3] = \tfrac{1}{4} + \tfrac{1}{2}\tfrac{1}{4}\tfrac{1}{2} = \tfrac{5}{16}.$$

Applying the same formula independently within each decimated array would use the wrong neighbors. Keeping the full channel tuple avoids that loss of information. The supplied suite compares every output sample in the two implementations using exact rational arithmetic. No convergence, contraction, or approximation property follows merely from writing these exact coordinates. Those require hypotheses on the nonlinear map and the problem being solved.

> **Read this as:** Exact packet coordinates remain available for nonlinear maps. They do not turn cross-channel interactions into independent channel problems.

Plate V.22 / THE CAYLEY CHART / EXACT QUARTER TURNS

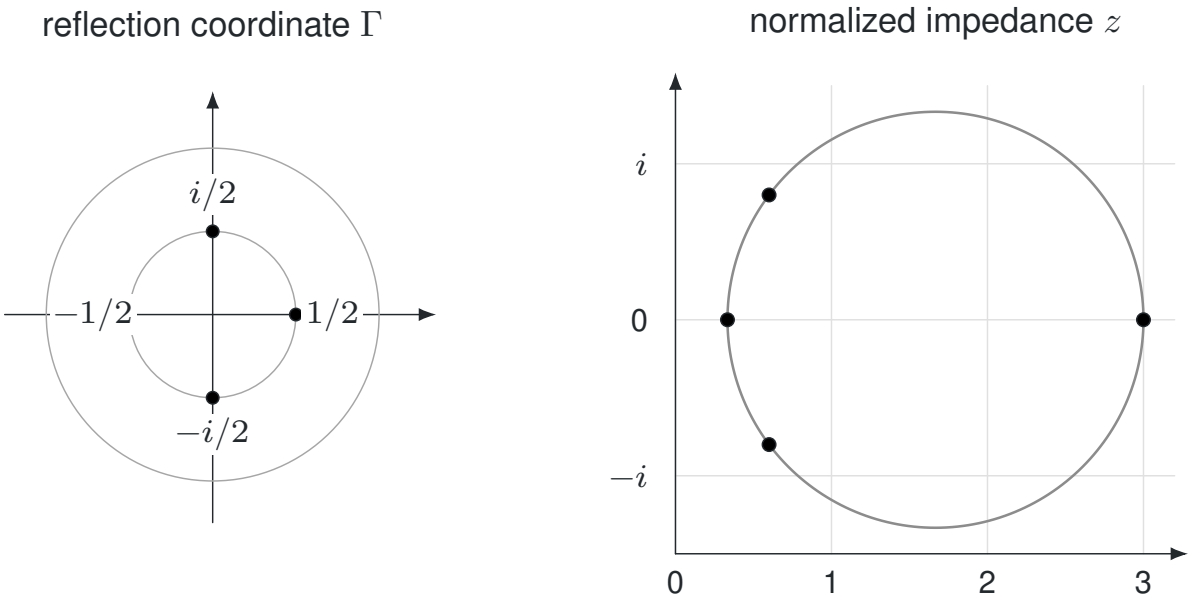
One reflection circle, four very different impedances

Chapter V: Theorem 13.2.

Use normalized impedance z and reflection coefficient Γ:

$$\Gamma = \frac{z-1}{z+1}, \qquad z = \frac{1+\Gamma}{1-\Gamma}.$$

The circle $|\Gamma| = 1/2$ is a constant-mismatch orbit. Its four quarter-turn points map to the exact impedances shown below.



$$3, \quad \tfrac{3}{5} + \tfrac{4}{5}i, \quad \tfrac{1}{3}, \quad \tfrac{3}{5} - \tfrac{4}{5}i$$

Γ	$1/2$	$i/2$	$-1/2$	$-i/2$
z	3	$3/5 + 4i/5$	$1/3$	$3/5 - 4i/5$

The coordinates preserve structure, not Euclidean distances

A quarter turn is a rotation in the reflection disk. After pulling back, it is a Möbius map, not a Euclidean rotation of the impedance plane. Equal reflection magnitudes need not give equal impedance magnitudes. The opposite points $1/2$ and $-1/2$ become 3 and $1/3$. The continuous impedance locus is the circle $(\operatorname{Re} z - 5/3)^2 + (\operatorname{Im} z)^2 = 16/9$, also drawn in the plate. Finite passive impedances with $\operatorname{Re} z \geq 0$ map to the closed disk except $\Gamma = 1$. That missing point is obtained only by adjoining impedance infinity. The interior maps to the open right half-plane. The classical Smith grid supplies the geometry. The cyclic labels specify exact rotations on it. Neither a new physical law nor a new optimal sampling theorem is implied by this coordinate description.

> **Read this as:** A simple action in one chart can be nonlinear in another. Conjugacy preserves the action law, not the apparent shape of a displacement.

Plate V.23 / HOMOGENEOUS COORDINATES / NO DIVISION BY ZERO

A pole is a chart boundary, not a broken group action

Chapter V: Theorem 13.2, Corollary 13.3.

Represent impedance by a nonzero pair $[u : v]$, where $[cu : cv] = [u : v]$ for $c \neq 0$. A finite impedance is $[z : 1]$. Infinity is $[1 : 0]$. For a nonzero reflection multiplier w, use

$$A(w) = \frac{1}{2}\begin{pmatrix} 1+w & 1-w \\ 1-w & 1+w \end{pmatrix}.$$

$$A(i) = \tfrac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$



$z = 0$	$z = -i$	$z = \infty$	$z = i$
$\Gamma = -1$	$\Gamma = -i$	$\Gamma = 1$	$\Gamma = i$

$A(i)^4 = I$: the fourth step returns to $[0 : 1]$

The action is defined before a ratio is taken

The determinant is w, so the matrix cannot send a nonzero pair to zero. Direct multiplication gives

$$A(w)A(w') = A(ww'), \qquad A(1) = I.$$

For $w = i$, the orbit through zero is

$$[0 : 1] \longmapsto [-i : 1] \longmapsto [1 : 0] \longmapsto [i : 1] \longmapsto [0 : 1].$$

The third point is a legitimate projective point, even though a finite impedance formula has a zero denominator there. The second iterate is $A(-1)$, which swaps the two coordinates. Thus a half turn exchanges 0 and ∞ and sends each finite nonzero z to $1/z$. The matched point $\Gamma = 0$, equivalently $z = 1$, is fixed by every phase multiplier. It is not part of the boundary orbit above. The exact checks test homogeneous pairs and their rescaling equivalence, rather than introducing an artificial complex value for $1/0$.

> **Read this as:** Compute with homogeneous pairs first. A denominator becomes a chart-selection issue rather than an exception to the cyclic action.

Plate V.24 / DYADIC–ODD FACTORIZATION / TWELVE ADDRESSES

Residue labels are not impedance averages

Chapter V: Proposition 14.1.

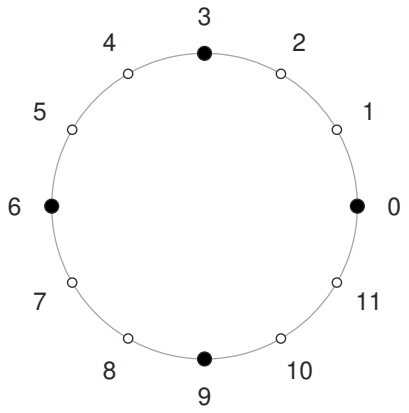
At phase order $12 = 4 \cdot 3$, the map

$$k \longmapsto (k \bmod 4,\ k \bmod 3)$$

is a bijection from twelve residue addresses to the 4×3 grid. Every cell below contains its unique original address.

	0	1	2
0	0	4	8
1	9	1	5
2	6	10	2
3	3	7	11

row $j \bmod 4$, column $j \bmod 3$



filled: the four coarse rays; open: the eight new rays

The factorization is an intersection of conditions

For integer sample addresses, including negative ones,

$$P_{12,k} = P_{4,k \bmod 4}\, P_{3,k \bmod 3}.$$

The product on the right is composition of two residue masks: both congruences must hold. Coprimality makes their intersection a single class modulo twelve. It is not a formula for multiplying or adding impedances.

Nested samples and nested projectors are different statements

The four coarse reflection rays occur at indices $0, 3, 6, 9$ of the twelve-ray grid. Thus the geometric sample set is nested. Separately, a coarse coefficient projector is the sum of its fine descendants in a common coefficient space. There is no corresponding rule saying that a coarse impedance is the average of its descendants. The inverse Cayley map is nonlinear. Even opposite reflection points have impedances 3 and $1/3$ in the preceding example. Their average $5/3$ is not the impedance 1 of their averaged reflection coefficient zero.

> **Read this as:** CRT factors addresses. Refinement conserves coefficient packets. Neither operation authorizes linear averaging through a nonlinear chart.

Plate V.25 / LOCAL DERIVATIVES / FINITE INCREMENTS

Sensitivity grows before the pole is reached

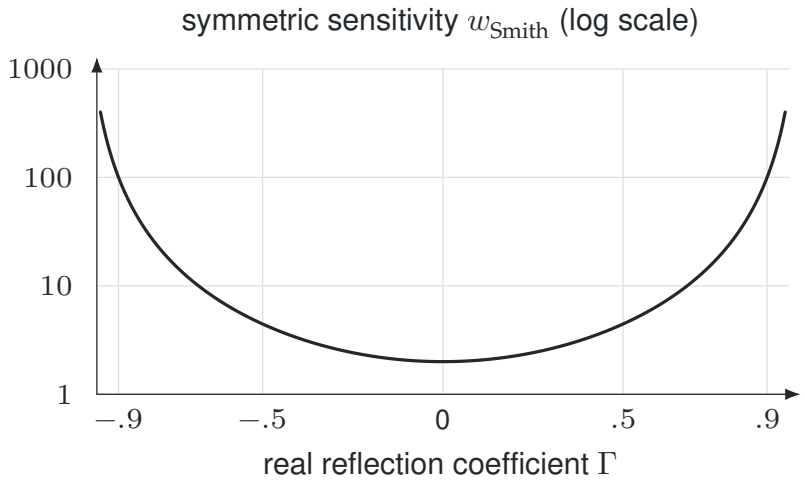
Chapter V: Section 14.

Away from $\Gamma = 1$, the inverse Cayley map satisfies

$$\frac{dz}{d\Gamma} = \frac{2}{(1-\Gamma)^2}.$$

Along the real axis, three exact derivatives already show the scale change:

$$\Gamma = \tfrac{1}{2} : 8, \qquad \Gamma = \tfrac{9}{10} : 200, \qquad \Gamma = \tfrac{99}{100} : 20000.$$



Γ	$\lvert dz/d\Gamma \rvert$
$1/2$	8
$9/10$	200
$99/100$	20000

$$\Delta z = 20/9 \neq 2$$

A differential is not the full finite-error formula

For a finite increment ε, with both denominators nonzero,

$$z(\Gamma+\varepsilon) - z(\Gamma) = \frac{2\varepsilon}{(1-\Gamma-\varepsilon)(1-\Gamma)}.$$

At $\Gamma = 9/10$ and $\varepsilon = 1/100$, the actual increment is $20/9$, while derivative times increment is 2. Both are calculated exactly. Their difference comes from using a differential in place of a finite increment. The derivative gives a first order approximation.

What a proposed weight can and cannot certify

Impedance is sensitive near $\Gamma = 1$, and admittance near $\Gamma = -1$. The symmetric weight

$$w(\Gamma) = |1-\Gamma|^{-2} + |1+\Gamma|^{-2}$$

reflects both chart sensitivities. Its growth motivates denser sampling near the endpoints, but does not establish an optimal grid, interpolation rule, or measurement allocation. A finite-error estimate must also control how close the perturbed point comes to either pole.

Read this as: Use the exact finite difference for an error budget. A large derivative diagnoses conditioning. Choosing an optimal algorithm requires further input.

Plate V.26 / EVIDENCE GUIDE / THE FORMAL CONTRACT IS UNCHANGED

What the reader can now verify

Chapter V: Section 21 (formal correspondence), Section E15.

Three worked structures, three distinct kinds of conclusion

Structure	What is established in the original worked layer
Finite signal	Exact signed addressing, filter transfer, Haar reconstruction and energy, sparse coefficient error, and full-tuple nonlinear evaluation.
Finite packet tree	All 26 complete prunings, actual projector sums, the unique cost-9 optimum, and a strict perturbation-margin certificate.
Reflection orbit	Exact rational-complex coordinates, a pole-safe projective action, CRT labels, and finite sensitivity calculations.

What each evidence layer means

Manuscript theorems. The Part I formal correspondence maps twenty theorem-like statements to 225 supporting Lean targets in 23 proof modules. Its compiler and axiom records provide the retained execution evidence. The formal companion gives their scope and reproduction instructions. The additional Part II developments remain outside this twenty-result map.

Recorded exact examples. The symbolic visual suite was rerun for the recorded monograph review. All 181 rational, algebraic, symbolic, and exhaustive finite checks passed. Seven deliberately false assertions were rejected. Numerical samples draw the continuous Fourier and sensitivity curves. They are not promoted to interval enclosures or universal error bounds.

Recorded executions. The recorded checker passed 20 exact tests containing 305,933 assertions and passed 16 verifier tests, including positive and negative controls. The numerical audit of the expanded source passed 11,052 finite checks, mixing exact and numerical calculations. The new extension examples separately passed 38 grouped exact checks and four false controls. These counts measure different operations and are not combined as theorem counts.

Reproducibility. The package preserves the original sources and their reports. Run `python3 build.py` here for an isolated PDF build. The extension finite checker is `checks/verify_extension_examples.py`. Its contract and raw execution manifest are included. The original visual checker requires the listed optional dependencies. PDF assembly does not run Lean or either mathematical checker.

The remaining boundaries are not decorative disclaimers

Known-edge synthesis requires the correct packet. A finite cost optimum depends on the supplied objective. An infinite packet tree, a universal segmentation procedure, minimax recovery rates, and optimal Smith sampling require additional theorems. The core's broad edge-recovery problems are not resolved by these finite examples. Plate V.27 introduces the additional separated-jump recovery results and their explicit hypotheses. The 16-sample running signal is not the 1024-sample manuscript experiment. The 26-pruning example is not its 458,330-pruning benchmark. Their smaller sizes allow every relevant number to be inspected exactly.

> **Read this as:** Coordinates, approximation, identification, optimization, and physical interpretation are separate layers. The examples make their interfaces visible without erasing their hypotheses.

Plate V.27 / PART II / REPRESENTATION BECOMES CONDITIONAL RECOVERY

What the extended developments add

Chapter V: Section 21 (formal correspondence), Section E1, Section E15.

The first twenty-six pages follow the corrected twenty-result formal core. The traditional chapter's second part restores the broader mathematical developments of the earlier expanded manuscript. The next thirteen worked pages make their additional hypotheses and conclusions visible.

Four transitions, with the missing input made explicit

Transition	Additional input	Resulting control
Residues to lattices	A finite-index sublattice and one representative per coset	Exact addresses in several dimensions
Invertibility to stability	Positive lower frame bound and bounded perturbation	Quantified coefficient and reconstruction error
Known jumps to recovery	Separated jumps, sampling, a Lipschitz background, and a strict contrast gap	Exact jump cells and bounded location/amplitude error
Recovered packets to approximation	A bounded linear approximation operator and its norm	A complete two-term error estimate

The two parts have different formal evidence

The inherited Lean map covers the twenty designated results of Part I through its 225 supporting targets. The additional traditional proofs in Part II are not silently included in that map. The following pages give mathematical explanations and exact worked specializations of those developments, with a separate checker for the finite examples. The original manuscript's broad edge-recovery questions remain broader than separated one-dimensional jumps. Conditions such as one jump per sample cell, an available wavelet support, and a positive detection margin do real work. In particular, the ambiguous observation on Plate V.19 still rules out an unrestricted recovery guarantee at its stated noise radius.

How to read the extension

Plates V.28–V.30 extend coordinates and stability. Plates V.31–V.34 identify what a singular packet contains and measure its error. Plates V.35–V.39 give three detection mechanisms. Plate V.40 closes the route by combining packet accuracy with a specified approximation operator.

> **Read this as:** A positive recovery theorem can coexist with an impossibility example. Their sampling models, noise bounds, and signal classes decide which conclusion applies.

Plate V.28 / LATTICE COORDINATES / A RECTANGULAR EXAMPLE

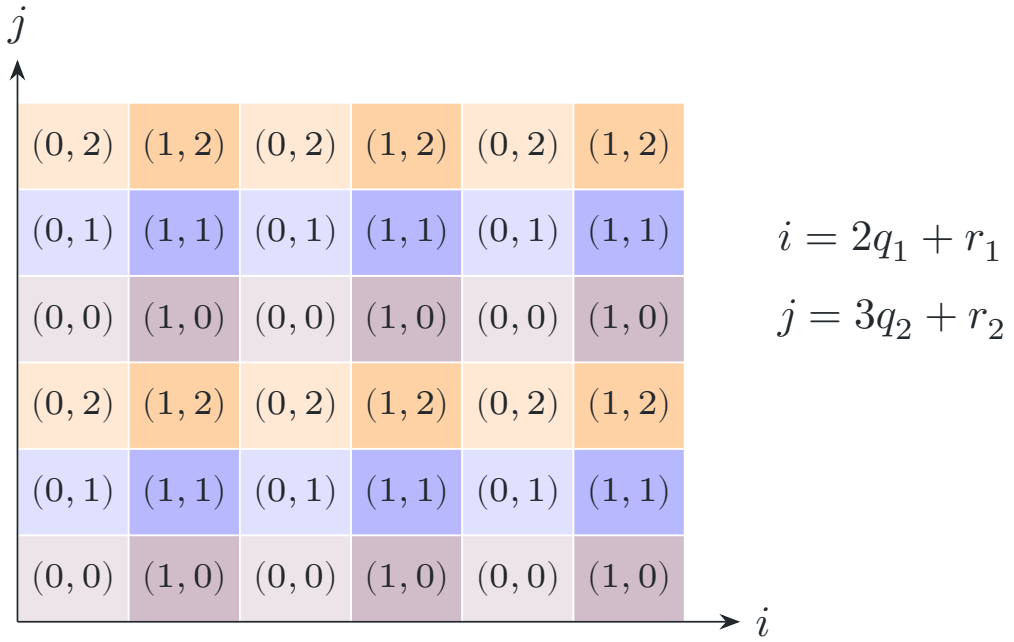
Six image phases are six exact address classes

Chapter V: Theorem E3.2, Theorem E3.5, Remark E2.2.

Take $L = 2\mathbb{Z} \times 3\mathbb{Z}$ inside $\mathbb{Z}^2$, with representatives

$$R = \{0, 1\} \times \{0, 1, 2\}.$$

Every address has a unique expression $(i, j) = (r_1, r_2) + (2q_1, 3q_2)$. The six channels are $x_r(2q_1, 3q_2) = x(r_1 + 2q_1, r_2 + 3q_2)$.



Each cell shows its residue label. Colors identify six disjoint classes. They do not average neighboring image pixels.

A negative address is still unique

The point $(-1, 7)$ has representative $(1, 1)$ and lattice displacement $(-2, 6)$, hence channel index $(-1, 2)$. Floor division gives the correct answer in both coordinates. For finite p, the channel norm uses the sum of the pth powers. At $p = \infty$, it uses their maximum. Both equal the norm on the original lattice because the cosets partition all addresses. In ℓ^∞, this coordinate identity does not imply that translates of an impulse are dense: their closure is only c_0.

Refinement partitions a class again

For $L' = 4\mathbb{Z} \times 6\mathbb{Z} \subset L$, the coarse representative $(1, 2)$ has fine representatives

$$(1, 2),\ (3, 2),\ (1, 5),\ (3, 5).$$

Their original-lattice masks sum to the coarse mask. As on Plate V.4, decimated arrays are interleaved, not added entrywise.

> **Read this as:** Dimension changes the address tuple. Unique coset decomposition still supplies exact synthesis and the correct norm identity.

Plate V.29 / FRAME STABILITY / AN EXACT TWO-CHANNEL BANK

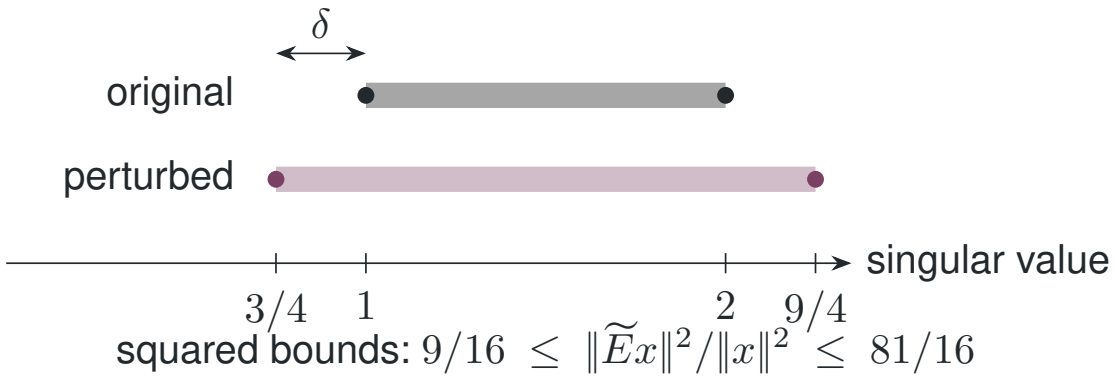
A positive singular-value gap survives perturbation

Chapter V: Theorem E5.2, Theorem E5.7, Corollary E5.6.

Consider the constant analysis bank

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \qquad E^*E = \operatorname{diag}(1, 4).$$

It has frame bounds $A = 1$, $B = 4$. For every perturbation with operator norm at most $\delta = 1/4$, the squared-norm bounds remain at least $(1 - 1/4)^2 = 9/16$ and at most $(2 + 1/4)^2 = 81/16$.



Both bounds can be attained

Choose $\Delta E = \operatorname{diag}(-1/4, 1/4)$. Then $\widetilde{E} = \operatorname{diag}(3/4, 9/4)$ has precisely those two frame bounds. Its inverse has operator norm $4/3$. A coefficient error $\nu = (3/4, 0)$ therefore produces signal error $(1, 0)$, attaining the noise bound. The strict condition $\delta < \sqrt{A}$ prevents the lower bound from vanishing. At $\delta = 1$, the perturbation $\operatorname{diag}(-1, 0)$ destroys the first channel entirely.

An inverse and an approximate inverse answer different questions

If the original synthesis $R = \operatorname{diag}(1, 1/2)$ is retained after this perturbation, then

$$R\widetilde{E} - I = \operatorname{diag}(-1/4, 1/8).$$

Its defect norm is $1/4$. Every input has relative reconstruction error at most that value, with equality on the first coordinate. For frequency-dependent banks the theorem uses an essential supremum on the unit circle, not a few sampled frequencies.

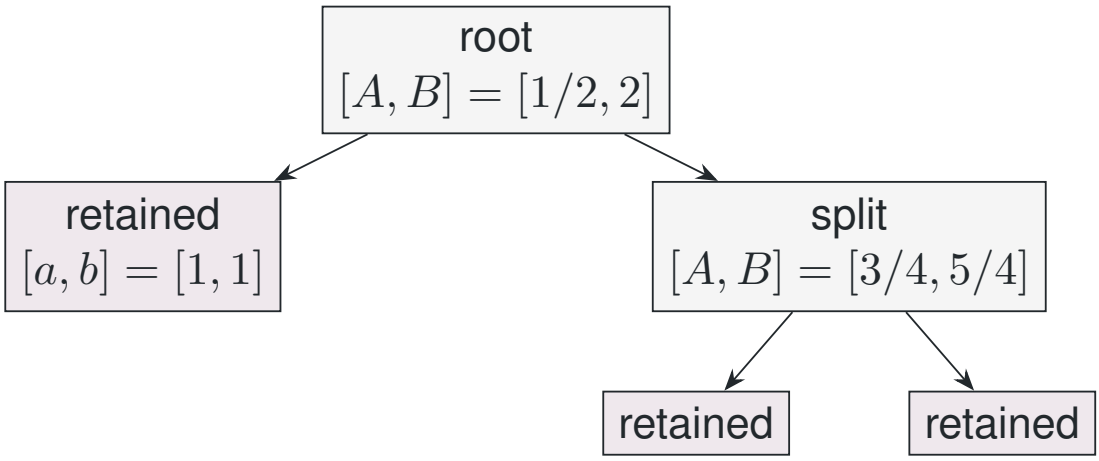
> **Read this as:** A lower singular-value gap quantifies stable recovery. A reconstruction defect quantifies the error of the actual synthesis bank being used.

Plate V.30 / LOCAL FRAME CONSTANTS / A MIXED-DEPTH CERTIFICATE

Nonunitary tree bounds propagate along branches

Chapter V: Proposition E6.4.

Suppose the root split has nonnegative squared-norm bounds $A = 1/2$ and $B = 2$. Retain the left child. Split the right child with bounds $3/4$ and $5/4$, then retain both grandchildren.



Start with one at every retained node

The left subtree has bounds $a_0 = b_0 = 1$. The right subtree has $a_1 = 3/4$ and $b_1 = 5/4$. The recursive certificate gives

$$a_{\mathrm{root}} = \tfrac{1}{2}\min\{1, \tfrac{3}{4}\} = \tfrac{3}{8}, \qquad b_{\mathrm{root}} = 2\max\{1, \tfrac{5}{4}\} = \tfrac{5}{2}.$$

Consequently the total leaf coefficient energy lies between $3/8$ and $5/2$ times the root energy. These are guaranteed bounds. No assertion that every admissible collection of local maps attains both constants is needed.

Why a minimum and maximum appear

After the root split, its energy may distribute unevenly between the two children. The lower estimate must survive the least favorable child lower bound. The upper estimate must survive the largest child upper bound. Replacing these by an unweighted average would presume an energy distribution that the hypotheses do not supply. When all local maps are unitary, every factor is one and the recursion recovers exact energy conservation. For general local maps, a positive lower bound alone establishes injectivity and stable inversion on the range. It does not by itself assert surjectivity onto every possible leaf tuple.

> **Read this as:** Exact tree coverage and good conditioning are separate inputs. The first chooses a coordinate route. The second bounds what that route does to norms.

Plate V.31 / TRACE DATA / A STEP AND A RAMP

Derivative jumps have their own packets

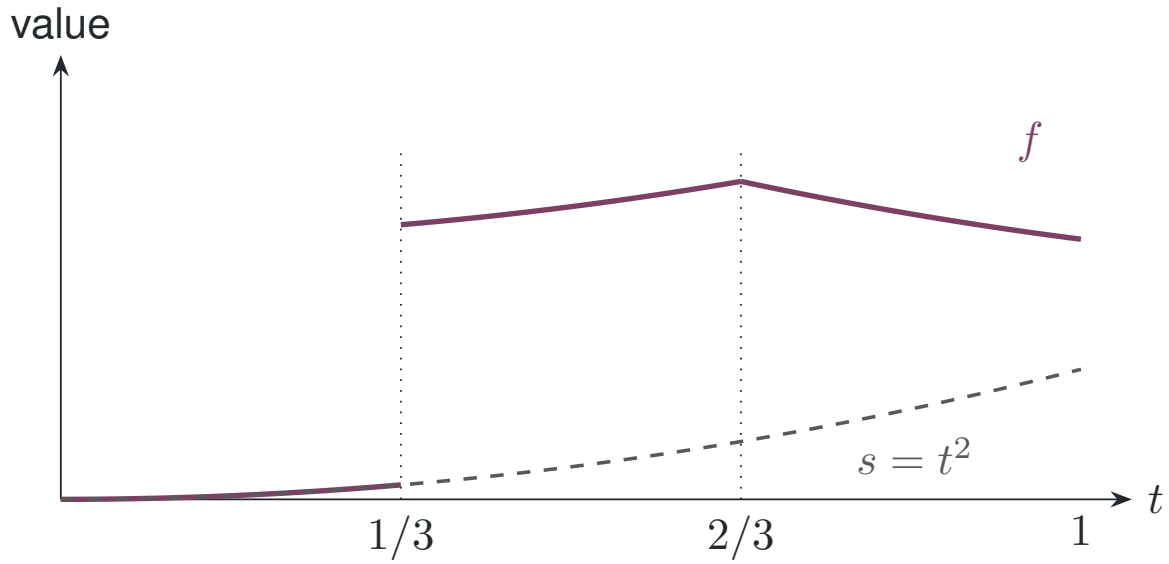
Chapter V: Theorem E7.2, Corollary E7.3.

On $[0,1]$, let $H_\tau = \mathbf{1}_{[\tau,1]}$ and take

$$f(t) = t^2 + 2H_{1/3}(t) - 3(t-2/3)_+.$$

The function has a value jump 2 at $1/3$ and a first-derivative jump -3 at $2/3$. Its packet and remainder are

$$j_f(t) = 2\phi_{0,1/3}(t) - 3\phi_{1,2/3}(t), \qquad s_f(t) = t^2.$$



The displayed curves are exact polynomial pieces. The solid jump is not connected across its discontinuity. The dashed curve is the global smooth remainder.

The normalization makes the trace coefficient literal

The general packet is $\phi_{m,\tau}(t) = (t-\tau)_+^m/m!$. Its derivative of order m has a unit trace jump, while its other classical trace jumps vanish. Thus its multiplier is exactly the jump being removed, with no hidden factorial. Here the remainder has matching value and first-derivative traces and belongs to $W^{2,p}(0,1)$ for every $1 \le p \le \infty$. Its second weak derivative is 2. The singular packet accounts for the distributional boundary terms that would otherwise arise at the breakpoints.

What the general theorem requires

Each piece must lie in the stated Sobolev class and possess the specified one-sided traces. Subtracting the finite trace packet then joins those pieces into a global Sobolev remainder. This is a theorem about finitely many declared breakpoints, not a representation of every possible singular measure or arbitrary image boundary.

> **Read this as:** A value jump uses a step. A slope jump uses a ramp. The derivative order determines the packet and its normalization.

Plate V.32 / ESTIMATED PACKETS / AN EXACT AMPLIFICATION EXAMPLE

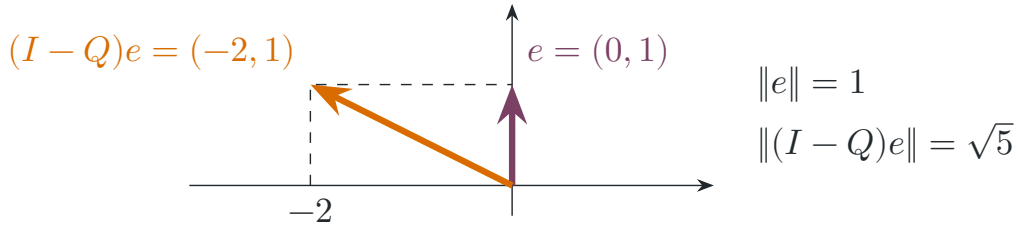
The approximation operator also acts on packet error

Chapter V: Theorem E7.8.

Let Q be a bounded *linear* operator and set $e = j - \hat{j}$. Substituting $f = s + j$ into $\hat{f} = Q(f - \hat{j}) + \hat{j}$ gives

$$f - \hat{f} = (I - Q)s + (I - Q)e.$$

The second term includes the action of Q on the packet error. Linearity is required for this split.



An oblique projection can increase the error

In $\mathbb{R}^2$, choose

$$Q = \begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix}, \qquad s = (1,0), \qquad e = (0,1).$$

Then $Q^2 = Q$ and $Qs = s$, but $(I - Q)e = (-2, 1)$. The smooth error is zero while the unit packet error becomes a reconstruction error of norm $\sqrt{5}$. For a fully specified realization, take $j = e$ and $\hat{j} = 0$. Then $f = (1, 1)$, $\hat{f} = Qf = (3, 0)$, and their difference is exactly $(-2, 1)$.

Use the actual operator bound

The estimate that always holds in this setting is

$$\|f - \hat{f}\| \le \|s - Qs\| + (1 + \|Q\|)\|e\|.$$

Here $\|Q\| = \sqrt{5}$, so the bound is conservative. For an orthogonal projection, $\|I - Q\| \le 1$ gives a sharper estimate. The operator's geometry determines which bound can be used.

> **Read this as:** Recovering the packet and approximating the background are separate tasks, but the chosen approximation operator couples their final errors.

Plate V.33 / REGION VALUES / EVERY MISCLASSIFICATION HAS A WEIGHT

An overlap matrix is an exact error ledger

Chapter V: Theorem E8.1, Theorem E8.4.

Let two piecewise-constant signals have region values $a = (0, 2, 5)$ and $b = (0, 2, 5)$ on a counting-measure domain of twelve samples. Suppose their overlap counts are

$$M = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 4 \end{pmatrix}.$$

The row totals $(4, 3, 5)$ are the true region sizes. The column totals are also $(4, 3, 5)$, although three samples are assigned to the wrong values.

	$b_0 = 0$	$b_1 = 2$	$b_2 = 5$
$a_0 = 0$	3	1	0
$a_1 = 2$	0	2	1
$a_2 = 5$	1	0	4

off-diagonal squared error

$$1 \cdot 2^2 + 1 \cdot 3^2 + 1 \cdot 5^2$$

$$= 38$$

Equal error counts need not mean equal reconstruction error

The three off-diagonal entries contribute one sample each. Their value differences have magnitudes 2, 3, and 5, so

$$\|j - \hat{j}\|_1 = 2 + 3 + 5 = 10, \qquad \|j - \hat{j}\|_2^2 = 4 + 9 + 25 = 38.$$

The counting-measure supremum error is 5. The full-domain mean squared error is $38/12 = 19/6$. A mismatch fraction of $3/12$ reports a different statistic and omits all three amplitude weights.

The formula requires partitions

For measurable partitions (E_i) and (F_j), with $M_{ij} = \mu(E_i \cap F_j)$,

$$\|j - \hat{j}\|_p^p = \sum_{i,j} |a_i - b_j|^p M_{ij}, \qquad 1 \le p < \infty.$$

Each location belongs to exactly one intersection. Overlapping indicator packets that are summed at the same location must first be converted to their actual combined value regions before this partition formula is applied.

> **Read this as:** Geometry supplies the overlap counts. Signal values supply the weights. Their combination gives the norm relevant to reconstruction.

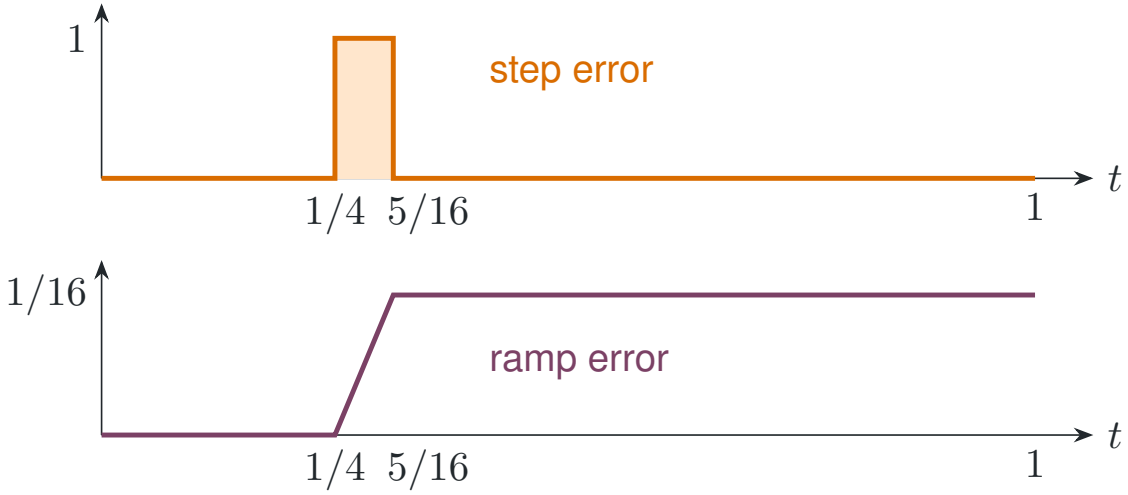
Plate V.34 / LOCATION STABILITY / A ONE-SIXTEENTH DISPLACEMENT

A shifted step and a shifted ramp have different moduli

Chapter V: Corollary E8.2, Theorem E8.3, Theorem E8.7.

Set $\tau = 1/4$, $\hat{\tau} = 5/16$, and $\delta = 1/16$. A unit step differs only on the interval between its two locations:

$$\|H_\tau - H_{\hat{\tau}}\|_2 = \sqrt{\delta} = \tfrac{1}{4}, \qquad \|H_\tau - H_{\hat{\tau}}\|_\infty = 1.$$



For a ramp, the error height shrinks too

Write $r_\tau(t) = (t - \tau)_+$. The difference grows linearly from zero to δ on $[\tau, \hat{\tau}]$ and stays at δ on $[\hat{\tau}, 1]$. Direct integration gives

$$\|r_\tau - r_{\hat{\tau}}\|_2^2 = \frac{\delta^3}{3} + (1 - \hat{\tau})\delta^2 = \frac{17}{6144} < \frac{1}{256} = \delta^2.$$

Thus the L^2 ramp error is at most δ, and its supremum error is exactly δ. Positive-order truncated powers are Lipschitz in location, with the order-dependent constants supplied by the theorem.

The geometric analogue needs a tube hypothesis

If $E \triangle F \subseteq N_\delta(\partial E)$ and the tube measure is bounded by $C\delta$ for the stated range $0 < \delta < \delta_0$, then an equal-amplitude packet has finite-p error at most $|a|(C\delta)^{1/p}$. A mere visual impression that two boundaries are close does not establish the inclusion or the tube estimate.

> **Read this as:** A displaced value jump keeps its full height on a smaller set. A displaced derivative-jump packet also decreases in height.

Plate V.35 / DETERMINISTIC RECOVERY / A NUMERICAL CERTIFICATE

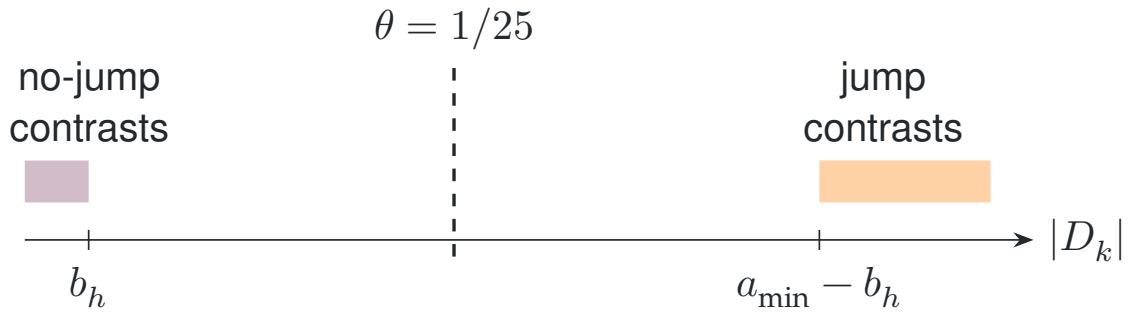
A strict contrast gap locates every separated jump cell

Chapter V: Lemma E9.1, Theorem E9.2, Example E9.7.

Assume a real signal is a Lipschitz background plus jumps. No jump occurs at a sample point, and each grid cell contains at most one jump. With spacing h, background constant M, and sample noise bounded by ε, write

$$b_h = Mh + 2\varepsilon.$$

A no-jump contrast has magnitude at most b_h. A jump contrast differs from its true amplitude by at most b_h.



Use the manuscript's exact numerical inputs

For $h = 1/256$, $M = 1/2$, $\varepsilon = 1/500$, and $a_{\min} = 2/25$,

$$b_h = \frac{381}{64000} < \frac{3}{500}, \qquad 2b_h < \frac{2}{25}.$$

The threshold $\theta = 1/25$ lies strictly between b_h and $a_{\min} - b_h$. Thresholding $|y_{k+1} - y_k|$ therefore identifies exactly the jump-containing cells. Their midpoints locate each jump within $1/512$, and each estimated amplitude differs by at most $381/64000$.

The recovered object is a cell, then an estimate

The theorem does not claim the exact real-valued breakpoint from a finite grid. It supplies the correct cell and a controlled representative of that cell. For real jumps, the amplitude error is too small to reverse the sign. Two opposite jumps in one cell could cancel in the contrast, which is why the separation assumption cannot be dropped. For Q jumps the finite-p packet bound contains both Qb_h and the location term $(\frac{h}{2}\sum_q |a_q|^p)^{1/p}$. A growing number of jumps or shrinking separation requires its own analysis.

> **Read this as:** Exact detection means exact cell identification under a strict margin. Location and amplitude remain quantitative estimates with explicit errors.

Plate V.36 / SUBGAUSSIAN RECOVERY / A CONSERVATIVE SAMPLE BUDGET

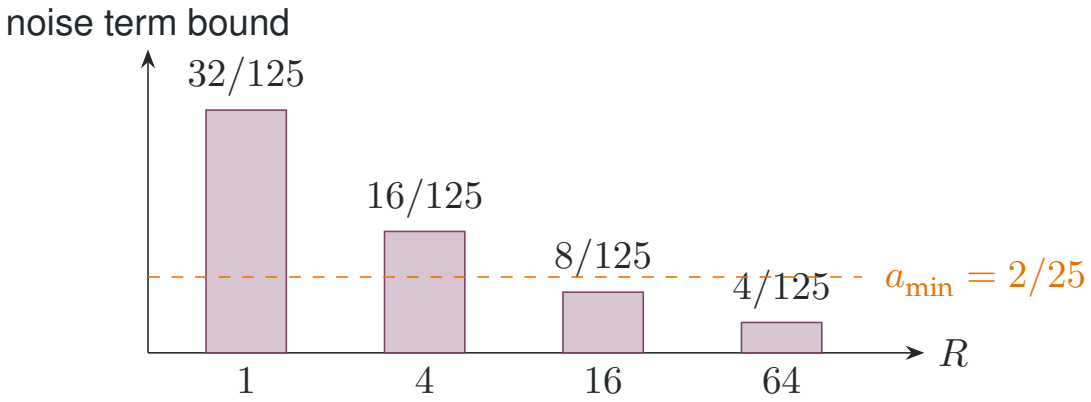
Repeated observations shrink the noise side of the gap

Chapter V: Theorem E9.8, Corollary E9.9, Corollary E9.11.

Suppose all noise observations are independent and centered, with moment-generating bound $\mathbb{E}e^{\lambda\eta} \le e^{\sigma^2\lambda^2/2}$. Averaging R observations at each of $n+1$ sample sites replaces σ by $\sigma/\sqrt{R}$. A sufficient recovery condition is

$$a_{\min} > 2Mh + \frac{4\sigma}{\sqrt{R}}\sqrt{\log(2n/\delta)}.$$

The probability of simultaneous recovery is then at least $1-\delta$, subject to the same separated-jump sampling model.



One explicit conservative choice

Take $n = 256$, $M = 1/2$, $\sigma = 1/50$, $\delta = 1/20$, and $a_{\min} = 2/25$. Since $\log(10240) < 10 < (16/5)^2$, choosing $R = 16$ makes the right-hand side strictly less than

$$\frac{1}{256} + \frac{1}{50}\frac{16}{5} = \frac{2173}{32000} < \frac{2}{25}.$$

Thus sixteen independent observations per sample site suffice for this bound. It is a sufficient budget, not a claim of optimal sample complexity. The exact checker certifies $e^{10} > 10240$ using a rational Taylor partial sum, so the displayed logarithm bound is not based on rounded numerical logarithms.

Dependence between adjacent contrasts is allowed

Neighboring differences share one sample. The proof does not require those differences to be independent: it bounds each tail and applies a union bound across all cells. Independence of the underlying sample noises supplies the stated sub-Gaussian difference proxy. For a fixed separated signal model, a noise scale satisfying $\sigma_n\sqrt{\log n} \to 0$ gives the stated recovery consistency in probability. A fixed positive noise scale with one observation per site does not satisfy that condition.

Read this as: More grid points and more repeated observations affect different terms. Finer spacing reduces background variation. Averaging reduces noise.

Plate V.37 / SCALE SEPARATION / A QUANTIFIED THRESHOLD

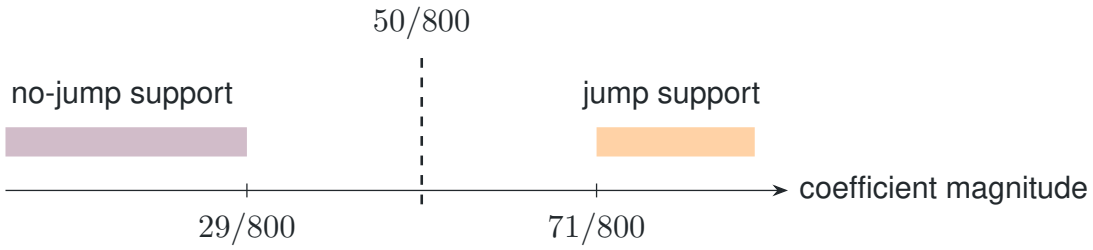
Haar sees a jump only through its position in the support

Chapter V: Theorem E10.1, Theorem E10.2, Corollary E10.3.

Use a normalized Haar wavelet on an available interval of length L on the real line, positive on its left half and negative on its right half. Here H_τ is the half-line step $\mathbf{1}_{[\tau,\infty)}$, so there is no artificial second endpoint jump. If u is the relative jump position in the support, then

$$|\langle aH_\tau, \psi\rangle| = |a|\sqrt{L}\min\{u, 1-u\}.$$

The coefficient vanishes when the jump lies exactly at a support boundary.



A worked gap with all three contributions

Take $L = 1/4$, $M = 1$, noise bound $\varepsilon = 1/100$, jump size $|a| = 1$, and relative boundary distance at least $\rho = 1/4$. The smooth and noise contributions together are bounded by

$$\frac{M}{4}L^{3/2} + \varepsilon\sqrt{L} = \frac{1}{32} + \frac{1}{200} = \frac{29}{800}.$$

The jump contributes at least $|a|\rho\sqrt{L} = 1/8$. The observed coefficient therefore has magnitude at least $71/800$. The threshold $1/16 = 50/800$ separates every such jump support from every support containing no jump.

The available support is part of the hypothesis

A fixed extension is needed if the observation domain ends inside a desired wavelet support. Also, the selected support must contain only the specified jump. An unmodeled second jump can reinforce or cancel the coefficient.

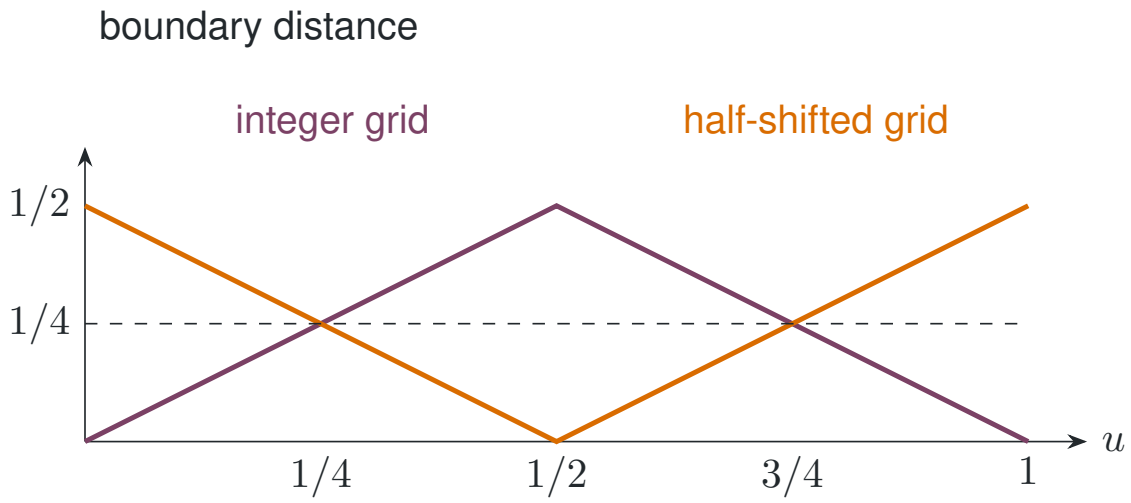
> **Read this as:** Scale determines the smooth contribution, but alignment determines the jump contribution. A detection guarantee needs both.

Plate V.38 / HAAR SUPPORT GEOMETRY / A QUARTER-CELL GUARANTEE

Two half-shifted grids remove exact alignment failure

Chapter V: Theorem E10.4, Corollary E10.5.

Normalize the support length to $L = 1$. The first grid has boundaries at integers. The second has boundaries at half-integers. For a point at position u modulo one, consider its distance to the nearest boundary of each grid.



At least one distance is one quarter or larger

If $1/4 \le u \le 3/4$, the original cell works. Outside that range, a shifted cell works. At $u = 1/8$, the original margin is $1/8$ but the shifted cell $[-1/2, 1/2)$ gives margin $3/8$. At $u = 1/4$, both margins equal $1/4$, showing that the uniform constant cannot be increased for these two grids. Restoring L, every jump has a support with boundary distance at least $L/4$ in one grid. If L is smaller than the minimum jump separation, that support contains no other jump.

Insert the geometric guarantee into the analytic estimate

The alignment-independent gap condition is

$$\frac{|a|}{4} > 2\varepsilon + \frac{M}{2}L.$$

For the numbers on Plate V.37, the right side is $1/50 + 1/8 = 29/200$, strictly below $1/4$. Hence one of the two grids supplies a detectable support for every jump, regardless of its position relative to the original grid. This certifies a useful support per jump. It does not by itself specify a unique rule for merging duplicate detections from two grids into exact breakpoint estimates.

> **Read this as:** A second grid repairs a geometric blind spot. The contrast inequality and separation hypotheses still supply the recovery guarantee.

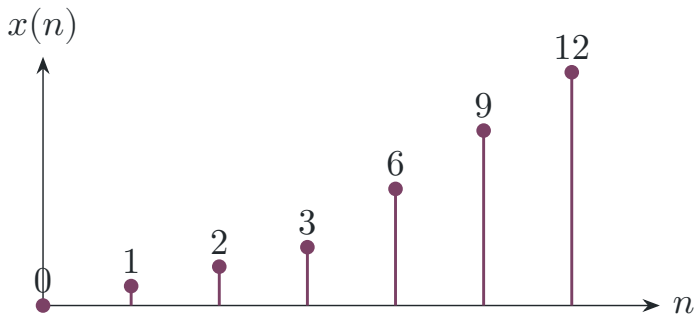
Plate V.39 / BOUNDARY STENCILS / SIGNAL RESPONSE AND NOISE GAIN

Higher differences annihilate polynomial interiors

Chapter V: Theorem E10.6, Proposition E10.7, Corollary E10.8.

Let $x(n) = n + 2(n-3)_+$ on the integer line. Its slope changes from one to three at the breakpoint 3. The second forward difference is

$$\Delta^2 x(n) = x(n+2) - 2x(n+1) + x(n).$$



$\Delta^2 x(2) = 2$

all other displayed second differences vanish

An exact small stencil calculation

The values at $n = 0, \dots, 6$ are $(0, 1, 2, 3, 6, 9, 12)$. The second differences at $n = 0, \dots, 4$ are $(0, 0, 2, 0, 0)$. The only nonzero stencil straddles the slope change. For a piecewise polynomial of degree at most d, Δ^{d+1} vanishes whenever its complete stencil lies inside one polynomial piece. Candidate responses are confined to at most $d+1$ starting addresses before each breakpoint. They are boundary bands, not necessarily exact breakpoint locations.

Differencing also acts on noise

The absolute binomial coefficients of Δ^m sum to 2^m. With $\|\eta\|_\infty \le 1/16$, second-difference noise is bounded by $4/16 = 1/4$. The true response 2 remains at least $7/4$, while all responses away from the breakpoint band remain at most $1/4$. Threshold one separates them. The general robust-band statement assumes that each breakpoint has at least one sufficiently large true response. A breakpoint can fail that assumption through its amplitude or local configuration. Annihilation away from boundaries does not prove a lower bound at every boundary.

> **Read this as:** A vanishing interior response is structural. Reliable detection also requires the boundary response to exceed the amplified noise.

Plate V.40 / THE TWO ERRORS / A COMPLETE READING CONTRACT

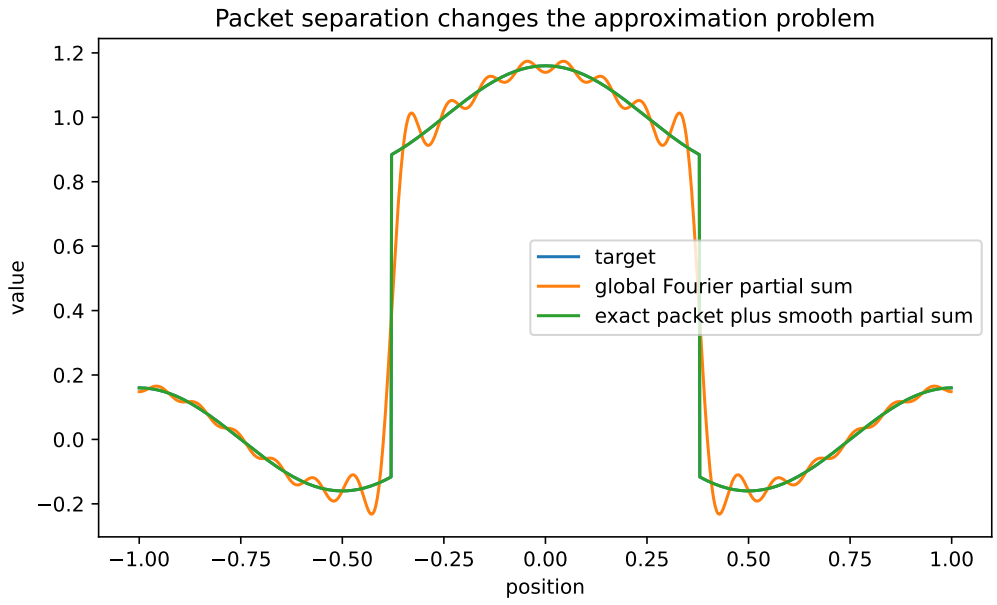
Conditional Gibbs control closes the reconstruction estimate

Chapter V: Theorem E11.2, Theorem E11.3.

The exact retained-packet approximation shares the same discontinuity as its target. If $f = s + j$ and $Q_m s \to s$ uniformly, then

$$\|f - (Q_m s + j)\|_\infty = \|s - Q_m s\|_\infty \longrightarrow 0.$$

The approximants themselves may be discontinuous. There is no conflict with the failure of continuous Fourier polynomials to converge uniformly to a jump.



Preserved vector figure from the expanded source. It illustrates the choice of approximation target. It is not an independent universal recovery certificate.

Estimated packets need the operator norm too

If the bounded linear operators satisfy $\sup_m \|Q_m\|_{L^p \to L^p} \le C_Q$, then

$$\|f - \hat{f}_m\|_p \le \|s - Q_m s\|_p + (1 + C_Q)\|j - \hat{j}_m\|_p.$$

For an orthogonal L^2 projection one may use $\|I - Q_m\| \le 1$, sharpening the packet multiplier to one. Such a bound must be justified in the norm actually being used. In particular, uniform boundedness of Fourier partial sums in the supremum norm is not supplied by this theorem.

What the complete guide establishes

Plates V.1–V.26 preserve the original worked layer. Plates V.27–V.40 explain the additional traditional developments, outside the inherited twenty-result Lean map. The new exact examples have their own finite arithmetic record. They do not extend the formal coverage merely by appearing beside formally mapped results. The final package keeps recorded computations and inherited compiler reports attached to their stated scope. The symbolic visual suite was rerun, with 181 checks passed and seven invalid controls rejected. Lean compiler evidence remains inherited. No fresh Lean run is claimed.

Read this as: A complete error claim specifies the signal class, the recovered packet, the approximation operator, and the norm. Each term can then be improved and checked on its own terms.

CHAPTER VI

Transfer Semigroups and Memory

Probabilistic refinement, fractional operators, and exact hidden-state elimination

Static descriptions now become evolution and conditioning problems.

Evolution, probability and conditioning share a transfer language. Eliminating hidden states produces memory terms whose hypotheses remain part of the model.

IN THIS CHAPTER

Mathematical text 511

Visual studies 616

Formal companion 1992

Sources and references 2678

Scope

Finite-state, common-gap and fixed-partition restrictions remain explicit. Completion, uniform refinement control and stable infinite latent-packet closure remain open. Floating residuals are not interval certificates.

The next chapter treats quantum seeds, operator grammars and finite effective dynamics.

How to use this volume

Part I. The thirty-three-result formal core, including nonnormal memory and canonical finite holomorphic naturality.

Part II. The complete strengthened mathematical development: bounded refinement towers, Bayesian information, Hermite transfer, quantitative memory compression, and competing risks.

Part III. Complementary results, exact examples, and the boundedoperator reciprocal calculus received from Chapter IV.

Part IV. Forty-two visual pages work through exact finite models, analytical examples, and the boundaries of the formal contract.

Reading this chapter

This chapter studies transfer semigroups in three complementary settings. Part I establishes the finite formal core. Part II develops bounded refinement towers, probabilistic conditioning and analytical realizations. Part III gives additional results on product-topology limits, reversible coarse dynamics, filtering, perturbation and operator substitution. Their hypotheses distinguish the constructions even where the underlying identities overlap.

The formal companion covers exactly the thirty-three scoped results of Part I. Its 341 supporting Lean theorem declarations lie in 47 proof modules. The wider namespace audit includes definitions and generated declarations. Formal coverage does not automatically extend to Parts II or III. The companion supplies the exact targets, assumptions and specialization boundaries.

Part I includes three explicitly open conjectures. Part II supplies bounded or finite theorems for several related questions. It does not settle arbitrary continuum limits, unbounded-generator convergence, general statistical sufficiency, or globally optimal nonnormal model reduction. The reader should keep each result's declared space, topology, positivity, and stability hypotheses. Part I's mass-only Bayesian commutation criterion assumes at least two coarse blocks. Part II's stronger sufficiency convention also preserves within-block conditionals, so its one-block case still requires a constant likelihood.

The worked atlas follows the mathematical development through exact finite models and analytical examples. Calculations illustrate stated ranges and assumptions. Their computational records remain distinct from both the written proofs and the formal declarations.

Read Part I for the common operator language and the exact formal contract. Read Part II for bounded Banach-space development and the broader probabilistic and analytical realizations. Read Part III for product-topology limits, reversible coarse dynamics, prior-restricted and coarse filtering, killed-transfer perturbations, rational realization, and the transferred operator reciprocal family. Section numbers in Part II have prefix E. Those in Part III have prefix R. Original labels and statement locations are mapped in the editable sources.

Contour winding, positive memory decay, finite moment closure and eventual absorption each require the hypotheses stated with their results. When the spaces vary along a refinement system, summability must be read in the specified ambient setting.

Part I The formal core

Projector-Resolved Seed Calculus VI

Transfer Semigroups: Stochastic, Bayesian, Fractional, and PDE Realizations

Overview

This chapter develops the transfer-semigroup realization of the projector-resolved seed calculus. A seed is propagated by a one-parameter semigroup, a Markov kernel, a normalized Bayesian transfer, a fractional spectral power, a Green operator, or a potential-weighted Feynman–Kac operator. Exact packet projectors resolve every transfer into input-output blocks, and the semigroup law becomes block convolution over intermediate packets. Coarse and fine generators are compared by a refinement defect. A Duhamel identity proves that this generator-level defect is exactly the source of the corresponding semigroup defect and gives a quantitative stability bound. In the stochastic realization, point-mass seeds generate random walks and continuous-time Markov evolution. Exact lumpability is the vanishing of the same refinement defect. Conditional expectation is treated as a weighted orthogonal packet projector, with nested partitions producing martingale-difference packets. Bayesian conditioning is a normalized nonlinear operator. Its independent-data composition law, packet-evidence decomposition, stability bound, conjugate updates, and continuous likelihood flow are proved explicitly. The Bayesian refinement conjecture is resolved in finite dimensions: for strictly positive likelihoods and at least two coarse packets, exact commutation for every prior is equivalent to blockwise-constant likelihoods up to global scale, and every failure factors through an explicit information-loss packet. Fractional powers are constructed packetwise through the spectral projectors of a positive operator, and the square-root semigroup is recovered by subordination. The fractional-naturality conjecture is also resolved for bounded nonnegative self-adjoint refinement systems: the fractional intertwining defect has an exact resolvent integral, a gap-free Hölder certificate, a sharper common-gap estimate, and a Duhamel semigroup bound. Green kernels, finitedifference Poisson problems, and second-order low-mode convergence provide the PDE realization. A finite-state Feynman–Kac formula is proved by path enumeration. Exact Schur–Feshbach elimination shows that discarded latent packets return as a frequency-dependent self-energy and, in time coordinates, as a memory kernel. Survival and hazard functions are generated by transient semigroups. Nineteen code-generated figures, machine-readable results, the original thirty-seven automated tests, and seventeen added boundary-case checks accompany the text. The saved Lean verification covers all 33 scoped theorem-like results in Appendix A. Finite-dimensional theorems, classical infinite-dimensional extensions, numerical certificates, and open continuum claims are kept separate.

Scope and claim boundary. The chapter proves the stated finite-dimensional semigroup, packet, Markov, Bayesian, conditional-expectation, fractional spectral, Green-kernel, Feynman–Kac, Schur–Feshbach, and survival identities. It also proves quantitative refinement-defect formulas under explicit norm hypotheses, including exact transfer of a bounded generator defect to every fractional power $0 < \alpha < 1$ for bounded nonnegative self-adjoint generators. It does not claim that an arbitrary infinite-dimensional operator generates a semigroup, that every stochastic process admits a finite packet representation, that every posterior family is conjugate, that finite Bayesian sufficiency automatically extends to infinite disintegrations, or that every PDE continuum limit follows from a finite test. Domain, closability, sectoriality, measurability, tightness, and compactness hypotheses remain necessary in their respective infinite-dimensional settings. Classical sources are retained in the separate end-of-series reference ledger rather than reproduced in this chapter.

1 The transfer-semigroup realization

Let X be a finite-dimensional real or complex vector space, let $A \in \mathrm{End}(X)$, and let $s \in X$ be a primitive seed. The basic continuous transfer is

$$T(t) = e^{tA}, \qquad t \geq 0,$$

while a positive operator L is more naturally written

$$H(t) = e^{-tL}.$$

Both satisfy the semigroup law. The sign convention changes with the domain: Markov generators and evolution equations are usually written with e^{tA}. Positive diffusion and elliptic operators are usually written with e^{-tL}.

The common operator chain is

$$\boxed{\text{seed} \longrightarrow \text{transfer} \longrightarrow \text{project} \longrightarrow \text{condition} \longrightarrow \text{refine} \longrightarrow \text{complete} \longrightarrow \text{reconstruct.}}$$

Definition 1.1 (Transfer-semigroup seed system)**.** A finite transfer-semigroup seed system is a tuple

$$\mathfrak{S} = (X, s, A, \{P_\alpha\}_{\alpha \in I}, \tau),$$

where X is the state space, s is the seed, A is a transfer generator, the P_α are pairwise orthogonal idempotents summing to the identity, and τ specifies the topology in which a completion or limit is interpreted.

Theorem 1.2 (Finite semigroup law)**.** *For every matrix A and all $s, t \in \mathbb{R}$,*

$$\boxed{e^{(s+t)A} = e^{sA} e^{tA}.}$$

Moreover, $T(0) = I$ and $T'(t) = AT(t) = T(t)A$.

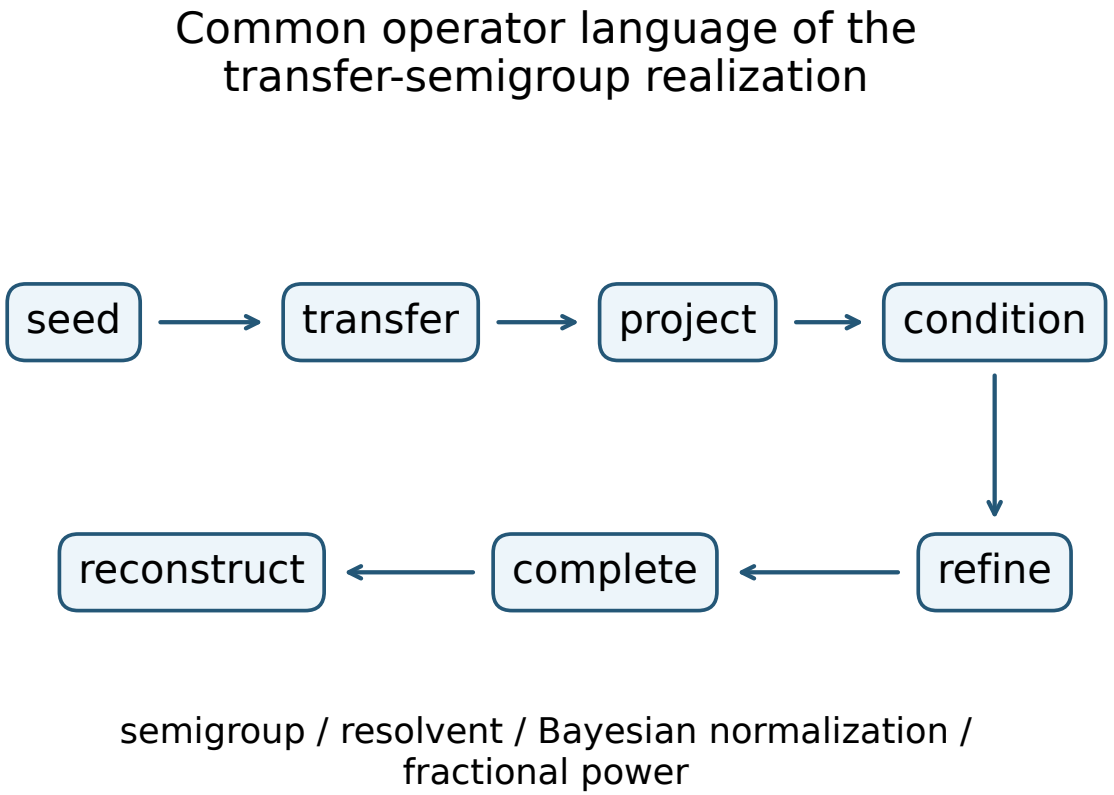


Figure 1. The common operator language used throughout the chapter. Semigroups, resolvents, normalized Bayesian maps, and fractional powers are distinct realizations of the same transfer stage.

Proof. Both exponentials are absolutely convergent matrix power series. Since every power of A commutes with every other power, the Cauchy product gives

$$e^{sA}e^{tA} = \sum_{m,n\geq 0} \frac{s^m t^n}{m!n!} A^{m+n} = \sum_{r\geq 0} \frac{(s+t)^r}{r!} A^r.$$

Termwise differentiation is valid in finite dimension and gives the derivative identities. □

Boundary 1.3 (Infinite-dimensional generators). For an unbounded operator, writing e^{tA} is not a definition unless a semigroup-generation theorem applies. The finite matrix identities in this chapter remain algebraically correct models, but a continuum realization must state the operator domain, closure, growth bound, and topology of convergence.

2 Packet blocks of a semigroup

The central idea

A packet block remembers both the origin and the destination. Composition must sum over every intermediate packet, because an unobserved intermediate coordinate can carry a real contribution. The original formulas below express this bookkeeping exactly.

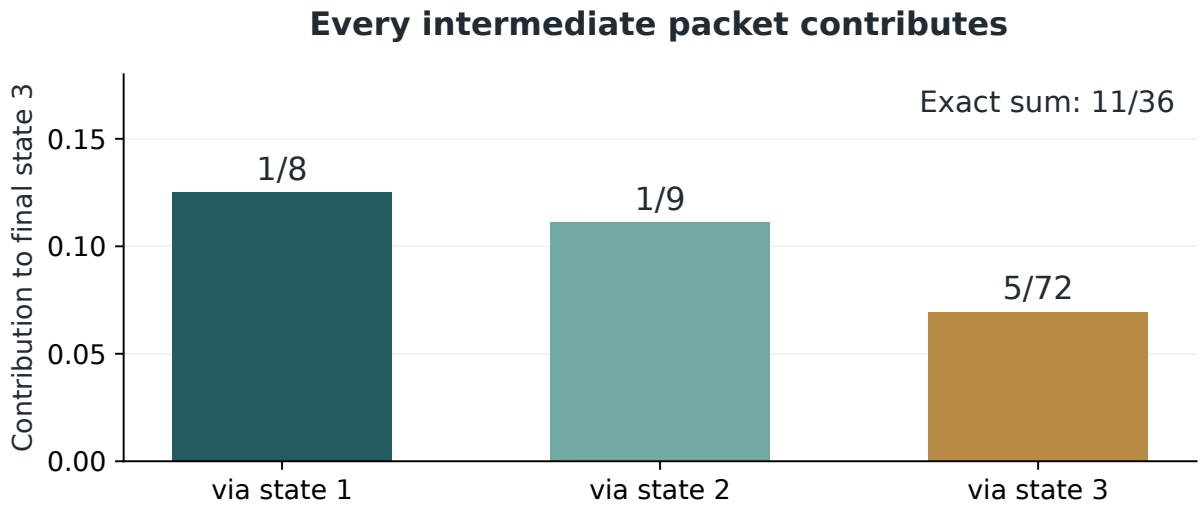


Figure 2. Two rational transfer steps resolved through all three intermediate states. The displayed products are computed exactly before plotting. This finite illustration explains the intermediate-packet sum.

Let

$$P_\alpha P_\beta = \delta_{\alpha\beta} P_\alpha, \qquad \sum_{\alpha\in I} P_\alpha = I.$$

For any operator B, define

$$B_{\beta\alpha} = P_\beta B P_\alpha.$$

Then

$$B = \sum_{\alpha,\beta} B_{\beta\alpha}.$$

For the semigroup, write

$$T_{\beta\alpha}(t) = P_\beta e^{tA} P_\alpha.$$

Theorem 2.1 (Packet semigroup convolution)**.** *For every s, t and every pair of packets α, β,*

$$\boxed{T_{\beta\alpha}(s+t) = \sum_{\gamma \in I} T_{\beta\gamma}(s) T_{\gamma\alpha}(t).}$$

Thus the semigroup law becomes ordinary matrix multiplication in the packet indices.

Proof. Insert the resolution of the identity between the two semigroup factors:

$$\begin{aligned} P_\beta e^{(s+t)A} P_\alpha &= P_\beta e^{sA} \left(\sum_\gamma P_\gamma \right) e^{tA} P_\alpha \\ &= \sum_\gamma (P_\beta e^{sA} P_\gamma)(P_\gamma e^{tA} P_\alpha). \end{aligned}$$

□

Corollary 2.2 (Reducing packets)**.** *If $P_\beta A P_\alpha = 0$ for every $\beta \neq \alpha$, then*

$$P_\beta e^{tA} P_\alpha = 0 \quad (\beta \neq \alpha)$$

for every t. Each packet evolves independently under $P_\alpha A P_\alpha$.

Proof. The hypothesis makes A block diagonal. Every power A^m is block diagonal with respect to the same projectors, and therefore so is the exponential series. □

Remark 2.3 (Path expansion). Expanding the exponential and inserting $I = \sum_\alpha P_\alpha$ between every pair of generators gives

$$P_{\alpha_m} A^m P_{\alpha_0} = \sum_{\alpha_1, \dots, \alpha_{m-1}} P_{\alpha_m} A P_{\alpha_{m-1}} \cdots P_{\alpha_1} A P_{\alpha_0}.$$

Every coefficient is therefore a sum over packet-transfer paths. The numerical library verifies that truncating this path series converges to the ordinary matrix exponential.

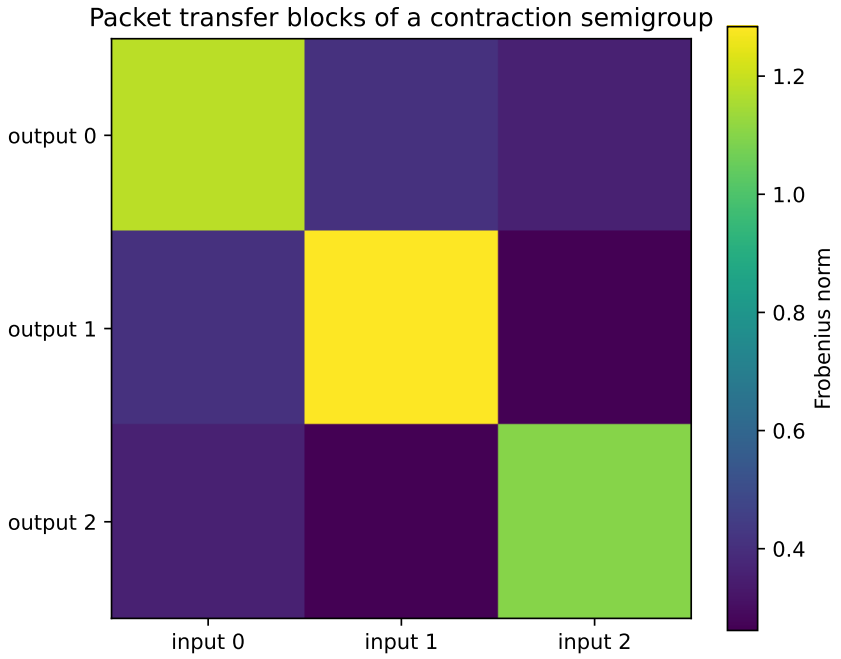


Figure 3. Norms of the packet blocks $P_\beta e^{tA} P_\alpha$. Off-diagonal entries quantify transfer between packets.

3 Refinement defects and the Duhamel identity

How to read the result

Read the generator defect as a mismatch between two declared paths: evolve and then coarsen, or coarsen and then evolve. The Duhamel identity transports that mismatch through time. Its estimate depends on the stated operator norms and growth bounds.

Let X_f and X_c be fine and coarse spaces, with generators A_f and A_c, and let

$$C : X_f \to X_c$$

be a coarsening map. Define the generator-level refinement defect

$$\boxed{\Delta = CA_f - A_c C.}$$

The corresponding semigroup defect is

$$\mathfrak{D}(t) = Ce^{tA_f} - e^{tA_c} C.$$

Theorem 3.1 (Exact Duhamel refinement identity)**.** *For every $t \geq 0$,*

$$\boxed{Ce^{tA_f} - e^{tA_c} C = \int_0^t e^{(t-r)A_c} \Delta e^{rA_f}\, dr.}$$

In particular, $\Delta = 0$ implies exact semigroup intertwining at every time.

Proof. Set

$$F(r) = e^{(t-r)A_c} C e^{rA_f}.$$

Then

$$F'(r) = e^{(t-r)A_c}(CA_f - A_c C)e^{rA_f}.$$

Integrating from 0 to t gives $F(t) - F(0) = Ce^{tA_f} - e^{tA_c}C$ and proves the formula. □

Corollary 3.2 (Norm bound). *Suppose*

$$\|e^{tA_f}\| \le M_f e^{\omega_f t}, \qquad \|e^{tA_c}\| \le M_c e^{\omega_c t}.$$

Then

$$\|\mathfrak{D}(t)\| \le M_f M_c \|\Delta\| \int_0^t e^{\omega_c(t-r)} e^{\omega_f r}\, dr.$$

For contraction semigroups,

$$\boxed{\|\mathfrak{D}(t)\| \le t\,\|\Delta\|\,.}$$

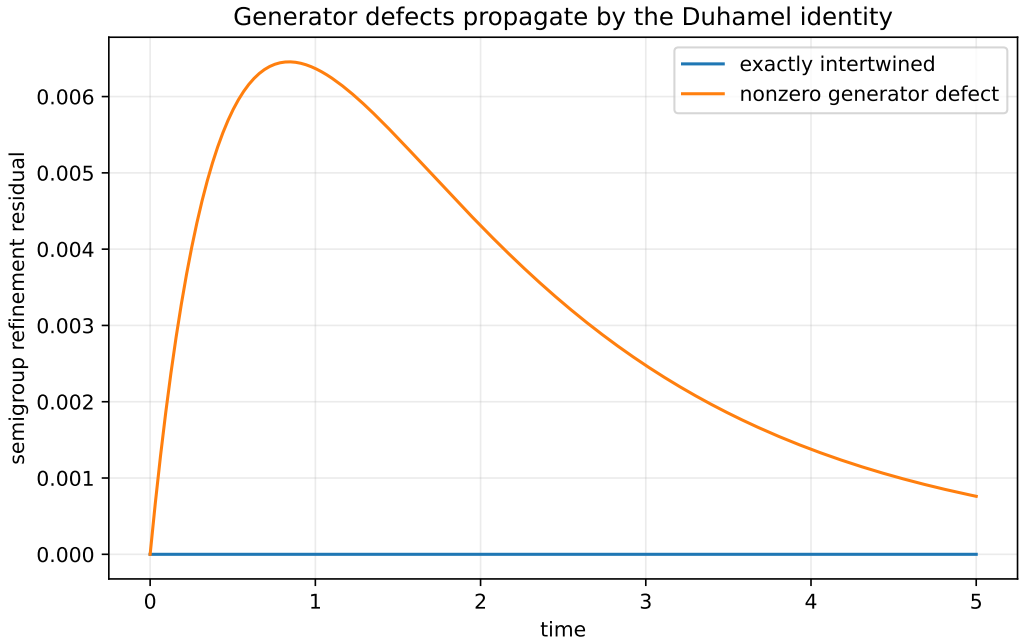


Figure 4. Exact generator intertwining gives a vanishing semigroup defect. A controlled perturbation produces a nonzero defect propagated by the Duhamel identity.

3.1 Resolvent form

For z in the common resolvent set, let

$$R_f(z) = (zI - A_f)^{-1}, \qquad R_c(z) = (zI - A_c)^{-1}.$$

Theorem 3.3 (Resolvent refinement identity)**.**

$$\boxed{CR_f(z) - R_c(z)C = R_c(z)\Delta R_f(z).}$$

Proof. Multiply the left side by $zI - A_c$ on the left and $zI - A_f$ on the right. The result is

$$(zI - A_c)C - C(zI - A_f) = CA_f - A_cC = \Delta.$$

Multiplication by the two resolvents recovers the stated identity. □

4 Markov seeds and stochastic transfer

Use column probability vectors. A finite continuous-time Markov generator Q has nonnegative off-diagonal entries and column sums zero. The stochastic semigroup is

$$K(t) = e^{tQ}.$$

Theorem 4.1 (Finite Markov semigroup)**.** *If Q has nonnegative off-diagonal entries and column sums zero, then e^{tQ} has nonnegative entries and column sums one for every $t \geq 0$.*

Proof. Choose $\lambda \geq \max_i(-Q_{ii})$. Then $B = Q + \lambda I$ has nonnegative entries, so e^{tB} is entrywise nonnegative by its power series. Hence

$$e^{tQ} = e^{-\lambda t}e^{tB}$$

is nonnegative. Since $1^TQ = 0$, differentiation gives

$$\frac{d}{dt}1^Te^{tQ} = 1^TQe^{tQ} = 0,$$

so $1^Te^{tQ} = 1^T$. □

Example 4.2 (Symmetric random walk)**.** Starting from δ_0 and applying the transfer

$$T = \frac{1}{2}(S + S^{-1})$$

for n steps gives

$$\mathbb{P}(X_n = k) = 2^{-n}\binom{n}{(n+k)/2}$$

when $k \equiv n \pmod 2$, and zero otherwise. Hence

$$\mathbb{E}[X_n] = 0, \qquad \mathrm{Var}(X_n) = n.$$

The parity packet is exact: all mass remains in the one residue class permitted by n.

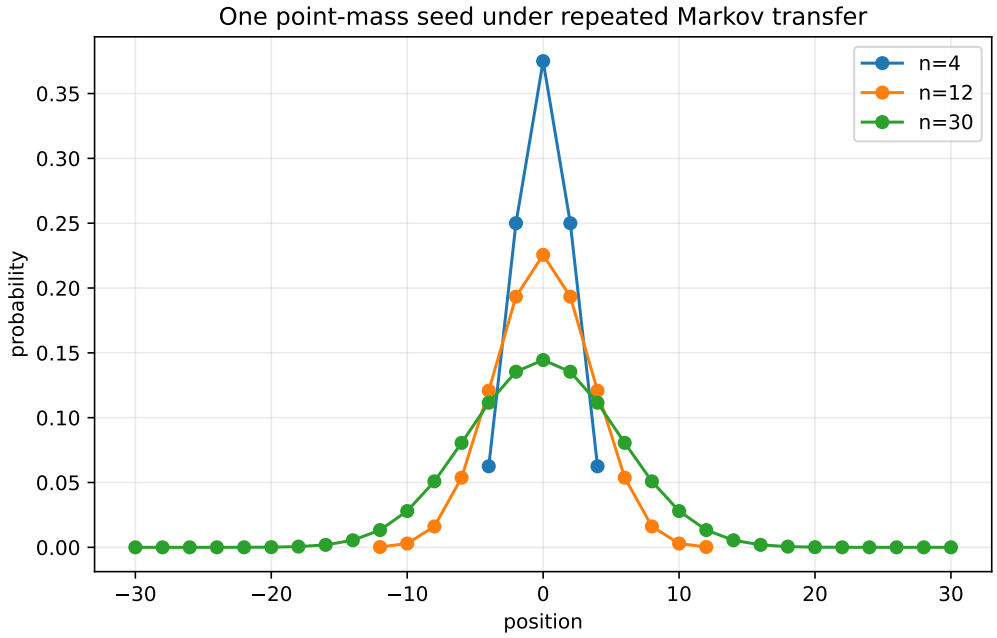


Figure 5. The point mass δ_0 propagated by repeated symmetric random-walk transfer.

5 Markov lumping as exact refinement

Let K be a column-stochastic fine transition matrix, let C aggregate fine states into coarse packets, and let $\bar{K}$ be a coarse transition matrix. Define

$$\Delta = CK - \bar{K}C.$$

Theorem 5.1 (Exact lumpability). *If $\Delta = 0$, then*

$$\boxed{CK^n = \bar{K}^n C}$$

for every $n \geq 0$. Thus fine evolution followed by coarse reconstruction equals coarse evolution from the beginning.

Proof. The result is immediate for $n = 0$. If $CK^n = \bar{K}^n C$, then

$$CK^{n+1} = \bar{K}^n CK = \bar{K}^{n+1} C.$$

□

Theorem 5.2 (Finite-horizon defect formula). *For arbitrary $\Delta = CK - \bar{K}C$,*

$$\boxed{CK^n - \bar{K}^n C = \sum_{j=0}^{n-1} \bar{K}^{n-1-j} \Delta K^j.}$$

If K and $\bar{K}$ are contractions in the chosen norm, then

$$\left\|CK^n - \bar{K}^n C\right\| \le n \left\|\Delta\right\|.$$

Proof. The identity follows by telescoping:

$$\bar{K}^{n-1-j} C K^{j+1} - \bar{K}^{n-j} C K^j = \bar{K}^{n-1-j}(CK - \bar{K}C)K^j.$$

Summing from $j = 0$ to $n - 1$ leaves $CK^n - \bar{K}^n C$. The norm bound follows termwise. □

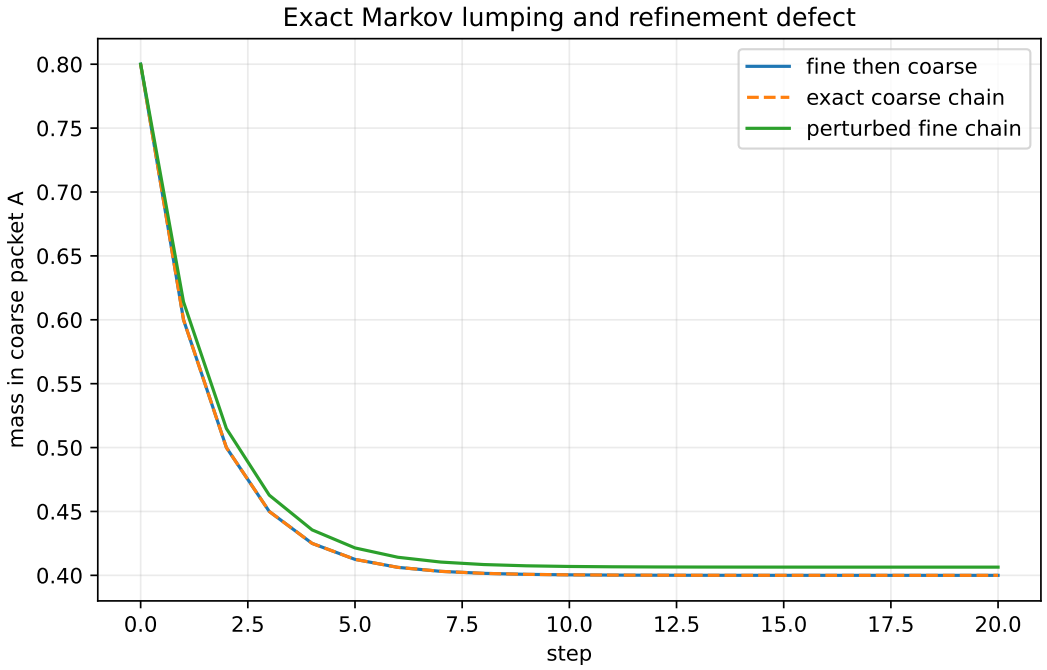


Figure 6. Exact lumping agrees at every horizon. A small generator-level or one-step defect creates a controlled coarse discrepancy.

6 Conditional expectation as a stochastic projector

A route through this section

Conditional expectation is a projector in a weighted Hilbert space. Nested partitions therefore split information into orthogonal increments. Keep the underlying measure fixed while reading the projector identities. A change of measure is a different operation.

Let $\Omega = \{1, \dots, n\}$ carry strictly positive probabilities π_i, and let $\mathcal{Q} = \{B_1, \dots, B_m\}$ be a partition. Define

$$(P_{\mathcal{Q}} f)_i = \frac{\sum_{j \in B(i)} \pi_j f_j}{\sum_{j \in B(i)} \pi_j},$$

where $B(i)$ is the block containing i.

Theorem 6.1 (Finite conditional-expectation projector)**.** *With the weighted inner product*

$$\langle f, g\rangle_\pi = \sum_i \pi_i f_i \overline{g_i},$$

the operator $P_{\mathcal{Q}}$ satisfies

$$P_{\mathcal{Q}}^2 = P_{\mathcal{Q}}, \qquad \langle P_{\mathcal{Q}} f, g\rangle_\pi = \langle f, P_{\mathcal{Q}} g\rangle_\pi.$$

Its range is the space of functions constant on each partition block.

Proof. After one application, the result is constant on every block, so applying the same block average again changes nothing. Weighted self-adjointness follows by expanding both inner products block by block. Both equal

$$\sum_{B\in\mathcal{Q}} \frac{1}{\pi(B)} \left(\sum_{i\in B} \pi_i f_i\right) \overline{\left(\sum_{j\in B} \pi_j g_j\right)}.$$

□

Theorem 6.2 (Tower law and martingale packets)**.** *If $\mathcal{Q}_c$ is coarser than $\mathcal{Q}_f$, then*

$$\boxed{P_c P_f = P_f P_c = P_c.}$$

For a nested tower $P_0 \le P_1 \le \cdots \le P_r$, define

$$D_0 = P_0, \qquad D_j = P_j - P_{j-1}.$$

Then

$$D_i D_j = \delta_{ij} D_j, \qquad P_r = \sum_{j=0}^{r} D_j.$$

Proof. A coarse-block average of a fine-block average is the coarse-block average, and a fine-block average of a coarse-constant function leaves it unchanged. The difference identities follow from $P_i P_j = P_{\min(i,j)}$. □

Classical Input 6.3 (Gaussian chaos realization). For the standard Gaussian measure, the normalized probabilists' Hermite polynomials form orthonormal chaos packets. The Ornstein–Uhlenbeck semigroup acts diagonally:

$$T_t H_n = e^{-nt} H_n.$$

Thus conditional expectation, martingale differences, and Gaussian chaos are all projector realizations of stochastic refinement.

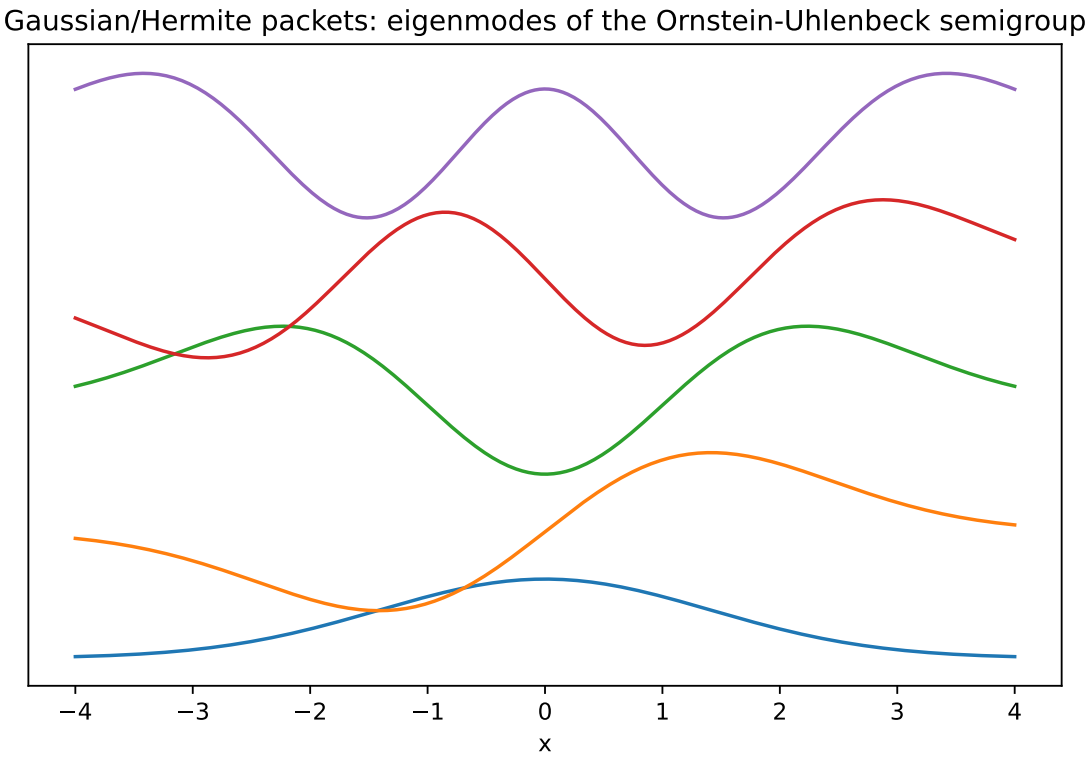


Figure 7. Hermite–Gaussian packets, which diagonalize the Ornstein–Uhlenbeck semigroup. The numerical study checks their Gaussian orthogonality through quadrature. The five displayed modes run from n = 0 at the bottom to n = 4 at the top. Vertical offsets separate the curves.

7 Bayesian conditioning as a normalized nonlinear operator

Let $p = (p_i)$ be a probability vector and let $\lambda = (\lambda_i)$ be a nonnegative likelihood with positive evidence

$$Z_\lambda(p) = \sum_i \lambda_i p_i.$$

Define

$$\boxed{\mathcal{B}_\lambda(p) = \frac{\lambda \odot p}{Z_\lambda(p)}.}$$

This map is nonlinear solely because of its scalar reconstruction factor.

Theorem 7.1 (Independent-likelihood composition)**.** *For nonnegative likelihoods λ and μ, assume $Z_\lambda(p) > 0$ and $Z_{\lambda\odot\mu}(p) > 0$. Equivalently, both successive updates have positive evidence. Then*

$$\boxed{\mathcal{B}_\mu(\mathcal{B}_\lambda(p)) = \mathcal{B}_{\lambda\odot\mu}(p).}$$

Proof. The first normalization appears in both numerator and denominator of the second update and cancels:

$$\frac{\mu_i(\lambda_i p_i/Z_\lambda)}{\sum_j \mu_j\lambda_j p_j/Z_\lambda} = \frac{\mu_i\lambda_i p_i}{\sum_j \mu_j\lambda_j p_j}.$$

□

7.1 Packet evidence and reconstruction

Let the state indices be partitioned into packets I_α. Define

$$Z_\alpha = \sum_{i\in I_\alpha} \lambda_i p_i, \qquad m_\alpha = \frac{Z_\alpha}{\sum_\beta Z_\beta}, \qquad q_{\alpha,i} = \begin{cases} \lambda_i p_i/Z_\alpha, & Z_\alpha > 0,\ i \in I_\alpha, \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 7.2 (Bayesian packet reconstruction)**.**

$$\boxed{\mathcal{B}_\lambda(p) = \sum_\alpha m_\alpha q_\alpha.}$$

Only packets with positive evidence carry a conditional probability vector. A zero-evidence packet has zero posterior mass and contributes the zero vector. The nonlinear coupling between packets occurs only through the total evidence used to normalize the packet evidences.

Proof. If $Z_\alpha = 0$, nonnegativity implies $\lambda_i p_i = 0$ for every $i \in I_\alpha$, so the reconstruction identity holds with both sides zero. For $Z_\alpha > 0$ and $i \in I_\alpha$,

$$m_\alpha q_{\alpha,i} = \frac{Z_\alpha}{Z}\frac{\lambda_i p_i}{Z_\alpha} = \frac{\lambda_i p_i}{Z}.$$

The packet supports are disjoint, so summing reconstructs the full posterior. □

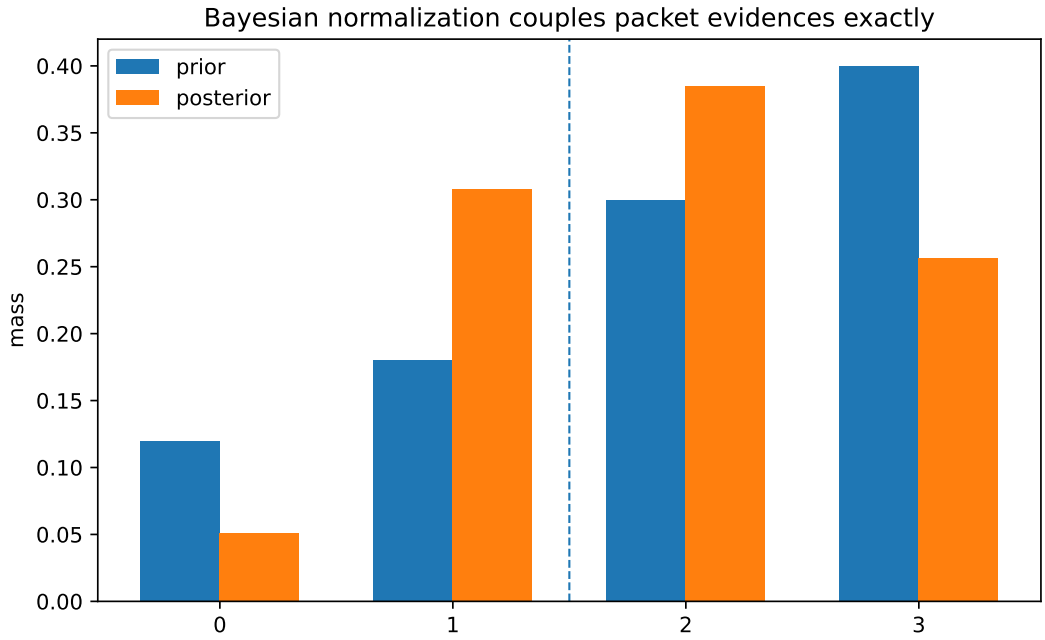


Figure 8. A finite prior and posterior split into two state packets. Evidence changes both the local packet shape and the global packet mass.

Theorem 7.3 (Likelihood perturbation bound). *Let p be fixed and let $\lambda, \mu \geq 0$ have positive evidences Z_λ, Z_μ. Then*

$$\boxed{\left\|\mathcal{B}_\lambda(p) - \mathcal{B}_\mu(p)\right\|_1 \leq \frac{2\sum_i |\lambda_i - \mu_i| p_i}{\min(Z_\lambda, Z_\mu)}.}$$

Proof. Add and subtract $\mu \odot p / Z_\lambda$ and use the triangle inequality:

$$\left\|\frac{\lambda \odot p}{Z_\lambda} - \frac{\mu \odot p}{Z_\mu}\right\|_1 \leq \frac{\|(\lambda - \mu) \odot p\|_1}{Z_\lambda} + \|\mu \odot p\|_1 \left|\frac{1}{Z_\lambda} - \frac{1}{Z_\mu}\right|.$$

The second term equals $|Z_\lambda - Z_\mu| / Z_\lambda$ and is bounded by the same weighted L^1 difference divided by Z_λ. Interchanging λ and μ gives the minimum-evidence form. □

8 Exact Bayesian refinement and information-loss packets

How to read the result

The all-prior criterion here tests coarse posterior masses for every prior and assumes at least two coarse blocks. Part II uses a stronger convention that also preserves each conditional distribution inside a block. These are different contracts, so the one-block boundary must be read with its own definition.

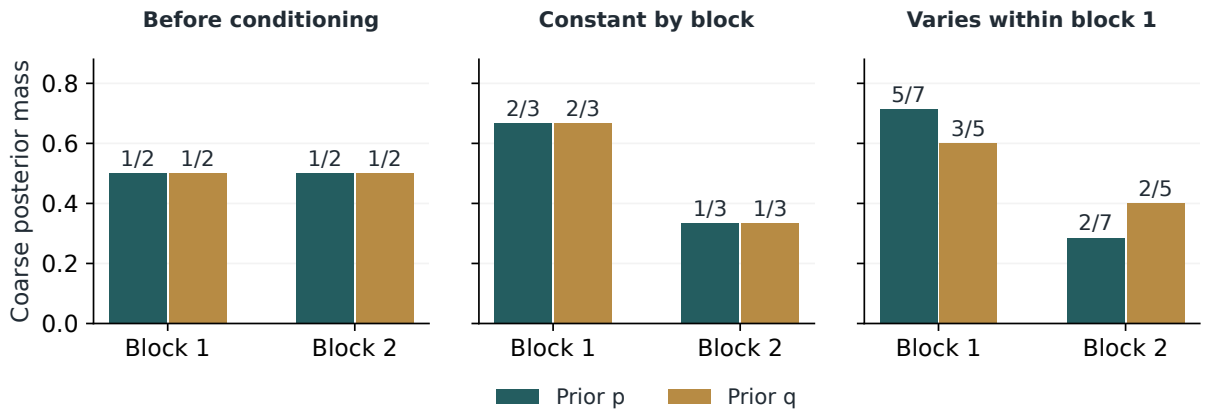


Figure 9. Two rational priors with the same two coarse masses. A blockwise constant likelihood gives the same coarse update, while a likelihood varying within a block separates the posteriors. The exact fractions are calculated before plotting.

Let the fine state set I be partitioned into nonempty coarse packets I_a, $a \in A$. The coarsening map on probability vectors is

$$(C_\pi p)_a = P_a = \sum_{i \in I_a} p_i.$$

For a fine likelihood $\lambda = (\lambda_i)_{i \in I}$ and a prior p, define the effective coarse likelihood

$$r_a(p) = \frac{\sum_{i \in I_a} \lambda_i p_i}{P_a} \qquad (P_a > 0).$$

Its value on a zero-mass packet is irrelevant and may be set to zero.

Theorem 8.1 (Exact effective-likelihood factorization). *Whenever the fine evidence is positive,*

$$\boxed{C_\pi \mathcal{B}_\lambda(p) = \mathcal{B}_{r(p)}(C_\pi p).}$$

Thus the only fine information needed for the coarse posterior, beyond the coarse prior masses, is the vector of conditional packet evidences $r(p)$.

Proof. For $P = C_\pi p$ and $Z = \sum_i \lambda_i p_i = \sum_a P_a r_a(p)$,

$$\left(C_\pi \mathcal{B}_\lambda(p)\right)_a = \frac{\sum_{i \in I_a} \lambda_i p_i}{Z} = \frac{P_a r_a(p)}{\sum_b P_b r_b(p)} = \left(\mathcal{B}_{r(p)}(P)\right)_a.$$

□

The next result gives the exact necessary and sufficient condition for a coarse likelihood to work for every prior.

Theorem 8.2 (Bayesian refinement criterion)**.** *Assume that the partition contains at least two coarse packets and that the fine and coarse likelihoods are strictly positive. The identity*

$$\boxed{C_\pi \mathcal{B}_\lambda(p) = \mathcal{B}_{\bar\lambda}(C_\pi p)}$$

holds for every fine prior p if and only if there exists one scalar $c > 0$ such that

$$\boxed{\lambda_i = c\,\bar\lambda_a \qquad (i \in I_a).}$$

Equivalently, the likelihood is constant on each coarse packet, with the coarse values determined up to the globally invisible Bayesian scaling c.

Proof. If $\lambda_i = c\bar\lambda_a$ on I_a, then $P_a r_a(p) = c\bar\lambda_a P_a$ for every prior, including zero-mass packets, and the preceding theorem gives the desired identity because Bayesian normalization removes the common scalar c.

Conversely, fix two fine states $i, i' \in I_a$ and one state $j \in I_b$ in another coarse packet. Vary the distribution of a fixed coarse mass inside I_a between i and i' while leaving every coarse packet mass unchanged. The right side depends only on those coarse masses and is therefore unchanged. The left-side posterior mass of packet a depends on the conditional average of λ_i and $\lambda_{i'}$. It can be independent of that variation only if $\lambda_i = \lambda_{i'}$. Hence λ is constant on every packet. Write the value on I_a as λ_a.

Now use priors supported on one state of packet a and one state of packet b. Equality of the two coarse posterior odds gives

$$\frac{\lambda_a}{\lambda_b} = \frac{\bar\lambda_a}{\bar\lambda_b}.$$

This holds for every pair of coarse packets, so $\lambda_a = c\bar\lambda_a$ for one common $c > 0$. □

Remark 8.3 (The one-packet boundary). If the coarse partition has only one packet, every posterior coarsens to the scalar 1, so commutation is automatic and no within-packet likelihood information can be identified. The nontrivial criterion begins with at least two coarse packets.

When sufficiency fails, the defect still has an exact finite factorization. Fix a declared coarse likelihood $\bar\lambda$, put

$$P = C_\pi p, \qquad \delta(p) = r(p) - \bar\lambda, \qquad \bar b = \mathcal{B}_{\bar\lambda}(P), \qquad Z_r = \langle r(p), P\rangle.$$

Theorem 8.4 (Bayesian information-loss packet)**.** *Assume nonnegative fine and declared coarse likelihoods, with both $Z_r > 0$ and $Z_{\bar\lambda} > 0$. The coarsening defect is*

$$\boxed{C_\pi \mathcal{B}_\lambda(p) - \mathcal{B}_{\bar\lambda}(P) = \frac{1}{Z_r}\left(\operatorname{diag}(P) - \bar b P^{\mathsf T}\right)\delta(p).}$$

The factor matrix annihilates $\bar{\lambda}$. This records the common-scale null direction at the declared coarse likelihood. The full exact defect remains homogeneous under simultaneous likelihood rescaling because the numerator and evidence denominator scale together. Moreover,

$$\boxed{\left\|C_\pi \mathcal{B}_\lambda(p) - \mathcal{B}_{\bar{\lambda}}(P)\right\|_1 \le \frac{2\sum_a P_a|\delta_a(p)|}{\min\{Z_r, Z_{\bar{\lambda}}\}}}$$

and

$$\sum_a P_a|\delta_a(p)| \le \sum_i p_i \left|\lambda_i - \bar{\lambda}_{\pi(i)}\right|.$$

Proof. The effective-likelihood theorem reduces the defect to

$$\mathcal{B}_{\bar{\lambda}+\delta}(P) - \mathcal{B}_{\bar{\lambda}}(P).$$

Since $Z_r = Z_{\bar{\lambda}} + P^\mathsf{T}\delta$, direct subtraction gives

$$\frac{1}{Z_r}\left(\operatorname{diag}(P) - \bar{b}P^\mathsf{T}\right)\delta.$$

Also,

$$\left(\operatorname{diag}(P) - \bar{b}P^\mathsf{T}\right)\bar{\lambda} = P \odot \bar{\lambda} - \bar{b}Z_{\bar{\lambda}} = 0.$$

The first norm bound is the likelihood perturbation theorem applied on the coarse simplex. For a zero-mass packet, both sides of the following estimate vanish. For $P_a > 0$,

$$P_a|\delta_a| = P_a\left|\sum_{i\in I_a}\frac{p_i}{P_a}(\lambda_i - \bar{\lambda}_a)\right| \le \sum_{i\in I_a} p_i|\lambda_i - \bar{\lambda}_a|,$$

and summing over a proves the second inequality. □

Corollary 8.5 (Nested Bayesian packet tower)**.** *Consider a nested finite packet tower with strictly positive likelihoods at every level. At each adjacent coarsening with at least two coarse packets, commutation for every prior is equivalent to the fine likelihood being blockwise proportional to the declared coarse likelihood, with one common positive proportionality factor. An adjacent coarsening to a single packet commutes automatically and imposes no such condition. Under these conditions, all composite coarsening squares commute.*

Proof. At each adjacent coarsening with at least two coarse packets, apply the refinement criterion. For a single coarse packet, both normalized posteriors equal 1. Composition of these commuting squares gives the full tower statement. □

Corollary 8.6 (Canonical coarsest likelihood-resolving partition)**.** *For a finite family of likelihoods $\{\lambda^{(y)}\}_{y\in Y}$, define*

$$i \sim j \iff \lambda_i^{(y)} = \lambda_j^{(y)} \text{ for every } y \in Y.$$

The equivalence classes form the unique coarsest likelihood-resolving partition: every likelihood is constant on each packet. Every update therefore closes on this partition whenever its evidence is positive. If all likelihoods are strictly positive and there are at least two profile classes, this is also the unique coarsest nontrivial partition on

which every update closes for every prior. The one-packet partition always closes, independently of the profiles, and is excluded from this last assertion.

Proof. Equality of likelihood profiles is an equivalence relation. Its classes are likelihood-resolving by definition, and every likelihood-resolving partition refines them. Exact closure follows by summing the weighted masses in each fiber, including zero-mass fibers. Under strict positivity, the refinement criterion implies that every nontrivial all-prior closed partition is likelihood-resolving. When every profile is identical, all partitions close. There need not be a unique coarsest nontrivial partition. □

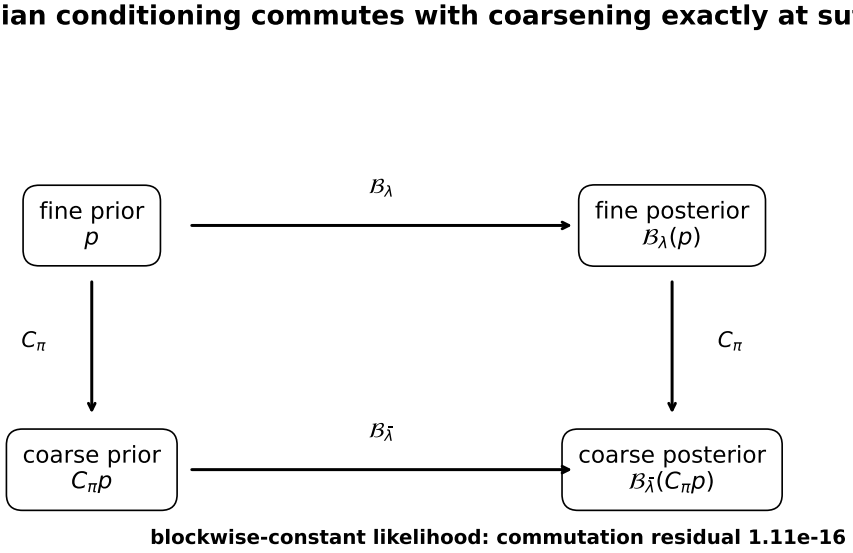


Figure 10. For strictly positive likelihoods and at least two coarse packets, the Bayesian refinement square commutes for every prior exactly when the likelihood is constant on each coarsening fiber, up to one global scale.

Numerical Certificate 8.7 (Bayesian refinement). For the finite test partition $\{0,1\} \sqcup \{2,3\}$, a blockwise-constant likelihood gave commutation residual 1.11×10^{-16}. A nonconstant likelihood produced an exact defect-factorization residual 7.29×10^{-17}. Its measured L^1 defect was 4.8787×10^{-2}, below the certified bound 1.1016×10^{-1}. Two priors with identical coarse masses but different hidden within-packet shapes produced coarse posteriors differing by 1.2400×10^{-1}, which is a direct necessity witness.

Boundary 8.8 (Finite exact theorem). The conjectured Bayesian refinement law is therefore resolved for finite packet systems and arbitrary priors. Infinite-dimensional disintegration, posterior consistency, and stability of data-dependent partitions require their own measure-theoretic and asymptotic hypotheses.

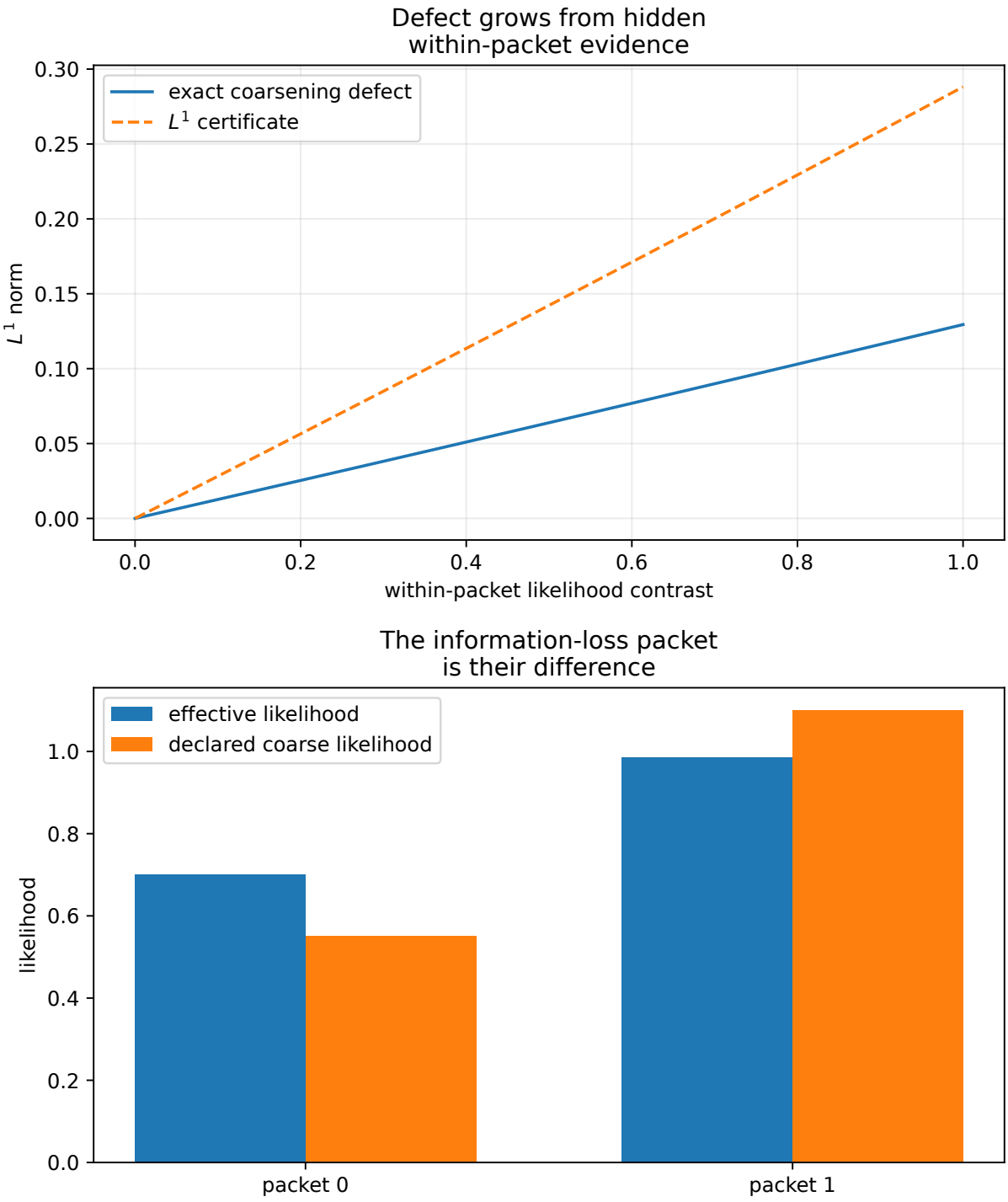


Figure 11. When packet sufficiency fails, the exact defect factors through the difference between the conditional effective likelihood and the declared coarse likelihood.

9 Bayesian flows, conjugacy, and hidden states

Fix a score vector g. The likelihood flow is

$$\lambda_i(t) = e^{tg_i},$$

and the Bayesian flow is

$$\Phi_t(p)_i = \frac{e^{tg_i}p_i}{\sum_j e^{tg_j}p_j}.$$

Theorem 9.1 (Bayesian nonlinear semigroup)**.**

$$\boxed{\Phi_{s+t} = \Phi_s \circ \Phi_t.}$$

Its generator is the centered multiplicative field

$$\boxed{\frac{d}{dt}p_i(t) = (g_i - \langle g, p(t)\rangle)p_i(t).}$$

Proof. The semigroup law is the likelihood-composition theorem with $e^{sg}e^{tg} = e^{(s+t)g}$. Differentiating the normalized exponential formula gives

$$\dot{p}_i = g_i p_i - p_i \frac{\sum_j g_j e^{tg_j}p_j(0)}{\sum_j e^{tg_j}p_j(0)}.$$

The quotient is $\langle g, p(t)\rangle$. □

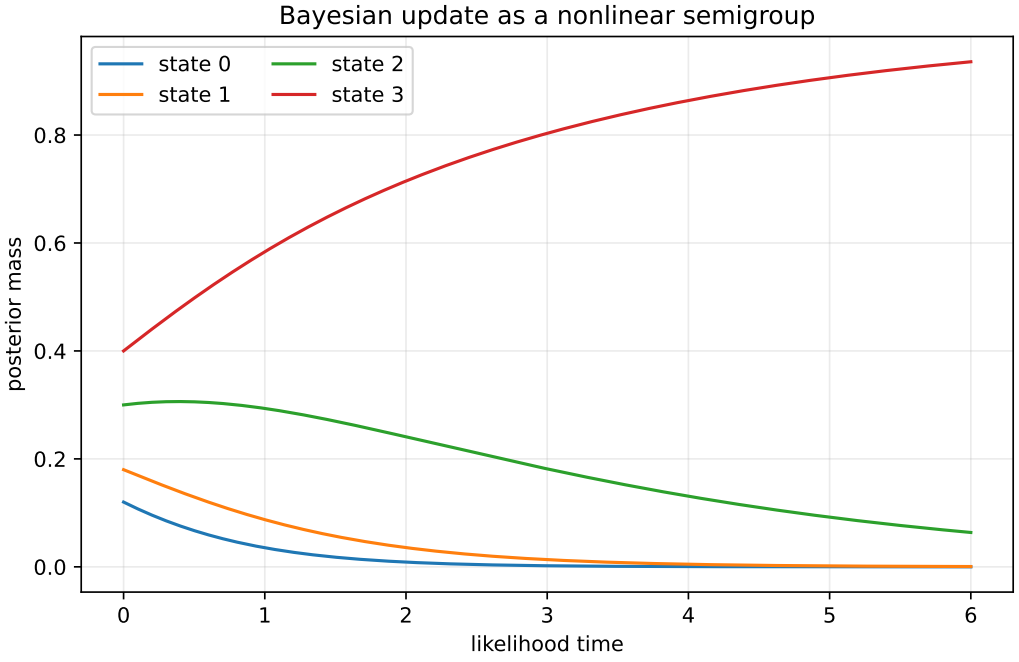


Figure 12. Continuous likelihood accumulation produces a nonlinear semigroup on the probability simplex.

Example 9.2 (Beta–Binomial conjugate packet)**.** A Beta(a, b) prior and s successes in n trials produce

$$\text{Beta}(a + s, b + n - s).$$

The prior exponents are additive packets: the likelihood increments them by $(s, n-s)$. In the numerical example,

$$(a, b) = (2, 3), \quad (s, n) = (7, 10),$$

so the posterior is Beta$(9, 6)$ with mean 0.6.

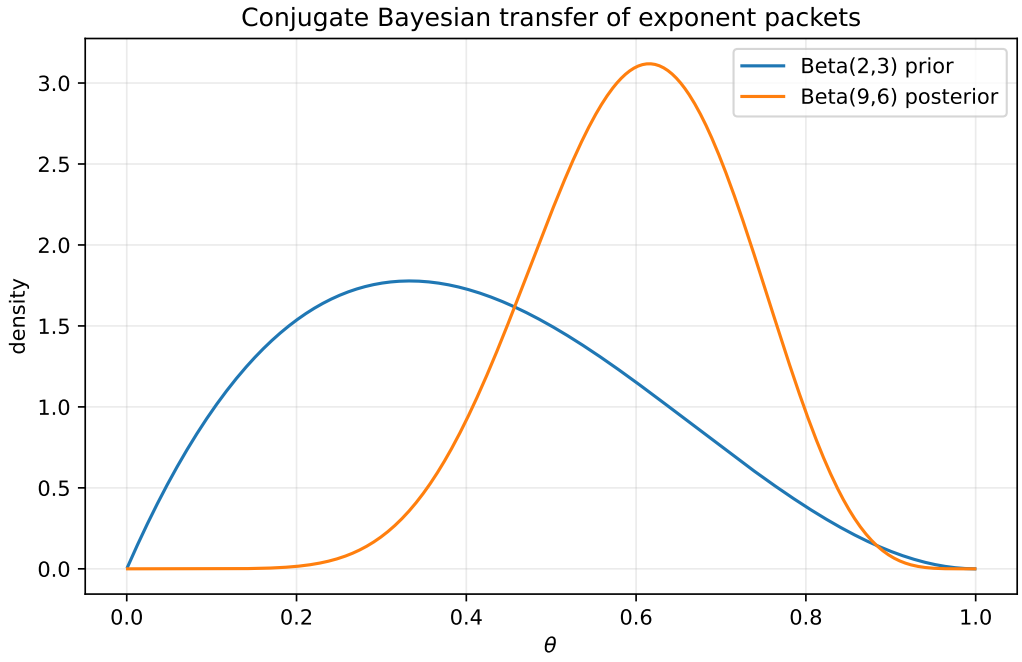


Figure 13. A conjugate Bayesian update viewed as exact transfer of exponent packets.

9.1 Hidden Markov filtering

Let K be column stochastic and let $\lambda^{(r)}$ be the likelihood associated with the rth observation. The filter recursion is

$$\hat{p}_r = K p_{r-1}, \qquad p_r = \mathcal{B}_{\lambda^{(r)}}(\hat{p}_r).$$

Proposition 9.3 (Prediction-update reconstruction)**.** *Every hidden Markov filtering step is the composition of a linear Markov transfer and a normalized nonlinear likelihood transfer. Assume a probability initial vector, a column-stochastic transition matrix, and nonnegative likelihoods with positive evidence at every step. Then every p_r is a probability vector.*

Proof. Column stochasticity gives $\hat{p}_r \geq 0$ and $1^T \hat{p}_r = 1$. Bayesian normalization then gives $p_r \geq 0$ and $1^T p_r = 1$. □

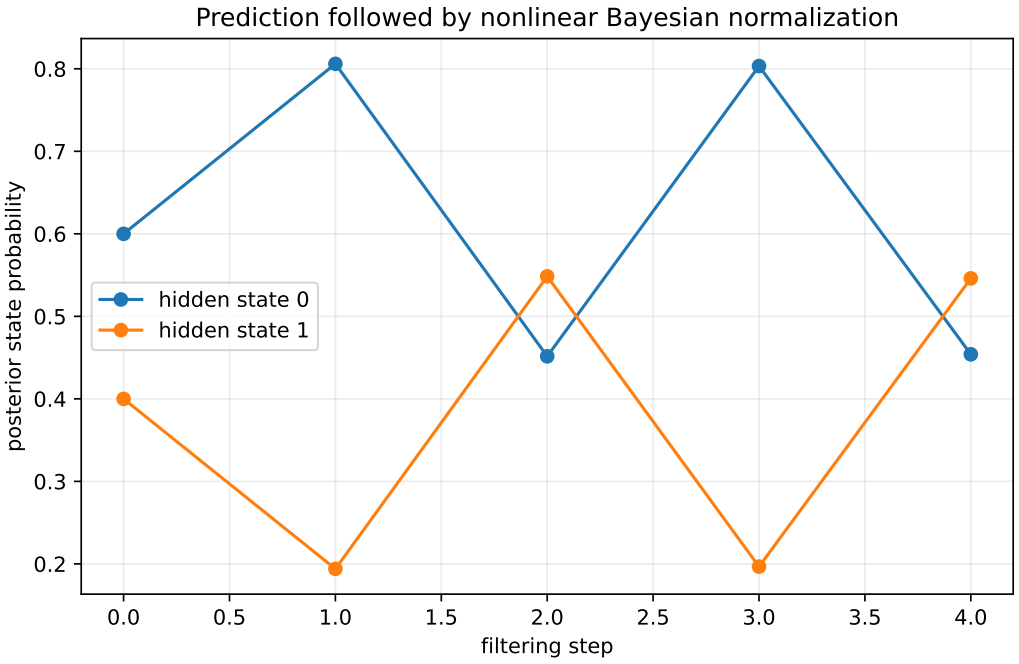


Figure 14. A hidden Markov filter alternates a linear prediction semigroup with nonlinear Bayesian normalization.

10 Fractional powers as spectral packet transfer

A route through this section

First identify the positive operator and its spectral packets. Then distinguish an exact spectral identity from a bound for a refinement mismatch. The gap-free certificate and the common-gap estimate have different hypotheses and different strengths.

Let $L = L^* \geq 0$ on a finite-dimensional Hilbert space. Its spectral resolution is

$$L = \sum_j \lambda_j P_j.$$

For $\alpha > 0$, define

$$\boxed{L^\alpha = \sum_j \lambda_j^\alpha P_j.}$$

The associated fractional semigroup is

$$T_\alpha(t) = e^{-tL^\alpha} = \sum_j e^{-t\lambda_j^\alpha} P_j.$$

Theorem 10.1 (Fractional packet calculus). *For $\alpha, \beta > 0$,*

$$L^\alpha L^\beta = L^{\alpha+\beta},$$

and for $s, t \geq 0$,

$$\boxed{T_\alpha(s)T_\alpha(t) = T_\alpha(s+t).}$$

Every spectral packet P_j reduces L^α and $T_\alpha(t)$.

Proof. All statements hold coefficientwise on the mutually orthogonal spectral projectors. For example,

$$L^\alpha L^\beta = \sum_{j,k} \lambda_j^\alpha \lambda_k^\beta P_j P_k = \sum_j \lambda_j^{\alpha+\beta} P_j.$$

□

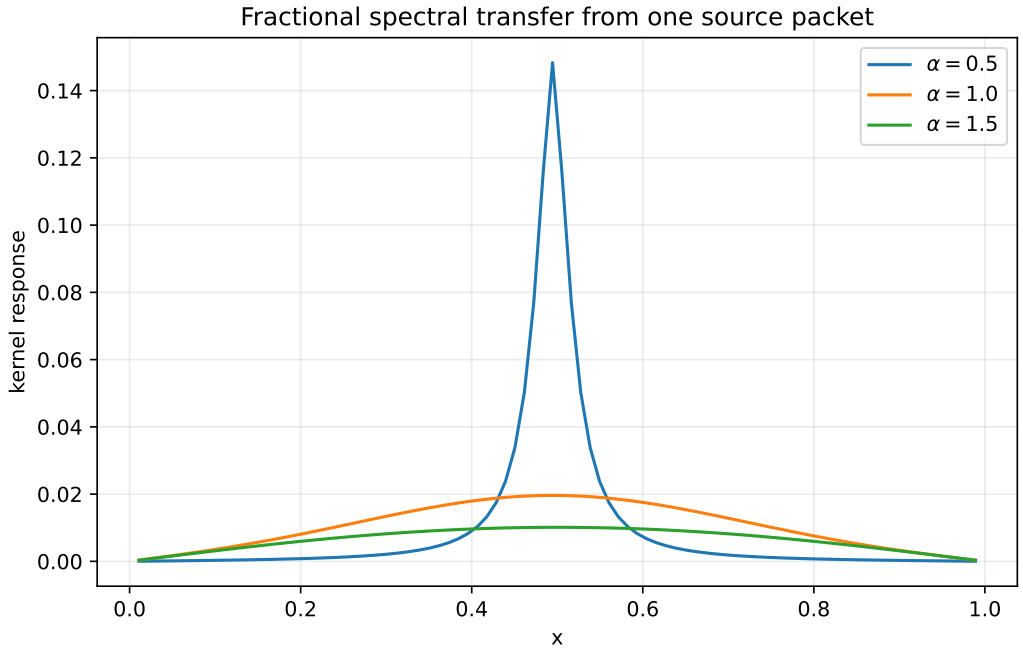


Figure 15. Different spectral exponents produce different transfer profiles from the same point seed.

10.1 Square-root subordination

Theorem 10.2 (Poisson subordination). *For $t > 0$ and $L \geq 0$,*

$$\boxed{e^{-t\sqrt{L}} = \frac{t}{2\sqrt{\pi}} \int_0^\infty s^{-3/2} e^{-t^2/(4s)} e^{-sL}\, ds.}$$

The integral converges in operator norm in finite dimension.

Proof. By the spectral theorem it suffices to prove the scalar identity

$$e^{-t\sqrt{\lambda}} = \frac{t}{2\sqrt{\pi}} \int_0^\infty s^{-3/2} e^{-t^2/(4s) - \lambda s}\, ds$$

for $\lambda \geq 0$. This is the classical Gaussian integral after the substitution $u = t/(2\sqrt{s})$. Applying the identity to every eigenvalue and recombining the spectral projectors proves the operator formula. □

10.2 Fractional naturality under a bounded refinement defect

Let $A \geq 0$ act on a fine Hilbert space, let $B \geq 0$ act on a coarse Hilbert space, and let C be a bounded coarsening map. Set

$$D = CA - BC, \qquad D_\alpha = CA^\alpha - B^\alpha C, \qquad 0 < \alpha < 1.$$

Theorem 10.3 (Fractional intertwining theorem). *For bounded nonnegative self-adjoint A and B,*

$$\boxed{D_\alpha = \frac{\sin(\pi\alpha)}{\pi} \int_0^\infty t^\alpha (tI + B)^{-1} D (tI + A)^{-1}\, dt.}$$

The improper integral converges in operator norm. Moreover,

$$\boxed{\|D_\alpha\| \leq K_\alpha \|C\|^{1-\alpha} \|D\|^\alpha, \qquad K_\alpha = \frac{2^{1-\alpha}\sin(\pi\alpha)}{\pi\alpha(1-\alpha)}.}$$

If $A, B \geq mI$ for some $m > 0$, then the stronger linear estimate holds:

$$\boxed{\|D_\alpha\| \leq \alpha m^{\alpha-1} \|D\|.}$$

In particular, exact generator intertwining $CA = BC$ implies exact fractional intertwining $CA^\alpha = B^\alpha C$.

Proof. For $0 < \alpha < 1$, the Balakrishnan formula gives

$$A^\alpha = \frac{\sin(\pi\alpha)}{\pi} \int_0^\infty t^{\alpha-1} A (tI + A)^{-1}\, dt,$$

and likewise for B. Using

$$A(tI + A)^{-1} = I - t(tI + A)^{-1}$$

and the rectangular resolvent identity

$$(tI + B)^{-1}C - C(tI + A)^{-1} = (tI + B)^{-1}(CA - BC)(tI + A)^{-1},$$

the difference of the two Balakrishnan integrals becomes the displayed resolvent-defect formula.

For small t,

$$\left\| CA(tI + A)^{-1} - B(tI + B)^{-1}C \right\| \leq 2\|C\|,$$

because the two positive resolvent factors are contractions. For large t, the resolvent identity gives the bound $\|D\|/t$. Splitting the integral at

$$r = \frac{\|D\|}{2\|C\|}$$

when C and D are nonzero yields

$$\begin{aligned}\|D_\alpha\| &\le \frac{\sin(\pi\alpha)}{\pi}\left(2\|C\|\int_0^r t^{\alpha-1}\,dt + \|D\|\int_r^\infty t^{\alpha-2}\,dt\right)\\ &= K_\alpha \|C\|^{1-\alpha}\|D\|^\alpha.\end{aligned}$$

The zero cases are immediate.

If $A, B \ge mI$, then both resolvents have norm at most $(t+m)^{-1}$, so

$$\|D_\alpha\| \le \frac{\sin(\pi\alpha)}{\pi}\|D\|\int_0^\infty \frac{t^\alpha}{(t+m)^2}\,dt.$$

After $t = mu$, the beta integral equals

$$m^{\alpha-1}\frac{\alpha\pi}{\sin(\pi\alpha)},$$

which gives the linear estimate. □

Corollary 10.4 (Fractional semigroup refinement)**.** *For $s \ge 0$,*

$$\boxed{Ce^{-sA^\alpha} - e^{-sB^\alpha}C = -\int_0^s e^{-(s-r)B^\alpha} D_\alpha e^{-rA^\alpha}\,dr.}$$

Hence

$$\left\|Ce^{-sA^\alpha} - e^{-sB^\alpha}C\right\| \le s\|D_\alpha\|.$$

Under the common gap $A, B \ge mI$,

$$\boxed{\left\|Ce^{-sA^\alpha} - e^{-sB^\alpha}C\right\| \le se^{-sm^\alpha}\alpha m^{\alpha-1}\|D\|.}$$

Proof. Differentiate

$$r \longmapsto e^{-(s-r)B^\alpha}Ce^{-rA^\alpha}$$

and integrate from 0 to s. Positive fractional semigroups are contractions. Under the gap, the product of their two norm bounds is e^{-sm^α}. □

Corollary 10.5 (Fractional refinement towers)**.** *Let $D_\ell = C_\ell A_{\ell+1} - A_\ell C_\ell$ in a bounded nonnegative self-adjoint tower, with $\sup_\ell \|C_\ell\| \le M$. If*

$$\sum_\ell \|D_\ell\|^\alpha < \infty,$$

then the fractional defects

$$C_\ell A_{\ell+1}^\alpha - A_\ell^\alpha C_\ell$$

have summable operator norms. If the tower has a uniform gap $m > 0$, it is enough that $\sum_\ell \|D_\ell\| < \infty$. On every bounded time interval, the series of corresponding fractional-semigroup defect norms converges uniformly. The maps may have different domain and codomain spaces. No operator sum across unrelated spaces is asserted.

Boundary 10.6 (Unbounded and nonnormal generators). The theorem resolves fractional naturality for bounded nonnegative self-adjoint refinement systems and gives the exact estimate needed for finite discretization towers. For unbounded positive generators, the same resolvent argument requires a declared common-domain condition and a bounded extension of $CA - BC$. Sectorial or nonnormal

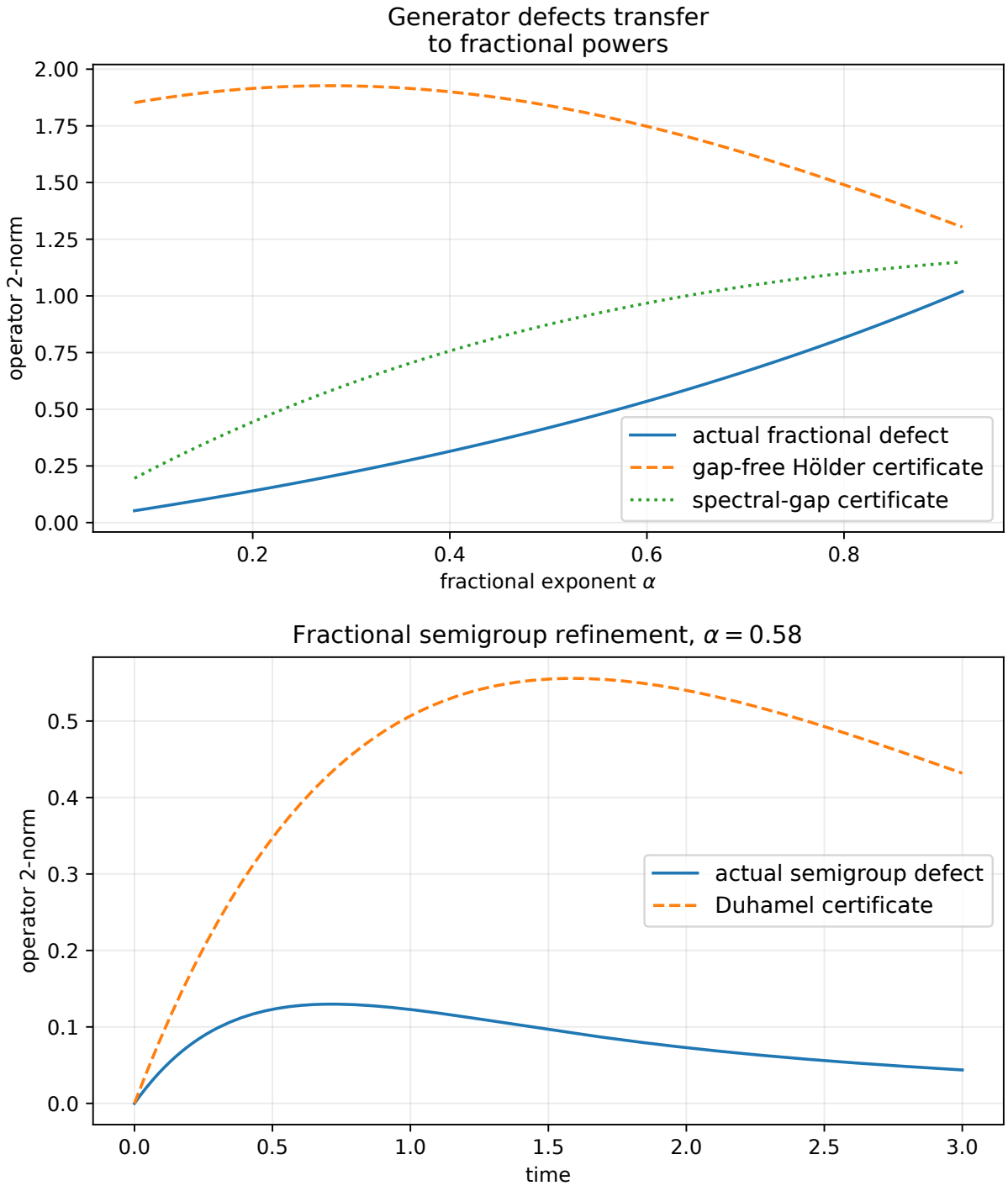


Figure 16. A generator-level refinement defect transfers quantitatively to fractional powers and then to fractional semigroups. The two upper curves are theorem certificates, not fitted error bars.

generators require their own contour calculus and resolvent bounds. Those domain questions are not inferred from the finite certificates.

Boundary 10.7 (Non-self-adjoint fractional powers). The spectral definition above is complete for finite self-adjoint positive operators. Sectorial and nonnormal operators require a resolvent or contour functional calculus. The same packet language can be used only after those analytic hypotheses are established.

11 Green operators and the Poisson seed

On the interval $(0,1)$ with Dirichlet boundary conditions, the Green kernel for $-u'' = f$ is

$$\boxed{G(x,y) = \min(x,y) - xy.}$$

Thus the point source δ_y generates the solution kernel, and a general source is reconstructed by

$$u(x) = \int_0^1 G(x,y) f(y)\, dy.$$

For n interior points with spacing $h = 1/(n+1)$, define

$$L_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

Theorem 11.1 (Exact discrete Green reconstruction at grid nodes). *Let $x_i = ih$. Then*

$$\boxed{\frac{1}{h}(L_h^{-1})_{ij} = G(x_i, x_j).}$$

Thus the scaled discrete inverse reproduces the continuum Dirichlet Green kernel exactly at the grid nodes.

Proof. The inverse of the unscaled tridiagonal matrix has entries

$$\frac{\min(i,j)(n+1-\max(i,j))}{n+1}.$$

Multiplying by h^2 and then dividing by h gives

$$h \min(i,j) \left(1 - \frac{\max(i,j)}{n+1}\right) = \min(x_i, x_j) - x_i x_j.$$

□

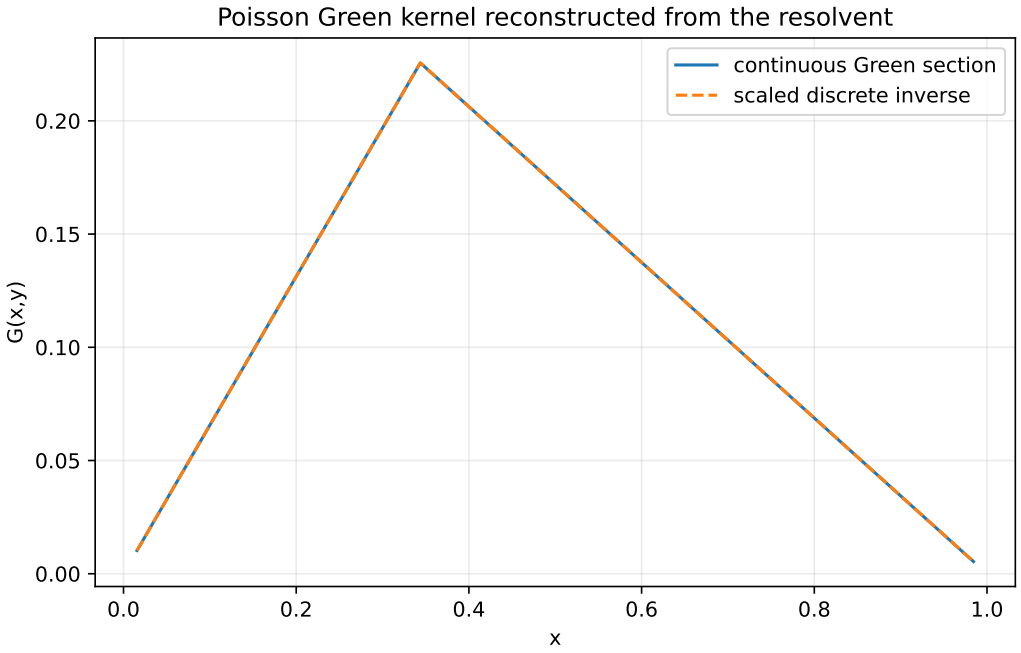


Figure 17. A section of the continuum Green kernel and the scaled discrete inverse. They agree at the grid nodes.

11.1 Low-mode convergence

The kth discrete eigenvalue is

$$\lambda_{k,h} = \frac{4}{h^2}\sin^2\left(\frac{k\pi h}{2}\right),$$

whereas the continuum eigenvalue is $(k\pi)^2$.

Theorem 11.2 (Second-order low-mode convergence). *For fixed k,*

$$\lambda_{k,h} = (k\pi)^2 + O(h^2).$$

More precisely, for sufficiently small h,

$$0 \le (k\pi)^2 - \lambda_{k,h} \le \frac{(k\pi)^4}{12}h^2.$$

Proof. For $x \ge 0$, $0 \le x^2 - \sin^2 x \le x^4/3$. Set $x = k\pi h/2$ and multiply by $4/h^2$. □

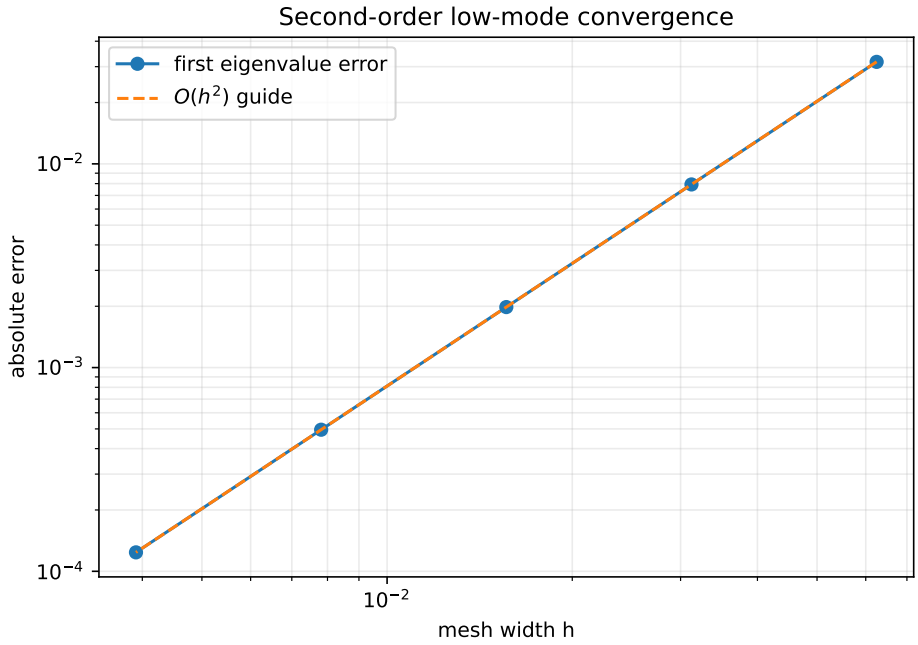


Figure 18. The first finite-difference eigenvalue converges with observed order approximately two.

12 Finite-state Feynman–Kac transfer

Let K be a column-stochastic matrix, let V be a potential on the states, and let f be terminal data. Define

$$M_V = \operatorname{diag}(e^{-V})K^T.$$

The backward Feynman–Kac value after n steps is

$$u_n = M_V^n f.$$

Theorem 12.1 (Discrete Feynman–Kac path formula)**.** *For an initial state i,*

$$\boxed{(u_n)_i = \mathbb{E}_i\left[\exp\left(-\sum_{r=0}^{n-1} V(X_r)\right) f(X_n)\right].}$$

Proof. For $n = 1$,

$$(M_V f)_i = e^{-V(i)} \sum_j K_{ji} f_j,$$

which is the stated expectation. Iterating and expanding the matrix product sums over every path $(i = i_0, i_1, \dots, i_n)$ with probability $\prod_r K_{i_{r+1}, i_r}$ and weight $\prod_r e^{-V(i_r)}$. □

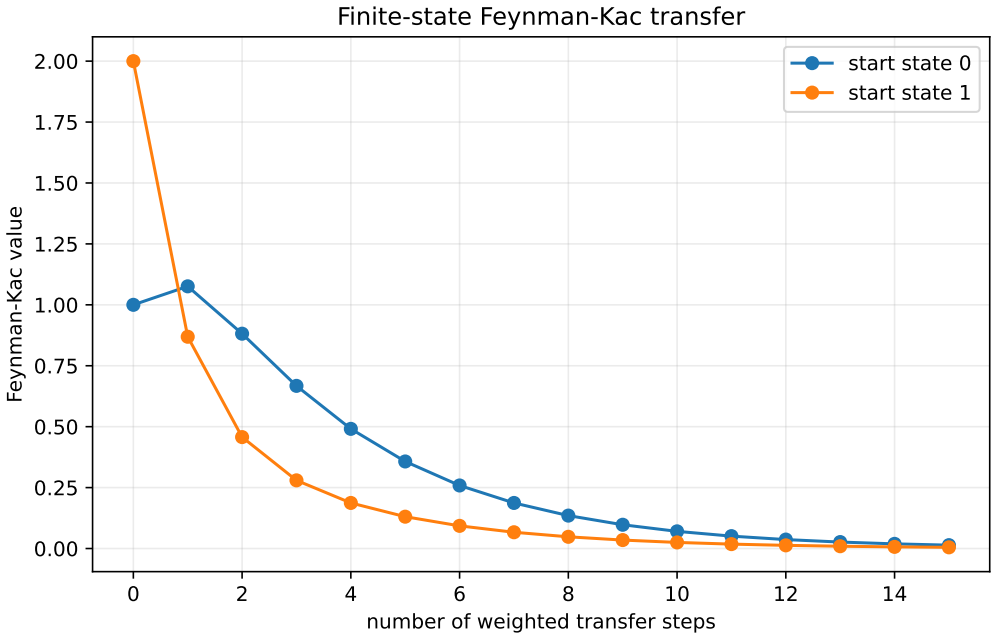


Figure 19. Potential-weighted transfer values computed by the finite Feynman–Kac operator. Exhaustive path enumeration agrees with the matrix formula in the supplied tests.

13 Exact elimination of latent packets

The central idea

Elimination removes hidden coordinates from the displayed state, but their future influence survives. The resolvent stores that influence as a self-energy. The time description stores it as a memory kernel. Both descriptions refer to the same declared block system.

Let $X = PX \oplus QX$ and write

$$A = \begin{pmatrix} A_{PP} & A_{PQ} \\ A_{QP} & A_{QQ} \end{pmatrix}.$$

Theorem 13.1 (Schur–Feshbach projected resolvent). *Assume both $zI - A$ and $zI - A_{QQ}$ are invertible. Define*

$$\boxed{F_P(z) = zI - A_{PP} - A_{PQ}(zI - A_{QQ})^{-1}A_{QP}.}$$

Then

$$\boxed{P(zI - A)^{-1}\big|_{\operatorname{Ran} P} = F_P(z)^{-1}.}$$

Proof. Solve the hidden block equation

$$(zI - A_{QQ})x_Q = A_{QP}x_P + y_Q$$

for x_Q and substitute it into the retained equation. The resulting retained coefficient is $F_P(z)$. Invertibility of the full operator and of the hidden block implies invertibility of this Schur complement. Setting

$y_Q = 0$ gives the stated identity as an operator on $\operatorname{Ran} P$. Equivalently, this is the block inverse formula. □

The self-energy

$$\Sigma_P(z) = A_{PQ}(zI - A_{QQ})^{-1}A_{QP}$$

contains the exact effect of the eliminated packets. A bare truncation A_{PP} discards this term and is generally not exact.

13.1 Time-domain memory

For the homogeneous system $\dot{x} = Ax$, variation of constants gives

$$x_Q(t) = e^{tA_{QQ}}x_Q(0) + \int_0^t e^{(t-r)A_{QQ}}A_{QP}x_P(r)\,dr.$$

Substitution yields

$$\dot{x}_P(t) = A_{PP}x_P(t) + A_{PQ}e^{tA_{QQ}}x_Q(0) + \int_0^t \mathcal{K}(t-r)x_P(r)\,dr,$$

where

$$\boxed{\mathcal{K}(t) = A_{PQ}e^{tA_{QQ}}A_{QP}.}$$

Theorem 13.2 (Memory/self-energy correspondence)**.** *For finite complex block matrices, assume* $\operatorname{Re} z > \operatorname{Re}\lambda$ *for every* $\lambda \in \sigma(A_{QQ})$*. Then the following improper integral is absolutely convergent and*

$$\boxed{\int_0^\infty e^{-zt}\mathcal{K}(t)\,dt = A_{PQ}(zI - A_{QQ})^{-1}A_{QP} = \Sigma_P(z).}$$

Proof. Put $B = A_{QQ}$ and $T_z(t) = e^{-zt}e^{tB} = e^{t(B-zI)}$. The generalized-eigenspace decomposition in finite dimensions writes T_z as a finite sum of constant matrix coefficients times $t^k e^{(\lambda-z)t}$. This follows by expanding the nilpotent part on each generalized eigenspace. Equivalently, apply the decomposition to left multiplication by B on the matrix algebra and evaluate the exponential at I. The strict spectral inequality makes every such term absolutely integrable on $(0,\infty)$ and forces it to tend to zero. Hence T_z is Bochner integrable and $T_z(t) \to 0$, including when B is nonnormal.

Let $J = \int_0^\infty T_z(t)\,dt$. Integrating both derivative identities $T_z' = (B - zI)T_z = T_z(B - zI)$ and using $T_z(0) = I$ gives

$$(zI - B)J = J(zI - B) = I.$$

Thus $J = (zI - B)^{-1}$. Continuous left and right multiplication carry the Bochner integral through the two rectangular transfer blocks, yielding the claim. □

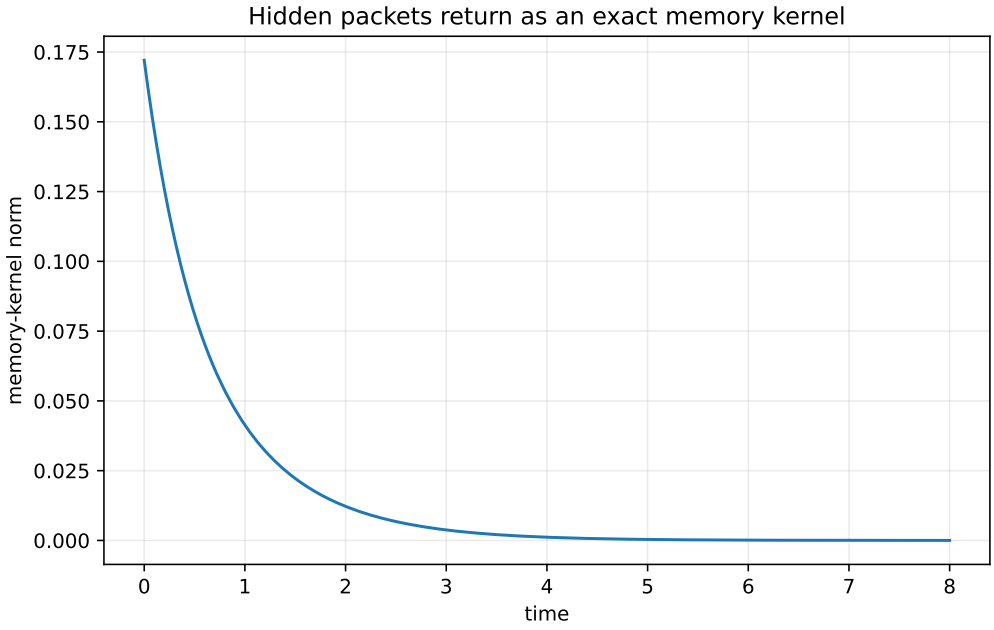


Figure 20. The norm of the exact memory kernel produced by eliminated packets.

14 Survival and hazard semigroups

Let A be a finite subgenerator: its off-diagonal entries are nonnegative and its column sums are nonpositive. These sign conditions alone allow nonabsorbing states. Eventual absorption is a separate hypothesis. Let p_0 be an initial transient-state distribution. Define

$$p(t) = e^{tA}p_0, \qquad S(t) = 1^T p(t).$$

The exit-rate row vector is

$$a^T = -1^T A \geq 0.$$

Theorem 14.1 (Semigroup survival identities). *For $t \geq 0$, the finite-time exit density and, wherever $S(t) > 0$, the hazard are*

$$\boxed{f(t) = a^T e^{tA} p_0 = -S'(t), \qquad h(t) = \frac{f(t)}{S(t)}.}$$

Consequently, S is nonincreasing and

$$S(t) = \exp\left(-\int_0^t h(r)\,dr\right)$$

whenever $S(t) > 0$. The density may be defective: its total mass is $1 - \lim_{t\to\infty} S(t)$. It is a probability density when absorption is eventual, so that this limit is zero.

Proof. Differentiate:

$$S'(t) = 1^T A e^{tA} p_0 = -a^T e^{tA} p_0.$$

The hazard identity is the definition of conditional instantaneous exit rate. Integrating $(\log S)' = -h$ gives the final formula. □

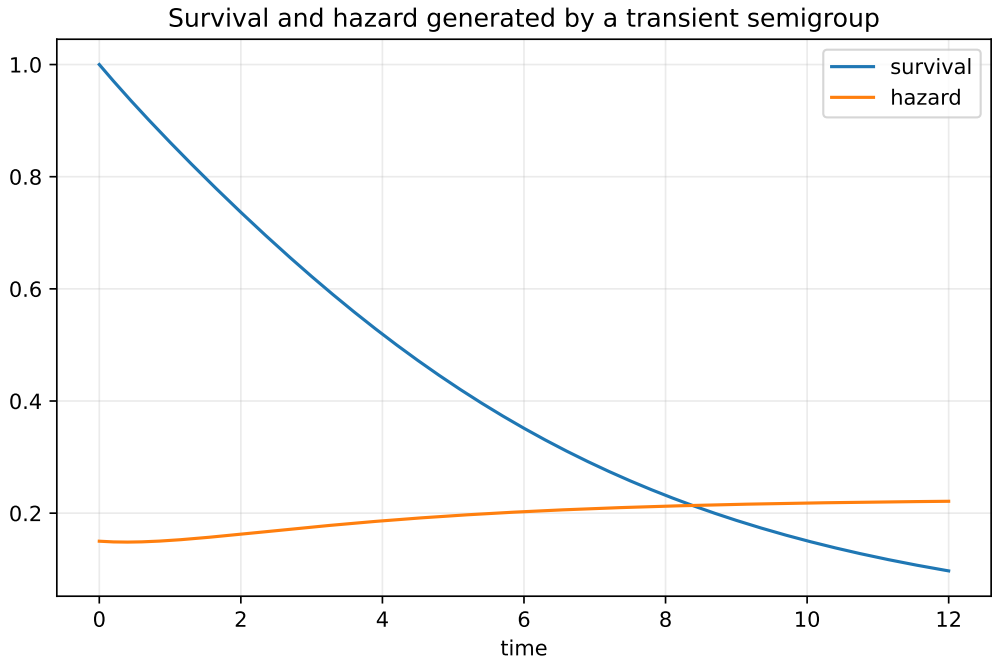


Figure 21. Survival and instantaneous hazard produced by one transient transfer semigroup.

15 One common operator language

The preceding constructions are not identical, but they use the same packet operations. The domain dictionary is summarized in table 1.

Realization	Seed	Transfer	Packet/reconstruction law
Markov	point mass δ_i	K^n or e^{tQ}	state lumping and $CK^n = \bar{K}^n C$
Conditional	random variable	P_Q	nested partitions and tower law
Bayesian	prior p	$\mathcal{B}_\lambda$	packet evidences and normalized synthesis
Fractional	spectral source	e^{-tL^α}	spectral projectors P_j
PDE	source f or δ_y	resolvent/Green operator	eigenmode or spatial packet synthesis
Feynman–Kac	terminal data f	$(\mathrm{diag}(e^{-V})K^T)^n$	weighted path reconstruction
Latent system	retained state	Feshbach/self-energy	hidden packets return as memory
Survival	initial transient law	e^{tA}	sum over transient packets

Table 1. The transfer-semigroup dictionary in the common operator language.

For clarity, the finite-dimensional holomorphic calculus used below has the following canonical definition. For A acting on $\mathbb{C}^d$, its generalized eigenspaces $E_\mu = \ker(A-\mu I)^d$ form a direct-sum decomposition over the eigenvalues. Define the unique linear operator $F(A)$ by

$$F(A)|_{E_\mu} = \sum_{k=0}^{d-1} \frac{F^{(k)}(\mu)}{k!}(A-\mu I)^k\Big|_{E_\mu}.$$

The formula is independent of a basis and of increasing the truncation index: the extra nilpotent powers vanish. It depends only on the finite Taylor jets of F at the spectrum. If F is holomorphic on an open neighborhood U of the spectrum, each eigenvalue admits a radius $r_\mu > 0$ with $\overline{D(\mu, r_\mu)} \subset U$, and Cauchy's formula gives

$$\frac{F^{(k)}(\mu)}{k!} = \frac{1}{2\pi i}\oint_{|z-\mu|=r_\mu} \frac{F(z)}{(z-\mu)^{k+1}}\,dz.$$

Consequently the same canonical operator is obtained from any such local Cauchy circles, even when U is disconnected. This is the standard finite Taylor-jet definition of the holomorphic matrix function.

Theorem 15.1 (Naturality under exact intertwining). *For finite complex matrices, let F be holomorphic on a neighborhood of the union of the spectra of A_f and A_c. Real matrices are treated by complexification. If*

$$CA_f = A_cC,$$

then

$$\boxed{CF(A_f) = F(A_c)C.}$$

In particular, exact generator refinement propagates simultaneously to semigroups, resolvents, integer powers, and analytic transfer functions.

Proof. The intertwining identity implies $C(A_f-\mu I)^k = (A_c-\mu I)^k C$ for every $k \geq 0$. It therefore sends the generalized μ-eigenspace of A_f into that of A_c. Use a common truncation index at least as large as both dimensions in the canonical formula above. Multiplying its finite sum by C now gives the desired identity on every generalized eigenspace. The direct-sum decomposition gives it on the full space. The argument covers nilpotent Jordan parts and uses only local spectral Taylor jets. It does not require one Taylor series to converge on both spectra. The zero-dimensional case is immediate. □

Remark 15.2 (Where nonlinearity enters). Bayesian conditioning does not fit the preceding linear functional calculus because normalization depends on the current state. It nevertheless obeys an exact composition law and has a closed packet reconstruction. Chapter III supplies the general nonlinear coordinate theorem. The Bayesian map is a probabilistic realization.

16 Numerical certificates

The recorded calculations were run on CPU. No GPU timing or performance claim is made. Floating-point residuals are implementation checks and are not substitutes for the symbolic proofs.

Numerical Certificate 16.1 (Test suite). The recorded automated test suite reports the following result:

$$\boxed{54/54 \text{ automated tests.}}$$

Certificate	Value
Packet-block semigroup composition residual	2.554×10^{-15}
Operator reconstruction residual	0
Random-walk mass error	0
Exact Markov generator-lumping defect	5.551×10^{-17}
Exact finite-horizon lumping error	2.220×10^{-16}
Perturbed ten-step lumping error	6.523×10^{-3}
Linear defect bound for that example	2.000×10^{-1}
Bayesian composition residual	5.551×10^{-17}
Bayesian packet reconstruction residual	0
Bayesian refinement commutation residual	1.110×10^{-16}
Bayesian defect-factorization residual	7.286×10^{-17}
Bayesian flow generator residual	1.959×10^{-11}
Conditional-expectation tower residual	0
Hermite Gram residual	9.992×10^{-15}
Fractional semigroup residual	2.082×10^{-17}
Maximum square-root subordination error	1.665×10^{-16}
Exact fractional-naturality residual	0
Fractional-power defect / gap certificate	$0.51059/0.95079$
Fractional-semigroup defect / certificate	$0.12969/0.44480$
Observed low-mode PDE convergence order	1.999854
Feynman–Kac enumeration residual	1.665×10^{-16}
Feshbach projected-resolvent residual	5.551×10^{-17}
Memory Laplace-transform residual	6.939×10^{-18}
Duhamel refinement identity residual	2.108×10^{-16}
Survival monotonicity violation	0
Hidden-Markov normalization error	1.110×10^{-16}

Table 2. Selected results from the machine-readable numerical record.

The tests cover projector reconstruction, packet-block semigroup composition, exact and approximate Markov lumping, Bayesian composition and flow, the exact Bayesian refinement criterion, information-loss factorization, likelihood-profile partitions, conditional-expectation towers, Hermite orthogonality, fractional semigroups, square-root subordination, fractional intertwining and semigroup-defect bounds, discrete Green kernels, PDE convergence, Feynman–Kac enumeration, Schur–Feshbach elimination, memory kernels, refinement defects, hidden Markov filtering, and survival curves.

17 Remaining conjectural extensions

The finite theory closes several operator identities but does not settle the corresponding universal continuum questions.

Conjecture 17.1 (Transfer-semigroup realization)**.** Every sufficiently regular realization of the seed calculus whose transfer stage is local in time admits a closed generator on a completed state space, and the packet analysis maps form a compatible system for its semigroup or evolution family.

Conjecture 17.2 (Uniform refinement control)**.** For a nested family of physically or analytically meaningful discretizations, there exist norms and constants independent of resolution such that

$$\left\|C_\ell e^{tA_{\ell+1}} - e^{tA_\ell} C_\ell\right\| \le M\, t\, \left\|C_\ell A_{\ell+1} - A_\ell C_\ell\right\|$$

on every bounded time interval. Summable generator defects would then produce a coherent inverse-limit evolution.

Conjecture 17.3 (Latent-packet closure)**.** For a stable infinite packet system, exact elimination of unresolved packets produces a causal memory kernel whose Laplace transform is the limiting packet self-energy. A Markovian closure exists precisely when this self-energy is constant, or can be represented by finitely many additional state packets.

Boundary 17.4 (What remains outside Chapter VI)*.* The following issues are not proved here:

1. existence and uniqueness for arbitrary stochastic differential equations;
2. posterior consistency or asymptotic normality in infinite statistical models;
3. domain invariance for unbounded fractional operators;
4. tightness and compactness required for stochastic continuum limits;
5. nonlinear PDE well-posedness beyond the stated finite and linear models;
6. an optimal packet choice for a general Markov, Bayesian, or PDE problem.

These belong to later realization-specific work or to external analytic theory.

18 Computational implementation

The accompanying Python package implements the formulas and finite models used in the chapter.

```
projector_transfer_semigroups
```

The computational organization mirrors the chapter:

`packets.py`	projectors, blocks, reconstruction, block composition
`semigroup.py`	matrix semigroups, defects, Feshbach operators, memory
`markov.py`	random walks, exact lumping, survival
`bayes.py`	normalized updates, packet evidence, conjugacy, flows, HMM
`conditional.py`	conditional expectation and Gaussian chaos
`fractional.py`	spectral powers, subordination, and fractional refinement certificates
`pde.py`	Dirichlet Laplacians, Green kernels, eigenvalue error
`feynman_kac.py`	weighted transfer and exhaustive path enumeration.

18.1 Representative algorithms

Listing 1. Normalized finite Bayesian update.

```
def bayes_update(prior, likelihood):
    p = normalize(prior)
    evidence = likelihood @ p
    if evidence <= 0:
        raise ValueError("positive evidence required")
    return likelihood * p / evidence
```

Listing 2. Packet block composition.

```
def block_composition(left, right):
    result = []
    for beta in range(len(left)):
        row = []
        for alpha in range(len(left)):
            value = sum(
                left[beta][gamma] @ right[gamma][alpha]
                for gamma in range(len(left))
            )
            row.append(value)
        result.append(row)
    return result
```

Listing 3. Finite Schur-Feshbach effective operator.

```
def feshbach_effective_operator(A, retained, z):
    App, Apq, Aqp, Aqq = split_blocks(A, retained)
    return z*I - App - Apq @ inv(z*I - Aqq) @ Aqp
```

The source corpus contains the implementation, tests, tables, figure-generation code and numerical records.

19 Conclusion

The transfer-semigroup realization gives one common operator language to several subjects that are often presented separately. Markov propagation is linear transfer of a probability seed. Conditional expectation is stochastic projection. Bayesian conditioning is normalized nonlinear transfer, and its exact finite coarsening law is characterized by packetwise likelihood sufficiency. Fractional calculus is spectral packet reweighting, and bounded positive refinement defects transfer to fractional powers and fractional semigroups by explicit resolvent and Duhamel estimates. A Green operator reconstructs a field from source packets. Feynman–Kac propagation adds path weights to stochastic transfer. Schur–Feshbach elimination shows that hidden packets are not destroyed. They return through self-energy and memory. Survival and hazard are scalar reconstructions of a transient semigroup.

The central structural identity is

$$\boxed{\text{generator defect} \longrightarrow \text{Duhamel semigroup defect} \longrightarrow \text{controlled refinement error.}}$$

Together with exact packet reconstruction, this makes the semigroup chapter the natural bridge from the foundational calculus to the quantum, computational, and arithmetic realizations that follow.

A Lean verification and scope

The formal development contains 341 explicit Lean theorem declarations across 47 proof modules. The saved compiler record reports a successful build and namespace audit under the pinned toolchain. The audit inspects 782 declarations, including definitions and generated auxiliaries. Only propext, Classical.choice and Quot.sound are allowed. These are different counting units from Part I's 33 theorem-like results.

The ledger has **33 complete, zero partial and zero pending results**. The last two results, 13.2 and 15.1, are complete. The three separately stated conjectures remain open and are excluded from this count.

"Complete" means the stated scope follows from the named checked targets and the ordinary specializations recorded in the ledger. Prose-to-Lean correspondence is manually reviewed. The fractional targets use complex Hilbert spaces. Complexification for real operators is an ordinary mathematical specialization, not a separately checked Lean construction. Completion of the theorem ledger does not certify every prose sentence, numerical figure or open conjecture.

Result	Manuscript statement	Lean scope
1.2	Finite semigroup law	Complete
2.1	Packet semigroup convolution	Complete
2.2	Reducing packets	Complete
3.1	Exact Duhamel refinement identity	Complete
3.2	Norm bound	Complete
3.3	Resolvent refinement identity	Complete
4.1	Finite Markov semigroup	Complete
5.1	Exact lumpability	Complete
5.2	Finite-horizon defect formula	Complete
6.1	Finite conditional-expectation projector	Complete
6.2	Tower law and martingale packets	Complete
7.1	Independent-likelihood composition	Complete
7.2	Bayesian packet reconstruction	Complete
7.3	Likelihood perturbation bound	Complete
8.1	Exact effective-likelihood factorization	Complete
8.2	Bayesian refinement criterion	Complete
8.4	Bayesian information-loss packet	Complete
8.5	Nested Bayesian packet tower	Complete
8.6	Canonical coarsest likelihood-resolving partition	Complete
9.1	Bayesian nonlinear semigroup	Complete
9.3	Prediction-update reconstruction	Complete
10.1	Fractional packet calculus	Complete
10.2	Poisson subordination	Complete
10.3	Fractional intertwining theorem	Complete
10.4	Fractional semigroup refinement	Complete
10.5	Fractional refinement towers	Complete
11.1	Exact discrete Green reconstruction at grid nodes	Complete
11.2	Second-order low-mode convergence	Complete
12.1	Discrete Feynman–Kac path formula	Complete
13.1	Schur–Feshbach projected resolvent	Complete
13.2	Memory/self-energy correspondence	Complete

Result	Manuscript statement	Lean scope
14.1	Semigroup survival identities	Complete
15.1	Naturality under exact intertwining	Complete

General nonnormal memory (13.2). The actual damped exponential is decomposed into finitely many terms of the form $t^k e^{(\mu-z)t}$. Integrability and convergence to zero follow from $\operatorname{Re}\mu < \operatorname{Re} z$. The formal proof uses generalized eigenspaces of left multiplication on the matrix algebra, so nilpotent Jordan parts are included. Both inverse identities follow by integrating the two exponential derivative formulas. A continuous linear sandwich map gives the literal rectangular matrix memory integral and the actual inverse. No normality or externally supplied growth bound is required.

Canonical holomorphic naturality (15.1). The standard finite Taylor-jet calculus defined before the theorem is constructed using the direct sum of generalized eigenspaces. Uniqueness, cutoff stability and exact rectangular naturality are proved. Cauchy's formula identifies the local coefficients with actual circle integrals. Admissible radii exist in every stated open holomorphic neighborhood, and changing them leaves the canonical operator unchanged. The argument permits disconnected neighborhoods and different matrix dimensions. It makes no assertion about general Banach-algebra contour deformation.

Fractional results (10.3–10.5). The formalization proves the scalar beta normalization, the Lebesgue integral representation of actual positive fractional powers, the literal resolvent-defect formula, both printed norm constants and exact intertwining. The rectangular complex-Hilbert-space semigroup proof substitutes those constants into Duhamel's identity. The varying-space tower proof gives summability of fractional-defect norms and uniform convergence of time-defect norm series on bounded time intervals. Here "summable" means summable in norm. Operators with unrelated codomains are not added together. A separate fixed-space theorem proves operator-series convergence when such a sum is defined.

Verification and numerical evidence. The saved records report the full pinned build, project-wide axiom audit, all 54 companion tests and six verifier controls passed.
The 54 tests comprise the original 37 tests and 17 boundary and validation cases.
The numerical records identify the 19 figure pairs and the separate 79-field comparison.
The numerical comparison and the visual replay have separate execution records.
Source contracts, exact theorem targets, full logs and checked ledgers accompany the source files.
Numerical tests are finite regressions. They are not substitutes for the Lean proofs.

Part II Extended mathematical development

Overview

A bounded generator and a finite packet system turn a semigroup into a matrix of transfer blocks. We prove convolution, differential, path, Laplace, resolvent, and defect formulas. Exact intertwining yields an inverse limit semigroup. Weighted summable adjacent defects give convergent reconstructed evolutions, while range leakage determines whether the limit remains a semigroup. For finite Markov chains we identify a canonical lumping obstruction and quantify its failure. Conditional expectations give orthogonal martingale packets, and the Ornstein–Uhlenbeck semigroup gives a countable Hermite model. Bayesian updating is treated as normalized transfer. We characterize exact coarse sufficiency, determine the coarsest nontrivial fixed sufficient partition for a likelihood family when the common level sets are nontrivial, and split relative information into coarse and local terms. Positive fractional powers are handled through spectral, Balakrishnan, and subordination formulas. Green operators, finite differences, and finite state Feynman–Kac formulas provide analytic realizations. Schur–Feshbach elimination produces a rational self-energy and an exact memory kernel. Every stable finite hidden system yields a polynomial exponential expansion, including Jordan blocks. A certified optimal primary block selection and a Volterra estimate control reduced trajectories. State dependent killing, survival laws, moments, and competing risks complete the finite theory.

E1 Introduction

One parameter semigroups provide a common language for evolution equations, Markov processes, diffusion, and resolvent methods. Standard analytic foundations appear in [Paz83, EN00]. Finite Markov chains and their coarse descriptions are treated in [KS60, Nor97, RP81]. Conditional expectation and martingale difference decompositions are basic parts of probability theory [Wil91]. Gaussian transfer and Hermite expansions provide a spectral model for the Ornstein–Uhlenbeck semigroup [Jan97]. Fractional powers of generators and subordinate semigroups belong to the functional calculus of positive and sectorial operators [Bal60, SSV12]. The Feynman–Kac formula, projection methods, and phase type laws add three further views of transfer [Kac49, Fes58, Zwa61, Neu81].

These subjects share a simple operation. A state is propagated, resolved into declared components, compared across levels, and reconstructed. The same operation can be linear, as in a Markov semigroup, or normalized and nonlinear, as in Bayesian conditioning. Hidden components can be discarded from the state vector without disappearing from the dynamics. Their exact influence returns through a frequency dependent self-energy or a time dependent memory kernel.

The first sections of this part treat bounded semigroups on Banach spaces. A finite packet system resolves every operator into input and output blocks. The semigroup law becomes block convolution. Splitting the generator into its packet diagonal and transfer part gives a convergent Dyson expansion over directed packet paths. For a coarse map C, the generator defect

$$\Delta = CA_f - A_c C$$

produces the entire semigroup defect through a Duhamel integral. The same defect controls resolvents and holomorphic functions of the generators.

The next question concerns refinement towers. Exact intertwining gives a coordinatewise semigroup on the bounded inverse limit. Approximate refinement is more delicate. Even if every adjacent defect is small, reconstruction from a fine level can introduce a second error because a coarse state need not remain inside the reconstructed fine subspace. We separate these two effects. Summability of adjacent defects gives a uniform limit of reconstructed coarse evolutions. An asymptotic closure condition is then sufficient for the limiting family to satisfy the semigroup law.

The probabilistic part begins with finite Markov chains. Exact lumpability is the vanishing of a canonical obstruction on the kernel of aggregation. In coordinates, this is equality of the transition mass sent from any two states in the same source block into every target block. The usual finite horizon defect formula follows from telescoping. Conditional expectation is then treated as an orthogonal projector. A nested family of partitions produces mutually orthogonal martingale packets and an exact energy decomposition.

Bayesian updating requires more care because normalization depends on the current prior. Given a partition, every fine update has an exact two stage description. The block masses are updated by prior dependent average likelihoods, while the conditional distribution inside each block is updated by its centered likelihood variation. A fixed coarse Bayesian rule works for every prior precisely when the likelihood is constant inside each block. This gives a finite and exact form of Bayesian sufficiency. For a family of likelihoods with at least two common level sets, those level sets form the unique coarsest nontrivial fixed sufficient partition. Relative entropy then separates exactly into information retained by the coarse packet masses and information carried by the local conditional changes. The same part derives perturbation bounds, the replicator flow, a variance identity for accumulated evidence, and stability estimates for hidden Markov filtering.

For a positive operator L, spectral packets define L^{α} and $\mathrm{e}^{-tL^{\alpha}}$. Exact intertwining passes to every fractional power. When both fine and coarse operators have a common positive lower bound m, the Balakrishnan formula gives the quantitative estimate

$$\left\| CL_f^{\alpha} - L_c^{\alpha} C \right\| \leq \alpha m^{\alpha-1} \left\| CL_f - L_c C \right\|, \qquad 0 < \alpha < 1.$$

A Duhamel argument then controls the fractional semigroup defect. This closes a finite version of fractional naturality under explicit spectral hypotheses.

The final sections move from generators to source reconstruction and hidden variables. The one-dimensional Dirichlet Green kernel is reproduced exactly by the scaled inverse of the second difference matrix at grid nodes. A discrete maximum principle gives a complete second order error estimate for smooth Poisson solutions. Finite state Feynman–Kac transfer is proved in discrete and continuous time. Schur–Feshbach elimination yields the exact projected resolvent, the retained memory equation, and the Laplace correspondence between memory and self-energy. Every stable finite hidden system gives a polynomial exponential memory expansion. Primary blocks can be ranked optimally for the certified L^1 bound, and a Volterra estimate carries kernel error to the retained solution. Survival, event density, moments, and competing risks are then read directly from a transient semigroup.

Several claims are intentionally absent. No arbitrary unbounded operator is asserted to generate a semigroup. The inverse limit theorem uses bounded generators or an explicit uniform continuity condition. Bayesian sufficiency is proved for a finite partition and a fixed likelihood. The fractional defect estimate uses a common spectral gap. The Poisson estimate concerns the stated one-dimensional scheme. Memory compression is controlled for the declared finite or integrable kernels. General stochastic differential equations, posterior consistency, nonlinear partial differential equations, and optimal model reduction remain outside this part.

Realization	Transfer	Exact coarse condition
Operator evolution	e^{tA}	generator intertwining
Markov chain	K^n or e^{tQ}	equality of aggregated transition masses
Conditional expectation	P_Q	nesting of the underlying partitions
Bayesian update	$\mathcal{B}_\lambda$	likelihood constant inside each coarse packet
Fractional diffusion	e^{-tL^α}	spectral or resolvent intertwining
Hidden dynamics	projected resolvent	vanishing self-energy, or an explicit memory state
Survival law	transient semigroup	summation over all transient states

Table E1. The transfer operations studied in this part and the condition that makes each coarse description exact.

The same words have different meanings in the realizations below. The table records the object that is transferred and the condition that makes its coarse description exact.

E2 Packet resolved semigroups

Let X be a complex Banach space. A finite packet system is a family of bounded idempotents

$$\mathcal{P} = (P_\alpha)_{\alpha \in I}$$

such that

$$P_\alpha P_\beta = \delta_{\alpha\beta} P_\alpha, \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

No orthogonality is needed in this section.

Definition E2.1 (Packet resolved transfer system)**.** A packet resolved transfer system is a triple

$$(X, A, \mathcal{P})$$

consisting of a Banach state space, a bounded generator, and a finite packet system. A marked transfer system also includes an initial state or source $s \in X$. The topology and all limiting operations are supplied by the ambient Banach space structure.

Definition E2.2 (Packet blocks)**.** For $B \in \mathcal{B}(X)$, define

$$B_{\beta\alpha} = P_\beta B P_\alpha.$$

If $A \in \mathcal{B}(X)$ and $T(t) = e^{tA}$, write

$$T_{\beta\alpha}(t) = P_\beta T(t) P_\alpha.$$

Theorem E2.3 (Semigroup and packet convolution). *For every bounded operator A, the family $T(t) = \mathrm{e}^{tA}$ is a uniformly continuous group and*

$$T(s+t) = T(s)T(t).$$

For every $\alpha, \beta \in I$,

$$T_{\beta\alpha}(s+t) = \sum_{\gamma \in I} T_{\beta\gamma}(s) T_{\gamma\alpha}(t).$$

Proof. The exponential series converges in operator norm. Since every power of A commutes with every other power, the Cauchy product gives the group law. Inserting $I_X = \sum_\gamma P_\gamma$ between the two factors gives the block formula. □

Proposition E2.4 (Block differential system). *The packet blocks are differentiable in operator norm and satisfy*

$$\frac{\mathrm{d}}{\mathrm{d}t} T_{\beta\alpha}(t) = \sum_{\gamma \in I} A_{\beta\gamma} T_{\gamma\alpha}(t) = \sum_{\gamma \in I} T_{\beta\gamma}(t) A_{\gamma\alpha}.$$

Their initial values are

$$T_{\beta\alpha}(0) = \delta_{\beta\alpha} P_\alpha.$$

Proof. Differentiate $T(t)$ in operator norm and insert the packet resolution between A and $T(t)$, then between $T(t)$ and A. □

Theorem E2.5 (Laplace resolvent and packet transform). *Let $A \in \mathcal{B}(X)$, let $T(t) = \mathrm{e}^{tA}$, and let*

$$\operatorname{Re} z > \|A\|.$$

Then the Bochner integral converges in operator norm and

$$(zI - A)^{-1} = \int_0^\infty \mathrm{e}^{-zt} T(t)\, \mathrm{d}t.$$

For every pair of packets,

$$P_\beta (zI - A)^{-1} P_\alpha = \int_0^\infty \mathrm{e}^{-zt} T_{\beta\alpha}(t)\, \mathrm{d}t.$$

Proof. The estimate $\|T(t)\| \le \mathrm{e}^{t\|A\|}$ gives absolute convergence of the operator valued integral. Let

$$R(z) = \int_0^\infty \mathrm{e}^{-zt} T(t)\, \mathrm{d}t.$$

Since $T'(t) = AT(t) = T(t)A$, integration by parts gives

$$(zI - A) R(z) = -\int_0^\infty \frac{\mathrm{d}}{\mathrm{d}t} (\mathrm{e}^{-zt} T(t))\, \mathrm{d}t = I.$$

The same calculation on the other side gives $R(z)(zI - A) = I$. Multiplication by P_β and P_α may be passed under the norm convergent integral, which gives the packet formula. □

Corollary E2.6 (Resolvent expansion over packet paths). *With $A = D + N$ as in the directed path decomposition, let*

$$R_D(z) = (zI - D)^{-1}.$$

If

$$\|R_D(z)N\| < 1,$$

then

$$(zI - A)^{-1} = \sum_{m=0}^{\infty} \bigl(R_D(z)N\bigr)^m R_D(z).$$

After multiplication by P_β and P_α, the term of order m is a sum over directed packet paths of length m from α to β.

Proof. Factor

$$zI - A = (zI - D)(I - R_D(z)N)$$

and expand the second inverse by its norm convergent Neumann series. The diagonal resolvent preserves each packet and every factor N changes the packet along an active transfer edge. □

Theorem E2.7 (Reducing packet criterion). *The following conditions are equivalent.*

1. *$AP_\alpha = P_\alpha A$ for every α.*
2. *$A_{\beta\alpha} = 0$ whenever $\beta \neq \alpha$.*
3. *$T_{\beta\alpha}(t) = 0$ whenever $\beta \neq \alpha$ and $t \in \mathbb{R}$.*
4. *Every packet space $P_\alpha X$ is invariant under $T(t)$ for every t.*

When these conditions hold,

$$T(t)|_{P_\alpha X} = \mathrm{e}^{tA_{\alpha\alpha}}.$$

Proof. The first two conditions are equivalent because the packet resolution gives

$$AP_\alpha = \sum_{\beta} A_{\beta\alpha}, \qquad P_\alpha A = \sum_{\gamma} A_{\alpha\gamma}.$$

A block diagonal generator has block diagonal powers, so the second condition implies the third. Differentiating the off-diagonal blocks at zero proves the converse. The equivalence with invariance follows from the definition of the blocks. The final identity follows by restricting the exponential series. □

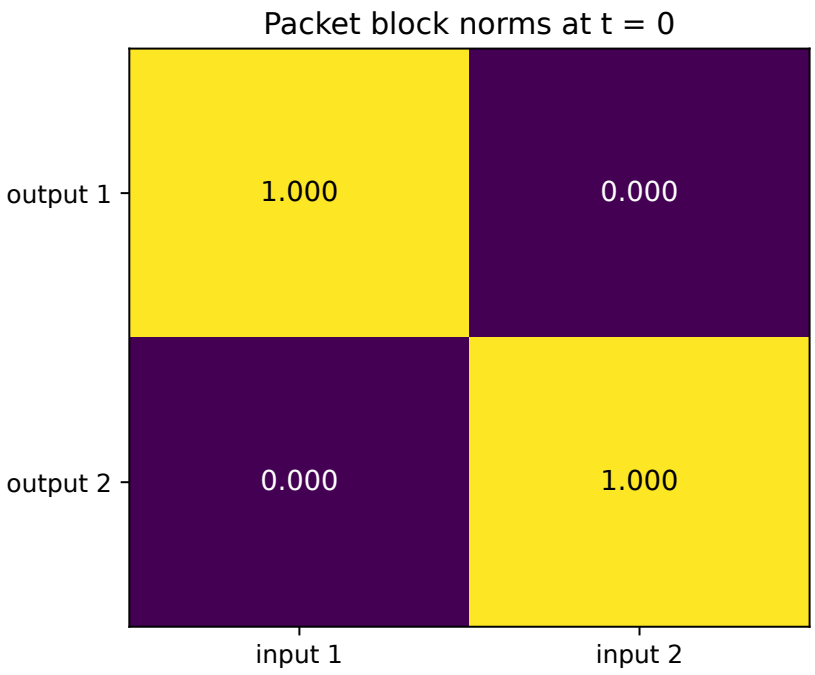


Figure E1. Packet block norms at three times. Transfer appears in the off-diagonal blocks as soon as the generator couples the two packet spaces. The three panels are shown in consecutive figures.

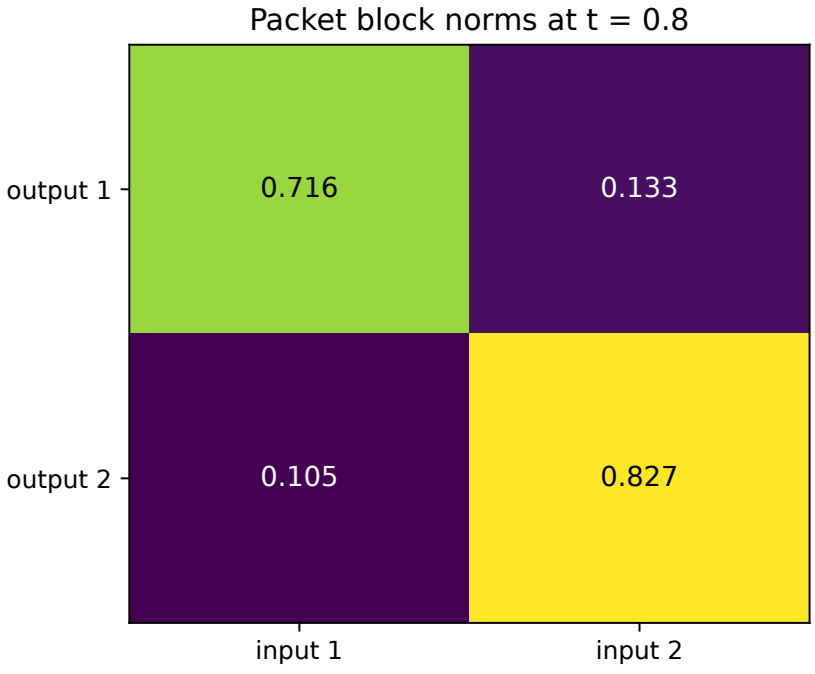


Figure E2. Packet block norms at time 0.8. The same fixed packet spaces and norm scale are used in the preceding panels.

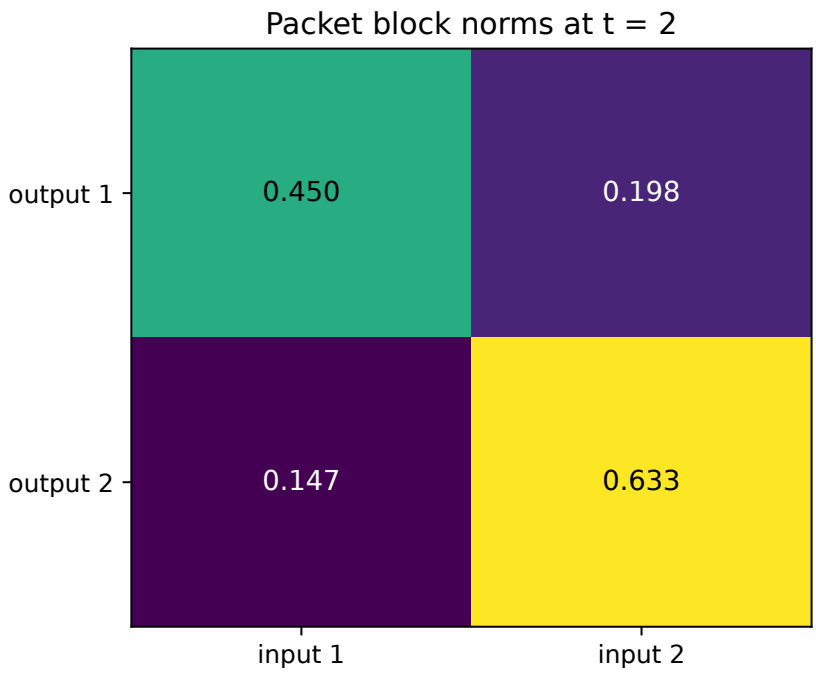


Figure E3. Packet block norms at time 2.0. The same fixed packet spaces and norm scale are used in the preceding panels.

E2.1 Directed path expansion

Set

$$D = \sum_{\alpha \in I} P_\alpha A P_\alpha, \qquad N = A - D.$$

The operator D preserves every packet and N contains only transfers between distinct packets.

Theorem E2.8 (Dyson expansion over packet paths)**.** *For $t \geq 0$,*

$$\mathrm{e}^{tA} = \mathrm{e}^{tD} + \sum_{m=1}^{\infty} \int_{0<s_1<\cdots<s_m<t} \mathrm{e}^{(t-s_m)D} N \mathrm{e}^{(s_m - s_{m-1})D} \cdots N \mathrm{e}^{s_1 D} \, \mathrm{d}s_1 \cdots \mathrm{d}s_m.$$

The series converges absolutely in operator norm. After multiplication by P_β and P_α, the term of order m is a sum over directed packet paths of length m from α to β.

Proof. Write the variation of constants identity

$$\mathrm{e}^{tA} = \mathrm{e}^{tD} + \int_0^t \mathrm{e}^{(t-s)D} N \mathrm{e}^{sA} \, \mathrm{d}s.$$

Iterating the identity gives the displayed series. The norm of the term of order m is at most

$$\mathrm{e}^{t\|D\|} \frac{(t\,\|N\|)^m}{m!}.$$

The majorant is summable. Inserting the packet resolution around every factor N leaves only products whose consecutive packet labels are joined by nonzero transfer blocks. □

Corollary E2.9 (Termination on an acyclic transfer graph)**.** *Suppose the directed graph with an edge $\alpha \to \beta$ whenever $P_\beta N P_\alpha \neq 0$ is acyclic. If its longest directed path has length q, then every Dyson term of order greater than q vanishes.*

Proof. Every nonzero term of order m contains a directed transfer path of length m. An acyclic graph has no path longer than q. □

E3 Generator defects and functional calculus

Let X_f and X_c be complex Banach spaces. Let $A_f \in \mathcal{B}(X_f)$, $A_c \in \mathcal{B}(X_c)$, and $C \in \mathcal{B}(X_f, X_c)$. Define

$$\Delta = CA_f - A_c C$$

and

$$\mathfrak{D}(t) = C\mathrm{e}^{tA_f} - \mathrm{e}^{tA_c} C.$$

Theorem E3.1 (Duhamel defect identity)**.** *For every $t \geq 0$,*

$$\mathfrak{D}(t) = \int_0^t \mathrm{e}^{(t-s)A_c} \Delta \mathrm{e}^{sA_f}\, \mathrm{d}s.$$

If

$$\left\|\mathrm{e}^{tA_f}\right\| \leq M_f \mathrm{e}^{\omega_f t}, \qquad \left\|\mathrm{e}^{tA_c}\right\| \leq M_c \mathrm{e}^{\omega_c t},$$

then

$$\|\mathfrak{D}(t)\| \leq M_f M_c \|\Delta\| \int_0^t \mathrm{e}^{\omega_c (t-s) + \omega_f s}\, \mathrm{d}s.$$

For contraction semigroups,

$$\|\mathfrak{D}(t)\| \leq t \|\Delta\|.$$

Proof. For fixed t, set

$$F(s) = \mathrm{e}^{(t-s)A_c} C \mathrm{e}^{sA_f}.$$

Then

$$F'(s) = \mathrm{e}^{(t-s)A_c} \Delta \mathrm{e}^{sA_f}.$$

Integrating from zero to t gives the identity. The estimates follow by taking norms under the integral. □

Proposition E3.2 (Infinitesimal recovery of the defect)**.** *The generator defect is the first order semigroup defect:*

$$\lim_{t \to 0} \frac{\mathfrak{D}(t)}{t} = \Delta$$

in operator norm. Consequently,

$$C\mathrm{e}^{tA_f} = \mathrm{e}^{tA_c} C \quad \text{for every } t$$

if and only if $CA_f = A_c C$.

Proof. The operator norm expansions

$$\mathrm{e}^{tA_f} = I + tA_f + o(t), \qquad \mathrm{e}^{tA_c} = I + tA_c + o(t)$$

give the limit. The equivalence follows from this limit and theorem E3.1. □

Theorem E3.3 (Resolvent defect identity). *Let z belong to the resolvent sets of A_f and A_c. With*

$$R_f(z) = (zI - A_f)^{-1}, \qquad R_c(z) = (zI - A_c)^{-1},$$

one has

$$CR_f(z) - R_c(z)C = R_c(z)\Delta R_f(z).$$

Proof. Multiply the proposed identity on the left by $zI - A_c$ and on the right by $zI - A_f$. Both sides reduce to

$$(zI - A_c)C - C(zI - A_f) = \Delta.$$

□

Theorem E3.4 (Holomorphic defect formula). *Let Γ be the positively oriented rectifiable boundary of a finite union of pairwise disjoint bounded Jordan domains. Assume that both spectra lie inside this union and that Γ lies in the common resolvent set. Thus its winding number is one about every spectral point. If g is holomorphic on a neighborhood of the closure of the union, then*

$$Cg(A_f) - g(A_c)C = \frac{1}{2\pi \mathrm{i}} \int_\Gamma g(z) R_c(z) \Delta R_f(z)\, \mathrm{d}z.$$

In particular,

$$\left\| Cg(A_f) - g(A_c)C \right\| \le \frac{\operatorname{length}(\Gamma)}{2\pi} \sup_{z \in \Gamma} |g(z)| \sup_{z \in \Gamma} \|R_c(z)\| \sup_{z \in \Gamma} \left\|R_f(z)\right\| \|\Delta\|.$$

Proof. Apply the Dunford integral formula to both operators and use theorem E3.3 under the contour integral. The norm estimate is immediate. □

Corollary E3.5 (Exact naturality). *If $CA_f = A_c C$, then*

$$Cg(A_f) = g(A_c)C$$

for every holomorphic function covered by theorem E3.4. The conclusion includes exponentials, resolvents, and polynomials.

Proof. Set $\Delta = 0$ in theorem E3.4. □

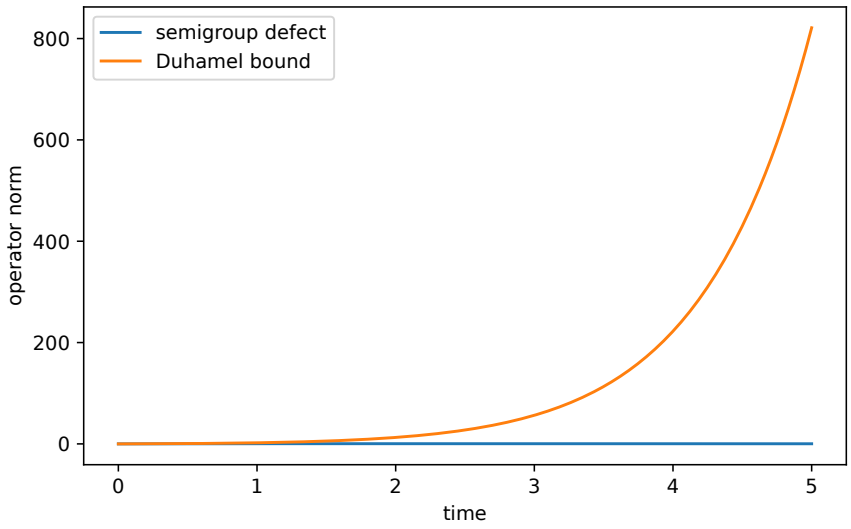


Figure E4. A finite semigroup defect and a direct Duhamel bound. The bound is designed to be uniform rather than sharp in this example.

E4 Refinement towers and limiting evolution

> **How to read the result**
>
> The limiting argument names its topology, reconstruction maps, and summability weights. Convergence of reconstructed trajectories is distinct from closure under the semigroup law. The range-leakage condition identifies the additional obstruction to that closure.

The exact inverse limit and the approximate completion problem require different hypotheses. Exact intertwining preserves the inverse limit coordinate by coordinate. Approximate intertwining gives a Cauchy family only after reconstruction maps are chosen.

E4.1 Exact inverse limits

Let

$$X_0 \xleftarrow{C_0} X_1 \xleftarrow{C_1} X_2 \leftarrow \cdots$$

be Banach spaces with bounded connecting maps. Define the bounded inverse limit

$$X_\infty = \left\{ (x_\ell)_{\ell \geq 0} : C_\ell x_{\ell+1} = x_\ell, \quad \sup_\ell \|x_\ell\| < \infty \right\}$$

with the supremum norm.

Lemma E4.1 (Completeness of the bounded inverse limit). *The space X_∞ is Banach.*

Proof. The product of the spaces X_ℓ, equipped with the bounded supremum norm, is Banach. The equations $C_\ell x_{\ell+1} = x_\ell$ define a closed linear subspace. □

Theorem E4.2 (Exact projective limit semigroup). *For each ℓ, let $T_\ell(t)$ be a strongly continuous semigroup on X_ℓ. Assume*

$$C_\ell T_{\ell+1}(t) = T_\ell(t) C_\ell$$

for every ℓ and $t \geq 0$, and assume the uniform bound

$$\sup_\ell \|T_\ell(t)\| \leq M \mathrm{e}^{\omega t}.$$

Define

$$T_\infty(t)(x_\ell)_\ell = (T_\ell(t) x_\ell)_\ell.$$

Then $T_\infty(t)$ is a bounded semigroup on X_∞. It is strongly continuous if

$$\lim_{t \downarrow 0} \sup_\ell \|T_\ell(t) x_\ell - x_\ell\| = 0$$

for every $(x_\ell)_\ell \in X_\infty$.
If $T_\ell(t) = \mathrm{e}^{tA_\ell}$, the generators are bounded, and

$$\sup_\ell \|A_\ell\| < \infty, \qquad C_\ell A_{\ell+1} = A_\ell C_\ell,$$

then the continuity condition holds and

$$A_\infty(x_\ell)_\ell = (A_\ell x_\ell)_\ell$$

is the bounded generator of T_∞.

Proof. Exact intertwining shows that $(T_\ell(t)x_\ell)_\ell$ remains compatible. The uniform growth bound makes the coordinatewise operator bounded on the supremum norm. The semigroup law holds in every coordinate. The stated continuity condition is exactly strong continuity in the supremum norm.

For bounded generators with common bound M_A,

$$\sup_\ell \left\|(\mathrm{e}^{tA_\ell} - I)x_\ell\right\| \leq (\mathrm{e}^{tM_A} - 1) \sup_\ell \|x_\ell\|.$$

This tends to zero with t. Exact generator intertwining makes $(A_\ell x_\ell)_\ell$ compatible, and coordinatewise differentiation identifies the generator. □

Remark E4.3. Strong continuity on each level does not by itself imply strong continuity in the supremum norm of the inverse limit. The additional uniform condition in theorem E4.2 cannot be omitted without replacing the topology.

E4.2 Summable adjacent defects

Assume now that each $C_\ell : X_{\ell+1} \to X_\ell$ has a bounded right inverse

$$R_\ell : X_\ell \to X_{\ell+1}, \qquad C_\ell R_\ell = I_{X_\ell}.$$

For $m < n$, set

$$C_{m,n} = C_m C_{m+1} \cdots C_{n-1}, \qquad R_{n,m} = R_{n-1} \cdots R_m.$$

The empty products are

$$C_{m,m} = I_{X_m}, \qquad R_{m,m} = I_{X_m}.$$

Definition E4.4 (Reconstructed fine evolution)**.** For a semigroup $T_n(t)$ on X_n, define the evolution seen from level m by

$$U_{m,n}(t) = C_{m,n} T_n(t) R_{n,m}.$$

Theorem E4.5 (Weighted summable defect completion)**.** *Define the adjacent semigroup defect*

$$E_n(t) = C_n T_{n+1}(t) - T_n(t) C_n.$$

Fix a coarse level m and a time horizon $T > 0$. For $n \geq m$, set

$$w_{m,n} = \|C_{m,n}\| \, \|R_n R_{n,m}\|.$$

If

$$\sum_{n=m}^{\infty} w_{m,n} \sup_{0 \leq t \leq T} \|E_n(t)\| < \infty,$$

then $U_{m,n}(t)$ converges in operator norm uniformly for $0 \leq t \leq T$. Its limit is denoted by $U_m(t)$, and

$$\sup_{0 \leq t \leq T} \|U_m(t) - U_{m,n}(t)\| \leq \sum_{j=n}^{\infty} w_{m,j} \sup_{0 \leq t \leq T} \|E_j(t)\|.$$

Proof. Since $C_n R_n = I$,

$$U_{m,n+1}(t) - U_{m,n}(t) = C_{m,n}(C_n T_{n+1}(t) - T_n(t) C_n) R_n R_{n,m}.$$

Therefore

$$\|U_{m,n+1}(t) - U_{m,n}(t)\| \leq w_{m,n} \, \|E_n(t)\|.$$

The weighted summability hypothesis makes the sequence uniformly Cauchy. Summing the tail estimates gives the stated bound. □

Corollary E4.6 (Contractive summable defect completion)**.** *Assume*

$$\|C_\ell\| \leq 1, \qquad \|R_\ell\| \leq 1, \qquad \|T_\ell(t)\| \leq 1$$

for every ℓ and $t \geq 0$. If

$$\sum_{\ell=0}^{\infty} \sup_{0 \leq t \leq T} \|E_\ell(t)\| < \infty,$$

then the conclusion of theorem E4.5 holds, and

$$\sup_{0 \leq t \leq T} \|U_m(t) - U_{m,n}(t)\| \leq \sum_{\ell=n}^{\infty} \sup_{0 \leq t \leq T} \|E_\ell(t)\|.$$

Proof. Every weight $w_{m,n}$ is at most one. □

Corollary E4.7 (Generator defect criterion). *Assume*

$$\|C_\ell\| \le 1, \qquad \|R_\ell\| \le 1,$$

and suppose $T_\ell(t) = \mathrm{e}^{tA_\ell}$ are contraction semigroups with bounded generators. Set

$$\Delta_\ell = C_\ell A_{\ell+1} - A_\ell C_\ell.$$

If

$$\sum_{\ell=0}^{\infty} \|\Delta_\ell\| < \infty,$$

then the conclusion of corollary E4.6 holds on every bounded time interval, and

$$\sup_{0\le t\le T} \left\|U_m(t) - U_{m,n}(t)\right\| \le T \sum_{\ell=n}^{\infty} \|\Delta_\ell\|.$$

Proof. Apply the contraction estimate in theorem E3.1 to every adjacent pair. □

The limit in theorem E4.5 need not satisfy the semigroup law. The obstruction comes from leaving the reconstructed fine subspace between two time steps.

Theorem E4.8 (Asymptotic closure criterion). *Assume the weighted convergence in theorem E4.5 holds for every bounded time interval, and define*

$$P_{m,n} = R_{n,m} C_{m,n}$$

on X_n. Suppose that for every $T > 0$,

$$\rho_{m,n}(T) = \sup_{0\le s,t\le T} \left\|C_{m,n} T_n(s)(I - P_{m,n}) T_n(t) R_{n,m}\right\| \longrightarrow 0.$$

Then

$$U_m(s+t) = U_m(s) U_m(t)$$

for all $s, t \ge 0$. If the convergence in theorem E4.5 is locally uniform and every T_n is norm continuous, then U_m is a norm continuous semigroup.

Proof. At a finite level,

$$U_{m,n}(s)U_{m,n}(t) - U_{m,n}(s+t) = C_{m,n} T_n(s)(P_{m,n} - I) T_n(t) R_{n,m}.$$

Its norm is at most $\rho_{m,n}(T)$ for $s, t \le T$. Pass to the limit using uniform convergence. Norm continuity follows from a uniform limit of norm continuous functions on compact intervals. □

Remark E4.9. Summable generator defects control consistency between adjacent levels. The quantity $\rho_{m,n}(T)$ controls closure of the reconstructed fine range under the fine evolution. These are different requirements.

Example E4.10 (Summable defects do not force a semigroup limit)**.** Let $X_0 = \mathbb{R}$ and let $X_n = \mathbb{R}^2$ for every $n \geq 1$. Set

$$C_0(x, y) = x, \qquad R_0 x = (x, 0),$$

and use identity connecting maps and reconstructions above level one. Let $T_0(t) = I$ and let every higher level carry the rotation group

$$T_n(t) = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}.$$

Only the first adjacent defect is nonzero, so every defect sum is finite. Yet for $n \geq 1$,

$$U_{0,n}(t) = C_0 T_n(t) R_0 = \cos t.$$

The limiting family is not a semigroup because in general

$$\cos(s + t) \neq \cos s \cos t.$$

The failure is exactly the excursion from the reconstructed line into its orthogonal complement between two time steps.

Proposition E4.11 (Range leakage criterion)**.** *Under the contractive assumptions of corollary E4.6, define*

$$\eta_{m,n}(T) = \sup_{0 \leq t \leq T} \left\| (I - P_{m,n}) T_n(t) R_{n,m} \right\|.$$

Then

$$\rho_{m,n}(T) \leq \eta_{m,n}(T).$$

Consequently, $\eta_{m,n}(T) \to 0$ for every $T > 0$ is sufficient for the limiting family U_m to satisfy the semigroup law.

Proof. For $s, t \leq T$, the contraction assumptions give

$$\begin{aligned} &\left\| C_{m,n} T_n(s) (I - P_{m,n}) T_n(t) R_{n,m} \right\| \\ &\quad \leq \left\| (I - P_{m,n}) T_n(t) R_{n,m} \right\|. \end{aligned}$$

Take the supremum and apply theorem E4.8. □

Theorem E4.12 (Generator control of range leakage)**.** *Let $T(t) = \mathrm{e}^{tA}$ on a Banach space X, let P be a bounded projection, and set $Q = I - P$. Assume*

$$\|T(t)\| \leq M \mathrm{e}^{\omega t}$$

and

$$\left\| \mathrm{e}^{t(QAQ)|_{QX}} \right\| \leq M_Q \mathrm{e}^{\omega_Q t}.$$

Then

$$\|QT(t)P\| \leq M M_Q \|P\|^2 \|QAP\| \int_0^t \mathrm{e}^{\omega_Q (t-s) + \omega s} \, \mathrm{d}s.$$

In particular, for orthogonal projections and contraction bounds,

$$\sup_{0\le t\le T}\|QT(t)P\| \le T\,\|QAP\|.$$

Proof. For $x \in PX$, write

$$p(t) = PT(t)x, \qquad q(t) = QT(t)x.$$

The hidden component satisfies

$$q'(t) = QAQq(t) + QAPp(t), \qquad q(0) = 0.$$

Variation of constants gives

$$q(t) = \int_0^t \mathrm{e}^{(t-s)(QAQ)|_{QX}} QAPp(s)\,\mathrm{d}s.$$

Since $\|p(s)\| \le \|P\|\,\|T(s)\|\,\|x\|$ and $x = Px$, the first estimate follows. The contraction case is immediate. □

Corollary E4.13 (A usable closure condition for refinement towers)**.** *Assume the contractive summability hypothesis of corollary E4.6 holds on every bounded time interval. Suppose every $P_{m,n}$ is an orthogonal projection on a Hilbert space and $T_n(t) = \mathrm{e}^{tA_n}$ is contractive. If, for every fixed m,*

$$\big\|(I - P_{m,n})A_nP_{m,n}\big\| \longrightarrow 0,$$

then the limiting family U_m is a semigroup.

Proof. A bounded generator of a contraction semigroup on a Hilbert space is dissipative. If $Q_{m,n} = I - P_{m,n}$ and $y \in Q_{m,n}X_n$, then

$$\operatorname{Re}\langle Q_{m,n}A_nQ_{m,n}y, y\rangle = \operatorname{Re}\langle A_ny, y\rangle \le 0.$$

Hence the compressed hidden semigroup is contractive. Apply theorem E4.12 to $P_{m,n}$. Since $P_{m,n}R_{n,m} = R_{n,m}$,

$$\eta_{m,n}(T) \le T\,\big\|(I - P_{m,n})A_nP_{m,n}\big\|\,\big\|R_{n,m}\big\|.$$

The right side tends to zero. Use proposition E4.11. □

E5 Finite Markov transfer and lumping

Probability vectors are columns. A column stochastic matrix K has nonnegative entries and column sums one. A continuous time generator Q has nonnegative off-diagonal entries and column sums zero.

Theorem E5.1 (Finite Markov semigroup)**.** *If Q is a finite Markov generator, then e^{tQ} is column stochastic for every $t \ge 0$.*

Proof. Choose $a \ge \max_i(-Q_{ii})$. The matrix $Q + aI$ is nonnegative, so its exponential is nonnegative. Hence

$$\mathrm{e}^{tQ} = \mathrm{e}^{-at}\mathrm{e}^{t(Q+aI)}$$

is nonnegative. Since $1^T Q = 0$, differentiation gives

$$1^T \mathrm{e}^{tQ} = 1^T.$$

□

Let $\{B_1, \dots, B_m\}$ be a partition of the fine state set. Its aggregation matrix is

$$C_{\alpha i} = 1_{\{i \in B_\alpha\}}.$$

Definition E5.2 (Canonical lumping obstruction)**.** For a linear fine transition operator K, define

$$\mathrm{Ob}_C(K) = (CK)|_{\mathrm{Ker}\, C} \in \mathcal{B}(\mathrm{Ker}\, C, \mathbb{R}^m).$$

Theorem E5.3 (Descent through an aggregation map)**.** *Let $C : X_f \to X_c$ be a surjective linear map between finite dimensional spaces. A unique operator $\overline{K} : X_c \to X_c$ satisfying*

$$CK = \overline{K}C$$

exists if and only if

$$\mathrm{Ob}_C(K) = 0.$$

For any right inverse R of C, the only possible coarse operator is

$$\overline{K} = CKR,$$

and

$$CK - \overline{K}C = CK(I - RC).$$

Proof. If $CK = \overline{K}C$, then CK vanishes on $\mathrm{Ker}\, C$. Conversely, suppose the obstruction vanishes and define

$$\overline{K}(Cx) = CKx.$$

If $Cx = Cy$, then $x - y \in \mathrm{Ker}\, C$, so the value does not depend on the representative. Surjectivity defines $\overline{K}$ on all of X_c, and the intertwining relation follows from the definition. It also gives uniqueness. If R is a right inverse, then

$$\overline{K} = \overline{K}CR = CKR.$$

Substitution gives the defect identity. □

Corollary E5.4 (Coordinate form of the lumping obstruction)**.** *For the partition aggregation above,* $\mathrm{Ker}\, C$ *is spanned by differences $e_i - e_j$ with i and j in one source block. Also,*

$$\mathrm{Ob}_C(K)(e_i - e_j) = \left(\sum_{r \in B_\alpha} (K_{ri} - K_{rj}) \right)_{\alpha=1}^{m}.$$

Proof. Inside each block, the vectors with coordinate sum zero are spanned by differences of coordinate vectors. Applying CK gives the displayed aggregated column differences. □

Theorem E5.5 (Characterization of exact discrete lumping). *Let K be column stochastic. A coarse column stochastic matrix $\overline{K}$ satisfying*

$$CK = \overline{K}C$$

exists if and only if, for every source block B_β, every pair $i, j \in B_\beta$, and every target block B_α,

$$\sum_{r \in B_\alpha} K_{ri} = \sum_{r \in B_\alpha} K_{rj}.$$

When it exists, $\overline{K}$ is unique and

$$\overline{K}_{\alpha\beta} = \sum_{r \in B_\alpha} K_{ri}$$

for any $i \in B_\beta$.

Proof. The jth column of CK records the transition mass from state j into every target block. The jth column of $\overline{K}C$ is the column of $\overline{K}$ indexed by the source block containing j. Equality therefore holds exactly under the stated condition. The formula gives uniqueness. Summing over α gives one, so the coarse matrix is column stochastic. □

Corollary E5.6 (Continuous time lumping). *A coarse Markov generator $\overline{Q}$ satisfying*

$$CQ = \overline{Q}C$$

exists if and only if the aggregated rates

$$\sum_{r \in B_\alpha} Q_{ri}$$

depend only on the source block of i. In that case

$$C\mathrm{e}^{tQ} = \mathrm{e}^{t\overline{Q}}C$$

for every $t \geq 0$.

Proof. The characterization is the same column argument as in theorem E5.5. The off-diagonal entries of $\overline{Q}$ are nonnegative and its column sums are zero. Exact semigroup intertwining follows from proposition E3.2. □

Theorem E5.7 (Finite horizon lumping defect). *For arbitrary fine and coarse operators K and $\overline{K}$, with the fixed aggregation map C, set*

$$\Delta_K = CK - \overline{K}C.$$

Then

$$CK^n - \overline{K}^n C = \sum_{j=0}^{n-1} \overline{K}^{n-1-j} \Delta_K K^j.$$

If K and $\overline{K}$ are column stochastic, then

$$\left\| CK^n - \overline{K}^n C \right\|_{1\to 1} \leq n \left\| \Delta_K \right\|_{1\to 1}.$$

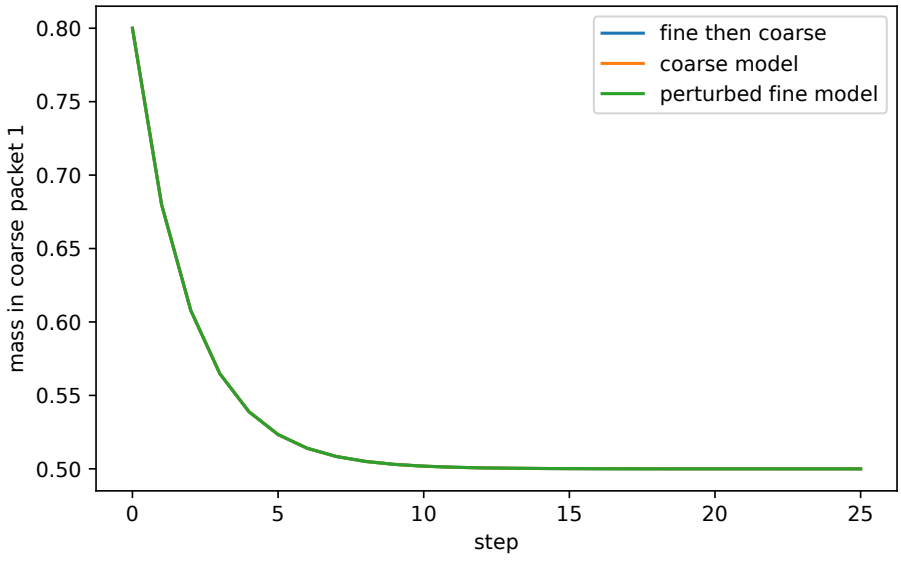


Figure E5. Exact lumping agrees at every horizon. A small violation of the block transition condition produces a visible but controlled discrepancy.

For every probability vector p, the total variation discrepancy is at most

$$\frac{n}{2}\,\|\Delta_K\|_{1\to 1}\,.$$

Proof. Sum the telescoping identities

$$\overline{K}^{n-1-j}CK^{j+1} - \overline{K}^{n-j}CK^{j} = \overline{K}^{n-1-j}\Delta_K K^{j}.$$

Stochastic matrices are contractions in the ℓ^1 norm. The last statement uses total variation equal to one half of the ℓ^1 distance. □

Corollary E5.8 (Stationary mass descends). *If $CK = \overline{K}C$ and $Kp = p$, then*

$$\overline{K}(Cp) = Cp.$$

Proof. Apply the intertwining identity to p. □

E6 Conditional expectation and martingale packets

Let $\Omega = \{1, \dots, n\}$ have strictly positive probabilities π_i. For a partition $\mathcal{Q}$, let $B(i)$ denote the block containing i. Define

$$(P_{\mathcal{Q}} f)_i = \frac{\sum_{j \in B(i)} \pi_j f_j}{\sum_{j \in B(i)} \pi_j}.$$

Equip $\mathbb{C}^n$ with

$$\langle f, g \rangle_\pi = \sum_i \pi_i f_i \overline{g_i}.$$

Theorem E6.1 (Finite conditional expectation)**.** *The operator $P_{\mathcal{Q}}$ is the orthogonal projector onto the functions that are constant on every block of $\mathcal{Q}$. In particular,*

$$P_{\mathcal{Q}}^2 = P_{\mathcal{Q}}, \qquad P_{\mathcal{Q}}^* = P_{\mathcal{Q}}.$$

Proof. One application produces a block constant function, so a second application changes nothing. Expanding both weighted inner products block by block gives

$$\langle P_{\mathcal{Q}} f, g\rangle_\pi = \sum_{B\in\mathcal{Q}} \frac{\left(\sum_{i\in B}\pi_i f_i\right)\overline{\left(\sum_{j\in B}\pi_j g_j\right)}}{\pi(B)} = \langle f, P_{\mathcal{Q}} g\rangle_\pi .$$

The range is exactly the block constant subspace. □

Theorem E6.2 (Tower law and martingale packets)**.** *Let $P_0, P_1, \dots, P_r$ come from increasingly fine partitions, so that*

$$\operatorname{Ran} P_0 \subseteq \operatorname{Ran} P_1 \subseteq \cdots \subseteq \operatorname{Ran} P_r.$$

Define

$$D_0 = P_0, \qquad D_j = P_j - P_{j-1} \quad (1 \le j \le r).$$

Then

$$P_i P_j = P_{\min(i,j)}, \qquad D_i D_j = \delta_{ij} D_j, \qquad P_r = \sum_{j=0}^{r} D_j.$$

Every D_j is an orthogonal projector, and

$$\|P_r f\|_\pi^2 = \sum_{j=0}^{r} \|D_j f\|_\pi^2$$

and

$$\|f\|_\pi^2 = \sum_{j=0}^{r} \|D_j f\|_\pi^2 + \|f - P_r f\|_\pi^2 .$$

Proof. A coarse average of a fine average is the coarse average. A fine average leaves a coarse constant function unchanged. Hence $P_i P_j = P_{\min(i,j)}$. Expanding products of the differences gives the orthogonality relations. The sum telescopes to P_r. Orthogonal Pythagorean identities give the norm formulas. □

Corollary E6.3 (Law of total variance)**.** *For a real valued f, let $P_0 f = \mathbb{E}_\pi[f]1$. Then*

$$\operatorname{Var}_\pi(f) = \operatorname{Var}_\pi(P_{\mathcal{Q}} f) + \|f - P_{\mathcal{Q}} f\|_\pi^2 .$$

Proof. Subtract the mean and apply the orthogonal decomposition into the range of $P_{\mathcal{Q}}$ and its orthogonal complement. □

Standard Fact E6.4 (General conditional expectations). On a probability space, conditional expectation onto a sub sigma algebra is the orthogonal projector in L^2. Nested sigma algebras satisfy the same

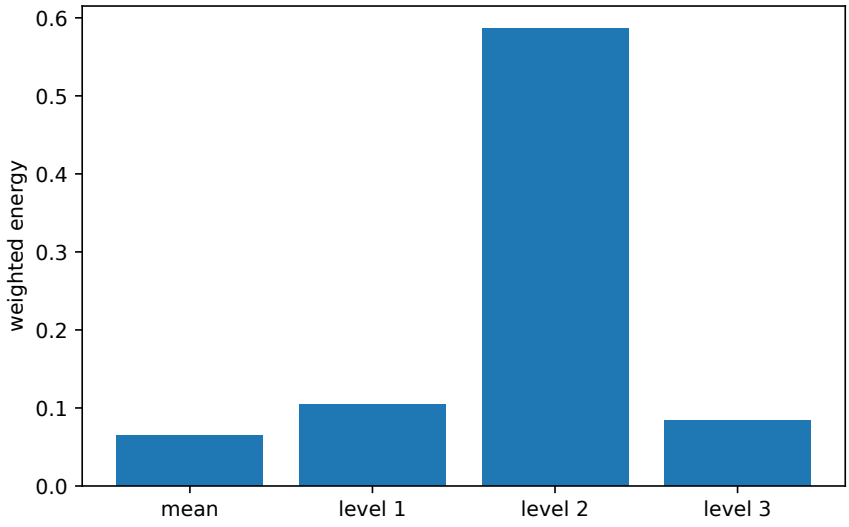


Figure E6. Energy carried by the mean packet and three successive martingale difference packets in a finite weighted example.

tower law. Under the usual hypotheses, martingale convergence gives a limit in L^2 and almost surely. Standard treatments include [Wil91].

E7 Gaussian transfer and Hermite packets

Let γ be the standard Gaussian probability measure on $\mathbb{R}$. For $t \geq 0$, define the Ornstein–Uhlenbeck transfer

$$(\mathcal{O}_t f)(x) = \int_{\mathbb{R}} f\left(\mathrm{e}^{-t}x + \sqrt{1-\mathrm{e}^{-2t}}\, y\right) \gamma(\mathrm{d}y).$$

Let H_n be the probabilists' Hermite polynomial defined by

$$\exp\left(ux - \frac{u^2}{2}\right) = \sum_{n=0}^{\infty} H_n(x)\frac{u^n}{n!},$$

and set $h_n = H_n/\sqrt{n!}$.

Theorem E7.1 (Ornstein–Uhlenbeck semigroup). *The operators $(\mathcal{O}_t)_{t\geq 0}$ form a positive Markov contraction semigroup on $L^p(\gamma)$ for every $1 \leq p \leq \infty$. On $L^2(\gamma)$, the semigroup is self-adjoint and*

$$\mathcal{O}_t h_n = \mathrm{e}^{-nt} h_n.$$

The family $(h_n)_{n\geq 0}$ is an orthonormal basis of $L^2(\gamma)$.

Proof. Positivity and preservation of constants follow from the integral formula. If X and Y are independent standard Gaussian variables, then

$$\mathrm{e}^{-s}\left(\mathrm{e}^{-t}x + \sqrt{1-\mathrm{e}^{-2t}}X\right) + \sqrt{1-\mathrm{e}^{-2s}}Y$$

has the same distribution as

$$\mathrm{e}^{-(s+t)}x + \sqrt{1-\mathrm{e}^{-2(s+t)}}Z$$

with Z standard Gaussian. This proves the semigroup law. Jensen's inequality and invariance of γ give the L^p contraction. The joint Gaussian pair

$$\left(X, \mathrm{e}^{-t}X + \sqrt{1-\mathrm{e}^{-2t}}Y\right)$$

is symmetric under interchange, which proves self-adjointness.

Apply $\mathcal{O}_t$ to the Hermite generating function. Gaussian integration gives

$$\begin{aligned}\mathcal{O}_t \exp\left(ux - \frac{u^2}{2}\right) &= \exp\left(\mathrm{e}^{-t}ux - \frac{\mathrm{e}^{-2t}u^2}{2}\right)\\ &= \sum_{n=0}^{\infty} \mathrm{e}^{-nt} H_n(x)\frac{u^n}{n!}.\end{aligned}$$

Comparison of coefficients proves the eigenvalue formula. The same generating function shows

$$\int H_m H_n \,\mathrm{d}\gamma = n!\delta_{mn}.$$

The Hermite polynomials span all polynomials, and polynomials are dense in $L^2(\gamma)$. Hence the normalized family is an orthonormal basis. See [Jan97] for the general Gaussian Hilbert space setting. □

Corollary E7.2 (Hermite packet reconstruction)**.** *Let P_n be the orthogonal projection onto $\mathbb{C}h_n$. Then*

$$I = \sum_{n=0}^{\infty} P_n$$

in the strong operator topology and

$$\mathcal{O}_t = \sum_{n=0}^{\infty} \mathrm{e}^{-nt} P_n.$$

For $f \in L^2(\gamma)$,

$$\|\mathcal{O}_t f\|_2^2 = \sum_{n=0}^{\infty} \mathrm{e}^{-2nt}\,\|P_n f\|_2^2 .$$

Proof. Apply Parseval's identity and the eigenvalue formula in theorem E7.1. □

Proposition E7.3 (Generator and Dirichlet form)**.** *On the polynomial subspace, the infinitesimal generator of $(\mathcal{O}_t)$ is the differential operator*

$$\mathcal{L}f = f'' - xf'.$$

It satisfies

$$\mathcal{L}h_n = -nh_n.$$

For every polynomial f,

$$\langle -\mathcal{L}f, f\rangle_{L^2(\gamma)} = \int_{\mathbb{R}} |f'(x)|^2 \,\gamma(\mathrm{d}x).$$

Proof. Write

$$\rho_t = \mathrm{e}^{-t} = 1 - t + O(t^2), \qquad \sigma_t^2 = 1 - \mathrm{e}^{-2t} = 2t + O(t^2).$$

For a polynomial f, Taylor expansion at x and Gaussian integration give

$$\begin{aligned}\mathcal{O}_t f(x) &= f(x) + f'(x)(\rho_t - 1)x + \frac{1}{2} f''(x)\sigma_t^2 + o(t) \\ &= f(x) + t(f''(x) - xf'(x)) + o(t).\end{aligned}$$

Every omitted term is a polynomial in the Gaussian variable multiplied by a quantity that is $o(t)$, so the expansion may be integrated term by term. Therefore

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{O}_t f\Big|_{t=0} = f'' - xf'.$$

The Hermite eigenvalue identity also follows by differentiating $\mathcal{O}_t h_n = \mathrm{e}^{-nt} h_n$ at zero. Gaussian integration by parts gives

$$\int_{\mathbb{R}} f'(x)\overline{g(x)}\gamma(\,\mathrm{d}x) = \int_{\mathbb{R}} f(x)\overline{xg(x) - g'(x)}\gamma(\,\mathrm{d}x).$$

Set $g = f'$ and rearrange. □

Corollary E7.4 (Gaussian spectral gap)**.** *If $f \in L^2(\gamma)$ has mean zero, then*

$$\|\mathcal{O}_t f\|_2 \le \mathrm{e}^{-t}\,\|f\|_2 .$$

For every polynomial f,

$$\mathrm{Var}_\gamma(f) \le \int_{\mathbb{R}} |f'(x)|^2\,\gamma(\,\mathrm{d}x).$$

Proof. A mean zero function has no h_0 component, so every surviving Hermite multiplier is at most e^{-t}. For a polynomial

$$f - \mathbb{E}_\gamma f = \sum_{n\ge 1} c_n h_n.$$

Then

$$\mathrm{Var}_\gamma(f) = \sum_{n\ge 1} |c_n|^2 \le \sum_{n\ge 1} n\,|c_n|^2 = \langle -\mathcal{L}f, f\rangle .$$

Apply proposition E7.3. □

Corollary E7.5 (Fractional Hermite transfer)**.** *Let $-\mathcal{L}$ be the nonnegative Ornstein–Uhlenbeck number operator with*

$$-\mathcal{L}h_n = nh_n.$$

For every $\alpha > 0$,

$$\mathrm{e}^{-t(-\mathcal{L})^\alpha} f = \sum_{n=0}^{\infty} \mathrm{e}^{-tn^\alpha} P_n f.$$

This is a contraction semigroup on $L^2(\gamma)$.

Proof. The spectral theorem applies to the orthonormal Hermite basis. Every multiplier has modulus at most one and obeys the scalar semigroup law. □

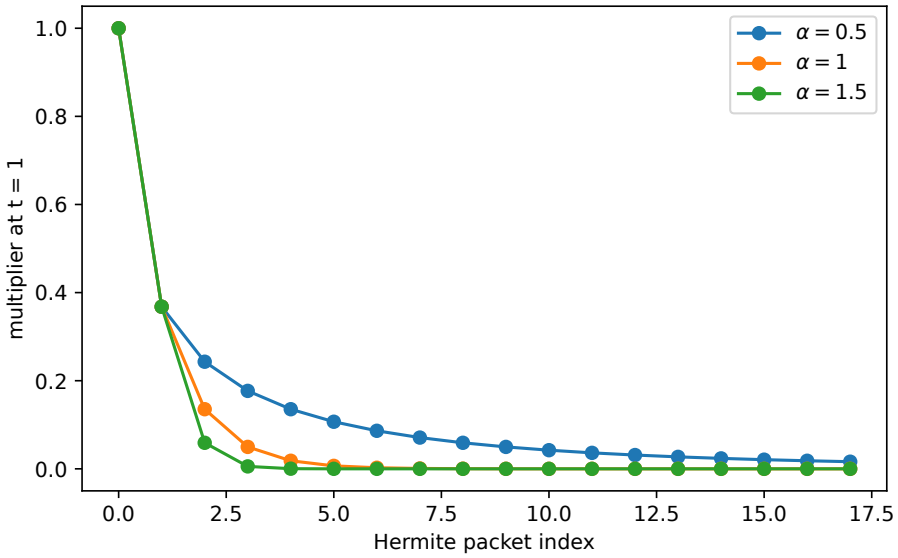


Figure E7. Hermite packet multipliers for the Ornstein–Uhlenbeck semigroup and two fractional powers of its number operator.

E8 Bayesian transfer and evidence packets

Let

$$\Delta_{n-1} = \left\{ p \in \mathbb{R}^n : p_i \geq 0, \quad \sum_i p_i = 1 \right\}$$

be the probability simplex. Let $\lambda \in [0, \infty)^n$ have positive evidence

$$Z_\lambda(p) = \sum_i \lambda_i p_i.$$

The Bayesian transfer is

$$\mathcal{B}_\lambda(p) = \frac{\lambda \odot p}{Z_\lambda(p)}.$$

Proposition E8.1 (Scale invariance and composition)**.** *For every $c > 0$,*

$$\mathcal{B}_{c\lambda} = \mathcal{B}_\lambda.$$

If all relevant evidences are positive, then

$$\mathcal{B}_\mu(\mathcal{B}_\lambda(p)) = \mathcal{B}_{\lambda \odot \mu}(p).$$

Proof. A common likelihood scale cancels between the numerator and denominator. In the second identity, the first evidence appears in both the numerator and the denominator of the second update and cancels. □

Let $I = I_1 \sqcup \cdots \sqcup I_m$ be a partition. Define packet evidences

$$Z_\alpha = \sum_{i \in I_\alpha} \lambda_i p_i.$$

For $Z_\alpha > 0$, define

$$m_\alpha = \frac{Z_\alpha}{\sum_\beta Z_\beta}, \qquad q_{\alpha,i} = \frac{\lambda_i p_i}{Z_\alpha} \quad (i \in I_\alpha).$$

Packets with zero evidence receive posterior mass zero.

Theorem E8.2 (Bayesian packet reconstruction). *The posterior is the disjoint packet synthesis*

$$\mathcal{B}_\lambda(p) = \sum_{\alpha : Z_\alpha > 0} m_\alpha q_\alpha.$$

The local shape q_α depends only on the prior and likelihood inside packet I_α. Coupling between different packets occurs through the normalization of the packet evidences.

Proof. For $i \in I_\alpha$,

$$m_\alpha q_{\alpha,i} = \frac{Z_\alpha}{\sum_\beta Z_\beta} \frac{\lambda_i p_i}{Z_\alpha} = \frac{\lambda_i p_i}{Z_\lambda(p)}.$$

The packet supports are disjoint. □

Theorem E8.3 (Likelihood perturbation). *Let $\lambda, \mu \geq 0$ have positive evidences under a fixed prior p. Set*

$$d_p(\lambda, \mu) = \sum_i |\lambda_i - \mu_i| \, p_i.$$

Then

$$\left\| \mathcal{B}_\lambda(p) - \mathcal{B}_\mu(p) \right\|_1 \leq \frac{2 d_p(\lambda, \mu)}{\max\{Z_\lambda(p), Z_\mu(p)\}}.$$

Proof. Add and subtract $\mu \odot p / Z_\lambda(p)$. This gives

$$\left\| \mathcal{B}_\lambda(p) - \mathcal{B}_\mu(p) \right\|_1 \leq \frac{2 d_p(\lambda, \mu)}{Z_\lambda(p)}.$$

Interchanging λ and μ gives the same estimate with $Z_\mu(p)$. Take the smaller of the two upper bounds. □

Theorem E8.4 (Stability with respect to the prior). *Assume*

$$0 < a \leq \lambda_i \leq b$$

for every i. Then

$$\left\| \mathcal{B}_\lambda(p) - \mathcal{B}_\lambda(q) \right\|_1 \leq \frac{2b}{a} \left\| p - q \right\|_1$$

for all priors $p, q \in \Delta_{n-1}$.

Proof. The evidences are at least a. Add and subtract $\lambda \odot q / Z_\lambda(p)$. The numerator difference is at most $b \left\| p - q \right\|_1$, and the evidence difference has the same bound. The two terms give the stated constant. □

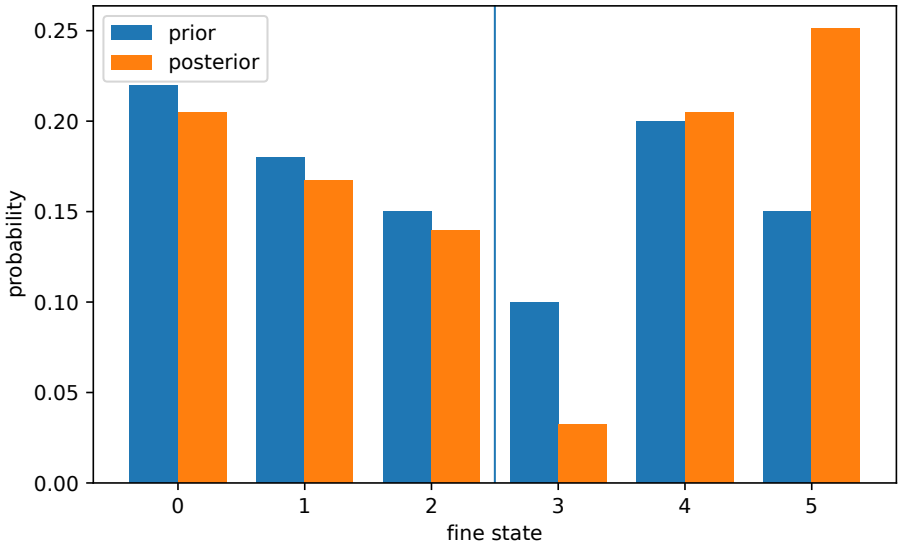


Figure E8. A prior and posterior on two coarse packets. The likelihood is constant in the first packet and varies inside the second, so only the second local conditional distribution changes.

E9 Bayesian coarsening and finite sufficiency

A fixed coarse description will be called sufficient when it reproduces the coarse posterior masses for every positive prior and leaves the conditional distribution inside every packet unchanged. This excludes a coarse rule that hides an unrecorded local update.

Let C send a fine probability vector to its block masses

$$(Cp)_\alpha = \sum_{i \in I_\alpha} p_i.$$

Assume first that every block mass is positive. Write

$$r_{\alpha,i} = \frac{p_i}{(Cp)_\alpha}$$

for the prior conditional distribution inside packet I_α. Define the prior dependent packet average likelihood

$$\overline{\lambda}_\alpha(p) = \sum_{i \in I_\alpha} \lambda_i r_{\alpha,i}.$$

Theorem E9.1 (Exact two stage Bayesian update). *For every strictly positive prior,*

$$C\mathcal{B}_\lambda(p) = \mathcal{B}_{\overline{\lambda}(p)}(Cp).$$

For every packet with $\overline{\lambda}_\alpha(p) > 0$, the posterior conditional distribution is

$$q_{\alpha,i} = \frac{\lambda_i r_{\alpha,i}}{\overline{\lambda}_\alpha(p)}.$$

A packet with zero average likelihood has zero posterior mass. Thus a fine Bayesian update consists of a coarse update by packet average likelihoods followed by independent local updates inside the packets that survive.

Proof. The posterior mass of packet I_α is

$$\frac{\sum_{i\in I_\alpha}\lambda_i p_i}{\sum_j \lambda_j p_j} = \frac{(Cp)_\alpha \overline{\lambda}_\alpha(p)}{\sum_\beta (Cp)_\beta \overline{\lambda}_\beta(p)}.$$

This is the coarse Bayesian formula. Dividing the fine posterior coordinates by the packet mass gives the local formula. □

Theorem E9.2 (Characterization of fixed coarse sufficiency)**.** *Assume $\lambda_i > 0$ for every state. The following conditions are equivalent.*

1. *There is a positive coarse likelihood $\ell \in (0,\infty)^m$ such that*

$$C\mathcal{B}_\lambda(p) = \mathcal{B}_\ell(Cp)$$

   for every strictly positive prior p, and the posterior conditional distribution inside every packet equals its prior conditional distribution.

2. *The likelihood is constant on every packet.*

When these conditions hold, ℓ_α can be chosen as the common value of λ on I_α, and it is unique up to a common positive scale.

Proof. If λ is constant on each packet, theorem E9.1 has packet averages independent of the prior and its local posterior formula gives $q_\alpha = r_\alpha$. This proves the first condition.

Conversely, fix any strictly positive prior. Equality of the local posterior and prior conditionals gives

$$\frac{\lambda_i r_{\alpha,i}}{\overline{\lambda}_\alpha(p)} = r_{\alpha,i}$$

for every $i \in I_\alpha$. Positivity of the conditional prior implies

$$\lambda_i = \overline{\lambda}_\alpha(p)$$

for every state in the packet. Thus the likelihood is constant on every packet. If two positive coarse likelihoods define the same update for every positive coarse prior, comparison of any two posterior odds shows that their coordinate ratios agree. Hence they differ only by a common positive factor. □

Theorem E9.3 (Coarsest fixed sufficient partition)**.** *Let $\mathcal{L}$ be a nonempty family of strictly positive likelihood vectors. Define*

$$i \sim_{\mathcal{L}} j \quad\Longleftrightarrow\quad \lambda_i = \lambda_j \quad \text{for every } \lambda \in \mathcal{L}.$$

Let $\mathcal{Q}_{\mathcal{L}}$ be the partition into equivalence classes. A partition $\mathcal{Q}$ gives a fixed coarse sufficient description in the sense of theorem E9.2 for every $\lambda \in \mathcal{L}$ if and only if every block of $\mathcal{Q}$ is contained in a block of $\mathcal{Q}_{\mathcal{L}}$. Hence $\mathcal{Q}_{\mathcal{L}}$ is the unique coarsest partition with this property.

Proof. Apply theorem E9.2 to every likelihood in the family. A fixed sufficient description exists exactly when each likelihood is constant on every block of $\mathcal{Q}$. This is equivalent to every block of $\mathcal{Q}$ being contained in one common level set of the family. The equivalence class partition is itself admissible, and every other admissible partition refines it. □

Definition E9.4 (Within packet information loss)**.** For a prior p, define

$$\eta_i(p) = \lambda_i - \overline{\lambda}_\alpha(p) \quad \text{when } i \in I_\alpha.$$

Then

$$\sum_{i \in I_\alpha} r_{\alpha,i}\eta_i(p) = 0.$$

Proposition E9.5 (Exact local distortion)**.** *For every packet with positive mass and evidence,*

$$q_{\alpha,i} - r_{\alpha,i} = \frac{r_{\alpha,i}\eta_i(p)}{\overline{\lambda}_\alpha(p)}.$$

Consequently,

$$\|q_\alpha - r_\alpha\|_1 = \frac{\sum_{i \in I_\alpha} r_{\alpha,i}\,|\eta_i(p)|}{\overline{\lambda}_\alpha(p)}.$$

If λ is positive on the packet, then

$$\|q_\alpha - r_\alpha\|_1 \le \frac{\operatorname{osc}_{I_\alpha}(\lambda)}{\min_{i \in I_\alpha} \lambda_i}.$$

Proof. Subtract $r_{\alpha,i}$ from the local posterior formula in theorem E9.1. The norm identity follows because the factor multiplying η_i is nonnegative. The last estimate uses

$$\sum_i r_{\alpha,i}\,|\eta_i| \le \operatorname{osc}_{I_\alpha}(\lambda)$$

and $\overline{\lambda}_\alpha(p) \ge \min_{I_\alpha} \lambda$. □

Theorem E9.6 (Approximate coarse Bayesian rule)**.** *Choose one positive number ℓ_α for every packet and define the block constant fine likelihood*

$$\tilde{\lambda}_i = \ell_\alpha \quad (i \in I_\alpha).$$

Then

$$\|C\mathcal{B}_\lambda(p) - \mathcal{B}_\ell(Cp)\|_1 \le \frac{2\sum_i p_i \left|\lambda_i - \tilde{\lambda}_i\right|}{\max\{Z_\lambda(p), Z_{\tilde{\lambda}}(p)\}}.$$

Proof. Block constancy gives

$$C\mathcal{B}_{\tilde{\lambda}}(p) = \mathcal{B}_\ell(Cp).$$

Aggregation is an ℓ^1 contraction. Apply theorem E8.3 to the two fine posteriors and then aggregate. □

Definition E9.7 (Relative information)**.** For probability vectors q and p, define

$$D(q\|p) = \sum_i q_i \log \frac{q_i}{p_i}.$$

The convention is $0 \log 0 = 0$. A term with $q_i > 0$ and $p_i = 0$ has value $+\infty$.

Theorem E9.8 (Exact information decomposition)**.** *Let p and q be probability vectors on a finite partition $(I_\alpha)_\alpha$, and assume $q_i = 0$ whenever $p_i = 0$. Let*

$$m = Cp, \qquad n = Cq,$$

and let p_α and q_α be the conditional distributions inside packet I_α whenever the corresponding mass is positive. Then

$$D(q\|p) = D(n\|m) + \sum_{\alpha : n_\alpha > 0} n_\alpha D(q_\alpha \| p_\alpha).$$

The first term is the information retained by the coarse masses. The second term is the exact information carried by changes inside the packets.

Proof. For $i \in I_\alpha$, write

$$p_i = m_\alpha p_{\alpha,i}, \qquad q_i = n_\alpha q_{\alpha,i}.$$

Then

$$\log \frac{q_i}{p_i} = \log \frac{n_\alpha}{m_\alpha} + \log \frac{q_{\alpha,i}}{p_{\alpha,i}}.$$

Multiply by q_i, sum first inside each packet, and then sum over the packets. This is the finite relative entropy chain rule. Standard information theoretic treatments include [CT06]. □

Corollary E9.9 (Coarse data processing and equality)**.** *Under the assumptions of theorem E9.8,*

$$D(Cq\|Cp) \le D(q\|p).$$

Equality holds if and only if

$$q_\alpha = p_\alpha$$

for every packet with positive posterior mass.

Proof. Every conditional relative information term in theorem E9.8 is nonnegative. It vanishes exactly when the two conditional distributions agree. □

Corollary E9.10 (Information created by a Bayesian update)**.** *Let $q = \mathcal{B}_\lambda(p)$ with positive likelihood and positive prior. Then*

$$D(q\|p) = \sum_i q_i \log \lambda_i - \log Z_\lambda(p).$$

Inside a packet,

$$D(q_\alpha \| r_\alpha) = \sum_{i \in I_\alpha} q_{\alpha,i} \log \frac{\lambda_i}{\overline{\lambda}_\alpha(p)}.$$

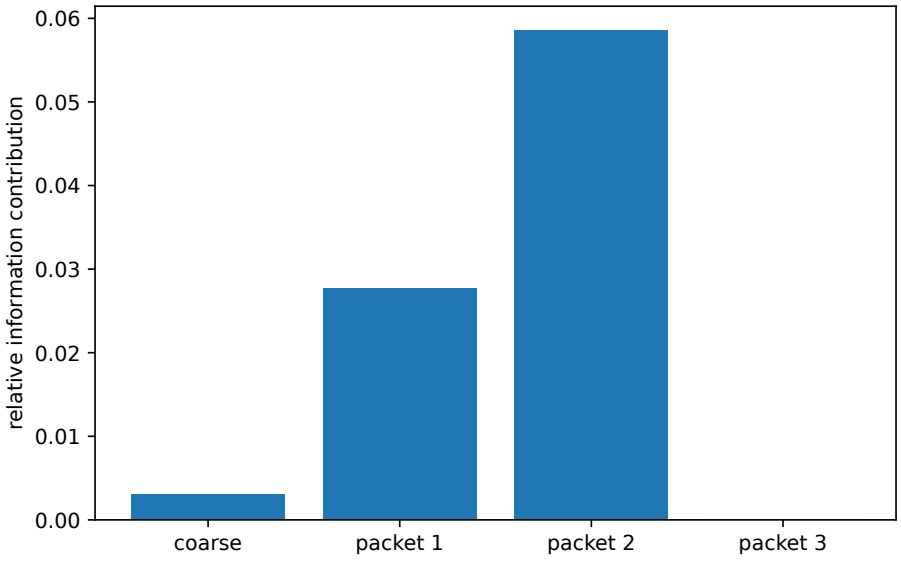


Figure E9. Relative information separated into a coarse packet contribution and conditional contributions inside the packets.

Hence

$$D(q_\alpha \| r_\alpha) \le \log \frac{\max_{i \in I_\alpha} \lambda_i}{\min_{i \in I_\alpha} \lambda_i}.$$

Proof. The posterior ratio is

$$\frac{q_i}{p_i} = \frac{\lambda_i}{Z_\lambda(p)}.$$

The local ratio is

$$\frac{q_{\alpha,i}}{r_{\alpha,i}} = \frac{\lambda_i}{\overline{\lambda}_\alpha(p)}.$$

Insert these identities into the definition of relative entropy. The final estimate follows because both λ_i and the packet average lie between the packet minimum and maximum. □

Corollary E9.11 (Information criterion for local sufficiency)**.** *For a fixed positive likelihood, the within packet term in theorem E9.8 vanishes for every positive prior if and only if the likelihood is constant on every packet.*

Proof. If the likelihood is block constant, every local posterior equals its local prior. Conversely, choose any positive prior. Vanishing relative entropy gives $q_\alpha = r_\alpha$. The local posterior formula then forces all likelihood values in that packet to equal their packet average. Repeat for every packet. □

Remark E9.12. The exact coarse likelihood in theorem E9.1 depends on the conditional prior inside each packet. Theorem E9.2 identifies when that dependence disappears. Theorem E9.8 measures what is lost when only the coarse posterior masses are retained.

E10 Bayesian flows and hidden Markov recursion

Fix a score vector $g \in \mathbb{R}^n$. For $t \geq 0$, define

$$\Phi_t(p)_i = \frac{\mathrm{e}^{tg_i} p_i}{\sum_j \mathrm{e}^{tg_j} p_j}.$$

Theorem E10.1 (Bayesian flow and replicator equation). *The maps Φ_t form a nonlinear semigroup:*

$$\Phi_{s+t} = \Phi_s \circ \Phi_t.$$

Every coordinate satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t} p_i(t) = (g_i - \langle g, p(t) \rangle) p_i(t).$$

The support of $p(t)$ is independent of t, and for any two coordinates in the support,

$$\frac{p_i(t)}{p_j(t)} = \mathrm{e}^{t(g_i - g_j)} \frac{p_i(0)}{p_j(0)}.$$

Proof. The semigroup law follows from multiplication of the likelihoods e^{sg} and e^{tg}. Differentiating the normalized formula gives the replicator equation. A coordinate that starts at zero remains zero, while a positive coordinate remains positive. Dividing two explicit coordinate formulas gives the odds identity. □

Theorem E10.2 (Evidence cumulants and score monotonicity). *Set*

$$\psi(t) = \log \left(\sum_i \mathrm{e}^{tg_i} p_i(0) \right).$$

Then

$$\psi'(t) = \langle g, p(t) \rangle, \qquad \psi''(t) = \mathrm{Var}_{p(t)}(g) \geq 0.$$

Consequently,

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle g, p(t) \rangle = \mathrm{Var}_{p(t)}(g).$$

Proof. Differentiate the log normalizing constant once and twice. The second derivative is the centered second moment under the tilted distribution. □

Corollary E10.3 (Concentration on the maximal score packet). *Let*

$$G = \max\{g_i : p_i(0) > 0\}$$

and let M be the set of supported indices with score G. If M is the full support, then the conclusion below is immediate. Otherwise, suppose

$$\delta = \min\{G - g_i : p_i(0) > 0, \quad i \notin M\} > 0.$$

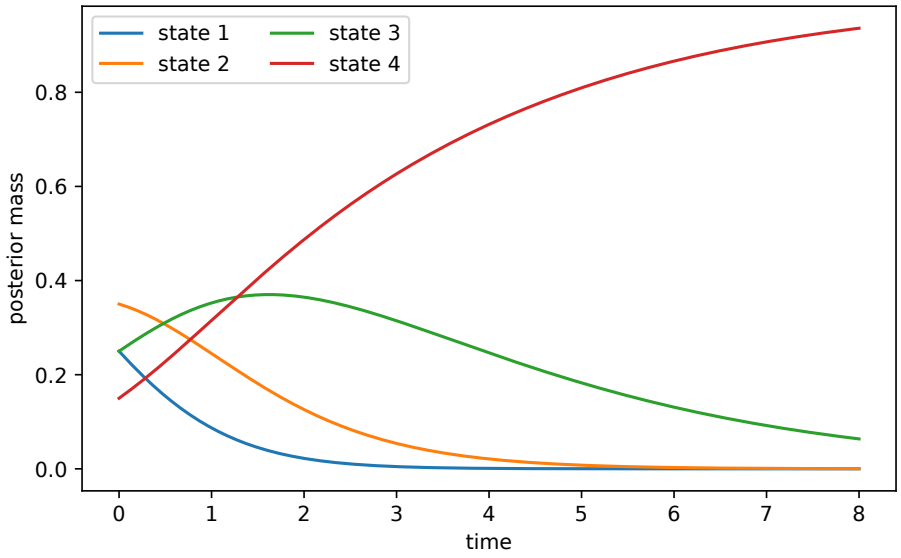


Figure E10. A finite Bayesian flow. Mass concentrates on the states with the largest score, while every relative odds ratio follows an exact exponential law.

Then

$$1 - \sum_{i \in M} p_i(t) \le \frac{1 - p_M(0)}{p_M(0)} \mathrm{e}^{-\delta t},$$

where $p_M(0) = \sum_{i \in M} p_i(0)$.

Proof. Factor e^{tG} from the normalizing constant. The total weight outside M is at most $(1 - p_M(0))\mathrm{e}^{-\delta t}$, while the weight on M is $p_M(0)$. □

E10.1 Prediction and update

Let K be column stochastic and let $\lambda^{(r)}$ be the likelihood of the rth observation. The hidden Markov filter is

$$\hat{p}_r = K p_{r-1}, \qquad p_r = \mathcal{B}_{\lambda^{(r)}}(\hat{p}_r).$$

A standard account of finite hidden Markov models appears in [Rab89].

Proposition E10.4 (Simplex preservation)**.** *Suppose p_{r-1} is a probability vector and the new evidence is positive. Prediction produces the probability vector $\hat{p}_r$, and normalization produces the probability vector p_r.*

Proof. Column stochasticity preserves nonnegativity and total mass. Bayesian normalization then preserves the same properties. □

Theorem E10.5 (One step filter stability)**.** *Let K and $\widetilde{K}$ be column stochastic. Assume*

$$0 < a \le \lambda_i \le b.$$

Then

$$\left\|\mathcal{B}_\lambda(Kp) - \mathcal{B}_\lambda(\widetilde{K}q)\right\|_1 \le \frac{2b}{a}\left(\|p-q\|_1 + \left\|K-\widetilde{K}\right\|_{1\to 1}\right).$$

Proof. Apply theorem E8.4 to Kp and $\widetilde{K}q$. Then use

$$\left\|Kp - \widetilde{K}q\right\|_1 \le \|K(p-q)\|_1 + \left\|(K-\widetilde{K})q\right\|_1$$

and the stochastic contraction property. □

Example E10.6 (Beta binomial transfer)**.** A beta prior with parameters (a, b) and binomial data containing s successes in n trials gives the posterior parameters

$$(a+s, b+n-s).$$

The update is additive in the two exponent coordinates. This is a finite dimensional conjugate realization of likelihood composition. Standard Bayesian references include [BS94].

E11 Fractional functional calculus

Let H be a finite dimensional Hilbert space and let $L = L^* \ge 0$. Write its spectral resolution as

$$L = \sum_j \lambda_j P_j.$$

For $\alpha > 0$, define

$$L^\alpha = \sum_j \lambda_j^\alpha P_j$$

and

$$T_\alpha(t) = \mathrm{e}^{-tL^\alpha} = \sum_j \mathrm{e}^{-t\lambda_j^\alpha} P_j.$$

Theorem E11.1 (Fractional packet calculus)**.** *For $\alpha, \beta > 0$,*

$$L^\alpha L^\beta = L^{\alpha+\beta}.$$

For $s, t \ge 0$,

$$T_\alpha(s)T_\alpha(t) = T_\alpha(s+t).$$

Every spectral packet P_jH reduces L^α and $T_\alpha(t)$.

Proof. The claims hold on each spectral packet and the projectors are mutually orthogonal. □

Theorem E11.2 (Balakrishnan formula in finite dimension)**.** *For $0 < \alpha < 1$,*

$$L^\alpha = \frac{\sin(\pi\alpha)}{\pi}\int_0^\infty s^{\alpha-1} L(sI+L)^{-1}\,\mathrm{d}s.$$

The integral converges in operator norm.

Proof. It is enough to prove the formula on an eigenvector with eigenvalue $x \geq 0$. For $x > 0$, the substitution $s = xu$ gives

$$\frac{\sin(\pi\alpha)}{\pi} \int_0^\infty s^{\alpha-1} \frac{x}{s+x}\, \mathrm{d}s = x^\alpha \frac{\sin(\pi\alpha)}{\pi} \int_0^\infty \frac{u^{\alpha-1}}{1+u}\, \mathrm{d}u = x^\alpha.$$

The value at $x = 0$ is zero. There are finitely many eigenvalues, so the operator integral converges. This is the finite spectral form of the fractional power construction in [Bal60]. □

Theorem E11.3 (Exact fractional naturality)**.** *Let L_f and L_c be finite dimensional positive self-adjoint operators and let C satisfy*

$$CL_f = L_c C.$$

Then, for every $\alpha > 0$ and $t \geq 0$,

$$CL_f^\alpha = L_c^\alpha C$$

and

$$C\mathrm{e}^{-tL_f^\alpha} = \mathrm{e}^{-tL_c^\alpha} C.$$

Proof. The union of the two finite spectra is a finite set. There is a polynomial that agrees with x^α on that set, and another polynomial that agrees with e^{-tx^α}. Exact intertwining passes to every polynomial by induction. □

Theorem E11.4 (Fractional defect bound)**.** *Let $0 < \alpha < 1$. Suppose*

$$L_f \geq mI, \qquad L_c \geq mI$$

for some $m > 0$. Define

$$\Delta_L = CL_f - L_c C.$$

Then

$$\left\| CL_f^\alpha - L_c^\alpha C \right\| \leq \alpha m^{\alpha-1} \left\| \Delta_L \right\|.$$

Consequently,

$$\left\| C\mathrm{e}^{-tL_f^\alpha} - \mathrm{e}^{-tL_c^\alpha} C \right\| \leq \alpha t m^{\alpha-1} \left\| \Delta_L \right\|.$$

Proof. Set

$$R_f(s) = (sI + L_f)^{-1}, \qquad R_c(s) = (sI + L_c)^{-1}.$$

The resolvent identity gives

$$CR_f(s) - R_c(s)C = -R_c(s)\Delta_L R_f(s).$$

Since $LR = I - sR$,

$$CL_f R_f(s) - L_c R_c(s) C = sR_c(s)\Delta_L R_f(s).$$

Insert this into theorem E11.2. The spectral lower bound gives

$$\left\|R_f(s)\right\|, \|R_c(s)\| \le \frac{1}{s+m}.$$

Therefore

$$\begin{aligned}\left\|CL_f^\alpha - L_c^\alpha C\right\| &\le \frac{\sin(\pi\alpha)}{\pi}\|\Delta_L\| \int_0^\infty \frac{s^\alpha}{(s+m)^2}\,\mathrm{d}s \\ &= \alpha m^{\alpha-1}\|\Delta_L\|.\end{aligned}$$

The semigroup estimate follows from theorem E3.1, applied to the contraction generators $-L_f^\alpha$ and $-L_c^\alpha$. □

Remark E11.5. The factor $m^{\alpha-1}$ records the singular behavior of x^α near zero. Without a common positive lower bound, a uniform Lipschitz estimate of this form cannot be expected.

Standard Fact E11.6 (Stable subordination). Let $0 < \alpha < 1$ and $t > 0$. There is a probability measure $\eta_{\alpha,t}$ on $[0,\infty)$ such that

$$\int_0^\infty \mathrm{e}^{-s\lambda}\eta_{\alpha,t}(\,\mathrm{d}s) = \mathrm{e}^{-t\lambda^\alpha}$$

for every $\lambda \ge 0$. Consequently, for a finite dimensional positive self-adjoint operator,

$$\mathrm{e}^{-tL^\alpha} = \int_0^\infty \mathrm{e}^{-sL}\eta_{\alpha,t}(\,\mathrm{d}s)$$

in operator norm. The measures satisfy the convolution law

$$\eta_{\alpha,s} * \eta_{\alpha,t} = \eta_{\alpha,s+t}.$$

The existence of $\eta_{\alpha,t}$ follows from complete monotonicity of $\mathrm{e}^{-t\lambda^\alpha}$. The operator identity is obtained by applying the scalar Laplace transform on every spectral packet. This is the finite spectral form of Bochner subordination. See [SSV12].

Theorem E11.7 (Poisson subordination)**.** *For $t > 0$,*

$$\mathrm{e}^{-t\sqrt{L}} = \frac{t}{2\sqrt{\pi}}\int_0^\infty s^{-3/2}\mathrm{e}^{-t^2/(4s)}\mathrm{e}^{-sL}\,\mathrm{d}s.$$

The integral converges in operator norm.

Proof. On a spectral packet with eigenvalue λ, the formula reduces to

$$\mathrm{e}^{-t\sqrt{\lambda}} = \frac{t}{2\sqrt{\pi}}\int_0^\infty s^{-3/2}\mathrm{e}^{-t^2/(4s)-\lambda s}\,\mathrm{d}s.$$

This scalar identity follows from the Gaussian integral. Recombine the finitely many spectral packets. □

Remark E11.8. For sectorial and nonnormal operators, fractional powers require a resolvent or contour calculus together with domain control. The finite self-adjoint theory proved here does not remove those analytic requirements. A broader account appears in [SSV12].

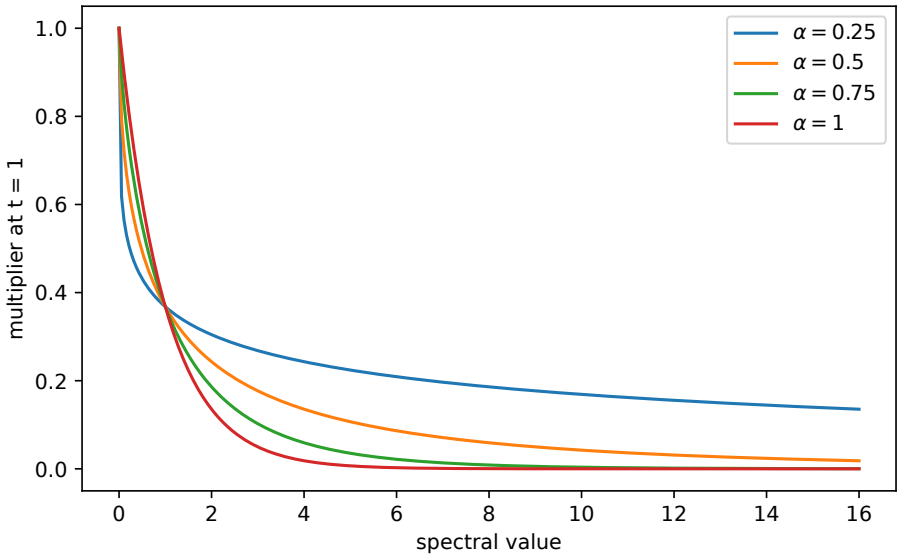


Figure E11. Fractional semigroup multipliers at time one. Smaller exponents preserve more of the high spectral range.

E12 Green operators and finite difference transfer

Consider

$$-u'' = f \quad \text{on } (0,1), \qquad u(0) = u(1) = 0.$$

The continuum Green kernel is

$$G(x,y) = \min(x,y) - xy.$$

Thus

$$u(x) = \int_0^1 G(x,y) f(y)\,\mathrm{d}y.$$

For n interior nodes, let

$$h = \frac{1}{n+1}, \qquad x_i = ih,$$

and define

$$L_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

Theorem E12.1 (Exact nodal Green reconstruction). *For $1 \le i, j \le n$,*

$$\frac{1}{h}(L_h^{-1})_{ij} = G(x_i, x_j).$$

Proof. The inverse of the unscaled tridiagonal matrix has entry

$$\frac{\min(i,j)(n+1-\max(i,j))}{n+1}.$$

Multiplying by h^2 and dividing by h gives

$$h\min(i,j)\left(1-\frac{\max(i,j)}{n+1}\right)=\min(x_i,x_j)-x_ix_j.$$

□

Corollary E12.2 (Discrete Green quadrature). *If u^h solves*

$$L_hu^h=f^h,$$

then every nodal value has the exact representation

$$u_i^h=h\sum_{j=1}^{n}G(x_i,x_j)f_j^h.$$

Thus the discrete Green operator is the nodal rectangle quadrature of the continuum Green kernel.

Proof. By theorem E12.1,

$$(L_h^{-1})_{ij}=hG(x_i,x_j).$$

Multiply the inverse matrix by the source vector. □

Proposition E12.3 (Discrete maximum bound). *The inverse L_h^{-1} has nonnegative entries and*

$$\left\|L_h^{-1}\right\|_{\infty\to\infty}=\max_{1\le i\le n}\frac{x_i(1-x_i)}{2}\le\frac{1}{8}.$$

Proof. Nonnegativity follows from the entry formula in theorem E12.1. Let $v_i=x_i(1-x_i)/2$. Since the second difference is exact on quadratics,

$$L_hv=1.$$

Hence the ith row sum of L_h^{-1} is v_i. For a nonnegative matrix, the infinity operator norm is the largest row sum. □

Theorem E12.4 (Second order Poisson solution error). *Suppose $u\in C^4[0,1]$ solves the Dirichlet problem above. Let*

$$f_i^h=f(x_i)$$

and let u^h solve

$$L_hu^h=f^h.$$

Then

$$\max_i\left|u_i^h-u(x_i)\right|\le\frac{h^2}{96}\left\|u^{(4)}\right\|_\infty.$$

Proof. Taylor expansion with remainder gives

$$\left\|L_h(u(x_i))_i-f^h\right\|_\infty\le\frac{h^2}{12}\left\|u^{(4)}\right\|_\infty.$$

Apply L_h^{-1} and use proposition E12.3. □

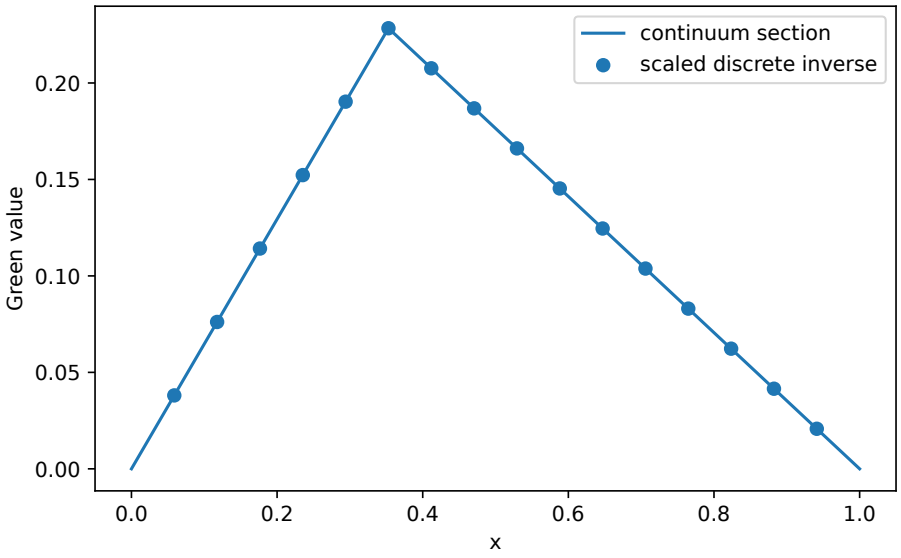


Figure E12. The continuum Dirichlet Green section and the scaled discrete inverse. The nodal values agree exactly.

The discrete eigenpairs are

$$\lambda_{k,h} = \frac{4}{h^2}\sin^2\left(\frac{k\pi h}{2}\right), \qquad v_{k,h}(i) = \sin(k\pi x_i).$$

Theorem E12.5 (Low mode eigenvalue control). *For $kh \le 1$,*

$$0 \le (k\pi)^2 - \lambda_{k,h} \le \frac{(k\pi)^4}{12}h^2.$$

Consequently, for every fixed k,

$$\lambda_{k,h} = (k\pi)^2 + O(h^2).$$

Proof. Set $x = k\pi h/2$. The condition $kh \le 1$ gives $0 \le x \le \pi/2$. On this interval,

$$\sin x \ge x - \frac{x^3}{6},$$

so

$$0 \le x^2 - \sin^2 x \le \frac{x^4}{3}.$$

Multiply by $4/h^2$. □

Corollary E12.6 (Heat and fractional mode transfer). *Let $\lambda_k = (k\pi)^2$. For $t \ge 0$,*

$$0 \le \mathrm{e}^{-t\lambda_{k,h}} - \mathrm{e}^{-t\lambda_k} \le \frac{t(k\pi)^4}{12}h^2\mathrm{e}^{-t\lambda_{k,h}}.$$

For $0 < \alpha \le 1$ and fixed k,

$$\lambda_{k,h}^{\alpha} = \lambda_k^{\alpha} + O(h^2)$$

and

$$\mathrm{e}^{-t\lambda_{k,h}^{\alpha}} = \mathrm{e}^{-t\lambda_k^{\alpha}} + O(th^2)$$

uniformly for t in bounded intervals.

Proof. The ordinary heat estimate follows from the mean value theorem and theorem E12.5. For fixed k, the eigenvalues stay uniformly away from zero once h is small. The function x^{α} is then Lipschitz on a common compact interval containing both eigenvalues. Apply the mean value theorem twice. □

Finite difference schemes and their stability theory are treated more broadly in [Str04]. The present result uses the special one-dimensional Green structure to obtain an explicit constant.

E13 Finite state Feynman–Kac transfer

How to read the result

The potential below is real. Its exponential is a potential weight in general, and a survival weight when the potential is nonnegative. This distinction does not change the finite transfer formula.

Let K be column stochastic. Let V be a real potential on the states and let

$$D_V = \operatorname{diag}(V).$$

For functions written as columns, the backward Markov operator is K^T.

Theorem E13.1 (Discrete Feynman–Kac formula). *Define*

$$M_V = \operatorname{diag}(\mathrm{e}^{-V})K^T.$$

For terminal data f,

$$(M_V^n f)_i = \mathbb{E}_i\left[\exp\left(-\sum_{r=0}^{n-1} V(X_r)\right) f(X_n)\right].$$

Proof. For one step,

$$(M_V f)_i = \mathrm{e}^{-V(i)} \sum_j K_{ji} f_j.$$

Iterating expands the matrix product over all paths. The product of transition entries is the path probability, and the diagonal factors give the exponential potential weight. □

Theorem E13.2 (Continuous time finite state formula). *Let Q be a finite Markov generator and let $(X_t)_{t\geq 0}$ be the corresponding chain. Then*

$$\left(\mathrm{e}^{t(Q^T - D_V)} f\right)_i = \mathbb{E}_i\left[\exp\left(-\int_0^t V(X_s)\,\mathrm{d}s\right) f(X_t)\right].$$

If $V \geq v_0$, then

$$\left\|\mathrm{e}^{t(Q^T - D_V)}\right\|_{\infty\to\infty} \leq \mathrm{e}^{-v_0 t}.$$

Proof. Conditioning over the first small time interval gives the backward equation

$$\dot{u} = (Q^T - D_V)u, \qquad u(0) = f.$$

The finite system has a unique solution, namely the displayed matrix exponential. The expectation bound follows from

$$\exp\left(-\int_0^t V(X_s)\,\mathrm{d}s\right) \le \mathrm{e}^{-v_0 t}.$$

This finite argument is the matrix form of the Feynman–Kac principle originating in [Kac49]. □

Theorem E13.3 (Alternating propagation and killing)**.** *For every $t \ge 0$,*

$$\mathrm{e}^{t(Q^T - D_V)} = \lim_{n\to\infty} \left(\mathrm{e}^{-tD_V/n}\mathrm{e}^{tQ^T/n}\right)^n$$

in operator norm. Each finite product alternates one Markov propagation step with one diagonal survival weight.

Proof. Set $B = -D_V$ and $C = Q^T$. For bounded matrices,

$$\mathrm{e}^{tB/n}\mathrm{e}^{tC/n} = I + \frac{t}{n}(B + C) + O(n^{-2})$$

uniformly in operator norm. The standard telescoping comparison with $(I + t(B + C)/n)^n$ shows that the product converges to $\mathrm{e}^{t(B+C)}$. This is the displayed factor order, so the limit follows. □

Corollary E13.4 (Positivity and composition)**.** *The operators*

$$F_V(t) = \mathrm{e}^{t(Q^T - D_V)}$$

form a positive semigroup. If $f \ge 0$, then $F_V(t)f \ge 0$.

Proof. Both claims follow from the expectation formula. □

E14 Schur–Feshbach elimination

Let $X = X_P \oplus X_Q$ and write

$$A = \begin{pmatrix} A_{PP} & A_{PQ} \\ A_{QP} & A_{QQ} \end{pmatrix}.$$

For $zI - A_{QQ}$ invertible, define

$$\Sigma_P(z) = A_{PQ}(zI - A_{QQ})^{-1}A_{QP}$$

and

$$F_P(z) = zI - A_{PP} - \Sigma_P(z).$$

Theorem E14.1 (Schur–Feshbach resolvent). *Assume $zI - A_{QQ}$ is invertible. Then $zI - A$ is invertible if and only if $F_P(z)$ is invertible. In that case,*

$$P(zI - A)^{-1}P = F_P(z)^{-1}.$$

In finite dimension,

$$\det(zI - A) = \det(zI - A_{QQ}) \det F_P(z).$$

Proof. Set

$$L_z = \begin{pmatrix} I & -A_{PQ}(zI - A_{QQ})^{-1} \\ 0 & I \end{pmatrix}$$

and

$$S_z = \begin{pmatrix} F_P(z) & 0 \\ -A_{QP} & zI - A_{QQ} \end{pmatrix}.$$

Direct multiplication gives $zI - A = L_z S_z$. The first factor is invertible. The claims follow from triangular inversion and the determinant of the second factor. This is the finite block form of the Feshbach reduction [Fes58]. □

E14.1 Time domain memory

For the homogeneous equation $\dot{x} = Ax$, variation of constants gives

$$x_Q(t) = \mathrm{e}^{tA_{QQ}} x_Q(0) + \int_0^t \mathrm{e}^{(t-s)A_{QQ}} A_{QP} x_P(s)\, \mathrm{d}s.$$

Substitution into the retained equation gives

$$\dot{x}_P(t) = A_{PP}x_P(t) + A_{PQ}\mathrm{e}^{tA_{QQ}}x_Q(0) + \int_0^t \mathcal{K}(t-s)x_P(s)\, \mathrm{d}s,$$

where

$$\mathcal{K}(t) = A_{PQ}\mathrm{e}^{tA_{QQ}}A_{QP}.$$

Theorem E14.2 (Memory and self-energy). *If z lies to the right of the spectrum of A_{QQ}, then*

$$\int_0^\infty \mathrm{e}^{-zt}\mathcal{K}(t)\, \mathrm{d}t = \Sigma_P(z).$$

If, for constants $M \geq 0$ and $\gamma > 0$,

$$\left\|\mathrm{e}^{tA_{QQ}}\right\| \leq M\mathrm{e}^{-\gamma t},$$

then

$$\|\mathcal{K}(t)\| \leq M\left\|A_{PQ}\right\|\left\|A_{QP}\right\|\mathrm{e}^{-\gamma t}$$

and

$$\|\mathcal{K}\|_{L^1(0,\infty)} \le \frac{M}{\gamma}\,\|A_{PQ}\|\|A_{QP}\|.$$

Proof. If $X_Q = \{0\}$, the memory kernel and self-energy are both zero, so the identity is immediate. Otherwise write $B = A_{QQ}$, which is a bounded operator on the hidden Banach space, and put $s(B) = \max \operatorname{Re}\operatorname{Spec}(B)$. Choose a real β with $s(B) < \beta < \operatorname{Re} z$. Spectral mapping and the spectral-radius formula give $r(\mathrm{e}^B) = \mathrm{e}^{s(B)} < \mathrm{e}^{\beta}$, hence $\|\mathrm{e}^{nB}\| \le M_\beta \mathrm{e}^{\beta n}$ for every integer $n \ge 0$, after increasing M_β to cover the finitely many initial values. Write $t = n + r$, where $0 \le r < 1$. Then

$$\|\mathrm{e}^{tB}\| \le M_\beta \exp(\|B\| + \max(0, -\beta))\mathrm{e}^{\beta t}.$$

Thus $\mathrm{e}^{-zt}\mathrm{e}^{tB}$ is Bochner integrable and tends to zero. Integrating its derivative gives both inverse identities for $R = \int_0^\infty \mathrm{e}^{-zt}\mathrm{e}^{tB}\,\mathrm{d}t$, so $R = (zI - B)^{-1}$. This is the bounded-generator growth principle in [EN00, Chapter IV, Corollary 3.12]. Multiplication by the coupling blocks gives the first identity. The remaining estimates follow directly from the assumed positive decay rate. □

Theorem E14.3 (Criterion for exact local closure)**.** *Assume $x_Q(0) = 0$ and $d = \dim X_Q < \infty$. The following conditions are equivalent.*

1. *For every retained initial state,*
$$P\mathrm{e}^{tA}P = \mathrm{e}^{tA_{PP}}P$$
for all $t \ge 0$.
2. *$\Sigma_P(z) = 0$ for every sufficiently large z.*
3. *$\mathcal{K}(t) = 0$ for every $t \ge 0$.*
4. *If $d = \dim X_Q$, then*
$$A_{PQ}A_{QQ}^k A_{QP} = 0$$
for $0 \le k < d$.

Proof. By theorem E2.5, the Laplace transform of the projected propagator is the projected resolvent. By theorem E14.1, the first condition is equivalent to

$$(zI - A_{PP} - \Sigma_P(z))^{-1} = (zI - A_{PP})^{-1}$$

for all sufficiently large z, which is equivalent to the second condition. The second and third conditions are equivalent by theorem E14.2 and uniqueness of the Laplace transform.

For large z,

$$\Sigma_P(z) = \sum_{k=0}^{\infty} \frac{A_{PQ}A_{QQ}^k A_{QP}}{z^{k+1}}.$$

Thus the second condition implies the moment identities. Conversely, the first d identities imply all later ones by the Cayley–Hamilton theorem, so the series vanishes. □

Remark E14.4. A finite hidden block produces a strictly proper rational self-energy. It tends to zero as z tends to infinity. Hence a nonzero constant self-energy cannot arise from hidden dynamics alone. A constant correction must already be included in the retained generator.

E15 Memory modes and controlled compression

How to read the result

Stable hidden modes may carry polynomial factors when a Jordan block is present. A spectral gap controls eventual decay, while nonnormal coupling can produce substantial intermediate memory. The compression certificate below is optimal for its declared primary-block selection problem.

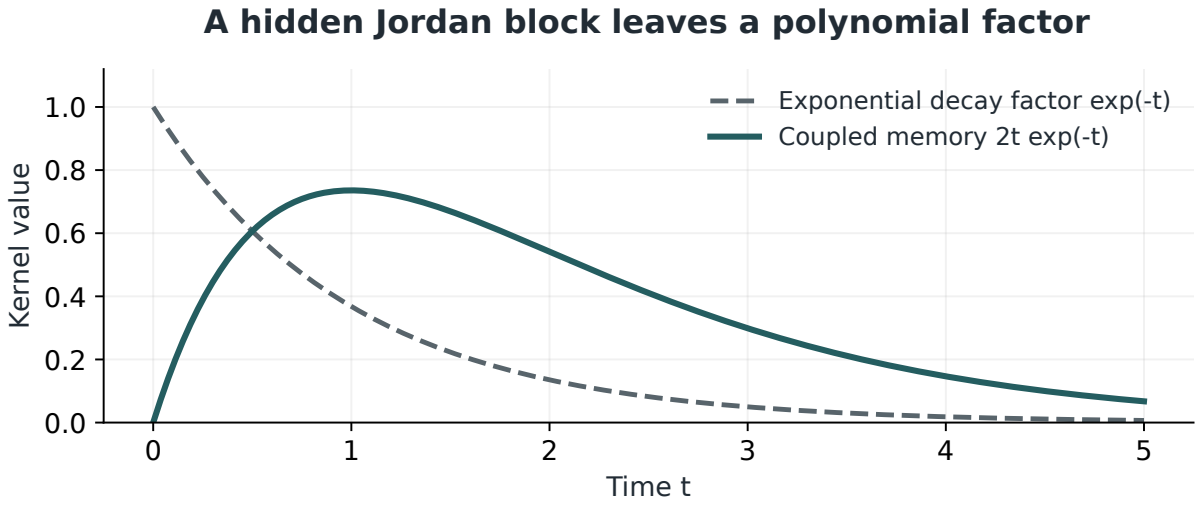


Figure E13. A two-coordinate Jordan hidden block has an exponential decay factor and a polynomial factor. The nilpotent polynomial is checked exactly at the displayed rational times. The exponential curve is a plotted illustration of the formula.

Finite hidden systems produce rational self-energies and finite modal memory. This is the state space viewpoint familiar from linear systems theory [Kai80]. The same formulas also underlie the Mori–Zwanzig representation of reduced dynamics [Zwa61, CHK00].

Theorem E15.1 (Polynomial exponential memory of a finite hidden system)**.** *Let X_Q be a finite dimensional complex space. Let $\mu_1, \dots, \mu_r$ be the distinct eigenvalues of A_{QQ}, let Q_j be the corresponding primary spectral projections, and set*

$$N_j = (A_{QQ} - \mu_j I)Q_j.$$

Let ν_j be the least positive integer for which $N_j^{\nu_j} = 0$, and define

$$B_{j,k} = A_{PQ} N_j^k Q_j A_{QP} \qquad (0 \le k < \nu_j).$$

Then

$$\mathcal{K}(t) = \sum_{j=1}^{r} \mathrm{e}^{\mu_j t} \sum_{k=0}^{\nu_j - 1} \frac{t^k}{k!} B_{j,k}$$

and

$$\Sigma_P(z) = \sum_{j=1}^{r} \sum_{k=0}^{\nu_j - 1} \frac{B_{j,k}}{(z-\mu_j)^{k+1}}.$$

Suppose $\operatorname{Re}\mu_j < 0$ *for every* j*. If* $J \subseteq \{1, \dots, r\}$ *and* $\mathcal{K}_J$ *retains the complete primary blocks indexed by* J*, then*

$$\|\mathcal{K} - \mathcal{K}_J\|_{L^1(0,\infty)} \le \sum_{j \notin J} \sum_{k=0}^{\nu_j - 1} \frac{\|B_{j,k}\|}{(-\operatorname{Re}\mu_j)^{k+1}}.$$

Proof. The primary decomposition gives

$$A_{QQ} Q_j = \mu_j Q_j + N_j$$

with commuting scalar and nilpotent parts on $Q_j X_Q$. Therefore

$$\mathrm{e}^{t A_{QQ}} Q_j = \mathrm{e}^{\mu_j t} \sum_{k=0}^{\nu_j - 1} \frac{t^k}{k!} N_j^k Q_j.$$

Multiplication by A_{PQ} and A_{QP} gives the memory formula. On the jth primary space, the nilpotent resolvent expansion is

$$(zI - A_{QQ})^{-1} Q_j = \sum_{k=0}^{\nu_j - 1} \frac{N_j^k Q_j}{(z-\mu_j)^{k+1}}.$$

Multiplication by the coupling blocks gives the self-energy formula for every z outside the hidden spectrum. Equivalently, on a right half plane it is the Laplace transform identity

$$\int_0^\infty \mathrm{e}^{-zt} \mathrm{e}^{\mu_j t} \frac{t^k}{k!} \, \mathrm{d}t = \frac{1}{(z-\mu_j)^{k+1}}.$$

Under stability,

$$\int_0^\infty \mathrm{e}^{\operatorname{Re}\mu_j t} \frac{t^k}{k!} \, \mathrm{d}t = \frac{1}{(-\operatorname{Re}\mu_j)^{k+1}}.$$

Sum the omitted terms. □

Corollary E15.2 (Optimal selection for the certified primary block bound)**.** *Under the stable assumptions of theorem E15.1, define the primary block weights*

$$w_j = \sum_{k=0}^{\nu_j - 1} \frac{\|B_{j,k}\|}{(-\operatorname{Re}\mu_j)^{k+1}}.$$

Among all sets J containing k primary blocks, the certified upper bound

$$\sum_{j\notin J} w_j$$

is minimized by retaining the k largest weights. If

$$w_{(1)} \geq w_{(2)} \geq \cdots \geq w_{(r)},$$

then the least number of retained primary blocks certified to give L^1 error at most ε is the least k for which

$$\sum_{j>k} w_{(j)} \leq \varepsilon.$$

Proof. If a retained set contains a block with smaller weight than an omitted block, exchanging them reduces the omitted weight sum. Repeating the exchange gives the set of the k largest weights. The tolerance statement is the corresponding sorted tail condition. □

Corollary E15.3 (Normal hidden dynamics)**.** *Suppose X_Q is a finite dimensional Hilbert space. If A_{QQ} is normal with spectral resolution*

$$A_{QQ} = \sum_{j=1}^{r} \mu_j Q_j,$$

then

$$\mathcal{K}(t) = \sum_{j=1}^{r} \mathrm{e}^{\mu_j t} B_j, \qquad \Sigma_P(z) = \sum_{j=1}^{r} \frac{B_j}{z-\mu_j},$$

where

$$B_j = A_{PQ} Q_j A_{QP}.$$

If $\operatorname{Re}\mu_j < 0$, then

$$\|\mathcal{K} - \mathcal{K}_J\|_{L^1(0,\infty)} \leq \sum_{j\notin J} \frac{\|B_j\|}{-\operatorname{Re}\mu_j}.$$

Proof. Normality removes every nilpotent part, so $\nu_j = 1$. Apply theorem E15.1. □

Corollary E15.4 (Pole order and hidden Jordan depth)**.** *Every finite hidden block produces a strictly proper rational self-energy. At an eigenvalue μ_j, its pole order is at most the size of the largest Jordan block of A_{QQ} at μ_j. Equality holds when the corresponding highest coefficient B_{j,ν_j-1} is nonzero.*

Proof. Read the pole order from the partial fraction expansion in theorem E15.1. □

Corollary E15.5 (Finite auxiliary state realization)**.** *Every kernel of the form*

$$\mathcal{K}(t) = B\mathrm{e}^{tD}C$$

can be represented exactly by adjoining the finite auxiliary state

$$\dot{y} = Dy + Cx_P, \qquad \dot{x}_P = A_{PP}x_P + By, \qquad y(0) = 0.$$

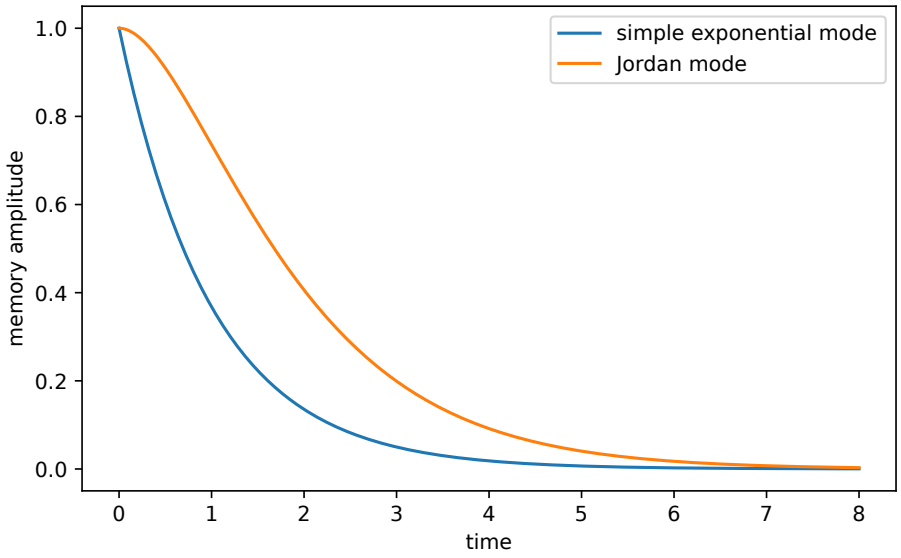


Figure E14. A stable Jordan block produces a polynomial exponential memory mode. A diagonal hidden block produces only pure exponentials.

A nonzero auxiliary initial state adds the explicit forcing $Be^{tD}y(0)$. The self-energy of the augmented system is

$$B(zI - D)^{-1}C.$$

Proof. Eliminate y by variation of constants or apply theorem E14.1 to the augmented block matrix. □

Theorem E15.6 (Volterra stability under kernel compression). *Let $S(t)$ be a strongly continuous semigroup on a Banach space X, fix $T > 0$, and assume*

$$\|S(t)\| \le M_T \quad (0 \le t \le T).$$

Let $K, \widetilde{K} \in L^1((0,T), \mathcal{B}(X))$, let $g \in L^1((0,T), X)$, and let $x, \tilde{x} \in C([0,T], X)$. Suppose these are mild solutions with the same initial state and forcing, except that K is replaced by $\widetilde{K}$:

$$x(t) = S(t)x_0 + \int_0^t S(t-s)g(s)\,\mathrm{d}s + \int_0^t S(t-s)\int_0^s K(s-r)x(r)\,\mathrm{d}r\,\mathrm{d}s.$$

Assume

$$\sup_{0\le t\le T} \|\tilde{x}(t)\| \le B.$$

Set

$$L = \|K\|_{L^1(0,T)}, \qquad \varepsilon = \left\|K - \widetilde{K}\right\|_{L^1(0,T)}.$$

Then

$$\sup_{0\le t\le T} \|x(t) - \tilde{x}(t)\| \le M_T B T \varepsilon \exp(M_T L T).$$

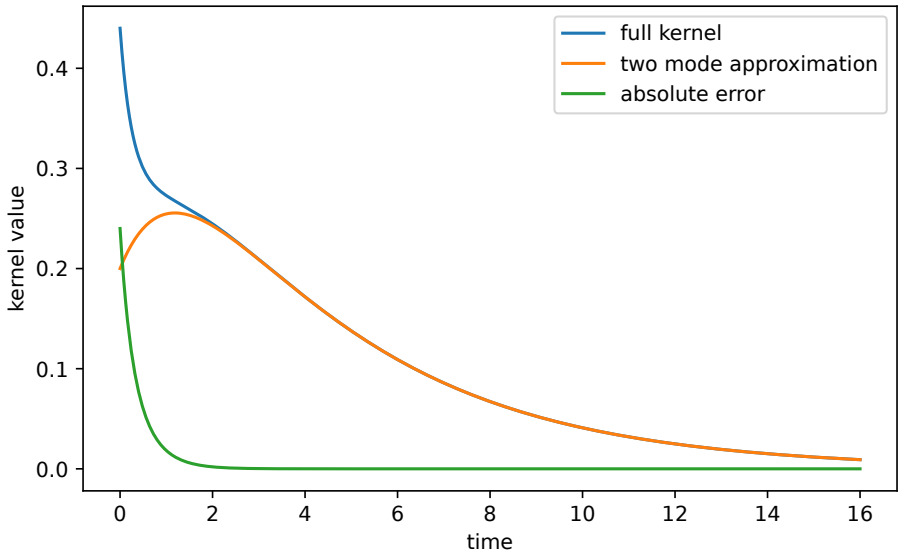


Figure E15. Full and truncated memory kernels with their pointwise difference. The bound uses integrated error.

Proof. Let

$$E(t) = \sup_{0 \le r \le t} \|x(r) - \tilde{x}(r)\|.$$

Subtract the two mild equations. Fubini's theorem and the triangle inequality give

$$E(t) \le M_T L \int_0^t E(r)\,\mathrm{d}r + M_T B t \varepsilon.$$

For $t \le T$, replace the last factor t by T and apply Gronwall's inequality. □

Corollary E15.7 (Trajectory bound for modal truncation)**.** *Under the assumptions of theorems E15.1 and E15.6, replacing $\mathcal{K}$ by $\mathcal{K}_J$ changes the retained trajectory by at most*

$$M_T B T \exp(M_T L T) \sum_{j \notin J} \sum_{k=0}^{\nu_j - 1} \frac{\|B_{j,k}\|}{(-\operatorname{Re} \mu_j)^{k+1}}.$$

Proof. Insert the L^1 error from theorem E15.1 into theorem E15.6. □

Remark E15.8. A finite hidden realization is exact for rational self-energy. A general memory kernel may require infinitely many modes or a continuum of modes. The compression theorem gives a stability estimate once an L^1 approximation has been supplied. The largest weight rule is optimal for the certified blockwise bound. It need not minimize the true kernel error when different modal terms cancel.

E16 Survival, moments, and competing risks

Let A be a transient state subgenerator. Its off-diagonal entries are nonnegative and its column sums are nonpositive. Let p_0 be a transient probability vector. Define

$$p(t) = \mathrm{e}^{tA}p_0, \qquad S(t) = 1^T p(t), \qquad a^T = -1^T A.$$

The row vector a^T contains the exit rates.

Theorem E16.1 (State dependent killing as survival). *Let Q be a finite Markov generator, let $V \geq 0$, and set*

$$A = Q - D_V.$$

For an initial distribution p_0, define

$$S_V(t) = 1^T \mathrm{e}^{tA}p_0.$$

Then

$$S_V(t) = \mathbb{E}_{p_0}\left[\exp\left(-\int_0^t V(X_s)\,\mathrm{d}s\right)\right].$$

Also,

$$-S_V'(t) = 1^T D_V \mathrm{e}^{tA}p_0.$$

The matrix $A = Q - D_V$ has nonnegative off-diagonal entries and nonpositive column sums, so it is the subgenerator of the killed, possibly nontransient chain. Thus the potential in the finite Feynman–Kac formula is a state dependent killing rate in the forward sub-Markov semigroup. Almost-sure absorption follows under the additional spectral stability hypothesis used below.

Proof. Average theorem E13.2 with terminal function $f = 1$ against the initial distribution. Transposition gives

$$p_0^T \mathrm{e}^{t(Q^T - D_V)}1 = 1^T \mathrm{e}^{t(Q - D_V)}p_0.$$

Since $1^T Q = 0$, differentiation gives

$$S_V'(t) = -1^T D_V \mathrm{e}^{tA}p_0.$$

□

Theorem E16.2 (Survival and hazard identities). *The event density and hazard are*

$$f(t) = a^T \mathrm{e}^{tA}p_0 = -S'(t), \qquad h(t) = \frac{f(t)}{S(t)}$$

whenever $S(t) > 0$. The survival function is nonincreasing and

$$S(t) = \exp\left(-\int_0^t h(s)\,\mathrm{d}s\right).$$

Proof. Because A is a Metzler matrix, e^{tA} is nonnegative. The column sum assumption makes a^T nonnegative. Differentiate $S(t)$:

$$S'(t) = 1^T A \mathrm{e}^{tA} p_0 = -a^T \mathrm{e}^{tA} p_0.$$

The remaining identity follows by integrating $(\log S)' = -h$. □

Theorem E16.3 (Laplace transform and moments). *Assume the spectrum of A lies in the open left half plane. Then absorption occurs with probability one. For $s \geq 0$,*

$$\mathbb{E}[\mathrm{e}^{-s\tau}] = a^T (sI - A)^{-1} p_0.$$

For every integer $m \geq 1$,

$$\mathbb{E}[\tau^m] = m! 1^T (-A)^{-m} p_0.$$

Proof. The Laplace transform follows by integrating the density in theorem E16.2. Stability of A gives

$$\int_0^\infty \mathrm{e}^{-st} \mathrm{e}^{tA} \, \mathrm{d}t = (sI - A)^{-1}.$$

At $s = 0$,

$$a^T (-A)^{-1} p_0 = 1^T p_0 = 1.$$

For the moments,

$$\int_0^\infty t^m \mathrm{e}^{tA} \, \mathrm{d}t = m! (-A)^{-m-1}.$$

Multiplying by $a^T = 1^T(-A)$ gives the result. These are the standard matrix formulas for phase type laws [Neu81, ANO96]. □

Proposition E16.4 (Residual life as a restarted semigroup). *For $S(t) > 0$, define the conditional transient distribution*

$$p_t = \frac{\mathrm{e}^{tA} p_0}{S(t)}.$$

Then

$$\frac{S(t+s)}{S(t)} = 1^T \mathrm{e}^{sA} p_t.$$

Proof. Use the semigroup law in the numerator $S(t+s) = 1^T \mathrm{e}^{sA} \mathrm{e}^{tA} p_0$. □

Theorem E16.5 (Competing risk packets). *Suppose the exit row splits into nonnegative cause rows*

$$a^T = \sum_{r=1}^{q} a_r^T.$$

Define

$$f_r(t) = a_r^T \mathrm{e}^{tA} p_0.$$

Then

$$f(t) = \sum_{r=1}^{q} f_r(t).$$

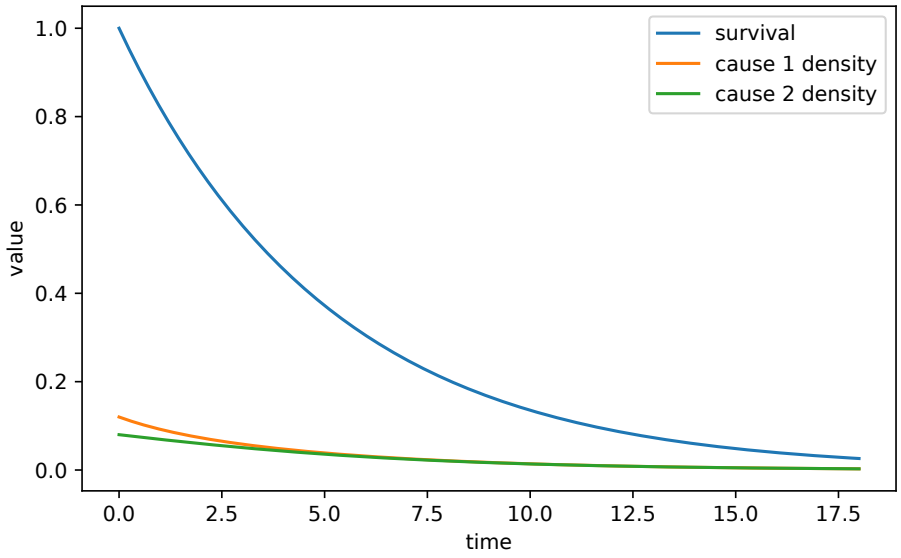


Figure E16. Survival and two cause specific event densities from one transient semigroup.

The cause specific hazard is

$$h_r(t) = \frac{f_r(t)}{S(t)},$$

and

$$h(t) = \sum_{r=1}^{q} h_r(t).$$

If the spectrum of A lies in the open left half-plane, the eventual probability of cause r is

$$\int_0^\infty f_r(t)\,\mathrm{d}t = a_r^T(-A)^{-1}p_0.$$

Proof. Every identity follows by linearity and the stable resolvent integral. □

Remark E16.6. A finite phase type hazard need not be monotone. The semigroup formulas guarantee nonnegative density and decreasing survival, not monotonicity of the hazard.

E17 Finite illustrations and computational checks

The figures in this part use small matrices chosen to make each mechanism visible. They are not evidence for claims outside the proved finite setting. The accompanying verification script recomputes the following identities with independently generated data.

1. Packet block convolution, the block differential system, and the Laplace resolvent formula.
2. The Duhamel and resolvent defect identities.
3. Exact and perturbed Markov lumping.

Identity	Typical residual
Packet block convolution	below 3×10^{-14}
Duhamel defect	below 3×10^{-12}
Markov lumping	below 3×10^{-14}
Bayesian reconstruction	below 3×10^{-15}
Fractional spectral law	below 3×10^{-12}
Nodal Green identity	below 3×10^{-13}
Feynman–Kac enumeration	below 3×10^{-14}
Schur–Feshbach resolvent	below 3×10^{-12}
Memory Laplace identity	below 3×10^{-11}
Survival moment formulas	below 3×10^{-11}

Table E2. Numerical tolerances in the finite verification suite. The proofs do not depend on these computations.

4. Conditional expectation, tower laws, martingale packet energy, and Hermite multipliers.
5. Bayesian reconstruction, sufficiency, local distortion, and information decomposition.
6. The evidence variance identity along a Bayesian flow.
7. The Balakrishnan formula and the fractional defect estimate.
8. Nodal Green reconstruction and the Poisson error bound.
9. Discrete Feynman–Kac path enumeration.
10. The Schur–Feshbach resolvent and memory transform.
11. Jordan memory modes, modal truncation, and the Volterra stability estimate.
12. Feynman–Kac killing, survival transforms, moments, and competing risk masses.

E18 Limits and further questions

The bounded inverse limit theorem isolates one clean route to continuum evolution. Unbounded generators require graph norms, domain compatibility, and a generation theorem. The range leakage theorem converts the closure quantity into an off-diagonal generator estimate for bounded orthogonal refinement models. Concrete discretizations still require a uniform estimate for that off-diagonal block.

The finite Bayesian sufficiency theorem is exact, but it concerns one likelihood and one fixed partition. The relative entropy chain rule measures both retained and discarded information for that update. Statistical sufficiency for a parametric family requires a common packet structure that preserves the full parameter dependence. In continuous models the corresponding object is a sufficient sigma algebra or statistic.

The fractional defect theorem uses a common spectral gap. Removing that gap requires weaker norms or a Hölder estimate near the null space. For sectorial operators, the same calculation should be carried out through resolvent bounds on a sector, with domains tracked explicitly. The finite spectral statement does not settle those issues.

The Green and Poisson results exploit a one-dimensional second difference matrix. Higher dimensions, irregular meshes, and variable coefficients require a separate stability theory. The packet language remains useful only after coercivity, consistency, and boundary approximation have been established.

Memory compression is exact once a finite hidden realization is known and quantitative once an integrable kernel approximation is supplied. The primary decomposition shows that a finite hidden block always produces polynomial exponential modes, including the effects of Jordan chains. Choosing a short approximation remains a different problem. Strongly nonnormal systems can exhibit transient growth and may require balanced or rational approximation methods. General Volterra kernels need not admit a short finite realization.

Finally, Markov lumping, Bayesian sufficiency, and memory closure answer different questions. Lumpability asks whether coarse probabilities evolve autonomously. Bayesian sufficiency asks whether a fixed coarse likelihood reproduces the posterior packet masses without hiding a change in the conditional distribution inside a packet. Memory closure asks whether discarded dynamics can be represented without history. Similar notation should not obscure these distinctions.

E19 Conclusion

A bounded semigroup becomes a matrix of packet transfers once a finite packet system is fixed. The matrix obeys convolution in time, a closed differential system, a Laplace resolvent formula, and a directed path expansion. A generator defect propagates through Duhamel's formula and through the resolvent calculus. Exact compatibility gives a semigroup on the bounded inverse limit. Summable adjacent defects give convergent reconstructed evolutions. Range leakage, and in particular an off-diagonal generator block, determines whether the limit retains the semigroup law.

In probability, the same structure appears in several forms. Markov transfer is linear and exact lumpability is the vanishing of a canonical kernel obstruction. Conditional expectation is an orthogonal stochastic projector whose differences form martingale packets. The Ornstein–Uhlenbeck semigroup resolves exactly into Hermite packets. Bayesian transfer is nonlinear because of evidence normalization. Its exact two stage decomposition separates coarse packet evidence from local conditional change. Relative entropy then records exactly how much information lies at each level. A fixed sufficient coarse description exists for every prior precisely when the likelihood is constant inside each packet. The common level sets of a likelihood family give its unique coarsest fixed sufficient partition.

Positive fractional powers preserve exact intertwining. Under a common spectral gap, the Balakrishnan formula turns an ordinary generator defect into an explicit fractional defect. Green operators and finite differences show how source reconstruction fits the same transfer language. Feynman–Kac weights Markov paths by a potential without leaving the semigroup framework.

Exact elimination returns hidden packets as a rational self-energy in the resolvent and a causal memory kernel in time. The primary decomposition of the hidden generator produces every finite memory kernel as a finite sum of polynomial exponential modes. Stable blocks can be truncated with an integrated error certificate, and the Volterra estimate transfers that certificate to the retained trajectory. State dependent killing identifies the Feynman–Kac potential with a survival mechanism. Survival and competing risks are then scalar reconstructions of a transient semigroup.

This part therefore separates four operations that are often compressed into one informal idea of coarse dynamics:

$$\begin{aligned} \text{intertwining} &\longrightarrow \text{refinement control} \\ &\longrightarrow \text{statistical sufficiency} \\ &\longrightarrow \text{memory closure}. \end{aligned}$$

Each arrow has its own hypotheses and its own obstruction.

Part III Retained results and operator calculus

R1 Transfer paths and product-topology limits

Let $T(t)$ be an arbitrary strongly continuous semigroup on a complex Banach space X, with generator A, and suppose $\|T(t)\| \le Me^{\omega t}$ for $M \ge 1$ and $\omega \in \mathbb{R}$. Let $(P_\alpha)_{\alpha\in I_0}$ be a finite family of bounded pairwise orthogonal idempotents summing to I, and put $T_{\beta\alpha}(t) = P_\beta T(t) P_\alpha$. The generator need not be bounded. In the following resolvent statement the integral is strong: apply the integrand to each $x \in X$ and take its Bochner integral in X. It is not asserted to be a Bochner integral in the operator-norm space $\mathcal{B}(X)$. The general Laplace formula is [EN00, Chapter II, Theorem 1.10].

Theorem R1.1 (Packet convolution). *For all $s, t \ge 0$ and all packets α, β,*

$$\boxed{T_{\beta\alpha}(s+t) = \sum_{\gamma\in I_0} T_{\beta\gamma}(s) T_{\gamma\alpha}(t).}$$

The original semigroup is reconstructed from its blocks by

$$T(t) = \sum_{\alpha,\beta\in I_0} T_{\beta\alpha}(t).$$

Proof. Insert the identity between the two semigroup factors:

$$\begin{aligned} P_\beta T(s+t) P_\alpha &= P_\beta T(s) \left(\sum_\gamma P_\gamma\right) T(t) P_\alpha \\ &= \sum_\gamma P_\beta T(s) P_\gamma T(t) P_\alpha. \end{aligned}$$

The reconstruction formula follows by inserting the packet identity on both sides of $T(t)$. □

Proposition R1.2 (Packet resolvent convolution). *Let z belong to the resolvent set of A. If $\operatorname{Re} z$ is larger than a growth bound for the semigroup, then*

$$P_\beta (zI - A)^{-1} P_\alpha = \int_0^\infty \mathrm{e}^{-zt} T_{\beta\alpha}(t)\,\mathrm{d}t.$$

Proof. The standard Laplace representation of the resolvent is

$$(zI - A)^{-1} = \int_0^\infty \mathrm{e}^{-zt} T(t)\,\mathrm{d}t.$$

Multiplication by the bounded packet projections may be passed through the Bochner integral. □

For the path-order statement, now assume A is bounded as in Part II. A directed edge from α to β means that $P_\beta A P_\alpha$ is nonzero, with $\alpha \ne \beta$. A packet always has the length-zero path to itself.

Corollary R1.3 (Minimum transfer order)**.** *Suppose every directed packet path from α to β uses at least m_0 off-diagonal transfers. Then*

$$P_\beta \mathrm{e}^{tA} P_\alpha = O(t^{m_0}) \quad \text{as } t \downarrow 0.$$

If there is no directed path from α to β, the block vanishes for all t.

Proof. Every nonzero term in theorem E2.8 contains at least m_0 time integrations. The simplex $0 < r_1 < \cdots < r_m < t$ has volume $t^m/m!$, so the first possible order is t^{m_0}. If there is no path, every packet term vanishes. □

Let $C_j : X_{j+1} \to X_j$ be bounded linear maps, and put $C_{\ell,m} = C_\ell \cdots C_{m-1}$ for $\ell < m$, with $C_{j,j} = I$. No common ambient space is assumed.

Theorem R1.4 (Multilevel telescoping identity)**.** *Let T_j be bounded operators on X_j, and set*

$$E_j = C_j T_{j+1} - T_j C_j.$$

Then

$$\boxed{C_{\ell,m} T_m - T_\ell C_{\ell,m} = \sum_{j=\ell}^{m-1} C_{\ell,j} E_j C_{j+1,m}.}$$

Proof. Expand each summand as

$$C_{\ell,j+1} T_{j+1} C_{j+1,m} - C_{\ell,j} T_j C_{j,m}.$$

The intermediate terms cancel. The first surviving term is $C_{\ell,m} T_m$, and the last is $-T_\ell C_{\ell,m}$. □

Corollary R1.5 (Summable generator defects)**.** *Suppose C_j are contractions and $(T_j(t))$ are contraction semigroups with bounded generators A_j. Set*

$$\delta_j = \left\| C_j A_{j+1} - A_j C_j \right\|.$$

Then

$$\left\| C_{\ell,m} T_m(t) - T_\ell(t) C_{\ell,m} \right\| \le t \sum_{j=\ell}^{m-1} \delta_j.$$

If $\sum_{j \ge 0} \delta_j < \infty$, the right side tends to zero as $\ell \to \infty$, uniformly in $m > \ell$ and uniformly for t in a fixed bounded interval.

Proof. By theorem E3.1, the interlevel semigroup defect has norm at most $t\delta_j$. Apply theorem R1.4 and use the contraction bounds on all coarsening maps. □

Definition R1.6 (Projective limit space)**.** Let (X_j, C_j) be a sequence of Banach spaces and bounded linear maps $C_j : X_{j+1} \to X_j$. Its projective limit is

$$X_\infty = \left\{ (x_j)_{j \ge 0} \in \prod_{j \ge 0} X_j : C_j x_{j+1} = x_j \text{ for every } j \right\}.$$

We equip X_∞ with the subspace topology inherited from the product topology.

Theorem R1.7 (Exact projective limit semigroup). *Suppose each X_j carries a strongly continuous semigroup $(T_j(t))_{t\geq 0}$, and*

$$C_j T_{j+1}(t) = T_j(t) C_j$$

for every j and every $t \geq 0$. Then

$$T_\infty(t)(x_j)_{j\geq 0} = (T_j(t)x_j)_{j\geq 0}$$

defines a strongly continuous semigroup on X_∞.

Proof. Exact intertwining gives

$$C_j T_{j+1}(t) x_{j+1} = T_j(t) C_j x_{j+1} = T_j(t) x_j.$$

Thus the coordinatewise evolution remains in the projective limit. The semigroup law holds in every coordinate. Strong continuity in the product topology follows because every coordinate map

$$t \longmapsto T_j(t)x_j$$

is continuous. □

Corollary R1.8 (Coordinate generators). *Assume the setting of theorem R1.7. If $x = (x_j)$ satisfies $x_j \in \mathrm{Dom}(A_j)$ and the sequence $(A_j x_j)$ belongs to X_∞, then x lies in the generator domain of T_∞ and*

$$A_\infty x = (A_j x_j)_{j\geq 0}.$$

Proof. The difference quotients converge in every coordinate to $A_j x_j$. The assumed compatibility places the coordinatewise limit in X_∞, which is exactly convergence in the product topology. □

The product topology used here differs from the bounded inverse-limit norm of Part II. No uniform boundedness of coordinate generators is required by the coordinatewise statement.

R2 Reversible coarse dynamics

Let Q be a finite Markov generator acting on column probability vectors, let (B_a) be a partition into nonempty blocks, and let $C_{ai} = 1_{i\in B_a}$. Strong lumpability means $CQ = \overline{Q}C$ for a coarse Markov generator. For observables, the backward generator is Q^T. Weighted L^2 inner products use the stated stationary probability.

Theorem R2.1 (Reversibility descends). *Assume Q is strongly lumpable and reversible with respect to a positive stationary distribution π, so*

$$Q_{ij}\pi_j = Q_{ji}\pi_i.$$

Define

$$\overline{\pi}_a = \sum_{i\in B_a} \pi_i.$$

Then $\overline{\pi}$ is stationary for $\overline{Q}$, and

$$\overline{Q}_{ab}\overline{\pi}_b = \overline{Q}_{ba}\overline{\pi}_a.$$

Thus the coarse chain is reversible.

Proof. Stationarity follows from

$$\overline{Q}\overline{\pi} = \overline{Q}C\pi = CQ\pi = 0.$$

For reversibility, use the aggregate rate formula:

$$\begin{aligned}\overline{Q}_{ab}\overline{\pi}_b &= \sum_{j\in B_b}\pi_j \sum_{i\in B_a} Q_{ij} \\ &= \sum_{i\in B_a}\sum_{j\in B_b}\pi_i Q_{ji} \\ &= \overline{Q}_{ba}\overline{\pi}_a.\end{aligned}$$

□

Corollary R2.2 (Coarse Dirichlet form)**.** *Under the hypotheses of theorem R2.1, let Jg be the fine function that is constant with value g_a on B_a. Then*

$$-\left\langle Jg, Q^T Jg\right\rangle_{L^2(\pi)} = -\left\langle g, \overline{Q}^T g\right\rangle_{L^2(\overline{\pi})}.$$

Proof. Expand the left side and group the fine indices by their coarse blocks. The aggregate rate criterion converts the resulting sums into the coarse generator entries. □

R3 Prior-restricted sufficiency and filtering

Let L be a strictly positive finite likelihood and let $\mathcal{B}_L(p) = L \odot p / \langle L, p\rangle$ for a probability vector p. Fix probability vectors $r^{(a)}$ supported on B_a, and restrict the priors to $p_i = q_a r_i^{(a)}$ for $i \in B_a$, with q a coarse probability vector.

Proposition R3.1 (Sufficiency relative to a prior disintegration)**.** *Define*

$$\overline{L}_a = \sum_{i\in B_a} L_i r_i^{(a)}.$$

For every prior in the declared family,

$$C\mathcal{B}_L(p) = \mathcal{B}_{\overline{L}}(q).$$

Proof. The evidence in block B_a is

$$Z_a = q_a \sum_{i\in B_a} L_i r_i^{(a)} = q_a \overline{L}_a.$$

Apply theorem E8.2. □

Lemma R3.2 (Normalization bound). *Let $a, b \in [0,\infty)^m$ be nonzero, and set $A = \|a\|_1$ and $B = \|b\|_1$. Then*

$$\boxed{\|a/A - b/B\|_1 \le \frac{2\,\|a-b\|_1}{\max\{A,B\}}.}$$

Proof. Using A in the denominator gives

$$\|a/A - b/B\|_1 \le \frac{\|a-b\|_1}{A} + \|b\|_1 \,|1/A - 1/B|\,.$$

The second term equals $|B-A|/A$, which is at most $\|a-b\|_1/A$. Interchanging a and b gives the analogous bound with B. Taking the smaller of the two bounds gives the result. □

For an arbitrary prior p, put $q = Cp$ and $Z(p,L) = \langle L, p\rangle$. Define $\ell_a(p) = \sum_{i\in B_a} L_i p_i / q_a$ when $q_a > 0$. Choose any positive value when $q_a = 0$. Only the product $q_a \ell_a(p)$ enters the formulas.

Theorem R3.3 (Exact information loss formula). *For every prior,*

$$\boxed{C\mathcal{B}_L(p) = \mathcal{B}_{\ell(p)}(q).}$$

Let $\overline{L}$ be any fixed positive coarse likelihood and set

$$d(p \mid L, \overline{L}) = \sum_{a=1}^{m} q_a \left|\ell_a(p) - \overline{L}_a\right|.$$

Then

$$\|C\mathcal{B}_L(p) - \mathcal{B}_{\overline{L}}(q)\|_1 \le \frac{2d(p \mid L, \overline{L})}{\max\{Z(p,L), \langle \overline{L}, q\rangle\}}.$$

Proof. The packet evidence in block a is $q_a \ell_a(p)$, which proves the exact identity. Apply lemma R3.2 to

$$a = q \odot \ell(p), \qquad b = q \odot \overline{L}.$$

Their ℓ^1 norms are the two evidences, and $\|a-b\|_1 = d(p \mid L, \overline{L})$. □

Corollary R3.4 (Oscillation bound). *Suppose $\overline{L}_a$ is any representative likelihood for block B_a, and set*

$$\omega_a = \max_{i\in B_a} \left|L_i - \overline{L}_a\right|.$$

Then

$$d(p \mid L, \overline{L}) \le \sum_a q_a \omega_a.$$

Thus small within block likelihood oscillation gives a small coarse posterior defect whenever the evidence is bounded away from zero.

Proof. The conditional average $\ell_a(p)$ lies in the convex hull of the likelihood values inside B_a. Hence

$$\left|\ell_a(p) - \overline{L}_a\right| \le \omega_a.$$

Multiply by q_a and sum. □

Along the Bayesian flow $p_i(t) = e^{tg_i}p_i(0)/\sum_j e^{tg_j}p_j(0)$, the following statement holds for every real observable h.

Corollary R3.5 (Covariance identity). *For any observable $h \in \mathbb{R}^n$,*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{p(t)}[h] = \mathrm{Cov}_{p(t)}(h, g).$$

In particular,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{p(t)}[g] = \mathrm{Var}_{p(t)}(g) \geq 0.$$

Proof. Differentiate $\sum_i h_i p_i(t)$ and insert the replicator equation. □

For stochastic matrices $K, \overline{K}$ and positive likelihoods $L_y, \overline{L}_y$, put $\Phi_y = \mathcal{B}_{L_y} \circ K$ and $\overline{\Phi}_y = \mathcal{B}_{\overline{L}_y} \circ \overline{K}$.

Theorem R3.6 (Exact coarse hidden Markov filter). *Assume*

$$CK = \overline{K}C$$

and every likelihood L_y is constant on each coarse block with block values $\overline{L}_y$. Then

$$C\Phi_y(p) = \overline{\Phi}_y(Cp)$$

for every prior. For any observation sequence $y_1, \dots, y_r$, the complete filtering recursion commutes with aggregation:

$$C\Phi_{y_r} \circ \cdots \circ \Phi_{y_1} = \overline{\Phi}_{y_r} \circ \cdots \circ \overline{\Phi}_{y_1} C.$$

Proof. Exact Markov lumping gives $CKp = \overline{K}Cp$. The Bayesian sufficiency criterion gives exact commutation of the update. Their composition proves the one step identity. Iterate it over the observation sequence. □

Theorem R3.7 (One step filter defect). *Let $r = Kp$, $q = Cr$, and let $\ell_y(r)$ be the exact coarse likelihood obtained by averaging L_y inside each block with respect to r. Assume*

$$0 < \overline{m} \leq (\overline{L}_y)_a \leq \overline{M}.$$

Then

$$\left\|C\Phi_y(p) - \overline{\Phi}_y(Cp)\right\|_1 \leq \frac{2d(r \mid L_y, \overline{L}_y)}{\max\{\langle L_y, r\rangle, \langle \overline{L}_y, q\rangle\}} + \frac{2\overline{M}}{\overline{m}}\left\|CKp - \overline{K}Cp\right\|_1.$$

Proof. Insert the intermediate posterior $\mathcal{B}_{\overline{L}_y}(CKp)$. The first difference is bounded by theorem R3.3. The second is bounded by theorem E8.4 applied to the coarse likelihood. □

R4 Fractional modes and killed transfer

For the one-dimensional Dirichlet difference operator, let $h = 1/M$ with an integer $M \geq 2$, let $1 \leq k \leq M-1$, and set $\lambda_{k,h} = 4h^{-2}\sin^2(k\pi h/2)$. The retained explicit coefficient below is valid for every discrete mode.

Corollary R4.1 (Fractional low mode dispersion). *Let $0 < \alpha \leq 1$. Then*

$$0 \leq (k\pi)^{2\alpha} - \lambda_{k,h}^{\alpha} \leq \frac{\alpha(4k^2)^{\alpha-1}(k\pi)^4}{12}h^2.$$

For every $t \geq 0$,

$$\left|\mathrm{e}^{-t\lambda_{k,h}^{\alpha}} - \mathrm{e}^{-t(k\pi)^{2\alpha}}\right| \leq \frac{t\alpha(4k^2)^{\alpha-1}(k\pi)^4}{12}h^2.$$

Proof. The chord bound $\sin x \geq 2x/\pi$ on $[0, \pi/2]$ gives $\lambda_{k,h} \geq 4k^2$. Since $x \mapsto x^{\alpha}$ is increasing and has decreasing derivative,

$$(k\pi)^{2\alpha} - \lambda_{k,h}^{\alpha} \leq \alpha\lambda_{k,h}^{\alpha-1}((k\pi)^2 - \lambda_{k,h}).$$

Use theorem E12.5 and the lower bound for $\lambda_{k,h}$. The exponential estimate follows from

$$\left|\mathrm{e}^{-ta} - \mathrm{e}^{-tb}\right| \leq t\,|a-b|$$

for nonnegative a and b. □

Let Q and $\overline{Q}$ be finite Markov generators in the column convention. All times in this section satisfy $t \geq 0$.

Theorem R4.2 (Potential perturbation bound). *Let V and W be nonnegative diagonal matrices. In the supremum norm,*

$$\boxed{\left\|\mathrm{e}^{t(Q^T-V)} - \mathrm{e}^{t(Q^T-W)}\right\|_{\infty\to\infty} \leq t\,\|V-W\|_{\infty\to\infty}.}$$

Proof. Both killed Markov semigroups are contractions in the supremum norm. Apply the Duhamel identity with generator defect $W - V$. □

Theorem R4.3 (Coarse Feynman–Kac transfer). *Assume $CQ = \overline{Q}C$, and let $J = C^T$ lift a coarse function to the corresponding block constant fine function. Suppose the potential is block constant, so*

$$VJ = J\overline{V}.$$

Then

$$\boxed{\mathrm{e}^{t(Q^T-V)}J = J\mathrm{e}^{t(\overline{Q}^T-\overline{V})}}$$

for every $t \geq 0$.

Proof. Transpose the Markov intertwining relation to obtain

$$Q^TJ = J\overline{Q}^T.$$

Together with $VJ = J\overline{V}$, this gives

$$(Q^T - V)J = J(\overline{Q}^T - \overline{V}).$$

The exponential intertwining follows from the power series. □

R5 Memory compression, rational realization, and phase-type consequences

Fix $T > 0$. For the next estimate let A be a bounded operator on a Banach space X, let $\mathcal{K}, \widetilde{\mathcal{K}} \in L^1((0,T), \mathcal{B}(X))$, and let f be continuous. Take continuously differentiable solutions of the displayed equations. This direct bounded-generator estimate retains a sharper exponential constant than the general mild-semigroup estimate in Part II.

Theorem R5.1 (Stability of memory compression). *Let x and $\tilde{x}$ solve*

$$\dot{x}(t) = Ax(t) + \int_0^t \mathcal{K}(t-s)x(s)\,\mathrm{d}s + f(t)$$

and

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + \int_0^t \widetilde{\mathcal{K}}(t-s)\tilde{x}(s)\,\mathrm{d}s + f(t)$$

with the same initial value. Fix $T > 0$, and assume

$$\sup_{0\le t\le T} \|\tilde{x}(t)\| \le M.$$

Set

$$a = \|A\|, \qquad k = \|\mathcal{K}\|_{L^1(0,T)}, \qquad \delta = \left\|\mathcal{K} - \widetilde{\mathcal{K}}\right\|_{L^1(0,T)}.$$

Then

$$\boxed{\sup_{0\le t\le T} \|x(t) - \tilde{x}(t)\| \le MT\delta \mathrm{e}^{(a+k)T}.}$$

Proof. Let $e = x - \tilde{x}$. Integrating the difference equation and changing the order of integration gives

$$\|e(t)\| \le (a+k)\int_0^t \|e(r)\|\,\mathrm{d}r + Mt\delta.$$

Since $t \le T$, Gronwall's inequality gives

$$\|e(t)\| \le MT\delta \mathrm{e}^{(a+k)t}.$$

Take the supremum over $[0,T]$. □

For the exact closure criterion of Part II, allowing an arbitrary bounded local generator B does not enlarge the class: differentiating $Pe^{tA}P = e^{tB}P$ at $t = 0$ forces $B = A_{PP}$ on the retained space. Thus its zero-memory criterion excludes every alternative bounded local generator.

A finite auxiliary realization of a matrix kernel means $\mathcal{K}(t) = Be^{tD}C$, with finite matrices of compatible sizes. Laplace transforms are taken in a right half-plane of absolute convergence. The rational characterization concerns continuous kernels of exponential order, so uniqueness of the ordinary Laplace transform applies.

Theorem R5.2 (Rational memory realization)**.** *A matrix valued memory transform has a finite auxiliary realization if and only if it is a strictly proper rational matrix function. More precisely,*

$$\boxed{\widehat{\mathcal{K}}(z) = B(zI - D)^{-1}C}$$

if and only if

$$\boxed{\mathcal{K}(t) = B\mathrm{e}^{tD}C.}$$

Proof. The forward implication follows by taking the Laplace transform. The resulting matrix is rational and tends to zero as z tends to infinity.

Conversely, decompose every entry of a strictly proper rational matrix into partial fractions. A term

$$\frac{c}{(z-\lambda)^r}$$

is an entry of the resolvent of an r by r Jordan block with eigenvalue λ. Taking a direct sum of these blocks and choosing the input and output maps entry by entry gives matrices B, D, C with the required resolvent. Inverse Laplace transformation gives the time domain formula. □

Corollary R5.3 (Exponential sum memory)**.** *If*

$$\mathcal{K}(t) = \sum_{j=1}^{m} B_j \mathrm{e}^{\lambda_j t} C_j,$$

then $\mathcal{K}$ has a finite auxiliary realization with one auxiliary block for each exponential. Repeated poles produce polynomial factors in t and are realized by Jordan blocks.

Proof. If B_j and C_j have an intermediate space of dimension d_j, use a block diagonal matrix D with blocks $\lambda_j I_{d_j}$, or the required Jordan blocks when poles repeat. Assemble the matrices B_j and C_j into the output and input maps. □

For the survival consequences, A is a finite Metzler matrix with nonpositive column sums and spectrum in the open left half-plane, p_0 is a probability vector, $S(t) = 1^T e^{tA} p_0$, and $f(t) = -S'(t)$. The one-phase model has $\lambda > 0$. The survival Laplace identity of Part II also holds for every complex s with $\operatorname{Re} s \geq 0$: stability gives absolute convergence of $\int_0^\infty e^{-st} e^{tA}\, dt = (sI - A)^{-1}$, and multiplying by the exit row and p_0 gives the same transform.

Corollary R5.4 (Phase type rationality)**.** *The Laplace transform of every finite phase type distribution is rational. If A is diagonalizable, then $S(t)$ and $f(t)$ are finite linear combinations of exponential functions. Jordan blocks add polynomial factors in t.*

Proof. The resolvent $(sI - A)^{-1}$ is rational. The time domain statement follows from the Jordan form of A. □

Corollary R5.5 (One phase model). *If $A = (-\lambda)$ and $p_0 = 1$, then*

$$S(t) = \mathrm{e}^{-\lambda t}, \qquad f(t) = \lambda \mathrm{e}^{-\lambda t}, \qquad h(t) = \lambda.$$

Proof. Insert the scalar generator into theorem E16.2. □

R6 Additional exact examples

These five examples use numerical parameters distinct from the visual studies.

Example R6.1 (One transfer between two packets). Let

$$A = \begin{pmatrix} a & 0 \\ c & b \end{pmatrix}$$

and let the two coordinate axes be the packet spaces. If $a \neq b$, then

$$\mathrm{e}^{tA} = \begin{pmatrix} \mathrm{e}^{at} & 0 \\ c\dfrac{\mathrm{e}^{at} - \mathrm{e}^{bt}}{a - b} & \mathrm{e}^{bt} \end{pmatrix}.$$

The lower left block is produced by one transfer from the first packet to the second. Its expansion begins with ct, in agreement with corollary R1.3. When $a = b$, the same block is $ct\mathrm{e}^{at}$.

Example R6.2 (A four state lumpable chain). Partition four states into $B_1 = \{1, 2\}$ and $B_2 = \{3, 4\}$. For positive rates a, b, u, v, set

$$Q = \begin{pmatrix} -u - a & u & b/2 & b/2 \\ u & -u - a & b/2 & b/2 \\ a/2 & a/2 & -v - b & v \\ a/2 & a/2 & v & -v - b \end{pmatrix}.$$

The rates u and v move inside the coarse blocks. Every state in B_1 leaves for B_2 at total rate a, and every state in B_2 leaves for B_1 at total rate b. Hence

$$\overline{Q} = \begin{pmatrix} -a & b \\ a & -b \end{pmatrix}$$

and $CQ = \overline{Q}C$. The internal rates disappear from the coarse generator even though they remain present in the fine evolution.

Example R6.3 (A visible likelihood loss). Let $B_1 = \{1, 2\}$, let $B_2 = \{3\}$, and take

$$p = (3/10, 2/10, 5/10), \qquad L = (1, 3, 2).$$

The coarse prior is $(1/2, 1/2)$. The exact block likelihoods are

$$\ell_1(p) = \frac{(3/10) + 3(2/10)}{1/2} = \frac{9}{5}, \qquad \ell_2(p) = 2.$$

Thus the exact coarse posterior is $(9/19, 10/19)$. If the first block is replaced by the fixed representative value 2, the approximate coarse posterior is $(1/2, 1/2)$. The ℓ^1 defect is $1/19$, while theorem R3.3 gives the valid upper bound $1/10$.

Example R6.4 (One hidden mode)**.** Consider

$$A = \begin{pmatrix} -a & c \\ d & -b \end{pmatrix}, \qquad a, b > 0,$$

and retain the first coordinate. The memory kernel and self-energy are

$$\mathcal{K}(t) = cd\mathrm{e}^{-bt}, \qquad \Sigma(z) = \frac{cd}{z+b}.$$

The retained coordinate has an exact time local closure for every retained initial value precisely when $cd = 0$. When $cd \neq 0$, one auxiliary scalar state realizes the memory exactly.

Example R6.5 (Two sequential phases)**.** Let

$$A = \begin{pmatrix} -\lambda & 0 \\ \lambda & -\mu \end{pmatrix}, \qquad p_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where $\lambda \neq \mu$ are positive. The first phase moves to the second at rate λ, and the second is absorbed at rate μ. Direct calculation gives

$$S(t) = \frac{\mu\mathrm{e}^{-\lambda t} - \lambda\mathrm{e}^{-\mu t}}{\mu - \lambda},$$

$$f(t) = \frac{\lambda\mu}{\mu - \lambda}\left(\mathrm{e}^{-\lambda t} - \mathrm{e}^{-\mu t}\right).$$

The ratio $f(t)/S(t)$ is generally not constant. A time local Markov model on two transient phases can therefore produce a nonconstant hazard after the phase coordinate is hidden.

R7 Holomorphic operator substitution

> **A route through this section**
>
> The scalar function, zero-free domain, and logarithm branch are fixed before substituting an operator. The bounded Banach-space functional calculus here is separate from the finite canonical-jet contract of Part I. Keep those spaces and hypotheses visible when using the reciprocal family.

This section presents the operator-substitution result transferred from Chapter IV. Fix an integer $N \geq 1$ and the entire scalar function

$$\mathsf{D}_N(z) = \sum_{m=0}^{\infty} \frac{(-1)^m z^{Nm}}{(Nm)!}.$$

More generally, the result applies to any specified holomorphic scalar function that is nonzero on the declared domain. On a simply connected zero-free domain U, choose $g = \log \mathsf{D}_N$ once and define $\mathsf{S}_N^{(r)}(z) = \exp(-rg(z))$ for complex r. The bounded Banach-space result below uses classical holomorphic functional calculus. It is separate from the finite canonical-jet calculus formalized in Part I.

Standard Fact R7.1 (Holomorphic functional calculus). Let A be a bounded operator on a complex Banach space. If f is holomorphic on a neighborhood of $\mathrm{Spec}(A)$, then

$$f(A) = \frac{1}{2\pi i} \int_\Gamma f(z)(zI - A)^{-1}\, dz$$

for an admissible positively oriented contour Γ lying in the holomorphy domain and the resolvent set, with winding number one about the spectrum and zero outside the domain. The calculus preserves sums, products, and composition. We use the standard form from [DS58]. An accessible proof is given in [Tay, Appendix A].

Proposition R7.2 (Operator reciprocal family)**.** *Let A be bounded and suppose $\mathrm{Spec}(A)$ lies in a simply connected region on which D_N has no zeros. Fix a holomorphic branch $g = \log \mathsf{D}_N$ there. Then*

$$\mathsf{S}_N^{(r)}(A) = \exp(-rg(A)) = (\mathsf{D}_N^{-r})(A)$$

is well defined and

$$\mathsf{S}_N^{(r)}(A)\mathsf{S}_N^{(s)}(A) = \mathsf{S}_N^{(r+s)}(A).$$

If (P_i) is a finite complete family of bounded idempotents, then

$$P_i f(A) P_j = \frac{1}{2\pi i} \int_\Gamma f(z) P_i (zI - A)^{-1} P_j\, dz$$

for every holomorphic f on a neighborhood of the spectrum.

Proof. The spectral mapping theorem makes $\mathsf{D}_N(A)$ invertible. The product identity follows from multiplicativity of the functional calculus and the fixed logarithm. Applying P_i and P_j to the Cauchy integral formula gives the packet formula. □

How to use this volume

The guide follows small models whose matrices, probability laws, and transfer formulas can be calculated explicitly. The traditional volume supplies the general arguments. The formal companion records a separate formal contract.

Routes through the guide

Plates VI.1 through VI.42 identify worked pages independently of PDF pagination. The references on Plates VI.1–VI.28 use the theorem numbering of Traditional Part I. References from Plate VI.29 onward name the corresponding strengthened result in Traditional Part II. These references distinguish preserved manuscript branches rather than merging their result counts.

Route	Mathematical question	Plates
Complete transfer	Which coordinates and modes reconstruct the same evolution?	1–6, 29
Probability and evidence	What survives averaging, aggregation, or an observation?	7–14, 26, 32, 34–36
Spectral evolution	How do rates, gaps, fractional clocks, and Gaussian modes interact?	15–17, 25, 33, 41
Green and path formulas	How do grid units and weighting conventions enter?	18–20
Eliminating hidden states	What memory returns, and how can it be compressed?	21–24, 37–40
Passing to limits	What additional conditions give convergence and closure?	27, 30–31
Evidence	Which calculations and formal records support which claims?	28, 42

Three distinctions to keep visible

A coordinate decomposition retains the whole vector if all components are kept. An aggregation map usually loses information. Elimination recovers exact retained dynamics only after adding a memory term and any hidden-state initial forcing.

An exact example checks its stated finite model. A theorem provides an argument under quantified assumptions. A Lean record certifies its declared formal statement under the recorded environment. None of these automatically certifies the other two.

Mathematical coverage

The first twenty-eight worked pages develop the finite models. Fourteen further pages connect them to the extended results. Forty editable vector figures illustrate these constructions. Their source files and retained generation records accompany the mathematical text and its formal correspondence.

The inherited formal baseline records 33 scoped results, 341 supporting theorem declarations, and 47 proof modules. The strengthened traditional branch is a separate collection of written proofs. Its broader content is not added to the formal denominator.

Traditional Part III retains complementary results with an R prefix: reversibility and Dirichlet forms (R2), prior disintegrations and filtering (R3), potential perturbation and coarse Feynman–Kac transfer (R4), and finite realizations (R5). Related models appear on Plates VI.5–VI.8, Plate VI.11, Plate VI.13, Plate VI.20, Plate VI.26, and Plates VI.39–VI.40. The bounded-operator reciprocal result migrated from Chapter IV is R7.2. Plate VI.25 and Plate VI.41 explain related finite matrix calculus. These models do not replace the supplementary hypotheses.

Read this as: Choose a question, work through its small model, then check the hypotheses in the corresponding traditional proof and formal contract.

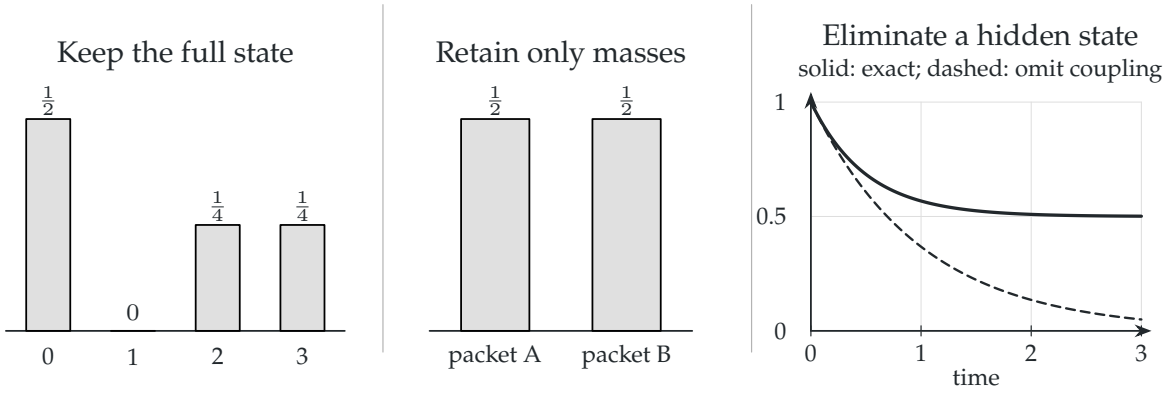
Part IV Worked visual atlas

Plate VI.1 / PAPER VI / A WORKED VISUAL ATLAS

Three ways to reduce a model

Chapter VI, Part I: Sections 1–3, 8, and 13. Read the three operations separately.

This guide follows small objects that can be computed completely. A four-state chain connects packet transfer, Markov aggregation, and Bayesian conditioning. A two-state coupled evolution exposes memory. A three-node Dirichlet problem makes the Green-operator normalization visible. These models are related by the chapter's language, not by pretending that their state spaces are identical.



Left and center use one probability vector and its two block masses. The right panel is a separate retained/hidden model, developed on Plate VI.22. Every curve is drawn from an explicit formula.

Coordinates retain, aggregation forgets, and elimination remembers

Keeping every component $P_a x$ is an exact change of representation when $\sum_a P_a = I$. Reconstruct by summing the components. In contrast, replacing (p_0, p_1, p_2, p_3) by $(p_0 + p_1, p_2 + p_3)$ forgets how mass was distributed inside either pair. That information can matter for later observations.

Eliminating a hidden state is different again. Its coordinates disappear from the displayed state, but the exact retained equation acquires a history integral. Simply deleting the coupling produces the dashed curve above, not the original evolution.

Question	Worked route	Plates
When is coarse evolution exact?	Aggregate columns, then test intertwining.	VI.4–VI.6
When is a coarse posterior exact?	Compare two hidden shapes with the same masses.	VI.9–VI.13
What changes under a fractional power?	Keep modes, change spectral rates.	VI.15–VI.17
What returns after elimination?	Compute the self-energy and memory kernel.	VI.21–VI.25

Read this as: The chapter supplies exact identities and conditional bounds. Whether a particular coarse model satisfies their hypotheses is a separate calculation.

Plate VI.2 / READ BEFORE MULTIPLYING / CONVENTIONS

A dictionary with orientations

Chapter VI, Part I: Sections 2, 4, 6, 7, and 12. Column probabilities and backward observables.

An apparent disagreement between two transfer formulas often comes from acting on different objects. Throughout this guide, probabilities are columns and observables are functions written as columns. Their pairing is $f^{\mathsf{T}}p$. The transpose distinguishes the forward action on distributions from the backward action on observations.

Object	Exact meaning	What to check
p	A nonnegative column with $\mathbf{1}^{\mathsf{T}}p = 1$.	It is not a vector of function values.
Q	Continuous Markov generator.	Off-diagonal entries are nonnegative and columns sum to zero.
K	One discrete stochastic step.	$\mathbf{1}^{\mathsf{T}}K = \mathbf{1}^{\mathsf{T}}$ and $K_{ij} \geq 0$.
$K^{\mathsf{T}}f$	Expected next observable given the current state.	The same dynamics, with the opposite action on the dual object.
P_a	A coordinate packet projector.	All packets together reconstruct the full state.
C	Sums fine probability masses into coarse blocks.	Usually rectangular and not invertible.
E_π	Conditional expectation of observables.	A weighted average, self-adjoint for the π inner product.
$\mathcal{B}_\lambda$	$\operatorname{diag}(\lambda)p/(\lambda^{\mathsf{T}}p)$.	Nonnegative likelihood and positive total evidence.
L	A nonnegative self-adjoint operator.	Its evolution is e^{-tL}, not e^{tL}.
$R(z)$	$(zI - A)^{-1}$.	z belongs to the resolvent set.
$\mathcal{K}(t)$	The memory kernel after hidden-state elimination.	It is not the transition matrix K.

One convention test

For

$$K_2 = \begin{pmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}, \qquad p = e_0, \qquad f = (2,1)^{\mathsf{T}},$$

we obtain $K_2 p = (3/4, 1/4)^{\mathsf{T}}$ and

$$f^{\mathsf{T}}K_2 p = 7/4 = (K_2^{\mathsf{T}}f)_0.$$

With departure weights $w_i = e^{-V(i)}$, the backward Feynman–Kac operator is $\operatorname{diag}(w)K_2^{\mathsf{T}}$. Its forward dual is $K_2 \operatorname{diag}(w)$. Interchanging these factors changes when the weight is applied.

> **Read this as:** Always name the object being transported. The transpose and the position of the likelihood or potential are mathematical choices, not typographic details.

Plate VI.3 / CONTINUOUS TRANSFER / THE ENTIRE SOLUTION

A clock generated by one matrix

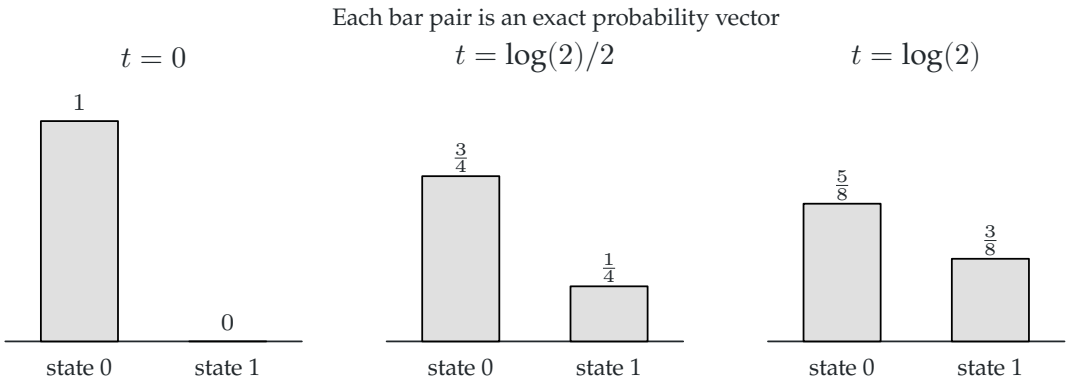
Chapter VI, Part I: Theorems 1.2 and 4.1. Finite matrix exponentials and Markov positivity.

Take the generator

$$G = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The total-mass direction $(1, 1)$ has eigenvalue zero. The contrast $(1, -1)$ has eigenvalue -2. Resolving those two directions gives the full transfer, not only its first-order approximation:

$$T(t) = e^{tG} = \frac{1}{2}\begin{pmatrix} 1 + e^{-2t} & 1 - e^{-2t} \\ 1 - e^{-2t} & 1 + e^{-2t} \end{pmatrix}.$$



A point seed at state zero approaches the uniform distribution. The displayed times were selected so that every bar label is rational. This is continuous-time evolution, not a simulated trajectory.

Two checks explain the mechanism

The constant mode is unchanged, while each elapsed interval multiplies the contrast by e^{-2t}. Consequently two intervals multiply their contrast factors:

$$e^{-2s}e^{-2t} = e^{-2(s+t)}, \qquad T(s)T(t) = T(s+t).$$

For $t \geq 0$, the entries are nonnegative and every column sums to one. Differentiation gives $T'(t) = GT(t)$. These three facts separately check the initial condition, the evolution equation, and preservation of probability.

At $t = \log 2$, the retained first coordinate is $5/8$. It would instead be $1/2$ under the isolated scalar equation $\dot{x} = -x$. The discrepancy is the mass that can leave state zero and return. On Plate VI.22, we remove the second coordinate correctly and recover exactly the same $5/8$ using a memory integral.

> **Read this as:** The semigroup lives on the complete state. One diagonal entry of a semigroup need not itself satisfy a scalar semigroup law.

Plate VI.4 / FOUR-STATE MODEL / COORDINATES AND MODES

One evolution, two packet choices

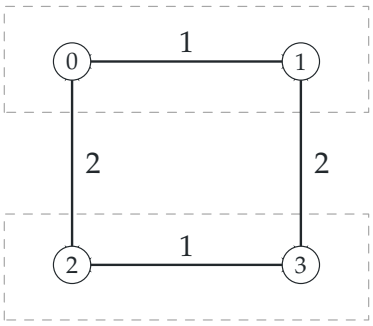
Chapter VI, Part I: Theorem 2.1, Corollary 2.2, and the packet path expansion.

Our main finite model has generator

$$Q = \begin{pmatrix} -3 & 1 & 2 & 0 \\ 1 & -3 & 0 & 2 \\ 2 & 0 & -3 & 1 \\ 0 & 2 & 1 & -3 \end{pmatrix}.$$

The graph shows rates between actual states. The four normalized sign vectors on the right form an orthonormal basis u_j. Their projectors $P_j = u_j u_j^{\mathsf{T}}$ sum to the identity and satisfy $P_i P_j = 0$ for $i \neq j$.

Off-diagonal rates of Q



Top pair A; bottom pair B

Mode signs (each divided by 2)

	0	1	2	3	rate
constant	1	1	1	1	0
within	1	−1	1	−1	−2
between	1	1	−1	−1	−4
mixed	1	−1	−1	1	−6

State coordinates have off-diagonal transfers. Spectral packets reduce the generator and evolve independently. The two dashed state groups are the coarse partition used next.

The same evolution has a closed formula

Writing $q = e^{-2t}$ gives

$$e^{tQ} = P_0 + qP_1 + q^2 P_2 + q^3 P_3.$$

At $q = 1/2$, the point seed e_0 becomes $(15, 5, 9, 3)^{\mathsf{T}}/32$. The polynomial identity $T(q)T(r) = T(qr)$ checks composition exactly, without numerical matrix exponentiation.

Any other complete packet family still has exact block convolution. For the coordinate groups A and B,

$$P_b e^{(s+t)Q} P_a = \sum_{c \in \{\mathrm{A},\mathrm{B}\}} (P_b e^{sQ} P_c)(P_c e^{tQ} P_a).$$

The intermediate packet cannot generally be omitted. Only a reducing family eliminates all cross terms.

For later discrete calculations define $K = I + Q/8$. This matrix is stochastic, but *it is not* $e^{Q/8}$. It is the step kernel in the uniformization identity $e^{tQ} = e^{-8t} \sum_{n \geq 0} (8t)^n K^n / n!$.

> **Read this as:** Changing packet coordinates does not change the evolution. Replacing continuous time by one discrete step does.

Plate VI.5 / EXACT LUMPABILITY / AGGREGATE COLUMNS

When two states are truly equivalent

Chapter VI, Part I: Theorem 5.1. The hypothesis is an intertwining identity.

Group states $0, 1$ into A and states $2, 3$ into B. The aggregation map and the candidate coarse kernel are

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \qquad \bar{K} = \begin{pmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}.$$

A fine state in A has probability $1/4$ of entering B in one step, regardless of which fine state in A was occupied. The analogous statement holds for B.

$8K$

	0	1	2	3
0	5	1	2	0
1	1	5	0	2
2	2	0	5	1
3	0	2	1	5

CK: aggregate transition columns

	0	1	2	3
A	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
B	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{3}{4}$

$$\bar{K} = \begin{pmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}$$

Fine trajectories differ. Coarse predictions agree.

Columns 0 and 1 agree; columns 2 and 3 agree.

The fine columns are visibly different. Their sums over each target packet agree within each source packet. That equality is the reason a two-state description works.

A one-step calculation controls every discrete horizon

The displayed columns give $CK = \bar{K}C$. Multiplication then yields

$$CK^n = \bar{K}^n C \qquad (n \geq 0).$$

For example, e_0 and e_1 have the same coarse initial vector $(1, 0)^{\mathsf{T}}$. Their fine evolutions differ, but their coarse vectors agree at every step. No choice of the within-A conditional distribution is needed for the prediction.

The continuous generator also closes:

$$CQ = \bar{Q}C, \qquad \bar{Q} = \begin{pmatrix} -2 & 2 \\ 2 & -2 \end{pmatrix}.$$

The Duhamel identity therefore gives $Ce^{tQ} = e^{t\bar{Q}}C$. This is a second consequence of the corresponding generator identity, not an inference that K is a fixed-time sample of Q.

Aggregation is still not invertible. The theorem says the discarded details are irrelevant for *these predictions*. It does not say they are irrelevant to every later likelihood. The observation counterexample on Plate VI.10 uses this exact same partition.

> **Read this as:** A partition is not automatically a valid Markov model. The aggregate transition columns must be constant within each source block.

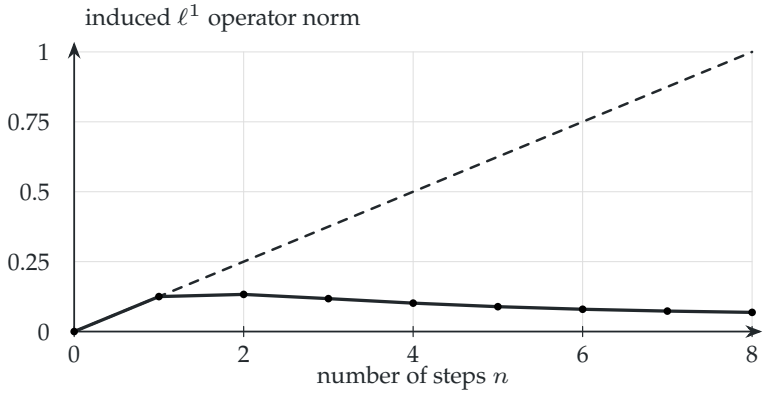
Plate VI.6 / DEFECTS / A GUARANTEE WITHOUT A FIT

A broken lump and its error budget

Chapter VI, Part I: Theorems 3.1, 3.3, and 5.2. Continuous and discrete defect identities.

Modify only column zero of K: move $1/16$ of its mass from target state zero to target state two. The perturbed matrix K_ε remains stochastic, but its aggregate column now differs from column one. Relative to the unchanged coarse kernel,

$$\Delta = CK_\varepsilon - \bar{K}C = \begin{pmatrix} -1/16 & 0 & 0 & 0 \\ 1/16 & 0 & 0 & 0 \end{pmatrix}, \qquad \|\Delta\|_1 = 1/8.$$



$\|\Delta\|_1 = 1/8$

solid: exact defect
dashed: $n/8$

The bound is a guarantee, not a fitted trend.

The solid values are computed with rational matrix powers. The dashed certificate uses only stochasticity and the exact one-step defect. It need not be sharp.

Telescoping tells us where every error originates

For $n \geq 0$,

$$CK_\varepsilon^n - \bar{K}^n C = \sum_{j=0}^{n-1} \bar{K}^{n-1-j} \Delta K_\varepsilon^j.$$

Each summand propagates one defect between an earlier fine evolution and a later coarse evolution. Column-stochastic matrices have induced ℓ^1 norm one, so the norm is at most $n/8$. The empty sum at $n = 0$ is zero.

The continuous counterpart is equally explicit. With fine generator -1, coarse generator -2, and $C = 1$, the generator defect is 1 and

$$e^{-t} - e^{-2t} = \int_0^t e^{-2(t-r)} e^{-r}\, dr.$$

Its resolvent version is $1/(z+1) - 1/(z+2) = 1/((z+1)(z+2))$ for $z > 0$.

Read this as: A small defect does not disappear. The identity specifies how it accumulates, and the norm assumptions turn that identity into a finite-horizon certificate.

Plate VI.7 / OBSERVABLES / WEIGHTED ORTHOGONALITY

Averaging is not summing

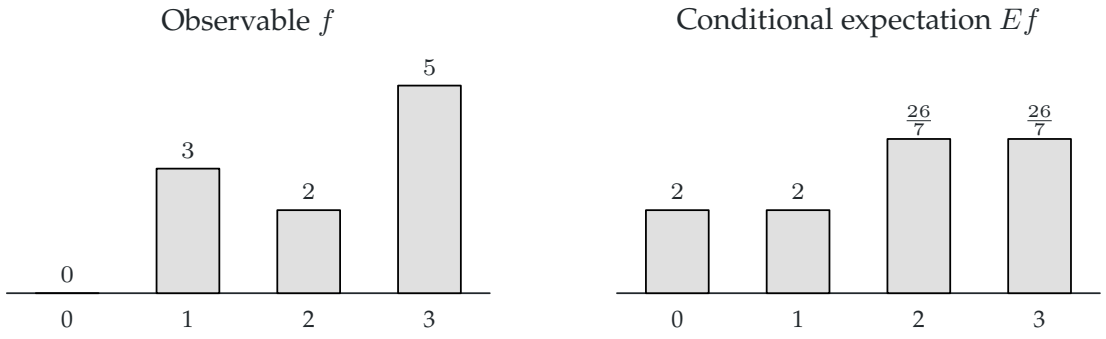
Chapter VI, Part I: Theorem 6.1. Strictly positive weights are part of the setting.

For this observable example, choose the fixed probability weights $\pi = (1, 2, 3, 4)/10$. They are not being identified with the stationary law of the earlier symmetric chain. On the same two-block partition, conditional expectation is

$$E = \begin{pmatrix} 1/3 & 2/3 & 0 & 0 \\ 1/3 & 2/3 & 0 & 0 \\ 0 & 0 & 3/7 & 4/7 \\ 0 & 0 & 3/7 & 4/7 \end{pmatrix}.$$

It replaces function values by their *weighted block average*. In particular, $f = (0, 3, 2, 5)^{\mathsf{T}}$ becomes $Ef = (2, 2, 26/7, 26/7)^{\mathsf{T}}$.

Weights $\pi = (1, 2, 3, 4)/10$; the block values are 2 and $26/7$.



The bars are function values, not state probabilities. Their total need not be one. The block masses $3/10$ and $7/10$ determine the denominators in the averaging formula.

Which inner product makes this an orthogonal projection?

Put $W = \mathrm{diag}(\pi)$. Direct multiplication checks

$$E^2 = E, \qquad WE = E^{\mathsf{T}}W, \qquad E\mathbf{1} = \mathbf{1}.$$

The middle equation is weighted self-adjointness. Ordinary symmetry fails: $E_{01} = 2/3$ whereas $E_{10} = 1/3$. Testing $E = E^{\mathsf{T}}$ would therefore reject a correct conditional-expectation projector.

The range consists exactly of functions constant on each block. The error $f - Ef$ has weighted mean zero inside every block, so it is weighted-orthogonal to every block-constant function. This is the finite projection interpretation of conditional expectation.

In contrast, C acts on probability columns by summing masses. The shapes of C and E differ, and their domains have different interpretations. The notation "coarsening" alone does not specify which operation is intended.

> **Read this as:** An exact weighted projector can be oblique in the unweighted Euclidean picture. Keep the measure and the inner product in the model.

Plate VI.8 / MARTINGALE PACKETS / THREE EXACT PIECES

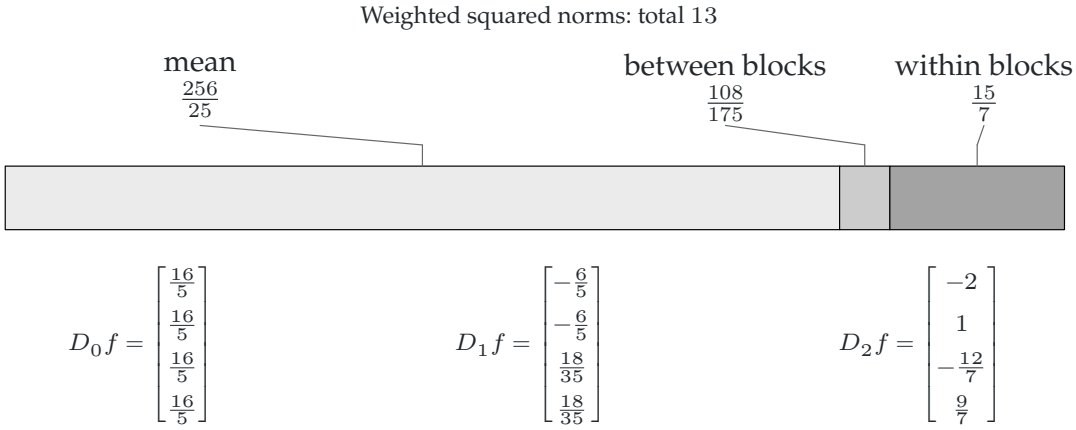
Where the variance goes

Chapter VI, Part I: Theorem 6.2. Nested weighted projections and their differences.

Keep the weights and observable from Plate VI.7. Let $E_0 f$ be the constant global mean, let $E_1 = E$ average within A and B, and let $E_2 = I$ retain every value. Define

$$D_0 = E_0, \qquad D_1 = E_1 - E_0, \qquad D_2 = I - E_1.$$

Then $D_0 f$ is the mean, $D_1 f$ describes the contrast between block means, and $D_2 f$ is the fluctuation left inside each block.



The strip lengths are proportional to the three exact weighted energies. The between-block component is small but not zero. The vectors below specify the corresponding directions, not only their sizes.

The energy identity follows from actual orthogonality

The tower identities give $E_0 E_1 = E_1 E_0 = E_0$. Together with weighted self-adjointness, these imply

$$D_i D_j = 0 \ (i \neq j), \qquad \sum_{j=0}^{2} D_j = I.$$

For the chosen f, the total weighted energy is $\|f\|_\pi^2 = 13$ and the mean is $16/5$. Consequently

$$\mathrm{Var}_\pi(f) = 13 - \left(\frac{16}{5}\right)^2 = \frac{69}{25} = \frac{108}{175} + \frac{15}{7}.$$

The first term is the energy of the between-block contrast, and the second is the weighted within-block variance. This is a decomposition of one observable in one fixed weighted space.

Changing the probability weights between levels would change the inner products and the averaging operators. The displayed orthogonal decomposition would no longer follow merely from nesting the sets. Likewise, choosing unrelated projections can preserve the telescope $\sum D_j = I$ while losing orthogonality and the energy sum.

> **Read this as:** The differences between nested conditional expectations are not just bookkeeping. They are orthogonal, separately measurable parts of the original signal.

Plate VI.9 / BAYESIAN TRANSFER / AN EXACT POSTERIOR

Evidence changes shape and mass

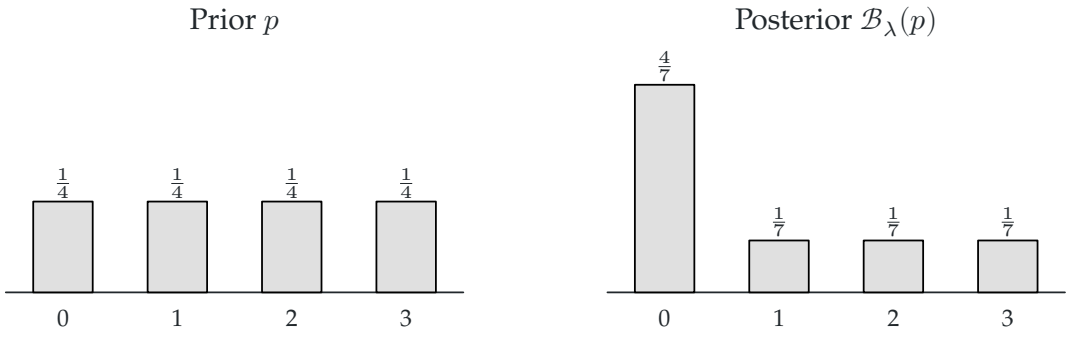
Chapter VI, Part I: Theorems 7.1–7.3. Normalization, packet reconstruction, and positive evidence.

Start with $p = (1,1,1,1)^{\mathsf{T}}/4$ and likelihood weights $\lambda = (4,1,1,1)^{\mathsf{T}}$. The unnormalized vector is $(1,1/4,1/4,1/4)^{\mathsf{T}}$, its evidence is $Z = 7/4$, and

$$\mathcal{B}_\lambda(p) = \frac{(4,1,1,1)^{\mathsf{T}}}{7}.$$

Likelihood weights may be rescaled, so an evidence larger than one is not a probability claim. Multiplying all likelihoods by one common positive factor leaves this posterior unchanged.

$\lambda = (4,1,1,1)$; evidence $Z = 7/4$



The A packet becomes heavier and its internal shape changes. The B packet retains an equal internal split but loses total mass. These are different effects of the same likelihood.

Reconstruct from packet mass times packet shape

The unnormalized packet evidences are $Z_{\mathrm{A}} = 5/4$ and $Z_{\mathrm{B}} = 1/2$. The resulting masses and conditional shapes are

$$m_{\mathrm{A}} = 5/7, \quad q_{\mathrm{A}} = (4/5, 1/5, 0, 0)^{\mathsf{T}}, \qquad m_{\mathrm{B}} = 2/7, \quad q_{\mathrm{B}} = (0, 0, 1/2, 1/2)^{\mathsf{T}}.$$

Thus $m_{\mathrm{A}} q_{\mathrm{A}} + m_{\mathrm{B}} q_{\mathrm{B}}$ is exactly the displayed posterior. Keeping only the masses would not reconstruct the fine state.

There is no need to divide by a zero packet evidence. For likelihood $(0,0,1,1)$, packet A has mass zero and contributes the zero vector. Only the occupied B packet has a conditional probability vector. If the *total* evidence is zero, however, the Bayesian update is not defined and the implementation rejects the input.

> **Read this as:** Bayesian reconstruction uses two kinds of data: how much posterior mass a packet carries and how that mass is distributed inside the packet.

Plate VI.10 / SUFFICIENCY / A COMPUTABLE FAILURE WITNESS

Two priors that fool the coarse model

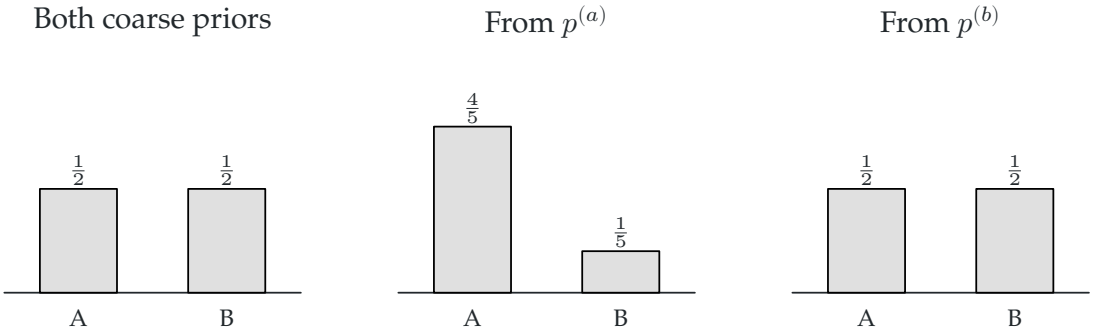
Chapter VI, Part I: Theorem 8.2 and the one-packet exception.

Keep the same likelihood $\lambda = (4, 1, 1, 1)^{\mathsf{T}}$ and partition A$= \{0, 1\}$, B$= \{2, 3\}$. Compare

$$p^{(a)} = (1/2, 0, 1/2, 0)^{\mathsf{T}}, \qquad p^{(b)} = (0, 1/2, 1/2, 0)^{\mathsf{T}}.$$

Both have coarse prior $(1/2, 1/2)^{\mathsf{T}}$. A procedure given only that two-entry vector cannot distinguish them.

Hidden redistribution changes the packet-A posterior from $1/2$ to $4/5$.



After the identical fine likelihood is applied, the coarse posteriors are $(4/5, 1/5)$ and $(1/2, 1/2)$. A single deterministic update of the common coarse prior cannot produce both.

Why this witnesses the necessity of blockwise constancy

The A evidence for $p^{(a)}$ is 2, whereas it is $1/2$ for $p^{(b)}$. Their B evidence is $1/2$ in both cases. The posterior odds therefore distinguish the hidden distribution inside A.

For strictly positive fine and coarse likelihoods, and at least two coarse blocks, the manuscript proves the all-prior criterion

$$C\mathcal{B}_\lambda(p) = \mathcal{B}_{\bar{\lambda}}(Cp) \quad \text{for every } p \quad \Longleftrightarrow \quad \lambda_i = c\bar{\lambda}_a \quad (i \in I_a)$$

for one global $c > 0$. The global scale is invisible after normalization. A different arbitrary scale for each packet is not invisible: it changes posterior odds.

If all states are merged into a single packet, every normalized posterior coarsens to (1). That degenerate commutation holds for every likelihood and cannot identify any within-packet information. It is explicitly excluded from the nontrivial criterion.

> **Read this as:** A prior-dependent effective likelihood can always summarize the update. A fixed likelihood that works for every prior is a much stronger requirement.

Plate VI.11 / BAYESIAN DEFECT / MORE THAN AN ERROR BAR

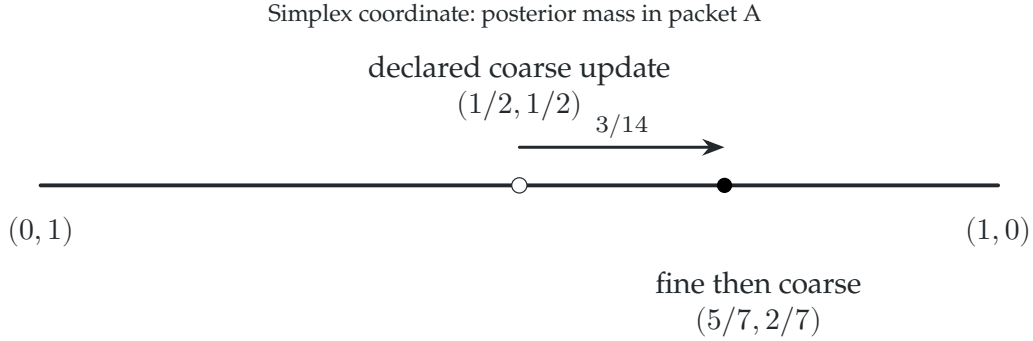
The exact information-loss vector

Chapter VI, Part I: Theorems 8.1 and 8.4, and Corollary 8.6.

For a prior p, let $P = Cp$ and let $r_a(p)$ be the conditional average of the fine likelihood inside packet a. For the uniform prior and $\lambda = (4, 1, 1, 1)$,

$$P = (1/2, 1/2)^{\mathsf{T}}, \qquad r = (5/2, 1)^{\mathsf{T}}.$$

The exact coarse posterior is $\mathcal{B}_r(P) = (5/7, 2/7)^{\mathsf{T}}$. Suppose instead that the declared coarse likelihood is $\bar{\lambda} = (1, 1)$.



The posterior error lies along the mass-preserving direction of the two-state simplex. Its entries are opposite because both endpoints have total mass one.

Compute the factorization, including the null direction

Here $\bar{b} = \mathcal{B}_{\bar{\lambda}}(P) = P$, $\delta = r - \bar{\lambda} = (3/2, 0)^{\mathsf{T}}$, and $Z_r = 7/4$. Hence

$$\frac{1}{Z_r}\left(\operatorname{diag}(P) - \bar{b}P^{\mathsf{T}}\right)\delta = \frac{4}{7}\begin{pmatrix} 1/4 & -1/4 \\ -1/4 & 1/4 \end{pmatrix}\begin{pmatrix} 3/2 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/14 \\ -3/14 \end{pmatrix}.$$

This is the actual fine-to-coarse discrepancy, not a fitted approximation. The matrix kills $\bar{\lambda}$, recording the common-scale null direction. Its ℓ^1 error is $3/7$.

For a family of likelihoods, states may be grouped by equality of their *whole likelihood profile*. The family $(3, 3, 1, 1)$ and $(1, 2, 1, 2)$ separates all four states even though either likelihood alone permits a coarser grouping. The common refinement is therefore the discrete partition. The theorem's one-packet exception still remains.

> **Read this as:** The defect factors through exactly the conditional evidence information missing from the declared coarse model. A new observation family can require a finer partition.

Plate VI.12 / BAYESIAN FLOW / COMPOSITION WITHOUT LINEARITY

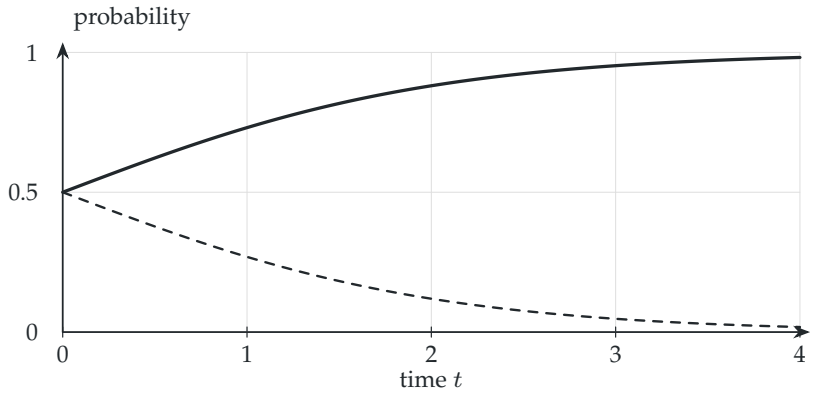
A nonlinear semigroup of odds

Chapter VI, Part I: Theorems 7.1 and 9.1. Likelihood composition and the replicator field.

Let the score vector be $g = (1, 0)$ and start at $(1/2, 1/2)$. The likelihood at time t is $(e^t, 1)$, so the first posterior coordinate is

$$p_0(t) = \frac{e^t}{1 + e^t}, \qquad p_1(t) = \frac{1}{1 + e^t}.$$

The odds p_0/p_1 grow exponentially, while the probabilities remain inside the simplex. At $t = \log 2$ the posterior is exactly $(2/3, 1/3)$.



$$p_0(t) = \frac{e^t}{1 + e^t}$$

$$p_1(t) = \frac{1}{1 + e^t}$$

$$\dot{p}_0 = p_0(1 - p_0)$$

Mass stays one.
Mixtures are not preserved.

The solid and dashed curves sum to one. The curvature is caused by the shared evidence denominator, not by a nonlinear likelihood multiplication law.

Two properties that must not be conflated

If $a > 0$ multiplies the odds, the first-coordinate map is

$$\phi_a(x) = \frac{ax}{1 + (a - 1)x}.$$

Direct algebra gives $\phi_a(\phi_b(x)) = \phi_{ab}(x)$. Thus $a = e^t$ gives a nonlinear semigroup. Nevertheless, $\phi_3(1/2) = 3/4$, while the mean of $\phi_3(0)$ and $\phi_3(1)$ is $1/2$. Superposition fails.

Differentiation of the normalized flow yields the centered score equation

$$\dot{p}_i = (g_i - \langle g, p \rangle) p_i.$$

For the displayed model, $\dot{p}_0 = p_0(1 - p_0)$. The negative average-score term is what keeps the total derivative zero. Omitting the derivative of the normalizer would not preserve mass.

Likelihood composition also preserves the evidence identity: the first evidence times the second conditional evidence equals the combined evidence. Normalization cancels in the posterior, but that factor still matters when comparing data models.

> **Read this as:** A semigroup is a composition law in time. It does not imply that the maps are linear or that they preserve mixtures.

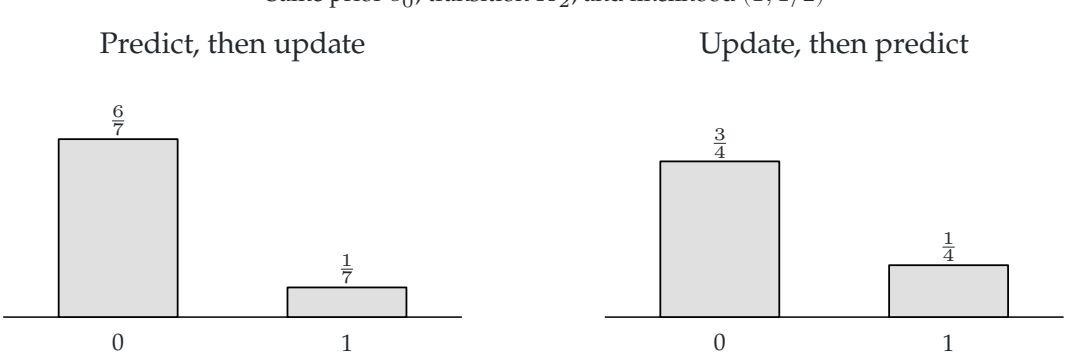
Plate VI.13 / HIDDEN MARKOV FILTERING / TIME ORDER

Prediction and observation do not commute

Chapter VI, Part I: Proposition 9.3. Linear prediction followed by a positive-evidence update.

Use the two-state kernel K_2 from Plate VI.2, the initial distribution e_0, and likelihood $(1, 1/2)$. Predicting first gives $(3/4, 1/4)$. Its likelihood-weighted vector is $(3/4, 1/8)$, with evidence $7/8$, so the posterior is $(6/7, 1/7)$.

Updating first leaves e_0 unchanged. Predicting afterward gives $(3/4, 1/4)$ instead. The order of operations expresses which time the observation concerns.



Both routes produce valid probability vectors. Their difference is therefore not a normalization bug. It is a different conditioning problem.

Keep an unnormalized path alongside the filter

For a sequence of observations, let $D_r = \operatorname{diag}(\lambda^{(r)})$. The forward recursion is

$$\hat{p}_r = K p_{r-1}, \qquad p_r = D_r \hat{p}_r / Z_r.$$

Its unnormalized companion is

$$v_r = D_r K \cdots D_1 K p_0, \qquad p_r = \frac{v_r}{\mathbf{1}^{\mathsf{T}} v_r}, \qquad \mathbf{1}^{\mathsf{T}} v_r = \prod_{j=1}^{r} Z_j.$$

The product of evidences records the normalization lost at each individual step. The four-state computation on Plate VI.26 checks both paths independently, including their coarse versions.

Here $D_r K$ means an observation after prediction. In the Feynman–Kac example on Plate VI.20, departure-state weighting has forward operator $K D_w$. Both conventions are legitimate once their meanings are fixed, but they cannot be substituted for one another.

> **Read this as:** Write the chronological sequence before multiplying matrices. A product can be normalized correctly and still answer the wrong time-ordered question.

Plate VI.14 / BETA MODEL / POSTERIOR AND EVIDENCE

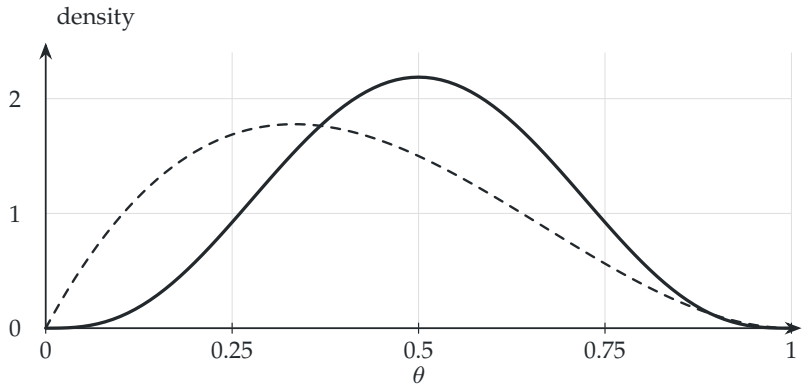
Conjugacy tracks exponents

Chapter VI, Part I: Example 9.2. A finite-parameter conjugate family, not a universal posterior model.

For a small exact example, take the Beta$(2, 3)$ density

$$\rho(\theta) = 12\theta(1-\theta)^2, \qquad 0 \le \theta \le 1.$$

Observe a specified ordered sequence with two successes and one failure. Its likelihood is $\theta^2(1-\theta)$. Multiplication adds exponents, so the posterior belongs to Beta$(4, 4)$ with density $140\theta^3(1-\theta)^3$.



The dashed prior is asymmetric. The solid posterior is symmetric with mean $1/2$. Polynomial integration verifies the normalizers exactly. The curves are sampled drawings of those polynomials.

The observation convention changes the evidence

The integral of the unnormalized posterior is

$$\int_0^1 12\theta^3(1-\theta)^3\,d\theta = \frac{3}{35}.$$

Dividing by $3/35$ gives the displayed normalized density. If the observation is instead the *count* of two successes in three trials, there are $\binom{3}{2} = 3$ compatible ordered sequences. The count likelihood is therefore $3\theta^2(1-\theta)$ and its evidence is $9/35$.

The extra factor does not depend on θ, so it cancels from the posterior. The two experiments thus have the same posterior distribution but different data probabilities. Evidence is not determined by the final normalized density alone.

The exact check also integrates both densities to one and integrates θ against the posterior to obtain $1/2$. This demonstrates one conjugate family in elementary terms. It does not show that arbitrary likelihoods preserve that family or that all posterior information can be compressed into two exponents.

Read this as: Conjugacy gives a closed parameter update when a specified likelihood respects a specified family. Preserve the evidence as well as the normalized posterior.

Plate VI.15 / FRACTIONAL DIFFUSION / NEW COUPLINGS

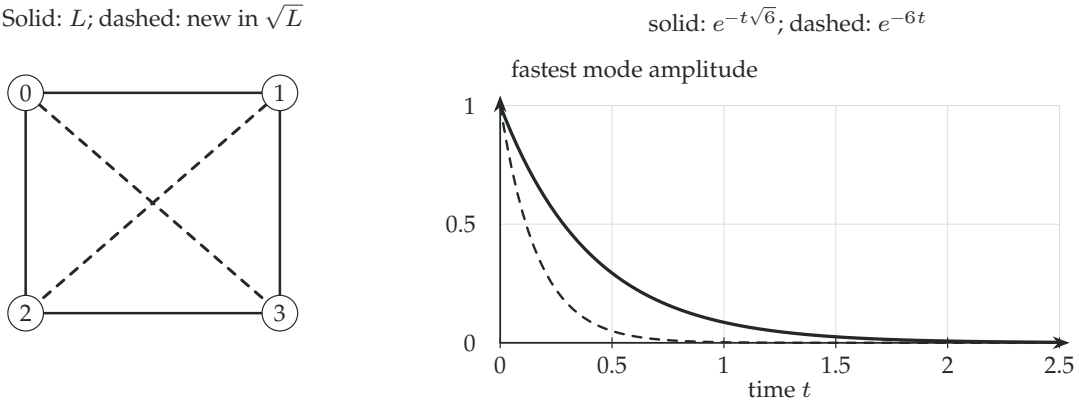
The modes stay as their clocks change

Chapter VI, Part I: Theorem 10.1. A positive self-adjoint finite model and its spectral powers.

Return to the four-state generator on Plate VI.4 and put $L = -Q$. It is real symmetric and nonnegative, with orthogonal projectors P_0, P_1, P_2, P_3 and eigenvalues $0, 2, 4, 6$. Its square root is

$$L^{1/2} = \sqrt{2}P_1 + 2P_2 + \sqrt{6}P_3, \qquad (L^{1/2})^2 = L.$$

Both heat evolutions retain these modes. Ordinary diffusion multiplies mode j by e^{-2jt}. Square-root diffusion multiplies it by $e^{-\sqrt{2j}\,t}$. The constant mode is unchanged in both.



The highest mode decays at rates 6 and $\sqrt{6}$. The graph beside the curves shows the additional couplings in the fractional operator. Curves display the explicit formulas, not an empirical fit.

Fractional does not mean entrywise

Although $L_{03} = 0$, its spectral square root satisfies

$$(L^{1/2})_{03} = \frac{\sqrt{6} - \sqrt{2} - 2}{4} < 0.$$

Thus $-L^{1/2}$ permits a direct transition between states 0 and 3, which were not neighbors for Q. Taking the square root of each matrix entry would neither produce this value nor square back to L.

The exact block aggregation remains compatible in this example:

$$CL^{1/2} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} C.$$

The coarse matrix is the positive square root of $-Q_c$. This is a concrete naturality identity, checked on the full matrices. It is not a claim that an arbitrary partition commutes with every fractional construction.

The computation uses a nonnegative self-adjoint operator and a specified spectral branch. General nonnormal matrices, unbounded domains, and positivity of other fractional evolutions need their own hypotheses.

> **Read this as:** Fractional powers alter modal decay rates. Reassembling those modes can introduce long-range spatial interactions even when the original generator was sparse.

Plate VI.16 / FRACTIONAL DEFECTS / NEAR ZERO VERSUS AWAY FROM ZERO

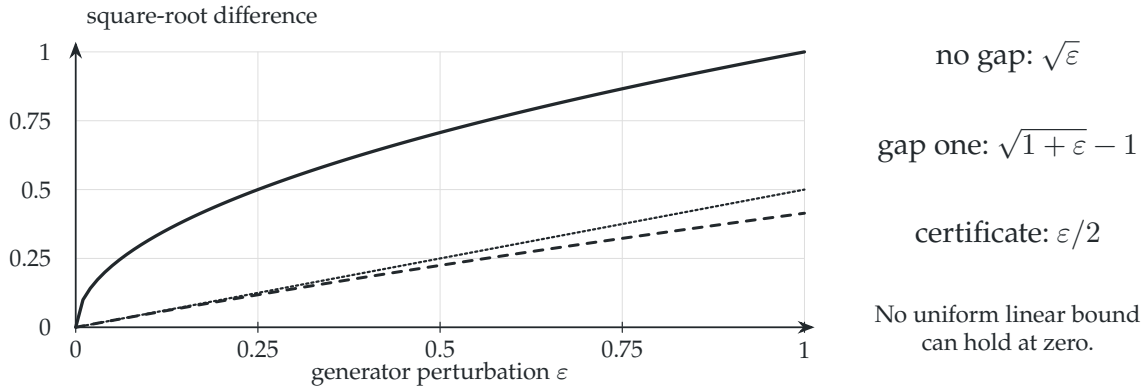
A gap changes the error scale

Chapter VI, Part I: Theorem 10.3 and Corollary 10.4. Separate generator error from fractional error.

The function $a \mapsto \sqrt{a}$ changes most rapidly near zero. A scalar model already shows why a spectral gap appears in fractional-defect estimates:

$$|\sqrt{\varepsilon} - \sqrt{0}| = \sqrt{\varepsilon}, \qquad \sqrt{1+\varepsilon} - 1 = \frac{\varepsilon}{\sqrt{1+\varepsilon}+1} \le \frac{\varepsilon}{2}.$$

The same input difference has a square-root scale at zero and a linear bound above a fixed positive lower bound.



The upper curve compares 0 with ε. The lower curve compares 1 with $1+\varepsilon$. The dotted line $\varepsilon/2$ bounds the latter. All plotted values come from the displayed scalar formulas.

Read an operator bound with its hypotheses

For positive self-adjoint finite matrices $A, B \ge mI$ and $0 < \alpha < 1$, the manuscript's gapped estimate has the form

$$\|CA^{\alpha} - B^{\alpha}C\| \le \alpha m^{\alpha-1} \|CA - BC\|.$$

It concerns an intertwining defect, not just two eigenvalue lists. With $C = 1$, $A = 4$, $B = 9$, $m = 4$, and $\alpha = 1/2$, the generator defect has size 5, the fractional defect has size 1, and the bound is $5/4$.

The common zero mode of a Markov generator cannot simply be included in a positive-gap estimate. One must either work on an appropriate gapped subspace, handle the zero mode separately, or use the gapless estimate. In particular, a good estimate on each finite level need not have a constant uniform in the refinement level.

On Plate VI.27, the sequences $1/n^2$ and $1/n$ turn this change in error scale into a concrete distinction between summable and nonsummable defects.

Read this as: A fixed gap is quantitative information. It prevents the derivative of the fractional function from becoming singular along the comparison.

Plate VI.17 / SUBORDINATION / A DISTRIBUTION OF ORDINARY TIMES

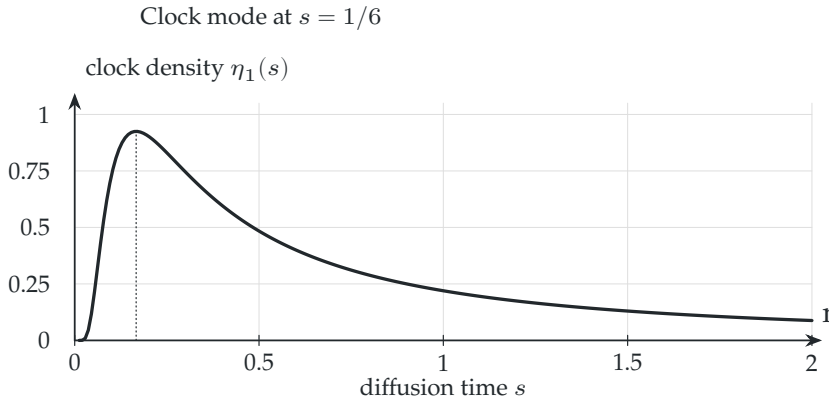
A random diffusion clock

Chapter VI, Part I: Theorem 10.2. The Poisson subordination formula for the finite positive model.

Square-root evolution can be written as an average of ordinary diffusion over a positive clock variable s:

$$e^{-t\sqrt{L}} = \int_0^\infty \eta_t(s) e^{-sL}\, ds, \qquad \eta_t(s) = \frac{t}{2\sqrt{\pi}} s^{-3/2} e^{-t^2/(4s)}, \quad t > 0.$$

Here t is the external time, and s is the ordinary diffusion time being mixed. They are not interchangeable variables.



$$\int_0^\infty \eta_t(s)\, ds = 1$$

$$\int_0^\infty s\eta_t(s)\, ds = \infty$$

A probability clock need not have a finite mean.

The density η_1 peaks at $s = 1/6$ but has a long right tail. The drawing ends at $s = 2$. It does not claim that the missing tail has zero mass.

Check the normalizer before interpreting a density

Under $s = t^2/(4u^2)$, reversing the integration limits gives

$$\eta_t(s)\,|ds| = \frac{2}{\sqrt{\pi}} e^{-u^2}\, du.$$

Its integral over $u > 0$ is one, so η_t is a probability density. On each eigenmode of L, the operator formula reduces to the scalar Laplace identity

$$\int_0^\infty \eta_t(s) e^{-\lambda s}\, ds = e^{-t\sqrt{\lambda}}.$$

The manuscript supplies that scalar identity and its spectral assembly. The appendix separately checks the change of variables and normalization. Sampled curves are not used to certify an improper integral.

A normalized density need not have a finite mean. For large s, the exponential factor tends to one and $\eta_t(s)$ is proportional to $s^{-3/2}$. Consequently $s\eta_t(s)$ has a nonintegrable $s^{-1/2}$ tail. The clock has infinite mean even though its total mass is one.

> **Read this as:** Subordination is a weighted mixture of diffusion times. Replacing the clock distribution by a single average time is not justified, and here that average is infinite.

Plate VI.18 / DIRICHLET GRID / THE MISSING FACTOR OF h

A Green kernel is an inverse with units

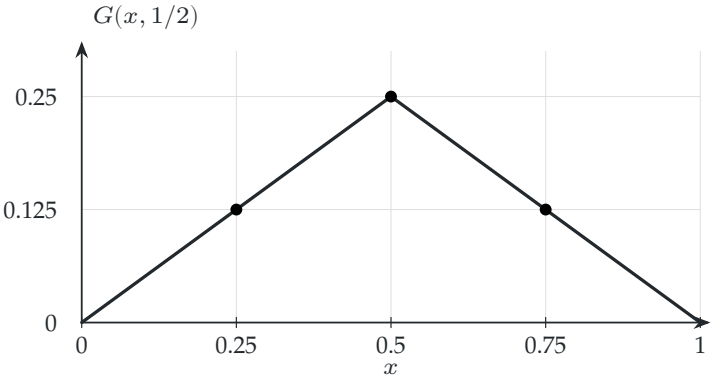
Chapter VI, Part I: Theorem 11.1. Exact nodal Green reconstruction for this one-dimensional stencil.

Take three interior nodes $x_i = i/4$ on $[0, 1]$, with zero boundary values and $h = 1/4$. The discrete negative second derivative is

$$L_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

For the continuous Dirichlet problem, the Green function is $G(x, y) = \min(x, y) - xy$. At these nodes the exact comparison is

$$L_h^{-1} = h\,[G(x_i, x_j)]_{i,j=1}^{3} = \frac{1}{64}\begin{pmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{pmatrix}.$$



$64L_h^{-1}$, $h = 1/4$

3	2	1
2	4	2
1	2	3

$L_h^{-1} = hG_{\text{nodes}}$

The tent is the central column of the continuous Green kernel. It has nodal values $(1, 2, 1)/8$. Its discrete inverse column is $(1, 2, 1)/32$, smaller by the factor h.

A forcing vector is not a quadrature-weighted measure

For sampled forcing f_j, the finite solution is

$$u_i = \sum_j (L_h^{-1})_{ij} f_j = h \sum_j G(x_i, x_j) f_j.$$

The factor h converts samples into the quadrature weights appropriate to this formula. Dropping it multiplies the computed solution by four on the displayed grid.

A discrete unit coordinate vector and a discretized unit point mass use different normalizations. For $f = e_2$, the result is the second inverse column. For $f = e_2/h$, it is the unscaled tent sampled at the nodes. Either convention can be useful, but it must be specified before comparing answers.

Exact checks multiply L_h by the proposed inverse at $1, 3, 7$, and 15 interior nodes. The general identity is the manuscript theorem. These finite checks do not extend it to arbitrary meshes, dimensions, or boundary conditions.

> **Read this as:** An inverse kernel carries a measure convention. A tent that looks correct can give a wrong solution if the grid weight is omitted.

Plate VI.19 / FINITE DIFFERENCES / A QUARTIC TEST PROBLEM

Convergence with a nonzero exact error

Chapter VI, Part I: Theorems 11.1–11.2. Exact nodal inversion and the scale of low-mode errors.

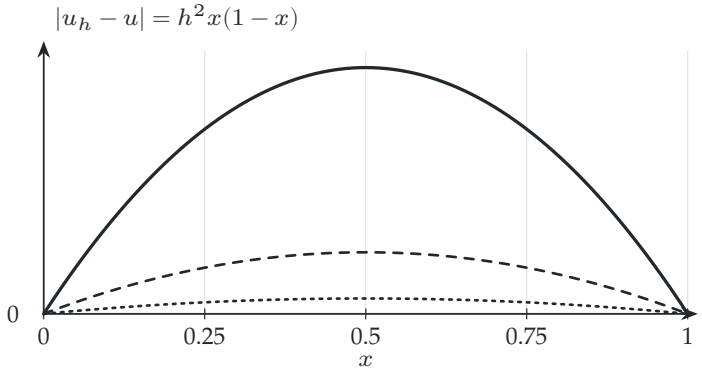
Use $u(x) = x - x^4$, which vanishes at both endpoints and solves $-u'' = 12x^2$. Its centered second difference is not exact:

$$L_h u|_{\text{nodes}} = 12x_i^2 + 2h^2.$$

Since the stencil differentiates the quadratic $x(1-x)$ exactly, the solution u_h to the sampled forcing $12x_i^2$ satisfies

$$\boxed{u_h(x_i) - u(x_i) = -h^2 x_i(1-x_i).}$$

This gives both the sign and size of the error, rather than merely a log–log slope.



grid spacing h	maximum error
$\frac{1}{4}$	$\frac{1}{64}$
$\frac{1}{8}$	$\frac{1}{256}$
$\frac{1}{16}$	$\frac{1}{1024}$

The three curves show magnitudes of the exact nodal errors for $h = 1/4, 1/8, 1/16$. Lines between grid points are visual guides, not an additional continuous numerical solution. The error is negative on every interior node.

One refinement divides this error by four

All three grids include $x = 1/2$, where the maximum absolute nodal error is $h^2/4$. It equals $1/64$, $1/256$, and $1/1024$, respectively. Exact rational matrix inversion verifies these values independently of the closed error formula.

The manuscript also compares the discrete eigenvalues

$$\lambda_{k,h} = 4h^{-2}\sin^2(k\pi h/2)$$

with $(k\pi)^2$, with error at most $(k\pi)^4 h^2/12$. For a fixed mode k, the bound tends to zero quadratically. It does not claim a uniform small error for modes whose index grows like $1/h$.

The nodal Green identity is exact, while the discretization of the differential equation still has a truncation error. These statements concern different comparisons and do not contradict one another.

> **Read this as:** An exact inverse of the discrete operator does not make that operator equal to the differential operator. Keep algebraic inversion and continuum approximation separate.

Plate VI.20 / FEYNMAN–KAC / DEPARTURE-STATE WEIGHTS

Eight paths, one weighted expectation

Chapter VI, Part I: Theorem 12.1. A finite path sum with its time convention displayed.

Let $K_2 = \frac{1}{4}\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$, start at state 0, use weights $w = (1, 1/2)$, and terminal observable $f = (2, 1)$. For three transitions, there are eight possible state sequences after the fixed initial state. We evaluate

$$\mathbb{E}_0\left[\prod_{r=0}^{2} w(X_r)\, f(X_3)\right].$$

Each path contributes its transition probability times its three departure weights times its terminal value.

Path	probability	weight	terminal	contribution
0000	$27/64$	1	2	$27/32$
0001	$9/64$	1	1	$9/64$
0010	$3/64$	$1/2$	2	$3/64$
0011	$9/64$	$1/2$	1	$9/128$
0100	$3/64$	$1/2$	2	$3/64$
0101	$1/64$	$1/2$	1	$1/128$
0110	$3/64$	$1/4$	2	$3/128$
0111	$9/64$	$1/4$	1	$9/256$

Total probability $= 1$; weighted terminal expectation $= 311/256$.

Every possible path is included. The probability column sums to one. The contribution column sums to $311/256$, agreeing exactly with the transfer calculation.

Check which side of the transition gets weighted

The backward operator on observables is $D_w K_2^{\mathsf{T}}$. Therefore the expectation is the zeroth coordinate of

$$(D_w K_2^{\mathsf{T}})^3 f.$$

Its forward dual is $K_2 D_w$: weight the mass at the departing state, then move it. Weighting arrival states instead gives $D_w K_2$, a different expression. The exact tests deliberately check that the two conventions do not agree in this example.

The answer $311/256$ exceeds one, but it is not a probability. The terminal observable can equal two. With $0 \le w_i \le 1$, the weighted expectation is bounded by the maximum terminal value, not necessarily by one.

This calculation certifies a finite three-step identity for a stated model. The continuous-time theorem uses the corresponding infinitesimal generator and potential, with the direction of action kept consistent.

> **Read this as:** A Feynman–Kac weight changes the value assigned to each path. Its location relative to a transition is part of the model, not a typographical choice.

Plate VI.21 / SCHUR–FESHBACH / RETAINED RESOLVENT

Elimination makes a frequency-dependent term

Chapter VI, Part I: Theorem 13.1. A two-state Schur complement with the inverse checked directly.

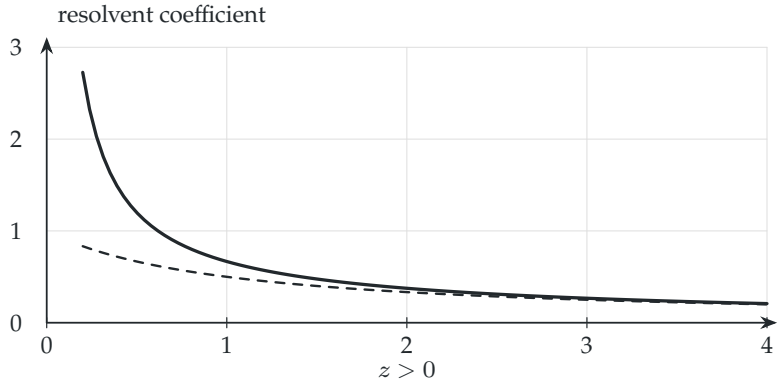
The two-state evolution from Plate VI.3 has generator

$$A = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Retain the first coordinate and eliminate the second. For z away from the relevant poles, the retained entry of $(zI - A)^{-1}$ is

$$R_x(z) = \left[z + 1 - \frac{1}{z+1} \right]^{-1} = \frac{z+1}{z(z+2)}.$$

The term $\Sigma(z) = 1/(z+1)$ is the hidden coordinate's self-energy.



exact: $\dfrac{z+1}{z(z+2)}$

omit hidden state: $\dfrac{1}{z+1}$

The pole at zero survives through hidden feedback.

The full inverse gives a rational response with two poles, at 0 and -2. Its partial fractions are $R_x(z) = \frac{1}{2}z^{-1} + \frac{1}{2}(z+2)^{-1}$. The hidden intermediate inverse has its own excluded point $z = -1$.

An eliminated variable has not disappeared

If the coupling to the second coordinate were simply deleted, the retained resolvent would be $(z+1)^{-1}$. That has a different pole and a different long-time limit. The correction is frequency dependent, so it cannot be represented by one constant scalar generator on the retained coordinate.

The Schur expression first requires $z + 1 \neq 0$, as well as invertibility of the full matrix. Its simplified rational form has a removable intermediate singularity at $z = -1$, where the retained entry of the full inverse is well-defined and equals zero. Simplification does not retroactively make the eliminated block invertible there.

For $z > 0$, the Laplace transform of $(1 + e^{-2t})/2$ is exactly $R_x(z)$. The next page derives the time-domain equation producing that response.

> **Read this as:** The Schur complement is exact elimination, not independent dynamics on a smaller state space. Hidden evolution returns as a z-dependent self-energy.

Plate VI.22 / EXACT RETAINED DYNAMICS / NOT A SCALAR SEMIGROUP

Hidden state returns as memory

Chapter VI, Part I: Theorem 13.2. The retained-state memory identity, including hidden initial data.

Write the same system as

$$x' = -x + y, \qquad y' = x - y.$$

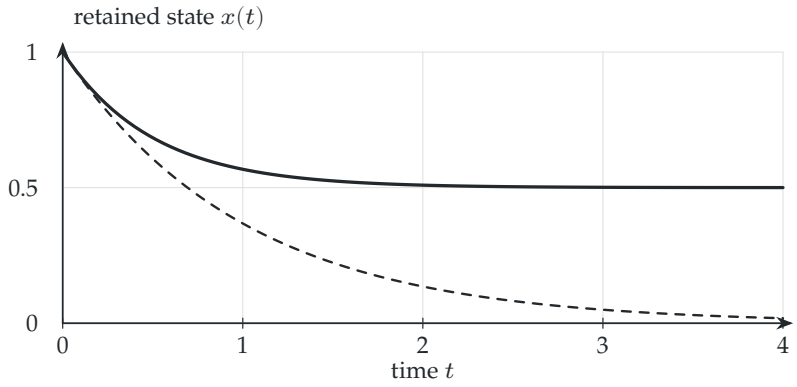
Solving the second equation first gives

$$y(t) = e^{-t}y(0) + \int_0^t e^{-(t-r)}x(r)\,dr.$$

Substitution is exact:

$$x'(t) = -x(t) + e^{-t}y(0) + \int_0^t e^{-(t-r)}x(r)\,dr.$$

The hidden initial state produces a separate forcing term. It must be retained even when the memory kernel is known.



exact: $\dfrac{1 + e^{-2t}}{2}$

naive deletion: e^{-t}

$K(t) = e^{-t}$
returns hidden excursions.

For $(x(0), y(0)) = (1, 0)$, the exact retained coordinate is $(1 + e^{-2t})/2$. The dashed curve e^{-t} solves the equation obtained by deleting the hidden coordinate rather than eliminating it.

Projection can destroy the semigroup law

Let $R(t) = (1 + e^{-2t})/2$, the response to this fixed initial preparation. At $t = \log 2$,

$$R(t) = \frac{5}{8}, \qquad R(2t) = \frac{17}{32}, \qquad R(2t) - R(t)^2 = \frac{9}{64}.$$

The full two-state propagator is a semigroup, but this retained scalar response is not. Restarting it as a scalar model silently resets the hidden coordinate, which generally carries information from the first interval.

The appendix verifies the full differential equation, initial condition, integral memory equation, and Laplace resolvent in this model. Its kernel is e^{-t}, whose transform is the self-energy on Plate VI.21.

Read this as: Exact reduction retains a record of the eliminated state through convolution and initial forcing. A memoryless approximation must justify the information it removes.

Plate VI.23 / NONDIAGONAL HIDDEN DYNAMICS / POLYNOMIAL EXPONENTIALS

A repeated pole remembers a Jordan chain

Chapter VI, Part I: Theorem 13.2 and Section 15. Functional calculus must retain nilpotent data.

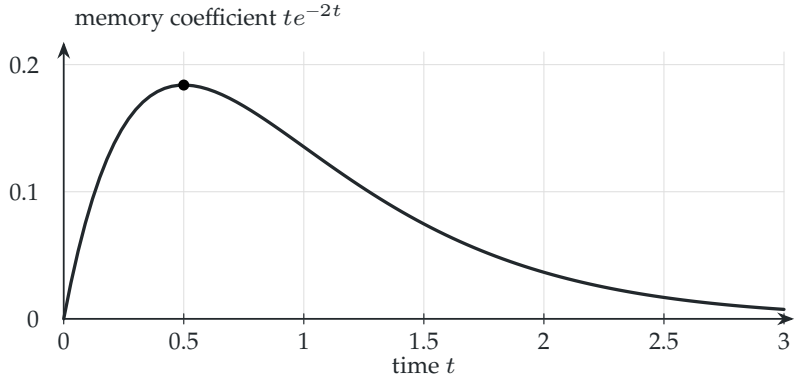
Now use a two-dimensional hidden block

$$J = -2I + N, \qquad N = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad N^2 = 0.$$

Choose the entrance column $V = (0,1)^{\mathsf{T}}$ and exit row $U = (1,0)$. The hidden exponential is not $e^{-2t}I$:

$$e^{tJ} = e^{-2t}(I + tN), \qquad K_{\text{mem}}(t) = Ue^{tJ}V = te^{-2t}.$$

The nilpotent link carries an input from the second hidden coordinate to the first.



$$J = -2I + N, \quad N^2 = 0$$

$$e^{tJ} = e^{-2t}(I + tN)$$

$$\widehat{K}(z) = \frac{1}{(z+2)^2}$$

A repeated pole carries a jet.

The memory kernel starts at zero, becomes positive, and peaks at $t = 1/2$. A rule that retained the eigenvalue -2 but discarded the Jordan chain would predict the zero kernel for these entrance and exit maps.

The transfer function sees pole order

The exact transform is

$$U(zI - J)^{-1}V = \frac{1}{(z+2)^2} = \int_0^\infty te^{-(z+2)t}\,dt, \qquad \operatorname{Re} z > -2.$$

The repeated pole and the polynomial factor t are the same information in two domains. For a longer nilpotent chain, higher powers of t occur with factorial denominators.

A zero value $K_{\text{mem}}(0)$ does not make the entire kernel zero. Nor do stable eigenvalues alone specify every transient contribution: entrance maps, exit maps, and nilpotent structure matter. Here the full kernel is nevertheless elementary, so its peak and transform can be checked without numerical fitting.

This page is an explicit finite realization. It does not show that an arbitrary continuum memory kernel admits a finite-dimensional representation of this form.

> **Read this as:** Repeated eigenvalues may carry directional information through Jordan chains. Keep the finite jet of the functional calculus, not just its values on the spectrum.

Plate VI.24 / PHASE-TYPE SURVIVAL / TWO SUCCESSIVE STAGES

A zero initial hazard can still lead to certain exit

Chapter VI, Part I: Theorem 14.1. Survival, density, hazard, and moments have distinct meanings.

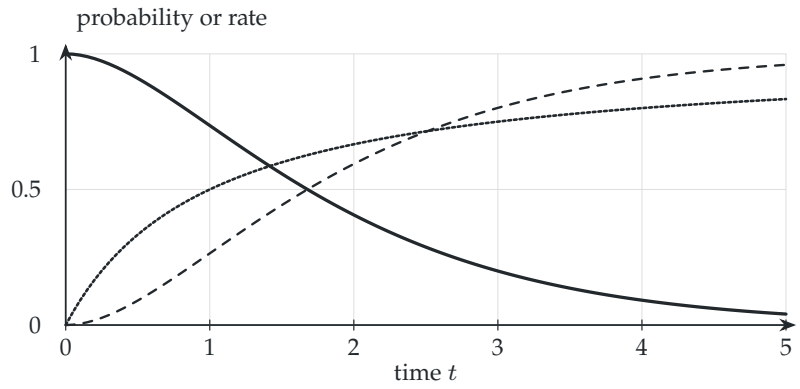
Let a transient process move through two successive rate-one stages, then leave the system. With column probabilities, the transient generator and starting vector are

$$T = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}, \qquad p(0) = e_0.$$

Thus $p(t) = e^{-t}(1, t)^{\mathsf{T}}$. Its total remaining mass, exit density, and hazard are

$$S(t) = (1 + t)e^{-t}, \qquad f(t) = te^{-t}, \qquad h(t) = \frac{t}{1 + t}.$$

The initial hazard is zero because an exit needs two stages, not one.



Survival falls from one to zero, while absorbed mass $1 - S$ rises to one. The dotted hazard is an instantaneous rate, not a probability, even though it shares the displayed numerical scale. Time is measured in inverse stage-rate units.

Missing transient mass is absorbed mass

The column sums of T are 0 and -1. The exit-rate row is therefore $(0, 1)$, and $f(t) = (0, 1)p(t)$. Adding an absorbing state restores a three-state probability vector with total mass one.

Exact integration gives

$$\int_0^\infty f(t)\, dt = 1, \qquad \mathbb{E}[\tau] = 2, \qquad \mathbb{E}[\tau^2] = 6, \qquad \mathrm{Var}(\tau) = 2.$$

Equivalently, the mean is $\mathbf{1}^{\mathsf{T}}(-T)^{-1}e_0$. A single exponential waiting time with mean two would have constant hazard $1/2$ and variance four, so matching the mean alone would not reproduce this distribution.

The hazard formula is used where $S(t) > 0$. The model has that property for every finite $t \geq 0$. No division by an already vanished survival mass is required.

Read this as: Survival tracks mass, density tracks exits per unit time, and hazard conditions on survival. Equal means or equal initial rates do not identify a waiting-time law.

Plate VI.25 / HOLOMORPHIC CALCULUS / A RECTANGULAR COMPARISON

Naturality also transports derivatives

Chapter VI, Part I: Theorem 15.1. Exact intertwining passes to the specified functional calculus.

Consider a fine operator, a coarse operator, and a rectangular map

$$A_f = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{pmatrix}, \qquad A_c = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}, \qquad C_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Direct multiplication gives $C_r A_f = A_c C_r$. For $F(z) = 1/(1+z)$, which is defined at both spectra, compute the matrices by inversion:

$$F(A_f) = \begin{pmatrix} 1/3 & -1/9 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/10 \end{pmatrix}.$$

A_f

2	1	0
0	2	0
0	0	9

$F(A_f)$ for $F(z) = (1+z)^{-1}$

$\frac{1}{3}$	$-\frac{1}{9}$	0
0	$\frac{1}{3}$	0
0	0	$\frac{1}{10}$

$$F(2) = 1/3$$

$$F'(2) = -1/9$$

The dashed block is $F(A_c)$, not just $F(2)I$.

The discarded eigenvalue 9 contributes $1/10$ only on the fine side. On the retained Jordan block, the off-diagonal entry is $F'(2) = -1/9$, not zero. The rectangular comparison preserves that entry.

Why spectral values are not enough

On the retained block $A_c = 2I + N$, with $N^2 = 0$, one has

$$F(A_c) = F(2)I + F'(2)N = \tfrac{1}{3}I - \tfrac{1}{9}N.$$

Hence $C_r F(A_f) = F(A_c) C_r$. Replacing the block by $F(2)I$ would lose the nonzero derivative term and no longer give the actual inverse of $I + A_c$.

For polynomials, naturality follows by transporting every operator power. The manuscript's finite holomorphic statement uses finite Taylor data at each primary block and the corresponding nilpotent orders. This example verifies a rational instance with an actual inverse rather than asserting that arbitrary scalar substitutions are meaningful.

The same mechanism explains why the Jordan memory kernel on Plate VI.23 contains te^{-2t}. Exponentiation also has nonzero derivatives at repeated spectral values.

Read this as: A functional calculus acts on operators, including their nilpotent parts. Exact intertwining transports that full action, not merely a list of scalar eigenvalue images.

Plate VI.26 / END-TO-END MODEL / TWO OBSERVATIONS

One complete coarse filter

Chapter VI, Part I: Proposition 9.3 with Theorems 5.1 and 8.2. Markov closure and likelihood sufficiency are separate.

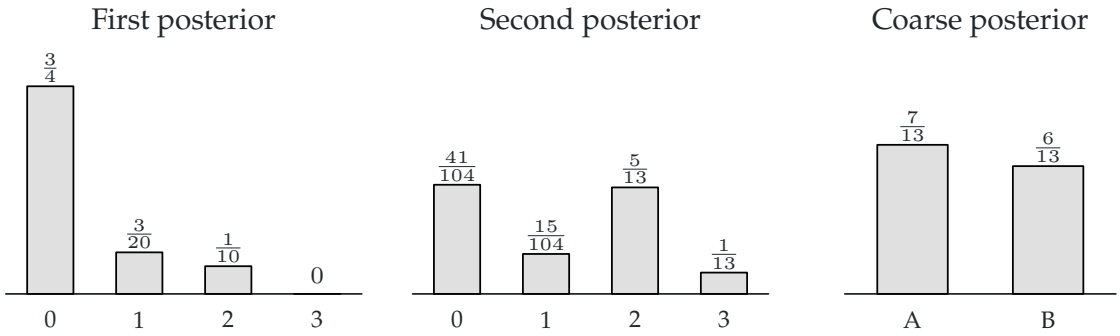
Use the four-state kernel K and aggregation C from Plate VI.5, starting at $p_0 = e_0$. Observe two likelihood vectors constant inside both packets:

$$\lambda^{(1)} = (3/4, 3/4, 1/4, 1/4), \qquad \lambda^{(2)} = (1/2, 1/2, 1, 1).$$

Predict, then update at each observation. The fine calculation gives

$$p_1 = \tfrac{1}{20}(15, 3, 2, 0)^{\mathsf{T}}, \qquad p_2 = \tfrac{1}{104}(41, 15, 40, 8)^{\mathsf{T}}.$$

Evidence $Z_1 = 5/8$, $Z_2 = 13/20$, and $Z_1 Z_2 = 13/32$



Within-packet probabilities change during the two observations, but the packet totals agree exactly with an independent two-state filter. The second coarse posterior is $(7/13, 6/13)$.

Check the normalizers, not only the final vector

The first and second predictive evidences are

$$Z_1 = \frac{5}{8}, \qquad Z_2 = \frac{13}{20}, \qquad Z_1 Z_2 = \frac{13}{32}.$$

The unnormalized product $D_2 K D_1 K p_0$ has mass $13/32$ and normalizes to p_2. Because the likelihood entries here lie in $[0, 1]$, they can serve as probabilities of the specified observations. This makes the product their joint evidence in the stated model.

Two distinct equalities justify the coarse computation. First, $CK = K_c C$ transports prediction. Second, the likelihood is constant on each packet, so $CD_r = \overline{D}_r C$ transports observation weighting. Their composition preserves both the posterior packet totals and the evidence factors.

The coarse filter does not recover the within-packet shape. If a later observation distinguishes states inside a packet, those missing probabilities can become necessary, as the counterexample on Plate VI.10 shows.

> **Read this as:** Exact filtering at a coarse level needs both dynamical lumpability and observation sufficiency. Agreement under one step does not excuse testing the other.

Plate VI.27 / REFINEMENT LIMITS / WHAT THE FINITE MODELS DO NOT PROVE

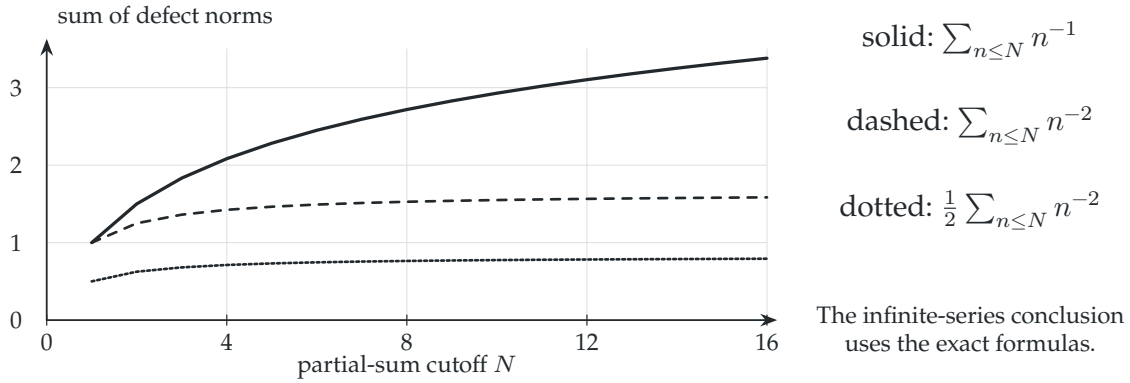
Small at each level is not summable across levels

Chapter VI, Part I: Corollary 10.5 and Section 17. Summability and uniform control are additional data.

For each $n \geq 1$, compare the two scalar operators $A_n = 0$ and $B_n = 1/n^2$ through the identity. Their ordinary defects have sizes $1/n^2$, while their square-root defects have sizes $1/n$:

$$\sum_n |A_n - B_n| < \infty, \qquad \sum_n |A_n^{1/2} - B_n^{1/2}| = \infty.$$

These are independent scalar comparisons, not a constructed inverse system. They isolate an error-scale obstruction without claiming a coherent refinement tower.



Partial sums through $n = 16$ illustrate the contrast. The convergence and divergence statements follow from the displayed series, not from the finite plot. A common positive gap gives the third, summable upper bound.

A uniform hypothesis can change the conclusion

Replace the pair by $A_n = 1$ and $B_n = 1 + 1/n^2$. Then

$$\left|B_n^{1/2} - A_n^{1/2}\right| = \frac{1/n^2}{\sqrt{1 + 1/n^2} + 1} \leq \frac{1}{2n^2}.$$

This is the elementary form of the uniform-gap mechanism on Plate VI.16. Without such information, transferring a summable generator-defect budget through a fractional function can fail.

The formal contract keeps three research questions separate: realization of broader transfer semigroups, uniform stochastic-refinement control, and latent packet closure. The exact finite examples here do not settle their existence, domain, topology, or uniformity requirements.

Likewise, exact hidden-state elimination only yields the finite memory models that were supplied. It does not establish a finite latent realization for every memory law. A complete list of matrix identities can still leave an infinite-dimensional approximation problem untouched.

Read this as: A finite example can expose a missing hypothesis. It cannot supply a projective-limit construction, a uniform constant, or a continuum realization merely by passing at several levels.

Plate VI.28 / READER'S GUIDE / CALCULATION, DRAWING, AND FORMAL CONTRACT

How to use the evidence

Chapter VI formal companion: thirty-three-result correspondence and reproduction record.

Plates VI.1–VI.28 develop the finite worked models corresponding to Part I. The references beneath their titles use Part I's theorem numbers. Plates VI.29–VI.42 address the broader results in Part II.

The formal contract covers 33 scoped results, with 341 supporting theorem declarations in 47 proof modules and a saved audit of 782 namespace declarations. These counts describe the recorded formal development rather than independent new mathematical discoveries. The three conjectures retain their stated status.

Read each kind of evidence at its actual strength

Evidence	What is checked here	What it does not establish
Exact matrices	Stochasticity, intertwining, projector identities, inverse formulas, and specific defects.	The same properties for arbitrary partitions or unbounded operators.
Rational algebra	Posterior/evidence identities, polynomial integrals, nodal errors, and complete finite path sums.	An unstated observation model or a continuum convergence theorem.
Symbolic formulas	The displayed exponentials, Laplace transforms, derivatives, and finite spectral powers under stated conditions.	A general-purpose formal proof of every analytic identity.
Vector drawings	Reproducible geometry and sampled curves from the explicit models.	Numerical quadrature certificates or an exhaustive infinite-time experiment.
Saved Lean records	The original contract, source files, dependency pins, and earlier verification logs.	Fresh kernel verification of this editorial addition.

Reproduce a calculation before adapting it

The accompanying files contain the mathematical models, drawing generator and exact-example verifier, with execution instructions. The recorded symbolic suite passed 204 exact or symbolic assertions and rejected 11 deliberately false controls. Its separate execution record is identified on Plate VI.42 and names every check and its finite scope.

The finite families are explicit: three-step Feynman–Kac paths are fully enumerated, selected rational priors are checked exhaustively within their stated denominator bound, and Green inverses are checked on four named grids. No finite count is used as a substitute for a general theorem.

To change a model, specify the state convention, partition, likelihood meaning, topology, and domain first. Then rerun the calculations and inspect the regenerated drawings. A preserved formula with changed assumptions is not automatically a preserved result.

> **Read this as:** Use the guide to see why a hypothesis matters and to compute a small case. Use the manuscript and its formal map for the scope of the general claim.

Plate VI.29 / DIRECTED TRANSFER / A FINITE DYSON EXAMPLE

Acyclic paths stop after finitely many transfers

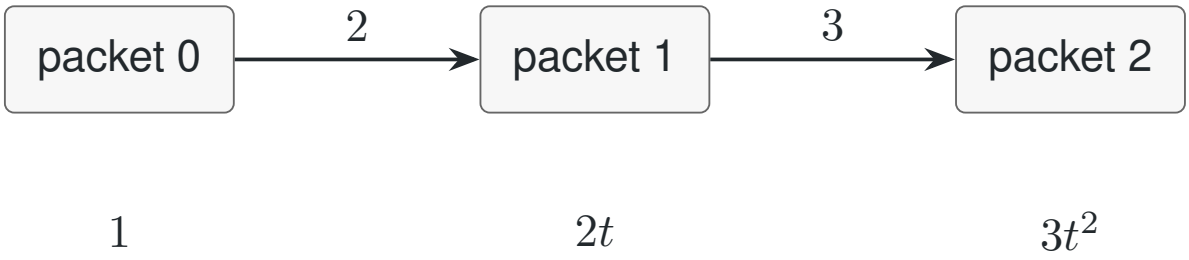
Chapter VI, Part II: Theorem E2.8 and Corollary E2.9. Acyclicity concerns the transfer graph.

Let three scalar packets form the directed chain $0 \to 1 \to 2$, with transfer coefficients 2 and 3. The generator and its powers are

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 6 & 0 & 0 \end{pmatrix}, \qquad A^3 = 0.$$

Therefore the exponential is the exact polynomial

$$e^{tA} = I + tA + \frac{t^2}{2}A^2.$$



At $t = 1$: $(1, 2, 3)$ At $t = 2$: $(1, 4, 12)$

Only the empty path, one-edge paths, and the two-edge path exist. The coefficient $3t^2$ includes both the edge product $2 \cdot 3$ and the ordered-time simplex area $t^2/2$.

Path length counts transfers, not elapsed time

Starting at e_0, the complete state is $(1, 2t, 3t^2)^{\mathsf{T}}$. No fourth packet or repeated cycle can contribute. Direct differentiation gives $x'(t) = Ax(t)$, including the relation $x_2'(t) = 3x_1(t)$.

If nonzero diagonal generators are added, a path acquires an integral of the within-packet evolutions between its transfer times. Acyclicity still bounds the number of transfers. It does not make the diagonal exponential factors polynomial in time.

This is a linear transfer example. The columns of A do not sum to zero. The total of the displayed state is $1 + 2t + 3t^2$. It is consequently not a Markov probability model. Positivity of off-diagonal coefficients alone does not imply conservation of probability.

> **Read this as:** An acyclic packet graph terminates the transfer expansion. Nilpotence makes this particular full exponential polynomial. The two statements have different hypotheses.

Plate VI.30 / REFINEMENT / A PERSISTENT RANGE EXCURSION

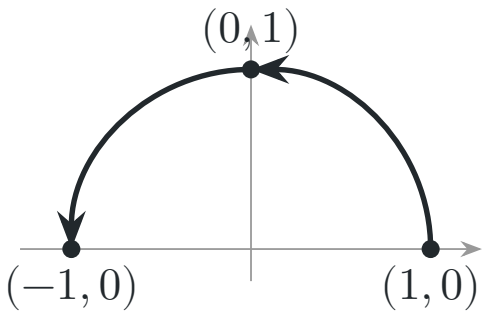
Convergence does not close the dynamics

Chapter VI, Part II: Theorem E4.8 and Example E4.10. Summability and closure are different tests.

Let the coarse space be $\mathbb{R}$ and every finer space be $\mathbb{R}^2$. The first connecting maps are $C(x,y) = x$ and $Rx = (x,0)$. All later maps are identities. Let the coarse evolution be the identity and all fine evolutions be rotations:

$$T(t) = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}, \qquad U_n(t) = CT(t)R = \cos t \quad (n \geq 1).$$

The reconstructed sequence is already constant above its first level. Only one adjacent defect is nonzero, so its defect sum is finite on every bounded time interval.



full two-step return: -1

project after each step: 0

The full state returns to the coarse line with negative sign after two quarter turns. Resetting the hidden coordinate after the first quarter turn loses the returning contribution.

A limit can fail the semigroup identity

At $s = t = \pi/2$,

$$U(s)U(t) = 0, \qquad U(s+t) = -1.$$

Write $P = RC = \operatorname{diag}(1,0)$. The exact algebraic difference is

$$U(s)U(t) - U(s+t) = CT(s)(P - I)T(t)R.$$

Its value is 1 at these times, matching the missing hidden excursion. This is a failure of closure, not a failure of convergence.

The strengthened closure theorem asks for the corresponding excursion norm to tend to zero with refinement, uniformly on bounded time intervals. Generator control of range leakage supplies a sufficient condition in its stated contractive setting. The present rotation example never meets that vanishing condition.

> **Read this as:** A convergent sequence of reconstructed evolutions need not converge to a semigroup. The limit must also forget no excursion that can later return.

Plate VI.31 / SUMMABLE DEFECTS / A SCALAR COMPLETION

A tail bound states what refinement buys

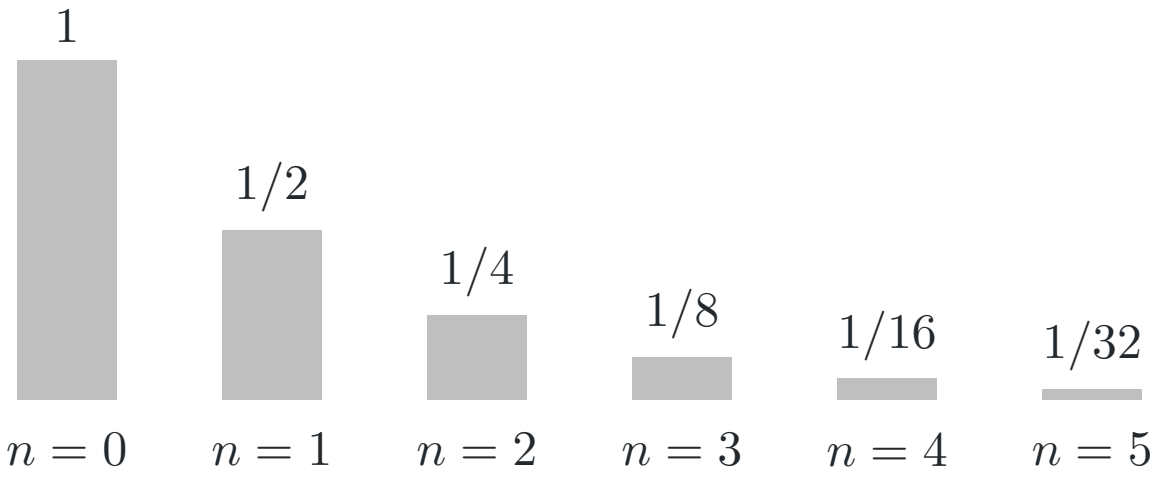
Chapter VI, Part II: Theorem E4.5 and Corollary E4.7. This example has identity connecting maps.

Take $X_n = \mathbb{R}$, $C_n = R_n = 1$, and

$$a_n = 1 - 2^{-n}, \qquad A_n = -a_n, \qquad T_n(t) = e^{-a_n t} \quad (n \geq 0).$$

Each semigroup is contractive for $t \geq 0$. The exact adjacent generator-defect magnitude is

$$|A_{n+1} - A_n| = 2^{-(n+1)}, \qquad \sum_{j=n}^{\infty} |A_{j+1} - A_j| = 2^{-n}.$$



The bars show the certified tail 2^{-n} at horizon $T = 1$. They are bounds on a whole time interval, not errors sampled at the right endpoint.

The estimate follows before any graph is drawn

For $0 \leq t \leq T$, the derivative of $a \mapsto e^{-at}$ has absolute value at most T when $a \geq 0$. Thus

$$\sup_{0 \leq t \leq T} |T_n(t) - e^{-t}| \leq T 2^{-n}.$$

The limit is e^{-t}, which does have the semigroup law. Here every reconstruction is the identity, so the hidden-range excursion is identically zero. That closure fact is additional to the tail estimate.

In a general tower, the weighted estimate includes norms of composed connecting and reconstruction maps. Dropping those weights without contraction or another uniform bound can change a convergent error budget into an invalid one. Even a small defect at each individual level does not control the sum over all levels.

> **Read this as:** A summable tail gives an explicit stopping rule for a fixed horizon. It does not remove weights, supply uniformity on all time, or prove closure without its own hypotheses.

Plate VI.32 / MARKOV DESCENT / TEST THE AGGREGATION KERNEL

The obstruction lives in the forgotten directions

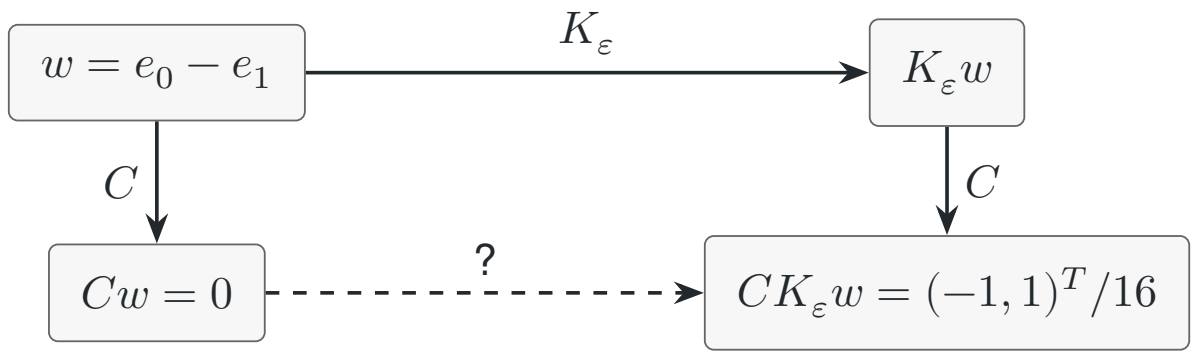
Chapter VI, Part II: Theorem E5.3 and Corollary E5.4. A quotient criterion with a concrete witness.

Return to the four-state model of Plate VI.4 and the perturbed stochastic matrix K_ε of Plate VI.6. Let

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \qquad w = e_0 - e_1.$$

The vector w changes the fine distribution inside the first packet without changing its mass: $Cw = 0$. Yet

$$CK_\varepsilon w = \begin{pmatrix} -1/16 \\ 1/16 \end{pmatrix} \neq 0.$$



No linear coarse map can make this step: 0 must map to 0.

The coarse input cannot see w. The next coarse output can. Therefore no linear coarse operator can reproduce this fine step for every input.

Why one witness rules out every coarse matrix

If some $\bar{K}$ satisfied $CK_\varepsilon = \bar{K}C$, then $CK_\varepsilon w = \bar{K}Cw = 0$, contradicting the displayed vector. The failure is not repaired by fitting a different coarse transition matrix to selected priors.

More generally, for a surjective aggregation map C, a linear map M descends exactly if CM vanishes on $\ker C$. The descended map is defined by $\bar{M}(Cx) = CMx$. The kernel condition makes this independent of the representative x.

For a deterministic partition of a finite stochastic system, differences $e_i - e_j$ within the same source block span the forgotten directions. Testing them gives the familiar equality of aggregate columns. Nonnegativity and column sums of the descended stochastic matrix then follow from the fine stochastic matrix and the partition structure.

> **Read this as:** The canonical obstruction is the action on information that aggregation discards. A nonzero witness in that kernel rules out exact descent at once.

Plate VI.33 / ORNSTEIN–UHLENBECK / HERMITE ENERGY

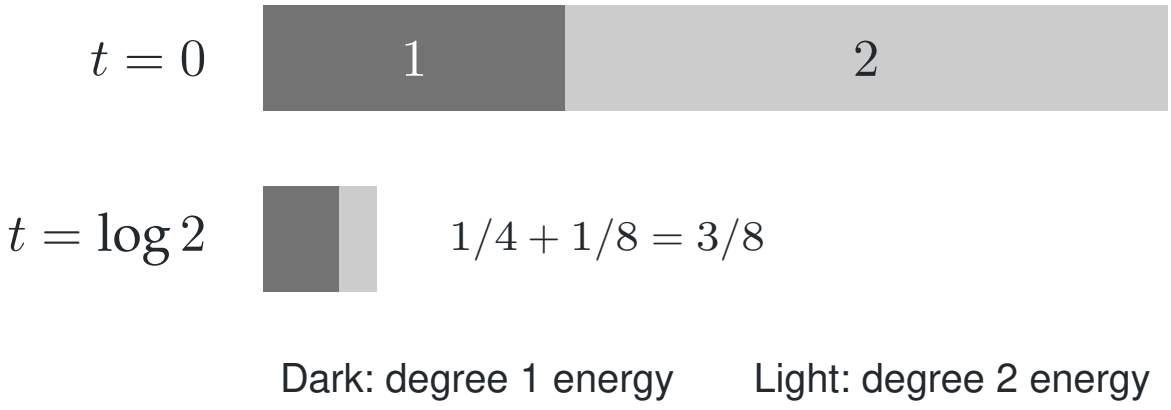
Gaussian packets have separate clocks

Chapter VI, Part II: Theorem E7.1 and Corollaries E7.2, E7.4, and E7.5. Standard Gaussian measure.

In $L^2(\gamma)$ for the standard Gaussian law on $\mathbb{R}$, use the probabilists' Hermite polynomials

$$H_1(x) = x, \qquad H_2(x) = x^2 - 1, \qquad H_4(x) = x^4 - 6x^2 + 3.$$

The Ornstein–Uhlenbeck generator is $\mathcal{L}f = f'' - xf'$. Direct polynomial differentiation gives $\mathcal{L}H_n = -nH_n$ for these three polynomials. Their squared norms are $n!$ and different Hermite degrees are orthogonal.



For $f = H_1 + H_2$, the two bars are its separate squared-energy contributions. At $t = \log 2$, their sum is $3/8$, below the spectral-gap bound $3/4$.

Reconstruction preserves the modes as time changes their weights

The semigroup sends $f = H_1 + H_2$ to

$$T(t)f = e^{-t}H_1 + e^{-2t}H_2, \qquad \|T(t)f\|_2^2 = e^{-2t} + 2e^{-4t}.$$

Its mean is zero. The Gaussian gap bounds this by $e^{-2t}\|f\|_2^2 = 3e^{-2t}$. It need not be an equality when multiple degrees are present.

For the nonnegative operator $-\mathcal{L}$, a square-root evolution replaces rate n by $\sqrt{n}$. On $g = H_1 + H_4$, the coefficients at $t = \log 2$ are $1/2$ and $1/4$, whereas ordinary OU evolution gives $1/2$ and $1/16$. The eigenfunctions agree. The rates differ.

These calculations concern polynomial elements of Gaussian L^2. The general expansion and generator-domain statements belong to the traditional proof. A finite list of Hermite checks does not establish completeness of the basis.

> **Read this as:** Orthogonal packets make the energy accounting exact. A fractional generator alters the spectral clock while keeping the same Gaussian modes.

Plate VI.34 / FIXED LIKELIHOODS / JOINT PROFILES

Several observations determine one sufficient partition

Chapter VI, Part II: Theorem E9.3. A finite family of strictly positive likelihoods.

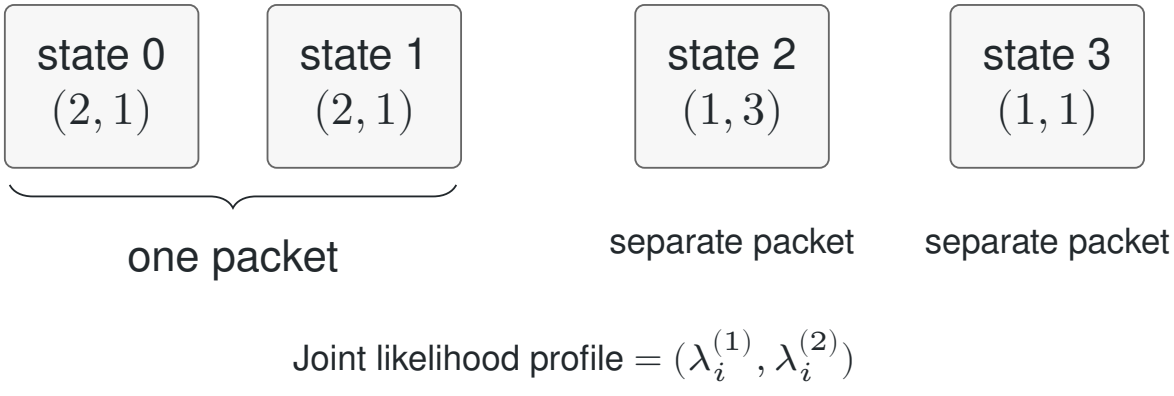
On four states, consider two likelihoods

$$\lambda^{(1)} = (2, 2, 1, 1), \qquad \lambda^{(2)} = (1, 1, 3, 1).$$

Two states may share a packet in the coarsest partition on which every listed likelihood is constant exactly when their entire likelihood profiles agree. The profiles are

$$(2, 1),\ (2, 1),\ (1, 3),\ (1, 1),$$

so the partition is $\{0, 1\}, \{2\}, \{3\}$.



The first observation alone allows the last two states to be merged. The second observation distinguishes them. The shared partition must respect both likelihoods.

A complete two-stage calculation

Start from the uniform prior. The coarse prior is $(1/2, 1/4, 1/4)$. After the first likelihood, it becomes $(2/3, 1/6, 1/6)$. After the second, it is

$$(1/2, 3/8, 1/8).$$

The direct fine update multiplies the likelihoods componentwise, giving weights $(2, 2, 3, 1)$ and posterior $(2, 2, 3, 1)/8$. Its packet masses are precisely the displayed coarse posterior.

The individual evidences are $Z_1 = 3/2$ and $Z_2 = 4/3$, and their product is 2, the direct evidence of the product likelihood. These are relative likelihood weights, so evidence larger than one is allowed.

This partition supplies fixed coarse likelihoods that work for every prior. A single-packet aggregation always gives posterior (1). That mass-only commutation does not identify likelihood profiles. A prior-dependent conditional average can summarize one update on a coarser partition, but requires the within-packet information that a fixed sufficient rule avoids.

> **Read this as:** Equal full likelihood profiles determine the coarsest partition on which every listed likelihood is constant. This partition admits fixed sufficient update rules.

Plate VI.35 / RELATIVE ENTROPY / COARSE AND CONDITIONAL TERMS

Information loss has an exact location

Chapter VI, Part II: Theorem E9.8 and Corollary E9.9. Positive reference masses in this example.

Take

$$p = (1/2, 1/4, 1/8, 1/8), \qquad q = (1/4, 1/4, 1/4, 1/4)$$

and group the first two and last two states. Their coarse laws are $P = (3/4, 1/4)$ and $Q = (1/2, 1/2)$. Define $D(p\|q) = \sum_i p_i \log(p_i/q_i)$ with natural logarithms.

$$D(p\|q) = \tfrac{1}{4}\log 2$$



coarse information
$\frac{3}{4}\log 3 - \log 2$

conditional information
$\frac{5}{4}\log 2 - \frac{3}{4}\log 3$

The bar lengths use the displayed exact formulas evaluated for illustration. The split is an identity: coarse information plus posterior-weighted conditional information. It is not a fitted allocation.

The chain rule identifies exactly what aggregation removes

Inside packet A, the conditional laws are $(2/3, 1/3)$ and $(1/2, 1/2)$. Inside B they agree. Hence

$$\begin{aligned} D(p\|q) &= \tfrac{1}{4}\log 2, \\ D(P\|Q) &= \tfrac{3}{4}\log 3 - \log 2, \\ \sum_a P_a D(p(\cdot \mid a)\|q(\cdot \mid a)) &= \tfrac{5}{4}\log 2 - \tfrac{3}{4}\log 3. \end{aligned}$$

Adding the last two lines recovers the first. Every conditional term is nonnegative, so aggregation cannot increase relative entropy. Equality holds precisely when the occupied packets have the same conditional laws, under the stated support convention.

The weighting is by the packet mass P_a from the first argument of $D(p\|q)$. Replacing it by Q_a generally breaks the identity. Likewise, forgetting zero-mass conventions or allowing a positive p_i over a zero q_i changes which expressions are finite.

> **Read this as:** Coarse relative entropy measures distinguishability that survives aggregation. The conditional term records exactly the distinguishability removed inside the packets.

Plate VI.36 / BAYESIAN FLOW / EVIDENCE CUMULANTS

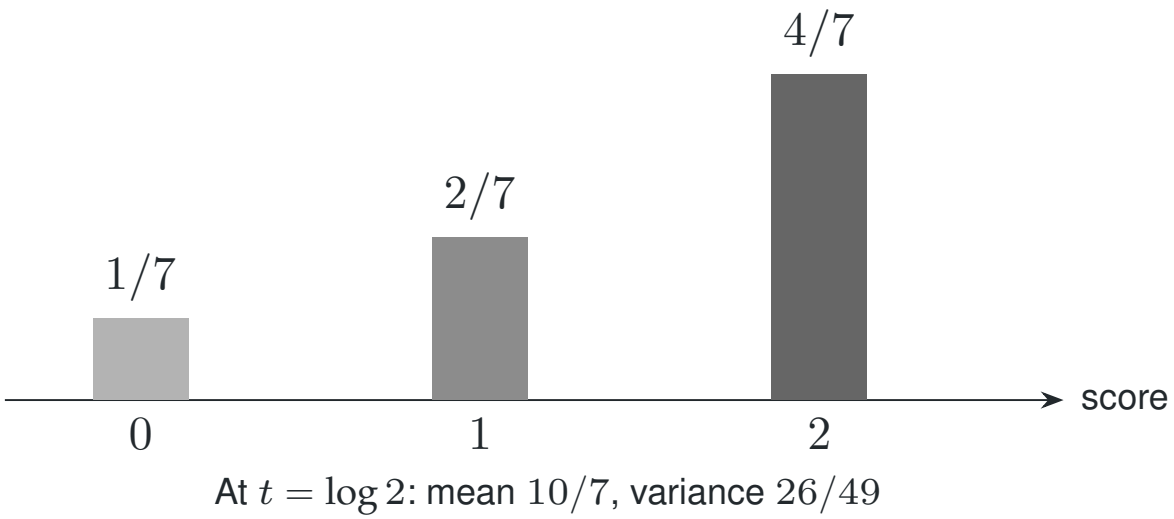
The score rises because it has variance

Chapter VI, Part II: Theorems E10.1 and E10.2. Exponential tilting of a fixed finite prior.

Start with equal prior masses on scores $u = (0, 1, 2)$ and likelihood e^{tu}. Put $q = e^t$. The evidence and tilted distribution are

$$Z(t) = \frac{1 + q + q^2}{3}, \qquad p(t) = \frac{(1, q, q^2)}{1 + q + q^2}.$$

At $t = \log 2$, they are $Z = 7/3$ and $p = (1, 2, 4)/7$.



The score values stay fixed. Their probabilities change. The rising mean score is a property of this tilted family, not an increase in the individual scores.

Two derivatives have different statistical meanings

The first derivative of log evidence is the current mean score, and the second is its variance:

$$\frac{d}{dt} \log Z(t) = \mathbb{E}_{p(t)} u, \qquad \frac{d^2}{dt^2} \log Z(t) = \mathrm{Var}_{p(t)}(u) \geq 0.$$

At $q = 2$ the exact values are

$$\mathbb{E}u = \frac{10}{7}, \qquad \mathbb{E}u^2 = \frac{18}{7}, \qquad \mathrm{Var}(u) = \frac{26}{49}.$$

Thus the derivative of the mean is $26/49$ at this time. The flow equation $p_i' = (u_i - \mathbb{E}u)p_i$ gives derivatives $(-10, -6, 16)/49$, which sum to zero.

As $t \to \infty$, mass concentrates on the largest score present in the initial support. A score with zero initial prior mass is never created by multiplication. If several occupied states tie for the maximum, their relative prior proportions remain unchanged within that maximal packet.

Read this as: Log evidence is a cumulant generator for the tilted law. Its curvature is variance, so the mean score is monotone even though the flow is nonlinear.

Plate VI.37 / MEMORY COMPRESSION / AN OPTIMIZATION WITH A DEFINED COST

Keep blocks by their certified contribution

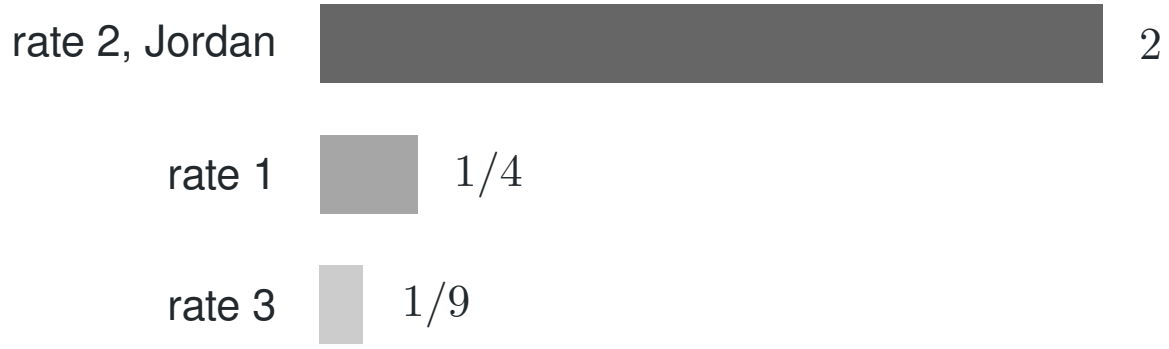
Chapter VI, Part II: Theorem E15.1 and Corollary E15.2. Stable primary blocks. Block count is the budget.

Consider the scalar memory kernel

$$\mathcal{K}(t) = \tfrac{1}{4}e^{-t} + (3+2t)e^{-2t} + \tfrac{1}{3}e^{-3t}.$$

It has three primary blocks. The middle block carries a Jordan term and must be retained or omitted as a whole for this selection rule. Their certified integrated weights are

$$w_1 = \frac{1}{4}, \qquad w_2 = \frac{3}{2} + \frac{2}{2^2} = 2, \qquad w_3 = \frac{1/3}{3} = \frac{1}{9}.$$



With one primary block retained, choose the middle block and certify omitted weight $13/36$. With two retained, add the rate-one block and the remaining certificate is $1/9$.

The slowest eigenvalue is not necessarily the largest effect

The rate-two block has the greatest weight because its couplings and nilpotent contribution dominate. Retaining the slowest rate alone would omit weight $2 + 1/9$, a much larger certificate.

The descending order is $2, 1/4, 1/9$. A target tolerance $1/8$ needs two blocks according to this bound: one block leaves $13/36 > 1/8$, while two leave $1/9 \le 1/8$. Every term in this particular scalar kernel is nonnegative, so its omitted integral equals the certificate.

In general matrix kernels, cancellations can make the true error smaller. The theorem optimizes the sum of omitted block weights, not every possible approximation error. Moreover, a Jordan block may cost several state dimensions. A fixed budget in total hidden dimension is a different optimization problem from a fixed number of primary blocks.

> **Read this as:** The largest-weight rule is exact for its stated certificate and block-count budget. It is not a claim of globally optimal model reduction.

Plate VI.38 / VOLTERRA STABILITY / SAME INITIAL DATA AND FORCING

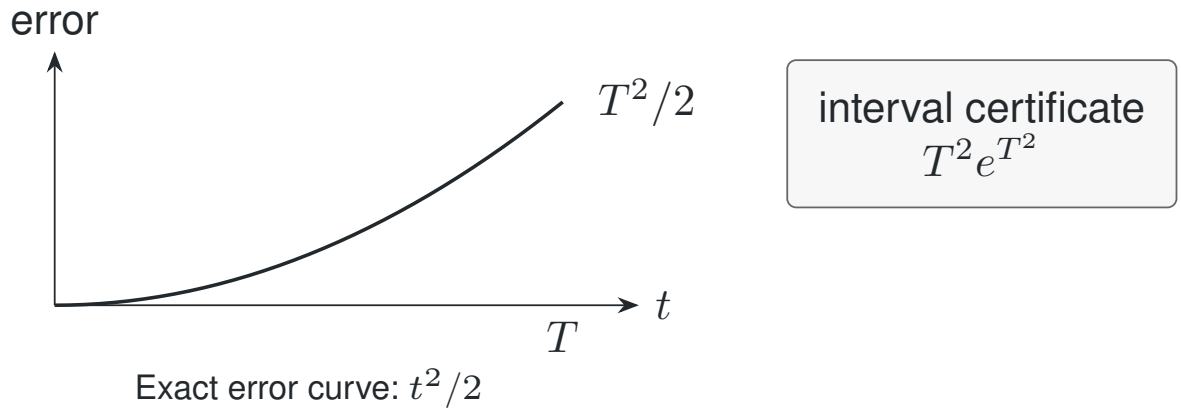
A kernel certificate becomes a trajectory certificate

Chapter VI, Part II: Theorem E15.6. Norms and a finite time horizon are explicit.

Let $N = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, so $N^2 = 0$ and $\|N\|_2 = 1$. On $[0, T]$, compare the memory kernels $K(t) = N$ and $\widetilde{K}(t) = 0$, with identity free semigroup, no forcing, and initial state $x_0 = e_1 = (0, 1)^\mathsf{T}$. The equations are

$$x'(t) = \int_0^t N x(r)\, dr, \qquad \tilde{x}'(t) = 0.$$

Their exact solutions are $x(t) = (t^2/2, 1)^\mathsf{T}$ and $\tilde{x}(t) = (0, 1)^\mathsf{T}$.



The first-coordinate error is exactly $t^2/2$. The certificate bounds its maximum on the entire interval. The bound is deliberately shown separately from the trajectory.

Track every constant before applying the estimate

Here $M_T = 1$, $B = \sup \|\tilde{x}\| = 1$, and

$$L = \|K\|_{L^1(0,T)} = T, \qquad \varepsilon = \|K - \widetilde{K}\|_{L^1(0,T)} = T.$$

The theorem yields

$$\sup_{0 \le t \le T} \|x(t) - \tilde{x}(t)\| = \frac{T^2}{2} \le T^2 e^{T^2}.$$

At $T = 1/2$, the exact error is $1/8$ and the certificate is $e^{1/4}/4$. Nilpotence makes this example polynomial. It does not remove the time horizon from the general theorem.

The two solutions share initial state, free semigroup, and forcing. If those are changed as well, their errors need additional terms. An integrated kernel error alone does not automatically control a trajectory on an infinite horizon.

> **Read this as:** The compression estimate transports an L^1 kernel budget into a finite-horizon state bound, using an explicit bound on the comparison trajectory.

Plate VI.39 / FINITE REALIZATION / AN EXACT DIFFERENTIAL MODEL

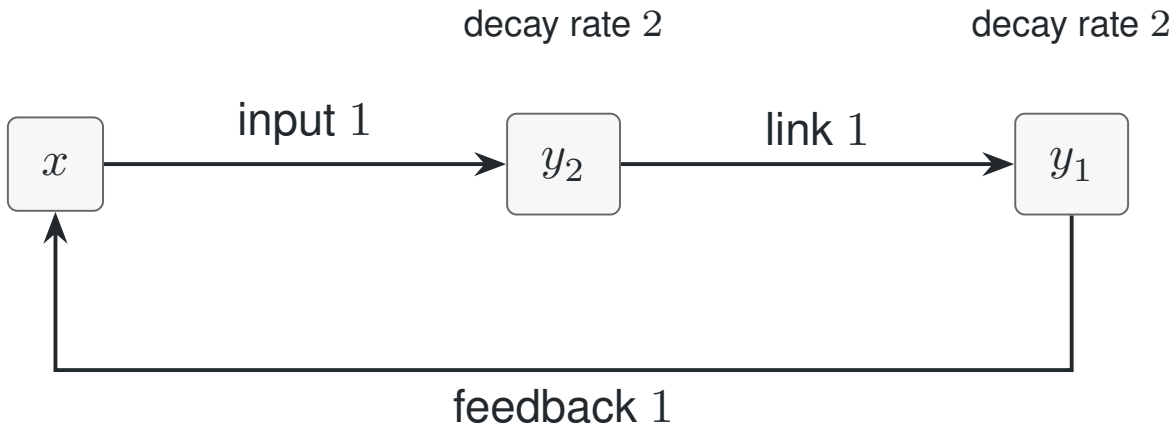
Two auxiliary states realize a repeated pole

Chapter VI, Part II: Corollaries E15.4 and E15.5. This is a supplied rational realization.

The kernel $K(t) = te^{-2t}$ from Plate VI.23 has transform $1/(z+2)^2$. It is realized by two auxiliary states:

$$\begin{aligned} \dot{y}_1 &= -2y_1 + y_2, \quad y_1(0) = 0, \\ \dot{y}_2 &= -2y_2 + x, \quad y_2(0) = 0, \\ \dot{x} &= ax + y_1. \end{aligned}$$

With $D = \begin{pmatrix} -2 & 1 \\ 0 & -2 \end{pmatrix}$, $B = (1, 0)$, and $C = (0, 1)^{\mathsf{T}}$, the feedback kernel is $Be^{tD}C = te^{-2t}$.



An input first enters y_2, transfers to y_1, and then returns to x. The two equal decay rates produce a repeated pole through the intervening link.

Elimination recovers the convolution, including its preparation

Variation of constants gives

$$y(t) = e^{tD}y(0) + \int_0^t e^{(t-r)D}Cx(r)\,dr,$$

so the retained equation is

$$\dot{x}(t) = ax(t) + Be^{tD}y(0) + \int_0^t (t-r)e^{-2(t-r)}x(r)\,dr.$$

The convolution-only version is exact for $y(0) = 0$. For a nonzero preparation, $Be^{tD}y(0) = e^{-2t}(y_1(0) + ty_2(0))$ is an additional forcing.

The highest pole coefficient is nonzero in this example. A hidden Jordan block can become invisible to a transfer function if its entrance or exit coupling annihilates that coefficient. Pole order is therefore bounded by hidden Jordan depth, with equality requiring a visible highest term.

> **Read this as:** A polynomial-exponential kernel has a concrete finite state realization when the matrices are supplied. This does not realize every possible memory law with finitely many states.

Plate VI.40 / COMPETING RISKS / RESIDUAL LIFE

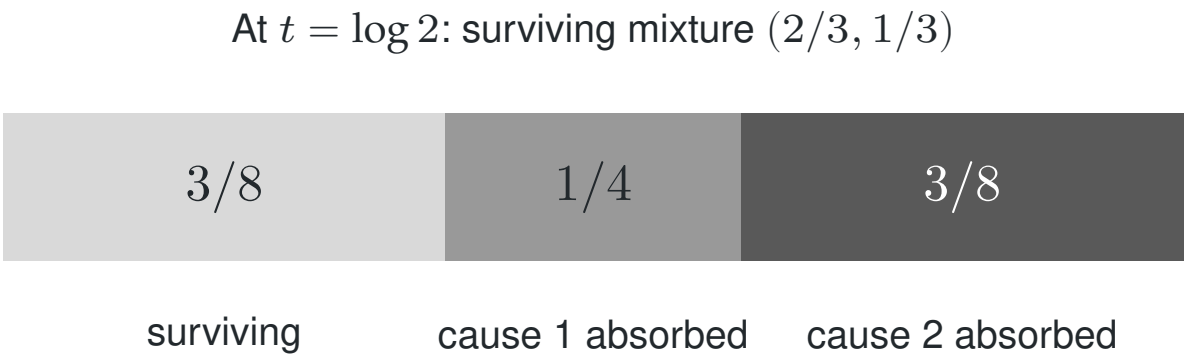
Survivors have a different mixture of causes

Chapter VI, Part II: Proposition E16.4 and Theorem E16.5. This example is stable and absorbs surely.

Two transient states have no internal transitions. Their exit rates are 1 and 2, their initial masses are both $1/2$, and the two states exit by distinct causes. Thus

$$A = \operatorname{diag}(-1, -2), \qquad S(t) = \tfrac{1}{2}e^{-t} + \tfrac{1}{2}e^{-2t},$$

with cause densities $f_1(t) = e^{-t}/2$ and $f_2(t) = e^{-2t}$.



At $t = \log 2$, surviving mass is $3/8$. Cause 1 has accumulated probability $1/4$ and cause 2 has accumulated probability $3/8$. The three pieces sum to one.

Condition on survival before restarting the clock

At this time the transient masses are $(1/4, 1/8)$, so the conditional distribution of survivors is $(2/3, 1/3)$. Consequently

$$\frac{S(t+s)}{S(t)} = \frac{2}{3}e^{-s} + \frac{1}{3}e^{-2s}, \qquad \mathbb{E}[\tau - t \mid \tau > t] = \frac{2}{3} + \frac{1}{6} = \frac{5}{6}.$$

The original mean lifetime was $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = 3/4$. Surviving longer changes the mixture toward the slower state. It does not restart the original mixture.

The cause densities at $t = \log 2$ both equal $1/4$, so the cause-specific hazards are both $2/3$ and total hazard is $4/3$. Hazards are rates and need not lie in $[0, 1]$. Each eventual cause probability is $1/2$, obtained by integrating its density. Stability of the subgenerator is part of that eventual-absorption statement.

> **Read this as:** Residual life restarts the same transient semigroup from the conditional surviving distribution. Cause probabilities, densities, and hazards answer different questions.

Plate VI.41 / FUNCTIONAL CALCULUS / AN EXACT RATIONAL TEST

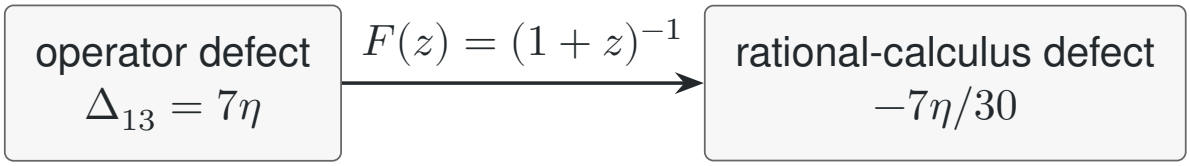
An operator substitution transports the defect too

Chapter VI, Part II: Theorem E3.4; compare Chapter VI, Part I, Theorem 15.1, and Plate VI.25.

Use the operators from Plate VI.25, but perturb the rectangular comparison map:

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}, \quad C_\eta = \begin{pmatrix} 1 & 0 & \eta \\ 0 & 1 & 0 \end{pmatrix}.$$

The generator defect is $\Delta = C_\eta A - BC_\eta$, with only one nonzero entry, $\Delta_{13} = 7\eta$ (matrix indices begin at one).



Left inverse factor $1/3$ Right inverse factor $1/10$ Sign $-$

At $\eta = 0$: both defects vanish exactly

The perturbation lets the discarded eigenvalue enter the comparison. Exact naturality holds at $\eta = 0$. A nonzero defect is transformed, not erased, by the rational calculus.

A resolvent identity supplies the sign and both inverse factors

For $F(z) = 1/(1+z)$, all required inverses exist, and

$$C_\eta F(A) - F(B)C_\eta = -(I+B)^{-1}\Delta(I+A)^{-1} = \begin{pmatrix} 0 & 0 & -7\eta/30 \\ 0 & 0 & 0 \end{pmatrix}.$$

This follows by multiplying the difference by $I+B$ on the left and $I+A$ on the right. The middle numerator is $(I+B)C_\eta - C_\eta(I+A) = -\Delta$.

The retained Jordan block still contributes $F'(2) = -1/9$, as on Plate VI.25. The new third-column discrepancy compares $F(9) = 1/10$ with $F(2) = 1/3$, multiplied by η.

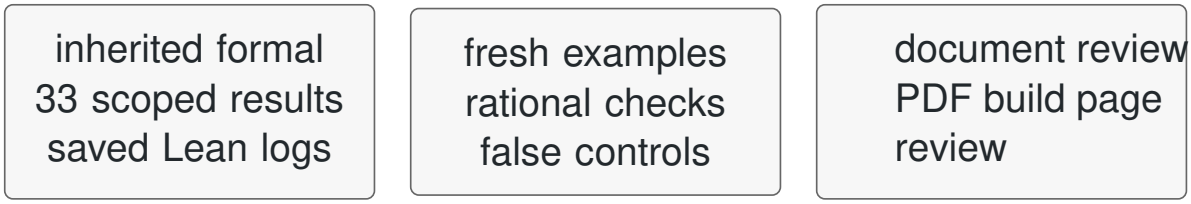
The general holomorphic defect formula integrates the corresponding resolvent defect under its contour and spectral assumptions. This rational example verifies an actual matrix inverse identity. It does not remove those analytic hypotheses or establish a calculus for arbitrary unbounded operators.

> **Read this as:** Exact intertwining gives exact naturality. When intertwining fails, the functional calculus has a structured defect whose factors can be computed.

Plate VI.42 / EVIDENCE MAP / FORMAL RECORDS AND FINITE EXAMPLES

How to read the evidence

Chapter VI formal companion: source, execution, and example records.



The three evidence tracks have distinct scopes. The formal contract, exact examples, and PDF assembly have separate verification records.

Read the records without adding incompatible counts

The recorded symbolic suite reports 204 exact or symbolic checks and 11 rejected false controls for its 28-page atlas. The scripts and saved report remain available. The recorded monograph review reran the original symbolic suite with SymPy and mpmath. Its recorded inputs, outputs, resource limits, and execution logs accompany the current result.

The recorded monograph review also reran the separate standard-library exact checker for Plates VI.29–VI.41. It uses rational matrices, finite polynomial algebra, Gaussian moments, and rational coefficients of logarithms. It checks the stated finite examples and rejects deliberately false alternatives. Its report records 120 assertions and 8 rejected false controls. A separate recorded verification reran 182 rational assertions and 4 rejected false controls for selected inherited models. That narrower check overlaps the full symbolic replay and remains separate from its count.

The new checks do not numerically certify improper integrals, all time horizons, or every analytic theorem. For example, the integrated memory weights use the stated exponential-integral formula. The checker verifies its coefficients and finite selection problem. Gaussian polynomial checks do not prove Hermite completeness. Rational entropy coefficient identities do not replace the general support-sensitive chain rule.

The formal and editorial boundaries stay separate

The inherited Lean record covers 33 scoped results with 341 supporting theorem declarations in 47 proof modules. This guide neither enlarges that denominator nor claims a fresh kernel build. Traditional Part II retains broader written proofs. Part III retains earlier complementary results and the operator-substitution material migrated from Chapter IV. The formal companion maps their relationship to the formal baseline.

All 40 plates are editable vector sources: the 26 original plates plus 14 editorial plates. The portable build compiles the shipped geometry without needing symbolic regeneration. The supplied models and Python generators were used to regenerate all forty plates. Geometry samples remain drawing coordinates rather than additional analytic proofs.

> Read this as: Read every proof, verification record and finite computation at its stated scope.

CHAPTER VII

Quantum Seeds and Effective Dynamics

Symmetry, geometric transport, quantum channels, and bounded refinement towers

The transfer viewpoint is now developed in quantum settings with their own algebraic and dynamical assumptions.

Quantum seeds generate spaces through an operator grammar. Symmetry, circuits, geometric transport and quantum channels lead to finite effective dynamics and bounded refinement questions.

IN THIS CHAPTER

Mathematical text 660

Visual studies 772

Formal companion 2054

Sources and references 2681

Scope

The finite formal contract does not certify the full continuum physics program. Anomaly-free refinement, uniform Feshbach control, Lorentzian reconstruction and the proposed gravitational limits remain open.

The next chapter turns to algorithms, separating exact structural reductions from numerical and hardware questions.

How to use this volume

Part I. The finite quantum seed calculus develops the packet language and states the precise scope of the archived formal proof.
Part II. The strengthened development extends this language through seventy-nine theorem, proposition, lemma, and corollary proofs.
Part III. Six supplements give the detailed mathematical constructions behind the formal proofs, followed by three alternative derivations.
Part IV. Forty worked visual pages connect the formulas to finite models and identify the hypotheses needed at each continuum boundary.

Reading this chapter

A seed is useful only together with the operators allowed to act on it. Chapter VII follows that principle through symmetry packets, quantum circuits, moving projectors, channels, effective dynamics, and refinement. The finite calculus in Part I supplies the common language. The strengthened development in Part II gives broader bounded constructions and limit criteria. Part III supplies the detailed proof completions.

The finite results use the prefix I. The bounded development uses II, and the supplementary constructions use III. This local numbering distinguishes their hypotheses and provides stable references throughout the chapter. Some constructions appear in more than one setting because the ambient spaces and conclusions differ.

The formal companion covers nineteen scoped manuscript items through 323 supporting Lean theorem declarations in twenty-five modules. The bounded development in Part II is outside that contract for nineteen items. The companion states the exact hypotheses, source declarations and verification provenance.

Five explicitly named conjectures remain open in the finite core: anomaly-free refinement, uniform Feshbach control, Lorentzian reconstruction, the Einstein infrared limit, and massless spin-two packets. Bounded direct limits and norm limits provide useful criteria. Each physical target still needs the additional structure stated in its conjecture.

Four distinctions to keep in view

A seed's algebraic orbit is different from a unitary state-preparation orbit. Compression is different from an effective resolvent. A normalized channel is different from an unnormalized error family. A finite spectrum is different from a continuum particle interpretation. Keeping these distinctions visible makes the later results easier to use.

Read Part I first for the packet grammar and the finite statements. Turn to Part II for bounded operator transport, spectral and channel constructions, and refinement towers. Use Part III when the geometric bundle construction or a formal specialization needs more detail. The visual atlas follows the same topics through explicit examples. It illustrates the statements with finite data and does not replace their proofs.

Part I Finite quantum seed calculus

A route through this section

The finite core starts with seed rank, then resolves symmetry and circuit transfer into packets. Its geometric statements use the actual range bundle, including the Lie-bracket term. Its channel and Feshbach statements keep normalization and invertibility visible. The three examples below introduce those distinctions with exact finite data.

Three exact finite models

A seed remembers multiplicity support

For $M_2(\mathbb{C}) \otimes I_3$, a coefficient matrix with rank one generates an orbit of dimension two. A matrix of rank two generates an orbit of dimension four. Neither single seed spans the full space of dimension six, because a 2×3 matrix cannot have rank three. Two seeds can nevertheless cover all three multiplicity coordinates together. The example illustrates the finite seed rank formula with $d = 2$ and $m = 3$.

Seed rank measures the available multiplicity support

```
C1 = [[1, 0, 0], [0, 0, 0]]

C2 = [[1, 0, 0], [0, 1, 0]]

C3 = [[0, 0, 1], [0, 0, 0]]

C2 and C3 cover all 3 columns.
```

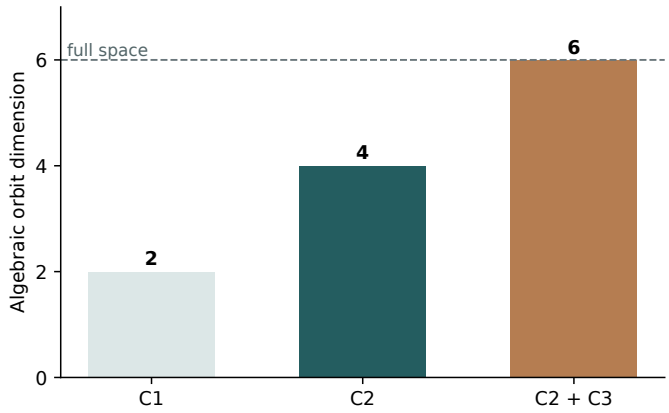


Figure 1. The figure displays exact coefficient matrices for $d = 2$, $m = 3$. The combined row support of two seeds has rank three. The bar heights record algebraic orbit dimensions and should not be read as probabilities of unitary state preparation.

An eliminated coordinate still contributes

Take $H = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 3 \end{smallmatrix}\right)$ and retain its first coordinate. The eliminated block is invertible and the Schur complement of H is $2 - 1/3 = 5/3$. Its inverse is $3/5$, exactly the retained entry of H^{-1}. Inverting the compressed entry alone gives $1/2$. The difference records the contribution of a path through the eliminated coordinate. Direct compression omits that contribution, while block inversion retains it.

Block inversion includes the discarded coordinate

```
H = [[2, 1], [1, 3]]

Schur complement of H:

2 - 1/3 = 5/3

Retained entry of H inverse:

(5/3) inverse = 3/5
```

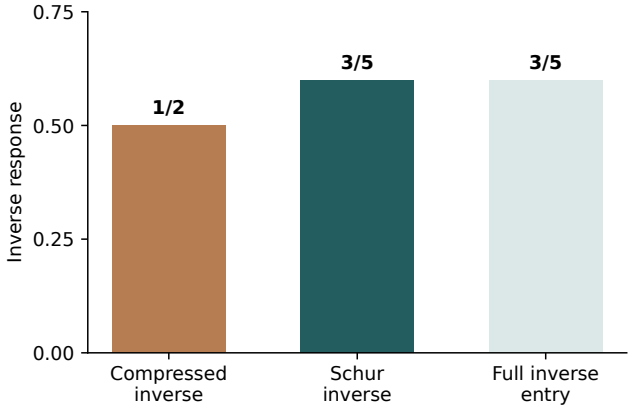


Figure 2. This rational model uses two coordinates to illustrate block inversion. Both the eliminated block and the full operator are invertible, and the plotted values are exact fractions. The example shows why direct compression and resolvent elimination give different results.

Normalization changes the map

Let $P = |0\rangle\langle 0|$ on $\mathbb{C}^2$ and $Z = \mathrm{diag}(1, -1)$. The normalized Kraus family $\sqrt{3/4}\, I$, $\sqrt{1/4}\, Z$ satisfies $\sum_j K_j^* K_j = I$ and maps P to P. The unnormalized family $\sqrt{3}\, I$, Z satisfies $\sum_j E_j^* E_j = 4I$ and maps P to $4P$. The two families have the same relative weights, but their total normalization differs. Both outputs are proportional to the same projector, but their traces differ.

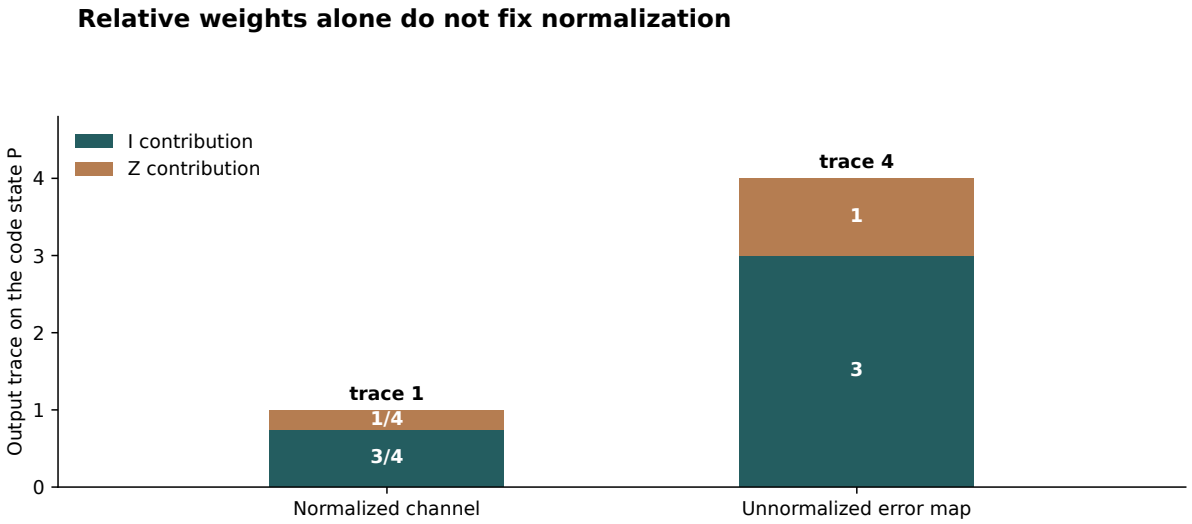


Figure 3. The figure shows exact weights for one supported code state. The normalized map preserves trace one, while the unnormalized positive map has output trace four. The illustration records this finite scaling distinction and does not assert a general recovery construction.

I.1 The quantum seed realization

Conventions. Unless the zero space is explicitly included as a boundary case, quantum state spaces are nonzero complex Hilbert spaces of finite dimension. A complete packet family is a finite collection satisfying

$$P_\alpha^* = P_\alpha, \qquad P_\alpha P_\beta = \delta_{\alpha\beta} P_\alpha, \qquad \sum_\alpha P_\alpha = I.$$

All operator norms in the analytic estimates are induced by the norm on the Hilbert space. The estimates do not use the entrywise maximum norm or the Frobenius norm. We write $[A, B] = AB - BA$ and use units with $\hbar = 1$. Parameter derivatives are real derivatives. A matrix X is supported on a code projector P when $PXP = X$.

The common operator sequence of the series becomes

quantum seed $\longrightarrow$ gate or observable address $\longrightarrow$ symmetry projection $\longrightarrow$ packet transfer $\longrightarrow$ constraint $\longrightarrow$ refinement $\longrightarrow$ reconstruction

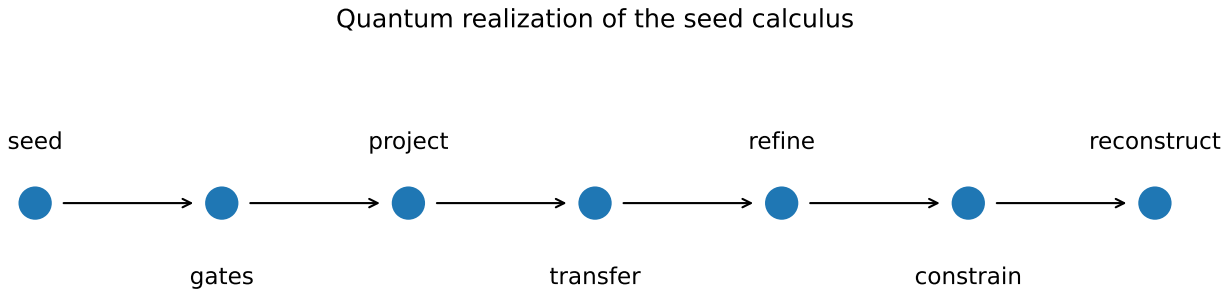


Figure 4. The diagram shows the quantum realization of the common sequence of operators in the seed calculus. The same finite packet language describes states, gates, Hamiltonians, channels, geometric projectors, and effective dynamics.

Definition I.1.1 (Finite quantum seed system)**.** A finite quantum seed system is a tuple

$$\mathfrak{Q} = \left(\mathcal{H}, \psi_0, \mathcal{O}, \mathcal{R}, \{P_{\ell,\alpha}\}, H, \tau\right),$$

where $\mathcal{H}$ is a finite-dimensional Hilbert space, $\psi_0 \in \mathcal{H}$ is a nonzero seed, $\mathcal{O}$ is the declared operator grammar, $\mathcal{R}$ is a collection of algebraic or physical constraints, $\{P_{\ell,\alpha}\}$ is an exact packet system at each refinement level, $H = H^*$ is a Hamiltonian when dynamics are present, and τ specifies the topology used in a later completion.

The generated state space is

$$\mathcal{H}(\psi_0; \mathcal{O}) = \operatorname{span}\left\{O_r \cdots O_1 \psi_0 : O_j \in \mathcal{O},\ r \geq 0\right\}.$$

The quantum seed is cyclic for the grammar when

$$\mathcal{H}(\psi_0;\mathcal{O}) = \mathcal{H}.$$

This is a statement about the pair $(\psi_0, \mathcal{O})$, not about the vector alone. A state may be cyclic for the full matrix algebra and noncyclic for a restricted physical gate set.

I.2 Quantum seed rank and cyclicity

I.2.1 Finite observable algebras

Let a finite-dimensional unital $*$-subalgebra of operators act in its standard multiplicity form

$$\mathcal{A} \cong \bigoplus_{i=1}^{r} \left(M_{d_i}(\mathbb{C}) \otimes I_{m_i} \right)$$

on

$$\mathcal{H} \cong \bigoplus_{i=1}^{r} \left(\mathbb{C}^{d_i} \otimes \mathbb{C}^{m_i} \right).$$

Here every d_i is a positive integer and every multiplicity m_i is a nonnegative integer. An empty block family represents the zero space and has seed rank zero. The maximum over an empty block family is also assigned the value zero. A vector $\psi_i \in \mathbb{C}^{d_i} \otimes \mathbb{C}^{m_i}$ is identified with a $d_i \times m_i$ coefficient matrix C_i.

Theorem I.2.1 (Orbit dimension of one quantum seed). *For integers $d \geq 1$ and $m \geq 0$, consider the action of $M_d(\mathbb{C}) \otimes I_m$ on $\mathbb{C}^d \otimes \mathbb{C}^m$. Let C be the coefficient matrix of a vector ψ. Then*

$$\boxed{\dim \operatorname{span}\{(A \otimes I_m)\psi : A \in M_d(\mathbb{C})\} = d \operatorname{rank} C.}$$

The seed is cyclic if and only if $\operatorname{rank} C = m$*, which is possible only when $m \leq d$.*

Proof. Under the identification by coefficient matrices, $(A \otimes I_m)\psi$ corresponds to AC. Every row of AC lies in the row space of C. Conversely, any $d \times m$ matrix whose rows lie in the row space of C equals AC for a suitable A: choose each row of A to produce the desired row combination. The orbit therefore consists of all matrices with d independently chosen rows in the $\operatorname{rank} C$-dimensional row space of C, and has dimension $d \operatorname{rank} C$. It equals the full dm-dimensional space exactly when $\operatorname{rank} C = m$. □

Theorem I.2.2 (Quantum seed-rank formula). *For*

$$\mathcal{A} = \bigoplus_i (M_{d_i}(\mathbb{C}) \otimes I_{m_i}) \quad \text{on} \quad \mathcal{H} = \bigoplus_i (\mathbb{C}^{d_i} \otimes \mathbb{C}^{m_i}),$$

the minimum number of seeds required to generate $\mathcal{H}$ is

$$\boxed{\operatorname{srank}_{\mathcal{A}}(\mathcal{H}) = \max_i \left\lceil \frac{m_i}{d_i} \right\rceil.}$$

Proof. Each seed contributes a coefficient matrix of rank at most d_i in the ith block. Thus k seeds can cover at most a kd_i-dimensional subspace of the multiplicity factor $\mathbb{C}^{m_i}$, so $k \geq \lceil m_i/d_i \rceil$ for every i.

For the converse, take

$$k = \max_i \left\lceil \frac{m_i}{d_i} \right\rceil .$$

In each block, partition a basis of $\mathbb{C}^{m_i}$ into at most k groups of size at most d_i. Choose the coefficient matrix of seed a so that its nonzero rows span the ath group. The combined row spaces of the k seeds then span $\mathbb{C}^{m_i}$ in every block. The preceding theorem shows that their $\mathcal{A}$-orbits generate the whole direct sum. □

Corollary I.2.3 (Local orbit and Schmidt rank)**.** *Let $\psi \in \mathcal{H}_A \otimes \mathcal{H}_B$, with $\dim \mathcal{H}_A = d_A \geq 1$, and let r be its Schmidt rank. The reduced positive operator on B is $\rho_B = \mathrm{Tr}_A(|\psi\rangle\langle\psi|)$. Normalization of ψ does not affect its support. Then*

$$\boxed{\dim \mathrm{span}\{(A \otimes I_B)\psi : A \in M_{d_A}(\mathbb{C})\} = d_A r.}$$

In particular, local operators on A generate the complete bipartite Hilbert space exactly when the reduced state on B has full support.

Proof. A Schmidt decomposition of length r gives a coefficient matrix of rank r. Applying Theorem I.2.1 gives the orbit dimension $d_A r$. The range of ρ_B is the span of the r Schmidt vectors on B, so full support is equivalent to $r = \dim \mathcal{H}_B$. Because $d_A > 0$, this is equivalent to the orbit having the full bipartite dimension. □

The conclusion concerns the algebraic span obtained from arbitrary local operators. A normalized unitary gate orbit need not contain every vector, and the span statement does not supply a deterministic protocol for preparing arbitrary remote states. Entanglement controls the multiplicity support appearing in that algebraic span.

I.2.2 A computational-basis seed

Theorem I.2.4 (One seed for the computational basis)**.** *For an integer $n \geq 0$, let*

$$\psi_0 = |0\rangle^{\otimes n}$$

and let X_j denote the Pauli X operator on qubit j. Then

$$\{X_1^{a_1} \cdots X_n^{a_n} \psi_0 : (a_1, \ldots, a_n) \in \{0,1\}^n\}$$

is the complete computational basis. Hence the seed is cyclic under the algebra generated by the X_j.

Proof. For a binary address $a = (a_1, \ldots, a_n)$,

$$X_1^{a_1} \cdots X_n^{a_n} |0 \cdots 0\rangle = |a_1 \cdots a_n\rangle .$$

Distinct binary addresses produce distinct orthonormal basis vectors. There are 2^n of them, equal to the dimension of the Hilbert space. □

Recorded Numerical Check I.2.5 (Three-qubit seed orbit). The original numerical implementation generated all eight computational basis states from $|000\rangle$ using the three local X address operators. It reported an orbit rank of $8/8$.

I.3 Cyclic symmetry packets

I.3.1 Character projectors

Let $N \geq 1$ be an integer, and let S be a unitary symmetry whose order divides N:

$$S^N = I.$$

Set $\omega = e^{2\pi i/N}$.

Theorem I.3.1 (Cyclic quantum projectors). *The operators*

$$\boxed{P_j = \frac{1}{N}\sum_{r=0}^{N-1} \omega^{-jr} S^r, \qquad 0 \leq j < N,}$$

are self-adjoint, pairwise orthogonal projectors satisfying

$$P_j P_k = \delta_{jk} P_j, \qquad \sum_{j=0}^{N-1} P_j = I, \qquad S P_j = \omega^j P_j.$$

Proof. Using $S^{r+s} = S^{r+s \bmod N}$ and the finite character orthogonality relation,

$$\frac{1}{N}\sum_{r=0}^{N-1} \omega^{(k-j)r} = \delta_{jk},$$

we obtain $P_j P_k = \delta_{jk} P_j$. Summing over j cancels every term indexed by $r \neq 0$ and retains the $r = 0$ term, which proves completeness. Unitarity and the change of residue $r \mapsto -r$ give

$$P_j^* = \frac{1}{N}\sum_{r=0}^{N-1} \omega^{jr} S^{-r} = P_j.$$

Finally, multiplication by S shifts the summation index and contributes the factor ω^j. □

For parity,

$$S = Z^{\otimes n}, \qquad P_{\text{even}} = \frac{I+S}{2}, \qquad P_{\text{odd}} = \frac{I-S}{2}.$$

I.3.2 Tensor refinement

Fix $N \geq 1$. Suppose every factor a has a complete orthogonal family $P_0^{(a)}, \dots, P_{N-1}^{(a)}$. The fine tensor packets are

$$P_{\mathbf{j}}^{\text{fine}} = P_{j_1}^{(1)} \otimes \cdots \otimes P_{j_m}^{(m)}.$$

Theorem I.3.2 (Tensor/polyphase refinement). *For each residue $r \in \mathbb{Z}/N\mathbb{Z}$, define*

$$\boxed{P_r^{\text{coarse}} = \sum_{j_1+\cdots+j_m \equiv r \pmod N} P_{j_1}^{(1)} \otimes \cdots \otimes P_{j_m}^{(m)}.}$$

Then the P_r^{coarse} are complete orthogonal projectors, and every coarse packet is the exact sum of its fine descendants.

Proof. The fine tensor projectors are mutually orthogonal and sum to the identity. Grouping the fine addresses by their total residue partitions the full address set. Sums over disjoint sets of orthogonal projectors remain orthogonal projectors, and the union of the residue classes is the full fine address set. □

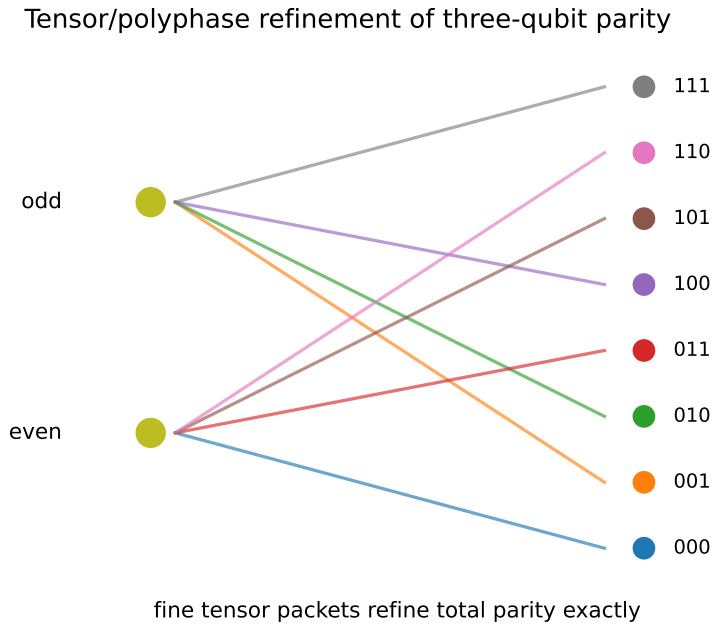


Figure 5. The eight computational packets of three qubits refine the two total-parity packets exactly.

Recorded Numerical Check I.3.3 (Exact parity refinement). For three qubits, the local computational projectors were grouped by the parity of their Hamming weight. The resulting coarse projectors agreed with $(I \pm Z^{\otimes 3})/2$ with residual 0.

I.4 Quantum circuits as packet paths

Let

$$U = U_m U_{m-1} \cdots U_1$$

be a circuit with $m \geq 1$, and let $\{P_\alpha\}$ be a complete packet system. For the empty circuit $m = 0$, set $U = I$. Its boundary block is $P_\beta P_\alpha = \delta_{\beta\alpha} P_\alpha$.

Theorem I.4.1 (Exact circuit path expansion). *For source packet α_0 and target packet α_m,*

$$\boxed{P_{\alpha_m} U P_{\alpha_0} = \sum_{\alpha_1,\ldots,\alpha_{m-1}} P_{\alpha_m} U_m P_{\alpha_{m-1}} \cdots P_{\alpha_1} U_1 P_{\alpha_0}.}$$

Summing over the source and target packets reconstructs the complete circuit operator.

Proof. Insert the identity resolution $I = \sum_\alpha P_\alpha$ between every adjacent pair of gates. Distributing the finite sums gives the path expansion. Summing the left and right boundary projectors reconstructs U. □

Theorem I.4.2 (Packetwise unitarity conservation). *If U is unitary and $U_{\beta\alpha} = P_\beta U P_\alpha$, then*

$$\boxed{\sum_\beta U^*_{\beta\alpha} U_{\beta\gamma} = \delta_{\alpha\gamma} P_\alpha.}$$

Similarly,

$$\sum_\alpha U_{\beta\alpha} U^*_{\delta\alpha} = \delta_{\beta\delta} P_\beta.$$

Proof. Using completeness and unitarity,

$$\sum_\beta U^*_{\beta\alpha} U_{\beta\gamma} = P_\alpha U^* \left(\sum_\beta P_\beta \right) U P_\gamma = P_\alpha P_\gamma.$$

For the second identity, sum over the source packets and use unitarity in the reverse order. □

Example I.4.3 (Bell seed circuit). In this example only, H denotes the Hadamard gate, not a Hamiltonian. Starting from $|00\rangle$,

$$(H \otimes I)\,|00\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}},$$

and then

$$\mathrm{CNOT}(H \otimes I)\,|00\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

The parity packets show that the final state lies entirely in the even sector, while the intermediate circuit path passes through both even and odd addresses.

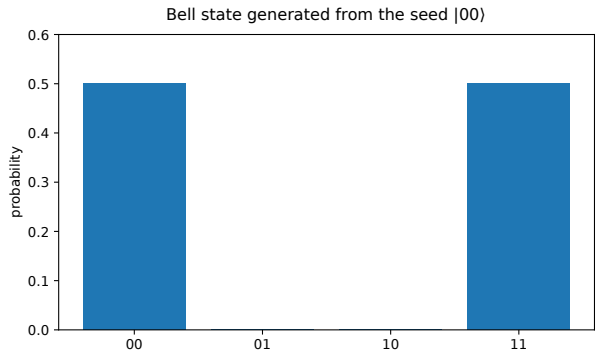


Figure 6. The Bell state generated from the seed $|00\rangle$.

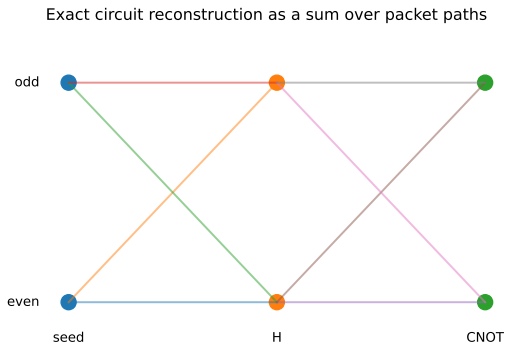


Figure 7. Circuit reconstruction as a sum over packet paths.

Recorded Numerical Check I.4.4 (Bell and path reconstruction). The original numerical implementation reported residuals of 0 for both the Bell state and its complete reconstruction from parity paths.

I.5 Symmetry-resolved Hamiltonians and leakage

Let $H = H^*$ and let P be a packet projector. If $[H, P] = 0$, then $P\mathcal{H}$ is invariant under e^{-itH}.

Theorem I.5.1 (Exact symmetry packets). *If $[H, P_\alpha] = 0$ for every packet, then*

$$H = \sum_\alpha P_\alpha H P_\alpha$$

and

$$e^{-itH} = \sum_\alpha P_\alpha e^{-itH} P_\alpha.$$

No amplitude transfers between distinct packets.

Proof. The commutation relation gives $P_\beta H P_\alpha = 0$ for $\beta \neq \alpha$. Every polynomial in H has the same block form, and the finite-dimensional exponential is the norm limit of its polynomial series. □

For two qubits and real parameters J, h, consider

$$H = JX \otimes X + h(Z \otimes I + I \otimes Z).$$

In the ordered even basis $\{|00\rangle, |11\rangle\}$ and odd basis $\{|01\rangle, |10\rangle\}$,

$$H_{\text{even}} = \begin{pmatrix} 2h & J \\ J & -2h \end{pmatrix}, \qquad H_{\text{odd}} = \begin{pmatrix} 0 & J \\ J & 0 \end{pmatrix}.$$

Therefore

$$\operatorname{Spec} H_{\text{even}} = \{\pm\sqrt{4h^2 + J^2}\}, \qquad \operatorname{Spec} H_{\text{odd}} = \{\pm J\}.$$

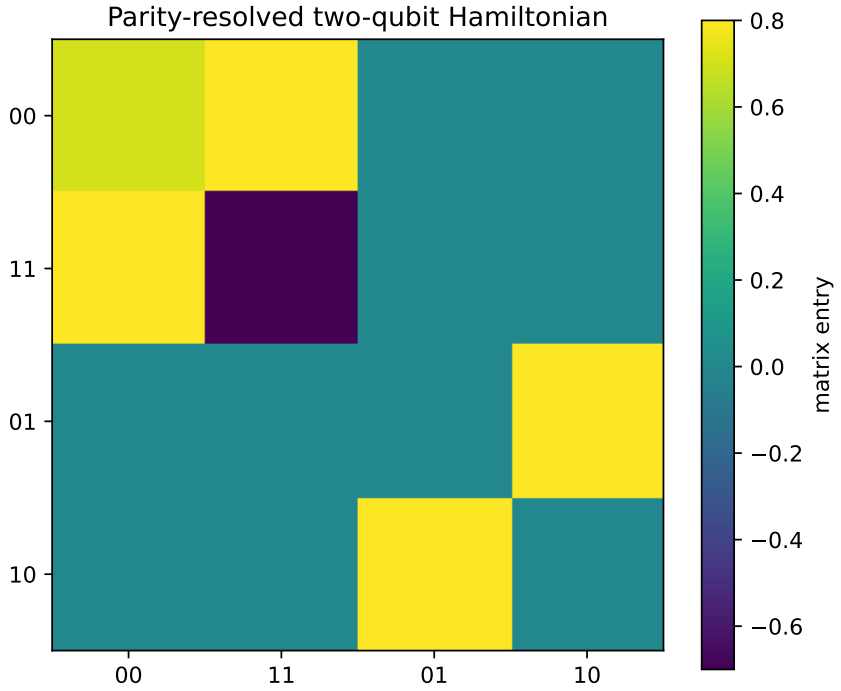


Figure 8. The Hamiltonian becomes exactly block diagonal when the even and odd basis vectors are grouped.

I.5.1 Leakage under broken symmetry

Set $Q = I - P$.

Theorem I.5.2 (Commutator leakage bound). *For every self-adjoint H, orthogonal projector P, and real time t,*

$$\boxed{\left\|Qe^{-itH}P\right\| \le |t|\ \|[H,P]\|\,.}$$

In particular, exact commutation gives zero leakage for all time.

Proof. Set $F(s) = e^{isH}Pe^{-isH}$. Differentiating the exponential gives

$$F'(s) = i\,e^{isH}[H,P]e^{-isH}, \qquad F(0) = P.$$

Unitary conjugation preserves the operator norm, so $\|F'(s)\| = \|[H,P]\|$. The mean-value estimate on the interval between 0 and t yields

$$\|F(t) - P\| \le |t|\,\|[H,P]\|.$$

Finally, $e^{itH}Qe^{-itH}P = (P - F(t))P$ and $\|P\| \le 1$. Left multiplication by e^{itH} preserves the norm and gives the desired inequality. This argument includes positive, negative and zero time. □

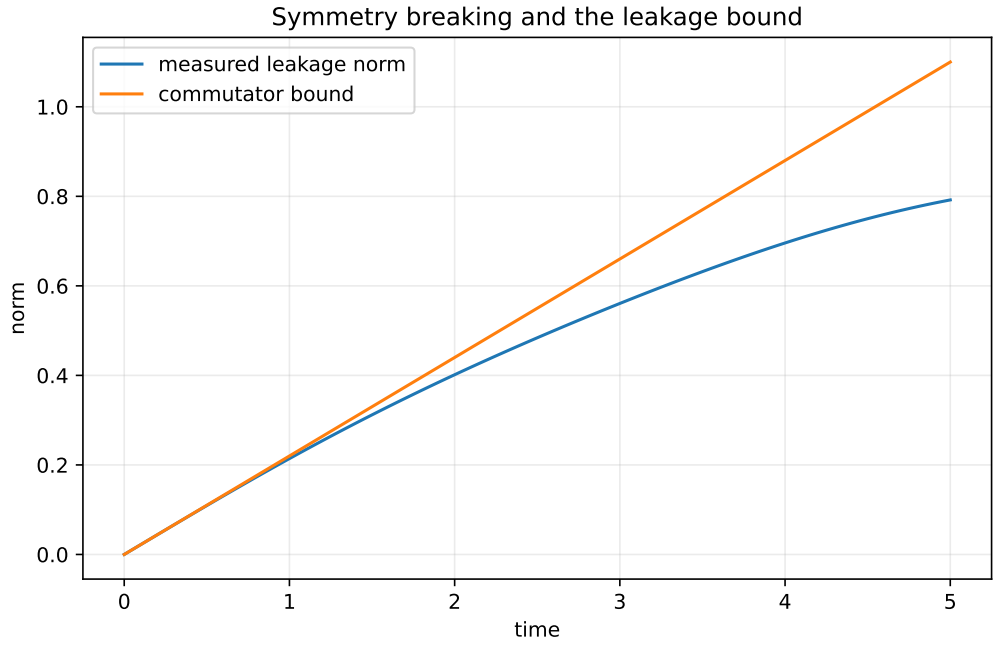


Figure 9. A perturbation that breaks parity creates nonzero transfer between the packets. The measured leakage remains below the commutator bound.

Recorded Numerical Check I.5.3 (Symmetry and negative control). For the Hamiltonian that preserves parity, the norm of the block between distinct packets and the maximum measured leakage were both 0. Adding an $X \otimes I$ perturbation broke parity and produced a maximum leakage of approximately 0.79196. The maximum violation of the proved bound remained 0.

I.6 Moving packets, connections, and curvature

I.6.1 Moving quantum coordinates

Theorem I.6.1 (Moving-frame Hamiltonian). *Let $U(t)$ and $\psi(t)$ be differentiable on an open real time interval, let $U(t)$ be unitary, and let $H(t) = H(t)^*$. Suppose*

$$i\dot{\psi}(t) = H(t)\psi(t).$$

Writing

$$\psi(t) = U(t)y(t)$$

gives

$$\boxed{i\dot{y}(t) = \big(U(t)^*H(t)U(t) - iU(t)^*\dot{U}(t)\big)\, y(t).}$$

Both the connection term $-iU^*\dot{U}$ and the effective Hamiltonian $U^*HU - iU^*\dot{U}$ are Hermitian.

Proof. Differentiate $\psi = Uy$:

$$i\dot{U}y + iU\dot{y} = HUy.$$

Multiplying by U^* gives the formula. Differentiating $U^*U = I$ shows $U^*\dot{U}$ is skew-Hermitian, so $-iU^*\dot{U}$ is Hermitian. □

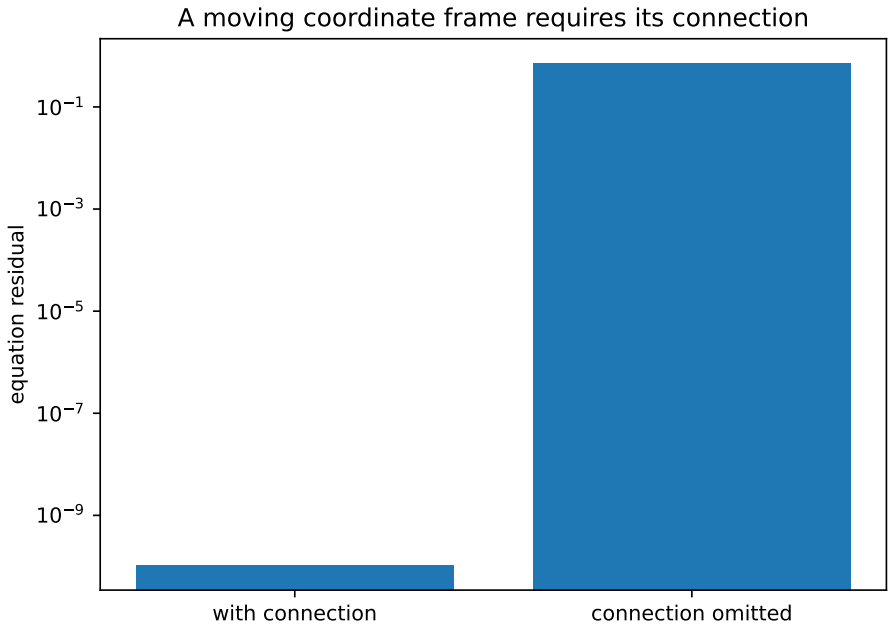


Figure 10. The equation in the moving frame holds to numerical precision only when the connection term is retained.

Recorded Numerical Check I.6.2 (Moving-frame correction). The reported residual for the complete equation in the moving frame was 1.0645×10^{-10}. Omitting the connection increased the residual to 0.7000.

I.6.2 A canonical projector connection

Let $P_1(t), \dots, P_m(t)$ be differentiable on an open real time interval, with $P_a(t)^* = P_a(t)$ and

$$P_a P_b = \delta_{ab} P_a, \qquad \sum_a P_a = I.$$

Theorem I.6.3 (Canonical packet connection). *Define*

$$\boxed{C(t) = \frac{1}{2}\sum_a [\dot{P}_a(t), P_a(t)] = \sum_a \dot{P}_a(t) P_a(t).}$$

Then

$$C(t)^* = -C(t)$$

and

$$\boxed{[C(t), P_b(t)] = \dot{P}_b(t)}$$

for every packet b.

Proof. Differentiating $P_a^2 = P_a$ gives

$$\dot{P}_a P_a + P_a \dot{P}_a = \dot{P}_a.$$

Summing over a and using $\sum_a \dot{P}_a = 0$ yields

$$\sum_a P_a \dot{P}_a = -\sum_a \dot{P}_a P_a,$$

which proves both displayed formulas for C and its skew-Hermiticity.

For fixed b,

$$C P_b = \dot{P}_b P_b.$$

When $a \neq b$, differentiating $P_b P_a = 0$ gives

$$P_b \dot{P}_a = -\dot{P}_b P_a.$$

Hence

$$P_b C = \sum_a P_b \dot{P}_a P_a = -\dot{P}_b (I - P_b),$$

and subtraction gives $[C, P_b] = \dot{P}_b$. □

The connection generates infinitesimal transport of the moving packet frame. The packet family determines it directly, so its construction does not require a basis inside any packet.

I.6.3 Projector curvature

Classical Input I.6.4 (Projected connection). Let M be a smooth finite-dimensional real manifold without boundary, let V be a finite-dimensional real normed vector space, and let $P : M \to \mathrm{End}_{\mathbb{R}}(V)$ be smooth with $P(x)^2 = P(x)$. In the quantum application, V is the underlying real space of a finite complex Hilbert space and P is an orthogonal complex-linear projector.

The rank of P is locally constant. On each open rank region, its ranges form a smooth vector bundle. For a smooth range section s, so that $Ps = s$, define

$$\nabla_X s = P\, ds(X).$$

This is a connection on smooth range sections. For smooth vector fields X, Y, its curvature satisfies

$$\boxed{(\nabla_X \nabla_Y - \nabla_Y \nabla_X - \nabla_{[X,Y]})s = P[dP(X), dP(Y)]P\, s.}$$

Equivalently, its endomorphism-valued curvature is

$$\boxed{F = P(dP \wedge dP)P.}$$

No global frame or globally constant rank is required, and the vector fields need not commute. The geometric formalization concerns real smooth structures. It does not cover complex analytic manifolds or extensions to a boundary.

Proof. Range coordinates. Fix $a \in M$ and put

$$T_a(x) = P(x)P(a) + (I - P(x))(I - P(a)).$$

Then $T_a(a) = I$ and $P(x)T_a(x) = T_a(x)P(a)$. Invertibility is open, so on a neighborhood of a, $T_a(x)$ restricts to an isomorphism from Ran $P(a)$ to Ran $P(x)$. This proves local constancy of rank. The local coordinates and their transitions are

$$(x, v) \longmapsto (x, T_a(x)^{-1}v), \qquad u \longmapsto T_b(x)^{-1}T_a(x)u.$$

Their smoothness follows from smoothness of P and inversion. The transition maps satisfy the cocycle identity by cancellation. These charts construct the smooth range bundles on the open rank regions, including when different connected components have different ranks.

Connection rules. The smoothness of $P\,ds(X)$ and the identity $P^2 = P$ show that $\nabla_X s$ is a smooth range section. Differentiation and projection give

$$\begin{aligned} \nabla_{X+Y}s = \nabla_X s + \nabla_Y s, \qquad \nabla_{fX}s = f\nabla_X s, \\ \nabla_X(s+t) = \nabla_X s + \nabla_X t, \quad \nabla_X(fs) = f\nabla_X s + (Xf)s \end{aligned}$$

for smooth real f. The range coordinates above identify ambient smooth range sections with smooth sections of the constructed bundle.

Curvature. Differentiating $P^2 = P$ gives $P(dP(X))P = 0$. Write $D_X s = ds(X)$. For ambient smooth sections,

$$D_X D_Y s - D_Y D_X s - D_{[X,Y]}s = 0.$$

Expansion of the projected derivatives therefore leaves

$$R(X, Y)s = P(dP(X))D_Y s - P(dP(Y))D_X s.$$

Since $Ps = s$, differentiation gives $D_Y s = (dP(Y))s + PD_Y s$. The terms with a middle P vanish. Thus

$$R(X, Y)s = P\big(dP(X)dP(Y) - dP(Y)dP(X)\big)P\,s.$$

This expression is alternating and bilinear in the tangent vectors, and it preserves the range. The ordinary chain rule and the transformation law for the Lie bracket make the local calculations agree on chart overlaps, which gives the global formula. □

For an orthogonal rank-one complex projector, use the scalar connection convention $A = -i\langle u, du\rangle$ in a local unit frame. The associated scalar curvature density is

$$\mathcal{F}_{\mu\nu} = -i\,\mathrm{Tr}\left(P[\partial_\mu P, \partial_\nu P]\right).$$

On the Bloch sphere,

$$P(\theta, \phi) = \frac{1}{2}\left(I + \mathbf{n}(\theta, \phi)\cdot\sigma\right),$$

where

$$\mathbf{n} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta).$$

The Pauli multiplication law gives

$$[\partial_\theta P, \partial_\phi P] = \frac{i}{2}(\partial_\theta \mathbf{n} \times \partial_\phi \mathbf{n})\cdot\sigma.$$

Using $\operatorname{Tr}(P\,\mathbf{a}\cdot\sigma)=\mathbf{n}\cdot\mathbf{a}$ therefore yields

$$\boxed{\mathcal{F}_{\theta\phi}=\frac{1}{2}\mathbf{n}\cdot(\partial_\theta\mathbf{n}\times\partial_\phi\mathbf{n})=\frac{1}{2}\sin\theta.}$$

The scalar sign changes if the opposite convention for A is chosen. This explicit calculation is not a separate entry among the 19 items in the Lean register.

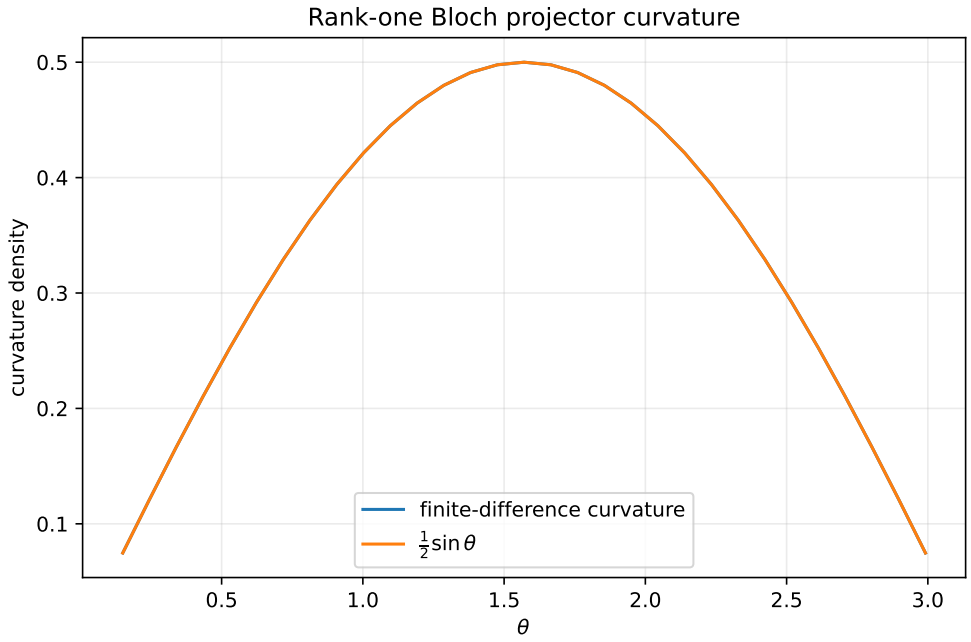


Figure 11. The curvature of the Bloch projector, evaluated by finite differences, is compared with the exact density $\frac{1}{2}\sin\theta$.

Recorded Numerical Check I.6.5 (Geometric packet identities). The canonical connection reconstructed every packet derivative with residual 0. The maximum residual for the Bloch curvature was 2.0803×10^{-11}.

I.7 Measurements and quantum channels

I.7.1 Packet measurements

Let ρ be a density matrix and let $\{E_k\}$ be a POVM:

$$E_k\geq 0,\qquad \sum_k E_k=I.$$

The outcome probabilities are

$$p_k=\operatorname{Tr}(E_k\rho).$$

Theorem I.7.1 (Block-diagonal measurement invariance). *Let $\{P_\alpha\}$ be a complete packet family, and suppose every measurement effect is packet diagonal:*

$$E_k = \sum_\alpha P_\alpha E_k P_\alpha.$$

Define the packet-dephased state

$$\mathcal{D}_P(\rho) = \sum_\alpha P_\alpha \rho P_\alpha.$$

Then

$$\boxed{\mathrm{Tr}(E_k\rho) = \mathrm{Tr}(E_k \mathcal{D}_P(\rho))}$$

for every outcome k.

Proof. Insert the packet decomposition of E_k and use cyclicity of the trace:

$$\mathrm{Tr}(E_k\rho) = \sum_\alpha \mathrm{Tr}(P_\alpha E_k P_\alpha \rho) = \sum_\alpha \mathrm{Tr}(E_k P_\alpha \rho P_\alpha).$$

□

The theorem identifies which coherences a declared measurement can detect. Observables that are diagonal in the packet decomposition cannot detect coherence between distinct packets. Coherence within a packet remains available to such observables.

I.7.2 Kraus transfer

A quantum channel can transfer both populations and coherences between packet addresses. A Kraus representation describes this transfer by summing the contributions of its individual branches. The normalization of those branches ensures that the total trace is preserved.

Classical Input I.7.2 (Finite Kraus channel). Let $K_1, \dots, K_r$ satisfy

$$\sum_a K_a^* K_a = I.$$

Then

$$\boxed{\Phi(\rho) = \sum_a K_a \rho K_a^*}$$

is completely positive and trace preserving.

Proof. Trace preservation follows from

$$\mathrm{Tr}\,\Phi(\rho) = \mathrm{Tr}\left(\rho \sum_a K_a^* K_a\right) = \mathrm{Tr}\,\rho.$$

For any auxiliary dimension and positive matrix X, each term

$$(I \otimes K_a) X (I \otimes K_a)^*$$

is positive, so their sum is positive. This is complete positivity. □

Packet superoperator blocks are defined by

$$\Phi_{\alpha\beta;\gamma\delta}(X) = P_\alpha \Phi(P_\gamma X P_\delta) P_\beta.$$

Summing all four packet indices reconstructs $\Phi(X)$ exactly.

For amplitude damping with a real parameter $0 \le \gamma \le 1$,

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \qquad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}.$$

The excited state becomes

$$\Phi(|1\rangle\langle 1|) = \gamma\,|0\rangle\langle 0| + (1-\gamma)\,|1\rangle\langle 1|\,.$$

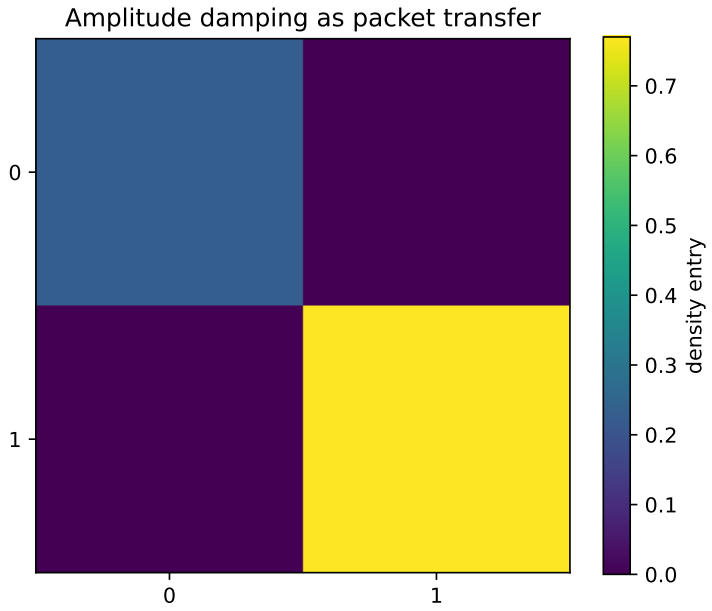


Figure 12. Amplitude damping transfers population from the excited packet to the ground packet while preserving trace and positivity.

Recorded Numerical Check I.7.3 (Channel reconstruction). For $\gamma = 0.23$, the residuals for Kraus completeness and the output trace were 0. The computed output from $|1\rangle\langle 1|$ was $\mathrm{diag}(0.23, 0.77)$. In the Bell state test, packet dephasing left every measurement probability in the computational basis unchanged.

I.8 Quantum error correction as obstruction-packet annihilation

Let $P = P^* = P^2$ project onto a code space $\mathcal{C} \subset \mathcal{H}$, and let $E_1, \dots, E_r$ be a finite family of complex error matrices. Neither a nonzero code nor channel normalization is assumed unless stated. The operators

$$PE_a^* E_b P$$

are the obstruction packets on the code. If they retain information about the logical state beyond one scalar multiple of P, the environment can distinguish logical states and exact correction is impossible.

We call a recovery complete when it is a Kraus map

$$\mathcal{R}(X) = \sum_j K_j X K_j^*, \qquad \sum_j K_j^* K_j = I.$$

The family is *exactly correctable* when one such recovery works for every coherent error combination $A_w = \sum_a w_a E_a$: for every w there is a real $t_w \geq 0$, independent of the logical input, such that

$$PXP = X \quad \Longrightarrow \quad \mathcal{R}(A_w X A_w^*) = t_w X.$$

The coefficient is an unnormalized weight. It is not necessarily a probability for an arbitrary matrix A_w.

Classical Input I.8.1 (Finite Knill–Laflamme criterion). The finite error family is exactly correctable in the sense just defined if and only if there is a positive semidefinite complex matrix $C = (c_{ab})$ such that

$$\boxed{PE_a^* E_b P = c_{ab} P \qquad \text{for every } a, b.}$$

If $\sum_a E_a^* E_a = I$, the equivalent normalized-channel formulation is the existence of a complete recovery satisfying

$$\mathcal{R}\left(\sum_a E_a X E_a^*\right) = X \quad \text{for every } X \text{ with } PXP = X.$$

The zero code and an empty error family are included. An empty family cannot define a channel that preserves trace on a nonzero ambient space.

Proof. Necessity. Fix an error E_a and let $L_j = K_j E_a P$. Perfect recovery on rank-one code operators gives

$$\sum_j |L_j\psi\rangle \langle L_j\psi| = t_a |\psi\rangle \langle\psi| \qquad (\psi \in \mathcal{C}).$$

For every vector orthogonal to ψ, the quadratic form on the left is a sum of nonnegative squared absolute values and equals zero. Each $L_j\psi$ therefore lies in the span of ψ. A linear operator that preserves every line in $\mathcal{C}$ is scalar there: apply it to two independent vectors and their sum. The one-dimensional case is immediate, and the claim is vacuous for the zero code. Consequently

$$K_j E_a P = d_{ja} P.$$

Recovery completeness now gives

$$PE_a^*E_bP = \sum_j (K_jE_aP)^*(K_jE_bP)$$
$$= \left(\sum_j \overline{d_{ja}}d_{jb}\right) P.$$

The coefficients form a Gram matrix, hence are positive semidefinite. For a normalized channel, apply the same rank-one argument to all composite branches K_jE_aP at once to obtain the necessary condition.

Sufficiency. Diagonalize $C = U\,\mathrm{diag}(\lambda_r)U^*$ with U unitary and $\lambda_r \geq 0$. Define compressed transformed errors

$$F_r = \sum_a U_{ar}E_aP.$$

Then $F_rP = F_r$ and $F_r^*F_s = \lambda_r\delta_{rs}P$. If $\lambda_r = 0$, $F_r^*F_r = 0$ forces $F_r = 0$. For positive eigenvalues only, set $V_r = F_r/\sqrt{\lambda_r}$. These satisfy

$$V_rP = V_r, \qquad V_r^*V_s = \delta_{rs}P.$$

Let $S = \sum_{\lambda_r>0} V_rV_r^*$. Orthogonality makes S an orthogonal projector. Use the recovery branches

$$K_r = V_r^*, \qquad K_{\text{unused}} = I - S.$$

They are complete on the whole ambient space because

$$\sum_r K_r^*K_r + K_{\text{unused}}^*K_{\text{unused}} = S + (I-S)^2 = I.$$

The inverse unitary change of coordinates reconstructs each E_aP from the F_r. Every recovery branch of a coherent combination is therefore scalar on the code, while the branch for the unused space vanishes. For all supported matrices, the resulting recovery is

$$\mathcal{R}(A_wXA_w^*) = (w^*Cw)X.$$

This establishes the statement for arbitrary families without assuming normalization. Under $\sum_a E_a^*E_a = I$ and $P \neq 0$, summing the diagonal KL equations gives $\mathrm{Tr}\,C = 1$, so the normalized-channel output is X. When $P = 0$, every supported X is zero and the conclusion is automatic. The recovery is trace preserving by completeness and completely positive at every finite ancillary dimension by the Kraus calculation in Classical Input I.7.2. □

For the three-qubit bit-flip code,

$$|0_L\rangle = |000\rangle\,, \qquad |1_L\rangle = |111\rangle\,,$$

and the error family is

$$I, \quad X\otimes I\otimes I, \quad I\otimes X\otimes I, \quad I\otimes I\otimes X.$$

The four error images of the code are pairwise orthogonal two-dimensional syndrome packets whose projectors sum to the identity on the eight-dimensional Hilbert space.

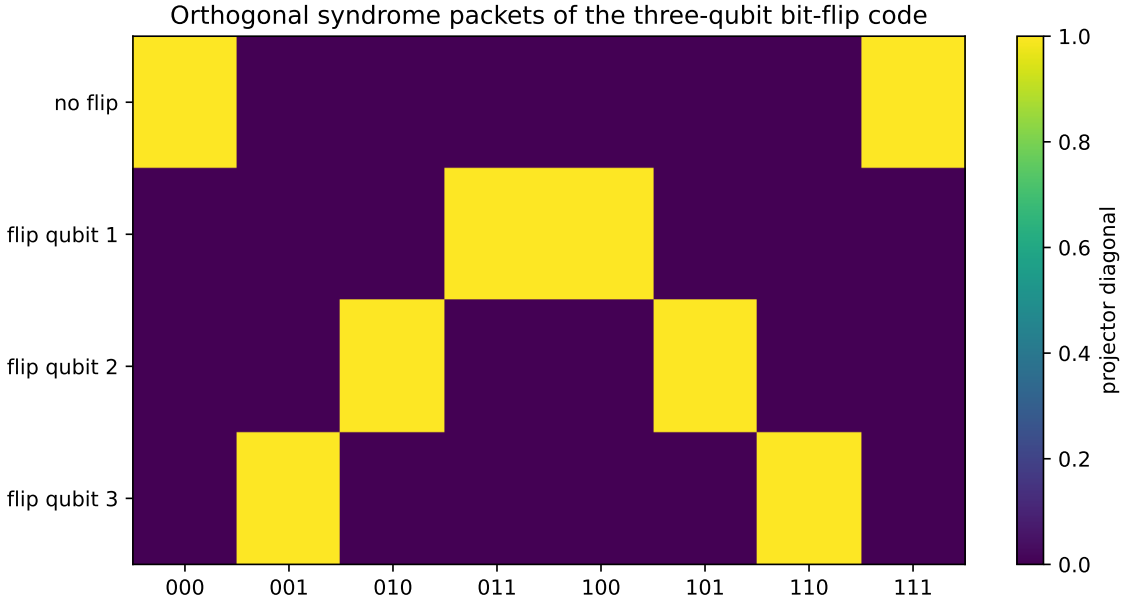


Figure 13. The four orthogonal syndrome packets of the three-qubit bit-flip code. Each packet identifies one correctable error without resolving the encoded logical state.

Recorded Numerical Check I.8.2 (Exact bit-flip correction). The residuals for the Knill–Laflamme condition and completeness of the syndrome packets were both 0. A mixed channel of single bit flip errors, applied to a nontrivial logical superposition, was reversed with a residual of 1.0385×10^{-16} in the density matrix.

I.9 Exact effective dynamics

I.9.1 Schur–Feshbach reduction

Let P be a retained packet projector and $Q = I - P$. In packet coordinates,

$$H = \begin{pmatrix} H_{PP} & H_{PQ} \\ H_{QP} & H_{QQ} \end{pmatrix}.$$

Theorem I.9.1 (Exact Feshbach resolvent). *Assume that both $zI_Q - H_{QQ}$ and $zI - H$ are invertible. Define*

$$\boxed{H_{\text{eff}}(z) = H_{PP} + H_{PQ}(z - H_{QQ})^{-1} H_{QP}.}$$

Then

$$\boxed{P(z - H)^{-1} P = \left(z - H_{\text{eff}}(z)\right)^{-1}}$$

where the left side is restricted to $P\mathcal{H}$, the right side is an inverse on that retained space, and both are written in the same orthonormal basis. Here $z - H_{QQ}$ and $z - H_{\text{eff}}$ abbreviate the appropriately sized scalar matrices minus the indicated blocks.

Proof. The block in the lower right corner of $z - H$ is $z - H_{QQ}$. Gaussian block elimination produces the Schur complement

$$z - H_{PP} - H_{PQ}(z - H_{QQ})^{-1}H_{QP}.$$

The assumed invertibility of this block and of the full matrix implies that the Schur complement is invertible. The block in the upper left corner of the full inverse is the inverse of that Schur complement, which is exactly the displayed operator. □

The second premise cannot be dropped: at $z = 0$ and $H = \mathrm{diag}(0, -1)$, the eliminated block is 1 but $zI - H$ is singular. This counterexample is also included in the Lean module.

The self-energy

$$\Sigma_P(z) = H_{PQ}(z - H_{QQ})^{-1}H_{QP}$$

contains the complete influence of the eliminated packet. Bare truncation drops $\Sigma_P(z)$ and is therefore exact only when the coupling between packets vanishes or the self-energy cancels for another stated reason.

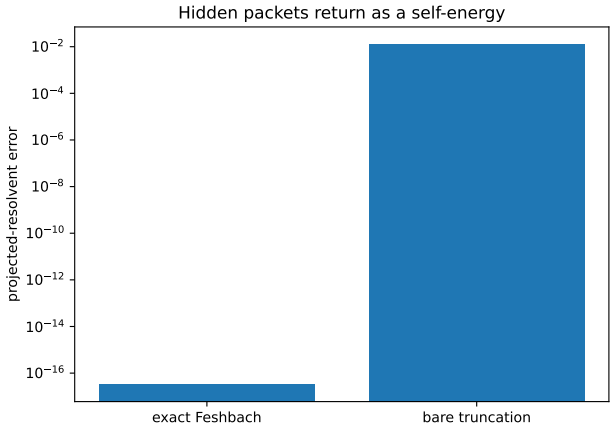


Figure 14. Exact Feshbach reduction versus bare truncation.

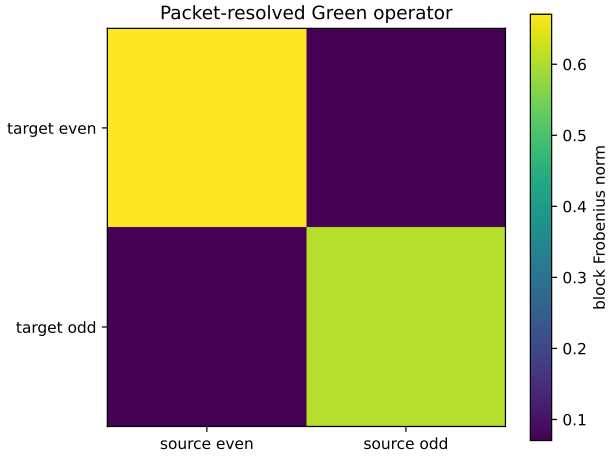


Figure 15. Norms of packet Green-function blocks.

Recorded Numerical Check I.9.2 (Effective resolvent). In the reported test with a random Hermitian matrix of dimension six, the exact Feshbach residual was 3.2291×10^{-17}. The corresponding residual for bare truncation was 1.3091×10^{-2}.

I.9.2 The hidden packet as memory

For a time-independent finite Hamiltonian H and fixed projectors P, Q, write $\psi = x + y$, with $x = P\psi$ and $y = Q\psi$. Then

$$i\dot{x} = H_{PP}x + H_{PQ}y,$$
$$i\dot{y} = H_{QP}x + H_{QQ}y.$$

Solving the second equation gives

$$y(t) = e^{-itH_{QQ}}y(0) - i\int_0^t e^{-i(t-s)H_{QQ}}H_{QP}x(s)\,ds.$$

Substitution yields the exact Volterra equation

$$\boxed{\begin{aligned} i\dot{x}(t) &= H_{PP}x(t) + H_{PQ}e^{-itH_{QQ}}y(0) \\ &\quad - i\int_0^t K(t-s)x(s)\,ds, \end{aligned}}$$

with memory kernel

$$\boxed{K(t) = H_{PQ}e^{-itH_{QQ}}H_{QP}.}$$

The influence of the eliminated degrees of freedom is therefore retained in a self-energy that depends on frequency, or equivalently in an exact memory term. For forward evolution the integral is causal. When the Hamiltonian depends on time, the corresponding propagator between two times replaces the stationary exponential used here. This Volterra derivation is not a separate entry in the finite Lean register.

I.10 Finite quantum geometry

I.10.1 A projector-resolved geometry

At resolution ℓ, define

$$\boxed{\mathfrak{Q}_\ell = \left(\mathcal{A}_\ell, \mathcal{H}_\ell, D_\ell, \{P_{\ell,\alpha}\}, E_\ell, C_\ell, \Pi_\ell^{\mathrm{phys}}, T_\ell\right).}$$

In order, the entries specify the observable algebra, Hilbert space, geometric or Dirac-type operator, packet projectors, embedding into the fine space, reconstruction onto the coarse space, physical constraint projector, and transfer operator. The maps have types $E_\ell : \mathcal{H}_\ell \to \mathcal{H}_{\ell+1}$ and $C_\ell : \mathcal{H}_{\ell+1} \to \mathcal{H}_\ell$. The left inverse relation $C_\ell E_\ell = I$ alone does not imply $C_\ell = E_\ell^*$ or make the embedding isometric.

A coherent refinement should satisfy

$$C_\ell E_\ell = I$$

and

$$\boxed{P_{\ell,\alpha} = C_\ell \left(\sum_{\beta \succ \alpha} P_{\ell+1,\beta}\right) E_\ell.}$$

For an operator D, define the refinement defect

$$\Delta_\ell(D) = C_\ell D_{\ell+1} E_\ell - D_\ell.$$

Exact resolution independence is $\Delta_\ell(D) = 0$. When exact equality is too strong, the correct continuum problem is to prove convergence of the resolvents and uniform control of the effective self-energies.

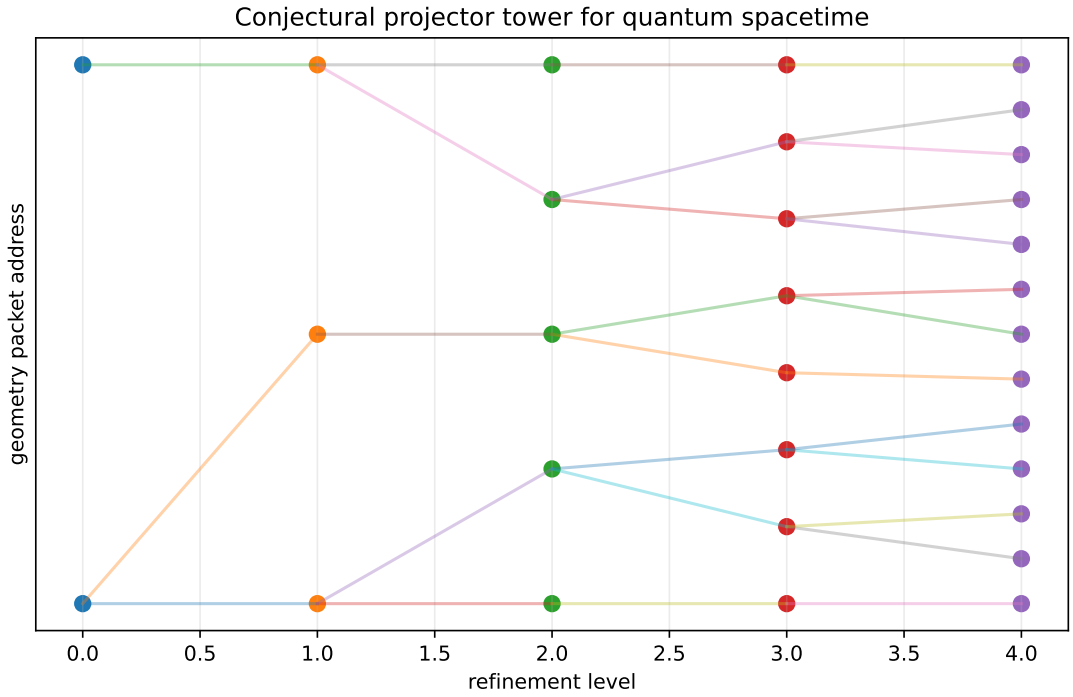


Figure 16. The diagram organizes a conjectural nested packet tower for quantum geometry. It does not provide a numerical reconstruction of physical spacetime.

I.10.2 Euclidean transfer

Theorem I.10.1 (Finite Euclidean transfer). *Let H be Hermitian on a nonzero finite-dimensional Hilbert space. For all real $\tau, \sigma \geq 0$,*

$$T(\tau) = e^{-\tau(H - E_0 I)}, \qquad E_0 = \min \operatorname{Spec} H,$$

is positive definite and satisfies

$$\boxed{T(\tau + \sigma) = T(\tau)T(\sigma).}$$

Its spectrum is contained in $(0, 1]$.

Proof. Diagonalize $H = \sum_j E_j P_j$. Then

$$T(\tau) = \sum_j e^{-\tau(E_j - E_0)} P_j.$$

Every eigenvalue is strictly positive and at most one. Multiplication adds the exponents, proving the semigroup law and $T(0) = I$. Positivity here is operator positivity, not an assertion that every matrix entry is nonnegative. □

The paper tests a controlled volume-state model

$$H_{nn} = 2\kappa + V(n), \qquad H_{n,n+1} = H_{n+1,n} = -\kappa,$$

with $\kappa > 0$ and a real confining potential. This is a finite transfer-matrix test only. Operator positivity alone is not reflection positivity of a Euclidean field measure. It is not derived from a gravitational path integral and is not evidence for a unique microscopic spacetime dynamics.

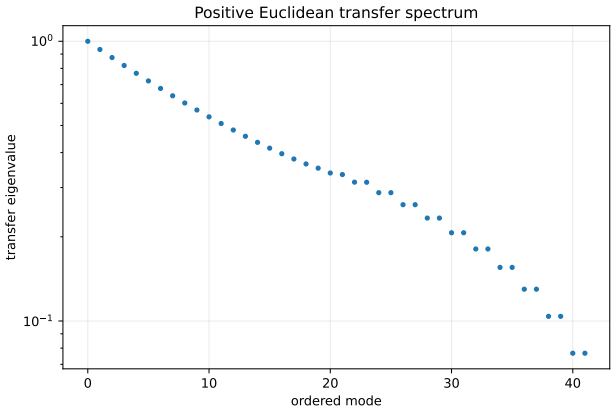


Figure 17. Positive spectrum of the Euclidean transfer operator.

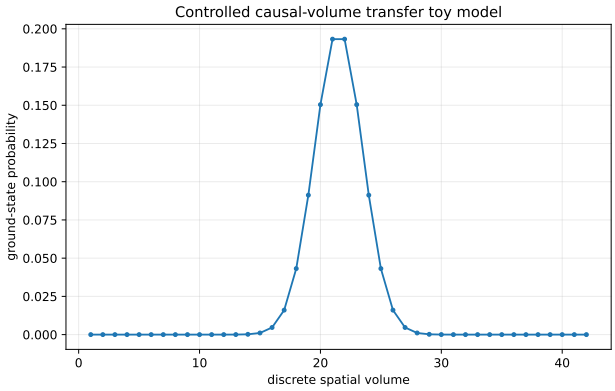


Figure 18. Ground-state volume packet in the controlled toy model.

Recorded Numerical Check I.10.2 (Finite transfer model). The minimum transfer eigenvalue was 0.0767054, the semigroup residual was 0, and reconstruction from four volume-residue packets had residual 0.

I.11 Spectral dimension and continuum diagnostics

Let $L \geq 0$ be a finite Laplacian on $N \geq 1$ states, with eigenvalues λ_j and at least one zero eigenvalue. For diffusion time $\tau > 0$, define the normalized heat trace

$$K(\tau) = \frac{1}{N} \sum_j e^{-\tau \lambda_j}$$

and the scale-dependent spectral dimension

$$\boxed{d_s(\tau) = -2 \frac{d \log K(\tau)}{d \log \tau}.}$$

For a finite system, $d_s(\tau)$ is a diagnostic, not a topological invariant. Indeed, finite differentiation gives

$$d_s(\tau) = 2\tau \frac{\sum_j \lambda_j e^{-\tau \lambda_j}}{\sum_j e^{-\tau \lambda_j}}.$$

It tends to zero as $\tau \downarrow 0$ and, under the stated assumption of a zero mode, as $\tau \to \infty$. If a general positive operator has no zero mode, its behavior at large times is instead asymptotic to $2\tau\lambda_{\min}$. The conclusion for operators with a zero mode therefore does not apply to that case. Inferring a geometric dimension requires a stable intermediate window and a controlled refinement limit.

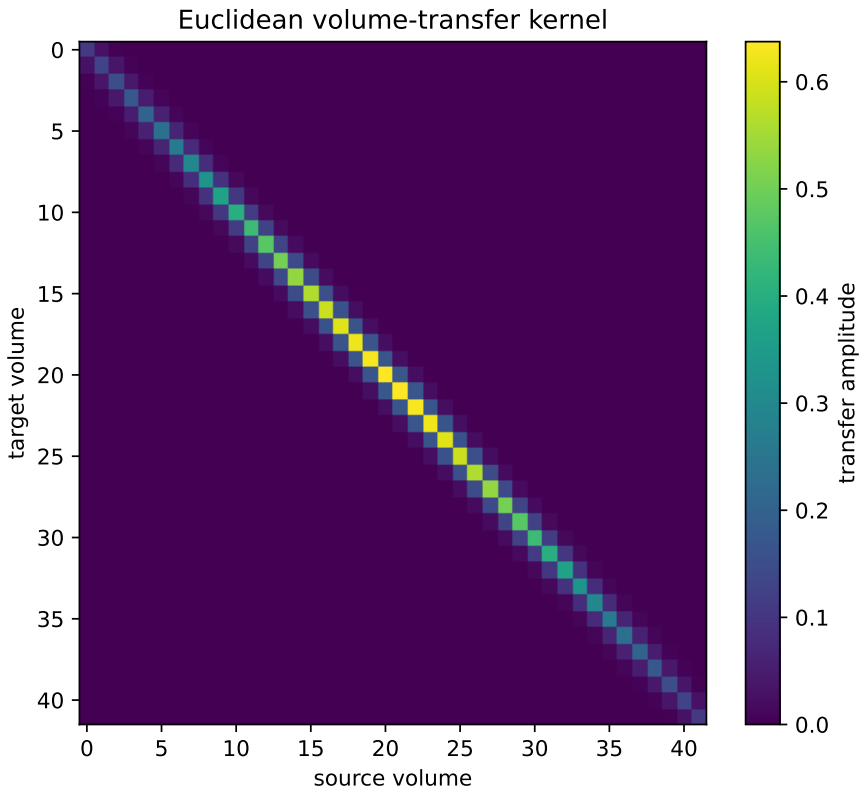


Figure 19. The figure shows the finite Euclidean kernel for volume transfer. Positivity and semigroup composition hold exactly in the finite model. Its gravitational interpretation remains provisional.

I.11.1 The circle dispersion bound

For a periodic grid with spacing $h > 0$ and physical wave number $k \in \mathbb{R}$,

$$\lambda_{k,h} = \frac{4}{h^2} \sin^2\left(\frac{kh}{2}\right).$$

Theorem I.11.1 (Low-mode second-order bound). *For every real k and every nonzero real h, define $\lambda_{k,h} = 4h^{-2}\sin^2(kh/2)$. Then*

$$\boxed{0 \le k^2 - \lambda_{k,h} \le \frac{k^4 h^2}{12}.}$$

Proof. Put $x = kh/2$. The inequalities $|\sin x| \le |x|$ and $|x - \sin x| \le |x|^3/6$, which hold for every real argument, give

$$0 \le x^2 - \sin^2 x = |x - \sin x|\,|x + \sin x| \le \frac{|x|^3}{6}\, 2|x| = \frac{x^4}{3}.$$

Multiplying by $4/h^2$ proves both bounds. The estimate holds for every frequency. Its interpretation as convergence of second order keeps k fixed while $h \to 0$, so it does not assert a uniformly small relative error across all grid modes. □

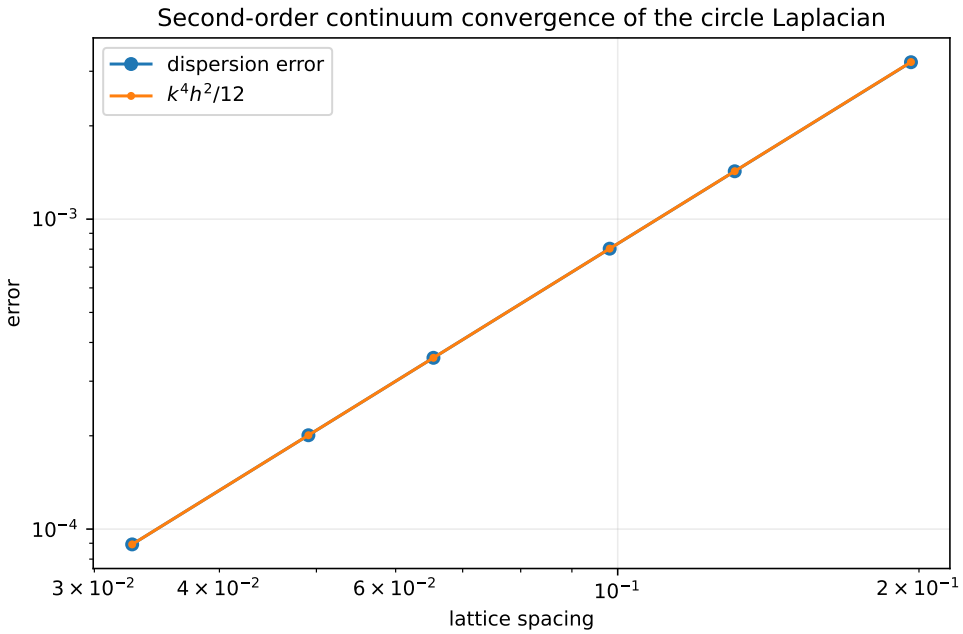


Figure 20. The first nonzero circle mode converges at second order and remains below the proved dispersion bound.

Recorded Numerical Check I.11.2 (Continuum dispersion). The reported convergence order was 1.99936. For every tested mode, the error was below the bound, so none of those tests violated the theorem.

I.11.2 One- and two-dimensional plateaus

At intermediate diffusion times, a finite cycle and a finite torus can exhibit plateaus near the dimensions of their underlying geometries. The following figure and numerical certificate show the reported behavior for these two examples. The plateaus occur between the small and large time regimes in which the spectral dimension returns to zero.

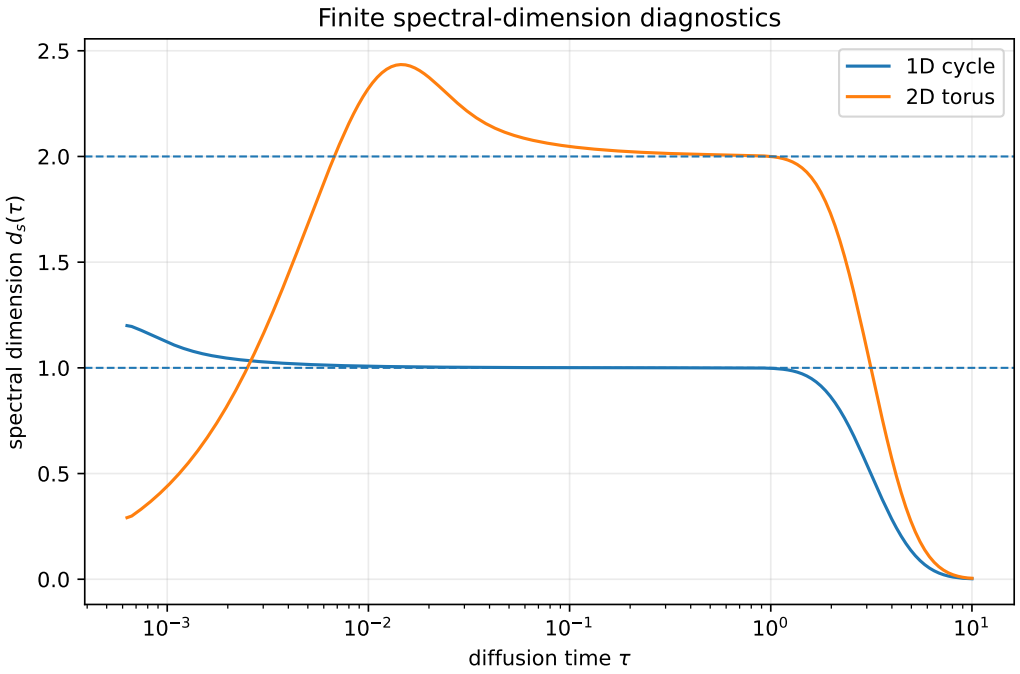


Figure 21. The curves show intermediate windows of spectral dimension for a finite cycle and a finite two-dimensional torus. Departures in the ultraviolet and infrared reflect the finite size of the models rather than additional dimensions.

Recorded Numerical Check I.11.3 (Spectral dimension). The median intermediate plateaus were

$$d_s^{(1)} = 1.00062, \qquad d_s^{(2)} = 2.03781.$$

These tests recover the expected dimensions of the chosen finite geometries. They do not select a physical theory of quantum spacetime.

I.12 Constraint packets and the quantum-spacetime boundary

At one finite level, let $\widehat{C}_A$ be a finite family of matrix constraints, with fixed scalar structure coefficients, and let Π^{phys} be the orthogonal projector onto their exact common kernel. If the classical constraint algebra has structure coefficients $f_{AB}^{\ C}$, define the anomaly operator

$$\boxed{\mathfrak{A}_{AB} = [\widehat{C}_A, \widehat{C}_B] - i \sum_C f_{AB}^{\ C} \widehat{C}_C.}$$

Its physical packet is

$$\Pi^{\text{phys}} \mathfrak{A}_{AB} \Pi^{\text{phys}}.$$

In this exact finite setting, $\widehat{C}_C \Pi^{\text{phys}} = 0$ for every C. Associativity gives

$$\mathfrak{A}_{AB} \Pi^{\text{phys}} = 0, \qquad \Pi^{\text{phys}} \mathfrak{A}_{AB} \Pi^{\text{phys}} = 0$$

automatically. This compression therefore cannot certify operator closure away from the physical range. A nonzero measured compression contradicts the exact-kernel assumptions or indicates an approximation or implementation error. For example, $C_1 = \operatorname{diag}(1, 0, 0)$ and $C_2 = E_{12} + E_{21}$ annihilate the same third coordinate, while their commutator $E_{12} - E_{21}$ is nonzero. With zero proposed structure coefficients, the anomaly is nonzero but its physical compression vanishes.

The closure target must be stated at the operator level, not inferred from that automatic compression. Unbounded constraints additionally require a specified common invariant domain. Neither such a domain theory nor gravitational structure functions are supplied by the finite check.

Conjecture I.12.1 (Anomaly-free refinement)**.** For a specified nontrivial quantum-geometric constraint model, there exists a coherent packet tower with nonzero physical spaces, compatible physical projectors and exact operator closure

$$\mathfrak{A}_{AB}^{(\ell)} = 0$$

at every finite level. The compatible physical compression then vanishes as a consequence, not as a substitute for closure.

The substantive research target is the operator condition. The earlier condition stated only in terms of compression was too weak. A choice of model, dynamics, and refinement maps is still needed before this program becomes a fully specified physical existence problem.

Conjecture I.12.2 (Uniform Feshbach control)**.** For the same specified tower, the self-energies from exact elimination of fine packets admit bounds uniform in cutoff and volume on a common spectral domain. Together with consistent identifications of the Hilbert spaces and a proved resolvent convergence criterion, they yield a nontrivial continuum limit in the strong resolvent sense. Uniform boundedness alone is not asserted to imply convergence.

Conjecture I.12.3 (Lorentzian reconstruction)**.** The Euclidean packet theory satisfies a positivity or reconstruction principle strong enough to produce a Lorentzian Hilbert space with causal propagation and positive physical norm.

Conjecture I.12.4 (Einstein infrared limit)**.** There exist semiclassical packet states $\Psi_{g,\ell}$ such that the effective action converges to

$$\frac{1}{16\pi G}\int \sqrt{-g}\,(R-2\Lambda)\,d^4x$$

with controlled higher-order corrections.

Conjecture I.12.5 (Massless spin-two sector)**.** The continuum two-point function possesses a positive transverse-traceless massless spin-two pole, while all unphysical packet sectors are removed by the constraints.

The exact finite identities establish none of these physical conclusions, and no formal theorem of logical independence is claimed. Exact packet reconstruction provides a kinematic and computational advantage. It does not determine the physical action, prove existence of the continuum theory, or guarantee the limit of general relativity.

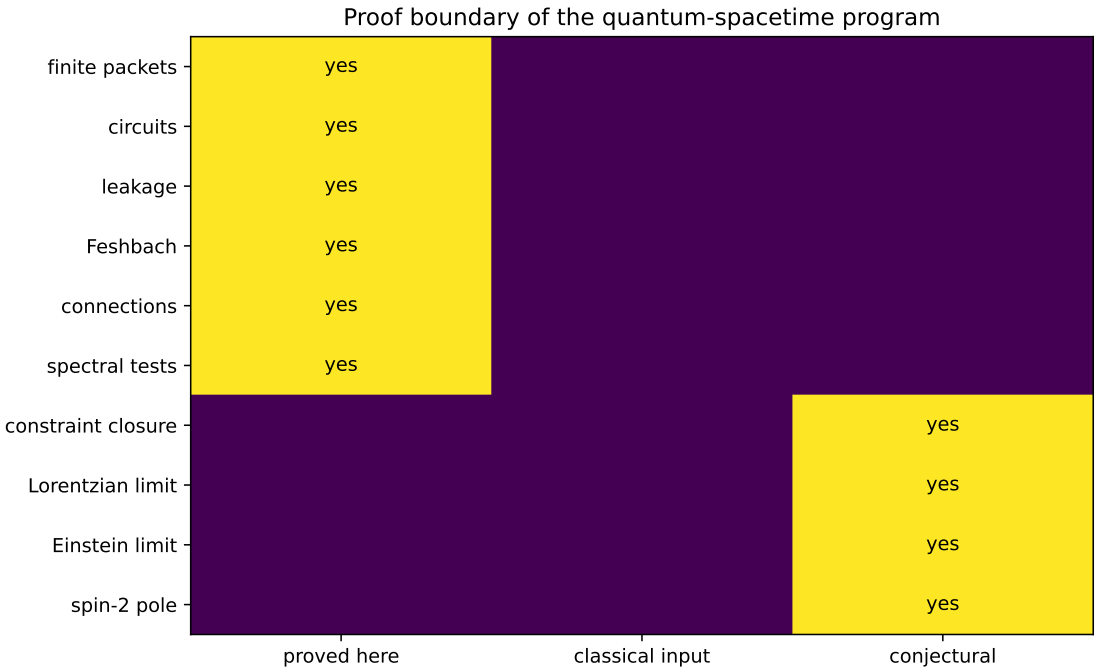


Figure 22. The proof boundary. Finite packet, circuit, leakage, Feshbach, connection, and spectral identities are established here. Constraint closure and the physical continuum claims remain conjectural.

I.13 Numerical certificates

The numerical study reports CPU experiments and a suite of 22 tests. The values below come from the accompanying numerical record. They are separate from the 323 supporting declarations in the formal companion. A displayed floating-point residual of zero describes the reported computation and does not independently prove equality in exact arithmetic.

Table 1. Recorded finite numerical checks.

Certificate	Value
Bell-state residual	0
Three-qubit seed-orbit rank	$8/8$
Parity off-block norm	0
Circuit path reconstruction residual	0
Exact-symmetry maximum leakage	0
Maximum leakage-bound violation	0
Feshbach projected-resolvent residual	3.2291×10^{-17}
Bare-truncation projected-resolvent residual	1.3091×10^{-2}
Moving-frame residual with connection	1.0645×10^{-10}
Moving-frame residual without connection	7.0000×10^{-1}
Canonical packet-connection residual	0
Bloch-curvature residual	2.0803×10^{-11}
Kraus completeness residual	0
Output trace residual	0
Knill–Laflamme residual	0
Bit-flip recovery residual	1.0385×10^{-16}
Tensor parity-refinement residual	0
Circle low-mode convergence order	1.99936
One-dimensional spectral plateau	1.00062
Two-dimensional spectral plateau	2.03781
Minimum Euclidean-transfer eigenvalue	0.0767054
Transfer semigroup residual	0
Volume-packet reconstruction residual	0

The recorded test count is

$$\boxed{22/22 \text{ recorded numerical tests presented here.}}$$

The 19 original figures are retained from the recovered numerical companion. The separate visual guide supplies a reading route through these constructions and the later strengthened results.

I.15 Conclusion

The quantum realization of the seed calculus begins with a primitive state and a declared operator grammar. Symmetry projectors resolve the state and every operator into exact packets. In these coordinates, circuit products become sums over packet paths, and Hamiltonian symmetry appears as a diagonal block structure. When that symmetry is broken, the resulting transfer is measured by the leakage estimates.

Moving packets carry a canonical connection and curvature. Channels and measurements act on the corresponding operator packets, while the conditions for error correction become statements about compression to the code. Exact elimination retains the influence of hidden sectors through self-energy and memory.

The finite theory therefore provides a coherent language for

$$\boxed{\begin{array}{c}\text{seed reachability} + \text{symmetry resolution} + \text{circuit paths} \\ +\text{effective dynamics} + \text{geometric transport} + \text{constraint diagnostics}\end{array}}$$

A physical continuum theorem remains open. The next required results are anomaly-free refinement, uniform control of effective operators, Lorentzian reconstruction, the Einstein infrared limit, and a positive massless spin-two sector. Each requires its own hypotheses and proof. Agreement in a finite numerical model cannot replace those arguments.

Part II Strengthened quantum packet calculus and continuum towers

A route through this section

This part broadens the packet calculus while keeping its hypotheses explicit. Read each result with its assumptions about nonzero spaces, a common ambient space, self-adjointness, finite address sets, and trace class operators. Its bounded limits do not establish the five open physical continuum targets.

II.1 Introduction

Quantum mechanics repeatedly uses the same operation. A Hilbert space is split into sectors, an operator is resolved between those sectors, and the pieces are propagated or recombined. Symmetry sectors, eigenspaces, syndrome spaces, adiabatic subspaces, and retained model spaces are familiar examples. This article treats them as instances of one finite packet calculus.

The elementary coordinate object is a complete family of orthogonal projectors

$$P_a P_b = \delta_{ab} P_a, \qquad \sum_{a \in I} P_a = I.$$

Every vector has packet coordinates $P_a \psi$. Every operator has transfer blocks

$$A_{ba} = P_b A P_a.$$

These identities are simple. Their value appears when the same coordinates are used across representation theory, circuits, channels, error correction, effective dynamics, and geometric limits.

Quantum information and operator theory in finite dimensions provide the standard background [18, 27, 3]. Finite group projectors come from character theory [23, 9]. Moving spectral subspaces and geometric phase originate in the work of Kato, Berry, Wilczek, and Zee [12, 13, 2, 28]. Completely positive maps and quantum dynamical semigroups are treated in [24, 5, 15, 16]. Exact error correction is governed by the Knill–Laflamme condition [14, 22, 1]. Projection methods and hidden state memory are linked by the Feshbach, Nakajima, and Zwanzig constructions [8, 17, 29, 4].

The first part begins with quantum seeds. A state is cyclic only relative to a declared observable algebra. For an algebra in multiplicity form, the local orbit dimension is controlled by the rank of a coefficient matrix. This recovers the relation between Schmidt rank and the dimension reached by local operations. The finite group section then constructs central isotypic projectors. Cyclic symmetry gives a sharper charge selection rule. An operator with charge q can move a state only from packet j to packet $j + q$.

A circuit is resolved by inserting the identity between successive gates. This produces a sum over packet paths. The target probability contains diagonal path terms and interference between distinct

paths. Symmetry can remove entire path families before any matrix multiplication is performed. Unitarity gives exact row and column relations among the transfer blocks.

The moving packet section treats time dependent projectors. The canonical generator

$$K(t) = \sum_a \dot{P}_a(t) P_a(t)$$

is skew adjoint and transports every packet. In the moving frame, the physical Hamiltonian acquires the geometric term $-\mathrm{i}W(t)^*\dot{W}(t)$. The leakage estimate must account for the motion of the packet itself. The physical coupling QHP is therefore replaced by

$$Q(t)(H(t) - \mathrm{i}K(t))P(t).$$

This is the exact finite distinction between following a moving subspace and remaining in a fixed one.

Density matrices require separate packet labels for their output and input coordinates. The superoperators

$$\mathcal{P}_{ab}(X) = P_a X P_b$$

form a complete orthogonal packet system on Hilbert–Schmidt operator space. Dephasing is the diagonal projection. Every channel has input and output superblocks, with the same composition law as ordinary matrix blocks. A covariant channel preserves the isotypic packets of the conjugation action. In the cyclic case these are coherence orders. The Choi matrix carries the equivalent symmetry under $U \otimes \overline{U}$.

Error correction is expressed through compression to a code. Let P be the code projector. The class of PAP modulo scalar multiples of P measures the obstruction to making A undetectable within the code. The Knill–Laflamme criterion says that this obstruction vanishes for every product $E_a^* E_b$ from the error family. Diagonalizing the resulting positive coefficient matrix produces orthogonal syndrome packets and an explicit recovery on the error span.

The effective dynamics section returns to ordinary Hilbert space packets. Schur complementation gives the exact retained resolvent and a frequency dependent self energy. The same hidden block gives a causal memory kernel in time. For a self adjoint hidden Hamiltonian, the self energy has a fixed sign in the upper half plane, and the time kernel is positive definite. Uniform spectral separation gives direct bounds that can be carried through a refinement tower.

The geometric part develops two finite constructions. A strictly positive contraction gives a reflection positive Hankel kernel, and its logarithm reconstructs a nonnegative Hamiltonian. A finite Laplacian gives a heat trace from which the spectral dimension is defined at each diffusion scale. The article proves convergence from periodic lattices to the continuum torus at every fixed positive diffusion time. Poisson summation then determines the continuum dimension as diffusion time tends to zero.

Constraint closure needs a different obstruction. If C_a are self adjoint constraints and P_{phys} projects onto their common kernel, then

$$P_{\mathrm{phys}}[C_a, C_b]P_{\mathrm{phys}} = 0$$

holds automatically. This compression cannot test closure. The correct finite obstruction is the class of $[C_a, C_b]$ in the quotient of $\mathrm{End}(\mathcal{H})$ by the span of the constraints. Vanishing means that the constraint span is closed under commutators.

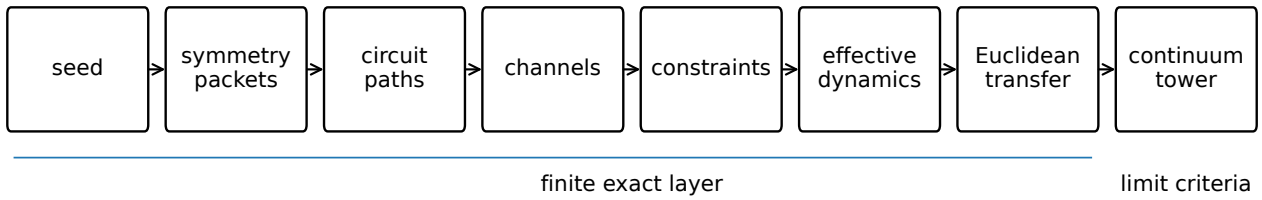


Figure 23. The finite mathematical layers developed in the article and the separate limit criteria needed at the continuum boundary.

Object	Packet coordinate	Exact structural condition
State	$P_a\psi$	completeness of the projector family
Operator	P_bAP_a	block multiplication
Circuit	intermediate packet path	insertion of complete resolutions
Density matrix	$P_a\rho P_b$	two sided operator packets
Channel	superoperator block	complete positivity and covariance when declared
Code	syndrome packet	scalar compression of error products
Effective model	Feshbach map	invertibility of the hidden resolvent
Euclidean model	positive transfer matrix	positivity and self adjointness
Constraint system	quotient obstruction	commutator closure modulo the constraint span
Continuum tower	direct or norm limit	compatibility plus explicit convergence

Table 2. The packet coordinates used in the article.

The last part gives two theorems for towers of bounded Hamiltonians. Exactly compatible finite Hamiltonians with a uniform norm bound define a bounded self adjoint operator on the Hilbert direct limit. When successive bounded self adjoint Hamiltonians on a common Hilbert space have norm summable differences, they converge to a bounded self adjoint limit. Their unitary evolutions, and their Euclidean evolutions under the stated positivity assumption, converge uniformly on bounded time intervals. Reflection positivity and closure with fixed structure coefficients pass to the limit under the stated convergence assumptions.

These results do not construct an interacting continuum quantum field theory. They do not prove anomaly freedom on an unbounded common domain, Osterwalder–Schrader reconstruction for a continuum measure, an Einstein infrared action, or a positive massless spin two sector. Each of those conclusions needs hypotheses absent from the finite calculations.

II.2 Quantum seeds and observable algebras

Definition II.2.1 (Finite quantum seed system)**.** A finite quantum seed system is a triple

$$(\mathcal{H}, \mathcal{A}, S),$$

where $\mathcal{H}$ is a finite dimensional Hilbert space, $\mathcal{A} \subseteq \mathrm{End}(\mathcal{H})$ is a unital star subalgebra, and $S \subseteq \mathcal{H}$ is a finite set. Its generated hull is

$$\mathcal{A}S = \mathrm{span}\{A\psi : A \in \mathcal{A},\ \psi \in S\}.$$

The seed set is cyclic when $\mathcal{A}S = \mathcal{H}$. The quantum seed rank is the least cardinality of a cyclic seed set.

The algebra is part of the data. A vector may be cyclic for $\mathrm{End}(\mathcal{H})$ and fail to be cyclic for a restricted observable or gate algebra.

Let

$$\mathcal{A} \cong \bigoplus_{r=1}^{s} (M_{d_r}(\mathbb{C}) \otimes I_{m_r})$$

act on

$$\mathcal{H} \cong \bigoplus_{r=1}^{s} (\mathbb{C}^{d_r} \otimes \mathbb{C}^{m_r}).$$

Assume $d_r, m_r \geq 1$ for each displayed block. For an empty block family, the seed rank and the displayed finite supremum are defined as zero. A vector in one block is identified with a d_r by m_r coefficient matrix.

Theorem II.2.2 (Orbit dimension of one seed)**.** *For integers $d \geq 1$, $m \geq 0$, and the action of $M_d(\mathbb{C}) \otimes I_m$ on $\mathbb{C}^d \otimes \mathbb{C}^m$, let C_ψ be the coefficient matrix of ψ. Then*

$$\dim \mathrm{span}\{(A \otimes I_m)\psi : A \in M_d(\mathbb{C})\} = d \,\mathrm{rank}\, C_\psi.$$

The vector is cyclic exactly when $\mathrm{rank}\, C_\psi = m$.

Proof. Under the coefficient matrix identification, $(A \otimes I_m)\psi$ corresponds to AC_ψ. Every row of AC_ψ lies in the row space of C_ψ. Conversely, a matrix whose rows lie in that row space equals AC_ψ for a suitable A. There are d independent row choices in a space of dimension $\mathrm{rank}\, C_\psi$. The orbit dimension is therefore $d\,\mathrm{rank}\, C_\psi$. It equals dm exactly when C_ψ has full column rank. □

Theorem II.2.3 (Quantum seed rank)**.** *For the displayed multiplicity decomposition,*

$$\mathrm{srank}_{\mathcal{A}}(\mathcal{H}) = \max_{1 \leq r \leq s} \left\lceil \frac{m_r}{d_r} \right\rceil.$$

Proof. A single seed contributes a row space of dimension at most d_r to the rth multiplicity factor. Hence k seeds cover at most kd_r multiplicity directions, which gives the lower bound.

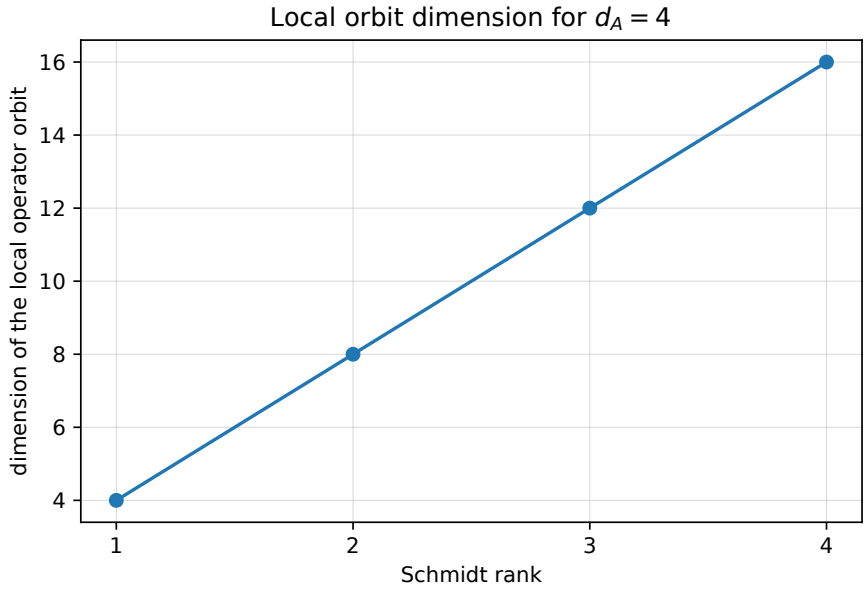


Figure 24. The dimension reached by all local operators is linear in Schmidt rank.

Set

$$k = \max_r \left\lceil \frac{m_r}{d_r} \right\rceil .$$

For each r, partition a basis of $\mathbb{C}^{m_r}$ into at most k groups of size at most d_r. Choose the coefficient matrix of the jth seed so that its row space spans the jth group. The combined row spaces span every multiplicity factor. The preceding theorem then shows that the k seeds generate every summand. □

Corollary II.2.4 (Local orbit and Schmidt rank)**.** *Let $\psi \in \mathcal{H}_A \otimes \mathcal{H}_B$, let $d_A = \dim \mathcal{H}_A \geq 1$, and let r be the Schmidt rank of ψ. Then*

$$\dim \operatorname{span}\{(A \otimes I_B)\psi : A \in \operatorname{End}(\mathcal{H}_A)\} = d_A r.$$

Local operators on A generate the whole bipartite Hilbert space exactly when the reduced state on B has full support.

Proof. The Schmidt rank is the rank of the coefficient matrix in any product basis. Substituting that rank into theorem II.2.2 gives the stated orbit dimension. □

Proposition II.2.5 (Computational address orbit)**.** *For n qubits, let*

$$\psi_0 = |0\rangle^{\otimes n}$$

and let X_j be the Pauli X operator on the jth qubit. Then

$$\{X_1^{a_1} \cdots X_n^{a_n} \psi_0 : (a_1, \ldots, a_n) \in \{0,1\}^n\}$$

is the computational basis.

Proof. The addressed vector is $|a_1 \cdots a_n\rangle$. Distinct binary addresses produce the 2^n distinct basis vectors. □

Remark II.2.6. The seed rank records generation by a declared algebra. It does not measure circuit depth, locality cost, control complexity, or physical reachability under a restricted time dependent Hamiltonian.

II.3 Symmetry packets and selection rules

Let G be a finite group and let

$$U : G \longrightarrow \mathcal{U}(\mathcal{H})$$

be a unitary representation. Write $\widehat{G}$ for the set of irreducible unitary representations. For $\lambda \in \widehat{G}$, let d_λ and χ_λ denote its dimension and character.

Theorem II.3.1 (Finite group isotypic projectors)**.** *The operators*

$$P_\lambda = \frac{d_\lambda}{|G|} \sum_{g \in G} \overline{\chi_\lambda(g)} U(g)$$

are orthogonal projectors. They satisfy

$$P_\lambda P_\mu = \delta_{\lambda\mu} P_\lambda, \qquad \sum_{\lambda \in \widehat{G}} P_\lambda = I_{\mathcal{H}}.$$

The range of P_λ is the λ isotypic component of $\mathcal{H}$.

Proof. Complete reducibility gives a unitary decomposition

$$\mathcal{H} \cong \bigoplus_{\mu \in \widehat{G}} V_\mu \otimes M_\mu, \qquad U(g) \cong \bigoplus_{\mu} \pi_\mu(g) \otimes I_{M_\mu}.$$

On the μ summand, the displayed average is central in the irreducible representation π_μ, so it is a scalar by Schur's lemma. Taking its trace and applying character orthogonality shows that the scalar is one when $\mu = \lambda$ and zero otherwise. The asserted identities follow on the direct sum. □

The central projectors separate irreducible types but do not choose coordinates inside the representation factor. Matrix units provide that finer resolution.

Theorem II.3.2 (Representation matrix units). *Fix an irreducible unitary representation $\pi_\lambda : G \to U(d_\lambda)$ and an orthonormal basis of its representation space. Define*

$$E_{ij}^{\lambda} = \frac{d_\lambda}{|G|} \sum_{g \in G} \big[\pi_\lambda(g^{-1})\big]_{ji} U(g).$$

Then

$$E_{ij}^{\lambda} E_{k\ell}^{\mu} = \delta_{\lambda\mu}\delta_{jk} E_{i\ell}^{\lambda}, \qquad \big(E_{ij}^{\lambda}\big)^{*} = E_{ji}^{\lambda},$$

and

$$P_\lambda = \sum_{i=1}^{d_\lambda} E_{ii}^{\lambda}.$$

Proof. Apply the matrix coefficient form of Schur orthogonality to the two group averages. It gives the matrix unit multiplication law. Unitarity gives the adjoint relation after replacing g by g^{-1}. Summing the diagonal coefficients gives the character average in theorem II.3.1. □

II.3.1 Cyclic charge packets

Let $N \geq 1$ be an integer, S unitary with $S^N = I$, and $\omega = \mathrm{e}^{2\pi \mathrm{i}/N}$.

Corollary II.3.3 (Cyclic charge projectors). *The operators*

$$P_j = \frac{1}{N} \sum_{r=0}^{N-1} \omega^{-jr} S^r, \qquad j \in \mathbb{Z}/N\mathbb{Z},$$

form a complete orthogonal packet system and satisfy

$$S P_j = \omega^j P_j.$$

Proof. Apply theorem II.3.1 to the cyclic group. Its irreducible representations are the one dimensional characters $r \mapsto \omega^{jr}$. □

Definition II.3.4 (Operator charge). An operator $A \in \mathrm{End}(\mathcal{H})$ has cyclic charge $q \in \mathbb{Z}/N\mathbb{Z}$ when

$$S A S^{*} = \omega^q A.$$

Theorem II.3.5 (Charge selection rule). *If A has charge q, then*

$$P_k A P_j = 0$$

unless $k = j + q$ in $\mathbb{Z}/N\mathbb{Z}$. In fact,

$$A P_j = P_{j+q} A P_j.$$

Proof. For $x \in P_j\mathcal{H}$,

$$S(Ax) = (SAS^{*})Sx = \omega^{q+j} Ax.$$

Thus $Ax \in P_{j+q}\mathcal{H}$. The block statement follows. □

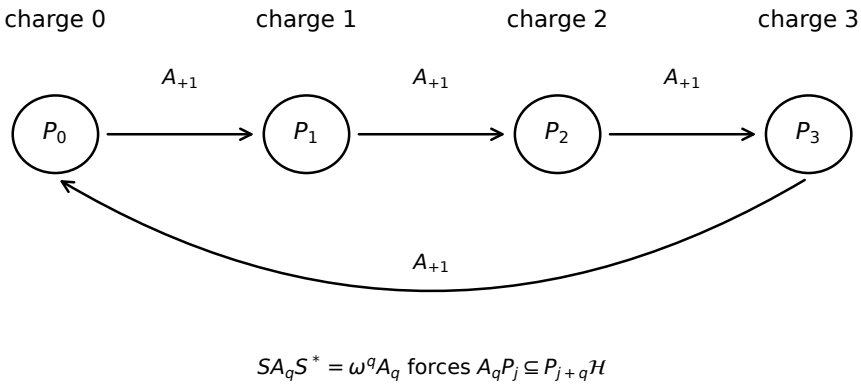


Figure 25. An operator of charge one moves each cyclic packet to the next packet and has no other transfer blocks.

Corollary II.3.6 (Charge addition)**.** *If A and B have charges q and r, then AB has charge $q + r$. The adjoint A^* has charge $-q$.*

Proof. Conjugation by S is a star automorphism. Applying it to AB adds the two charges, while applying it to A^* changes the sign of the charge. □

II.3.2 Tensor refinement

Suppose the ath tensor factor carries a cyclic packet family

$$\left(P_j^{(a)}\right)_{j\in\mathbb{Z}/N\mathbb{Z}}.$$

The fine tensor packets are

$$P_{\boldsymbol{j}}^{\text{fine}} = P_{j_1}^{(1)} \otimes \cdots \otimes P_{j_m}^{(m)}.$$

Theorem II.3.7 (Tensor charge refinement)**.** *For $r \in \mathbb{Z}/N\mathbb{Z}$, define*

$$P_r^{\text{total}} = \sum_{j_1+\cdots+j_m=r} P_{j_1}^{(1)} \otimes \cdots \otimes P_{j_m}^{(m)}.$$

Then the P_r^{total} form a complete orthogonal packet system. Every total charge packet is the exact sum of its fine tensor descendants.

Proof. The fine tensor projectors are mutually orthogonal and sum to the identity. The total residue classes partition the fine address set. Summing the fine projectors over each class gives the result. □

Corollary II.3.8 (Tensor selection rule). *If A_a has local charge q_a, then*

$$A_1 \otimes \cdots \otimes A_m$$

has total charge $q_1 + \cdots + q_m$.

Proof. Conjugate the tensor product by the total cyclic symmetry and multiply the local phase factors. □

Remark II.3.9. For a nonabelian group, tensor products are refined by irreducible fusion multiplicities rather than residue addition. The central isotypic projectors remain exact, but a complete tensor refinement also requires a choice of Clebsch–Gordan intertwiners.

II.4 Quantum circuits as packet paths

Let

$$U = U_m U_{m-1} \cdots U_1$$

be a finite circuit with $m \geq 1$, and let $(P_a)_{a \in I}$ be a complete orthogonal packet family. For the empty circuit, $U = I$ and its transfer block is $P_b P_a = \delta_{ba} P_a$, whose endpoint sum reconstructs I.

Theorem II.4.1 (Exact circuit path expansion). *For source packet a_0 and target packet a_m,*

$$P_{a_m} U P_{a_0} = \sum_{a_1, \ldots, a_{m-1}} P_{a_m} U_m P_{a_{m-1}} \cdots P_{a_1} U_1 P_{a_0}.$$

Summing over a_0 and a_m reconstructs U.

Proof. Insert $I = \sum_a P_a$ between every adjacent pair of gates and distribute the finite sums. Completeness at the two endpoints gives the full operator. □

For a packet path

$$\gamma = (a_0, a_1, \ldots, a_m),$$

write

$$U_\gamma = P_{a_m} U_m P_{a_{m-1}} \cdots P_{a_1} U_1 P_{a_0}.$$

Theorem II.4.2 (Packet unitarity identities). *If U is unitary and $U_{ba} = P_b U P_a$, then*

$$\sum_b U_{ba}^* U_{bc} = \delta_{ac} P_a$$

and

$$\sum_a U_{ba} U_{da}^* = \delta_{bd} P_b.$$

Proof. Use completeness and unitarity:

$$\sum_b U_{ba}^* U_{bc} = P_a U^* \left(\sum_b P_b \right) U P_c = P_a P_c.$$

Summing instead over the source packets and using unitarity in the reverse order gives the second identity. □

Corollary II.4.3 (Probability conservation from one packet)**.** *If $\psi \in P_a\mathcal{H}$, then*

$$\sum_b \|P_bU\psi\|^2 = \|\psi\|^2 .$$

Proof. Take the inner product of the first identity in theorem II.4.2 with ψ and set $c = a$. □

Theorem II.4.4 (Path interference formula)**.** *Fix a source vector $\psi \in P_{a_0}\mathcal{H}$ and a target packet b. Let Γ_b be the set of packet paths from a_0 to b, and set*

$$\eta_\gamma = U_\gamma\psi.$$

Choose any total order on this finite path set. Then

$$\|P_bU\psi\|^2 = \sum_{\gamma\in\Gamma_b} \|\eta_\gamma\|^2 + 2\,\mathrm{Re} \sum_{\substack{\gamma,\gamma'\in\Gamma_b \\ \gamma<\gamma'}} \langle \eta_\gamma, \eta_{\gamma'} \rangle .$$

Consequently,

$$\|P_bU\psi\| \le \sum_{\gamma\in\Gamma_b} \|\eta_\gamma\| .$$

If the path outputs are pairwise orthogonal, the target probability is the sum of the path probabilities.

Proof. The path expansion gives

$$P_bU\psi = \sum_{\gamma\in\Gamma_b} \eta_\gamma.$$

Expanding the squared norm and grouping conjugate cross terms gives the interference formula. The triangle inequality gives the norm bound. When the path outputs are orthogonal, every cross term vanishes and only the individual path probabilities remain. □

Proposition II.4.5 (Charge pruning of circuit paths)**.** *Suppose every gate has a finite charge decomposition*

$$U_r = \sum_q U_r^{(q)}, \qquad SU_r^{(q)}S^* = \omega^q U_r^{(q)}.$$

A term with gate charges $q_1, \dots, q_m$ can contribute from source charge a_0 only to target charge

$$a_m = a_0 + q_1 + \cdots + q_m.$$

Every packet path violating one of the intermediate charge updates vanishes.

Proof. Apply theorem II.3.5 after each gate component. The packet address is forced at every step. □

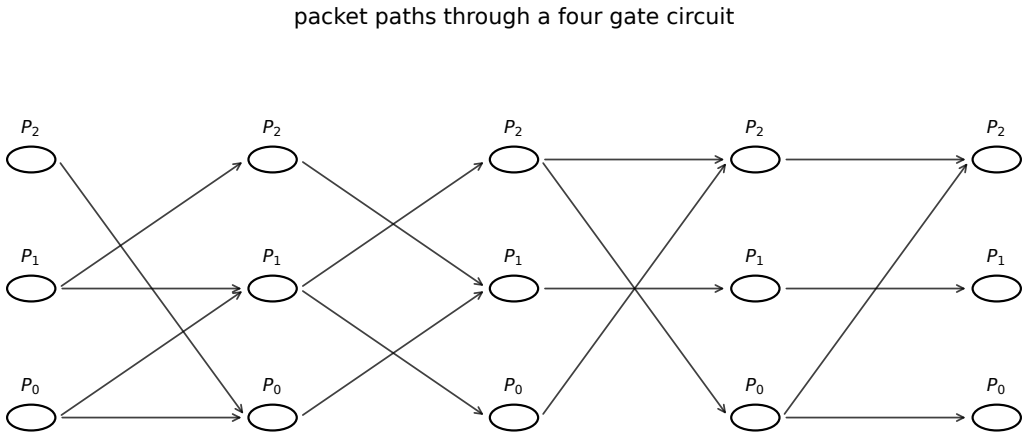


Figure 26. A circuit block is the coherent sum of its allowed packet paths. Selection rules remove forbidden paths before amplitudes are evaluated.

Example II.4.6 (Bell circuit)**.** Starting from $|00\rangle$, apply a Hadamard gate to the first qubit and then a controlled NOT gate. The final state is

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

It lies in the even parity packet. The intermediate state has one even and one odd computational contribution, so the final even amplitude is reconstructed by more than one packet path.

II.5 Symmetry resolved dynamics and leakage

Let $H = H^*$ and let $(P_a)_{a\in I}$ be a complete orthogonal packet system.

Theorem II.5.1 (Exact invariant packets)**.** *The following conditions are equivalent:*

1. *The Hamiltonian commutes with each projector:* $[H, P_a] = 0$ *for every* a*.*
2. *Transfer blocks between distinct packets vanish:* $P_b H P_a = 0$ *whenever* $a \neq b$*.*
3. *Every packet* $P_a\mathcal{H}$ *is invariant under* $\mathrm{e}^{-\mathrm{i}tH}$ *for every real* t*.*

When they hold,

$$H = \sum_a P_a H P_a$$

and

$$\mathrm{e}^{-\mathrm{i}tH} = \sum_a P_a \mathrm{e}^{-\mathrm{i}tH} P_a.$$

Proof. Commutation is equivalent to vanishing of every off diagonal block. The same block decomposition passes to polynomials in H, then to the exponential by norm convergence. Invariance of every packet implies invariance under both $\mathrm{e}^{-\mathrm{i}tH}$ and its adjoint. Differentiation at $t = 0$ gives commutation with H. □

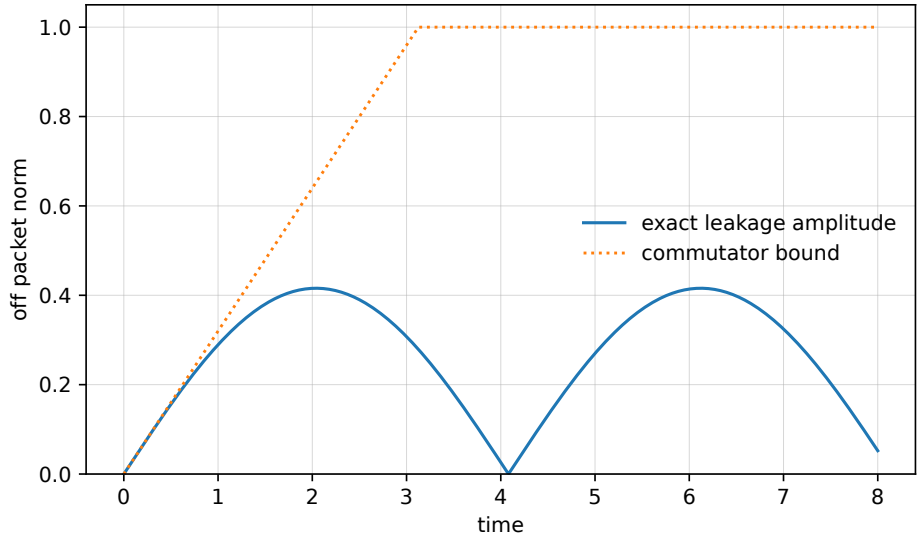


Figure 27. Exact transition amplitude in a two packet model and the linear commutator bound.

Fix one packet projector P and set $Q = I - P$.

Lemma II.5.2 (Off diagonal norm)**.** *For self adjoint H,*

$$\|[H, P]\| = \|QHP\| = \|PHQ\|\,.$$

Proof. Relative to $P\mathcal{H} \oplus Q\mathcal{H}$,

$$[H, P] = \begin{pmatrix} 0 & -PHQ \\ QHP & 0 \end{pmatrix}.$$

Its square has diagonal blocks $-PHQHP$ and $-QHPHQ$. The norm is therefore the largest singular value of QHP. □

Theorem II.5.3 (Fixed packet leakage)**.** *For every real t,*

$$\left\|Q\mathrm{e}^{-\mathrm{i}tH}P\right\| \le |t|\,\|QHP\| = |t|\,\|[H, P]\|\,.$$

Proof. Let

$$H_{\mathrm{d}} = PHP + QHQ, \qquad H_{\mathrm{o}} = PHQ + QHP.$$

The propagator generated by H_{d} preserves $P\mathcal{H}$ and $Q\mathcal{H}$. Duhamel's formula relative to H_{d} gives

$$Q\mathrm{e}^{-\mathrm{i}tH}P = -\mathrm{i}\int_0^t Q\mathrm{e}^{-\mathrm{i}(t-s)H}H_{\mathrm{o}}\mathrm{e}^{-\mathrm{i}sH_{\mathrm{d}}}P\,\mathrm{d}s.$$

Since $H_{\mathrm{o}}P = QHP$ and both propagators are unitary, the norm is at most $|t|\,\|QHP\|$. Use lemma II.5.2 for the last equality. □

Corollary II.5.4 (Leakage probability). *If $\psi \in P\mathcal{H}$ is a unit vector, then*

$$\left\|Q\mathrm{e}^{-\mathrm{i}tH}\psi\right\|^2 \leq t^2 \left\|QHP\right\|^2 .$$

Proof. Apply theorem II.5.3 to the unit vector, then square the resulting nonnegative bound. □

II.6 Moving packets and geometric transport

Let $(P_a(t))_{a\in I}$ be a twice continuously differentiable family of complete orthogonal packet systems on a finite dimensional Hilbert space.

Theorem II.6.1 (Canonical packet connection). *Define*

$$K(t) = \sum_a \dot{P}_a(t) P_a(t).$$

Then

$$K(t)^* = -K(t)$$

and

$$[K(t), P_a(t)] = \dot{P}_a(t)$$

for every a.

Proof. Differentiating $P_a^2 = P_a$ gives

$$\dot{P}_a P_a + P_a \dot{P}_a = \dot{P}_a.$$

Summing over a and using $\sum_a \dot{P}_a = 0$ yields

$$\sum_a P_a \dot{P}_a = -\sum_a \dot{P}_a P_a.$$

This proves skew adjointness. For fixed b,

$$K P_b = \dot{P}_b P_b.$$

Differentiating $P_b P_a = 0$ for $a \neq b$ gives

$$P_b \dot{P}_a = -\dot{P}_b P_a.$$

Hence

$$P_b K = -\dot{P}_b (I - P_b).$$

Subtracting gives $[K, P_b] = \dot{P}_b$. □

Theorem II.6.2 (Exact Kato transport). *Let $W(t)$ solve*

$$\dot{W}(t) = K(t)W(t), \qquad W(0) = I.$$

Then $W(t)$ is unitary and

$$P_a(t)W(t) = W(t)P_a(0)$$

for every packet a.

Proof. Skew adjointness gives

$$\frac{\mathrm{d}}{\mathrm{d}t} W(t)^* W(t) = 0.$$

Thus $W(t)$ is unitary. Next,

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(W(t)^* P_a(t) W(t) \right) = W(t)^* \left(\dot{P}_a - [K, P_a] \right) W(t) = 0.$$

The expression equals $P_a(0)$ at $t = 0$, which proves the intertwining identity. □

Theorem II.6.3 (Moving frame Hamiltonian). *Let $U(t)$ solve*

$$\mathrm{i}\dot{U}(t) = H(t)U(t), \qquad U(0) = I,$$

where $H(t) = H(t)^*$ is continuous on the compact time interval under consideration. Set

$$V(t) = W(t)^* U(t).$$

Then

$$\mathrm{i}\dot{V}(t) = H_{\mathrm{mov}}(t)V(t),$$

where

$$H_{\mathrm{mov}}(t) = W(t)^* H(t) W(t) - \mathrm{i} W(t)^* \dot{W}(t) = W(t)^* (H(t) - \mathrm{i}K(t)) W(t).$$

The moving frame Hamiltonian is self adjoint.

Proof. Differentiate $V = W^* U$. Since $\dot{W} = KW$ and $K^* = -K$,

$$\dot{W}^* = -W^* K.$$

Substitution gives the equation. The first term in H_{mov} is self adjoint. The operator $-\mathrm{i}W^*\dot{W}$ is self adjoint because $W^*\dot{W}$ is skew adjoint. □

Theorem II.6.4 (Moving packet leakage). *Fix a packet $P(t) = P_a(t)$ and put $Q(t) = I - P(t)$. For $0 \le t \le T$,*

$$\|Q(t)U(t)P(0)\| \le \int_0^t \left\| Q(s)(H(s) - \mathrm{i}K(s))P(s) \right\| \mathrm{d}s.$$

In particular, the physical evolution follows the moving packet exactly on $[0, T]$ when the displayed off packet block vanishes for every time in that interval.

Proof. In the Kato frame, $P(0)$ is fixed and

$$Q(t)U(t)P(0) = W(t)Q(0)V(t)P(0).$$

Apply the time dependent version of the Duhamel argument in theorem II.5.3 to $H_{\mathrm{mov}}(t)$. The off diagonal block is

$$Q(0)H_{\mathrm{mov}}(s)P(0) = W(s)^*Q(s)(H(s) - \mathrm{i}K(s))P(s)W(s).$$

Unitary conjugation preserves the norm. □

The exact coupling test therefore includes the geometric term. Even if the Hamiltonian is diagonal in the packet decomposition at every fixed time, its evolution can fail to follow a packet whose range rotates.

II.6.1 Discrete packet transport

Let (P_a) and (Q_a) be two complete packet systems with the same index set. Define

$$S(\mathcal{Q}, \mathcal{P}) = \sum_a Q_a P_a.$$

Theorem II.6.5 (Direct packet rotation). *Assume*

$$\|S(\mathcal{Q}, \mathcal{P}) - I\| < 1.$$

Then $S(\mathcal{Q}, \mathcal{P})$ is invertible. Its unitary polar factor

$$\mathcal{U}(\mathcal{Q}, \mathcal{P}) = S(S^*S)^{-1/2}$$

satisfies

$$Q_a\mathcal{U}(\mathcal{Q}, \mathcal{P}) = \mathcal{U}(\mathcal{Q}, \mathcal{P})P_a$$

for every a.

Proof. The norm condition gives invertibility by the Neumann series. The identity

$$Q_aS = SP_a = Q_aP_a$$

shows that S^*S commutes with every P_a. Its inverse square root also commutes with every P_a. Hence

$$Q_a\mathcal{U} = Q_aS(S^*S)^{-1/2} = SP_a(S^*S)^{-1/2} = \mathcal{U}P_a.$$

The polar factor is unitary because S is invertible. □

Theorem II.6.6 (Convergence of short packet rotations)**.** *Let $(P_a(t))$ be twice continuously differentiable on $[0,T]$. For a partition*

$$0 = t_0 < t_1 < \cdots < t_n = T,$$

let

$$W_\pi = \mathcal{U}(\mathcal{P}(t_n), \mathcal{P}(t_{n-1})) \cdots \mathcal{U}(\mathcal{P}(t_1), \mathcal{P}(t_0)).$$

For sufficiently fine partitions the factors are defined, and

$$\|W_\pi - W(T)\| \le C_T \, |\pi|$$

for a constant depending on the packet path but not on the partition. Here the size of the partition is measured by its mesh, which is the length of its largest subinterval.

Proof. Taylor expansion gives

$$S(\mathcal{P}(t+h), \mathcal{P}(t)) = I + hK(t) + O(h^2)$$

uniformly on $[0,T]$. Since $K(t)^* = -K(t)$,

$$S^*S = I + O(h^2),$$

so the polar factor satisfies

$$\mathcal{U}(\mathcal{P}(t+h), \mathcal{P}(t)) = I + hK(t) + O(h^2).$$

This gives a first order approximation to $\dot{W} = KW$ over a single time step. A telescoping product estimate over the partition gives a total error bounded by a constant times $\sum_j (t_{j+1} - t_j)^2$, which is at most $T\,|\pi|$. □

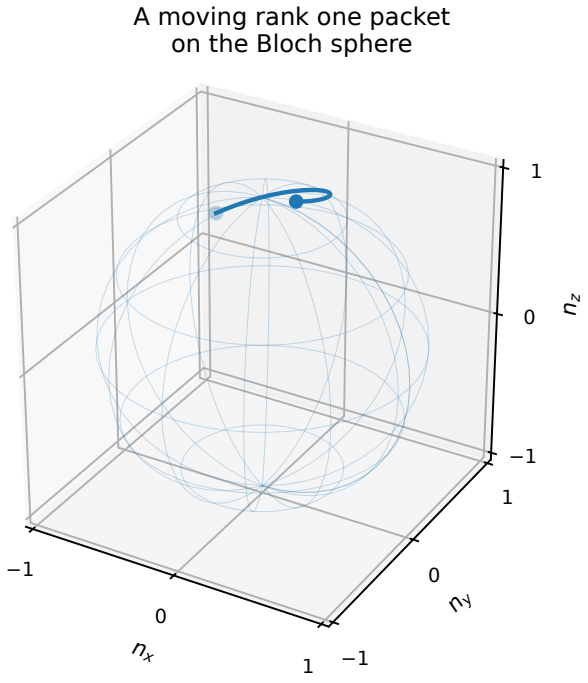


Figure 28. A moving rank one packet on the Bloch sphere. The Kato evolution transports the range without choosing a phase convention for its eigenvector.

II.6.2 Local frames and curvature

Let $P(x)$ be a smooth rank r orthogonal projector over a parameter manifold. Choose a local isometry

$$V(x) : \mathbb{C}^r \longrightarrow \mathcal{H}, \qquad V(x)V(x)^* = P(x).$$

Theorem II.6.7 (Nonabelian packet connection)**.** *The one form with matrix values*

$$\mathcal{A} = V^* \, \mathrm{d}V$$

is skew adjoint. Under a frame change $V' = VG$, where $G(x) \in U(r)$,

$$\mathcal{A}' = G^* \mathcal{A} G + G^* \, \mathrm{d}G.$$

Its curvature

$$\mathcal{F} = \mathrm{d}\mathcal{A} + \mathcal{A} \wedge \mathcal{A}$$

satisfies

$$\mathcal{F} = V^* (\, \mathrm{d}P \wedge \mathrm{d}P) V.$$

Proof. Differentiate $V^*V = I$ to obtain $\mathcal{A}^* = -\mathcal{A}$. The gauge transformation law follows by differentiating VG. Since $P = VV^*$ and $(I - P)\, \mathrm{d}V = \mathrm{d}P\, V$, direct expansion gives

$$\mathrm{d}\mathcal{A} + \mathcal{A} \wedge \mathcal{A} = \mathrm{d}V^* (I - P) \wedge \mathrm{d}V = V^* (\, \mathrm{d}P \wedge \mathrm{d}P) V.$$

□

Corollary II.6.8 (Bloch sphere curvature)**.** *Let*

$$P(\theta,\phi)=\frac{1}{2}\left(I+\boldsymbol{n}(\theta,\phi)\cdot\boldsymbol{\sigma}\right),$$

where

$$\boldsymbol{n}=(\sin\theta\cos\phi,\sin\theta\sin\phi,\cos\theta).$$

With the convention

$$\mathcal{F}_{\theta\phi}=-\mathrm{i}\,\mathrm{Tr}(P[\partial_\theta P,\partial_\phi P]),$$

one has

$$\mathcal{F}_{\theta\phi}=\frac{1}{2}\sin\theta$$

and

$$\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\pi}\mathcal{F}_{\theta\phi}\,\mathrm{d}\theta\,\mathrm{d}\phi=1.$$

Proof. The Pauli identity

$$(\boldsymbol{a}\cdot\boldsymbol{\sigma})(\boldsymbol{b}\cdot\boldsymbol{\sigma})=(\boldsymbol{a}\cdot\boldsymbol{b})I+\mathrm{i}(\boldsymbol{a}\times\boldsymbol{b})\cdot\boldsymbol{\sigma}$$

gives

$$-\mathrm{i}\,\mathrm{Tr}(P[\partial_\theta P,\partial_\phi P])=\frac{1}{2}\boldsymbol{n}\cdot(\partial_\theta\boldsymbol{n}\times\partial_\phi\boldsymbol{n})=\frac{1}{2}\sin\theta.$$

Integrating the displayed curvature over the sphere and applying the stated normalization gives one. □

II.7 Operator packets, measurements, and quantum channels

Density matrices and observables carry both an input packet and an output packet. Equip $\mathrm{End}(\mathcal{H})$ with the Hilbert–Schmidt inner product

$$\langle X,Y\rangle_{\mathrm{HS}}=\mathrm{Tr}(X^*Y).$$

Definition II.7.1 (Two sided operator packet)**.** For a complete orthogonal family (P_a), define

$$\mathcal{P}_{ab}(X)=P_aXP_b.$$

The range

$$\mathcal{B}_{ab}=P_a\,\mathrm{End}(\mathcal{H})P_b$$

is the operator packet with output address a and input address b.

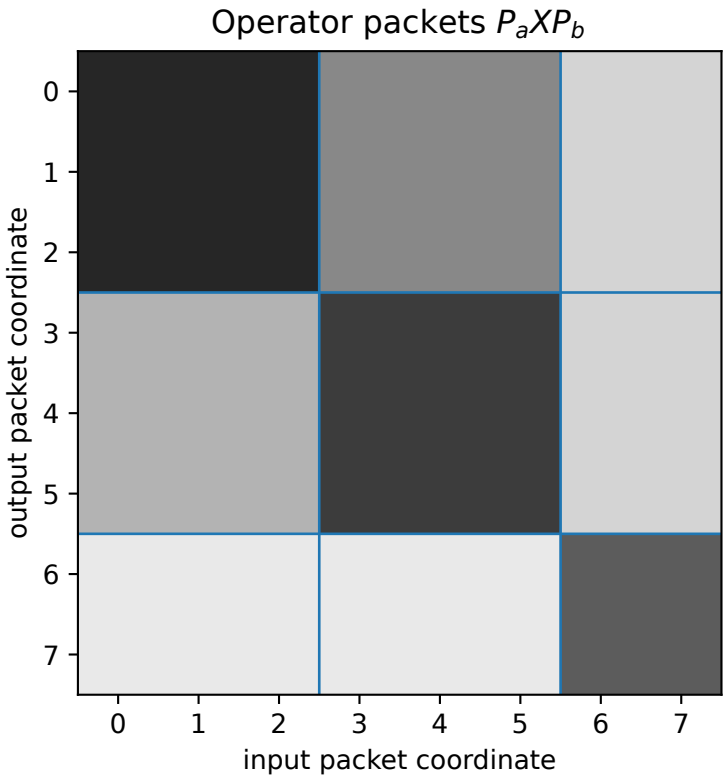


Figure 29. A density matrix or observable carries two packet addresses. Diagonal packets contain populations and internal observables. Off diagonal packets contain coherences.

Theorem II.7.2 (Complete operator packet system)**.** *The superoperators* $(\mathcal{P}_{ab})_{a,b\in I}$ *are mutually orthogonal projections on Hilbert–Schmidt operator space. They satisfy*

$$X = \sum_{a,b} P_a X P_b$$

and

$$\|X\|_{\mathrm{HS}}^2 = \sum_{a,b} \|P_a X P_b\|_{\mathrm{HS}}^2 .$$

Proof. Idempotence and completeness follow from the projector identities. For Hilbert–Schmidt orthogonality,

$$\begin{aligned} \langle P_a X P_b, P_c Y P_d \rangle_{\mathrm{HS}} &= \mathrm{Tr}(P_b X^* P_a P_c Y P_d) \\ &= \delta_{ac}\delta_{bd}\,\mathrm{Tr}(P_b X^* P_a Y P_b). \end{aligned}$$

Since the packets are orthogonal and sum to the original operator, their squared norms add to its squared norm. □

Definition II.7.3 (Packet dephasing)**.** The packet dephasing map is

$$\mathcal{D}_{\mathcal{P}}(X) = \sum_a P_a X P_a.$$

Theorem II.7.4 (Dephasing as conditional expectation)**.** *The map $\mathcal{D}_{\mathcal{P}}$ is completely positive, trace preserving, unital, and idempotent. It is the Hilbert–Schmidt orthogonal projection onto the block diagonal algebra*

$$\mathcal{A}_{\mathcal{P}} = \{X : [X, P_a] = 0 \text{ for every } a\}.$$

In particular,

$$\|\mathcal{D}_{\mathcal{P}}(X)\|_{\mathrm{HS}} \le \|X\|_{\mathrm{HS}}.$$

Proof. The projectors P_a form a Kraus family because $\sum_a P_a^* P_a = I$. Thus the map is completely positive and trace preserving. It is unital for the same reason. Orthogonality and idempotence follow from theorem II.7.2. Its range is the sum of the diagonal operator packets. This is exactly the commutant of the projector family. □

Corollary II.7.5 (Block diagonal measurement invariance)**.** *Let (E_k) be a positive operator valued measurement, and suppose*

$$E_k = \mathcal{D}_{\mathcal{P}}(E_k)$$

for every k. Then every density matrix ρ satisfies

$$\mathrm{Tr}(E_k \rho) = \mathrm{Tr}(E_k \mathcal{D}_{\mathcal{P}}(\rho)).$$

Proof. Since $\mathcal{D}_{\mathcal{P}}$ is self adjoint for the Hilbert–Schmidt inner product,

$$\mathrm{Tr}(E_k \rho) = \mathrm{Tr}(\mathcal{D}_{\mathcal{P}}(E_k)\rho) = \mathrm{Tr}(E_k \mathcal{D}_{\mathcal{P}}(\rho)).$$

□

II.7.1 Channel superblocks

Let $\Phi : \mathrm{End}(\mathcal{H}_X) \to \mathrm{End}(\mathcal{H}_Y)$ be linear. Choose packet systems (P_a) on $\mathcal{H}_X$ and (Q_c) on $\mathcal{H}_Y$. Define

$$\Phi_{cd,ab} = \mathcal{Q}_{cd} \Phi \mathcal{P}_{ab}.$$

Theorem II.7.6 (Exact channel block calculus)**.** *Every linear map has the exact reconstruction*

$$\Phi = \sum_{a,b,c,d} \Phi_{cd,ab}.$$

If $\Psi : \mathrm{End}(\mathcal{H}_Y) \to \mathrm{End}(\mathcal{H}_Z)$, then

$$(\Psi\Phi)_{ef,ab} = \sum_{c,d} \Psi_{ef,cd} \Phi_{cd,ab}.$$

Proof. Insert the complete operator packet resolutions before and after Φ. For composition, insert the middle resolution between Ψ and Φ and collect the output and input packets. □

Standard Fact II.7.7 (Finite Kraus and Choi criteria)**.** A linear map $\Phi : M_n(\mathbb{C}) \to M_m(\mathbb{C})$ is completely positive exactly when it has a Kraus representation

$$\Phi(X) = \sum_r K_r X K_r^*.$$

It is trace preserving exactly when

$$\sum_r K_r^* K_r = I.$$

Equivalently, its Choi matrix

$$J(\Phi) = \sum_{i,j} \Phi(|i\rangle\langle j|) \otimes |i\rangle\langle j|$$

is positive, and trace preservation is

$$\operatorname{Tr}_{\text{out}} J(\Phi) = I_{\text{in}}.$$

These finite criteria are classical [24, 5, 15, 27].

II.7.2 Covariance and coherence order

Let S be unitary and $N \geq 1$ an integer with $S^N = I$, and define the conjugation action

$$\operatorname{Ad}_S(X) = SXS^*.$$

Theorem II.7.8 (Coherence order projectors)**.** *For $q \in \mathbb{Z}/N\mathbb{Z}$, define*

$$\mathcal{L}_q(X) = \frac{1}{N} \sum_{r=0}^{N-1} \omega^{-qr} S^r X S^{-r}.$$

Then the $\mathcal{L}_q$ form a complete orthogonal packet system on Hilbert–Schmidt operator space. Their ranges are the operators satisfying

$$SXS^* = \omega^q X.$$

If P_j are the state charge projectors, then

$$\operatorname{Ran} \mathcal{L}_q = \bigoplus_j P_{j+q} \operatorname{End}(\mathcal{H}) P_j.$$

Proof. Apply the cyclic character projector formula to the unitary representation Ad_S on operator space. The last identity follows from theorem II.3.5 applied to each operator packet. □

Theorem II.7.9 (Covariant channels preserve coherence order)**.** *Suppose*

$$\Phi(SXS^*) = S\Phi(X)S^*$$

for every X. Then

$$\Phi\mathcal{L}_q = \mathcal{L}_q\Phi$$

for every q. Thus Φ cannot mix distinct coherence orders.

Proof. The covariance identity says that Φ commutes with every power of Ad_S. It therefore commutes with each finite character average $\mathcal{L}_q$. □

Theorem II.7.10 (Choi symmetry criterion)**.** *Let $U : G \to \mathcal{U}(\mathcal{H})$ be a finite group representation. A channel Φ is covariant,*

$$\Phi\big(U(g)XU(g)^*\big) = U(g)\Phi(X)U(g)^*,$$

if and only if

$$(U(g) \otimes \overline{U(g)})J(\Phi)(U(g)^* \otimes U(g)^{\mathsf{T}}) = J(\Phi)$$

for every $g \in G$.

Proof. Let

$$\Omega = \sum_i |i\rangle \otimes |i\rangle .$$

Then

$$J(\Phi) = (\Phi \otimes \mathrm{id})(|\Omega\rangle\langle\Omega|).$$

The vector Ω is invariant under $U \otimes \overline{U}$. Covariance therefore implies the displayed Choi invariance. Conversely, the displayed transformation is the Choi matrix of

$$\mathrm{Ad}_{U(g)} \circ \Phi \circ \mathrm{Ad}_{U(g)^*} .$$

The Choi correspondence is injective, so invariance implies that this map equals Φ, which is covariance. □

Corollary II.7.11 (Choi packet decomposition)**.** *The Choi matrix of a covariant channel is block diagonal with respect to the isotypic decomposition of $U \otimes \overline{U}$.*

Proof. Apply theorem II.3.1 to the representation $U \otimes \overline{U}$ and use theorem II.7.10. □

Example II.7.12 (Amplitude damping)**.** For $0 \le \gamma \le 1$, let

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \qquad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}.$$

The channel

$$\Phi_\gamma(\rho) = K_0\rho K_0^* + K_1\rho K_1^*$$

is trace preserving. It sends the off diagonal packet $|0\rangle\langle 1|$ to $\sqrt{1-\gamma}\,|0\rangle\langle 1|$, while the excited population transfers partly into the ground population.

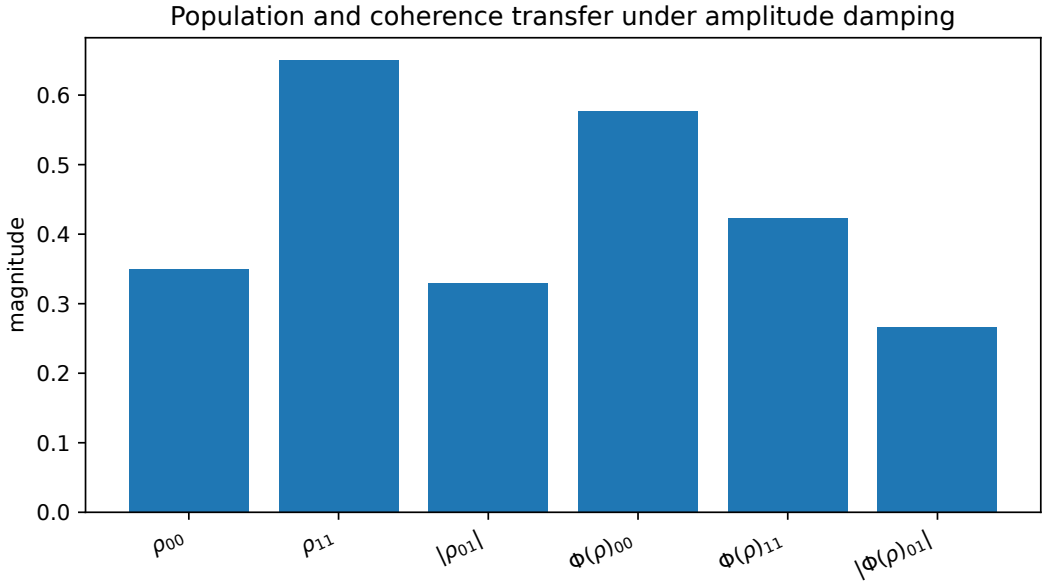


Figure 30. Amplitude damping transfers population and contracts the coherence packet.

II.7.3 Quantum dynamical semigroups

Standard Fact II.7.13 (Finite Lindblad generator)**.** A norm continuous semigroup of completely positive trace preserving maps on a finite matrix algebra, in the Schrödinger picture, has a generator of the form

$$\mathcal{G}(X) = -\mathrm{i}[H, X] + \sum_r \left(L_r X L_r^* - \frac{1}{2} L_r^* L_r X - \frac{1}{2} X L_r^* L_r \right),$$

where $H = H^*$. Conversely, every operator of this form generates a completely positive trace preserving semigroup [16].

Theorem II.7.14 (Charge preservation by a Lindblad generator)**.** *Let S be unitary and $N \geq 1$ an integer with $S^N = I$. Assume*

$$SHS^* = H$$

and, for every jump operator, there is a charge q_r such that

$$SL_rS^* = \omega^{q_r} L_r.$$

Then the Lindblad generator $\mathcal{G}$ preserves every coherence order:

$$\mathcal{G}\mathcal{L}_p = \mathcal{L}_p \mathcal{G}.$$

The quantum dynamical semigroup $\mathrm{e}^{t\mathcal{G}}$ has the same property.

Proof. Let X have coherence order p. The Hamiltonian commutator has order p because H has charge zero. The jump term $L_r X L_r^*$ has charge

$$q_r + p - q_r = p.$$

Also $L_r^* L_r$ has charge zero, so both anticommutator terms have charge p. Thus $\mathcal{G}$ preserves Ran $\mathcal{L}_p$. The exponential series then shows that $\mathrm{e}^{t\mathcal{G}}$ preserves the same packets. □

Corollary II.7.15 (Independent coherence evolution)**.** *Under the hypotheses of theorem II.7.14, each coherence component of an initial operator evolves within its own order. In particular, a density matrix in the zero order sector remains in that sector.*

Proof. Decompose the initial operator by theorem II.7.8 and apply the invariant packet statement to each component. □

Preservation of coherence order does not by itself give a closed classical Markov chain for the packet populations. Such scalar population equations require additional isotropy within each packet.

II.8 Quantum error correction as a compression obstruction

Let P be the orthogonal projector onto a code space

$$\mathcal{C} = P\mathcal{H}, \qquad k = \dim \mathcal{C} \geq 1.$$

The compression of $A \in \mathrm{End}(\mathcal{H})$ to the code is $PAP \in \mathrm{End}(\mathcal{C})$. The zero code has only the zero supported operator and no normalized code state. Part III treats that branch in the mathematical completion.

Definition II.8.1 (Code compression obstruction)**.** Let

$$q_P : \mathrm{End}(\mathcal{C}) \longrightarrow \mathrm{End}(\mathcal{C})/(\mathbb{C} I_{\mathcal{C}})$$

be the quotient map. Define

$$\mathrm{Ob}_P(A) = q_P(PAP).$$

Equivalently, one may use the traceless representative

$$\mathfrak{o}_P(A) = PAP - \frac{\mathrm{Tr}(PAP)}{k} P.$$

The class vanishes exactly when A acts as a scalar after compression to the code. Such an operator cannot distinguish normalized code vectors by expectation value.

Exact correction of an error span means that one completely positive trace preserving recovery $\mathcal{R}$ satisfies $\mathcal{R}(F\rho F^*) = q_F \rho$ for every F in that span and every code-supported matrix ρ, with a nonnegative scalar q_F depending on F but not on the logical input. For every normalized channel with Kraus operators in the same span, the same recovery restores the code input with total scalar one. Individual unnormalized errors are not required to have unit recovery weight.

Theorem II.8.2 (Knill–Laflamme as obstruction vanishing). *Let $E_1, \dots, E_r$ be an error family. The code $\mathcal{C}$ corrects their linear span exactly if and only if*

$$\mathrm{Ob}_P(E_a^* E_b) = 0$$

for all a, b. Equivalently, there is a matrix $c = (c_{ab})$ such that

$$P E_a^* E_b P = c_{ab} P$$

for all a, b.

Proof. The equivalence between the quotient statement and the scalar compression identity is immediate from the definition. The latter identity is the finite Knill–Laflamme criterion [14, 22]. □

The next result turns the scalar compression identities into a family of orthogonal syndrome packets. Those packets provide the coordinates used by the recovery map.

Theorem II.8.3 (Orthogonal syndrome construction). *Assume the Knill–Laflamme identities. Then the coefficient matrix c is positive semidefinite. Choose a unitary matrix W such that*

$$W c W^* = \mathrm{diag}(\lambda_1, \dots, \lambda_r), \qquad \lambda_j \geq 0,$$

and define

$$F_j = \sum_a \overline{W_{ja}} E_a.$$

For every j with $\lambda_j > 0$, the operator

$$V_j = \lambda_j^{-1/2} F_j P$$

is an isometry from the code into $\mathcal{H}$. Its range projector

$$Q_j = V_j V_j^*$$

is a syndrome packet, and the nonzero Q_j are pairwise orthogonal.

Proof. For $z \in \mathbb{C}^r$, let $F_z = \sum_a z_a E_a$. Then

$$P F_z^* F_z P = (z^* c z) P \geq 0,$$

so $c \geq 0$. The unitary change of error coordinates gives

$$P F_j^* F_\ell P = \lambda_j \delta_{j\ell} P.$$

Thus

$$V_j^* V_\ell = \delta_{j\ell} P.$$

Each V_j is an isometry on $P\mathcal{H}$, and distinct ranges are orthogonal. Their range projectors are therefore pairwise orthogonal. □

Theorem II.8.4 (Exact recovery on the error span)**.** *Under the hypotheses of theorem II.8.3, define on the syndrome support*

$$Q = \sum_{\lambda_j > 0} Q_j$$

the completely positive map

$$\mathcal{R}_0(X) = \sum_{\lambda_j > 0} V_j^* Q_j X Q_j V_j.$$

It extends to a quantum channel $\mathcal{R}$ on $\mathrm{End}(\mathcal{H})$*. If*

$$\Phi(X) = \sum_a E_a X E_a^*$$

is trace preserving on code states, then

$$\mathcal{R}\Phi(\rho) = \rho$$

for every density matrix $\rho = P\rho P$.

Proof. The map $\mathcal{R}_0$ is completely positive. It is trace preserving on operators supported in $Q\mathcal{H}$ because

$$\mathrm{Tr}(V_j^* Q_j X Q_j V_j) = \mathrm{Tr}(Q_j X Q_j).$$

Fix a code state ρ_0 and write $Q^\perp = I - Q$. Then

$$\mathcal{R}(X) = \mathcal{R}_0(QXQ) + \mathrm{Tr}(Q^\perp X)\rho_0$$

is completely positive and trace preserving. It extends $\mathcal{R}_0$ on operators supported in $Q\mathcal{H}$.

Unitary mixing of a Kraus family does not change the channel, so

$$\Phi(\rho) = \sum_j F_j \rho F_j^* = \sum_{\lambda_j > 0} \lambda_j V_j \rho V_j^*.$$

Recovery gives

$$\mathcal{R}\Phi(\rho) = \left(\sum_j \lambda_j \right) \rho.$$

Trace preservation on a code state implies $\sum_j \lambda_j = 1$. □

Corollary II.8.5 (Syndrome measurement preserves logical information)**.** *For a channel trace preserving on code states as in the preceding theorem, the completed projective measurement consisting of $(Q_j)_{\lambda_j>0}$ and $I - \sum_{\lambda_j>0} Q_j$ identifies the orthogonal error syndrome without resolving the encoded state inside the code. The complement outcome has probability zero on the corrupted code state.*

Proof. On syndrome j, the encoded state is carried by the isometry V_j. The outcome probability is λ_j, independent of the normalized code state, while the postmeasurement state is $V_j \rho V_j^*$. □

Definition II.8.6 (Approximate Knill–Laflamme defect)**.** For a finite error family, define the following defect. When the family is empty, assign the maximum the value zero.

$$\varepsilon_{\mathrm{KL}} = \max_{a,b} \inf_{c\in\mathbb{C}} \|PE_a^* E_b P - cP\| .$$

Proposition II.8.7 (Meaning of the approximate defect)**.** *For each pair (a, b), choose a minimizing scalar c_{ab}. For unit code vectors ψ, ϕ,*

$$|\langle E_a\psi, E_b\phi\rangle - c_{ab}\,\langle\psi,\phi\rangle| \le \varepsilon_{\mathrm{KL}}.$$

Thus $\varepsilon_{\mathrm{KL}}$ controls the deviation of error overlaps from their scalar code form.

Proof. The left side is

$$|\langle\psi, (PE_a^* E_b P - c_{ab}P)\phi\rangle| ,$$

which is bounded by the operator norm. □

No recovery fidelity estimate is inferred from this norm alone. Approximate error correction requires a separate continuity theorem and a declared channel metric.

Example II.8.8 (Three qubit repetition code)**.** Let

$$|0_L\rangle = |000\rangle , \qquad |1_L\rangle = |111\rangle ,$$

and take the errors

$$I, \quad X\otimes I\otimes I, \quad I\otimes X\otimes I, \quad I\otimes I\otimes X.$$

Their images of the code are four pairwise orthogonal two dimensional syndrome packets whose sum is the full eight dimensional Hilbert space.

Standard Fact II.8.9 (Correctable observable algebras)**.** Let Φ be a finite completely positive trace preserving channel with Kraus operators (E_a). A finite dimensional unital star subalgebra $\mathcal{M} \subseteq P\,\mathrm{End}(\mathcal{H})P$ is correctable when a completely positive trace preserving recovery $\mathcal{R}$ satisfies $P(\Phi^* \circ \mathcal{R}^*)(X)P = X$ for every $X \in \mathcal{M}$, where stars on maps denote Heisenberg adjoints. This holds exactly when every compressed product $PE_a^* E_b P$ belongs to the commutant $\mathcal{M}'$ on the code. The scalar code criterion is the special case $\mathcal{M} = P\,\mathrm{End}(\mathcal{H})P$ [1].



Figure 31. The syndrome packets of the three qubit repetition code. Each packet has dimension two and carries the full logical state.

II.9 Feshbach effective dynamics and hidden packet memory

Let P be an orthogonal projector, put $Q = I - P$, and write a self adjoint Hamiltonian as

$$H = \begin{pmatrix} A & B \\ B^{*} & D \end{pmatrix}$$

relative to $P\mathcal{H} \oplus Q\mathcal{H}$.

Definition II.9.1 (Feshbach map and self energy)**.** For $z \notin \operatorname{Spec} D$, define

$$\Sigma(z) = B(z - D)^{-1} B^{*}$$

and

$$F_{P}(z) = z - A - \Sigma(z).$$

Theorem II.9.2 (Exact projected resolvent)**.** *If $z \notin \operatorname{Spec} D$ and $F_{P}(z)$ is invertible, then*

$$P(z - H)^{-1} P = F_{P}(z)^{-1}$$

as an operator on $P\mathcal{H}$.

Proof. The lower right block of $z - H$ is $z - D$. Block Gaussian elimination gives

$$\begin{pmatrix} I & B(z-D)^{-1} \\ 0 & I \end{pmatrix} (z - H) \begin{pmatrix} I & 0 \\ (z-D)^{-1} B^{*} & I \end{pmatrix} = \begin{pmatrix} F_{P}(z) & 0 \\ 0 & z - D \end{pmatrix}.$$

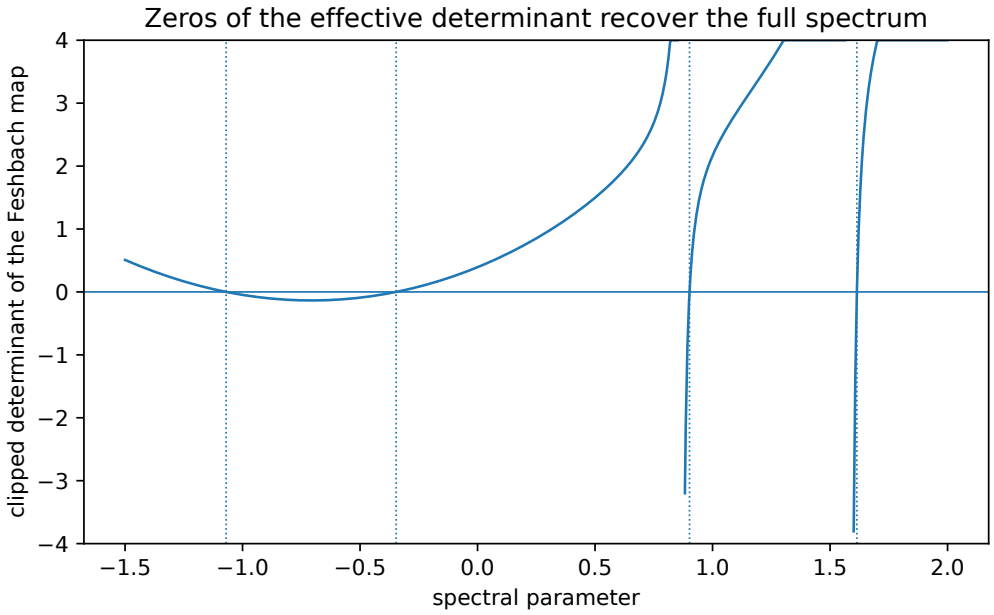


Figure 32. The zeros of the Feshbach determinant recover the eigenvalues of the full finite Hamiltonian away from the hidden spectrum.

Invert the factors. The retained resolvent is the block in the upper left corner of the resulting inverse. □

Theorem II.9.3 (Spectral equivalence). *Let* $z \notin \operatorname{Spec} D$. *Then*

$$z \in \operatorname{Spec} H \quad \Longleftrightarrow \quad 0 \in \operatorname{Spec} F_P(z).$$

The map

$$x \longmapsto \begin{pmatrix} x \\ (z-D)^{-1} B^* x \end{pmatrix}$$

is an isomorphism from $\operatorname{Ker} F_P(z)$ *onto* $\operatorname{Ker}(z-H)$.

Proof. The equation $(z-H)(x,y)^{\mathsf{T}} = 0$ is

$$(z-A)x - By = 0, \qquad -B^* x + (z-D)y = 0.$$

The second equation gives $y = (z-D)^{-1}B^*x$. Substitution into the first gives $F_P(z)x = 0$. The construction is reversible. □

Theorem II.9.4 (Upper half plane sign). *If* $\operatorname{Im} z > 0$, *then*

$$\operatorname{Im} \Sigma(z) = -(\operatorname{Im} z) B (z-D)^{-1} (\overline{z}-D)^{-1} B^* \leq 0.$$

Thus $-\Sigma$ *is an operator valued Herglotz function.*

Proof. For $R(z) = (z - D)^{-1}$, the resolvent identity gives

$$R(z) - R(z)^* = (\overline{z} - z)R(z)R(z)^*.$$

Divide by 2i, then multiply the result on the left by B and on the right by B^*. □

Proposition II.9.5 (Uniform self energy bounds)**.** *Let*

$$\delta = \operatorname{dist}(z, \operatorname{Spec} D) > 0, \qquad g = \|B\|.$$

Then

$$\|\Sigma(z)\| \leq \frac{g^2}{\delta}$$

and

$$\|\Sigma'(z)\| \leq \frac{g^2}{\delta^2}.$$

For real E, E' in an interval at distance at least δ from Spec D*,*

$$\|\Sigma(E) - \Sigma(E')\| \leq \frac{g^2}{\delta^2} |E - E'|.$$

Proof. The self adjoint resolvent satisfies

$$\left\|(z - D)^{-1}\right\| = \frac{1}{\operatorname{dist}(z, \operatorname{Spec} D)}.$$

Differentiate the resolvent to obtain

$$\Sigma'(z) = -B(z - D)^{-2}B^*.$$

Taking operator norms gives the first two estimates. Applying the derivative bound along the real interval gives the final mean value estimate. □

Proposition II.9.6 (Low energy expansion)**.** *Assume D is invertible and $|z| \left\|D^{-1}\right\| < 1$. Then*

$$\Sigma(z) = -\sum_{n=0}^{\infty} z^n B D^{-n-1} B^*$$

with norm convergence. In particular,

$$F_P(z) = z - A + BD^{-1}B^* + O(|z| \|B\|^2 \left\|D^{-1}\right\|^2).$$

Proof. Write

$$(z - D)^{-1} = -D^{-1}(I - zD^{-1})^{-1} = -\sum_{n=0}^{\infty} z^n D^{-n-1}$$

and multiply on the left by B and on the right by B^*. □

Proposition II.9.7 (Bare truncation error identity)**.** *If both $F_P(z)$ and $z - A$ are invertible, then*

$$P(z-H)^{-1}P - (z-A)^{-1} = F_P(z)^{-1}\Sigma(z)(z-A)^{-1}.$$

Proof. Use the inverse difference identity

$$X^{-1} - Y^{-1} = X^{-1}(Y-X)Y^{-1}$$

with $X = F_P(z)$ and $Y = z - A$. □

II.9.1 Time memory

Write a Schrödinger state as $\psi = x + y$, where $x = P\psi$ and $y = Q\psi$.

Theorem II.9.8 (Exact retained memory equation)**.** *The retained coordinate satisfies*

$$\begin{aligned} \mathrm{i}\dot{x}(t) &= Ax(t) + B\mathrm{e}^{-\mathrm{i}tD}y(0) \\ &\quad - \mathrm{i}\int_0^t K(t-s)x(s)\,\mathrm{d}s, \end{aligned}$$

where

$$K(t) = B\mathrm{e}^{-\mathrm{i}tD}B^*.$$

For $\operatorname{Re} s > 0$,

$$\int_0^\infty \mathrm{e}^{-st}K(t)\,\mathrm{d}t = B(s+\mathrm{i}D)^{-1}B^*.$$

Proof. The hidden equation is

$$\mathrm{i}\dot{y} = B^*x + Dy.$$

Variation of constants gives

$$y(t) = \mathrm{e}^{-\mathrm{i}tD}y(0) - \mathrm{i}\int_0^t \mathrm{e}^{-\mathrm{i}(t-s)D}B^*x(s)\,\mathrm{d}s.$$

Substitute into $\mathrm{i}\dot{x} = Ax + By$. The Laplace formula follows by diagonalizing D or integrating the norm convergent finite dimensional exponential. □

Theorem II.9.9 (Positive definite quantum memory)**.** *The kernel $K(t-s)$ is positive definite. For times $t_1, \dots, t_m$ and retained vectors $v_1, \dots, v_m$,*

$$\sum_{j,k=1}^m \left\langle v_j, K(t_j - t_k)v_k \right\rangle \geq 0.$$

Proof. Set

$$w = \sum_{k=1}^m \mathrm{e}^{\mathrm{i}t_kD}B^*v_k.$$

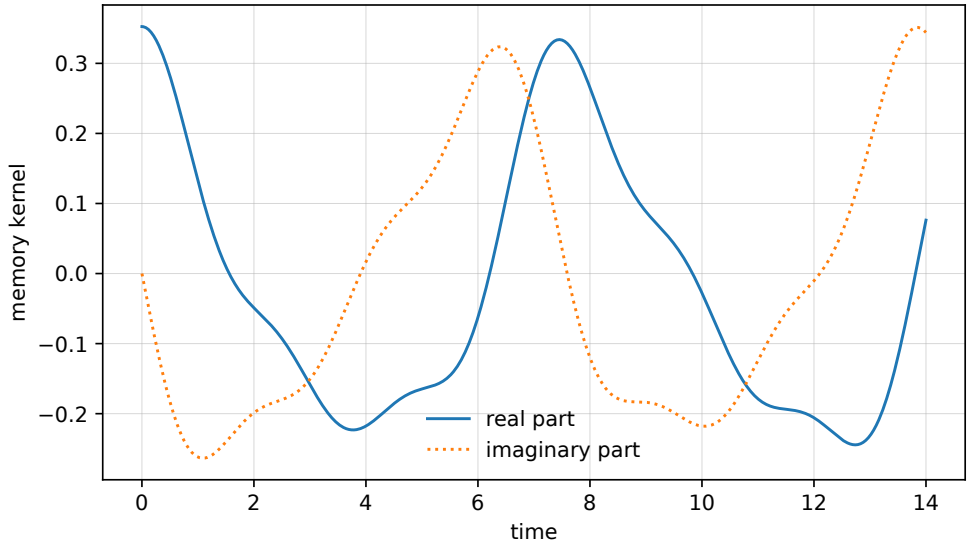


Figure 33. A finite hidden spectrum produces an oscillatory memory kernel. The retained dynamics therefore depend on the history of the retained state.

Then

$$\|w\|^2 = \sum_{j,k} \left\langle B^* v_j, \mathrm{e}^{-\mathrm{i}(t_j - t_k)D} B^* v_k \right\rangle = \sum_{j,k} \left\langle v_j, K(t_j - t_k) v_k \right\rangle .$$

□

Corollary II.9.10 (Finite oscillatory memory expansion)**.** *If*

$$D = \sum_r E_r Q_r$$

is the spectral decomposition of the hidden Hamiltonian, then

$$K(t) = \sum_r \mathrm{e}^{-\mathrm{i}tE_r} B Q_r B^* .$$

Every coefficient BQ_rB^* is positive semidefinite.

Proof. Inserting the spectral decomposition into the exponential gives the stated sum. The spectral projectors are positive semidefinite, so multiplication by the coupling and its adjoint preserves positivity of each coefficient. □

Theorem II.9.11 (Exact local closure criterion)**.** *Assume $y(0) = 0$. The retained equation is local and equal to*

$$\mathrm{i}\dot{x} = Ax$$

for every retained initial state exactly when $B = 0$. Equivalently, exact local closure holds exactly when the retained and hidden packets reduce H.

Proof. If $B = 0$, the retained and hidden equations decouple. Conversely, suppose that every solution with $y(0) = 0$ satisfies $x(t) = \mathrm{e}^{-\mathrm{i}tA}x(0)$. The full block equation gives

$$\ddot{x}(0) = -(A^2 + BB^*)x(0),$$

while the declared local equation gives

$$\ddot{x}(0) = -A^2x(0).$$

Since this holds for every retained initial vector, $BB^* = 0$. Hence $B = 0$. □

II.9.2 Stability across models

Theorem II.9.12 (Stability of the self energy)**.** *Let $D = D^*$ and $\widetilde{D} = \widetilde{D}^*$ act on one fixed hidden Hilbert space, and let $B, \widetilde{B}$ map that hidden space to one fixed retained Hilbert space. Define*

$$\Sigma(z) = B(z - D)^{-1}B^*, \qquad \widetilde{\Sigma}(z) = \widetilde{B}(z - \widetilde{D})^{-1}\widetilde{B}^*.$$

Assume

$$\|B\|, \left\|\widetilde{B}\right\| \leq g$$

and

$$\operatorname{dist}(z, \operatorname{Spec} D), \operatorname{dist}(z, \operatorname{Spec} \widetilde{D}) \geq \delta > 0.$$

Then

$$\left\|\Sigma(z) - \widetilde{\Sigma}(z)\right\| \leq \frac{2g}{\delta}\left\|B - \widetilde{B}\right\| + \frac{g^2}{\delta^2}\left\|D - \widetilde{D}\right\|.$$

Proof. Add and subtract $\widetilde{B}(z - D)^{-1}B^*$ and $\widetilde{B}(z - D)^{-1}\widetilde{B}^*$. The first two differences give the coupling term. For the last, use

$$(z - D)^{-1} - (z - \widetilde{D})^{-1} = (z - D)^{-1}(D - \widetilde{D})(z - \widetilde{D})^{-1}.$$

Apply the resolvent bounds. □

Corollary II.9.13 (Uniform Feshbach control)**.** *For a family represented on fixed retained and hidden Hilbert spaces, with self adjoint hidden Hamiltonians, a common hidden spectral separation δ, and a common coupling bound g, the self energies are uniformly bounded and uniformly Lipschitz on the declared spectral region. If the couplings and hidden Hamiltonians converge in norm, then the self energies converge uniformly there.*

Proof. The common gap and coupling bounds allow proposition II.9.5 and theorem II.9.12 to be applied uniformly across the family. □

This result identifies hypotheses that make Feshbach control uniform in the cutoff. Whether a chosen geometric refinement tower satisfies those hypotheses remains a separate question.

II.10 Euclidean transfer and finite reflection positivity

Let $H = H^*$ on a nonzero finite dimensional Hilbert space and let

$$E_0 = \min \operatorname{Spec} H.$$

For a lattice time step $a > 0$, define

$$T = \mathrm{e}^{-a(H - E_0 I)}.$$

Theorem II.10.1 (Finite Euclidean transfer). *The operator T is positive definite, self adjoint, and contractive. Its spectrum lies in $(0, 1]$, and*

$$T^n = \mathrm{e}^{-na(H - E_0 I)}$$

for every $n \in \mathbb{N}$.

Proof. Diagonalize H. Each eigenvalue of T is

$$\mathrm{e}^{-a(E_j - E_0)}$$

and therefore lies in $(0, 1]$. The power law follows from the functional calculus. □

Theorem II.10.2 (Hamiltonian reconstruction from transfer). *Let T be positive definite, self adjoint, and contractive. Then*

$$H_T = -a^{-1} \log T$$

is a nonnegative self adjoint operator and

$$T^n = \mathrm{e}^{-naH_T}.$$

It is the unique nonnegative self adjoint operator with $T = \mathrm{e}^{-aH_T}$.

Proof. Apply the real logarithm to the positive spectrum of T. The spectral calculus gives existence and the power identity. Injectivity of the scalar exponential on $[0, \infty)$ gives uniqueness. □

Theorem II.10.3 (Finite reflection positivity). *For vectors $v_1, \dots, v_m \in \mathcal{H}$ and nonnegative integers $n_1, \dots, n_m$, the matrix*

$$G_{jk} = \left\langle v_j, T^{n_j + n_k} v_k \right\rangle$$

is positive semidefinite. Equivalently,

$$\sum_{j,k} \overline{c_j} c_k \left\langle v_j, T^{n_j + n_k} v_k \right\rangle \geq 0$$

for every $c \in \mathbb{C}^m$.

Proof. Since T is positive and self adjoint,

$$\left\langle v_j, T^{n_j + n_k} v_k \right\rangle = \left\langle T^{n_j} v_j, T^{n_k} v_k \right\rangle.$$

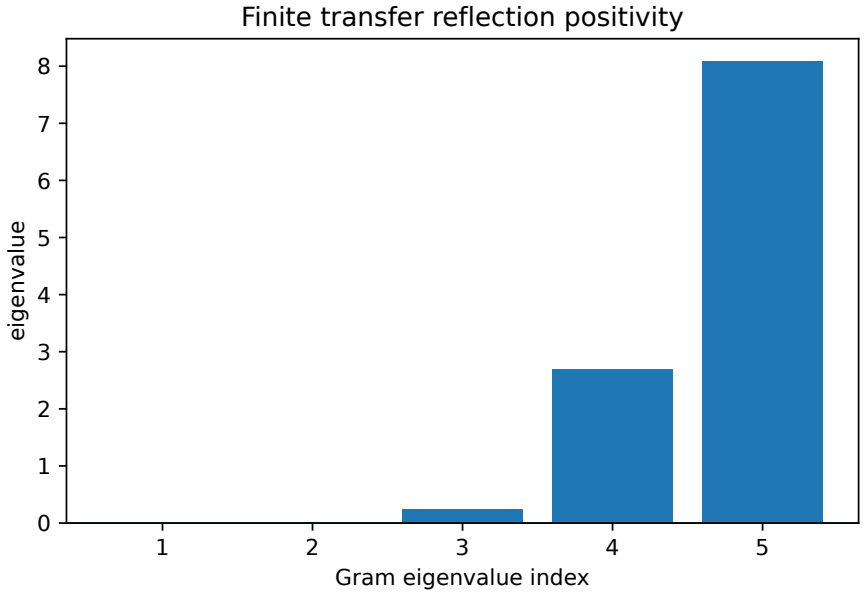


Figure 34. Eigenvalues of a finite reflection positive Gram matrix. Small numerical negative values would signal either roundoff or failure of the transfer assumptions.

The quadratic form is the squared norm of

$$\sum_k c_k T^{n_k} v_k.$$

□

Corollary II.10.4 (Reflection positive correlators)**.** *Fix a vector Ω and operators $A_1, \dots, A_m$. The matrix*

$$C_{jk} = \langle A_j \Omega, T^{n_j + n_k} A_k \Omega \rangle$$

is positive semidefinite.

Proof. Apply theorem II.10.3 with $v_j = A_j \Omega$. □

Proposition II.10.5 (Ground packet convergence)**.** *Let P_0 be the spectral projector of H at E_0, and let*

$$\Delta = \min\{E_j - E_0 : E_j > E_0\}.$$

If an excited eigenvalue exists, then

$$\|T^n - P_0\| = \mathrm{e}^{-na\Delta}.$$

Proof. In the spectral decomposition, $T^n - P_0$ has eigenvalues zero on the ground space and $\mathrm{e}^{-na(E_j - E_0)}$ on the excited eigenspaces. The largest is $\mathrm{e}^{-na\Delta}$. □

Proposition II.10.6 (Symmetry resolved transfer)**.** *If $[H, P_a] = 0$ for a complete packet family, then*

$$T = \sum_a P_a T P_a$$

and each reflection positive Gram matrix decomposes as the sum of its packet Gram matrices.

Proof. The functional calculus preserves commutation. Decompose every vector into packet components and use orthogonality. □

Finite reflection positivity is a necessary structural property of transfer data. It is not by itself the continuum Osterwalder–Schrader theorem, which also requires a suitable algebra of Euclidean fields, covariance, regularity, clustering, and a controlled continuum limit [19, 20, 10].

II.11 Finite spectral geometry and diffusion dimension

A finite spectral datum consists of a finite dimensional Hilbert space, a nonnegative self adjoint operator L, and any declared packet family. One may take $L = D^2$ for a finite Dirac type operator or a graph Laplacian. Related geometric viewpoints appear in [6, 26].

Definition II.11.1 (Heat trace and spectral dimension)**.** Let the eigenvalues of L be $\lambda_1, \dots, \lambda_N$, with at least one zero eigenvalue. Define

$$Z_L(t) = \mathrm{Tr}(\mathrm{e}^{-tL})$$

and

$$d_L(t) = -2\frac{\mathrm{d}\log Z_L(t)}{\mathrm{d}\log t} = 2t\frac{\mathrm{Tr}(L\mathrm{e}^{-tL})}{\mathrm{Tr}(\mathrm{e}^{-tL})}.$$

Proposition II.11.2 (Finite scale limits)**.** *For every finite nonnegative L with nontrivial kernel,*

$$\lim_{t\downarrow 0} d_L(t) = 0$$

and

$$\lim_{t\to\infty} d_L(t) = 0.$$

For every $t > 0$,

$$0 \le d_L(t) \le 2t\,\|L\|\,.$$

Proof. The formula in definition II.11.1 is twice t times a weighted average of the eigenvalues. This gives the bound and the limit at zero. At large time, the denominator converges to $\dim \mathrm{Ker}\, L$, while every positive eigenvalue contribution in the numerator decays exponentially. □

Thus a finite system cannot have a nonzero spectral dimension at the extreme ultraviolet or infrared ends. A geometric dimension can appear only in a refinement window.

Proposition II.11.3 (Heat packet decomposition). *If $[L, P_a] = 0$ for every packet, then*

$$Z_L(t) = \sum_a Z_{L,a}(t), \qquad Z_{L,a}(t) = \mathrm{Tr}(P_a \mathrm{e}^{-tL}).$$

The global spectral dimension is the average of the packet dimensions weighted by their heat traces:

$$d_L(t) = \sum_{a: Z_{L,a}(t)>0} \frac{Z_{L,a}(t)}{Z_L(t)} d_{L,a}(t),$$

where

$$d_{L,a}(t) = 2t \frac{\mathrm{Tr}(P_a L \mathrm{e}^{-tL})}{Z_{L,a}(t)}$$

for nonempty packet heat traces.

Proof. Commutation gives a block diagonal heat operator. Add the traces and the corresponding first spectral moments, then divide by the total heat trace. □

II.11.1 Periodic lattices

Let $N \geq 2$ and let $\Lambda_N = (\mathbb{Z}/N\mathbb{Z})^d$ be the unit flat torus grid with spacing $1/N$. The standard periodic difference Laplacian has eigenvalues

$$\lambda_N(k) = 4N^2 \sum_{j=1}^{d} \sin^2\left(\frac{\pi k_j}{N}\right),$$

where k runs through centered residue representatives. This is the standard periodic finite difference Laplacian [25].

Theorem II.11.4 (Low mode dispersion in dimension d). *For every centered mode k,*

$$0 \leq 4\pi^2 |k|^2 - \lambda_N(k) \leq \frac{4\pi^4}{3N^2} \sum_{j=1}^{d} k_j^4.$$

Proof. For $x \in [0, \pi/2]$,

$$\sin x \leq x$$

and

$$\sin x \geq x - \frac{x^3}{6}.$$

Thus

$$0 \leq x^2 - \sin^2 x \leq \frac{x^4}{3}.$$

Apply this with $x = \pi |k_j| / N$, multiply by $4N^2$, and sum over j. □

Theorem II.11.5 (Fixed time heat trace convergence)**.** *For every $t > 0$,*

$$\sum_{k \in \Lambda_N^*} \mathrm{e}^{-t\lambda_N(k)} \longrightarrow \sum_{k \in \mathbb{Z}^d} \mathrm{e}^{-4\pi^2 t|k|^2}$$

as $N \to \infty$. The first spectral moments also converge:

$$\sum_{k \in \Lambda_N^*} \lambda_N(k) \mathrm{e}^{-t\lambda_N(k)} \longrightarrow \sum_{k \in \mathbb{Z}^d} 4\pi^2 |k|^2 \mathrm{e}^{-4\pi^2 t|k|^2}.$$

Consequently, the discrete spectral dimensions converge to the continuum torus spectral dimension at every fixed positive time.

Proof. For a fixed integer mode, theorem II.11.4 gives

$$\lambda_N(k) \longrightarrow 4\pi^2 |k|^2 .$$

For centered modes, the inequality

$$\sin x \geq \frac{2x}{\pi}, \qquad 0 \leq x \leq \frac{\pi}{2},$$

gives

$$\lambda_N(k) \geq 16 |k|^2 .$$

Also $\lambda_N(k) \leq 4\pi^2 |k|^2$. Extend the discrete summands by zero outside the centered residue box. The heat trace terms are dominated by

$$\mathrm{e}^{-16t|k|^2},$$

and the moment terms by

$$4\pi^2 |k|^2 \mathrm{e}^{-16t|k|^2}.$$

Both dominating sequences are summable on $\mathbb{Z}^d$. Dominated convergence proves the two limits. Their quotient gives spectral dimension convergence. □

Standard Fact II.11.6 (Theta transformation)**.** The continuum flat torus heat trace satisfies

$$Z_d(t) = \sum_{k \in \mathbb{Z}^d} \mathrm{e}^{-4\pi^2 t|k|^2} = (4\pi t)^{-d/2} \sum_{n \in \mathbb{Z}^d} \mathrm{e}^{-|n|^2/(4t)}.$$

This is the Poisson summation formula for a Gaussian.

Theorem II.11.7 (Continuum torus small time dimension)**.** *Let*

$$d_{\text{torus}}(t) = -2 \frac{\mathrm{d} \log Z_d(t)}{\mathrm{d} \log t}.$$

Then

$$d_{\text{torus}}(t) = d + O\left(t^{-1} \mathrm{e}^{-1/(4t)}\right)$$

as $t \downarrow 0$.

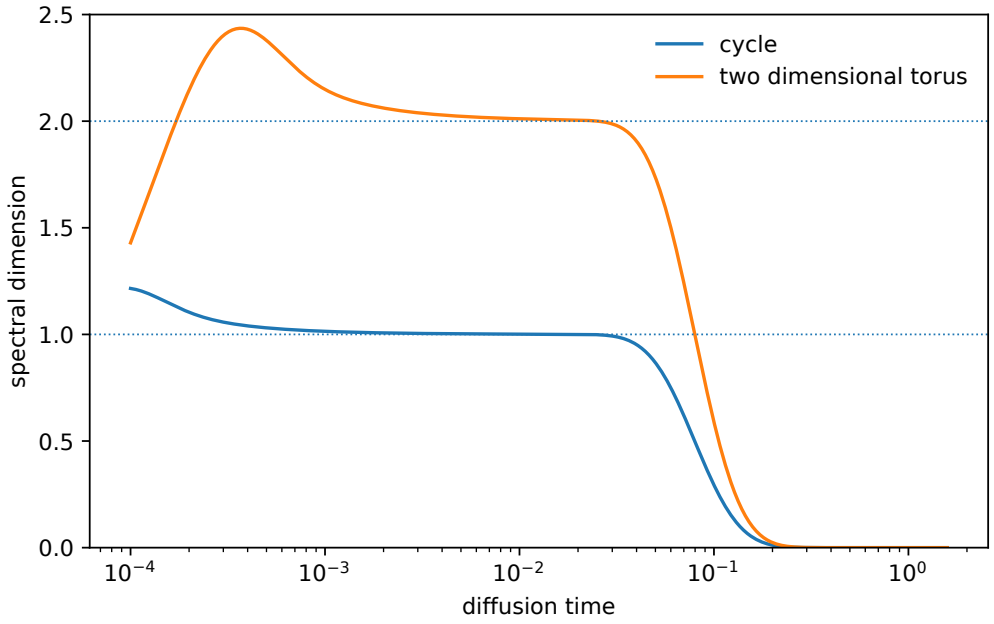


Figure 35. Finite periodic lattices show an intermediate dimension window. The ultraviolet and infrared limits return to zero because the state space is finite.

Proof. By standard fact II.11.6,

$$Z_d(t) = (4\pi t)^{-d/2}(1 + R(t)),$$

where

$$R(t) = \sum_{n \neq 0} \mathrm{e}^{-|n|^2/(4t)} = O(\mathrm{e}^{-1/(4t)}).$$

Differentiation gives

$$tR'(t) = O\left(t^{-1}\mathrm{e}^{-1/(4t)}\right).$$

Substituting this estimate into the logarithmic derivative gives the stated small time expansion. □

Corollary II.11.8 (Rigorous intermediate dimension window)**.** *For every $\varepsilon > 0$, there is a diffusion time $t_\varepsilon > 0$ and an integer N_ε such that*

$$\left|d_{L_N}(t_\varepsilon) - d\right| < \varepsilon$$

for every $N \geq N_\varepsilon$.

Proof. Choose t_ε so that theorem II.11.7 is within $\varepsilon/2$ of d. Then apply theorem II.11.5 to choose N_ε so that the discrete spectral dimension is within $\varepsilon/2$ of the continuum value. □

The order of limits matters. Taking $t \downarrow 0$ at fixed N gives zero. Taking $N \to \infty$ at fixed positive t, then taking $t \downarrow 0$, gives the continuum dimension.

II.12 Constraint closure and anomaly packets

Let $C_1, \dots, C_m$ be self adjoint constraint operators on a finite dimensional Hilbert space. The distinction between constraint action and constraint closure is standard in constrained Hamiltonian theory [7, 11]. Write

$$\mathfrak{c} = \operatorname{span}_{\mathbb{C}}\{C_1, \dots, C_m\} \subseteq \operatorname{End}(\mathcal{H}).$$

Definition II.12.1 (Universal closure obstruction)**.** Let

$$q_{\mathfrak{c}} : \operatorname{End}(\mathcal{H}) \longrightarrow \operatorname{End}(\mathcal{H})/\mathfrak{c}$$

be the quotient map. Define

$$\mathrm{Ob}^{\mathrm{cl}}_{ab} = q_{\mathfrak{c}}([C_a, C_b]).$$

Theorem II.12.2 (Finite constraint closure criterion)**.** *The constraint span $\mathfrak{c}$ is closed under commutators exactly when*

$$\mathrm{Ob}^{\mathrm{cl}}_{ab} = 0$$

for all a, b.

Proof. The quotient class vanishes exactly when $[C_a, C_b] \in \mathfrak{c}$. This holds for every pair exactly when $\mathfrak{c}$ is a Lie algebra under the commutator. □

If structure coefficients ${f_{ab}}^{c}$ are prescribed, define the fixed structure anomaly

$$\mathfrak{A}_{ab} = [C_a, C_b] - \mathrm{i}\sum_c {f_{ab}}^{c} C_c.$$

Exact realization of the prescribed algebra is the stronger condition $\mathfrak{A}_{ab} = 0$.

Proposition II.12.3 (Physical compression is not a closure test)**.** *Let*

$$\mathcal{H}_{\mathrm{phys}} = \bigcap_a \operatorname{Ker} C_a$$

and let P_{phys} be its orthogonal projector. Then

$$P_{\mathrm{phys}}[C_a, C_b]P_{\mathrm{phys}} = 0$$

for every pair a, b, whether or not the constraint span closes.

Proof. Self adjointness and $C_a P_{\mathrm{phys}} = 0$ give $P_{\mathrm{phys}} C_a = 0$. Both terms in the compressed commutator therefore vanish. □

The quotient in definition II.12.1 is needed because closure concerns the operator algebra. Compression to physical states tests only the action on the common kernel, where the constraints already vanish.

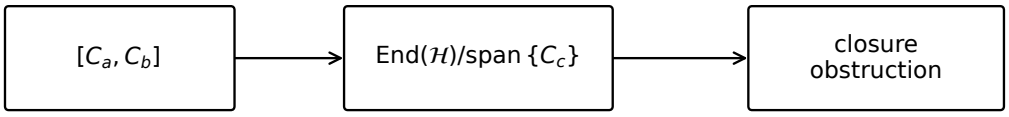


Figure 36. The closure obstruction lives in a quotient by the constraint span. Physical compression is automatic for self adjoint constraints and cannot replace this quotient.

Theorem II.12.4 (Invariance of the physical space)**.** *Let $H = H^*$. Suppose*

$$[H, C_a] = \sum_c b_a{}^c C_c$$

for every a. Then H preserves $\mathcal{H}_{\text{phys}}$, and so does $\mathrm{e}^{-\mathrm{i}tH}$ for every real t.

Proof. For $\psi \in \mathcal{H}_{\text{phys}}$,

$$C_a H\psi = HC_a\psi - [H, C_a]\psi = 0.$$

Thus $H\psi \in \mathcal{H}_{\text{phys}}$. The exponential preserves every invariant finite dimensional subspace. □

II.12.1 Constraint refinement

Let $(\mathcal{H}_\ell, C_{\ell,a})$ and $(\mathcal{H}_{\ell+1}, C_{\ell+1,a})$ be two levels, and let

$$R_\ell : \mathcal{H}_\ell \longrightarrow \mathcal{H}_{\ell+1}$$

be an isometry.

Theorem II.12.5 (Exact constraint transport). *Assume*

$$C_{\ell+1,a}R_\ell = R_\ell C_{\ell,a}$$

for every a. Then

$$[C_{\ell+1,a}, C_{\ell+1,b}]R_\ell = R_\ell[C_{\ell,a}, C_{\ell,b}].$$

If the same structure coefficients are used at both levels, then

$$\mathfrak{A}_{\ell+1,ab}R_\ell = R_\ell\mathfrak{A}_{\ell,ab}.$$

The isometry maps coarse physical states into fine physical states.

Proof. Apply the intertwining identity twice to each product in the commutator. The anomaly identity follows by subtracting the same linear combination of constraints. If $C_{\ell,a}\psi = 0$, then

$$C_{\ell+1,a}R_\ell\psi = R_\ell C_{\ell,a}\psi = 0.$$

□

Proposition II.12.6 (Approximate constraint transport). *Assume*

$$\|C_{\ell+1,a}R_\ell - R_\ell C_{\ell,a}\| \le \varepsilon_a$$

and

$$\|C_{\ell,a}\|, \|C_{\ell+1,a}\| \le M.$$

Then a coarse physical unit vector ψ satisfies

$$\|C_{\ell+1,a}R_\ell\psi\| \le \varepsilon_a.$$

Also,

$$\begin{aligned}\|[C_{\ell+1,a}, C_{\ell+1,b}]R_\ell - R_\ell[C_{\ell,a}, C_{\ell,b}]\| \\ \le 2M(\varepsilon_a + \varepsilon_b).\end{aligned}$$

Proof. The first claim follows by applying the intertwining defect to ψ. Write

$$E_a = C_{\ell+1,a}R_\ell - R_\ell C_{\ell,a}.$$

Expanding the two commutators gives four error terms:

$$E_a C_{\ell,b}, \quad C_{\ell+1,a}E_b, \quad -E_b C_{\ell,a}, \quad -C_{\ell+1,b}E_a.$$

The triangle inequality and the stated bounds on the constraints and defects give the claimed estimate. □

Approximate transport controls finite violations. It does not turn a nonzero anomaly into an exact continuum algebra without a limit theorem and a common operator domain.

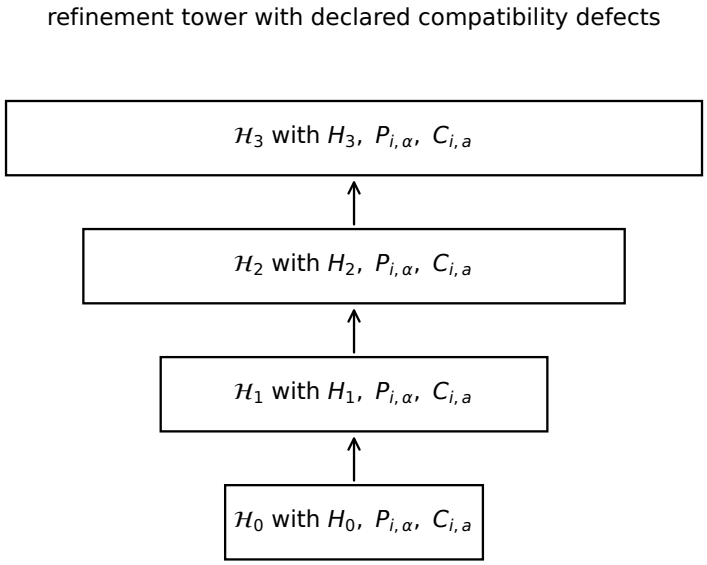


Figure 37. A refinement tower carries Hamiltonians, packets, constraints, and transfer data. Each family has its own compatibility defect.

II.13 Quantum refinement towers

Definition II.13.1 (Finite quantum refinement tower)**.** A finite quantum refinement tower consists of Hilbert spaces $\mathcal{H}_\ell$, isometries

$$R_\ell : \mathcal{H}_\ell \longrightarrow \mathcal{H}_{\ell+1},$$

self adjoint Hamiltonians H_ℓ, packet projectors $P_{\ell,\alpha}$, constraints $C_{\ell,r}$, and positive transfer operators T_ℓ. Exact compatibility means that every declared object intertwines with R_ℓ. Approximate compatibility is recorded by the corresponding operator norm defects.

II.13.1 Exact bounded direct limits

Let $\mathcal{H}_\infty$ be the Hilbert direct limit of the isometric system, and let

$$\iota_\ell : \mathcal{H}_\ell \longrightarrow \mathcal{H}_\infty$$

be the canonical isometries.

Theorem II.13.2 (Bounded Hamiltonian direct limit)**.** *Assume*

$$H_{\ell+1}R_\ell = R_\ell H_\ell$$

and

$$\sup_\ell \|H_\ell\| < \infty.$$

Then there is a unique bounded self adjoint operator H_∞ on $\mathcal{H}_\infty$ such that

$$H_\infty \iota_\ell = \iota_\ell H_\ell$$

for every ℓ. It satisfies

$$\mathrm{e}^{-\mathrm{i}tH_\infty}\iota_\ell = \iota_\ell \mathrm{e}^{-\mathrm{i}tH_\ell}$$

and

$$\mathrm{e}^{-tH_\infty}\iota_\ell = \iota_\ell \mathrm{e}^{-tH_\ell}$$

when the Hamiltonians are nonnegative.

Proof. Define H_∞ on the algebraic union of the embedded levels by

$$H_\infty \iota_\ell x = \iota_\ell H_\ell x.$$

Compatibility makes the definition independent of the chosen level. The uniform norm bound makes the operator bounded on the dense union, so it extends uniquely to the completion. It is symmetric because every H_ℓ is self adjoint. A bounded symmetric operator defined on the whole Hilbert space is self adjoint. The functional calculus intertwining follows first for polynomials and then for exponentials by uniform convergence. □

Theorem II.13.3 (Exact packet direct limit)**.** *Assume the packet families have one fixed finite address set at every level and each packet address a is compatible in the sense*

$$P_{\ell+1,a}R_\ell = R_\ell P_{\ell,a}.$$

Then there is an orthogonal projector $P_{\infty,a}$ on $\mathcal{H}_\infty$ satisfying

$$P_{\infty,a}\iota_\ell = \iota_\ell P_{\ell,a}.$$

Finite completeness and orthogonality pass to the direct limit.

Proof. Use the same construction as in theorem II.13.2. Every projector has norm at most one, so no additional boundedness assumption is needed. The projector identities hold on the dense algebraic union and hence on the completion. □

II.13.2 Norm summable approximation

The next theorem places all approximating Hamiltonians on a common Hilbert space. It applies to bounded effective Hamiltonians. An unbounded field Hamiltonian requires a different convergence argument.

Theorem II.13.4 (Norm summable Hamiltonian tower)**.** *Let (H_n) be bounded self adjoint operators on one Hilbert space and assume*

$$\sum_{n=0}^{\infty} \|H_{n+1} - H_n\| < \infty.$$

Then H_n converges in norm to a bounded self adjoint operator H. For every $T > 0$,

$$\sup_{|t|\le T} \left\|\mathrm{e}^{-\mathrm{i}tH_n} - \mathrm{e}^{-\mathrm{i}tH}\right\| \le T \left\|H_n - H\right\|.$$

If every $H_n \ge 0$, then

$$\sup_{0\le t\le T} \left\|\mathrm{e}^{-tH_n} - \mathrm{e}^{-tH}\right\| \le T \left\|H_n - H\right\|.$$

Proof. The summability condition makes (H_n) Cauchy in operator norm. The limit is self adjoint. Duhamel's identity gives

$$\mathrm{e}^{-\mathrm{i}tH_n} - \mathrm{e}^{-\mathrm{i}tH} = -\mathrm{i}\int_0^t \mathrm{e}^{-\mathrm{i}(t-s)H_n}(H_n - H)\mathrm{e}^{-\mathrm{i}sH}\,\mathrm{d}s.$$

Both unitary factors have norm one, which gives the first estimate. The same argument gives the Euclidean estimate because the semigroups generated by nonnegative Hamiltonians are contractions. □

Corollary II.13.5 (Limit of packet leakage)**.** *Let P_n be orthogonal projectors and H_n bounded self adjoint operators on one Hilbert space, with $P_n \to P$ and $H_n \to H$ in norm. Then for every bounded time interval,*

$$(I - P_n)\mathrm{e}^{-\mathrm{i}tH_n}P_n \longrightarrow (I - P)\mathrm{e}^{-\mathrm{i}tH}P$$

uniformly in time.

Proof. Add and subtract terms so that one factor changes at a time. The projector terms converge uniformly because $P_n \to P$ in norm. Duhamel's identity gives

$$\sup_{|t|\le T} \left\|\mathrm{e}^{-\mathrm{i}tH_n} - \mathrm{e}^{-\mathrm{i}tH}\right\| \le T \left\|H_n - H\right\|,$$

which proves uniform convergence of the middle factor. □

Theorem II.13.6 (Reflection positivity is closed under limits)**.** *Suppose $C_n(j,k)$ are reflection positive kernels on $\mathbb{N}$, meaning every finite matrix*

$$\left(C_n(j_r, j_s)\right)_{r,s}$$

is positive semidefinite. If

$$C_n(j,k) \longrightarrow C(j,k)$$

for every pair, then C is reflection positive.

Proof. For a fixed finite set of times and coefficients, the corresponding quadratic forms are nonnegative and converge to the quadratic form of C. The limit is nonnegative. □

Theorem II.13.7 (Constraint closure under norm convergence)**.** *Let $C_{n,a} \to C_a$ in norm for every a, and fix structure coefficients ${f_{ab}}^c$. If*

$$\left\| [C_{n,a}, C_{n,b}] - \mathrm{i} \sum_c {f_{ab}}^c C_{n,c} \right\| \longrightarrow 0,$$

then

$$[C_a, C_b] = \mathrm{i} \sum_c {f_{ab}}^c C_c.$$

Proof. Operator multiplication and the commutator are continuous in operator norm. Pass to the limit in the anomaly identity. □

Theorem II.13.8 (Heat trace convergence from trace norm)**.** *Let L_n, L be nonnegative self adjoint operators on one Hilbert space. For a fixed $t > 0$, assume their heat operators are trace class and that*

$$\mathrm{e}^{-tL_n} \longrightarrow \mathrm{e}^{-tL}$$

in trace norm. Then

$$\mathrm{Tr}(\mathrm{e}^{-tL_n}) \longrightarrow \mathrm{Tr}(\mathrm{e}^{-tL}).$$

If also the heat moment operators are trace class and

$$L_n \mathrm{e}^{-tL_n} \longrightarrow L\mathrm{e}^{-tL}$$

in trace norm and the limiting heat trace is nonzero, then the spectral dimensions defined by the heat moment ratio converge at that time.

Proof. The trace is continuous in trace norm. Apply this to the heat operators and to the first heat moments, then take the quotient in definition II.11.1. □

Remark II.13.9. The theorems above produce bounded continuum Hamiltonians. A physically useful continuum Hamiltonian is often unbounded. Its construction generally requires strong resolvent convergence, convergence of closed quadratic forms, or another method that controls operator domains [13, 21]. None of those conclusions follows from finite matrix convergence without additional hypotheses.

II.14 Continuum requirements and scope

The finite theorems establish exact identities and criteria for bounded limits. Passing from a finite transfer model to a continuum theory still requires several separate arguments. This section states those remaining mathematical tasks.

Definition II.14.1 (Anomaly free continuum constraint system)**.** A continuum constraint system consists of a Hilbert space, a common invariant dense domain $\mathcal{D}$, and operators C_a on $\mathcal{D}$. It is anomaly free for prescribed structure coefficients when

$$[C_a, C_b]\psi = \mathrm{i}\sum_c f_{ab}{}^c C_c \psi$$

for every $\psi \in \mathcal{D}$, and when the associated gauge evolution is defined on the physical quotient or physical subspace.

Norm closure of bounded finite constraints, as in theorem II.13.7, is one special case. It does not treat unbounded constraints, changing domains, structure functions, or renormalized operator products.

Definition II.14.2 (Reflection positive Euclidean limit)**.** A reflection positive Euclidean limit requires a limiting algebra of observables, Euclidean time translations, a reflection operation, and correlation functions whose reflected quadratic forms are nonnegative. A field theoretic reconstruction also requires the regularity and covariance assumptions needed to construct a Hilbert space, a vacuum, and a self adjoint time translation generator.

Theorem II.13.6 shows that positivity of each fixed finite Gram matrix survives pointwise convergence. It does not prove existence, locality, clustering, or the remaining Osterwalder–Schrader axioms.

Definition II.14.3 (Einstein infrared criterion)**.** A quantum geometric tower meets an Einstein infrared criterion only if an effective action is controlled at long wavelengths and has an expansion whose leading geometric terms take the form

$$\Gamma_{\text{eff}}[g] = \int \sqrt{|g|}\left(\Lambda + \frac{1}{16\pi G}R + \text{controlled higher order terms}\right)\mathrm{d}^d x.$$

Here the volume coefficient includes the sign and normalization written explicitly in the cosmological term in Part I. The expansion must respect the limiting gauge identities and must include an estimate for the omitted terms on the declared infrared regime.

A spectral dimension close to four does not imply this criterion. A heat trace records eigenvalue growth. It does not determine the tensor structure, gauge identities, sign, or interaction coefficients of an effective action.

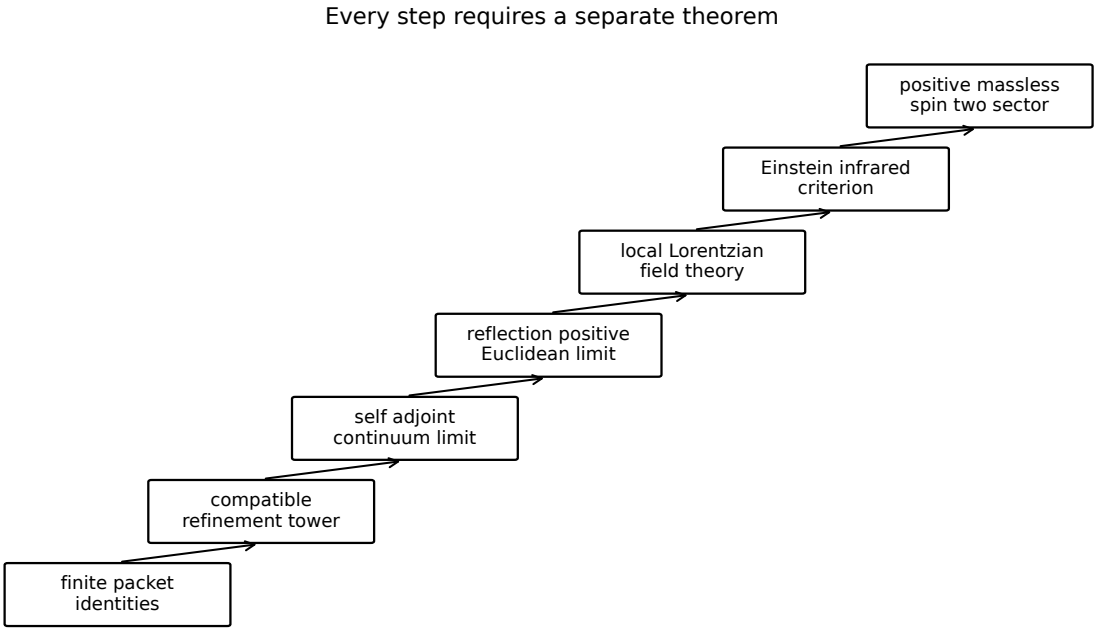


Figure 38. The finite identities occupy the first step. Every later step needs a separate convergence or reconstruction theorem.

Definition II.14.4 (Massless spin two criterion)**.** A positive massless spin two sector requires a physical two point function or resolvent with a massless pole in the transverse traceless sector, positive residue on physical states, absence of negative norm modes, and compatibility with the limiting gauge constraints. Interactions must preserve the constraint structure.

A low energy eigenvalue, a spectral dimension plateau, or a finite rank two tensor packet does not establish a massless spin two particle.

II.15 Conclusion

A complete packet family gives one coordinate language for finite quantum structure. Group characters produce symmetry packets. Charged operators obey exact selection rules. Circuits become coherent sums over packet paths. The canonical moving connection separates geometric transport from physical mixing. Density matrices and channels use two sided packets, while channel covariance becomes block preservation on operator space and symmetry of the Choi matrix.

Compression to a code gives a natural error correction obstruction. Its vanishing is the Knill–Laflamme condition. Diagonalizing the resulting coefficient matrix produces orthogonal syndrome packets and an exact recovery on the error span. Schur–Feshbach elimination gives an equally exact description of hidden Hamiltonian packets. Their frequency response is the self energy, and their time response is a positive definite memory kernel.

Positive Euclidean transfer gives finite reflection positivity. Periodic lattice heat traces converge to the continuum torus at every fixed positive diffusion time, and the continuum small time spectral dimension is recovered after the refinement limit. Constraint closure is measured by a quotient of operator space, not by compression to the physical kernel.

Continuum objective	Finite result available here	Missing continuum input
Symmetry and constraints	exact packets and quotient obstruction	common unbounded domain and renormalized closure
Effective dynamics	exact Feshbach map and uniform gap bounds	cutoff independent spectral separation and locality
Euclidean transfer	positive finite transfer and finite reflection positivity	convergent Euclidean fields and reconstruction axioms
Spectral dimension	lattice heat trace convergence	relation to a physical metric and interacting observables
Lorentzian theory	bounded unitary direct limits	unbounded generator, causality, locality, and field algebra
Einstein regime	a definition of the required expansion	derivation and error control for the effective action
Spin two sector	a precise spectral criterion	positive physical pole and interaction consistency

Table 3. What the finite results establish and what remains outside the article.

The theorems for bounded towers show how exact compatibility and summable differences in operator norm produce limiting quantum evolutions. Their hypotheses also mark the scope of the argument. Unbounded field Hamiltonians, continuum gauge constraints, reflection positive field measures, the Einstein infrared action, and a massless spin two sector each require additional theorems. The finite packet calculus gives a common framework in which those questions can be stated and studied, while leaving their physical resolution open.

Part III Bundle constructions and complementary derivations

A route through this section

The six supplements retain the substantive algebra and geometry of the formal completion. The final supplement constructs local range-bundle charts, their overlaps, and a fiber-valued connection. Three earlier derivations follow because they offer a distinct route through the same finite mathematics.

III.1 Exact packets and arbitrary circuit length

Let R be a unital ring, not necessarily commutative, and let I be a finite address set. An exact packet family consists of elements $P_a \in R$ satisfying

$$P_a P_b = \begin{cases} P_a, & a = b, \\ 0, & a \neq b, \end{cases} \qquad \sum_{a \in I} P_a = 1.$$

Adjoint conditions are added when needed. For an operator U, define the transfer corner

$$U_{ba} = P_b U P_a.$$

The target index comes first. Completeness and distributivity give

$$\sum_{b,a} U_{ba} = U, \qquad (UV)_{ba} = \sum_c U_{bc} V_{ca}.$$

The second identity uses $P_c^2 = P_c$ when adjacent transfer corners are multiplied.

III.1.1 Circuit paths: source Theorem 4.1

For $U = U_m \cdots U_1$, the checked result is

$$P_{a_m} U P_{a_0} = \sum_{a_1, \ldots, a_{m-1}} P_{a_m} U_m P_{a_{m-1}} \cdots P_{a_1} U_1 P_{a_0}.$$

The Lean implementation defines a recursive path sum and proves the identity by induction over an arbitrary finite gate list. It is not restricted to a fixed circuit depth. The empty list has circuit operator 1 and path value $P_b P_a$. Summing the source and target indices reconstructs the full circuit.

III.1.2 Adjoint conservation: source Theorem 4.2

Assume a star operation with $P_a^* = P_a$. Then

$$(U_{ba})^* = P_a U^* P_b.$$

The two checked conservation laws are

$$U^*U = 1 \quad \Longrightarrow \quad \sum_b (U_{ba})^* U_{bc} = \delta_{ac} P_a,$$
$$UU^* = 1 \quad \Longrightarrow \quad \sum_a U_{ba} (U_{da})^* = \delta_{bd} P_b.$$

For the first law, expand each corner, collapse P_b^2 and sum:

$$\sum_b P_a U^* P_b U P_c = P_a U^* \left(\sum_b P_b \right) U P_c = P_a P_c.$$

The second law follows by the analogous calculation and has its own formal proof. For complex matrices, the adjoint is the conjugate transpose. The order of the operator products is preserved throughout.

III.2 Symmetry and measurement

III.2.1 Exact symmetry: source Theorem 5.1

If $HP_a = P_a H$ for every packet, then

$$P_b H P_a = 0 \quad (b \neq a), \qquad H = \sum_a P_a H P_a.$$

In a complex Banach algebra, commutation also propagates to the actual exponential:

$$e^{zH} P_a = P_a e^{zH} \qquad (z \in \mathbb{C}).$$

Consequently,

$$e^{zH} = \sum_a P_a e^{zH} P_a, \qquad P_b e^{zH} P_a = 0 \quad (b \neq a).$$

Taking $z = -it$ gives evolution in real time. The Lean proof applies the exponential commutation theorem directly, so invariance of the packets is a proved consequence of commutation. Self-adjointness of H is not needed for the algebraic reducing property, so this result applies more broadly than the Hermitian specialization in the paper.

This result describes exact symmetry. The quantitative leakage estimate when symmetry is broken appears in Theorem 5.2 and requires the separate Duhamel and norm arguments developed later.

III.2.2 Measurements: source Theorem 7.1

For finite complex matrices, define packet dephasing by

$$\mathcal{D}_P(X) = \sum_a P_a X P_a.$$

The code proves idempotence and trace duality:

$$\mathcal{D}_P(\mathcal{D}_P(X)) = \mathcal{D}_P(X), \qquad \operatorname{tr}(\mathcal{D}_P(E)\rho) = \operatorname{tr}(E\mathcal{D}_P(\rho)).$$

Trace duality follows term by term from cyclicity:

$$\operatorname{tr}(P_a E P_a \rho) = \operatorname{tr}(E P_a \rho P_a).$$

Therefore an effect that is diagonal in the packets, $\mathcal{D}_P(E) = E$, satisfies

$$\boxed{\operatorname{tr}(E\rho) = \operatorname{tr}(E\mathcal{D}_P(\rho)).}$$

The proof holds for arbitrary finite complex matrices E and ρ. Density matrices and positive measurement effects are valid specializations rather than assumptions needed for this trace identity. Taking $E = 1$ also gives trace preservation of packet dephasing.

The full theorem for Kraus channels and the Knill–Laflamme recovery criterion require further arguments about positivity, auxiliary systems, and recovery maps. The trace identity above supplies only the measurement statement.

III.3 Corrected Feshbach resolvent

Write a finite complex matrix in retained and eliminated coordinates:

$$H = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \qquad zI - H = \begin{pmatrix} zI_P - A & -B \\ -C & zI_Q - D \end{pmatrix}.$$

Set

$$D_z = zI_Q - D, \qquad S_z = zI_P - A - BD_z^{-1}C.$$

The effective Hamiltonian is $H_{\text{eff}}(z) = A + BD_z^{-1}C$, so $S_z = zI_P - H_{\text{eff}}(z)$.

III.3.1 Correct statement: source Theorem 9.1

Assume both D_z and $zI - H$ are invertible. Then S_z is invertible and

$$\boxed{\left((zI - H)^{-1}\right)_{PP} = S_z^{-1}.}$$

The left side is the submatrix of the full inverse on the retained coordinates. This matches the paper's compressed operator when written in an orthonormal basis of the retained subspace. The proof is valid for arbitrary complex blocks of compatible sizes and does not require Hermiticity.

The checked invertibility criterion is

$$D_z \text{ invertible} \quad \Longrightarrow \quad (zI - H \text{ invertible} \iff S_z \text{ invertible}).$$

Thus one may equivalently assume invertibility of D_z and S_z.

Why the original assumption is insufficient. Take $z = 0$ and $H = \operatorname{diag}(0, -1)$, retaining the first coordinate. Then

$$D_z = 1, \qquad zI - H = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

The eliminated block is invertible, but the full matrix is singular. Hence the original hypothesis does not ensure the existence of its stated resolvent. The Lean counterexample proves invertibility of the scalar 1 and noninvertibility of this full 2×2 matrix.

III.3.2 Relation to memory and limits

The general block inverse uses the same finite identity as the Schur–Feshbach construction in Chapter VI. Its formal specialization and the counterexample above are included in the Chapter VII source. The identity is read with both stated invertibility assumptions.

The memory equation in the time domain in Section 9.2, resolvent control uniform in the cutoff, and continuum limits require arguments beyond this finite inverse identity. The source amendment makes the two invertibility assumptions explicit without changing the intended effective Hamiltonian.

III.4 Dispersion with an explicit spacing condition

The grid eigenvalue in source Theorem 11.1 is

$$\lambda_{k,h} = \frac{4}{h^2} \sin^2\left(\frac{kh}{2}\right).$$

The original low-mode condition is $0 \le kh/2 \le 1$. A nonzero spacing is also required for the quotient to have its intended meaning. For a physical grid, one normally takes $h > 0$.

III.4.1 Checked stronger statement

For every $k, h \in \mathbb{R}$ with $h \neq 0$,

$$\boxed{0 \le k^2 - \lambda_{k,h} \le \frac{k^4 h^2}{12}.}$$

The code proves this without the low-mode restriction. The bound may be loose away from the low-mode regime, but it remains valid.

The proof begins with an estimate valid for every real argument:

$$0 \le x^2 - \sin^2 x \le \frac{x^4}{3}.$$

The lower bound follows from $|\sin x| \le |x|$. For the upper bound, use

$$|x - \sin x| \le \frac{|x|^3}{6}, \qquad |x + \sin x| \le 2|x|.$$

Factoring the difference of squares gives

$$0 \le x^2 - \sin^2 x = |x - \sin x|\,|x + \sin x| \le \frac{x^4}{3}.$$

With $x = kh/2$ and $h \neq 0$,

$$k^2 - \lambda_{k,h} = \frac{4}{h^2}\left(x^2 - \sin^2 x\right),$$
$$\frac{4}{h^2}\frac{x^4}{3} = \frac{k^4 h^2}{12}.$$

Every algebraic transformation and inequality in the active source was accepted by Lean.

III.4.2 What this does not establish

This is an exact deterministic error bound for the displayed dispersion relation. The numerical convergence experiment and spectral dimension measurements provide finite computational evidence. The bound alone does not select a physical spacetime geometry.

III.5 A constraint diagnostic that vanishes automatically

Section 12 defines the physical projector using the exact common kernel of the constraints. In finite dimensions this gives $C_a P = 0$ for every a. Absorb the factor i and any structure coefficient into f_c, and write

$$\mathcal{A}_{ab} = C_a C_b - C_b C_a - \sum_c f_c C_c.$$

Proposition III.5.1 (Exact common-kernel compression)**.** *If $C_c P = 0$ for every c, then*

$$\mathcal{A}_{ab} P = 0, \qquad P \mathcal{A}_{ab} P = 0.$$

No idempotence or adjoint property of P is needed for this implication.

Proof. Associativity gives

$$\mathcal{A}_{ab} P = C_a(C_b P) - C_b(C_a P) - \sum_c f_c (C_c P) = 0.$$

Left multiplication by P gives the second identity. □

III.5.1 Compressed vanishing without operator closure

Over the rationals, take

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

These displayed matrices are symmetric. The code checks that $P^2 = P$, $P \neq 0$ and $XP = ZP = 0$, but

$$[X, Z] = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{pmatrix} \neq 0, \qquad P[X, Z]P = 0.$$

The commutator therefore violates the closure relation with zero structure coefficients outside the physical range, even though its physical compression vanishes exactly.

III.5.2 Interpretation for the paper

Vanishing of the compressed anomaly is automatic for the exact finite common kernel. It therefore cannot by itself certify full operator closure. A nonzero computed compression would instead contradict those exact assumptions or expose an approximation or implementation defect.

Existence of a nontrivial compatible refinement tower and the broader continuum program remain separate questions. A stronger closure claim must test more than this compression, and unbounded constraints need their operator domains specified. This audit is additional to the six checked source items and is not counted as a seventh original theorem.

III.6 One seed for the computational basis

Manuscript item 2.4. `ComputationalSeed.lean` 14 declarations

For $n \geq 0$, let

$$A_n = (\mathbb{Z}/2\mathbb{Z})^{\{0,\dots,n-1\}}, \qquad V_n = \{v : A_n \to \mathbb{C}\}.$$

For an address $a \in A_n$, define the standard coordinate vector

$$|a\rangle\,(x) = \begin{cases} 1 & x = a, \\ 0 & x \neq a. \end{cases}$$

The seed is $\psi_0 = |0\rangle$. Define a translation operator by

$$(T_a v)(x) = v(x - a).$$

At position j, the local Pauli X is $X_j = T_{e_j}$, where e_j has a single nonzero binary coordinate. The formal circuit uses the identity when $a_j = 0$ and X_j when $a_j = 1$.

Supplementary statement. For every binary address a,

$$W_a = \prod_{j=0}^{n-1} X_j^{a_j} = T_a, \qquad W_a\psi_0 = |a\rangle\,.$$

The family $(W_a\psi_0)_{a\in A_n}$ is linearly independent, spans V_n, and has exactly 2^n elements. It is the standard computational basis.

Proof. Composition gives $T_aT_b = T_{a+b}$, and direct evaluation gives $T_a\,|b\rangle = |a+b\rangle$. Every binary address satisfies $a + a = 0$, so each translation is an involution. In particular, the local operator changes precisely the addressed bit:

$$X_j\,|a\rangle = \big|a \text{ with coordinate } j \text{ replaced by } a_j + 1\big\rangle\,.$$

The selected factor at j equals translation by the address supported at j with value a_j. Induction on a list proves that the product of translations is translation by their sum. Each of these addresses contributes one coordinate, and their sum is a. Thus $W_a = T_a$ and $W_a\psi_0 = |a\rangle$.

Lean identifies the resulting family with `Pi.basisFun` over $\mathbb{C}$. Its standard basis theorems supply linear independence and full span. The finite function-space cardinality gives $|A_n| = 2^n$. □

Formal scope. The proof constructs the sequence of gates that produces each addressed state and then identifies the resulting family with the standard coordinate basis. It includes the empty circuit for $n = 0$. Orthogonality belongs to the usual computational basis interpretation. The formal endpoint records the basis identification, together with separate explicit theorems for span and linear independence.

III.7 Tensor and polyphase refinement

Manuscript item 3.2. `TensorRefinement.lean` 11 declarations

Let F be an arbitrary finite set of factors and let D_i be a finite coordinate index for factor i. For matrix families A_i, define the coordinate tensor product on the product index space by

$$\mathsf{T}(A)(x,y) = \prod_{i\in F} A_i(x_i, y_i).$$

The proof first establishes the laws for multiplication, identity, adjoints, and finite sums. These laws apply to any finite number of factors with the stated finite coordinate sets.

Suppose $N > 0$ and each factor has a complete orthogonal self-adjoint projector family $(P_j^{(i)})_{j\in\mathbb{Z}/N\mathbb{Z}}$. For a fine address $a : F \to \mathbb{Z}/N\mathbb{Z}$, set

$$Q_a = \mathsf{T}(i \mapsto P_{a_i}^{(i)}), \qquad C_r = \sum_{\sum_i a_i = r} Q_a.$$

Supplementary statement. The fine and coarse families are complete orthogonal self-adjoint projector systems. In particular,

$$C_r C_s = \delta_{rs} C_r, \qquad \sum_r C_r = I, \qquad C_r^* = C_r.$$

Each coarse projector is exactly the displayed sum of its fine descendants.

Proof. Expanding the finite product of sums gives

$$\mathsf{T}(A)\mathsf{T}(B) = \mathsf{T}(i \mapsto A_i B_i).$$

Consequently, if two fine addresses differ in any coordinate, the corresponding product contains a zero factor. If the addresses agree, the product is the same fine projector. The adjoint passes through the finite coordinate product, so $Q_a^* = Q_a$. Summing over all address functions and using the same expansion yields

$$\sum_a Q_a = \mathsf{T}\left(i \mapsto \sum_j P_j^{(i)}\right) = I.$$

A separate algebraic regrouping theorem applies to any finite orthogonal family and any address map into a finite set. In the double sum for $C_r C_s$, orthogonality removes all terms with different fine addresses. A surviving address can lie in both classes only when $r = s$. Completeness follows by summing each fine projector exactly once. Adjoint preservation follows term by term. Specializing the address map to total residue proves the claim. □

Formal scope. Residue grouping is proved for every positive modulus and arbitrary finite factor dimensions. It assumes the supplied local projector laws. Constructing those local projectors from cyclic characters, as in item 3.1, requires the separate proof in the next supplement.

III.8 Differentiated moving coordinates

Manuscript item 6.1. `MovingFrames.lean` 9 declarations

Fix a real time t and a finite complex coordinate space. Let $U(s)$ be a unitary matrix curve, differentiable entrywise at t, and let $\psi(s)$ be differentiable componentwise at t. Write $\dot{U}$ and $\dot{\psi}$ for those derivatives. Suppose

$$i\dot{\psi} = H\psi(t).$$

Only the value of the Hamiltonian at the chosen time is needed. No derivative of H is assumed.

Supplementary statement. The explicitly defined coordinate curve

$$y(s) = U(s)^* \psi(s)$$

is differentiable at t, with

$$\dot{y} = \dot{U}^* \psi(t) + U(t)^* \dot{\psi}.$$

It satisfies

$$\boxed{i\dot{y} = H_{\text{eff}} y(t), \qquad H_{\text{eff}} = U(t)^* H U(t) - iU(t)^* \dot{U}.}$$

The connection term $-iU(t)^*\dot{U}$ is Hermitian. If $H^* = H$, then $H_{\text{eff}}^* = H_{\text{eff}}$.

Proof. Applying the product rule entry by entry proves the differentiation formulas for matrix products and for matrices acting on vectors. Since the parameter is real, conjugation commutes with differentiation, giving the stated derivative of U^* and hence of y.

Differentiate $U^*U = I$ using those rules and uniqueness of derivatives:

$$\dot{U}^*U + U^*\dot{U} = 0.$$

Thus $(U^*\dot{U})^* = -U^*\dot{U}$, so multiplication by $-i$ produces a Hermitian matrix.

The identity $UU^* = I$ gives $\psi = Uy$ for the constructed coordinate curve. Differentiating and inserting the original equation yields

$$i\dot{U}y + iU\dot{y} = HUy.$$

Multiplying by U^* and using $U^*U = I$ gives the transformed equation, with the negative sign on the connection term. Finally, conjugation by U preserves Hermiticity of H, and the sum of the two Hermitian terms is Hermitian. □

Formal scope. The main certificate constructs y and its derivative rather than assuming a separate derivative witness for the desired answer. Both unitary identities are explicit. The proofs are pointwise and apply at any time where the stated hypotheses hold. They do not assert existence of a global solution to an evolution equation.

III.9 The canonical packet connection

Manuscript item 6.3. `CanonicalConnection.lean` 9 declarations

Let $P_a(s)$ be a finite family in a normed real algebra with a continuous compatible adjoint. Assume actual derivatives $D_a = \dot{P}_a(t)$ exist and, for all parameter values under consideration,

$$P_aP_b = \delta_{ab}P_a, \qquad \sum_a P_a = I, \qquad P_a^* = P_a.$$

In the supplied Lean theorem, the curves are defined on $\mathbb{R}$ and the projector laws hold at every real parameter. The algebraic formulation applies to matrices and bounded operators whenever they satisfy the stated typeclass hypotheses.

Supplementary statement. Define $C = \sum_a D_aP_a(t)$. Then

$$\boxed{C = \frac{1}{2}\sum_a [D_a, P_a(t)], \qquad C^* = -C,}$$

and, for every b,

$$\boxed{[C, P_b(t)] = D_b.}$$

Proof. The derivative product rule and uniqueness of derivatives give

$$D_aP_b + P_aD_b = \delta_{ab}D_a.$$

Differentiating completeness and adjoint invariance gives

$$\sum_a D_a = 0, \qquad D_a^* = D_a.$$

These are proved derivative consequences, not additional axioms supplied in place of differentiation.

On the diagonal, $D_a P_a + P_a D_a = D_a$. Summing shows

$$\sum_a P_a D_a = -\sum_a D_a P_a = -C.$$

This proves the formula involving half the sum of commutators, together with $C^* = -C$.

Orthogonality directly yields $C P_b = D_b P_b$. For the left product, multiply the differentiated identity for (b, a) on the right by P_a and sum over a. Completeness reduces the result to

$$D_b + P_b C = D_b P_b.$$

Subtracting this equation from the identity for multiplication on the right gives $C P_b - P_b C = D_b$. □

Formal scope. The proof derives the differentiated identities before using their algebraic consequences, so the role of the derivative hypotheses remains explicit. It establishes the canonical connection identity for one parameter. The range bundle over a parameter manifold, the exterior differential calculus, and the curvature theorem of item 6.4 are developed in the later supplements.

III.10 Finite Kraus channels and full amplification

Manuscript item 7.2. `KrausChannels.lean` 7 declarations

For a finite family of complex matrices K_a, define

$$\Phi(\rho) = \sum_a K_a \rho K_a^*.$$

The formalization constructs this as a complex-linear map and proves the following properties directly in the finite matrix representation.

Supplementary statement. The map Φ is completely positive. If

$$\sum_a K_a^* K_a = I,$$

then $\operatorname{tr} \Phi(\rho) = \operatorname{tr} \rho$ for every matrix ρ. In particular, positive semidefinite matrices of trace one are mapped to positive semidefinite matrices of trace one.

Proof. Matrix multiplication distributes across addition and scalar multiplication, which gives complex linearity. If $\rho \succeq 0$, each term $K_a \rho K_a^*$ is positive semidefinite, hence so is their finite sum.

To prove complete positivity rather than positivity alone, let J be an arbitrary finite auxiliary index set. For a matrix X indexed by $J \times n$, define its blockwise amplification by

$$[\Phi^{(J)}(X)]_{(i,u),(j,v)} = \Phi\big(X_{(i,\cdot),(j,\cdot)}\big)_{u,v}.$$

The explicit index calculation proves the identity

$$\boxed{\Phi^{(J)}(X) = \sum_a (I_J \otimes K_a) X (I_J \otimes K_a)^*.}$$

The displayed identity identifies the defined amplification with tensoring by the identity, which is the operation needed for complete positivity. Every summand on the right is positive when X is positive, and this holds for every finite J.

Finally, cyclicity and linearity of the trace give

$$\begin{aligned} \operatorname{tr} \Phi(\rho) &= \sum_a \operatorname{tr}(K_a \rho K_a^*) \\ &= \operatorname{tr}\left(\left(\sum_a K_a^* K_a \right) \rho \right) = \operatorname{tr} \rho. \end{aligned}$$

Positivity together with trace preservation gives the stated conclusion for density matrices. □

Formal scope. Complete positivity is proved for every finite auxiliary dimension. Normalization is required only for trace preservation and the conclusion about density matrices. The converse representation theorem for arbitrary completely positive maps lies outside this argument.

III.11 Actual orbits and sharp seed counts

Let $C \in M_{d\times m}(\mathbb{C})$ be the coefficient matrix of one seed. Define its orbit by actual matrix multiplication:

$$\mathcal{O}(C) = \{AC : A \in M_d(\mathbb{C})\}.$$

This set is already a linear subspace. The source proves that it equals the linear span of the orbit before computing its dimension.

A matrix X belongs to $\mathcal{O}(C)$ exactly when each of its rows belongs to $\operatorname{Row}(C)$. One direction follows by expanding AC. For the converse, choose the coefficients producing each row separately and place them in the corresponding row of A. An explicit linear equivalence therefore gives

$$\mathcal{O}(C) \simeq \operatorname{Row}(C)^d, \qquad \boxed{\dim_{\mathbb{C}} \mathcal{O}(C) = d \operatorname{rank} C.}$$

If $d > 0$, equality with the full $d \times m$ coefficient space is equivalent to $\operatorname{rank} C = m$. Consequently a cyclic seed requires $m \le d$. The assumption that there is at least one row is essential for canceling the factor d.

III.11.1 Several seeds: both necessity and construction

Continue under the assumption above that the number of rows is positive. For seeds $C_1, \dots, C_k$, generation means the concrete statement

$$\forall X \in M_{d\times m}(\mathbb{C}), \quad \exists A_1, \dots, A_k \in M_d(\mathbb{C}) : \quad X = \sum_{a=1}^{k} A_a C_a.$$

Stack the C_a vertically into a $kd \times m$ matrix D. The source identifies the action on the stacked matrix and proves the equivalence

$$\text{the seeds generate} \quad \Longleftrightarrow \quad \operatorname{rank} D = m.$$

Thus $m \le kd$ is necessary. It is sufficient as well: embed the m coordinate columns into kd coordinate positions to obtain a matrix D of full column rank, then split its rows back into k seeds. This construction proves that generating matrices with the required dimensions exist.

III.11.2 Simultaneous generation in a direct sum

For the stipulated block representation

$$\mathcal{A} = \bigoplus_i (M_{d_i}(\mathbb{C}) \otimes I_{m_i}), \qquad d_i > 0,$$

the proof establishes that k seeds generate all blocks simultaneously exactly when $m_i \le kd_i$ for every i. It follows that

$$\boxed{\operatorname{srank}_{\mathcal{A}}(\mathcal{H}) = \max_i \left\lceil \frac{m_i}{d_i} \right\rceil .}$$

The final Lean proposition is an `IsLeast` statement over the set of feasible seed counts. It proves both attainment and minimality. For an empty block family the finite supremum is defined as 0, matching the zero module.

III.12 Schmidt rank and the reduced state

Use coordinates ψ_{ij} in the standard product basis and let $C_{ij} = \psi_{ij}$. An explicit linear equivalence identifies the state vector on the product index set with its coefficient matrix. The implementation checks the actual Kronecker action:

$$\operatorname{coeff}((A \otimes I_B)\psi) = AC.$$

It also proves that the transported matrix orbit equals the span of the state vectors obtained by this Kronecker action. The dimension formula therefore applies to the physical local action on the state space.

III.12.1 An independent definition of Schmidt rank

The source defines Schmidt rank as the least number of summands that are product vectors. A decomposition of length s is the concrete equality

$$C_{ij} = \sum_{a=1}^{s} u_a(i) v_a(j).$$

It factors $C = UV$ through an s-dimensional intermediate space, so $\operatorname{rank} C \le s$. Conversely, choose a basis of the row space of C. Expanding each row in this basis constructs a decomposition with exactly $\operatorname{rank} C$ product terms. Thus

$$\boxed{r(\psi) = \operatorname{rank} C.}$$

The definition uses a least natural number, and the equality with matrix rank is a theorem. This argument establishes the minimum number of product terms. It does not construct an orthonormal Schmidt decomposition or prove a new theorem on singular value decomposition.

The matrix-orbit dimension and the coefficient equivalence now yield

$$\boxed{\dim_{\mathbb{C}} \operatorname{span}\{(A \otimes I_B)\psi : A \in M_{d_A}(\mathbb{C})\} = d_A r(\psi).}$$

III.12.2 The actual partial trace and its support

The reduced matrix is defined by tracing the addressed index of the density matrix of the pure state:

$$(\rho_B)_{j\ell} = \sum_i \psi_{ij} \overline{\psi_{i\ell}}.$$

For the coefficient convention used here, the exact Gram identity is

$$\rho_B = C^{\mathsf{T}} (C^{\mathsf{T}})^*, \qquad \operatorname{rank} \rho_B = \operatorname{rank} C = r(\psi).$$

The transpose is important. With this convention, writing C^*C instead would conjugate the reduced matrix. The Lean source checks the entrywise identity and positivity of the resulting matrix.

Full support is expressed as surjectivity of the reduced matrix's linear map. Assuming $d_A > 0$, the checked equivalence is

$$\boxed{\mathcal{O}_A(\psi) = \mathcal{H}_A \otimes \mathcal{H}_B \quad \Longleftrightarrow \quad \operatorname{Ran} \rho_B = \mathcal{H}_B.}$$

Normalization of ψ is unnecessary for these dimension and support statements. The zero state has rank 0 because its decomposition needs no product terms.

III.13 Constructed cyclic character projectors

Let $N > 0$ and let S be a unitary complex matrix with $S^N = I$. The implementation uses the standard additive character of $\mathbb{Z}/N\mathbb{Z}$ and proves its expression through a root of unity:

$$\omega = \exp(2\pi i/N), \qquad \chi(r) = \omega^{\mathrm{val}(r)}.$$

Here $\mathrm{val}(r)$ is the canonical representative in $\{0, \dots, N-1\}$. The representation is constructed as

$$\rho(r) = S^{\mathrm{val}(r)}.$$

Reduction of exponents modulo N is proved from $S^N = I$, giving $\rho(r+s) = \rho(r)\rho(s)$ and $\rho(0) = I$. The final matrix theorem derives the representation law and character orthogonality from the stated inputs.

III.13.1 Exact coefficient formula

The packet definition is a finite sum over residue classes:

$$P_j = \frac{1}{N} \sum_{r \in \mathbb{Z}/N\mathbb{Z}} \chi(-rj)\rho(r).$$

The source proves

$$\chi(-rj) = \left(\omega^{\mathrm{val}(r)\,\mathrm{val}(j)}\right)^{-1},$$

so this is precisely the manuscript's N-term Fourier projector formula, with canonical representatives.

III.13.2 Algebraic identities

Primitivity of the standard character yields the checked finite sum

$$\sum_r \chi(tr) = \begin{cases} N, & t = 0, \\ 0, & t \neq 0. \end{cases}$$

Reindexing finite sums then gives

$$\boxed{P_j P_k = \delta_{jk} P_j, \qquad \sum_j P_j = I, \qquad S P_j = \omega^j P_j.}$$

Multiplicativity and completeness are first proved for a general representation in a complex algebra, then instantiated with the actual powers of S. This organization makes the algebra of finite sums reusable while preserving the final hypotheses.

III.13.3 Orthogonality of the projectors

Unitarity supplies $\rho(r)^* = \rho(-r)$, while the character obeys $\overline{\chi(r)} = \chi(-r)$. Reindexing $r \mapsto -r$ proves $P_j^* = P_j$. The final concrete matrix theorem includes Hermiticity together with multiplication, completeness, and eigenaction. Together these identities establish orthogonal spectral packets.

The proof includes the degenerate modulus $N = 1$: finite order then forces $S = I$ and the sole projector is I. It does not need an artificial assumption $N > 1$.

III.14 The actual Euclidean transfer exponential

Let H be a Hermitian matrix on a nonzero complex state space of finite dimension. The source defines

$$T_e(t) = \exp(-t(H - eI))$$

using Lean's exponential defined by a power series. The proof then relates this defined operator to its spectral expansion.

III.14.1 Connecting power series and spectral calculus

The shifted generator is proved Hermitian. The implementation then identifies the actual matrix exponential with continuous functional calculus:

$$\boxed{T_e(t) = f_t(H), \qquad f_t(x) = e^{-t(x-e)}.}$$

The formal proof uses composition, scalar multiplication, subtraction, and the established agreement between the exponential in functional calculus and the exponential defined by a power series. Spectral mapping is consequently available for the operator that was actually defined:

$$\mathrm{Spec}_{\mathbb{R}}\, T_e(t) = \{e^{-t(x-e)} : x \in \mathrm{Spec}_{\mathbb{R}}\, H\}.$$

Hermiticity also identifies the full complex spectrum with the image of these real eigenvalues. A separate checked theorem states the interval containing the complex spectrum explicitly.

III.14.2 The minimum eigenvalue

The ground energy is the minimum of the finite, nonempty eigenvalue set. The proof establishes both

$$E_0 \in \mathrm{Spec}_{\mathbb{R}}\, H, \qquad E_0 \le x \quad (x \in \mathrm{Spec}_{\mathbb{R}}\, H).$$

The assumption that the state space is nonzero is recorded in the theorem because the minimum of an empty spectrum is not defined by the manuscript's formula.

III.14.3 Positivity, semigroup composition, and the interval

Every eigenvalue of $T_e(t)$ is strictly positive, for arbitrary real e, t, since the real exponential is positive. For $e = E_0$ and $t \geq 0$ it is also at most one. Therefore

$$\boxed{T_{E_0}(t) > 0, \qquad \operatorname{Spec} T_{E_0}(t) \subseteq (0, 1].}$$

The two scalar multiples of $H - eI$ commute, so the exponential defined by the power series obeys

$$\boxed{T_e(t+s) = T_e(t)T_e(s), \qquad T_e(0) = I.}$$

The semigroup identity is proved for all real t, s. Nonnegative time is needed for the upper spectral bound, not for the composition identity or strict positivity.

These results concern operators on a finite dimensional space. Continuum transfer limits, gravitational path integrals and physical spacetime models require further structure. The numerical figures provide separate computational illustrations.

III.15 Quantitative leakage from the actual exponential

Let A be a unital complex C-star algebra, let $H = H^*$, and let $P = P^* = P^2$. Put $Q = 1 - P$. The checked statement is

$$\boxed{\|Q \exp(-itH)P\| \leq |t| \, \|HP - PH\| \qquad (t \in \mathbb{R}).}$$

The matrix specialization uses the operator norm on the Hilbert space, implemented by `Matrix.Norms.L2Operator`. It is not an entrywise norm or the row-sum operator norm. The theorem includes the zero projector and empty matrix index type. The theorem also applies at time zero.

III.15.1 Deriving the estimate by differentiation

For a skew-adjoint generator G, define the actual conjugation path

$$T_P(t) = \exp(tG)P\exp(-tG).$$

The formalization derives its derivative from the exponential power series and the product rule:

$$T_P'(t) = \exp(tG)(GP - PG)\exp(-tG).$$

Both exponential factors are proved unitary. Left and right multiplication by a unitary preserve the C-star norm. Therefore

$$\|T_P'(t)\| = \|GP - PG\|, \qquad \|T_P(t) - T_P(s)\| \leq |t - s| \, \|GP - PG\|.$$

The second inequality follows from the mean value inequality in normed spaces and covers either ordering of s and t. It is not a finite-difference estimate or a numerical time discretization.

Multiplication by $\exp(tG)$ gives

$$(T_P(t) - P)\exp(tG) = \exp(tG)P - P\exp(tG).$$

For every orthogonal projector, $\|P\| \le 1$ and $\|1 - P\| \le 1$. The exact corner identity

$$(1 - P)UP = (1 - P)(UP - PU)P$$

then bounds leakage by the exponential commutator. Taking $G = -iH$ gives the stated theorem because $\|[G, P]\| = \|[H, P]\|$.

III.15.2 Checked consequences

Unitarity also gives the capped estimate

$$\|Q \exp(-itH)P\| \le \min\{1,\ |t|\, \|[H, P]\|\}.$$

If $H = H_0 + V$, both summands are self-adjoint, and $[H_0, P] = 0$, then

$$\|Q \exp(-it(H_0 + V))P\| \le 2|t|\, \|V\|.$$

The latter uses $\|[V, P]\| \le 2\|V\|$. Its constant is not claimed to be optimal. Exact commutation gives zero leakage. The work does not include unbounded generators or time-dependent Hamiltonians.

III.16 Actual projected differentiation and curvature

Let E, F be real normed spaces. Let $P : E \to \mathcal{L}(F, F)$ be differentiable with $P(x)^2 = P(x)$ for every x. Let $s : E \to F$ be C^2 with $P(x)s(x) = s(x)$, and let $X, Y : E \to E$ be differentiable vector fields. The checked connection is the actual expression

$$\nabla_X s(x) = P(x)\, Ds(x)[X(x)].$$

It is proved to preserve the range condition. It is linear in the vector field direction and obeys the scalar Leibniz rule

$$\nabla_X(cs) = Dc[X]\, s + c\nabla_X s$$

for differentiable real scalar functions c. No differentiated projector identities are inserted as hypotheses in the main curvature theorem.

III.16.1 Differentiated projector and section identities

Differentiating $P^2 = P$ and $Ps = s$ produces

$$\begin{aligned} DP[v]P + PDP[v] &= DP[v], \\ PDP[v]P &= 0, \\ Ds[v] &= DP[v]s + PDs[v]. \end{aligned}$$

The zero sandwich follows algebraically from the first identity. All products preserve their noncommutative order.

III.16.2 Retaining the bracket of vector fields

Define the actual curvature action by

$$R_P(X,Y)s = \nabla_X \nabla_Y s - \nabla_Y \nabla_X s - \nabla_{[X,Y]} s.$$

The source uses the Lie bracket defined through Fréchet derivatives. Symmetry of second derivatives is supplied by the C^2 hypothesis, not postulated separately. Expanding the connection gives

$$R_P(X,Y)s = P(DP[X]Ds[Y] - DP[Y]Ds[X]).$$

Substituting the section identity and removing the zero sandwich terms proves

$$\boxed{R_P(X,Y)s = P(DP[X]DP[Y] - DP[Y]DP[X])Ps.}$$

Thus the coefficient is precisely $P(\,\mathrm{d}P \wedge \mathrm{d}P)P$ with the convention $(\,\mathrm{d}P \wedge \mathrm{d}P)(v,w) = DP[v]DP[w] - DP[w]DP[v]$.

III.16.3 Tensor properties and the remaining global boundary

The coefficient is proved bilinear, alternating, and supported on the range on both sides. The chain rule proves the pullback identity

$$F_{P\circ f}(x)(v,w) = F_P(f(x))(Df(x)v, Df(x)w).$$

The main theorem has global regularity hypotheses on normed parameter spaces. The pullback identity alone does not supply localization, smooth local trivializations, or gluing. The final supplement constructs the global range bundle and proves that the local connections glue.

III.17 A constructive and total syndrome recovery

Let P be an orthogonal code projector. The input certificate consists of a finite family V_a satisfying

$$V_a P = V_a, \qquad V_a^* V_b = \delta_{ab} P.$$

These hypotheses state that the operators are partial isometries with orthogonal syndromes. The next supplement derives such a family from arbitrary Knill–Laflamme Gram data.

III.17.1 Constructing the recovery on the whole ambient space

Define the syndrome projectors and their complement by

$$S_a = V_a V_a^*, \qquad S = \sum_a S_a, \qquad Q = 1 - S.$$

The source proves self-adjointness, $S_a S_b = \delta_{ab} S_a$, $S^2 = S$, and $Q^2 = Q = Q^*$. It also proves $QV_a = 0$ and $V_a^* Q = 0$.

The recovery is not left undefined outside the syndrome subspaces. Its explicit Kraus operators are $R_a = V_a^*$ together with $R_{\text{extra}} = Q$, giving

$$\mathcal{R}(\rho) = \sum_a V_a^* \rho V_a + Q\rho Q.$$

They satisfy exact completeness:

$$\sum_a R_a^* R_a + R_{\text{extra}}^* R_{\text{extra}} = \sum_a V_a V_a^* + Q = S + Q = I.$$

For finite complex matrices, trace preservation and complete positivity at every finite auxiliary dimension are proved for this actual map. The amplified map is identified with the Kraus family $I_J \otimes R_a$, including the complement term. Positivity is not merely checked on a single matrix size.

III.17.2 Exact correction, including coherent errors

For every operator supported on the code, $P\rho P = \rho$, the central identity is

$$\boxed{\mathcal{R}(V_a \rho V_b^*) = \delta_{ab}\rho.}$$

Consequently normalized mixtures satisfy

$$\mathcal{R}\left(\sum_a p_a V_a \rho V_a^*\right) = \rho \quad \text{when} \quad \sum_a p_a = 1.$$

The identity is algebraic and is proved for complex weights. Physical mixtures use the nonnegative real case. For a coherent error $E_z = \sum_a z_a V_a$, the calculation of the cross terms yields

$$\mathcal{R}(E_z \rho E_z^*) = \left(\sum_a z_a \overline{z_a}\right) \rho.$$

Thus the same map also corrects normalized coherent superpositions. This construction supplies the recovery stage used in the full Knill–Laflamme proof that follows.

III.18 The precise finite channel statement

Let $\mathcal{H} = \mathbb{C}^n$ be a finite coordinate space and let $P = P^* = P^2$. A matrix is code supported when $P\rho P = \rho$. For a finite family of square matrices E_a, put

$$\mathcal{N}_E(\rho) = \sum_a E_a \rho E_a^*.$$

The input channel normalization is the actual matrix identity

$$\sum_a E_a^* E_a = I.$$

A finite Kraus recovery has the form $\mathcal{R}(\sigma) = \sum_r R_r \sigma R_r^*$, with $\sum_r R_r^* R_r = I$. This representation, together with the checked amplification results, provides complete positivity and trace preservation on the entire ambient matrix space.

III.18.1 Necessary and sufficient conditions

Under the stated normalization, the following equivalence is proved:

$$\boxed{\begin{array}{c}\exists\, \mathcal{R} \text{ of the above finite Kraus form,} \\ \mathcal{R}\mathcal{N}_E(\rho) = \rho \quad \text{for every } \rho \succeq 0,\ \operatorname{Tr}\rho = 1,\ P\rho P = \rho\end{array}} \iff \boxed{\begin{array}{c}\exists C \succeq 0, \\ PE_a^* E_b P = C_{ab} P \quad \text{for all } a, b.\end{array}}$$

The statement applies throughout the code space. The formalization separately proves equivalence with correction of *every* complex matrix supported on the code. It also proves preservation with arbitrary finite reference systems, as detailed in the discussion of density matrices and reference systems below.

III.18.2 Including degenerate cases

The matrix C need not be positive definite. The proof includes zero eigenvalues, redundant error operators, an empty set of positive eigenvalues, and the zero projector. If $P = 0$, the only supported matrix is zero and no trace-one code density matrix exists. The proofs handle that case separately rather than dividing by the code rank or by $\operatorname{Tr} P$.

For an unnormalized error family, the correct statement is generally

$$\mathcal{R}\mathcal{N}_E(\rho) = \operatorname{Tr}(C)\rho,$$

not unconditional equality to ρ. Normalization forces $\operatorname{Tr} C = 1$ on a nonzero code. The discussion of coherent errors below treats unrestricted error families and shows when the same decoder works throughout their span.

III.19 From Gram data to an actual total recovery

Assume $C \succeq 0$ and $PE_a^* E_b P = C_{ab} P$. First restrict the error operators to the code, $F_a = E_a P$. The finite Hermitian spectral theorem supplies a unitary U such that

$$U^* C U = \operatorname{diag}(\lambda_r), \qquad \lambda_r \geq 0.$$

Define the transformed errors by an actual finite sum,

$$B_r = \sum_a U_{ar} F_a.$$

The formal module proves the identities

$$B_r P = B_r, \qquad B_r^* B_s = \delta_{rs} \lambda_r P, \qquad \sum_r B_r \rho B_r^* = \sum_a F_a \rho F_a^*.$$

For ρ supported on the code, the restricted and unrestricted error sums are equal. Thus the unitary change of error coordinates preserves the actual channel on the code.

III.19.1 Null directions are removed by proof

If $\lambda_r = 0$, then $B_r^* B_r = 0$. Positivity of the ordinary complex matrix inner product gives $B_r = 0$. This is the checked declaration `DiagonalErrors.zero_weight_error`. The normalized syndrome operators are defined only on

$$A_+ = \{r : \lambda_r > 0\}, \qquad V_r = \lambda_r^{-1/2} B_r.$$

Consequently no inverse square root at a null eigenvalue is needed. Lean proves

$$V_r P = V_r, \qquad V_r^* V_s = \delta_{rs} P.$$

The recovery from the previous block is instantiated from these derived syndromes, rather than from an assumed orthogonal error family.

III.19.2 A channel on the whole ambient space

Put $Q = I - \sum_{r \in A_+} V_r V_r^*$. The recovery is

$$\boxed{\mathcal{R}(\sigma) = \sum_{r \in A_+} V_r^* \sigma V_r + Q \sigma Q.}$$

The complement is an orthogonal projector and

$$\sum_{r \in A_+} V_r V_r^* + Q^* Q = I.$$

Hence the recovery is trace preserving. Its finite Kraus form also proves positivity after amplification by the identity on every finite auxiliary space. The complement term matters: the map is defined as a channel even outside the syndrome ranges.

Finally the checked calculation gives $\mathcal{R}\mathcal{N}_E(\rho) = \sum_r \lambda_r \rho = \mathrm{Tr}(C)\rho$. Under normalization of the input channel and $P \neq 0$, summing the diagonal Gram identities gives $\mathrm{Tr}(C)P = P$, hence $\mathrm{Tr}\, C = 1$. The proof for the zero code does not use this cancellation.

III.20 Deriving the Gram matrix from recovery

Suppose a finite recovery that preserves trace corrects every operator supported on the code. To derive the Knill–Laflamme condition without assuming a syndrome decomposition, the proof expands the composite channel into the finite Kraus family

$$A_{ra} = R_r E_a P.$$

For every ambient matrix ρ, correction on the code implies

$$\sum_{r,a} A_{ra} \rho A_{ra}^* = P \rho P.$$

This equality on all matrices follows by applying correction to the sandwich supported on the code, $P\rho P$, using the projector identities.

III.20.1 Rank-one rigidity of the corrected channel

The module tests the map equality on actual matrix units and flattens each A_{ra} into a row. If X is the matrix of those rows and p is the vector of entries of P, the resulting Gram identity is

$$X^*X = \overline{p}\,p^{\mathsf{T}}.$$

The checked rank-one Gram lemma proves that every row of X is proportional to p. For nonzero p, a nonzero coordinate is selected and a test vector supported on two coordinates is placed in the null space of the Gram matrix. The implication $X^*Xv = 0 \Rightarrow Xv = 0$ then gives the required proportionality. When $p = 0$, the Gram identity forces $X = 0$ directly.

Applied to the corrected Kraus family, this yields scalars z_{ra} with

$$R_r E_a P = z_{ra} P.$$

The essential declarations are `rank_one_gram_rigidity`, `kraus_rank_one_rigidity` and `corrected_kraus_family`. No rank-one Choi characterization is inserted as an axiom or as the desired conclusion in hypothesis form.

III.20.2 Trace-preserving recovery supplies the original Gram data

Let $Z_{ra} = z_{ra}$. The recovery normalization gives

$$\begin{aligned} PE_a^* E_b P &= PE_a^* \Big(\sum_r R_r^* R_r\Big) E_b P \\ &= \sum_r (R_r E_a P)^* (R_r E_b P) \\ &= \Big(\sum_r \overline{z_{ra}} z_{rb}\Big) P = (Z^*Z)_{ab} P. \end{aligned}$$

Thus $C = Z^*Z$ is positive semidefinite and satisfies the required compressed identities. The theorem `gram_of_recovery` implements this calculation with the original error and recovery matrices.

A second rigidity argument handles a nonnegative scalar multiple of the identity code map. It uses $\sqrt{t}\,P$ as the rank-one map generator and includes $t = 0$. That argument proves necessity for unnormalized error families, which need not define channels.

III.21 Density matrices, all operators and finite references

The necessity argument is stated for all operators, so it must be connected to correction of physical density matrices. The proof establishes this connection explicitly for complex linear maps on finite matrices.

III.21.1 Normalization is not an extra assumption

A nonzero positive semidefinite matrix has strictly positive real trace. If a complex-linear map fixes every normalized code density matrix, apply that hypothesis to

$$\frac{\rho}{\operatorname{Tr}\rho}$$

for each nonzero positive code-supported ρ. Linearity then gives equality on ρ itself. The zero matrix is handled separately. The source proves that the trace is real and nonzero, then justifies cancellation of the scalar. These steps establish that normalization is valid.

III.21.2 Complex polarization reaches the whole operator space

Write $q(x) = xx^*$. The module proves the exact identity

$$4xy^* = q(x+y) - q(x-y) + i(q(x+iy) - q(x-iy)).$$

Agreement on all positive pure outer products therefore implies agreement on every outer product. Actual matrix units then span the full matrix space. Applying this argument to the sandwich by P proves equality on every code-supported operator, including non-Hermitian ones.

The result `density_corrects_iff_operators` establishes the equivalence for the actual composite Kraus maps. It does not replace the set of density matrices by an assumed basis of logical states.

III.21.3 An arbitrary finite reference system

For any finite reference index set A, the amplified channel is defined using the actual Kronecker Kraus operators,

$$\widetilde{\mathcal{N}}_E(\rho) = \sum_a (I_A \otimes E_a)\rho(I_A \otimes E_a)^*.$$

Its reference blocks are proved equal to the original channel applied blockwise. If

$$(I_A \otimes P)\rho(I_A \otimes P) = \rho,$$

then every reference block is code supported. Applying the checked correction theorem for operators to those blocks gives

$$\boxed{\widetilde{\mathcal{R}}\,\widetilde{\mathcal{N}}_E(\rho) = \rho.}$$

This includes entangled density matrices supported on the combined reference and code space. In fact, the identity holds for every jointly supported matrix and every finite reference dimension.

The declaration `recovery_with_reference` proves the equality. The finite Kraus amplification separately establishes complete positivity of the recovery, which is needed to preserve states in the presence of a reference system.

III.22 Coherent combinations and normalization

A correction proof for only one selected Kraus presentation is weaker than a proof that one decoder works throughout the error span. The formal module connects the constructed spectral recovery back to the original error operators.

III.22.1 Cross terms for the original errors

For arbitrary positive Gram data, the same recovery satisfies

$$\boxed{\mathcal{R}(E_a \rho E_b^*) = C_{ba}\rho \qquad (P\rho P = \rho).}$$

The index order ba is intentional: the defining Gram relation is $PE_a^*E_bP = C_{ab}P$. The proof first derives scalar corrected Kraus composites for the constructed recovery and then uses its trace-preserving completeness to identify the original Gram entries.

For $E(c) = \sum_a c_a E_a$, finite-sum expansion gives

$$\mathcal{R}(E(c)\rho E(c)^*) = \left(\sum_{a,b} c_a \overline{c_b} C_{ba}\right)\rho = (c^* C c)\rho.$$

The formula with a double sum is the literal checked conclusion. The final notation as a quadratic form follows by renaming finite indices. It is not restricted to orthogonal errors or real coefficients.

III.22.2 Every normalized channel in the same span

Suppose $F_j = \sum_a c_{ja} E_a$ and $\sum_j F_j^* F_j = I$. The decoder is *not* reconstructed for the F_j. Summing the coherent identities gives a scalar multiple $q\rho$. Trace preservation of both the F channel and the recovery forces $q = 1$ on a nonzero code. The proof uses positivity of P to justify cancellation of its trace and handles $P = 0$ separately. Therefore

$$\boxed{\mathcal{R}\Big(\sum_j F_j \rho F_j^*\Big) = \rho.}$$

This is `Data.recovery_span_channel`, a quantified statement over arbitrary finite F families and their complex coefficients.

III.22.3 The equivalence for unnormalized error families

For arbitrary finite E, define exact family recovery to mean that a finite trace-preserving Kraus recovery and a single $t \geq 0$ exist such that

$$\mathcal{R}\mathcal{N}_E(\rho) = t\rho \quad \text{for every code-supported } \rho.$$

Then `error_family_correctable_iff_gram` proves that this condition is equivalent to the existence of $C \succeq 0$ satisfying the Gram identities. Sufficiency chooses $t = \operatorname{Tr} C$. Necessity constructs C from the scaled rank-one rigidity argument.

The scale is essential. Zero errors can give $t = 0$ and cannot be inverted to a nonzero normalized state. An input with a larger total trace is not silently declared a channel. For normalized input channels, the separate channel theorem removes the scale exactly as described in the sufficiency argument above.

III.23 Exact examples, trust audit and remaining work

III.23.1 Singular, nonorthogonal errors are accepted

On a qubit with full code $P = I$, take

$$E_0 = \frac{3}{5} I, \qquad E_1 = \frac{4}{5} I, \qquad C = \begin{pmatrix} 9/25 & 12/25 \\ 12/25 & 16/25 \end{pmatrix}.$$

Lean checks the channel normalization, the Gram identities, positivity of C, $\det C = 0$, and the nonzero off-diagonal entry. It then applies the criterion for density matrices to prove correctability. The calculation is exact and uses rational complex entries.

III.23.2 A genuine channel can still be uncorrectable

For full qubit dephasing, $E_0 = \operatorname{diag}(1, 0)$ and $E_1 = \operatorname{diag}(0, 1)$, the Kraus normalization is checked. But $E_0^* E_0$ cannot be a scalar multiple of I, as its two diagonal entries differ. Necessity therefore proves that no finite recovery channel corrects the full code space. The case of the zero code is also checked independently.

III.24 The global projected connection

Let M be a smooth real manifold without boundary, F a complete normed real vector space and $P : M \to \operatorname{End}(F)$ a smooth family of bounded idempotents. For smooth range sections s, define

$$\nabla_X s = P \, \mathrm{d}s[X].$$

The proof begins by constructing the range bundle from P. Its local connection formulas glue, and the actual covariant commutator satisfies

$$\boxed{(\nabla_X \nabla_Y - \nabla_Y \nabla_X - \nabla_{[X,Y]}) s = P(\, \mathrm{d}P[X] \, \mathrm{d}P[Y] - \, \mathrm{d}P[Y] \, \mathrm{d}P[X]) P \, s.}$$

The derivative and Lie bracket are Mathlib's actual manifold operators. No global coordinate chart is assumed. The result requires idempotence, not an additional orthogonality hypothesis.

III.25 Local products constructed from the projector

The fiber over x is the actual fixed subspace

$$\mathcal{E}_x = \{v \in F : P(x)v = v\} = \operatorname{Ran} P(x).$$

The total space is the subset $\mathcal{E} = \{(x, v) : P(x)v = v\} \subseteq M \times F$, with the induced topology. The construction provides local homeomorphisms of this total space as well as the transition functions between them.

III.25.1 An invertible intertwiner near each anchor

For idempotents p, q, set

$$T(p, q) = pq + (I - p)(I - q).$$

Lean proves $T(p, p) = I$ and $pT(p, q) = T(p, q)q$. For an anchor $a \in M$, define

$$U_a = \{x : T(P(x), P(a)) \text{ is invertible}\}.$$

The invertible elements form an open subset of the algebra of bounded operators. Consequently U_a is open and contains a. The family (U_a) covers the whole base. Invertibility is *derived locally*, not assumed at every pair of points.

III.25.2 The mutually inverse local maps

Write $T_a(x) = T(P(x), P(a))$. For $x \in U_a$,

$$B_a(x) : \mathcal{E}_a \longrightarrow \mathcal{E}_x, \quad v \longmapsto T_a(x)v,$$

$$A_a(x) : \mathcal{E}_x \longrightarrow \mathcal{E}_a, \quad w \longmapsto P(a)T_a(x)^{-1}w.$$

The intertwining identity and its inverse counterpart show that the displayed targets are correct. Lean proves both inverse laws, continuity, linearity in the fiber and smooth dependence on the base. Thus

$$\mathcal{E}|_{U_a} \cong U_a \times \mathcal{E}_a$$

is a local product homeomorphism that preserves the base point.

III.25.3 Smooth overlap maps and rank

The coordinate transition is $g_{ab}(x) = A_b(x)B_a(x)$. It is smooth on $U_a \cap U_b$, agrees with the actual change between local homeomorphisms, and satisfies

$$g_{aa} = I, \qquad g_{bc}g_{ab} = g_{ac}.$$

A continuous linear equivalence between $\mathcal{E}_a$ and $\mathcal{E}_x$ also gives local constancy of fiber dimension. No single constant rank across disconnected components is assumed.

III.26 A connection on the constructed bundle

The global formula is defined using the actual manifold derivative:

$$\nabla_X s(x) = P(x)\, D_M s(x)[X(x)].$$

Here D_M is represented by `mvfderiv`. A second definition, `GlobalConnection.covariant`, has values in the actual fiber:

$$\nabla s(x) : T_x M \longrightarrow \mathcal{E}_x.$$

It is not only an ambient expression asserted to land in the range.

III.26.1 Connection axioms and smoothness

The formal proofs establish range preservation, additivity in the section, linearity over functions in the vector field argument, and the scalar Leibniz identity

$$\nabla_X(fs) = \mathrm{d}f[X]\, s + f \nabla_X s.$$

Smooth P, smooth sections and smooth vector fields give smooth covariant derivatives. Equality of section germs gives equality of covariant derivatives at the point, so the operation is local.

III.26.2 Smoothness of sections in bundle coordinates

Let $x \in U_a$. If $P(y)s(y) = s(y)$ in a neighborhood of x, then

$$s \text{ is smooth at } x \quad \Longleftrightarrow \quad (y \mapsto A_a(y)s(y)) \text{ is smooth at } x.$$

The range condition is only local. The proof uses the actual inverse local maps and equality of germs. It follows that smoothness in bundle coordinates does not depend on the chosen chart.

III.26.3 The local connection and its change of coordinates

For a coordinate function u valued in $\mathcal{E}_a$, define

$$\nabla^{(a)}_X u = A_a(P\, D_M(B_a u)[X]).$$

For every overlap point, Lean proves the operator identity

$$\boxed{\nabla^{(b)}_X(g_{ab}u) = g_{ab}\nabla^{(a)}_X u.}$$

This is proved for arbitrary local coordinate functions. It does not assume that u was obtained from a globally smooth section or that a smooth extension has been supplied. Together with the local smoothness criterion, this closes the gluing step left open in the fifth block.

III.27 From differentiated projectors to the two-form

The curvature is defined by the covariant commutator, including the *actual manifold Lie bracket*. It is not defined to equal the desired projector expression.

III.27.1 The flat ambient identity is proved in charts

For a smooth vector-valued function s, the proof identifies directional derivatives with the pullback into each extended chart. It then proves

$$D_M s([X,Y]) = D_M(D_M s[Y])[X] - D_M(D_M s[X])[Y].$$

This uses the chart pullback of the actual vector fields and the ordinary differential identity, then transfers the result back to the manifold. The auxiliary module works locally at a point, so the argument applies within arbitrary open chart domains.

III.27.2 Projector and section identities are derived

Differentiating $P^2 = P$ in a tangent direction v yields

$$(\,\mathrm{d}P[v])P + P\,\mathrm{d}P[v] = \,\mathrm{d}P[v], \qquad P\,\mathrm{d}P[v]P = 0.$$

Differentiating $Ps = s$ gives

$$\mathrm{d}s[v] = \,\mathrm{d}P[v]s + P\,\mathrm{d}s[v].$$

Both statements are consequences of actual derivative rules. They are not additional assumptions.

III.27.3 Cancellation of the second derivatives

The product rule and the identity for the ambient Lie bracket reduce curvature to

$$R^{\nabla}(X,Y)s = P\,\mathrm{d}P[X]\,\mathrm{d}s[Y] - P\,\mathrm{d}P[Y]\,\mathrm{d}s[X].$$

Substitute the identity obtained by differentiating the range condition for the section. The terms containing $P\,\mathrm{d}P[v]P$ vanish, leaving

$$R^{\nabla}(X,Y)s = P(\,\mathrm{d}P[X]\,\mathrm{d}P[Y] - \,\mathrm{d}P[Y]\,\mathrm{d}P[X])Ps.$$

Lean also proves that this depends only on $X(x)$, $Y(x)$ and $s(x)$, not on their extensions. The operator-valued form is alternating, additive and homogeneous, and lands in the actual fiber. Constant projector bundles have zero curvature form.

III.27.4 Geometry and scalar conventions

The main global curvature theorem uses a smooth real manifold without boundary, modeled on a complete normed real space, and bounded real-linear endomorphisms of a complete normed real space. Finite-dimensional parameter and quantum spaces fall within the corresponding real differential calculus. Complex quantum spaces can be viewed through their underlying real normed spaces. This block does not introduce a separate typeclass wrapper for complex bundles.

The atlas and global connection constructions also support the more general infrastructure for models with corners wherever their hypotheses apply. The *curvature completion claim* concerns the stated theorem for manifolds without boundary. Extending that curvature result to boundaries or corners requires further work.

III.27.5 An explicit smooth atlas

The implementation exposes the actual total space, local product homeomorphisms, linear transitions, smoothness proofs and cocycle laws. It does not register these as a `ContMDiffVectorBundle` instance. This is an interface choice. The range bundle atlas, local trivializations, overlap identities, and curvature formula are all constructed or proved from the stated inputs.

The equivalence for smooth sections and the gauge law make the atlas usable intrinsically. The connection is also defined with values in `fiber(Px)`. Six constructed objects that contain proof fields, including both local equivalences and the derivative with values in the fiber, have separate axiom audits.

III.28 Earlier compact derivations retained as alternatives

The following derivations retain valid mathematical arguments from the original manuscript. Their hypotheses and notation agree with the corrected finite core. The earlier abbreviated argument for error correction is replaced by the complete normalized and scaled recovery proofs above, and its substantive conclusions are included there.

III.28.1 Duhamel proof of leakage from a fixed packet

Let $H = H^*$ on a finite-dimensional complex Hilbert space, $P = P^* = P^2$, $Q = I - P$, and $t \in \mathbb{R}$.

Alternative proof of the commutator leakage bound. Because $QP = 0$,

$$Qe^{-itH}P = Q[e^{-itH}, P]P.$$

The Duhamel identity gives

$$[e^{-itH}, P] = -i \int_0^t e^{-i(t-s)H}[H, P]e^{-isH}\, ds.$$

Both unitary factors have norm one, so taking norms and using the integration interval's length $|t|$ yields $\|Qe^{-itH}P\| \le |t|\, \|[H, P]\|$. □

III.28.2 Proof of projected curvature with exterior forms

Let P be a smooth idempotent field and extend the projected connection to differential forms with values in the range bundle by $\nabla\omega = P\,d\omega$. Thus ∇^2 denotes the exterior covariant square, including the Lie bracket term in the exterior derivative when evaluated on vector fields.

Alternative proof of the curvature formula. For a section s with $Ps = s$,

$$\nabla^2 s = P\,d(P\,ds) = P(dP) \wedge ds.$$

Since $ds = d(Ps) = (dP)s + Pds$ and $P(dP)P = 0$, the surviving term is

$$P(dP \wedge dP)Ps.$$

□

III.28.3 Restricted low-mode proof

Let $h \neq 0$, $x = kh/2$, and $0 \leq x \leq 1$.

Alternative proof in the original low-mode regime. Since $0 \leq \sin x \leq x$, the error is nonnegative. On $[0, 1]$,

$$\sin x \geq x - \frac{x^3}{6} \geq 0.$$

Therefore

$$\sin^2 x \geq \left(x - \frac{x^3}{6}\right)^2 \geq x^2 - \frac{x^4}{3}.$$

Hence $x^2 - \sin^2 x \leq x^4/3$. Multiplication by $4/h^2$ gives

$$0 \leq k^2 - \frac{4}{h^2}\sin^2(kh/2) \leq \frac{k^4 h^2}{12}.$$

□

How to use this volume

Follow each small model through its matrices and formulas, then read the cited general argument in the traditional volume. The Lean companion records the formal statements and their precise hypotheses.

Routes through the guide

Plates VII.1 through VII.40 identify worked pages independently of PDF pagination. The source line below each title points into the strengthened quantum development in Traditional Part II. The traditional volume also retains the corrected finite core and the completion supplements as distinct branches.

Route	Mathematical question	Plates
Seeds and symmetry	Which algebra and multiplicities determine the generated span?	1–7, 29
Coherent circuits	Which amplitudes interfere before an outcome is measured?	8–11, 27
Moving subspaces	How do transport, local charts, and curvature enter?	12–15, 30
Channels and recovery	What survives dephasing and which errors can be corrected?	16–20, 31–33
Hidden dynamics	What self-energy and memory return after elimination?	21–23, 34–36
Limits and constraints	Which additional hypotheses support continuum passage?	24–26, 37–40
Evidence	What is checked, proved, or still open?	28, 40

Keep the operation visible

A complete packet decomposition retains the vector. Measurement and dephasing can remove coherence. Eliminating hidden states produces an exact memory equation, while deleting their coupling usually produces a different model.

A finite example checks its specified matrices or formulas. A written theorem gives an argument under quantified assumptions. A compiled Lean theorem certifies its formal statement in the recorded environment. The reading links help compare these accounts while keeping their hypotheses and evidence scopes distinct.

Preservation and evidence

The first 28 worked pages retain the mathematical content of the original appendix. Twelve further pages explain the strengthened results. All 38 editable TikZ plates and their Python generators remain available. The retained execution records describe the example suites, their tested inputs, and their rejected controls. The formal companion distinguishes these finite checks from the theorem correspondence.

The finite formal correspondence covers nineteen scoped results. The broader quantum development retains its written hypotheses, and the five continuum research targets remain open. The examples explain how the finite results work and where further arguments are needed.

Read this as: Choose a question, calculate its small model, and then check the hypotheses of the traditional result and the declared formal statement.

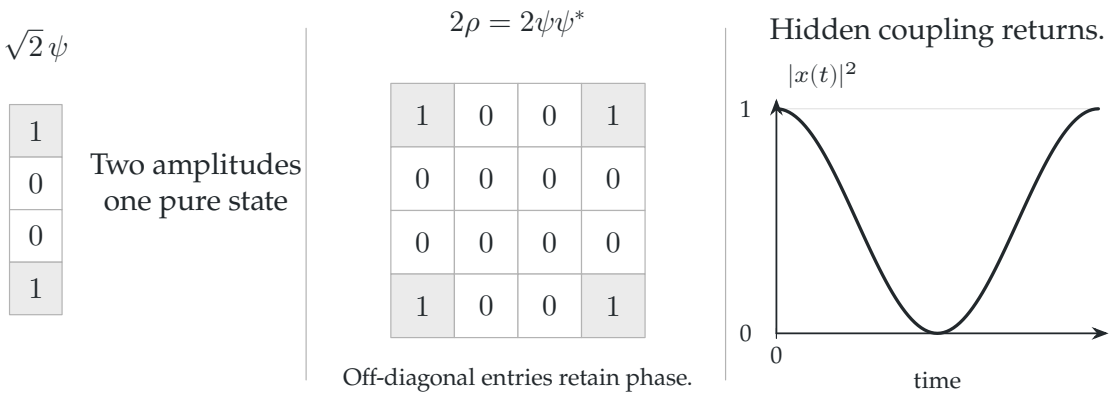
Part IV Worked visual atlas

Plate VII.1 / HOW TO READ THIS APPENDIX

Three objects, three questions

Chapter VII: Section II.1, Section II.7, Section II.9.

A quantum calculation becomes easier to follow when its objects remain visible. This appendix follows a state vector, a density matrix, and a retained part of a dynamical system. They may describe the same experiment, but they do not answer the same question.



Amplitudes retain relative phase

For the Bell vector $\psi = (|00\rangle + |11\rangle)/\sqrt{2}$, the two nonzero entries are amplitudes. The density matrix $\rho = \psi\psi^*$ also records coherence in its off-diagonal entries. Measurements that compare the two addresses can detect this coherence. Deleting those entries applies a physical channel and changes the state.

How hidden dynamics returns

When a Hamiltonian couples a retained state to a hidden state, the retained amplitude can leave and later return. Its exact equation then remembers the hidden coupling. A smaller matrix obtained by deleting rows and columns does not generally reproduce that equation.

A route through the paper

Plate VII.3–Plate VII.10 develop seeds, symmetry, and coherent circuit paths. Plate VII.11–Plate VII.15 track leakage and moving subspaces. Plate VII.16–Plate VII.20 distinguish measurement, channels, and recovery. Plate VII.21–Plate VII.27 follow hidden dynamics and finite geometric diagnostics.

Read these examples alongside the mathematical text. The first twenty-eight plates develop the finite constructions and their geometric interpretation. Plate VII.29–Plate VII.40 develop the further results on bounded limits and constraints. The example checks and the inherited Lean certificates have separately recorded scopes.

> **Read this as:** A packet coordinate change preserves information. A measurement, dephasing operation, or truncation may not. Always identify which operation is actually being performed.

Plate VII.2 / AN ATLAS OF TYPES AND CONVENTIONS

The objects on the page

Chapter VII: Definition II.2.1, Definition II.7.1.

The same letter can stand for different kinds of data in different calculations. Keep the following types fixed while reading a formula. In this appendix, vectors are columns, operators compose from right to left, and A^* denotes the conjugate transpose.

Symbol	Object	What it records
$\mathcal{H}, \psi$	Hilbert space and state vector	They specify the space of addresses and its coherent amplitudes.
$\mathcal{A}$	Declared operator algebra	It specifies the available linear operations.
P_a	Orthogonal packet projector	It selects an entire subspace.
ρ	Positive matrix of trace one	It records populations and coherences.
E_k	Positive measurement effect	It gives the outcome probability $\mathrm{Tr}(E_k\rho)$.
K_a	Kraus operator	It describes one channel branch and need not be a projector.
H	Self-adjoint Hamiltonian	It generates evolution in real time through $e^{-\mathrm{i}tH}$.
C, ∇	Packet transport generator and connection	They describe moving subspaces and differentiation within their ranges.
P, Q	Retained and hidden projectors	They define a decomposition with $Q = I - P$.
$\Sigma(z)$	Self-energy	It records the spectral influence of the hidden sector.
$L, T(\tau)$	Nonnegative operator and Euclidean transfer	They determine decay factors and finite heat trace diagnostics.

Normalization and indexing

The tensor factors are ordered from left to right, so $X_1 = X \otimes I \otimes I$. A normalized pure state gives $\rho = \psi\psi^*$. A coefficient matrix used to count a seed orbit need not be normalized. Scaling a nonzero seed does not change its generated linear span.

A Hamiltonian and a Hadamard gate are both traditionally denoted H. Where both appear, the gate is H_{Had} and the Hamiltonian keeps the letter H. Numerical labels on a drawing are coordinates or exact example values, not observations of a physical device.

> **Read this as:** Check the type before interpreting the picture. An idempotent projector, a density matrix, and a channel matrix have different defining equations.

Plate VII.3 / CYCLICITY DEPENDS ON THE GRAMMAR

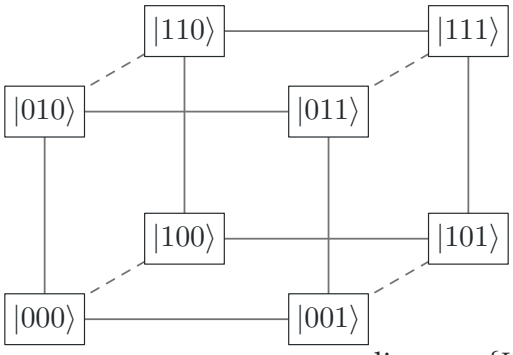
One seed, eight addresses

Chapter VII: Proposition II.2.5.

Start with $\psi_0 = |000\rangle$ and allow the three bit flip operators X_1, X_2, X_3. For a binary address $a = (a_1, a_2, a_3)$, put

$$W_a = X_1^{a_1} X_2^{a_2} X_3^{a_3}. \qquad W_a \psi_0 = |a_1 a_2 a_3\rangle .$$

The drawing shows the actual eight states reached by these words. Dashed edges connect the two squares by changing the first bit. The other edges change one of the remaining bits.



Three actual words

$$W_{101} = X_1 X_3$$

$$W_{011} = X_2 X_3$$

$$W_{110} = X_1 X_2$$

$$\dim \operatorname{span}\{W_a |000\rangle\} = 8$$

Two different meanings of “reach”

The eight vectors are orthonormal, so their *linear span* is all of $\mathbb{C}^8$. This proves cyclicity for the algebra generated by the flips. Preparing every normalized vector requires more than products of these flips alone.

For example, no such word sends $|000\rangle$ to $(|000\rangle + |001\rangle)/\sqrt{2}$. A Hadamard gate, a suitable linear combination in the observable algebra, or a larger preparation grammar is required for that operation. Counting linearly independent addressed vectors and synthesizing a physical gate sequence are separate tasks.

The exact check constructs each 8×8 word matrix, applies it to the seed, and verifies that the matrix of all eight outputs is the identity. It also tests the displayed superposition against every word output. No search over an unspecified gate set is involved.

> **Read this as:** A cyclic seed generates a vector space under an algebra. That conclusion does not automatically provide a unitary preparation procedure for every vector in it.

Plate VII.4 / SCHMIDT RANK AS ACCESSIBLE MULTIPLICITY

Entanglement changes the local span

Chapter VII: Theorem II.2.2, Corollary II.2.4.

Write a bipartite vector as a coefficient matrix C, with the first subsystem indexing rows. Acting only on that subsystem transforms C to AC. Every resulting row remains in the row space of C, and every independent choice of such rows can be obtained in this way.

$$\dim \operatorname{span}\{(A \otimes I)\psi : A \in M_d(\mathbb{C})\} = d \operatorname{rank} C.$$

Product seed matrix

1	0
0	0

row rank 1
orbit dimension 2

Bell seed matrix $\sqrt{2}C$

1	0
0	1

row rank 2
orbit dimension 4

A acts on the left. It cannot invent a new direction in the row space.

The product vector $|00\rangle$ has matrix $C = \operatorname{diag}(1, 0)$. Its local orbit span allows arbitrary vectors in the first subsystem but only the second subsystem direction $|0\rangle$, so its dimension is two. The Bell vector has $C = I/\sqrt{2}$, rank two, and a four-dimensional local orbit span.

What the formula does not say

The action here allows arbitrary matrices and then takes a linear span. It is not the orbit under local unitary gates alone. Local unitary operations preserve Schmidt coefficients. They therefore cannot turn the Bell vector into an arbitrary product vector by a deterministic transformation between pure states.

For the exact calculation, apply the four matrix units E_{ab} to each seed matrix, flatten the outputs, and compute their column rank. The observed dimensions are two and four, respectively. The calculation directly exhibits the independent rows counted by the theorem.

> **Read this as:** Schmidt rank measures the remote multiplicity directions accessible to the local operator span. It is not permission for unrestricted deterministic local state preparation.

Plate VII.5 / AN EXACT MULTIPLICITY LOWER BOUND

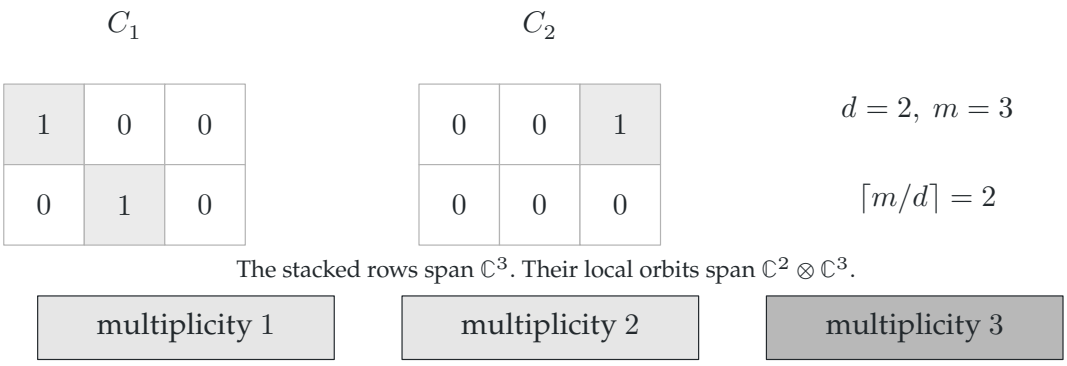
Why one seed is not always enough

Chapter VII: Theorem II.2.3.

Consider $M_2(\mathbb{C}) \otimes I_3$ acting on $\mathbb{C}^2 \otimes \mathbb{C}^3$. A seed is a 2×3 coefficient matrix. Its two rows span at most two directions in the three-dimensional multiplicity factor. Therefore no single seed is cyclic for this algebra.

$$C_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad C_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

These two matrices provide a complete set of row directions. Their individual orbit dimensions are four and two, and together those spans fill the six-dimensional state space.



From the example to the finite formula

For a block $M_d(\mathbb{C}) \otimes I_m$, k seed matrices contribute at most kd independent multiplicity directions. Hence $k \geq \lceil m/d \rceil$. Partitioning a multiplicity basis into groups of size at most d attains this bound. For a direct sum of such blocks, the worst block determines the required count:

$$\operatorname{srank}_{\mathcal{A}}(\mathcal{H}) = \max_i \lceil m_i/d_i \rceil.$$

This is an invariant of the declared representation of the algebra, and depends on more than the dimension of the Hilbert space alone. If the available algebra were enlarged to the full matrix algebra on $\mathbb{C}^6$, every nonzero seed would instead be cyclic.

The example checks both the ranks of the separate coefficient matrices and the rank three of their stacked row matrix. The argument explains why the construction meets the lower bound and therefore proves minimality.

> **Read this as:** Missing multiplicity directions cannot be created by reusing the same left action. Adding seeds and enlarging the operator algebra provide different ways to supply those directions.

Plate VII.6 / CHARACTER PROJECTORS YOU CAN MULTIPLY

Four phases resolve a cyclic shift

Chapter VII: Corollary II.3.3.

On four sites, let $S\,|k\rangle = |k+1 \bmod 4\rangle$ and let $\omega = \mathrm{i}$. The character projectors are the exact matrices

$$P_j = \frac{1}{4}\sum_{r=0}^{3} \mathrm{i}^{-jr} S^r, \qquad v_j(k) = \frac{1}{2}\mathrm{i}^{-jk}, \qquad P_j = v_j v_j^*.$$

All entries belong to the Gaussian rationals. Thus no approximation to a root of unity is needed in this model.

$2V$, with $V^*V = I$

	$j=0$	$j=1$	$j=2$	$j=3$
site 0	1	1	1	1
site 1	1	$-i$	-1	i
site 2	1	-1	1	-1
site 3	1	i	-1	$-i$

$$Sv_0 = 1v_0 \qquad P_0 = v_0 v_0^*$$

$$Sv_1 = iv_1 \qquad P_1 = v_1 v_1^*$$

$$Sv_2 = -1v_2 \qquad P_2 = v_2 v_2^*$$

$$Sv_3 = -iv_3 \qquad P_3 = v_3 v_3^*$$

Each column has four entries of magnitude $1/2$.

Reading the columns

Each column v_j is a normalized wave around the four sites. Advancing the address multiplies it by i^j. The entries show phase differences rather than four different probability distributions: every column has uniform site probability $1/4$.

Multiplying the actual projectors gives $P_jP_k = 0$ for $j \neq k$, $P_j^2 = P_j$, and $\sum_j P_j = I$. In particular, the complete packet tuple $(P_j\psi)_j$ retains the original vector exactly. Recording only the four packet norms is a different operation and can discard phase information between packets.

The same shift on four sites returns on Plate VII.27, where it diagonalizes the Hamiltonian $L = 2I - S - S^*$, and its four character phases explain perfect transfer between opposite sites. The same fixed projectors serve as coordinates, symmetry sectors, and a complete interference ledger.

> **Read this as:** A character projector retains a complex subspace component. Its norm summarizes the size of that component while leaving its phase and direction unspecified.

Plate VII.7 / TENSOR REFINEMENT BY TOTAL PARITY

A coarse packet can still contain coherence

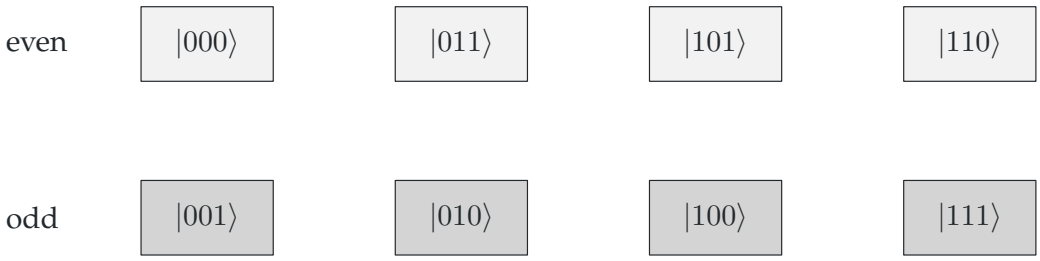
Chapter VII: Theorem II.3.7, Corollary II.3.8.

For three qubits, the eight fine projectors are $P_a = |a\rangle\langle a|$. Grouping addresses according to the parity of their Hamming weight gives

$$P_{\text{even}} = P_{000} + P_{011} + P_{101} + P_{110} = \frac{I + Z \otimes Z \otimes Z}{2}, \qquad P_{\text{odd}} = I - P_{\text{even}}.$$

This is a sum of subspaces, not an averaging of their coordinates. The rank-four even packet retains all complex amplitudes on its four children.

Each coarse projector has rank four. All eight fine projectors have rank one.



Refinement is not measurement

The fine packet description and the coarse packet description each reconstruct the full vector when every component is retained. Replacing a state by the mixture $\sum_a P_a \rho P_a$ is instead a dephasing channel. That operation removes coherence between the fine packets and can differ from dephasing only between the even and odd sectors.

For instance, a superposition of $|000\rangle$ and $|011\rangle$ lies entirely in the even packet. Its internal relative phase survives a parity measurement but is lost by a complete measurement in the computational basis if outcomes are forgotten. The two measurements use projectors from the same refinement hierarchy but expose different information.

The exact check verifies all eight addresses, their parity grouping, the rank of both coarse projectors, and equality with the tensor expression above. The general residue version replaces parity by total label modulo N and groups the tensor descendants by their summed residue.

> **Read this as:** A coarse packet is a whole subspace. Internal coherence survives a change of address description. It is lost only when an operation discards it.

Plate VII.8 / INTERMEDIATE PACKETS ARE NOT FINAL PACKETS

Track the Bell state through the circuit

Chapter VII: Theorem II.4.1, Example II.4.6.

The input is $|00\rangle$. First apply a Hadamard to the first qubit, then a controlled flip:

$$|00\rangle \xrightarrow{H_{\text{Had}}\otimes I} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

The displayed columns keep amplitudes, including their normalization, at every step. The middle vector occupies both parity packets. The final vector occupies only the even packet.

	seed		after $H \otimes I$		after CNOT
00	1	00	$\frac{\sqrt{2}}{2}$	00	$\frac{\sqrt{2}}{2}$
01	0	01	0	01	0
10	0	10	$\frac{\sqrt{2}}{2}$	10	0
11	0	11	0	11	$\frac{\sqrt{2}}{2}$
	even only		both parity packets		even only, with coherence

How the exact path expansion works

Insert $P_{\text{even}} + P_{\text{odd}} = I$ between the two gates. Each summand records a particular intermediate packet, while the boundary projectors select its input and output packets. Summing all terms recovers the original 4×4 circuit matrix:

$$\sum_{a,b,c} P_b \,\text{CNOT}\, P_a (H_{\text{Had}} \otimes I) P_c = \text{CNOT}(H_{\text{Had}} \otimes I).$$

This expansion is an identity even though the intermediate parity is not measured. Physically inserting a measurement would alter the problem by removing interference between its alternatives. The derivation resolves intermediate coordinates without physically measuring them.

In the finite check, all eight matrix terms indexed by parity are constructed and summed. The matrix equality is checked before specializing to the input vector. It therefore verifies the decomposition for every input of this example, not merely agreement on the Bell seed.

> **Read this as:** A final invariant packet does not imply that each preceding gate preserved it. Follow the amplitudes through the actual intermediate resolutions.

Plate VII.9 / HOW TWO PATH AMPLITUDES CANCEL

Why path probabilities do not add

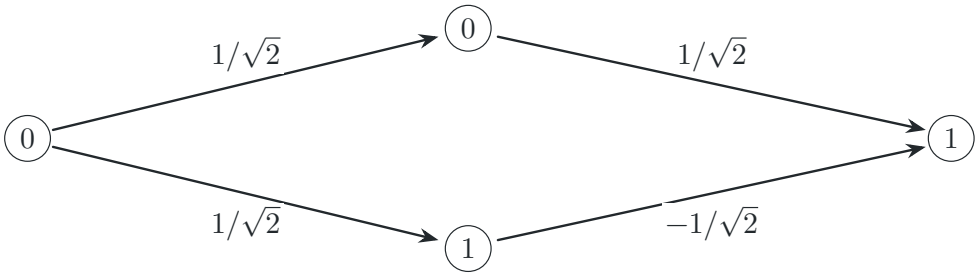
Chapter VII: Theorem II.4.2, Theorem II.4.4.

Two consecutive Hadamard gates multiply to the identity. Starting from $|0\rangle$, there are two resolved paths to output $|1\rangle$. Their signed products are

$$\langle 1|\, H_{\mathrm{Had}}\, |0\rangle\, \langle 0|\, H_{\mathrm{Had}}\, |0\rangle = \frac{1}{2}, \qquad \langle 1|\, H_{\mathrm{Had}}\, |1\rangle\, \langle 1|\, H_{\mathrm{Had}}\, |0\rangle = -\frac{1}{2}.$$

Their sum is zero. Squaring each separately gives a different answer.

Two actual Hadamard paths, not classical alternatives



$$\text{amplitude} = 1/2 - 1/2 = 0 \quad \neq \quad \text{sum of path probabilities} = 1/2$$

Where the missing term goes

For path amplitudes a, b, the outcome probability is $|a+b|^2 = |a|^2 + |b|^2 + 2\Re(a\bar{b})$. Here the last term is $-1/2$, canceling the first two terms. The path picture retains this information only if its edges carry complex amplitudes rather than unsigned reachability or probabilities.

The packetwise unitarity identity is likewise an operator identity, including the off-diagonal cancellations:

$$\sum_b U_{ba}^* U_{bc} = \delta_{ac} P_a, \qquad U_{ba} = P_b U P_a.$$

It cannot be replaced by the assertion that every path individually has unit norm. The exact checks verify all four choices of a, c for the two rank-one packets, as well as the two path products leading to the opposite output.

A measurement at the intermediate stage would remove the cross term. In that modified experiment the two alternatives really would contribute total probability $1/2$ to output one. Adding the two path probabilities would therefore describe the experiment with an intermediate measurement.

> **Read this as:** Insert projectors algebraically to resolve paths, then add amplitudes. Add probabilities only when the physical setup has actually made the alternatives incoherent.

Plate VII.10 / ONE HAMILTONIAN IN TWO ORDERS

A symmetry reveals blocks, not approximations

Chapter VII: Theorem II.5.1.

Take the exact rational Hamiltonian

$$H = X \otimes X + \frac{3}{8}(Z \otimes I + I \otimes Z).$$

It commutes with total parity $Z \otimes Z$. Reordering the computational basis into even addresses $(00, 11)$ and odd addresses $(01, 10)$ displays the invariant subspaces explicitly.

Computational order

	00	01	10	11
00	$\frac{3}{4}$	0	0	1
01	0	0	1	0
10	0	1	0	0
11	1	0	0	$-\frac{3}{4}$

same operator
different order

Parity order

	00	11	01	10
00	$\frac{3}{4}$	1	0	0
11	1	$-\frac{3}{4}$	0	0
01	0	0	0	1
10	0	0	1	0

Even eigenvalues: $\pm 5/4$. Odd eigenvalues: ± 1.

$$H_{\text{even}} = \begin{pmatrix} 3/4 & 1 \\ 1 & -3/4 \end{pmatrix}, \qquad H_{\text{odd}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The even block squares to $(25/16)I$, so its eigenvalues are $\pm 5/4$. The odd block squares to I, so its eigenvalues are ± 1. Exponentiating either block gives the corresponding part of the exact evolution. Transfer between the packets remains zero for every real time.

What has changed?

Only the order of basis vectors has changed. No matrix entries, amplitudes, or energy levels were discarded. This is why the decomposition is exact. Keeping just the even block would be a separate restriction of the state space, justified for initial data known to be even and for dynamics that preserve it.

The next page breaks a symmetry in a smaller model. There the entries coupling the packets no longer vanish, so their effect needs an explicit bound. The exact tests here multiply the commutator, verify the displayed permutation, and compute all four eigenvalues symbolically.

> **Read this as:** Block diagonalization uses an invariant decomposition already present in the operator. It is not evidence that an arbitrary chosen packet is invariant.

Plate VII.11 / COMMUTATORS AND PROBABILITY BOUNDS

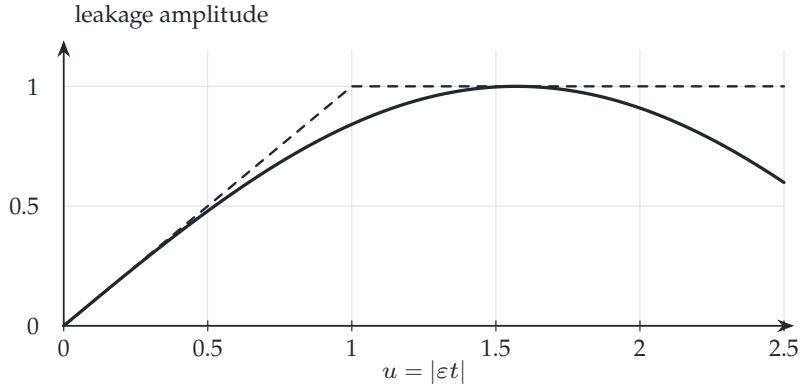
Quantify the amplitude that escapes

Chapter VII: Theorem II.5.3, Corollary II.5.4.

For a two-level system, choose $H = \varepsilon X$, $P = |0\rangle\langle 0|$, and $Q = I - P$. Direct exponentiation gives

$$e^{-\mathrm{i}tH} = \cos(\varepsilon t) I - \mathrm{i}\sin(\varepsilon t) X, \quad \left\| Q e^{-\mathrm{i}tH} P \right\| = |\sin(\varepsilon t)|, \quad \|[H, P]\| = |\varepsilon|.$$

The general commutator estimate becomes the elementary bound $|\sin u| \le |u|$. Unitarity also bounds the amplitude by one.



$H = \varepsilon X$

solid: $|\sin u|$
dashed: $\min(1, u)$

Probability is $\sin^2 u$,
not $|\sin u|$.

From a norm to a probability

For this initial state, the escape probability is $\sin^2(\varepsilon t)$, not $|\sin(\varepsilon t)|$. Consequently

$$\Pr(\text{hidden packet at } t) \le \min\{1, \varepsilon^2 t^2\}.$$

At $\varepsilon t = \pi/6$, the amplitude is $1/2$ and the probability is $1/4$. Near zero, amplitudes are first order while escape probabilities are second order. Mixing those two quantities produces an incorrect interpretation of a leakage plot.

The figure displays samples of the explicit functions above. The supporting check verifies the exact exponential identity, its unitarity, the squared leakage matrix, and the special-time value. The paper's theorem supplies the bound for all times within its stated scope.

At larger times amplitude can return from the hidden packet. The commutator bound therefore limits the largest possible leakage without predicting that it must rise monotonically.

Read this as: The norm estimate controls amplitude. For a normalized state, square it to obtain a probability bound, and retain the distinction between a bound and the actual oscillatory dynamics.

Plate VII.12 / NOT ALL MOTION IS PHYSICAL EVOLUTION

A moving frame creates a connection term

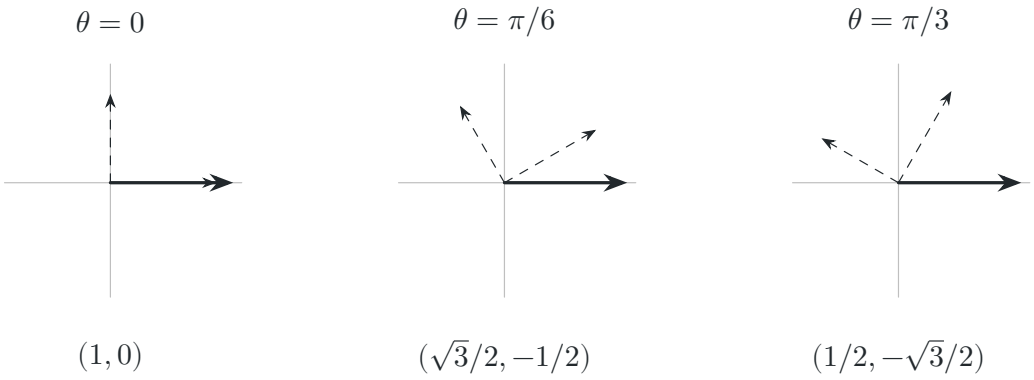
Chapter VII: Theorem II.6.3.

Set the laboratory Hamiltonian to zero and keep $\psi(t) = |0\rangle$ fixed. Describe it in rotating coordinates,

$$U(t) = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}, \qquad y(t) = U(t)^* \psi(t) = \begin{pmatrix} \cos t \\ -\sin t \end{pmatrix}.$$

The vector has not moved in the laboratory, but its coordinate entries have changed.

Solid vector: fixed in the laboratory. Dashed axes: rotating frame.



The missing term can be computed exactly

Let $J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Since $U^*\dot{U} = J$, the coordinate Hamiltonian is

$$H_{\text{frame}} = -\mathrm{i}J = \begin{pmatrix} 0 & \mathrm{i} \\ -\mathrm{i} & 0 \end{pmatrix}, \qquad \mathrm{i}\dot{y} = H_{\text{frame}} y.$$

This matrix is Hermitian. Omitting it would predict a constant y, contradicting the displayed coordinates. In this example the norm of the omitted time derivative is exactly one, so the failure is not a small numerical residual.

For a nonzero laboratory Hamiltonian, the same product rule adds the transformed physical term U^*HU. Thus $U^*HU - \mathrm{i}U^*\dot{U}$ separates the physical generator from the coordinate correction. Both pieces belong to the same equation but have different origins.

The drawing shows actual rotated axes at three angles, not three different laboratory trajectories. The check differentiates the exact matrices and verifies the coordinate Schrödinger equation symbolically. The explicit derivatives make a numerical difference approximation unnecessary.

> **Read this as:** When coordinates move, even a stationary physical vector acquires changing components. The connection term accounts for that change without adding a force to the laboratory system.

Plate VII.13 / THE CANONICAL PACKET CONNECTION

Differentiate the subspace itself

Chapter VII: Theorem II.6.1, Theorem II.6.2.

Continue with $P(t) = U(t)\,\mathrm{diag}(1,0)U(t)^*$ and $Q(t) = I - P(t)$. Differentiating $P^2 = P$ gives

$$\dot{P}P + P\dot{P} = \dot{P}, \qquad P\dot{P}P = 0.$$

The second identity says that the derivative has no component entirely inside the current packet. It moves the subspace relative to its complement.

$P(0)$

1	0
0	0

$\dot{P}(0)$

0	1
1	0

C

0	-1
1	0

$$P\dot{P}P = 0, \qquad C^* = -C, \qquad [C, P] = \dot{P}.$$

For this complete pair of packets, define

$$C = \dot{P}P + \dot{Q}Q.$$

Exact calculation yields the constant matrix $C = J$, with $C^* = -C$ and $[C, P] = \dot{P}$. At $t = 0$, the displayed small matrices allow these identities to be checked by direct multiplication.

From two packets to a complete family

For differentiable orthogonal projectors P_a summing to the identity, the paper constructs

$$C = \sum_a \dot{P}_a P_a = \frac{1}{2}\sum_a [\dot{P}_a, P_a].$$

This single connection reproduces the motion of every packet in the family. No basis within a packet is needed for this formula. A basis choice would add internal coordinate data without changing which subspaces are selected.

The completeness of the family matters in deriving skew-Hermiticity and the simultaneous transport identities. Using only one summand of an incomplete list is not automatically the same connection. The example retains both P and Q throughout, even when the picture emphasizes only the first line.

> **Read this as:** The projector derivative describes motion between subspaces. The canonical connection reconstructs that motion from the complete family without choosing internal packet bases.

Plate VII.14 / THE RANGE BUNDLE IS CONSTRUCTED LOCALLY

A chart failure is not a singular fiber

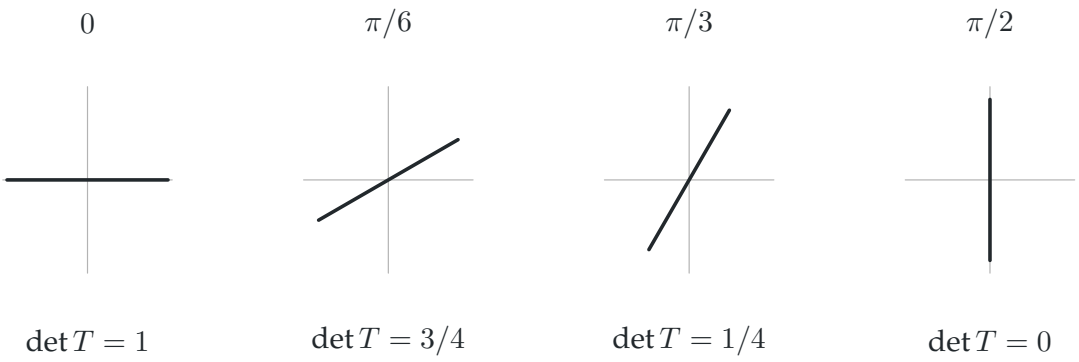
Chapter VII: Theorem II.6.7. Details of the range atlas: Chapter VII, Part III, Sections III.24–III.26 (range-bundle construction).

The fiber of a projector field is the actual fixed subspace $E_x = \{v : P(x)v = v\}$. To compare it with a nearby anchor fiber E_a, the completion proof constructs

$$T_a(x) = P(x)P(a) + (I - P(x))(I - P(a)).$$

It equals the identity at the anchor and intertwines the two projectors. Where it is invertible, its restriction provides local fiber coordinates.

The ranges of $P(\theta)$ are shown with a horizontal anchor.



At the last position, use a new anchor. The fiber still has dimension one.

An exact example of the domain restriction

For the rotating line and anchor $P(0)$,

$$T_0(t) = \cos t\, U(t), \qquad \det T_0(t) = \cos^2 t.$$

The chart fails at $t = \pi/2$, but $P(\pi/2) = \mathrm{diag}(0, 1)$ is a perfectly regular rank-one projector. Choosing a vertical anchor supplies a new chart there. What failed was the comparison with one reference fiber, not the fiber itself.

In general, the open sets where $T_a(x)$ is invertible cover the base. The inverse chart maps and smooth overlap maps are constructed from the actual matrices. Their cocycle identities express consistent coordinate changes, rather than assuming a global basis exists.

The range-bundle construction in Part III proves the intrinsic connection and its curvature using actual manifold derivatives. Its scope includes bounded idempotents, not only orthogonal ones. The illustration uses a finite-dimensional orthogonal projector to make the construction visible. Those simplifying choices do not restrict the recorded theorem.

> **Read this as:** A smooth bundle is built from compatible local charts. The singularity of one chart determinant can be harmless, provided another anchor covers the same smooth fiber.

Plate VII.15 / AN EXACT BLOCH PROJECTOR CALCULATION

Curvature measures a local area response

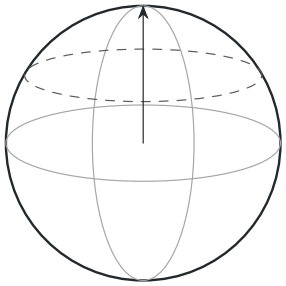
Chapter VII: Theorem II.6.7, Corollary II.6.8.

On the unit sphere, set $n = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ and

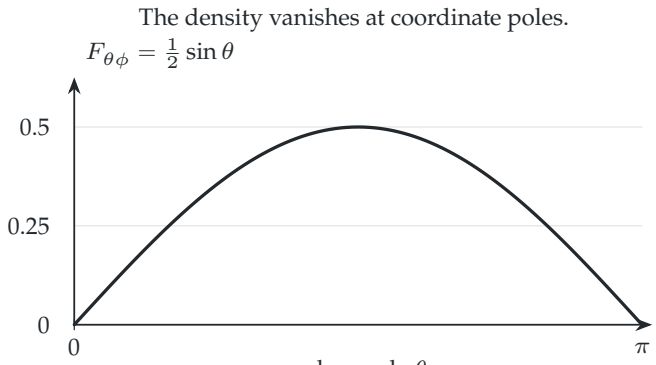
$$P(\theta,\phi) = \frac{1}{2}(I + n\cdot\sigma).$$

This is a smooth rank-one orthogonal projector. The intrinsic curvature is the endomorphism-valued two-form $P(\,\mathrm{d}P \wedge \mathrm{d}P)P$. With the paper's scalar convention, its component is

$$F_{\theta\phi} = -\mathrm{i}\,\mathrm{Tr}\big(P[\partial_\theta P, \partial_\phi P]\big) = \frac{1}{2}\sin\theta.$$



Cap flux at $\theta = \pi/3$ is $\pi/2$.



A local frame gives the same answer

Away from the south pole, use $u_N = (\cos(\theta/2), e^{\mathrm{i}\phi}\sin(\theta/2))^T$. Then $P = u_N u_N^*$ and the scalar one-form is

$$A_N = -\mathrm{i}u_N^*\,\mathrm{d}u_N = \frac{1-\cos\theta}{2}\,\mathrm{d}\phi, \qquad \mathrm{d}A_N = \frac{1}{2}\sin\theta\,\mathrm{d}\theta\wedge\mathrm{d}\phi.$$

Thus the curvature flux through the cap $0 \le \theta \le \theta_0$ is $\pi(1-\cos\theta_0)$. At $\theta_0 = \pi/3$, it is exactly $\pi/2$. The latitude drawn in the plate illustrates this cap. The exact flux comes from the formula above.

The vanishing of the coordinate density at a pole does not mean the geometric curvature disappears there. The spherical area form also carries the factor $\sin\theta$. Assigning a phase to parallel transport also requires an orientation and a sign convention for the connection. These choices determine how flux relates to phase.

> **Read this as:** The projector formula and a local frame calculation agree. Curvature is intrinsic even when no single smooth frame covers the whole sphere.

Plate VII.16 / THE ANSWER DEPENDS ON ITS EFFECTS

Which coherence can a measurement see?

Chapter VII: Theorem II.7.2, Theorem II.7.4, Corollary II.7.5.

For the Bell density matrix $\rho = |\Phi^+\rangle\langle\Phi^+|$, parity dephasing does nothing: both nonzero amplitudes already lie in the even subspace. Dephasing in the computational basis removes the entries between addresses 00 and 11.

2ρ, also $2D_{\text{parity}}(\rho)$

	00	01	10	11
00	1	0	0	1
01	0	0	0	0
10	0	0	0	0
11	1	0	0	1

same diagonal
different coherence

$2D_{\text{computational}}(\rho)$

	00	01	10	11
00	1	0	0	0
01	0	0	0	0
10	0	0	0	0
11	0	0	0	1

Bell-projector outcome: 1 on the left, $1/2$ on the right.

Every effect in the computational basis $E_a = |a\rangle\langle a|$ has the same expectation on both matrices, namely $1/2$ at 00 and 11 and zero elsewhere. These measurements read only the diagonal. They cannot distinguish the pure Bell state from this incoherent mixture.

A measurement that resolves the difference

Use the valid two-outcome measurement with effects $E = \rho$ and $I - E$. Then

$$\operatorname{Tr}(E\rho) = 1, \qquad \operatorname{Tr}\Big(E D_{\text{computational}}(\rho)\Big) = \frac{1}{2}.$$

The effect itself is not diagonal in the fine computational packets. It compares their coherent combination, so it falls outside the paper's invariance result for block-diagonal effects.

This example contains three operations that should not be conflated: representing a state by all of its packet matrix blocks, applying a dephasing channel, and reading one specified measurement. Only the second discards the off-diagonal blocks. The third may or may not notice that loss, depending on its effects.

The exact checks construct all four fine projectors, both parity projectors, both dephased states, and the outcome traces. They reject the stronger statement that *every* measurement is invariant under dephasing.

> **Read this as:** A measurement's inability to see a coherence does not make that coherence absent. It states a restriction on the observable, not an equality of the underlying states.

Plate VII.17 / POPULATIONS AND COHERENCES TRANSFORM DIFFERENTLY

A channel acts on four matrix entries

Chapter VII: Theorem II.7.6, Example II.7.12.

Choose amplitude damping with $\gamma = 9/25$:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & 4/5 \end{pmatrix}, \qquad K_1 = \begin{pmatrix} 0 & 3/5 \\ 0 & 0 \end{pmatrix}, \qquad K_0^* K_0 + K_1^* K_1 = I.$$

The channel $\Phi(\rho) = \sum_a K_a \rho K_a^*$ is trace preserving and completely positive. The special parameter keeps every displayed coefficient rational.

Superoperator on matrix entries

	00	01	10	11
00	1	0	0	$\frac{9}{25}$
01	0	$\frac{4}{5}$	0	0
10	0	0	$\frac{4}{5}$	0
11	0	0	0	$\frac{16}{25}$

$\gamma = 9/25$

coherence factor $4/5$

Output from $|+\rangle$

$\frac{17}{25}$	$\frac{2}{5}$
$\frac{2}{5}$	$\frac{8}{25}$

$$\operatorname{Tr}\rho = 1, \quad \det\rho = 36/625 > 0$$

The superoperator uses the entry order $(\rho_{00}, \rho_{01}, \rho_{10}, \rho_{11})$. Population leaves the excited state with factor $9/25$, while coherences contract by $4/5$. These are not the same rate or the same coordinate.

For $\rho_+ = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, the exact output is

$$\Phi(\rho_+) = \begin{pmatrix} 17/25 & 2/5 \\ 2/5 & 8/25 \end{pmatrix}.$$

Its trace is one and determinant is $36/625 > 0$, so it is a mixed positive state. Arrows that show population transfer alone would leave its off-diagonal entries unspecified.

How complete positivity is certified

The Choi matrix is an explicit sum of two outer products of vectorized Kraus operators. Its eigenvalues are $41/25, 9/25, 0, 0$, all nonnegative. This gives an independent finite certificate in addition to Kraus normalization. The Choi matrix has trace two under the convention used by the check, which leaves the maximally entangled vector unnormalized.

> **Read this as:** A packet channel acts on pairs of packet addresses, because density matrices have two indices. Population transfer alone does not reconstruct a quantum channel.

Plate VII.18 / WHY COMPLETE POSITIVITY IS THE RELEVANT CONDITION

An ancilla tests more than positivity

Chapter VII: Classical input II.7.7.

Ordinary transpose T sends a positive matrix on one system to another positive matrix and preserves its trace. Those properties are insufficient for a quantum channel on a subsystem of an entangled state. The test must include $\mathrm{id} \otimes \mathsf{T}$ on the larger system.

$$2(\mathrm{id} \otimes \mathsf{T})(\rho_{\mathrm{Bell}})$$

00	1	0	0	0
01	0	0	1	0
10	0	1	0	0
11	0	0	0	1

$$w = (|01\rangle - |10\rangle)/\sqrt{2}$$

$$w^*(\mathrm{id} \otimes \mathsf{T})(\rho)w = -1/2$$

Transpose is positive on one system.
It is not completely positive.

Apply this extended map to the Bell density matrix. The result is

$$(\mathrm{id} \otimes \mathsf{T})(\rho_{\mathrm{Bell}}) = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Its middle block has one negative eigenvalue. More concretely, the normalized vector $w = (|01\rangle - |10\rangle)/\sqrt{2}$ satisfies

$$w^*(\mathrm{id} \otimes \mathsf{T})(\rho_{\mathrm{Bell}})w = -\frac{1}{2}.$$

This exact negative value of the quadratic form certifies failure of positivity on the composite system. No numerical eigenvalue tolerance is needed.

Why the Kraus form avoids this problem

For every auxiliary dimension and positive X, each term $(I \otimes K_a)X(I \otimes K_a)^*$ is positive. Their sum is positive as well. This argument explains the word *completely*. The conclusion holds for every auxiliary dimension, which a test of one input density matrix on the original system cannot establish.

The transpose counterexample is an explanatory finite calculation, not a new addition to the formal theorem count. It makes visible the hypothesis that the manuscript's Kraus result already supplies. The exact check constructs the partial transpose with the stated tensor indexing and evaluates its negative witness directly.

> **Read this as:** A map can look harmless on an isolated system and fail on an entangled extension. Complete positivity retains compatibility with every auxiliary system.

Plate VII.19 / FOUR EXACT SYNDROME PACKETS

Find the error without reading the logical state

Chapter VII: Theorem II.8.2, Theorem II.8.4, Example II.8.8.

Encode a qubit by $|0_L\rangle = |000\rangle$ and $|1_L\rangle = |111\rangle$. Let P project onto this code and take errors $E_a \in \{I, X_1, X_2, X_3\}$. Their images are four orthogonal rank-two packets $Q_a = E_a P E_a^*$ that sum to the full identity.

Channel probabilities: $1/10,\ 2/10,\ 3/10,\ 4/10$.

no error	X_1	X_2	X_3
$\|000\rangle$	$\|100\rangle$	$\|010\rangle$	$\|001\rangle$
$\|111\rangle$	$\|011\rangle$	$\|101\rangle$	$\|110\rangle$
ratio $3:4$	ratio $3:4$	ratio $3:4$	ratio $3:4$

Use the logical state $\psi = (3\,|0_L\rangle + 4\,|1_L\rangle)/5$ and channel probabilities $w = (1, 2, 3, 4)/10$. The noisy state is $\sum_a w_a E_a \psi\psi^* E_a^*$. A syndrome measurement returns label a with probability w_a, independent of the logical amplitudes.

The recovery is an actual channel

The operators $R_a = E_a^* Q_a$ satisfy $\sum_a R_a^* R_a = I$. Applying the recovery $\mathcal{R}(X) = \sum_a R_a X R_a^*$ to the noisy state gives exactly $\psi\psi^*$. The off-diagonal logical coherence $12/25$ is recovered as well as the populations $9/25$ and $16/25$.

The Knill–Laflamme compressions are $P E_a^* E_b P = \delta_{ab} P$. Their scalar form says that the error labels do not resolve the encoded vector. The exact checks evaluate all sixteen compressions, all sixteen syndrome products, the four probabilities, the recovery normalization, and the full recovered matrix.

This code corrects the specified flips of individual bits. That result does not cover arbitrary errors on three qubits or arbitrary errors on a single qubit. The next example exhibits an error outside its protected family.

> **Read this as:** A syndrome identifies an error subspace while leaving a two-dimensional logical subspace intact. Measuring the two logical basis states separately would destroy the information the recovery must preserve.

Plate VII.20 / THE SCALAR COMPRESSION TEST MATTERS

An unchanged syndrome can hide a logical error

Chapter VII: Theorem II.8.2, Proposition II.8.7.

Apply $Z_1 = Z \otimes I \otimes I$ to the same repetition code. It sends $|000\rangle$ to itself and $|111\rangle$ to its negative. Both states remain inside the original code, so the previous syndrome partition reports no bit flip.

$$\frac{3\,|000\rangle + 4\,|111\rangle}{5} \longmapsto \frac{3\,|000\rangle - 4\,|111\rangle}{5}.$$

The populations are unchanged, but the relative phase is not.

$V^* Z_1 V$

1	0
0	-1

before: $(3, 4)/5$

after: $(3, -4)/5$

same code support
different relative phase

Required scalar form cI

c	0
0	c

A trace of zero does not make $\mathrm{diag}(1, -1)$ the zero operator.

Let V be the code isometry with columns $|000\rangle, |111\rangle$. For the proposed error family $\{I, Z_1\}$, its off-diagonal error compression is

$$V^* I^* Z_1 V = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

This matrix is not cI for any scalar c. Thus the Knill–Laflamme condition fails for that family. Its trace is zero even though its action on the code is nonzero. A trace test alone would miss the obstruction.

Known correction versus family recovery

If it is known with certainty that Z_1 occurred, applying Z_1 again reverses it. The failed criterion concerns a *single recovery valid for every unknown error in the declared family*, not invertibility of each error considered separately. This distinction prevents a misleading conclusion that no phase error can ever be undone.

The finite calculation checks the actual compressed matrix and rejects its equality to zero. The scalar obstruction is then transparent from the two unequal diagonal entries.

> **Read this as:** Correction must preserve all logical superpositions for the declared error family. Staying in the code and preserving basis probabilities are not sufficient.

Plate VII.21 / TWO INVERTIBILITY CONDITIONS, NOT ONE

Hidden states return as spectral dependence

Chapter VII: Theorem II.9.2, Theorem II.9.3.

Take $H = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$ and retain its first coordinate. The eliminated block is $H_{QQ} = 2$. Whenever $z - 2$ and the full $zI - H$ are invertible,

$$H_{\text{eff}}(z) = \frac{1}{z-2}, \qquad \left[(zI - H)^{-1}\right]_{00} = \frac{1}{z - 1/(z-2)} = \frac{z-2}{z^2 - 2z - 1}.$$

These are the hypotheses used in the formal companion.

Filled: full resolvent poles. Open at 2: only the eliminated inverse fails.



At $z = 3$: exact retained resolvent $1/2$, bare truncation $1/3$.

At $z = 3$, the exact retained resolvent is $1/2$. Bare truncation uses only $H_{PP} = 0$ and gives $1/3$, an error of $1/6$. The omitted coupling has changed the response even though the retained diagonal entry did not change.

A pole can belong to the elimination chart

At $z = 2$, the eliminated inverse does not exist, so the displayed Schur formula cannot be applied there as written. Nevertheless, the full resolvent exists and its retained entry is zero. The rational expression has a continuation to this value. By contrast, $1 \pm \sqrt{2}$ are poles of the full resolvent.

Invertibility of the full matrix is an independent requirement. At $z = 0$, $H = \operatorname{diag}(0, -1)$ has an invertible eliminated block in $zI - H$, but the full matrix is singular. The original prose omitted this condition. The preserved amendments restore it. This appendix uses the corrected form throughout.

Read this as: An exact elimination formula has a spectral domain. Check both inverses before evaluating it, and distinguish poles of an intermediate inverse from poles of the full physical resolvent.

Plate VII.22 / AN EXACT QUANTUM MEMORY EQUATION

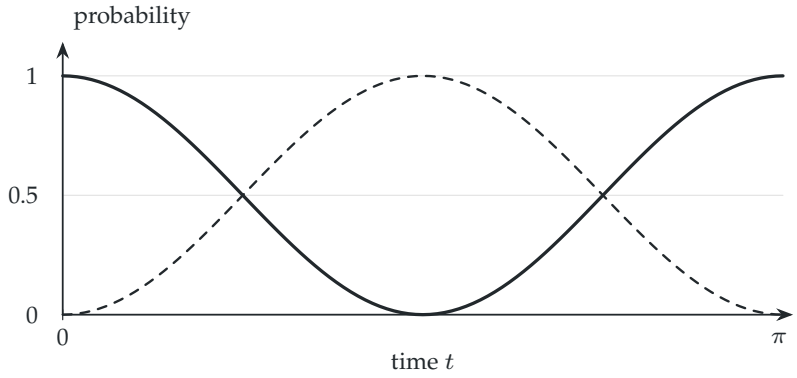
The smaller system remembers the discarded one

Chapter VII: Theorem II.9.8, Theorem II.9.11.

Choose $H = X$, initial state $(1,0)^T$, and retain the first coordinate. The coupled equations and their solution are

$$\mathrm{i}\dot{x} = y, \qquad \mathrm{i}\dot{y} = x, \qquad x(t) = \cos t, \quad y(t) = -\mathrm{i}\sin t.$$

The hidden probability returns periodically. Deleting the hidden row and column would instead leave the zero Hamiltonian and predict $x(t) = 1$.



solid: retained
dashed: hidden

$$K(t) = 1$$

The two probabilities sum to one.

Eliminate exactly, not approximately

Integrating the second equation with $y(0) = 0$ gives $y(t) = -\mathrm{i}\int_0^t x(s)\,ds$. Substitution produces

$$\dot{x}(t) = -\int_0^t x(s)\,ds.$$

The memory kernel is the constant function one. Differentiating once more recovers $\ddot{x} = -x$ with $x(0) = 1$ and $\dot{x}(0) = 0$, explaining the cosine solution. A nonzero hidden initial state would add a forcing term that must also be retained.

The full propagator is a unitary group, but its retained scalar entry is not a semigroup. At $t = s = \pi/4$,

$$x(t+s) - x(t)x(s) = 0 - \frac{1}{2} = -\frac{1}{2}.$$

There is no contradiction: compression of a product need not equal the product of the compressed factors when the hidden subspace carries intermediate amplitude.

The check verifies the differential and integral equations symbolically and the displayed nonzero semigroup defect exactly. The curves visualize the explicit solution and its periodic return of amplitude.

Read this as: Exact elimination replaces a hidden coordinate with memory and forcing from the initial data. Replacing that memory by a constant generator is an additional approximation, not part of the exact theorem.

Plate VII.23 / SPECTRAL POSITIVITY IS BASIS INDEPENDENT

Positive transfer is not a stochastic matrix

Chapter VII: Theorem II.10.1, Theorem II.10.3.

For $H = I - X$, the two eigenvalues are zero and two. Its Euclidean transfer is

$$T(\tau) = e^{-\tau H} = \frac{1}{2}\begin{pmatrix} 1+q & 1-q \\ 1-q & 1+q \end{pmatrix}, \qquad q = e^{-2\tau}.$$

For $\tau \geq 0$, the eigenvalues $1, q$ lie in $(0, 1]$, and multiplying two such matrices replaces q, r by qr. This is the semigroup law in spectral coordinates.

T at $e^{-2\tau} = 1/2$

$\frac{3}{4}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{3}{4}$

Both have spectrum $\{1, 1/2\}$.

ZTZ in another basis

$\frac{3}{4}$	$-\frac{1}{4}$
$-\frac{1}{4}$	$\frac{3}{4}$

Positive-definite operator: $v^* T v > 0$ for $v \neq 0$. Not an entrywise test.

At $q = 1/2$, the displayed T happens also to have nonnegative entries and column sums one. Conjugate by the unitary Z. The spectrum and positivity of the quadratic form are preserved, but both off-diagonal entries become $-1/4$. Thus positivity as an operator is not entrywise positivity and does not by itself give a Markov interpretation.

A useful finite positive kernel

For this nonnegative H, vectors u_a and times $t_a \geq 0$ satisfy

$$\sum_{a,b} \langle u_a, e^{-(t_a+t_b)H} u_b \rangle = \left\| \sum_a e^{-t_a H} u_a \right\|^2 \geq 0.$$

The factorization makes positivity transparent. It remains a finite operator identity. A continuum reconstruction theorem requires a compatible field structure, limits, positivity hypotheses, and the other assumptions of the chosen reconstruction framework. Each of these assumptions requires its own justification beyond the two positive eigenvalues.

The exact check verifies the matrix semigroup law as a polynomial identity in q, r and verifies the unchanged spectrum after conjugation. It rejects entrywise nonnegativity for the conjugated matrix.

> **Read this as:** Euclidean positivity refers to quadratic forms and spectral values. A stochastic interpretation or a continuum physical reconstruction needs additional structure.

Plate VII.24 / THE EXACT HEAT TRACE ON FOUR SITES

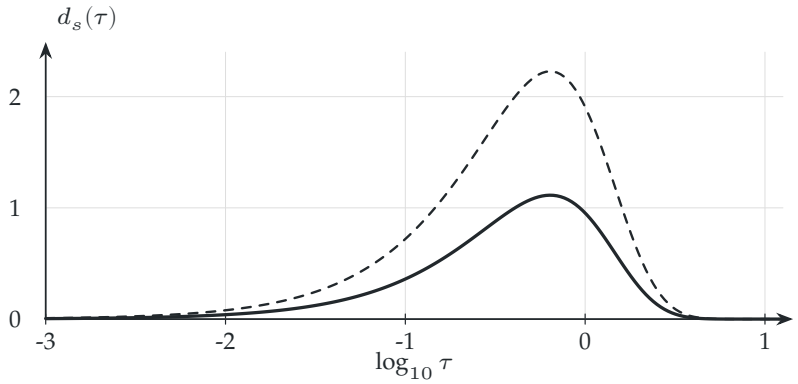
A dimension curve is a scale diagnostic

Chapter VII: Definition II.11.1, Proposition II.11.2.

For the Laplacian on a cycle with four sites $L = 2I - S - S^*$, the eigenvalues are $0, 2, 2, 4$. The normalized heat trace and its diagnostic of spectral dimension are therefore

$$K(\tau) = \frac{1 + 2e^{-2\tau} + e^{-4\tau}}{4} = \frac{(1 + e^{-2\tau})^2}{4}, \qquad d_s(\tau) = -2\tau \frac{K'(\tau)}{K(\tau)} = \frac{8\tau}{e^{2\tau} + 1}.$$

No numerical differentiation is needed for this finite example.



solid: four-cycle
dashed: product of two

No broad plateau is promised at four sites.

Finite diagnostic, not a dimension proof.

Both ends reflect the finite size of the system

As $\tau \to 0$, every eigenmode contributes approximately one and the dimension diagnostic tends to zero. As $\tau \to \infty$, the zero mode dominates, so it again tends to zero. A finite spectrum cannot sustain a nonzero plateau at all scales.

For the Cartesian product of two identical cycles with four sites, the Laplacian eigenvalues add and the heat traces multiply: $K_2 = K^2$. Consequently $d_{s,2} = 2d_s$ exactly. The product identity determines the second curve exactly from the first.

At $\tau = \log 2/2$, the first curve equals $4 \log 2/3$. This noninteger value is a spectral quantity that depends on scale, not a claim that the graph has a fractional topological dimension. Four sites are too few to promise a broad geometric plateau. Experiments on larger grids remain finite diagnostics. Convergence requires the analytic hypotheses developed on Plate VII.37.

> **Read this as:** A convincing continuum dimension needs a stable scale window that persists under controlled refinement. One appealing finite curve is not that theorem.

Plate VII.25 / AN EXACT ERROR ENCLOSURE, NOT A FITTED SLOPE

Refine the grid while holding the mode fixed

Chapter VII: Theorem II.11.4.

For the circle symbol, the continuum value is k^2 and the grid value is

$$\lambda_{k,h} = \frac{4}{h^2}\sin^2\left(\frac{kh}{2}\right), \qquad h \neq 0.$$

For a fixed low mode, the paper bounds the error by $0 \leq k^2 - \lambda_{k,h} \leq k^4 h^2/12$. The spacing condition matters because the displayed expression divides by h^2.

$h = \frac{1}{4}$	$h = \frac{1}{8}$	$h = \frac{1}{16}$
error (approx.)	error (approx.)	error (approx.)
$\simeq 0.005197495$	$\simeq 0.001301405$	$\simeq 0.000325478$
bound $h^2/12$	bound $h^2/12$	bound $h^2/12$
$\frac{1}{192}$	$\frac{1}{768}$	$\frac{1}{3072}$

Successive error ratios lie strictly between 3.99 and 4.

What is certified in the displayed table?

Take $k = 1$ and $h = 1/4, 1/8, 1/16$. For $x = h/2 \in [0, 1]$, alternating Taylor sums give rational lower and upper bounds for $\sin x$. Squaring these positive endpoints and multiplying by $4/h^2$ yields exact rational enclosures for each error. The displayed decimals only locate those much narrower rational intervals.

Each interval lies strictly inside $(0, h^2/12)$. Dividing successive interval endpoints proves that both error ratios are between 3.99 and 4. The check verifies these inequalities by rational arithmetic, without relying on approximate sine values or a fitted logarithmic slope.

Why the mode regime matters

If instead $k = \pi/h$ moves with the grid, the ratio $\lambda_{k,h}/k^2 = 4/\pi^2$ stays constant, not one. The bound's factor k^4 warns that it is not uniform in arbitrary moving high modes. The displayed rational test evaluates the analytic symbol. It does not assert that every chosen spacing defines an integer periodic grid.

Read this as: Second-order convergence is a quantified statement with a specified mode regime. Keep that regime fixed when reading an error curve or comparing refinements.

Plate VII.26 / TEST THE CONSTRAINTS BEFORE THE PROJECTION

A zero compressed anomaly can be automatic

Chapter VII: Theorem II.12.2, Proposition II.12.3.

Take the self-adjoint matrices

$$C_1 = \operatorname{diag}(0, 1, -1), \qquad C_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad P = \operatorname{diag}(1, 0, 0).$$

Both constraints annihilate $P\mathbb{C}^3$, and their common kernel is exactly that one-dimensional space. Nevertheless, their full commutator is nonzero.

$[C_1, C_2]$

0	0	0
0	0	2
0	−2	0

$P = \operatorname{diag}(1, 0, 0)$

$C_1 P = C_2 P = 0$

$P[C_1, C_2]P$

0	0	0
0	0	0
0	0	0

The right matrix is zero by the common-kernel choice, not by operator closure.

Why the compressed test cannot decide closure

Since $C_1 P = C_2 P = 0$, each product $C_1 C_2 P$ and $C_2 C_1 P$ vanishes. Therefore $P[C_1, C_2]P = 0$ without any requirement that the unprojected commutator close in the chosen constraint span. In this example the skew off-diagonal commutator is not a linear combination of the two displayed symmetric constraints.

More generally, an anomaly expression made from their commutators and linear combinations of constraints also kills an exact projector onto the common kernel on the right. The formal companion makes this limit of the diagnostic explicit. A nonzero compressed residual can reveal inconsistent proposed data, but an automatic zero does not construct an anomaly-free constraint algebra.

A substantive closure analysis must inspect the operator relations, their domains, the actual physical subspace, and compatibility under refinement. For infinite systems, these questions also involve domain and limit control that finite matrix arithmetic cannot replace.

The exact example computes both constraint products, the full commutator, and its compression. It deliberately rejects the inference from compressed zero to unprojected zero.

> **Read this as:** Before interpreting a vanishing residual, check whether the definition of its projector already forces it to vanish. A tautological diagnostic cannot supply the missing existence or closure theorem.

Plate VII.27 / SEEDS, CHARACTERS, INTERFERENCE, AND DEPHASING

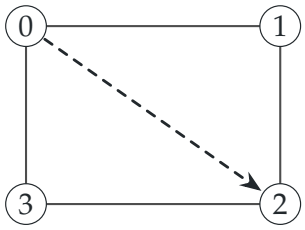
One calculation of transfer to the opposite site

Chapter VII: Corollary II.3.3, Theorem II.4.4, Theorem II.7.4.

Return to the shift on four sites S and Laplacian $L = 2I - S - S^*$. The same character projectors have energies $(0, 2, 4, 2)$. At time $t = \pi/2$, their evolution phases are $(1, -1, 1, -1)$, exactly the character eigenvalues of S^2. Hence

$$e^{-\mathrm{i}\pi L/2} = S^2, \qquad |0\rangle \longmapsto |2\rangle\,.$$

$e^{-\mathrm{i}\pi L/2} = S^2$



start	time $\pi/2$
1	0
0	0
0	1
0	0

Target probability after partial momentum dephasing: $1 - 3\eta/4$.

At halfway time $\pi/4$, the actual amplitude vector is $(-\mathrm{i}, 1, \mathrm{i}, 1)^T/2$, giving uniform site probabilities. The subsequent phases recombine those four amplitudes into a single output at the opposite site. Uniform intermediate probabilities do not imply an incoherent intermediate state.

A channel changes the conclusion predictably

Before evolving, mix the seed density matrix with its complete momentum dephasing:

$$\rho_\eta = (1-\eta)\,|0\rangle\langle 0| + \eta \sum_j P_j\,|0\rangle\langle 0|\,P_j = (1-\eta)\,|0\rangle\langle 0| + \eta I/4, \quad 0 \le \eta \le 1.$$

At transfer time, the probability at the opposite site is exactly $1 - 3\eta/4$. Complete momentum dephasing reduces it to $1/4$. This calculation isolates how discarded coherence changes the interference result.

Finally, the shift grammar generates four independent vectors from $|0\rangle$, while the Hamiltonian powers generate only three: its repeated energy links two momentum directions. The same seed can therefore be cyclic for the shift algebra and noncyclic for the Hamiltonian algebra, even though perfect transfer occurs at one time.

> **Read this as:** Perfect transfer is an exact finite interference statement. It neither implies cyclicity for every operator grammar nor survives an unaccounted dephasing operation.

Plate VII.28 / AN EVIDENCE GUIDE FOR THE READER

What these pictures establish

Chapter VII: Section II.14. Formal scope: the nineteen-result correspondence in the Lean companion.

The appendix uses small objects to make the paper's constructions inspectable. Every matrix and finite example is supplied in editable source, with a separate check record. The plates make those calculations visible. Their appearance helps explain the result, while the equations and check records supply its evidence.

Evidence	What it supports	What it does not supply
Exact matrices	The displayed finite identities, witnesses, and counterexamples.	A classification of arbitrary systems.
Symbolic derivatives	The explicit rotating and Bloch examples.	A proof inferred from numerical differentiation.
Rational enclosures	The three displayed dispersion intervals and ratio inequalities.	A continuum theorem from three sample points.
Formal companion	Its 19 registered finite items under their recorded hypotheses.	A fresh kernel run for this appendix.
Numerical record	The original finite software examples in the tested environment.	Physical experiments or unrestricted convergence.

Which source is authoritative?

The formal companion records 323 supporting theorem declarations in 25 modules for nineteen scoped finite items. The range-bundle construction and the finite examples have the distinct hypotheses stated in their respective treatments. Source and verification records identify the evidence for each claim.

The five research boundaries remain separate: an anomaly-free coherent refinement, uniform control of effective operators, Lorentzian reconstruction, an Einstein infrared limit, and a positive physical massless spin-two sector. A finite projector bundle, positive transfer matrix, or spectral diagnostic does not settle them.

A practical way to use the appendix

Choose one page, reproduce its matrix calculation, and then read the cited theorem with its hypotheses. The retained execution records give the exact assertions, tested ranges, and rejected controls for the finite and symbolic suites. The additional examples and their recorded bounds are described on Plate VII.40. In particular, retain the two resolvent assumptions on Plate VII.21, the distinction between a known error and an error family on Plate VII.20, and the limitation of testing only the common kernel on Plate VII.26.

> **Read this as:** The finite constructions have explicit scope. The appendix adds understanding and reproducibility, not unproved continuum claims or a new theorem count.

Plate VII.29 / A SIX-DIMENSIONAL REGULAR REPRESENTATION

Nonabelian packets carry multiplicity

Chapter VII: Theorem II.3.1, Theorem II.3.2.

Let $\mathcal{H} = \mathbb{C}[S_3]$ have the six orthonormal basis vectors $|g\rangle$, with left action $U(h)\,|g\rangle = |hg\rangle$. The irreducible characters on the identity, transposition, and three-cycle classes are

	e	(12)	(123)	d_λ
trivial	1	1	1	1
sign	1	-1	1	1
standard	2	0	-1	2

$$P_\lambda = \frac{d_\lambda}{6} \sum_{g \in S_3} \chi_\lambda(g^{-1}) U(g).$$

All matrices in this example are rational. Multiplying them gives self-adjoint idempotents of ranks $1, 1, 4$, with pairwise zero products and sum I_6.



$$\mathbb{C}[S_3] \cong \mathbb{C} \oplus \mathbb{C} \oplus (\mathbb{C}^2 \otimes \mathbb{C}^2)$$

The standard packet has dimension four

The standard representation occurs twice in the regular representation. Its isotypic component therefore has dimension $2 \cdot 2 = 4$: one factor carries the two-dimensional irreducible action and the other records multiplicity. A character projector selects the whole isotypic component. It does not select a particular copy or a particular internal vector.

Representation matrix units refine the irreducible factor, while leaving the multiplicity factor untouched. Their multiplication law is stronger information than a list of packet ranks. A basis chosen inside this packet supplies coordinates. It does not canonically distinguish the two copies of the standard representation.

What the retained check establishes

The retained exact check constructs all six permutation matrices, tests all 36 multiplication relations, computes the character averages, and checks ranks independently by Gaussian elimination and traces. It rejects the claim that the standard isotypic projector has rank two. This is one specified representation of S_3, not a computational classification of all finite group representations.

> **Read this as:** For a nonabelian symmetry, distinguish irreducible dimension from multiplicity. A central character projector retains both.

Plate VII.30 / AN EXACT LOCAL CHART AND ITS FAILURE

Rotate whole packets by a polar factor

Chapter VII: Theorem II.6.5, Theorem II.6.6.

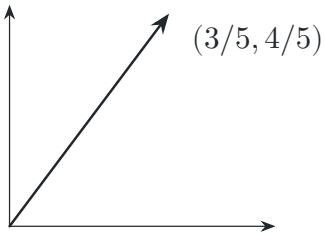
On $\mathcal{H} = \mathbb{C}^2$, put $P_0 = \operatorname{diag}(1,0)$, $P_1 = I - P_0$, and

$$R = \begin{pmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{pmatrix}, \qquad Q_a = RP_aR^*, \qquad S = \sum_{a=0}^{1} Q_aP_a = \frac{3}{5}R.$$

The exact Gram matrices give

$$(S-I)^*(S-I) = \frac{16}{25}I, \quad S^*S = \frac{9}{25}I, \quad \|S-I\| = \frac{4}{5} < 1.$$

Consequently the unitary polar factor is $S(S^*S)^{-1/2} = R$, and $Q_aR = RP_a$ for both complete packets.



$$S = (3/5)R$$
$$S^*S = (9/25)I$$
$$\downarrow$$
$$S(S^*S)^{-1/2} = R$$

The norm condition selects a chart

If the target rank-one packet is orthogonal to the initial one, the two labels are exchanged and $S = 0$. Its polar inverse square root is undefined. This is a failure of this direct local construction, although a unitary carrying one packet system to the other still exists. The strict norm assumption is sufficient, not a global existence criterion for all packet rotations.

Short steps can be composed before reaching a distant endpoint. For example, the rational step $R_1 = \begin{pmatrix} 4/5 & -3/5 \\ 3/5 & 4/5 \end{pmatrix}$ obeys

$$R_1^2 = \begin{pmatrix} 7/25 & -24/25 \\ 24/25 & 7/25 \end{pmatrix}.$$

The multiplication is exact and each step has a valid local overlap. The convergence theorem has a separate analytic contract: a twice continuously differentiable complete projector path on a compact time interval, sufficiently fine partitions, and a constant depending on the path in an $O(|\pi|)$ estimate in operator norm.

Evidence boundary

The retained run checks the Gram identities, polar factor, intertwining, and product of two rational steps. It also checks that the overlap has rank zero when the endpoint packets are orthogonal. It does not certify the analytic convergence theorem by sampling a path.

> **Read this as:** A local polar chart can fail while the subspaces remain regular. Use the stated chart hypothesis and the separate convergence theorem when composing small rotations.

Plate VII.31 / CHANNEL, CHOI, AND GENERATOR CHECKS

Coherence order survives a covariant channel

Chapter VII: Theorem II.7.8, Theorem II.7.10, Theorem II.7.14.

Take $S = Z = \mathrm{diag}(1, -1)$ on $\mathbb{C}^2$. Under conjugation $X \mapsto ZXZ$, diagonal matrices have coherence order zero and off-diagonal matrices have coherence order one modulo two. Its projectors on Hilbert–Schmidt operator space are $\mathcal{L}_0(X) = (X + ZXZ)/2$ and $\mathcal{L}_1(X) = (X - ZXZ)/2$.

Choose amplitude damping with $K_0 = \mathrm{diag}(1, 4/5)$ and $K_1 = (3/5)\,|0\rangle\langle 1|$. The channel is trace preserving, population flows from state one to state zero, and each coherence contracts by $4/5$.

even Choi addresses $00, 11$	odd Choi addresses $01, 10$
$\begin{pmatrix} 1 & 4/5 \\ 4/5 & 16/25 \end{pmatrix}$	$\mathrm{diag}(9/25, 0)$

$$\Phi(|0\rangle\langle 1|) = (4/5)\,|0\rangle\langle 1|$$

With the unnormalized Choi convention $J = \sum_{a,b} \Phi(|a\rangle\langle b|) \otimes |a\rangle\langle b|$ and basis $(00, 01, 10, 11)$, the exact matrix is

$$J = \begin{pmatrix} 1 & 0 & 0 & 4/5 \\ 0 & 9/25 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4/5 & 0 & 0 & 16/25 \end{pmatrix}, \qquad [J, Z \otimes Z] = 0.$$

The $00, 11$ block is positive and has rank one. The other nonzero eigenvalue is $9/25$. The retained check also verifies that every principal minor is nonnegative.

Symmetry in continuous time has its own hypotheses

For the Lindblad generator with Hamiltonian $H = \mathrm{diag}(0, 2)$ and jump $L = (3/5)\,|0\rangle\langle 1|$, H has charge zero and L has charge one. Each matrix unit stays within its coherence order under the generator. Replacing H by X breaks that conclusion: the generator then sends a diagonal matrix to off-diagonal entries.

All four input matrix units are checked exactly for channel covariance, generator order preservation, and zero generator trace. The paper uses the finite exponential series to extend generator invariance to the semigroup. The retained calculation uses no numerical solver for the master equation. Symmetry alone does not close the scalar population equations inside packets of higher rank.

> **Read this as:** Charge is a property of operator space. Covariance preserves its coherent sectors, while complete positivity is checked separately through Kraus or Choi data.

Plate VII.32 / A TOTAL CHANNEL FROM SINGULAR ERROR DATA

Null Gram directions are removed before recovery

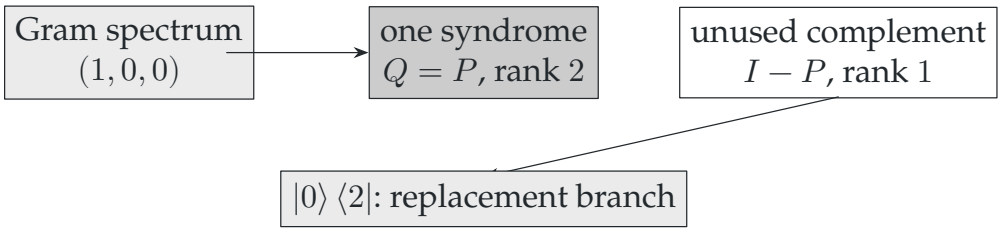
Chapter VII: Theorem II.8.3, Theorem II.8.4.

Let $\mathcal{H} = \mathbb{C}^3$, with nonzero code projector $P = \mathrm{diag}(1, 1, 0)$. On the code use the error family $E_1 = (3/5)P$, $E_2 = (4/5)P$, and $E_3 = 0$. Its Knill–Laflamme Gram matrix is

$$c = \frac{1}{25}\begin{pmatrix} 9 & 12 & 0 \\ 12 & 16 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad PE_a^* E_b P = c_{ab} P.$$

The errors are redundant rather than three distinct syndromes. A real unitary change of error coordinates gives

$$W = \begin{pmatrix} 3/5 & 4/5 & 0 \\ -4/5 & 3/5 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad WcW^* = \mathrm{diag}(1, 0, 0), \quad (F_1, F_2, F_3) = (P, 0, 0).$$



Normalize only positive eigenvalues

The positive branch has $V_1 = P$ as an isometry *on the code*, and syndrome support $Q = P$. No factor $\lambda^{-1/2}$ is formed on either zero branch: those branches vanish on the code. Treating them as extra recoverable packets would divide by zero.

The syndrome support is smaller than the ambient space. Define a total recovery with Kraus operators

$$R_1 = P, \qquad R_2 = |0\rangle\langle 2|, \qquad R_1^* R_1 + R_2^* R_2 = I_3.$$

It acts as the identity on operators supported on the code and sends the unused complement population into a chosen code state. The declared error map preserves the trace of code states, so recovery returns every code matrix unit, including both off-diagonal units, exactly. It need not preserve states on the unused complement.

Exceptional cases and scope

This construction assumes a nonzero code so that a complement replacement state exists. The case of a zero code is vacuous and has its own formal statement in the saved ledger. The retained exact run verifies all nine Gram compressions, the diagonalization, both null branches, total channel normalization, and all four code matrix units. It adds no Lean coverage and does not implement a numerical spectral decoder for arbitrary Gram data.

> **Read this as:** Positive Gram support defines genuine syndrome isometries. Null branches vanish. Specifying the action on the complement makes the recovery a total channel.

Plate VII.33 / THE OPTIMAL SCALAR FOR TWO SPECTRAL POINTS

An approximate defect has a precise norm meaning

Chapter VII: Definition II.8.6, Proposition II.8.7.

On a two-dimensional code, consider the overlap compression

$$A = PE_a^* E_b P = \operatorname{diag}(11/10, 9/10).$$

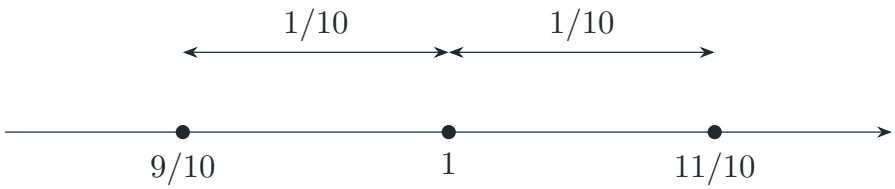
For instance, within the code take $E_a = I$ and $E_b = A$. This declares an error family, not by itself a normalized noise channel. The optimal scalar compression has

$$\inf_{c\in\mathbb{C}} \|A - cI\| = \frac{1}{10}, \qquad c = 1 \text{ attains it.}$$

Indeed, for every complex c, the triangle inequality gives

$$\frac{1}{5} = \left|\frac{11}{10} - \frac{9}{10}\right| \le \left|\frac{11}{10} - c\right| + \left|\frac{9}{10} - c\right|.$$

Thus the larger of the two distances is at least $1/10$, while both distances equal $1/10$ at $c = 1$.



one pair: $1/10$, full family $\{I, A\}$: $1/5$

What one pair contributes to the family defect

For unit code vectors ψ, ϕ, this pair therefore satisfies

$$|\langle E_a\psi, E_b\phi\rangle - \langle\psi, \phi\rangle| \le \frac{1}{10}.$$

The bound is attained on either coordinate vector. For the *whole* declared family $\{I, A\}$, the A^*A compression also enters the maximum: its diagonal entries are $121/100, 81/100$, giving pair defect $1/5$. Hence the full family value is $\varepsilon_{\mathrm{KL}} = 1/5$, not $1/10$.

This distinction prevents a tempting shortcut from one favorable overlap to an approximate correction claim. The quantity measures how close all overlap operators are to scalar action on the code. A recovery fidelity bound requires a declared normalized channel, a metric, and a separate continuity result.

What is checked exactly

For $\varepsilon \in \{0, 1/10, 1/4, 1\}$ the script checks $(A - I)^*(A - I) = \varepsilon^2 I$ for $A = \operatorname{diag}(1 + \varepsilon, 1 - \varepsilon)$ and several rational scalar controls. The triangle argument proves the lower bound for every complex scalar. Checking a finite list of candidates would not establish that conclusion. The family maximum above is also recorded separately.

> **Read this as:** An overlap norm has a useful operational interpretation. Evaluate every error pair, then keep that statement separate from an unproved recovery fidelity estimate.

Plate VII.34 / SIGN, STABILITY, AND A CONVERGENT EXPANSION

Control the hidden spectral response

Chapter VII: Theorem II.9.4, Proposition II.9.5, Theorem II.9.12.

Retain one coordinate in $H = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$, so $A = 0$, $B = 1$, and $D = 2$. In its elimination domain, $\Sigma(z) = 1/(z-2)$. At $z = 3 + \mathrm{i}$,

$$\Sigma(3+\mathrm{i}) = \frac{1-\mathrm{i}}{2}, \qquad \Im\Sigma(z) = -(\Im z)\left|(z-2)^{-1}\right|^2 \le 0.$$

The sign in the upper half plane comes from self-adjoint hidden data. It is not an entrywise sign condition on an arbitrary effective matrix.

$$\boxed{\begin{matrix} B = 1,\ D = 2 \\ \Sigma(3) = 1 \end{matrix}} \longleftrightarrow \boxed{\begin{matrix} \widetilde{B} = 9/8,\ \widetilde{D} = 17/8 \\ \widetilde{\Sigma}(3) = 81/56 \end{matrix}}$$

$$25/56 \le 21/32$$

common hypotheses: $g = 9/8$, $\delta = 3/4$

A stability estimate with a common spectral gap

Compare $(B, D) = (1, 2)$ with $(\widetilde{B}, \widetilde{D}) = (9/8, 17/8)$ at $z = 3$. Both hidden spectral distances are at least $\delta = 3/4$, and both couplings are bounded by $g = 9/8$. Then

$$|\Sigma - \widetilde{\Sigma}| = \left|1 - \frac{81}{56}\right| = \frac{25}{56}, \qquad \frac{2g}{\delta}|B - \widetilde{B}| + \frac{g^2}{\delta^2}|D - \widetilde{D}| = \frac{21}{32}.$$

The theorem bounds the difference using a common spectral gap. Applying the bound uniformly to a refinement family requires uniform spectral separation and control of the couplings.

A series near zero energy and its exact remainder

At $z = 1/2$, $|z|\,|D^{-1}| = 1/4 < 1$, so

$$\Sigma(z) = -\sum_{n=0}^{\infty} \frac{z^n}{2^{n+1}}.$$

After the first four terms, the exact remainder has magnitude $1/384$. The geometric series controls the infinite tail. The retained check verifies that remainder, the complex sign identity, and the identities for the projected resolvent and bare truncation at four specified real spectral parameters.

At $z = 2$ the hidden inverse fails. At an eigenvalue of the full Hamiltonian, the retained Feshbach inverse fails. Neither point belongs to the resolvent identity's domain. Analytic continuation of a scalar expression does not erase its theorem hypotheses.

> **Read this as:** A uniform effective response requires a uniform gap and coupling bound. Exact small examples show how those hypotheses enter each estimate.

Plate VII.35 / TWO HIDDEN FREQUENCIES AND A TEST OF LOCAL CLOSURE

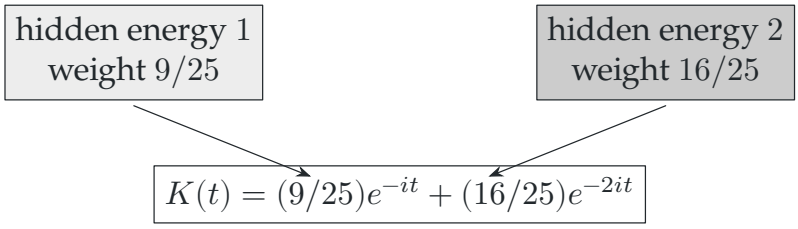
Memory positivity is a Gram identity

Chapter VII: Theorem II.9.9, Theorem II.9.11.

Use retained space $\mathbb{C}$ and hidden space $\mathbb{C}^2$, with $A = 0$, $B = (3/5, 4/5)$, and $D = \operatorname{diag}(1, 2)$. The full Hamiltonian is $\begin{pmatrix} 0 & 3/5 & 4/5 \\ 3/5 & 1 & 0 \\ 4/5 & 0 & 2 \end{pmatrix}$. Its memory kernel is

$$K(t) = \frac{9}{25} e^{-\mathrm{i}t} + \frac{16}{25} e^{-2\mathrm{i}t}, \qquad K(0) = 1.$$

Each oscillatory coefficient is positive although $K(t)$ need not be real or pointwise nonnegative. At $t = \pi/2$, it equals $-16/25 - 9\mathrm{i}/25$.



Positive definite means a matrix test at arbitrary times

For retained scalars v_j and real times t_j, the structural identity is

$$\sum_{j,k} \overline{v}_j K(t_j - t_k) v_k = \left\| \sum_k e^{\mathrm{i}t_k D} B^* v_k \right\|^2 \geq 0.$$

The exact worked check specializes to $t = (0, \pi/2, \pi)$ and $v = (1, 2, -1)$. Its phases lie in $\mathbb{Q}(\mathrm{i})$, so both sides are computed without evaluating trigonometric functions. This Gram test illustrates the factorization. The written identity proves positivity for all choices of times and vectors.

Why this does not close to a bare local equation

With hidden initial state zero, the full system has $\ddot{x}(0) = -(A^2 + BB^*)x(0) = -x(0)$. The bare retained equation $\mathrm{i}\dot{x} = Ax$ would instead give zero acceleration. Therefore exact local closure for every retained initial state fails. More generally, equality of these accelerations forces $BB^* = 0$ and hence $B = 0$.

A nonzero initial hidden vector also supplies the forcing term $Be^{-\mathrm{i}tD}y(0)$. It is additional data, not part of the convolution kernel. The retained checks cover the displayed kernel values, exact Gram factorization, and acceleration defect. The integral equation follows from the written elimination argument.

> **Read this as:** Memory is positive definite through a Gram factorization, not through pointwise signs. Exact bare local closure requires the coupling to vanish.

Plate VII.36 / STRICT POSITIVITY AND THE GROUND GAP

A positive transfer determines its Hamiltonian

Chapter VII: Theorem II.10.2, Proposition II.10.5.

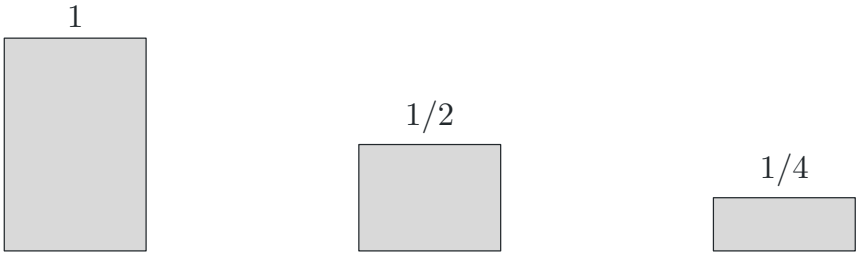
On $\mathcal{H} = \mathbb{C}^3$, take the strictly positive contraction

$$T = \operatorname{diag}(1, 1/2, 1/4), \qquad H = -\log T = \operatorname{diag}(0, \log 2, 2\log 2).$$

At unit Euclidean time, $T = e^{-H}$. The self-adjoint logarithm is unique because the logarithm acts on positive spectral values. At integer time $n \geq 0$,

$$T^n = \operatorname{diag}(1, 2^{-n}, 4^{-n}), \qquad P_0 = \operatorname{diag}(1, 0, 0), \qquad \|T^n - P_0\| = 2^{-n}.$$

The gap $\Delta = \log 2$ determines the rate $e^{-n\Delta}$ exactly in this example.



Strict positivity is an essential finite hypothesis

For $T' = \operatorname{diag}(1, 0)$, the operator is positive semidefinite, but there is no finite self-adjoint H' on all of $\mathbb{C}^2$ with $T' = e^{-H'}$: every finite eigenvalue of H' exponentiates to a strictly positive value. One may restrict to the positive support, but that changes the space on which the logarithm is defined.

For a family with nontrivial excited spectra, a uniform exponential rate needs a common positive lower bound on the gap. Ground multiplicity does not cause a problem: P_0 selects the entire eigenspace at the minimum energy. For a Hamiltonian whose minimum is $E_0 \neq 0$, the normalized transfer is $e^{-t(H - E_0 I)}$ before taking the limit.

Positivity is retained at several Euclidean times

For arbitrary vectors u_j and nonnegative integer times n_j, the reflected kernel obeys

$$\sum_{j,k} \langle u_j, T^{n_j + n_k} u_k \rangle = \left\| \sum_j T^{n_j} u_j \right\|^2 .$$

This is a finite Gram factorization. The retained run checks the integer semigroup and diagonal ground error for $n = 0, \ldots, 12$, and rejects a logarithm on the full space for the singular control. The written spectral argument justifies the logarithms and extends the functional calculus statement to all times.

> **Read this as:** A positive definite transfer can be logarithmically reconstructed. Ground convergence needs a declared normalization and gap, while continuum reconstruction still needs its own structure.

Plate VII.37 / FINITE HEAT DATA AND THE QUANTIFIERS

The dimension window has an order of limits

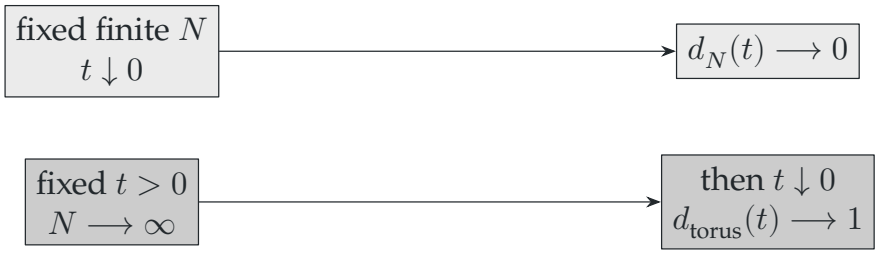
Chapter VII: Theorem II.11.5, Corollary II.11.8.

For the unit circle grid with N sites, the scaled eigenvalues are $\lambda_N(k) = 4N^2 \sin^2(\pi k/N)$. Put $Z_N(t) = \sum_k e^{-t\lambda_N(k)}$ and $d_N(t) = 2t \sum_k \lambda_N(k) e^{-t\lambda_N(k)}/Z_N(t)$.

At the single time $t = \log 2/16$, two small grids admit exact rational heat weights:

N	spectrum	Z_N	$\sum \lambda e^{-t\lambda}$	$d_N/\log 2$
2	$(0, 16)$	$3/2$	8	$2/3$
4	$(0, 32, 32, 64)$	$25/16$	20	$8/5$

These are unnormalized traces. Normalizing by N changes Z_N but leaves the logarithmic derivative at that fixed grid unchanged.



The theorem uses two separate choices

For each $\varepsilon > 0$, first choose a fixed positive time t_ε so that the continuum torus dimension differs from one by less than $\varepsilon/2$. Then choose N_ε so that every $N \geq N_\varepsilon$ differs from that continuum value by less than $\varepsilon/2$ at the *same time*. This proves one controlled intermediate time for all sufficiently fine grids.

It does not assert a uniform interval width from that argument, a moving time t_N with no extra estimates, or a dimension plateau at every scale. At fixed finite N, $d_N(t) \to 0$ as $t \downarrow 0$ and as $t \to \infty$. By contrast, first taking $N \to \infty$ at positive t and then taking $t \downarrow 0$ yields the continuum dimension.

Where the infinite tail is controlled

At a fixed time, the proof bounds centered modes below by $16k^2$ and above by $4\pi^2 k^2$. The summable Gaussian dominators control both the heat trace and its first moment. A finite grid table supplies neither those tails nor the quantifiers. The retained check verifies the two displayed spectra, rational weights, traces, and dimension coefficients. It also distinguishes the unscaled cycle with four sites from the scaled unit circle grid.

> **Read this as:** The strengthened window result proves convergence after fixing a suitable time. The exact table is a worked calculation, not its proof.

Plate VII.38 / THE IMAGE OF AN ISOMETRY IS NOT THE WHOLE LEVEL

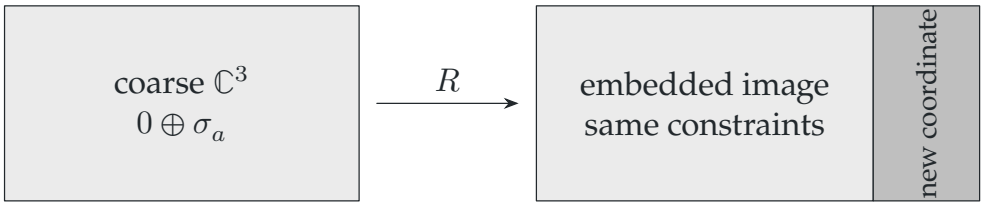
Transport constraints without hiding closure

Chapter VII: Theorem II.12.2, Theorem II.12.5, Proposition II.12.6.

On $\mathbb{C}^3 = \mathbb{C}\,|0\rangle \oplus \mathbb{C}^2$, use $C_a = 0 \oplus \sigma_a$ for $a = x, y, z$. The common kernel is $\mathbb{C}\,|0\rangle$, and the full algebra closes:

$$[C_x, C_y] = 2\mathrm{i}C_z, \qquad [C_y, C_z] = 2\mathrm{i}C_x, \qquad [C_z, C_x] = 2\mathrm{i}C_y.$$

These are uncompressed identities. They supply an actual closure test in the quotient by the constraint span, unlike the automatic compressed zero on the physical kernel.



closure on im R transfers, and the new complement needs its own test

Exact transport holds on the coarse image

Embed by the coordinate isometry $R : \mathbb{C}^3 \to \mathbb{C}^4$ and set $\widetilde{C}_a = C_a \oplus 1$. Then $\widetilde{C}_a R = R C_a$, so

$$[\widetilde{C}_a, \widetilde{C}_b]R = R[C_a, C_b].$$

The coarse physical vector maps into the fine common kernel. However, the same prescribed Pauli structure does *not* close on the whole fine level: on the added coordinate the commutator is zero, whereas $2\mathrm{i}\widetilde{C}_z$ is $2\mathrm{i}$. Exact transport constrains the image of R and leaves its orthogonal complement as separate data.

An approximate residual is also explicit

Perturb only $\widetilde{C}_x$ by $(1/10)\,|0\rangle\langle 0|$. For the unit coarse physical vector, the fine constraint residual is exactly $1/10$. The intertwining defect has the same norm. The general result for approximate transport controls these residuals and commutator errors using the defect sizes and a common bound on the constraint norms. A nonzero anomaly remains an error that must be controlled.

The retained checks verify all three coarse commutators, the isometry, each constraint intertwining, all nine commutator transports, and the residual. The added coordinate provides a counterexample to extending the identity on the image to the whole space. Unbounded constraints would additionally require a common invariant domain.

> **Read this as:** Closure is a statement about the operator algebra. Refinement intertwining transports it to the embedded image. The new degrees of freedom require their own control.

Plate VII.39 / EXACT EMBEDDINGS AND A SUMMABLE TAIL

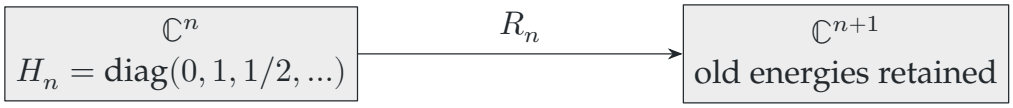
Two routes to a bounded Hamiltonian limit

Chapter VII: Theorem II.13.2, Theorem II.13.4, Corollary II.13.5.

For the exact route, let $\mathcal{H}_n = \mathbb{C}^n$ with coordinate inclusions and

$$H_n = \operatorname{diag}(0, 1, 1/2, \dots, 1/(n-1)), \qquad H_1 = (0).$$

The old energies are retained at every level, $H_{n+1}R_n = R_nH_n$, and $\|H_n\| \le 1$. On the Hilbert space of the direct limit $\ell^2(\mathbb{N}_0)$, the diagonal sequence defines a bounded self-adjoint H_∞. Projectors onto even addresses also intertwine and give a fixed complete system of two packets in the limit.



fixed $\mathbb{C}^2$: $H_n = \operatorname{diag}(0, 1 + 2^{-n})$
tail $\|H_n - H\| = 2^{-n}$

Compatibility and boundedness are different assumptions

The equally compatible tower $\operatorname{diag}(0, 1, 2, \dots, n-1)$ has growing norms. It does not satisfy the bounded direct limit theorem. Its natural diagonal operator needs a separately specified domain. Dropping the uniform norm hypothesis does not supply that domain argument.

For the norm route, work instead on the *single fixed space* $\mathbb{C}^2$ and set

$$H_n = \operatorname{diag}(0, 1 + 2^{-n}), \qquad H = \operatorname{diag}(0, 1).$$

Here $\|H_{n+1} - H_n\| = 2^{-(n+1)}$ and $\|H_n - H\| = 2^{-n}$. The summable tail supplies an error bound for every n. For $|t| \le T$, the paper's Duhamel estimate gives

$$\left\|e^{-\mathrm{i}tH_n} - e^{-\mathrm{i}tH}\right\| \le T2^{-n}.$$

The Euclidean counterpart holds on $0 \le t \le T$ because both Hamiltonians are nonnegative. If packet projectors also converge in norm, leakage operators converge uniformly on these time intervals.

Execution scope

Exact inclusions and projector compatibility are checked through level eight. The two-dimensional tail identities are checked for $n = 0, \dots, 12$. Their infinite conclusion uses the displayed geometric series and the theorem's analytic estimate. A finite prefix cannot certify an unspecified continuation of a tower.

> **Read this as:** Exact embedded compatibility uses a uniform norm bound. Approximate convergence uses a common space and summable defects. Keep the hypotheses of these two routes explicit.

Plate VII.40 / REFLECTION GRAMS, TRACES, AND FINAL SCOPE

Choose the topology that preserves the quantity

Chapter VII: Theorem II.13.6, Theorem II.13.8, Section II.14.

Let $q_n = 1/2 + 2^{-(n+2)}$ and define the scalar reflected kernel $C_n(j,k) = 1 + q_n^{j+k}$ on nonnegative integers. Each finite Gram matrix is a sum of two positive rank-one matrices, and pointwise convergence gives $C(j,k) = 1 + (1/2)^{j+k}$. For times $(0,1,2)$ and coefficients $(1,-2,1)$,

$$\sum_{j,k} v_j C_n(j,k) v_k = (1-q_n)^4 \geq 0.$$

This exact check illustrates how finite quadratic forms pass positivity to the limit.



positive trace-class control: norm convergence alone does not preserve a trace

Operator norm does not preserve trace in growing rank

On ℓ^2, let P_n select the first n coordinates and set $A_n = P_n/n$. Then $\|A_n\| = 1/n \to 0$ while $\operatorname{Tr} A_n = 1$ for every n. These positive trace-class operators show why the choice of topology matters. Their zero limit is not the heat operator of a finite Hamiltonian.

For transfers on two states $T_n = \operatorname{diag}(1, q_n) = e^{-H_n}$, with $H_n = \operatorname{diag}(0, -\log q_n)$ at unit time, the error in trace norm is $2^{-(n+2)}$ and the heat traces converge to $3/2$. The criterion for convergence of spectral dimension additionally assumes convergence of the first heat moments in trace norm and a nonzero limiting heat trace. Convergence of the heat trace alone does not determine that quotient.

The retained evidence and the remaining open targets

The extension includes retained exact checks over specified finite ranges in $\mathbb{Q}(\mathrm{i})$ using `Fraction` from Python's standard library. The original appendix's record of 180 checks, including symbolic derivatives, is also retained. The formal companion records the provenance of these checks. All model and vector sources remain supplied.

The formal correspondence lists nineteen finite items with their stated hypotheses. Its five original targets for continuum and physical reconstruction remain open. Part II supplies analytic proofs and criteria for bounded limits. These results and examples do not establish the unresolved physical existence claims.

Read this as: Use pointwise convergence for finite positivity tests, operator norm for bounded dynamics and constraints, and trace norm for heat traces. Every conclusion keeps its own hypotheses and evidence scope.

CHAPTER VIII

Packet Operator Algorithms

Exact compilation and GPU realization

The earlier packet constructions become inputs to exact compilation and numerical implementation.

Structured packet representations remove work before execution. Exact compilation connects algebraic structure, recompression, precision and stated hardware observations.

IN THIS CHAPTER

Mathematical text 814

Visual studies 897

Formal companion 2132

Sources and references 2684

Scope

Small-core optimality uses supplied ordered thin-SVD equations and actual compact ranges. General numerical SVD verification, unrestricted discovery and CUDA-kernel verification require further results.

The book then turns to arithmetic, beginning with Euler congruences, completion and dyadic geometry.

How to use this volume

The complete finite manuscript comes first. Six supplementary proof developments follow as Appendices A–F. A worked visual atlas closes the volume. The separate visual and Lean companions provide another reading route and an explicit evidence register.

Reading this chapter

An operator representation determines what a computation means before it determines how a device executes it. Chapter VIII develops that distinction through packet blocks, low rank corrections, graph support, preconditioning, precision, and finite compilation. Three exact examples introduce the main choices. The manuscript then gives their general statements and proofs.

The main argument keeps its local section and item numbers. Appendices A–F give the additional constructions and proofs for exact semantics, compilation, structured transforms, scheduling and small-core recompression. These developments make assumptions and intermediate objects explicit when a shorter main-text argument leaves them implicit.

The formal companion covers all 56 finite manuscript-proved items under their stated hypotheses. Its 392 supporting theorem declarations span 40 modules, including 19 reused quantum-packet declarations. The 66 construction audits and four classical inputs are tracked separately. The companion supplies the declaration map and the provenance of the recorded compiler evidence.

The source already includes positive precision budgets and caps, nonempty allocation sets, positive physical denominators and transform orders, valid band widths, actual positive square roots, reflexive reachability, and the empty-graph depth-zero case. Its arithmetic counts describe declared event models. Compiler stability uses solved child scores, and local recompilation assumes correct unaffected sibling caches. These hypotheses are part of the results.

The small-core recompression result starts from supplied ordered thin-SVD equations and handles arbitrary ambient competitors by compression. It does not establish SVD existence, a floating-point QR/SVD algorithm, or a canonical singular-values API identification. The other three classical inputs concern exact-arithmetic conjugate gradients, a stated floating-point model, and simultaneous spectral decomposition. A rounded action on varying inputs requires a uniform certificate before it can be treated as one fixed linear operator.

Representation, output, model, observation

A compressed action and an explicit matrix are different outputs. An additive cost model and an observed execution time are different evidence. A finite compiler optimum and unrestricted structure discovery are different problems. Keep the representation, output contract, mathematical model, and recorded measurement together when using a result.

All hardware times and speed ratios are recorded observations from the benchmark corpus. Missing source-value hashes and explicit correctness-reference fields limit these comparisons. CPU, graph-replay and two-stream rows retain their scope qualifications. In the negative-control figure, “matched baseline” identifies the recorded comparison baseline. It does not certify the full benchmark fairness contract.

Appendices A–F complete the finite development under its stated hypotheses. Three broader targets remain open: unrestricted discovery, compilation with nonlocal overlapping models, and a packet-native communication theorem for a specified architecture. The exact examples and visual atlas illustrate finite mathematics. They add no hardware performance claim and do not replace the proofs.

Abstract

A packet operator is a linear map written between the members of a finite direct sum. We study the resulting computational representation before choosing a processor or kernel. Exact formulas are given for block sparse application, block diagonal plus low rank algebra, Woodbury inversion, support propagation, block preconditioning, and packetwise perturbation. Products of low rank corrections admit optimal recompression through a small singular value problem, with explicit error propagation through a product chain. A rooted candidate decomposition supports an exact compiler when representation cost and error are additive. Dynamic programming gives the optimal Lagrangian pruning, an exact quantized error budget algorithm, and a local update rule. Discrete precision assignment becomes a multiple choice budget problem. Arithmetic, storage, data movement, launch count, discovery, transfer, and materialization are kept as separate costs. A preserved corpus of 154 recorded CPU and GPU measurements illustrates regular packet gains, precision frontiers, crossover behavior, and retained negative controls. Its largest reported structured ratio is 14.89 against a device resident dense GPU calculation, with archived matching-metadata limitations stated explicitly. The chapter proves finite algebraic and optimization statements. Hardware measurements remain specific to their recorded environment, and no generic dense matrix exponent or universal discovery algorithm is claimed.

Three exact models before the general theory

These models use exact integers and rational numbers. Their costs and integer errors are declared illustrative quantities. The figures display the computed records. They do not report timings or identify actual hardware precision types.

A low rank action and its inverse

Set

$$D = \operatorname{diag}(2, 3, 4, 5), \qquad u = (1, 0, 1, 0)^T, \qquad v = (1, 1, 0, 0)^T,$$

and define $A = D + uv^T$. Its entries are shown in the figure. The same operator can be stored by its diagonal and two thin vectors rather than by every matrix entry. For $x = (1, 2, 3, 4)^T$, first form $v^T x = 3$. Then

$$Dx = (2, 6, 12, 20)^T, \qquad u(v^T x) = (3, 0, 3, 0)^T,$$

so $Ax = (5, 6, 15, 20)^T$. Only the first and third output coordinates receive the correction.

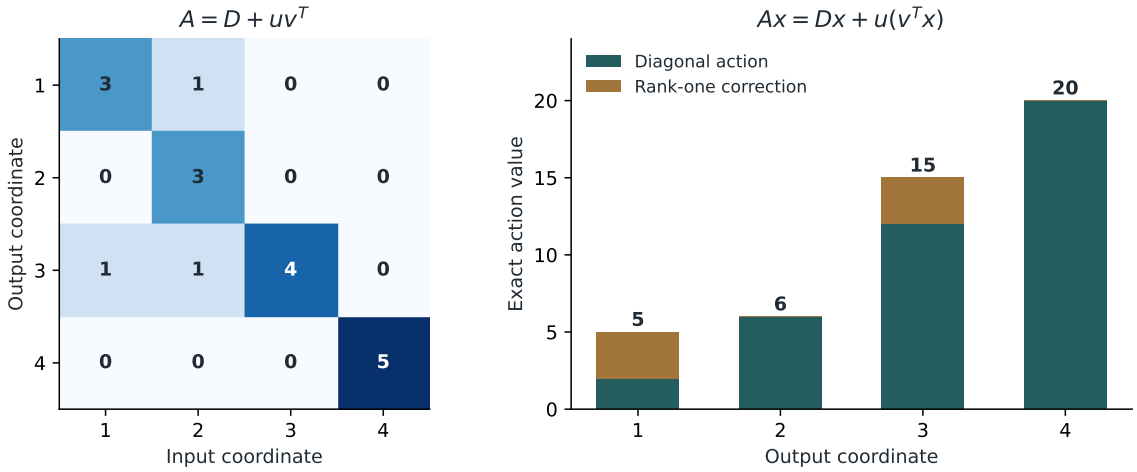


Figure 1. Exact finite action of a diagonal operator plus one thin correction. The bars display the same action as the explicit matrix.

The inverse starts with the same representation. Write $y = Ax$. Since D is invertible,

$$D^{-1}u = (1/2, 0, 1/4, 0)^T, \qquad 1 + v^T D^{-1} u = 3/2 \neq 0.$$

The scalar Woodbury core is therefore invertible. Its correction coefficient is

$$\frac{v^T D^{-1} y}{1 + v^T D^{-1} u} = \frac{9/2}{3/2} = 3.$$

Subtracting that correction recovers the original input:

$$D^{-1}y - 3D^{-1}u = (5/2, 2, 15/4, 4)^T - (3/2, 0, 3/4, 0)^T = x.$$

> **What the output contract preserves**
>
> The diagonal and thin vectors use twelve scalar slots, compared with sixteen for the explicit matrix. This small storage count assumes the full thin vectors are stored. It does not predict elapsed time. A compressed action returns Ax. Materializing all entries of A produces another output and incurs its own work. The general formulas appear in theorems 2.3 and 5.3.

A budget can select a point that no Lagrangian selects

Consider a rooted candidate tree. The root r has children a, b, and a has children a_1, a_2. Split overheads are zero. Each retained node has the following declared cost and integer error:

Node	r	a	b	a_1	a_2
Cost	1	2	1	3	1
Error	6	4	0	1	0

Exactly three prunings are possible:

Selected nodes	Total cost	Total error
r	1	6
a, b	3	4
a_1, a_2, b	5	1

For a Lagrangian multiplier $\lambda \geq 0$, their objectives are $1 + 6\lambda$, $3 + 4\lambda$, and $5 + \lambda$. The middle pruning can be optimal only if

$$3 + 4\lambda \leq 1 + 6\lambda \quad \text{and} \quad 3 + 4\lambda \leq 5 + \lambda.$$

These inequalities require $\lambda \geq 1$ and $\lambda \leq 2/3$ simultaneously. There is no such multiplier. The root and full split exchange optimality at $\lambda = 4/5$, where both have objective $29/5$.

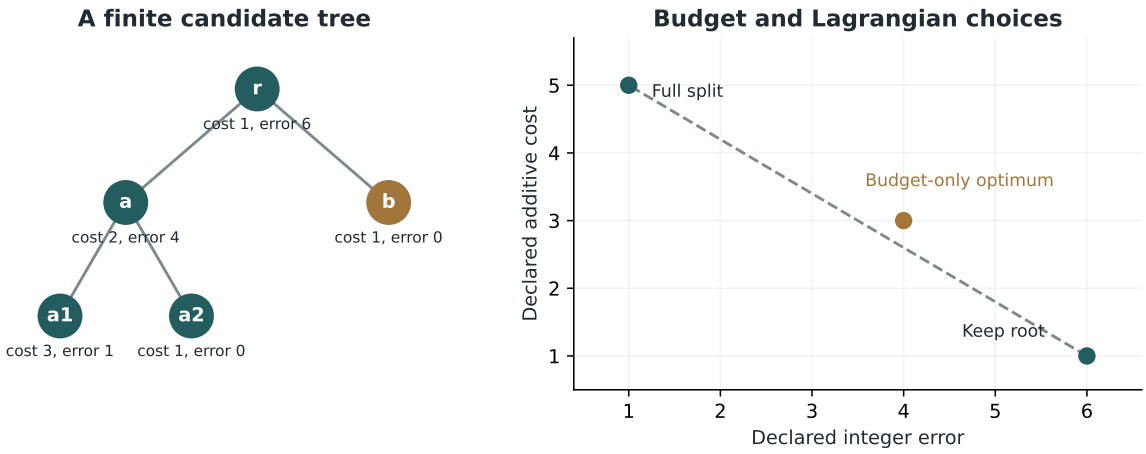


Figure 2. The finite candidate tree has three prunings. The middle point is optimal under error budget four, although no nonnegative Lagrangian multiplier selects it.

The budget problem still selects that middle pruning. Under the constraint err ≤ 4, the root is infeasible. The two feasible costs are three and five, so the minimum is three. Under budget one, only the full split is feasible. Under budget six, the root is cheapest. Thus the budget program adds something that changing λ cannot recover.

The dynamic recurrence gives the same Lagrangian choices without listing prunings. At a, it compares $2 + 4\lambda$ with $4 + \lambda$. At r, it compares $1 + 6\lambda$ with $1 + F_\lambda(a)$. Its exactness follows from the finite tree and additive model in theorems 9.4 and 9.8. Overlapping decompositions or nonlocal execution costs need another argument.

Discrete precision follows the sensitive edge

Suppose three packet edges each admit a fast option F and an accurate option A. The names describe the supplied model, rather than any actual scalar type. Exactly one option is chosen on each edge:

	F		A	
Edge	Cost	Error	Cost	Error
1	1	5	4	0
2	1	2	3	0
3	1	1	3	0

The all-fast assignment FFF has cost three and error eight. It exceeds budget three. Changing only edge one to A gives AFF, with cost $4 + 1 + 1 = 6$ and error $0 + 2 + 1 = 3$. Changing only edges two and three gives FAA, with cost seven and error five. That choice still exceeds the budget.

All eight assignments can be checked directly:

Assignment	Cost	Error	Assignment	Cost	Error
FFF	3	8	AFF	6	3
FFA	5	7	AFA	8	2
FAF	5	6	AAF	8	1
FAA	7	5	AAA	10	0

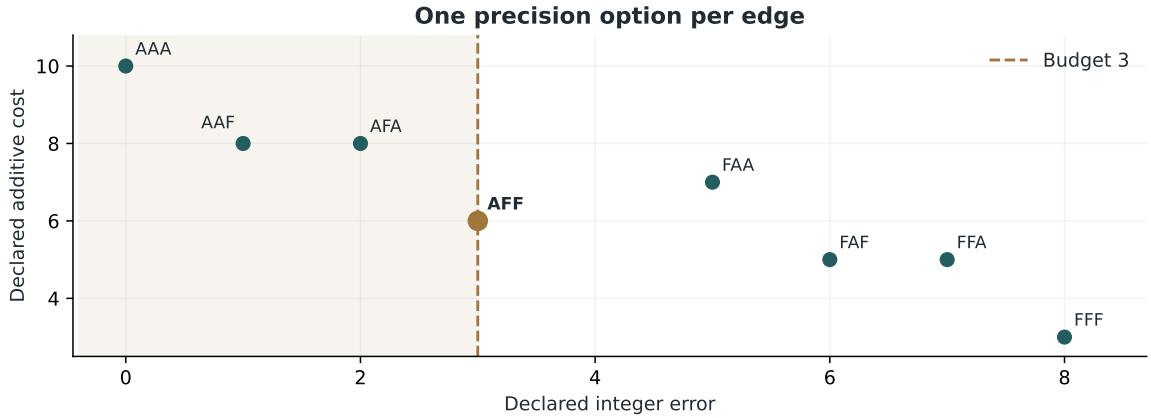


Figure 3. The shaded region satisfies the supplied error budget. AFF is the unique minimum-cost assignment in that region. The model protects the first edge before spending on less sensitive edges.

The exact recurrence reaches the same result. After edge one, only $(\mathrm{err}, \mathrm{cost}) = (5, 1)$ and $(0, 4)$ are reachable. After edge two, the reachable states are $(7, 2)$, $(5, 4)$, $(2, 5)$, and $(0, 7)$. Extending these with either option on edge three gives the final eight assignments. Among states of error at most three, the lowest cost is six.

The certificate belongs to the model

The integer error weights must be supplied by a stated certificate or quantization rule. The recurrence solves that finite model exactly. Kernel accumulation, overflow, underflow, scaling, and shape support require their own assumptions before an actual implementation can use the result. The precise recurrence is theorem 8.7.

The complete rational records, dynamic tables, and five rejected false controls are reproduced by `generate_monograph_examples.py`. Figure generation converts those records to plotting coordinates only after the exact checks pass.

The finite manuscript

The main reading path follows exact packet semantics, algebra, support, numerical certificates, and optimization before reaching the archived measurement corpus. The cost and error assumptions determine where each algorithm is exact.

1 Introduction

A matrix can look dense while its action is not. Independent diagonal blocks, sparse transfers, low rank couplings, tensor factors, and transform coordinates can remove most of the arithmetic and storage before a kernel is launched. The gain is mathematical first. A device can exploit only the structure that the representation exposes.

This article begins with a finite direct sum

$$X = \bigoplus_{\alpha \in I} X_\alpha$$

and a complete family of coordinate projectors

$$P_\alpha P_\beta = \delta_{\alpha\beta} P_\alpha, \qquad \sum_{\alpha \in I} P_\alpha = I.$$

An operator has packet blocks

$$A_{\beta\alpha} = P_\beta A P_\alpha.$$

The active addresses form a directed graph. Block dimensions, block types, approximation tolerances, and output requirements turn this graph into a computational intermediate representation.

The underlying linear algebra is classical. Direct and iterative methods are treated in [14, 25, 9]. Low rank approximation and updates are governed by singular value decomposition and the Sherman–Morrison–Woodbury identity [11, 22, 15, 4]. Floating point analysis follows the standard model in [17, 20]. Randomized discovery of low rank structure is developed in [16, 21]. Communication and attainable performance are shaped by data movement and arithmetic intensity [19, 27, 2, 12].

The first task is semantic. A compressed representation must specify the operator that it denotes. It must also specify what an algorithm returns. A product represented by block and low rank factors is not the same output as an explicit matrix. A timing comparison is valid only after the output contract and timing scope agree.

The second task is algebraic. Block diagonal plus low rank operators are closed under multiplication before recompression. Their unrecompressed rank grows additively. Woodbury inversion reduces a large inverse to packet solves and one small dense inverse. The low rank factors can then be recompressed by orthogonalizing them and taking the singular value decomposition of a small core. This produces the best truncated approximation in spectral and Frobenius norm.

The third task is combinatorial. Sparse transfer graphs predict possible support under powers, polynomials, and Schur elimination. A directed acyclic graph gives a finite path expansion. A block triangular solve can be scheduled by topological level. Graph regularity matters because equal shapes can be batched, while irregular edges require indexing and accumulation.

The fourth task is numerical. Packet perturbations can be certified by their block sensitivities. A continuous model assigns tolerances through a convex calculation. Upper bounds on available tolerances

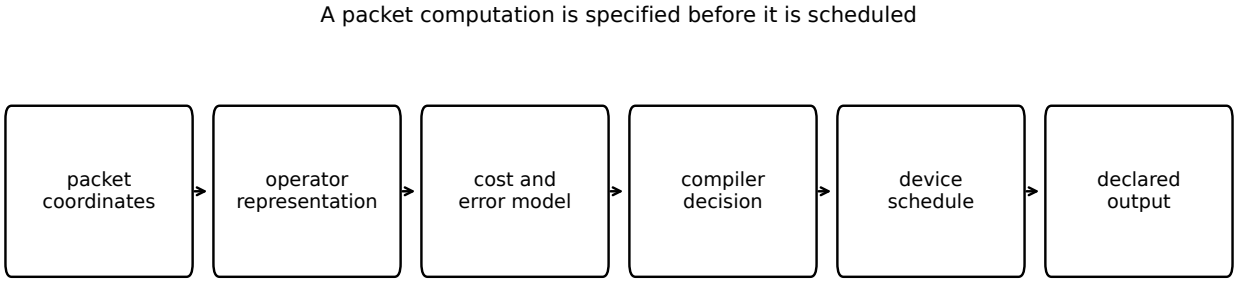


Figure 4. The representation and the output contract are fixed before the device schedule is chosen.

produce a clipped allocation. A finite set of data types gives a multiple choice budget problem that can be solved exactly after the errors are quantized.

The central computational result concerns compilation. Suppose a finite rooted tree records candidate refinements. Each node may be kept as one representation or split into its children. If cost and error are additive across the selected leaves, the optimal pruning is found exactly by dynamic programming. A quantized error budget has a second exact dynamic program. The decision is stable when the score gap exceeds the perturbation in the local models. A change inside one subtree requires recomputation only along the ancestor chain. These results prove a restricted packet compiler. They do not solve unrestricted discovery or arbitrary device scheduling.

The measured part uses a sanitized corpus collected on one recorded CPU and one NVIDIA GeForce RTX 3070 Ti Laptop GPU. The preserved corpus contains 154 recorded rows from sixteen suites. The timings are inherited observations. They are not new timing measurements. Regular equal block operators and block diagonal plus low rank operators show large gains. Recursive HODLR, Strassen multiplication, iterative refinement, a small Walsh–Hadamard transform, and an attempted two stream overlap provide negative controls. These rows are retained because structure does not guarantee speed.

The article separates four sources of gain:

$$\text{representation}, \qquad \text{device}, \qquad \text{kernel}, \qquad \text{amortization}.$$

A single speedup number can mix all four. The benchmark protocol therefore requires case identity, output contract, timing scope, source precision, correctness reference, discovery time, compilation time, and reuse count. The archived rows do not satisfy every field of this ideal contract: source-value hashes and explicit correctness-reference fields are absent, while setup, discovery, and reuse metadata are sparse. Recorded ratios are consequently descriptive comparisons under their stated scopes, rather than fair matches with complete provenance.

The mathematical statements are finite. The empirical statements apply only to the recorded environment and generated cases. The article does not claim a new generic matrix multiplication exponent. Fast matrix multiplication and its stability form a separate theory [26, 10]. It also does not claim that every structured matrix has a profitable GPU realization.

2 Packet operator representations

2.1 Exact semantics

Definition 2.1 (Packet layout)**.** A packet layout is a finite family of finite dimensional Hilbert spaces

$$\mathcal{P} = (X_\alpha)_{\alpha \in I}$$

with direct sum

$$X = \bigoplus_{\alpha \in I} X_\alpha.$$

Write

$$n_\alpha = \dim X_\alpha, \qquad n = \sum_{\alpha \in I} n_\alpha.$$

The coordinate inclusion and projection are $\iota_\alpha : X_\alpha \to X$ and $\pi_\alpha : X \to X_\alpha$. The ambient coordinate projector is $P_\alpha = \iota_\alpha \pi_\alpha$.

Definition 2.2 (Packet operator representation)**.** A finite packet operator representation consists of

$$\mathsf{IR} = (\mathcal{P}, E, (B_{\beta\alpha})_{(\beta,\alpha)\in E}),$$

where

$$E \subseteq I \times I$$

and

$$B_{\beta\alpha} : X_\alpha \longrightarrow X_\beta.$$

Local blocks are identified with their ambient extensions by inclusion and projection. In particular, the notation $P_\beta A P_\alpha$ denotes either the ambient block or its restriction $X_\alpha \to X_\beta$ according to the displayed domain, and $\iota_\beta B_{\beta\alpha} \pi_\alpha$ is the corresponding ambient extension. Its evaluation is

$$\operatorname{eval}(\mathsf{IR}) = \sum_{(\beta,\alpha)\in E} \iota_\beta B_{\beta\alpha} \pi_\alpha,$$

where $\iota_\beta : X_\beta \to X$ is the coordinate inclusion.

Theorem 2.3 (Exact evaluation)**.** *Let $A \in \operatorname{End}(X)$. Set*

$$B_{\beta\alpha} = \pi_\beta A \iota_\alpha$$

and let E contain exactly the nonzero blocks. Then

$$\boxed{A = \sum_{(\beta,\alpha)\in E} P_\beta A P_\alpha.}$$

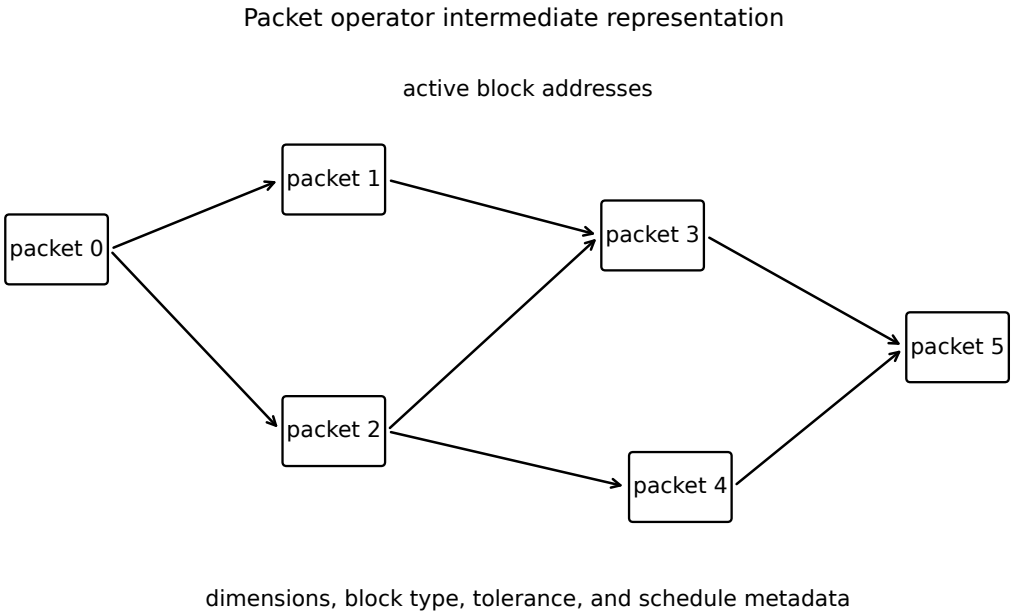


Figure 5. A packet intermediate representation stores active block addresses together with dimensions, block types, tolerances, and scheduling metadata.

For every matrix of right hand sides Z,

$$AZ = \sum_{(\beta,\alpha)\in E} \iota_\beta B_{\beta\alpha} \pi_\alpha Z.$$

Proof. Insert the identity on both sides of A:

$$A = \left(\sum_\beta P_\beta\right) A \left(\sum_\alpha P_\alpha\right) = \sum_{\alpha,\beta} P_\beta A P_\alpha.$$

The zero blocks can be removed. Applying the identity to Z gives the second formula. □

Definition 2.4 (Support graph)**.** The support graph G_A has vertex set I. It has a directed edge

$$\alpha \longrightarrow \beta$$

when $P_\beta A P_\alpha \neq 0$.

Theorem 2.5 (Block composition). *Let $A, B \in \mathrm{End}(X)$. Then*

$$\boxed{(AB)_{\gamma\alpha} = \sum_{\beta \in I} A_{\gamma\beta} B_{\beta\alpha}.}$$

Consequently,

$$\mathrm{supp}(AB) \subseteq \mathrm{supp}(A) \circ \mathrm{supp}(B),$$

where relational composition is defined by

$$(\gamma, \alpha) \in E \circ F$$

when some β satisfies

$$(\gamma, \beta) \in E, \qquad (\beta, \alpha) \in F.$$

Proof. Insert the packet identity between A and B:

$$P_\gamma A B P_\alpha = \sum_\beta P_\gamma A P_\beta B P_\alpha.$$

A summand can be nonzero only when both block addresses are active. □

Corollary 2.6 (Adjoint and tensor semantics). *The adjoint representation has blocks*

$$(A^*)_{\alpha\beta} = A^*_{\beta\alpha}.$$

If A and B are packet resolved on X and Y, then

$$(A \otimes B)_{(\beta,\delta),(\alpha,\gamma)} = A_{\beta\alpha} \otimes B_{\delta\gamma}.$$

Proof. Both identities follow by inserting the corresponding coordinate projectors and using

$$(P_\beta A P_\alpha)^* = P_\alpha A^* P_\beta.$$

The tensor identity follows from

$$(P_\beta \otimes Q_\delta)(A \otimes B)(P_\alpha \otimes Q_\gamma) = A_{\beta\alpha} \otimes B_{\delta\gamma}.$$

□

2.2 Output contracts

Definition 2.7 (Output contract). An output contract specifies the mathematical object returned by an algorithm. The contract may require an explicit vector, an explicit matrix, block factors, low rank factors, spectral factors, or an implicit operator with a declared application rule.

Definition 2.8 (Timing scope)**.** A timing scope states which costs are included. Possible costs include structure discovery, compilation, host transfer, device transfer, kernel execution, graph capture, graph replay, factor reconstruction, and explicit materialization.

Proposition 2.9 (Contract conversion)**.** *Suppose one algorithm returns a compressed representation R and another returns an explicit object Y. A direct timing comparison is valid only if*

$$\mathrm{decode}(R) = Y$$

and the time required by decode *is included, or if the downstream task consumes R directly.*

Proof. Without decoding, the two procedures return different mathematical objects. A timing ratio between different outputs does not compare implementations of the same map. Including the decoding cost gives two procedures with the same output. If the next operation consumes R, decoding is not part of the declared computation. □

Remark 2.10. A compressed product and an explicit matrix product may both be exact. They still have different contracts. This distinction becomes decisive when materialization costs n^2 entries.

3 Arithmetic, storage, and data movement

We count one scalar multiplication and one scalar addition as two modeled floating point operations. These counts describe a declared event model for a representation. They do not predict wall time without a memory and launch model. A multiplication with accumulation charges two scalar operations even for the first coefficient. Thin corrections accumulate into an already computed diagonal or sparse result. Index metadata, temporary workspace, launch overhead, discovery, transfer, and materialization are accounted for separately.

Theorem 3.1 (Dense and packet application counts)**.** *A dense n by n matrix applied to q right hand sides has modeled cost*

$$\boxed{\mathrm{flop}_{\mathrm{dense}} = 2n^2q.}$$

A packet representation with active edge set E has modeled cost

$$\boxed{\mathrm{flop}_{\mathrm{packet}} = 2q \sum_{(\beta,\alpha)\in E} n_\beta n_\alpha.}$$

It stores

$$\boxed{M_{\mathrm{packet}} = \sum_{(\beta,\alpha)\in E} n_\beta n_\alpha}$$

scalar entries.

Proof. The active block $B_{\beta\alpha}$ is an n_β by n_α matrix. Applying it to an n_α by q input costs $2n_\beta n_\alpha q$ modeled operations and stores $n_\beta n_\alpha$ scalars. Summation over the active addresses gives both formulas. □

Corollary 3.2 (Equal diagonal packets)**.** *Let $p \geq 1$ and $b \geq 1$ be integers and $n = pb$. Suppose n is divided into p equal diagonal packets of width b. Then*

$$\mathrm{flop}_{\mathrm{block}} = 2qnb, \qquad M_{\mathrm{block}} = nb.$$

The following factor describes the dense-to-block storage reduction and, when $q \geq 1$, the arithmetic reduction:

$$\boxed{\frac{n}{b}.}$$

Proof. There are n/b active diagonal blocks. Each contains b^2 entries. □

Definition 3.3 (Block diagonal plus low rank operator)**.** A block diagonal plus low rank operator has the form

$$A = D + UV^*,$$

where

$$D = \bigoplus_{\alpha \in I} D_\alpha, \qquad U, V \in \mathbb{C}^{n \times r}.$$

The abbreviation BDLR will be used below.

Theorem 3.4 (BDLR application and storage)**.** *For q right hand sides,*

$$AZ = DZ + U(V^*Z)$$

has modeled cost

$$\boxed{2q\left(\sum_\alpha n_\alpha^2 + 2nr\right).}$$

The representation stores

$$\boxed{\sum_\alpha n_\alpha^2 + 2nr}$$

scalars. For $n = pb$, with integers $p, b \geq 1$, the following factor describes the ideal dense storage reduction and, when $q \geq 1$, the arithmetic reduction:

$$\boxed{\frac{n}{b + 2r}.}$$

Proof. The block diagonal product costs

$$2q \sum_\alpha n_\alpha^2.$$

The products V^*Z and $U(V^*Z)$ cost $2nrq$ each. Storage consists of the diagonal blocks and the two low rank factors. For equal packets,

$$\sum_\alpha n_\alpha^2 = nb.$$

□

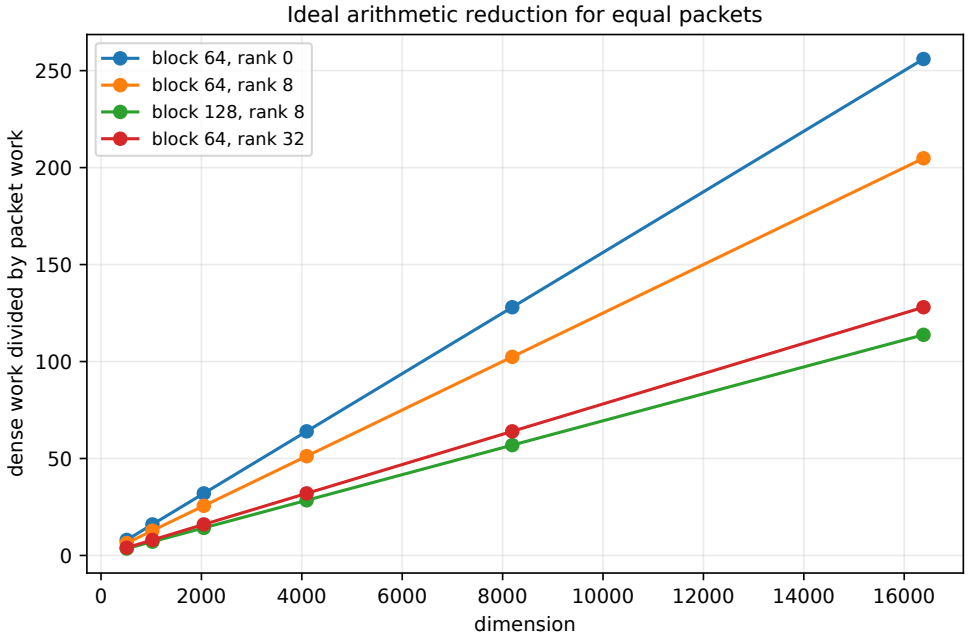


Figure 6. The ideal reduction grows with dimension only when packet width and transfer rank remain controlled.

Definition 3.5 (One pass streaming model)**.** Fix a scalar size s bytes. For every active block, the one pass streaming model reads the block once, reads its source packet once, and reads and writes its target packet once. It does not assume cache reuse across edges.

Proposition 3.6 (Streaming byte count)**.** *Under definition 3.5, applying a packet operator to q right hand sides moves*

$$\boxed{B_{\text{stream}} = s \sum_{(\beta,\alpha)\in E} \left(n_\beta n_\alpha + n_\alpha q + 2n_\beta q\right)}$$

bytes.

Proof. The three terms count the matrix block, the source values, and the read plus write traffic for the target accumulation. The model counts these quantities independently for every active edge. □

Definition 3.7 (Arithmetic intensity)**.** When the modeled byte traffic is strictly positive, the modeled arithmetic intensity is

$$\mathcal{I} = \frac{\text{flop}}{\text{bytes}}.$$

Theorem 3.8 (Roofline lower bound)**.** *Suppose a device sustains at most $P > 0$ modeled operations per second and at most $W > 0$ bytes per second. A computation that requires F operations and B bytes satisfies*

$$\boxed{T \geq \max\left\{\frac{F}{P}, \frac{B}{W}\right\}.}$$

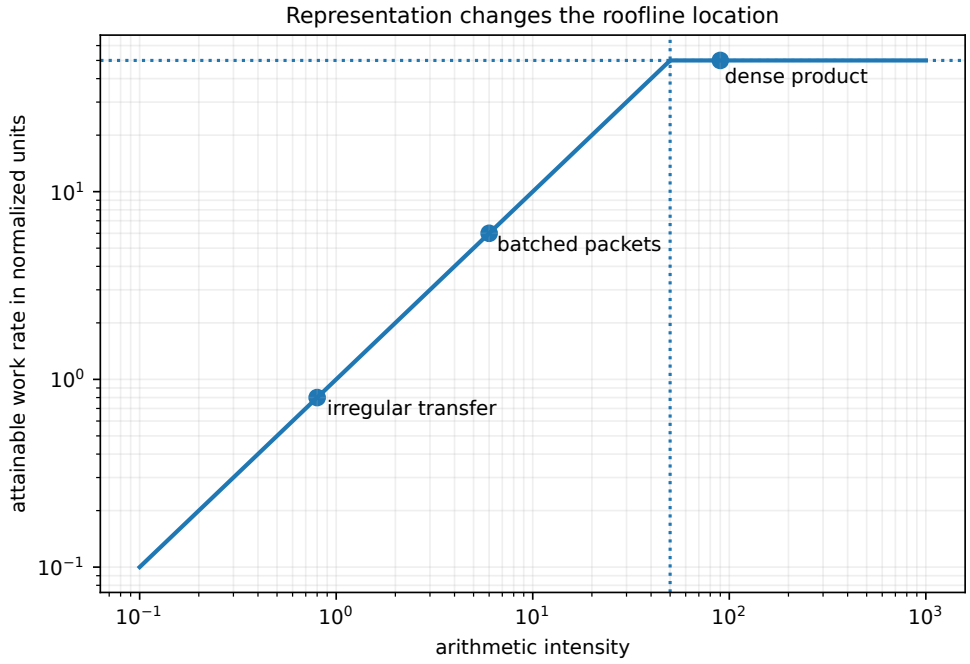


Figure 7. A representation can move the same mathematical problem between memory limited and compute limited regimes.

For positive elapsed time and positive byte traffic, its work rate is at most

$$\min\{P, W\mathcal{I}\}.$$

Proof. The operation capacity gives $T \geq F/P$. The bandwidth capacity gives $T \geq B/W$. Both restrictions hold at once. □

The roofline model is a guide rather than an exact execution law [27]. Packet structure changes both F and B. An irregular representation may remove arithmetic while lowering intensity enough to become bandwidth or launch limited.

Corollary 3.9 (Capacity threshold)**.** *Let a device have available memory $M \geq 0$ bytes and scalar size $s > 0$, and let $n \geq 0$. The matrix representation alone can fit only when*

$$n \leq \sqrt{M/s}.$$

The equal packet BDLR representation alone fits when

$$n(b + 2r)s \leq M.$$

Proof. Substitute the dense and BDLR storage counts into the capacity inequality. □

Proposition 3.10 (Materialization cost)**.** *Materializing $D + UV^*$ as an explicit matrix requires storage for n^2 scalars and modeled arithmetic*

$$2n^2r + O(n^2)$$

when the low rank correction is formed by dense multiplication.

Proof. The product UV^* multiplies an n by r matrix by an r by n matrix. Adding the diagonal part changes only the lower order term. □

4 Structured packet primitives

The packet representation can store different exact local operator types. This section records the finite formulas used by several measured suites. The results concern representation and arithmetic. They do not assert that one implementation is fastest at every size.

Proposition 4.1 (Equal block banded count)**.** *Let $p, b \geq 1$ be integers, $n = pb$, and let $p = n/b$ equal packets have width b. Suppose a block is active precisely when*

$$|i - j| \leq w, \qquad 0 \leq w \leq p - 1.$$

The number of active blocks is

$$\boxed{N_{p,w} = p(2w + 1) - w(w + 1).}$$

For q right hand sides, the modeled application cost and scalar storage are

$$\boxed{2qb^2N_{p,w} \quad \textit{and} \quad b^2N_{p,w}.}$$

Proof. The diagonal contributes p blocks. At offset k, where $1 \leq k \leq w$, the upper and lower block diagonals contribute $2(p - k)$ blocks. Therefore

$$p + 2\sum_{k=1}^{w}(p - k) = p(2w + 1) - w(w + 1).$$

The cost and storage follow from theorem 3.1. □

Proposition 4.2 (Sparse plus low rank application)**.** *Let*

$$A = S + UV^*,$$

where S has z stored scalar nonzeros and $U, V \in \mathbb{C}^{n\times r}$. In the model that charges two operations per stored sparse entry and right hand side, applying A to q right hand sides costs

$$\boxed{2q(z + 2nr)}$$

modeled operations and stores

$$\boxed{z + 2nr}$$

scalar values, apart from sparse index metadata.

Proof. The sparse product costs $2zq$. The two dense thin products cost $2nrq$ each. The stored values are the nonzeros of S and the entries of the two factors. □

Theorem 4.3 (Circulant spectral representation). *Let $n \geq 1$ be an integer and $c \in \mathbb{C}^n$, with indices taken modulo n, and define*

$$C(c)_{jk} = c_{j-k}.$$

Let F be the unitary discrete Fourier matrix

$$F_{jk} = \frac{1}{\sqrt{n}} \exp\left(-\frac{2\pi \mathrm{i} jk}{n}\right).$$

Then

$$\boxed{C(c) = F^* \operatorname{diag}(\sqrt{n}\, Fc) F.}$$

The operator is determined by n scalars and can be applied through two Fourier transforms and one pointwise product.

Proof. For each frequency k, let

$$v_j^{(k)} = \frac{1}{\sqrt{n}} \exp\left(\frac{2\pi \mathrm{i} jk}{n}\right).$$

Then

$$(C(c)v^{(k)})_j = \sum_{r=0}^{n-1} c_r v_{j-r}^{(k)} = \left(\sum_{r=0}^{n-1} c_r \exp\left(-\frac{2\pi \mathrm{i} rk}{n}\right)\right) v_j^{(k)}.$$

The eigenvalue is $\sqrt{n}(Fc)_k$. The vectors $v^{(k)}$ are the columns of F^*, which gives the formula. □

The fast Fourier transform evaluates the two transforms in $O(n \log n)$ arithmetic for the usual composite lengths [7]. This is an algorithmic factorization of the exact identity in theorem 4.3.

Corollary 4.4 (Toeplitz circulant embedding). *Let $n \geq 1$ be an integer and $T \in \mathbb{C}^{n \times n}$ be Toeplitz, with first column*

$$(c_0, c_1, \dots, c_{n-1})^T$$

and first row

$$(c_0, r_1, \dots, r_{n-1}).$$

Set

$$g = (c_0, c_1, \dots, c_{n-1}, 0, r_{n-1}, \dots, r_1)^T \in \mathbb{C}^{2n}$$

and append n zeros to $x \in \mathbb{C}^n$. The first n entries of the circulant product $C(g)(x, 0)^T$ equal Tx.

Proof. For $0 \leq j < n$, the circular convolution sum separates into indices $k \leq j$, where the coefficient is c_{j-k}, and indices $k > j$, where the wrapped coefficient is r_{k-j}. The padded coordinates do not contribute. The resulting sum is exactly the jth Toeplitz row applied to x. □

Theorem 4.5 (Kronecker modewise application). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, and $X \in \mathbb{C}^{n\times m}$. With column major vectorization,*

$$\boxed{(A \otimes B)\,\mathrm{vec}(X) = \mathrm{vec}(BXA^T).}$$

The right side has modeled cost

$$\boxed{2n^2m + 2nm^2}$$

and stores $n^2 + m^2$ factor entries instead of n^2m^2 entries.

Proof. The (i, j) entry of BXA^T is

$$\sum_{k=1}^{n}\sum_{\ell=1}^{m} B_{ik}X_{k\ell}A_{j\ell}.$$

This is the coordinate formula for $(A \otimes B)\,\mathrm{vec}(X)$. Multiplying first by B and then by A^T gives the stated count. □

Corollary 4.6 (Equal tensor factors). *For integers $k, d \geq 0$, a product of k square factors of order d, acting on a tensor with d^k entries, can be applied by k modewise contractions, reusing intermediate slices, with modeled cost*

$$\boxed{2kd^{k+1}}$$

and factor storage kd^2.

Proof. For $k = 0$ there are no contractions and both modeled work and factor storage are zero. For $k \geq 1$, one mode has d^{k-1} fibers of length d. Multiplying each fiber by a d by d matrix costs $2d^2$, so one mode costs $2d^{k+1}$. Sum over the k modes. □

Theorem 4.7 (Walsh–Hadamard recursion). *Let $N = 2^m$, and define*

$$H_1 = [1], \qquad H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}.$$

Then

$$H_N H_N^T = NI.$$

The product $H_N x$ can be evaluated in exactly $N \log_2 N$ additions or subtractions through the recursion.

Proof. The orthogonality identity follows by induction from block multiplication. One recursive stage forms $x_0 + x_1$ and $x_0 - x_1$, which uses N additions or subtractions, and then applies two transforms of size $N/2$. Thus

$$T(N) = 2T(N/2) + N, \qquad T(1) = 0,$$

whose solution is $T(N) = N \log_2 N$. □

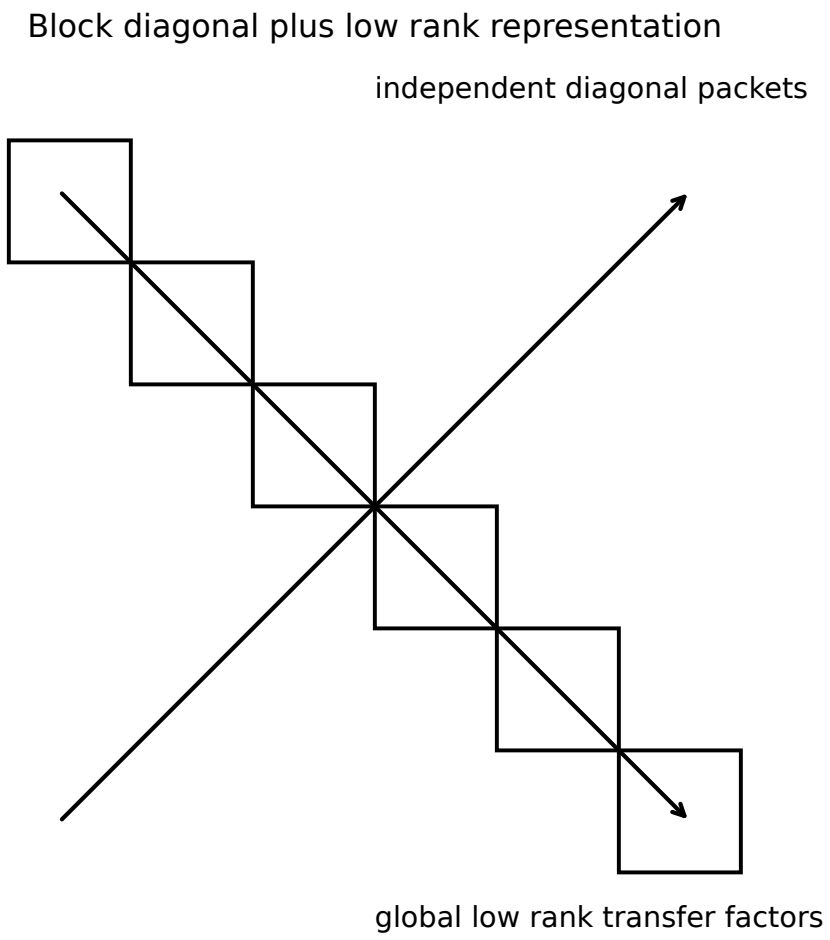


Figure 8. A regular diagonal packet structure is coupled by global low rank factors.

5 Exact low rank algebra and recompression

5.1 Products and inverses

Theorem 5.1 (Exact BDLR product). *Let*

$$A = D_A + U_A V_A^*, \qquad B = D_B + U_B V_B^*$$

use the same diagonal packet layout. Then

$$AB = D_A D_B + U_C V_C^*,$$

where

$$U_C = \begin{bmatrix} D_A U_B + U_A (V_A^* U_B) & U_A \end{bmatrix}$$

and

$$V_C = \begin{bmatrix} V_B & D_B^* V_A \end{bmatrix}.$$

The unrecompressed correction rank is at most $r_A + r_B$.

Proof. Expand the product:

$$AB = D_A D_B + D_A U_B V_B^* + U_A V_A^* D_B + U_A (V_A^* U_B) V_B^*.$$

The second and fourth terms share the right factor V_B^*. The third term equals

$$U_A (D_B^* V_A)^*.$$

Concatenation gives the stated factors. □

Corollary 5.2 (Rank growth in a product chain)**.** *Let*

$$A_j = D_j + U_j V_j^*, \qquad j = 1, \dots, m,$$

share one diagonal packet layout. Before recompression, the product

$$A_m \cdots A_1$$

has a BDLR representation whose correction rank is at most

$$\boxed{\sum_{j=1}^{m} r_j.}$$

Proof. Apply theorem 5.1 inductively. Each new factor adds at most its own correction rank. □

Theorem 5.3 (Woodbury inversion)**.** *Assume D and*

$$S = I + V^* D^{-1} U$$

are invertible. Then

$$\boxed{(D + UV^*)^{-1} = D^{-1} - D^{-1} U S^{-1} V^* D^{-1}.}$$

Proof. Multiply the proposed inverse by $D + UV^*$. The coefficient of $V^* D^{-1}$ is

$$U - U S^{-1} (I + V^* D^{-1} U) = 0.$$

The remaining term is the identity. The right inverse calculation is the same. □

The formula and its history are discussed in [15]. It reduces one large solve to packet solves with D and one dense solve with the small matrix S.

Corollary 5.4 (Inverse norm bound)**.** *Under the hypotheses of theorem 5.3,*

$$\left\| (D + UV^*)^{-1} \right\| \le \left\| D^{-1} \right\| + \left\| D^{-1} U \right\| \left\| S^{-1} \right\| \left\| V^* D^{-1} \right\|.$$

Proof. Apply the triangle inequality and submultiplicativity to the Woodbury formula. □

5.2 Optimal factor recompression

Standard Fact 5.5 (Eckart–Young–Mirsky theorem)**.** If M has singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots,$$

pad this finite sequence with zeros. For each integer $k \geq 0$, its truncated singular value decomposition M_k is a best rank-at-most-k approximation. It satisfies

$$\|M - M_k\|_2 = \sigma_{k+1}$$

and

$$\|M - M_k\|_{\mathrm{F}} = \left(\sum_{j>k} \sigma_j^2 \right)^{1/2}.$$

This is classical [11, 22]. The next theorem shows that an n dimensional singular value decomposition is unnecessary when a correction is already factored.

Theorem 5.6 (Small core recompression)**.** *Let*

$$L \in \mathbb{C}^{m \times r}, \qquad R \in \mathbb{C}^{n \times r}.$$

Take compact QR factorizations

$$L = Q_L R_L, \qquad R = Q_R R_R,$$

where the columns of Q_L and Q_R are orthonormal. Form the small core

$$G = R_L R_R^*$$

and let

$$G = W \Sigma Z^*$$

be an ordered thin singular value decomposition, with nonnegative nonincreasing singular values. For an integer $k \geq 0$, retain $\min\{k, d\}$ columns, where d is the number of supplied thin singular directions. Tail singular values beyond d are zero. Define

$$L_k = Q_L W_k \Sigma_k^{1/2}, \qquad R_k = Q_R Z_k \Sigma_k^{1/2}.$$

Then

$$L_k R_k^*$$

is a best rank at most k approximation to LR^*. Its spectral and Frobenius errors are the singular value tails in standard fact 5.5. The core dimensions equal the column ranks of the two input factors. A noncompact reduced QR may retain extra zero singular directions without changing the conclusion.

Proof. The exact factor product is

$$LR^* = Q_L G Q_R^*.$$

The two outer maps are isometries on their column spaces. They preserve the nonzero singular values and the spectral and Frobenius norms of the core. For every ambient rank-at-most-k competitor B, its compression $Q_L^* B Q_R$ has rank at most k, and compression cannot increase either norm. Thus the

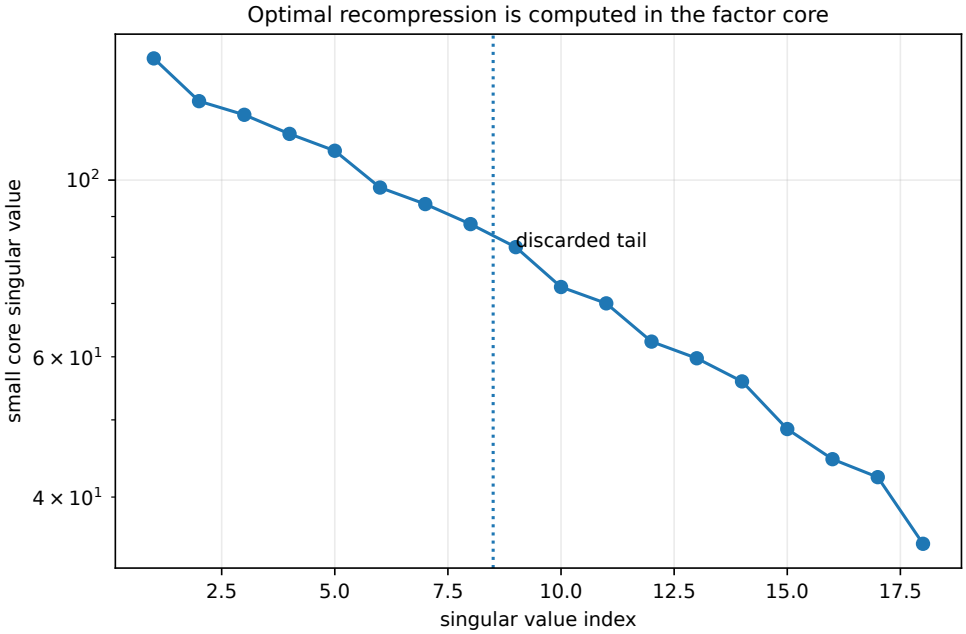


Figure 9. The approximation decision is made from the singular values of a small core whose dimension is the unrecompressed rank.

core best-approximation inequality also bounds the error of every ambient competitor from below. Truncating the singular value decomposition of G and applying standard fact 5.5 gives the result. The compact QR dimensions equal the ranks of the factor column spaces. □

Remark 5.7. The large matrices enter through QR factorizations of the factors. The singular value problem has dimension at most the unrecompressed rank. Updating and recompressing low rank factors is standard numerical linear algebra [13, 4].

Theorem 5.8 (Product chain perturbation). *Let $A_1, \dots, A_m$ be exact factors and let*

$$\widehat{A}_j = A_j + E_j.$$

Then

$$\boxed{\left\| \widehat{A}_m \cdots \widehat{A}_1 - A_m \cdots A_1 \right\| \le \sum_{j=1}^{m} \left(\prod_{i=j+1}^{m} \left\| \widehat{A}_i \right\| \right) \|E_j\| \left(\prod_{i=1}^{j-1} \|A_i\| \right).}$$

Empty products are one.

Proof. Use the telescoping identity

$$\widehat{A}_m \cdots \widehat{A}_1 - A_m \cdots A_1 = \sum_{j=1}^{m} \widehat{A}_m \cdots \widehat{A}_{j+1} E_j A_{j-1} \cdots A_1.$$

Take norms term by term. □

Theorem 5.9 (Certified repeated recompression). *Let*

$$P_0 = I, \qquad P_j = A_j P_{j-1},$$

be exact partial products. Suppose an implemented chain satisfies

$$\widehat{P}_0 = I, \qquad \widehat{P}_j = A_j \widehat{P}_{j-1} + E_j,$$

where E_j is the additive error introduced after the jth multiplication, for example by factor recompression. Then

$$\boxed{\left\|\widehat{P}_m - P_m\right\| \le \sum_{j=1}^{m} \left(\prod_{i=j+1}^{m} \|A_i\| \right) \|E_j\|.}$$

If $\|E_j\| \le \delta_j$, the same formula with δ_j gives a certified final error.

Proof. Set $D_j = \widehat{P}_j - P_j$. Then

$$D_j = A_j D_{j-1} + E_j, \qquad D_0 = 0.$$

Repeated substitution gives

$$D_m = \sum_{j=1}^{m} A_m A_{m-1} \cdots A_{j+1} E_j.$$

Taking norms yields the result. □

6 Sparse transfer graphs and schedules

6.1 Support under algebraic operations

Theorem 6.1 (Path support of powers). *For every positive integer m,*

$$P_\gamma A^m P_\alpha$$

is the sum over directed walks (vertices may repeat)

$$\alpha = \alpha_0 \longrightarrow \alpha_1 \longrightarrow \cdots \longrightarrow \alpha_m = \gamma$$

in G_A, with contribution

$$A_{\alpha_m \alpha_{m-1}} \cdots A_{\alpha_1 \alpha_0}.$$

If no such path exists, the block vanishes.

Proof. Insert the packet identity between every two consecutive factors of A. Every nonzero term determines one path of length m. □

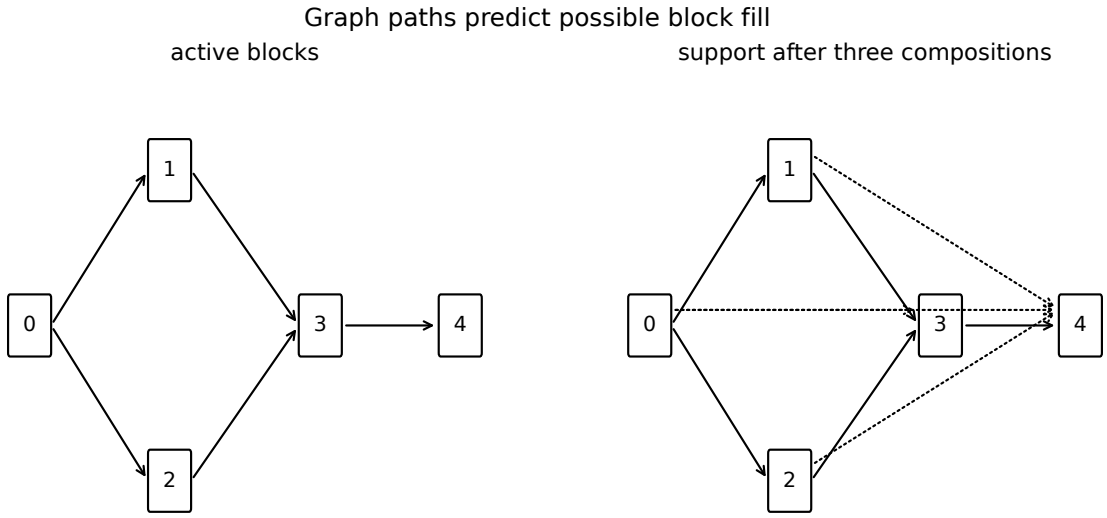


Figure 10. Composition creates possible support along directed paths. Cancellation can remove an allowed block, so the graph gives an inclusion rather than an equality.

Corollary 6.2 (Polynomial support)**.** *If*

$$p(z) = \sum_{j=0}^{d} c_j z^j,$$

then

$$\operatorname{supp}(p(A))$$

is contained in the union of the support relations of A^j for $0 \le j \le d$.

Proof. Apply theorem 6.1 to every power in the polynomial. □

Corollary 6.3 (Acyclic transfer nilpotence)**.** *Let K contain only strict transfer blocks. If its support graph is acyclic and its longest directed path has h edges, then*

$$K^{h+1} = 0.$$

Proof. There is no directed path of length $h+1$. Every block of K^{h+1} therefore vanishes by theorem 6.1. □

6.2 Block triangular solves

Definition 6.4 (Topological level)**.** Let G be a directed acyclic graph. The first level contains its source vertices. Remove them. The next level contains the new sources. Continue until every vertex has been assigned.

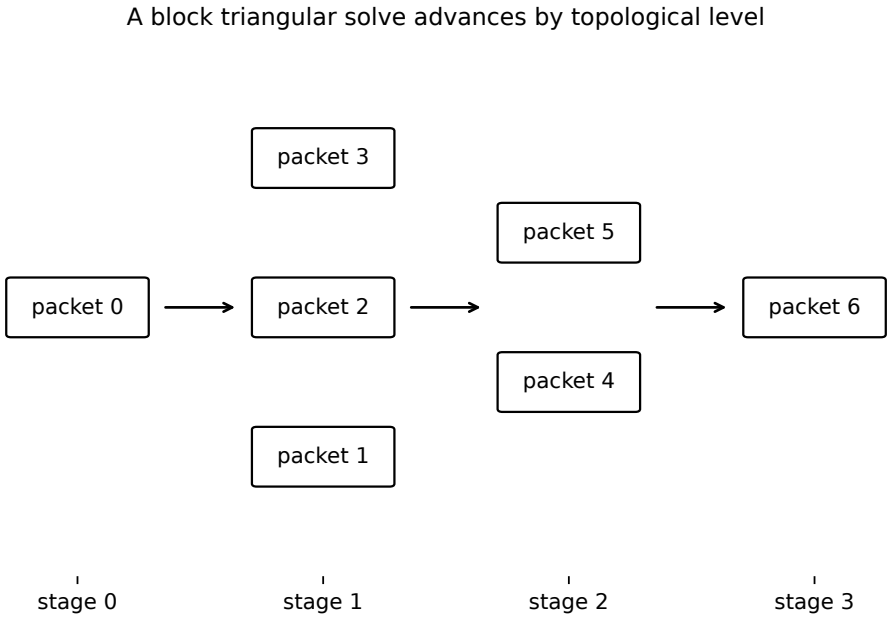


Figure 11. Packets in the same topological level can be processed concurrently in a triangular solve.

Theorem 6.5 (Level schedule for a triangular solve)**.** *Let $A = D + K$ be block lower triangular after a topological ordering. Assume every diagonal block D_α is invertible. The solve*

$$Ax = b$$

can be completed in one sequential stage per topological level. Blocks in the same level may be solved independently after all earlier levels are known.

Proof. The equation in packet β is

$$D_\beta x_\beta = b_\beta - \sum_{\alpha \to \beta} K_{\beta\alpha} x_\alpha.$$

Every predecessor of β lies in an earlier level. Once those values are available, x_β is obtained from the local solve with D_β. Vertices in the same level have no directed edge between them, so their solves are independent. □

Corollary 6.6 (Sequential depth)**.** *For a nonempty finite graph, the number of sequential stages in the level schedule equals one plus the length of the longest directed path. The empty graph has depth zero.*

Proof. The recursive level of a vertex is one plus the largest level of its predecessors. For a nonempty graph, the largest level therefore counts the vertices on a longest path. For an empty graph, no task is scheduled and the depth is zero. □

6.3 Fill under elimination

Theorem 6.7 (Inverse support lies in the transitive closure)**.** *Let A be an invertible packet operator. Reachability here includes a zero-length path from a packet to itself. If there is no directed path from packet α to packet β in G_A, then*

$$\boxed{P_\beta A^{-1} P_\alpha = 0.}$$

Proof. Let R be the set of packets reachable from α, including α, and set

$$X_R = \bigoplus_{\gamma \in R} X_\gamma.$$

No active edge leaves R, so $AX_R \subseteq X_R$. The restriction of A to the finite dimensional space X_R is injective. It is therefore onto X_R. Hence $A^{-1}X_R \subseteq X_R$. Since $\beta \notin R$, the stated block of A^{-1} vanishes. □

Proposition 6.8 (Schur support inclusion)**.** *Split the packets into retained indices R and hidden indices H. Suppose the hidden block A_{HH} is invertible. The retained Schur complement is*

$$S = A_{RR} - A_{RH} A_{HH}^{-1} A_{HR}.$$

A retained block (β, α) can be active only if it was already active in A_{RR}, or if there is a directed walk from α through hidden packets to β. The internal hidden path may have length zero, so entrance and exit may use the same hidden packet.

Proof. The correction block is

$$P_\beta A_{RH} A_{HH}^{-1} A_{HR} P_\alpha.$$

It can be nonzero only when $A_{HR}P_\alpha$ enters a hidden packet, the inverse hidden operator connects hidden packets, and $P_\beta A_{RH}$ exits to β. By theorem 6.7, the hidden inverse can connect only packets joined by a hidden directed path. Cancellation may still remove an allowed block. □

Sparse elimination and fill are treated systematically in [9]. The packet graph records the same structural issue at block scale.

7 Krylov methods and packet preconditioning

Definition 7.1 (Krylov space)**.** For a linear operator A, a vector v, and a positive integer m, define

$$\mathcal{K}_m(A, v) = \mathsf{span}\{v, Av, \dots, A^{m-1}v\}.$$

Proposition 7.2 (Invariant packet Krylov decomposition)**.** *Assume*

$$[A, P_\alpha] = 0$$

for every packet. If

$$v = \sum_\alpha v_\alpha, \qquad v_\alpha = P_\alpha v,$$

then

$$A^j v = \sum_\alpha A_\alpha^j v_\alpha$$

for every $j \geq 0$, where

$$A_\alpha = P_\alpha A P_\alpha.$$

Hence

$$\mathcal{K}_m(A, v) \subseteq \bigoplus_\alpha \mathcal{K}_m(A_\alpha, v_\alpha).$$

Proof. Commutation gives

$$A^j P_\alpha = P_\alpha A^j.$$

Apply this identity to each packet component of v. □

Theorem 7.3 (Approximate packet Krylov bound)**.** *Write*

$$A = D + K,$$

where D is packet diagonal. For every positive integer j,

$$\boxed{\left\|A^j - D^j\right\| \leq j M^{j-1} \left\|K\right\|,}$$

where

$$M = \max\{\left\|A\right\|, \left\|D\right\|\}.$$

Consequently,

$$\operatorname{dist}(A^j v, \bigoplus_\alpha \mathcal{K}_{j+1}(D_\alpha, v_\alpha)) \leq j M^{j-1} \left\|K\right\| \left\|v\right\|.$$

Proof. Use the telescoping identity

$$A^j - D^j = \sum_{k=0}^{j-1} A^{j-1-k} K D^k.$$

Each term has norm at most $M^{j-1} \left\|K\right\|$. The vector $D^j v$ lies in the direct sum of local Krylov spaces. □

Theorem 7.4 (Block Jacobi condition bound)**.** *Let*

$$A = D + K$$

act on a nonempty finite coordinate space and be Hermitian positive definite with positive definite block diagonal part D. Here $D^{-1/2}$ is the actual inverse of the positive square root of D. Suppose

$$E = D^{-1/2} K D^{-1/2}$$

is Hermitian and

$$\left\|E\right\|_2 \leq \eta < 1.$$

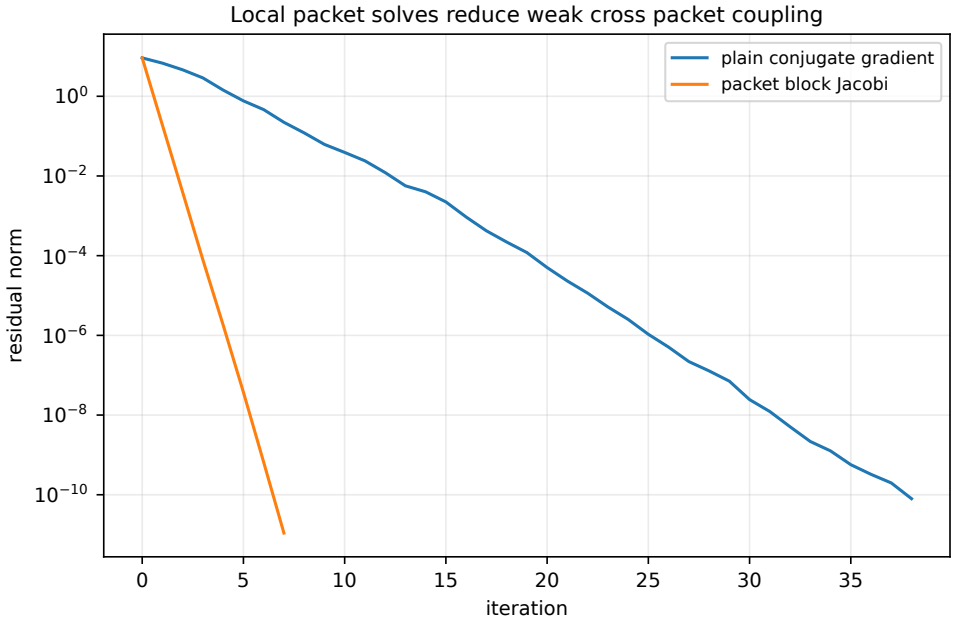


Figure 12. The block diagonal packet operator serves as a reusable local preconditioner in a weakly coupled positive system.

Then

$$\boxed{\operatorname{cond}_2(D^{-1/2}AD^{-1/2}) \le \frac{1+\eta}{1-\eta}.}$$

Proof. For every unit vector x,

$$|x^{*}Ex| \le \eta.$$

The Rayleigh quotient of $I + E$ lies in $[1-\eta, 1+\eta]$. The quotient of the extreme eigenvalue bounds gives the result. □

Standard Fact 7.5 (Preconditioned conjugate gradient convergence)**.** Let A and M be Hermitian positive definite on a nonempty finite coordinate space, and set

$$\kappa = \operatorname{cond}_2(M^{-1/2}AM^{-1/2}).$$

In exact arithmetic, the energy norm error of preconditioned conjugate gradient satisfies

$$\|x_m - x_*\|_A \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{m} \|x_0 - x_*\|_A .$$

This standard estimate is discussed in [25]. Combining it with theorem 7.4 yields an explicit iteration certificate when the normalized cross packet coupling is small.

Corollary 7.6 (Certified packet preconditioning)**.** *Under the hypotheses of theorem 7.4, the convergence factor in standard fact 7.5 is bounded by the value obtained from*

$$\kappa = \frac{1+\eta}{1-\eta}.$$

Proof. Insert the condition bound from theorem 7.4 into the monotone convergence factor in standard fact 7.5. □

8 Packetwise mixed precision

8.1 One operator

Theorem 8.1 (Packet perturbation certificate)**.** *Let*

$$A = \sum_{e \in E} A_e, \qquad \widehat{A} = \sum_{e \in E} (A_e + E_e),$$

where edge e starts at packet $\alpha(e)$, and assume

$$A_e = A_e P_{\alpha(e)}, \qquad E_e = E_e P_{\alpha(e)}.$$

Then

$$\boxed{\left\|(\widehat{A} - A)x\right\| \le \sum_{e \in E} \|E_e\| \left\|P_{\alpha(e)} x\right\|.}$$

If

$$\|E_e\| \le \varepsilon_e \|A_e\|,$$

then

$$\boxed{\left\|(\widehat{A} - A)x\right\| \le \sum_{e \in E} \varepsilon_e \|A_e\| \left\|P_{\alpha(e)} x\right\|.}$$

Proof. Exact packet evaluation gives

$$(\widehat{A} - A)x = \sum_{e} E_e P_{\alpha(e)} x.$$

Apply the triangle inequality and the operator norm bound to each term. □

Standard Fact 8.2 (Floating point matrix product model)**.** For real $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$, consider ordinary dot-product matrix multiplication in uniform precision with unit roundoff u, no overflow or underflow, and $pu < 1$. Its computed output $\widehat{C}$ satisfies the componentwise forward-error bound

$$|\widehat{C} - AB| \le \gamma_p |A||B|, \qquad \gamma_p = \frac{pu}{1-pu}.$$

Absolute values and the inequality are entrywise. Complex arithmetic, fused kernels, mixed accumulation, and tensor kernels require their own stated constants. This fact does not assert one common small

input perturbation for every column, and an input-dependent rounded action is not automatically a fixed linear operator.

Detailed forms and their hypotheses are given in [17, 18]. For a fixed computed matrix, the bound can certify its error matrix. A rounded application to varying inputs needs a separately justified uniform certificate before it can be used as one fixed operator error in theorem 8.1.

8.2 Several operators

Corollary 8.3 (Mixed precision product chain)**.** *If each fixed linear factor A_j is replaced by a fixed packetwise linear approximation $\widehat{A}_j$, write $E_j = \widehat{A}_j - A_j = \sum_e \Delta_{j,e} P_{\alpha(j,e)}$. A uniform certificate is*

$$\|E_j\| \le \delta_j := \sum_e \|\Delta_{j,e}\| \, \|P_{\alpha(j,e)}\|.$$

For orthogonal packet projectors, $\delta_j \le \sum_e \|\Delta_{j,e}\|$. The final operator error is bounded by theorem 5.8 with these uniform δ_j. The corresponding vector error is this operator bound multiplied by the input norm.

Proof. Take the supremum over unit inputs in theorem 8.1, using $\|P_\alpha x\| \le \|P_\alpha\|\|x\|$. Substitute the resulting uniform operator error bounds into theorem 5.8. The action estimate follows from submultiplicativity. □

8.3 Continuous allocation

Theorem 8.4 (Continuous precision allocation)**.** *Let E be a finite nonempty edge set, let $s_e > 0$ be sensitivities and $c_e > 0$ be costs, and let $\tau > 0$. Consider*

$$\min_{\varepsilon_e > 0} \sum_e c_e \log \frac{1}{\varepsilon_e}$$

subject to

$$\sum_e s_e \varepsilon_e = \tau.$$

The unique minimizer is

$$\boxed{\varepsilon_e = \frac{\tau c_e}{s_e \sum_f c_f}.}$$

Proof. The Lagrangian derivative gives

$$-\frac{c_e}{\varepsilon_e} + \lambda s_e = 0.$$

Thus

$$\varepsilon_e = \frac{c_e}{\lambda s_e}.$$

The constraint gives

$$\lambda = \frac{\sum_f c_f}{\tau}.$$

The objective is strictly convex on the positive orthant, so the stationary point is the unique minimizer. □

Theorem 8.5 (Clipped precision allocation)**.** *Retain the finite nonempty edge set, strictly positive sensitivities and costs, and add strictly positive caps $m_e > 0$ with upper bounds*

$$0 < \varepsilon_e \le m_e.$$

Assume

$$0 < \tau \le \sum_e s_e m_e.$$

There is a unique $\lambda > 0$, except at the maximal tolerance endpoint, such that

$$\boxed{\varepsilon_e = \min\left\{m_e, \frac{c_e}{\lambda s_e}\right\}}$$

and

$$\sum_e s_e \varepsilon_e = \tau.$$

This vector minimizes the same objective over the clipped feasible set.

Proof. The Karush–Kuhn–Tucker conditions give the unconstrained form on every nonactive coordinate and equality with m_e on every active upper bound. The weighted sum

$$\phi(\lambda) = \sum_e s_e \min\left\{m_e, \frac{c_e}{\lambda s_e}\right\}$$

is continuous and nonincreasing. It decreases from the maximal feasible value to zero. For a strictly submaximal budget, once any coordinate is unsaturated, ϕ is strictly decreasing, which gives a unique multiplier. At maximal budget the only feasible vector is $(m_e)_e$ and every $0 < \lambda \le \min_e c_e/(s_e m_e)$ produces it. Thus the minimizing vector is unique at this endpoint while its multiplier need not be. Strict convexity gives uniqueness of the minimizer in every case. □

8.4 Discrete data types

Definition 8.6 (Discrete precision option)**.** For edge e, a discrete precision option k has a measured or modeled cost $c_{e,k}$ and a nonnegative integer error weight $w_{e,k}$. Exactly one option must be selected on every edge.

Theorem 8.7 (Exact discrete precision program)**.** *Let the total integer error budget be B. Define*

$$F_j(b)$$

as the minimum cost for assigning precisions to the first j edges with total weight exactly b. Then

$$F_0(0) = 0, \qquad F_0(b) = +\infty \quad \text{for } b \neq 0.$$

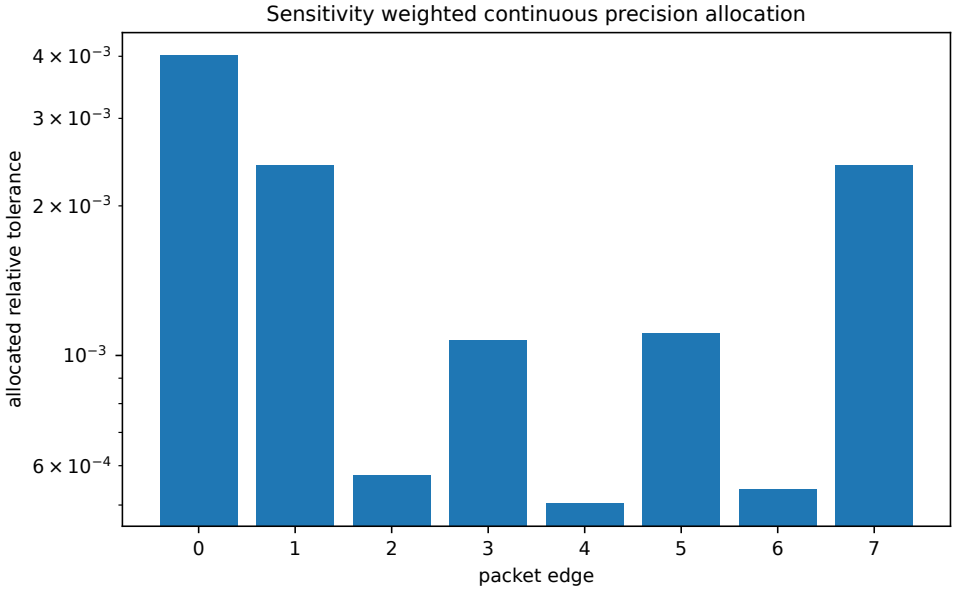


Figure 13. More sensitive packet edges receive tighter continuous tolerances.

Set $F_{j-1}(u) = +\infty$ whenever $u < 0$ or the state is unreachable. Then

$$\boxed{F_j(b) = \min_k \left\{ F_{j-1}(b - w_{j,k}) + c_{j,k} \right\}.}$$

The minimum of $F_m(b)$ over $0 \le b \le B$ is the exact minimum cost for the quantized model.

Proof. Every feasible assignment for the first j edges ends with one option k on edge j. Removing it leaves an optimal or suboptimal assignment for the first $j - 1$ edges with residual weight $b - w_{j,k}$. Replacing a suboptimal prefix by an optimal one can only lower cost. The recurrence therefore has optimal substructure. Induction on j proves the result. □

Remark 8.8. The discrete program is exact only for the supplied options and quantized error weights. Real kernels also depend on accumulation type, scaling, overflow, underflow, tensor core availability, and shape restrictions. Mixed precision iterative refinement has its own conditioning requirements [5].

9 Discovery and exact packet compilation

9.1 Exact spectral discovery

Standard Fact 9.1 (Simultaneous spectral decomposition)**.** A finite family of pairwise commuting normal matrices admits a common orthonormal basis of joint eigenvectors. The joint eigenspaces associated with distinct eigenvalue tuples give the coarsest orthogonal packet system on which every matrix in the family acts as a scalar.

Proposition 9.2 (Exact block discovery certificate)**.** *For an orthogonal packet system (P_α), define the block diagonal compression*

$$\Pi_{\mathcal{P}}(A) = \sum_{\alpha} P_\alpha A P_\alpha.$$

Then

$$\boxed{\|A - \Pi_{\mathcal{P}}(A)\|_{\mathrm{F}}^2 = \sum_{\alpha \neq \beta} \|P_\beta A P_\alpha\|_{\mathrm{F}}^2 .}$$

In operator norm,

$$\boxed{\|A - \Pi_{\mathcal{P}}(A)\|_2 \leq \sum_{\alpha \neq \beta} \|P_\beta A P_\alpha\|_2 .}$$

Thus any proposed packet discovery has an exact finite residual certificate after its projectors are known.

Proof. Distinct matrix blocks are orthogonal in the Hilbert–Schmidt inner product. Their squared Frobenius norms therefore add. The operator norm estimate follows from the triangle inequality. □

This supplies one exact discovery regime and one exact a posteriori certificate. Perturbation of spectral packets requires a gap. The Davis–Kahan theorem gives quantitative control for Hermitian spectral subspaces under such a gap [8]. Randomized low rank discovery supplies another controlled regime [16, 21]. Neither result gives globally optimal packet discovery for arbitrary operators.

9.2 Candidate packet trees

Definition 9.3 (Candidate representation tree)**.** A candidate representation tree is a finite rooted tree. Every node v has

$$\kappa_v \geq 0, \qquad \epsilon_v \geq 0,$$

representing the cost and error of keeping v as one leaf. Every internal node also has a split overhead $\sigma_v \geq 0$. An admissible pruning contains exactly one selected node on every root to terminal path.

The cost and error of a pruning $\mathcal{L}$ are

$$\mathrm{cost}(\mathcal{L}) = \sum_{v \in \mathcal{L}} \kappa_v + \sum_{u \text{ split}} \sigma_u$$

and

$$\mathrm{err}(\mathcal{L}) = \sum_{v \in \mathcal{L}} \epsilon_v.$$

The model can include arithmetic, storage, launch, transfer, and precision costs inside κ_v. It remains exact only when the declared totals are additive.

Theorem 9.4 (Exact Lagrangian packet compiler). *Fix $\lambda \geq 0$ and define*

$$J_\lambda(\mathcal{L}) = \mathrm{cost}(\mathcal{L}) + \lambda\, \mathrm{err}(\mathcal{L}).$$

For every node v, set

$$K_\lambda(v) = \kappa_v + \lambda \epsilon_v$$

and

$$S_\lambda(v) = \sigma_v + \sum_{w \in \mathrm{child}(v)} F_\lambda(w).$$

For a terminal node, set

$$F_\lambda(v) = K_\lambda(v).$$

For an internal node, the recurrence

$$\boxed{F_\lambda(v) = \min\{K_\lambda(v), S_\lambda(v)\}}$$

computes the exact minimum Lagrangian objective over every admissible pruning of the subtree rooted at v.

Proof. A pruning of the subtree rooted at v has two possible forms. It keeps v, in which case its objective is $K_\lambda(v)$. Otherwise it splits v and independently chooses an admissible pruning in every child subtree. By induction, the optimal child values are $F_\lambda(w)$. The split objective is therefore $S_\lambda(v)$. Taking the smaller value is exact. □

This recurrence is a computational version of best basis selection and Bellman optimality [3, 6]. The node costs here may include operator application, transfer, launch, precision, and output conversion rather than only a signal entropy.

Corollary 9.5 (Shared compilation for an operator family). *Let operators $A^{(1)}, \dots, A^{(m)}$ use the same candidate tree. Give operator j a nonnegative reuse weight r_j. Replace the local keep cost and error by*

$$\kappa_v = \kappa_v^{\mathrm{setup}} + \sum_{j=1}^{m} r_j \kappa_v^{(j)}$$

and

$$\epsilon_v = \sum_{j=1}^{m} r_j \epsilon_v^{(j)}.$$

Then theorem 9.4 gives the exact optimal shared pruning for the additive weighted objective.

Proof. The weighted family objective is still the sum of local leaf and split terms. Apply theorem 9.4 to the aggregated quantities. □

Proposition 9.6 (Monotone Lagrangian frontier). *Let $0 \leq \lambda_1 < \lambda_2$. Choose optimal prunings $\mathcal{L}_1$ and $\mathcal{L}_2$ for the two Lagrangian parameters. Then*

$$\boxed{\mathrm{err}(\mathcal{L}_2) \leq \mathrm{err}(\mathcal{L}_1)}$$

and

$$\boxed{\mathrm{cost}(\mathcal{L}_2) \geq \mathrm{cost}(\mathcal{L}_1).}$$

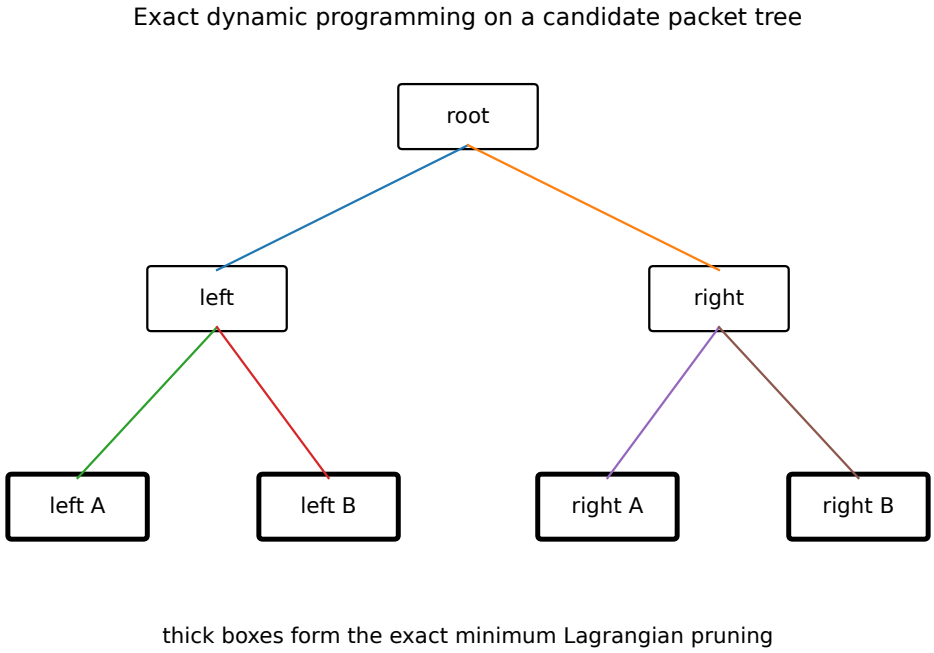


Figure 14. The thick leaves are selected by the exact Lagrangian recurrence.

Proof. Optimality gives

$$\operatorname{cost}(\mathcal{L}_1) + \lambda_1 \operatorname{err}(\mathcal{L}_1) \le \operatorname{cost}(\mathcal{L}_2) + \lambda_1 \operatorname{err}(\mathcal{L}_2)$$

and

$$\operatorname{cost}(\mathcal{L}_2) + \lambda_2 \operatorname{err}(\mathcal{L}_2) \le \operatorname{cost}(\mathcal{L}_1) + \lambda_2 \operatorname{err}(\mathcal{L}_1).$$

Adding the inequalities yields

$$(\lambda_2 - \lambda_1)(\operatorname{err}(\mathcal{L}_2) - \operatorname{err}(\mathcal{L}_1)) \le 0.$$

The error ordering follows. Substitution into the first inequality gives the cost ordering. □

Corollary 9.7 (Supported Pareto points)**.** *For $\lambda > 0$, every Lagrangian optimum is Pareto optimal for the pair*

$$(\operatorname{cost}, \operatorname{err}).$$

The quantized budget compiler may recover Pareto points that are not optimal for any Lagrangian parameter.

Proof. If another pruning had no larger cost and no larger error, with one strict inequality, its Lagrangian objective would be smaller. This contradicts optimality. A finite nonconvex frontier can contain points that are not supported by a line of slope $-\lambda$, so the budget formulation is genuinely additional. □

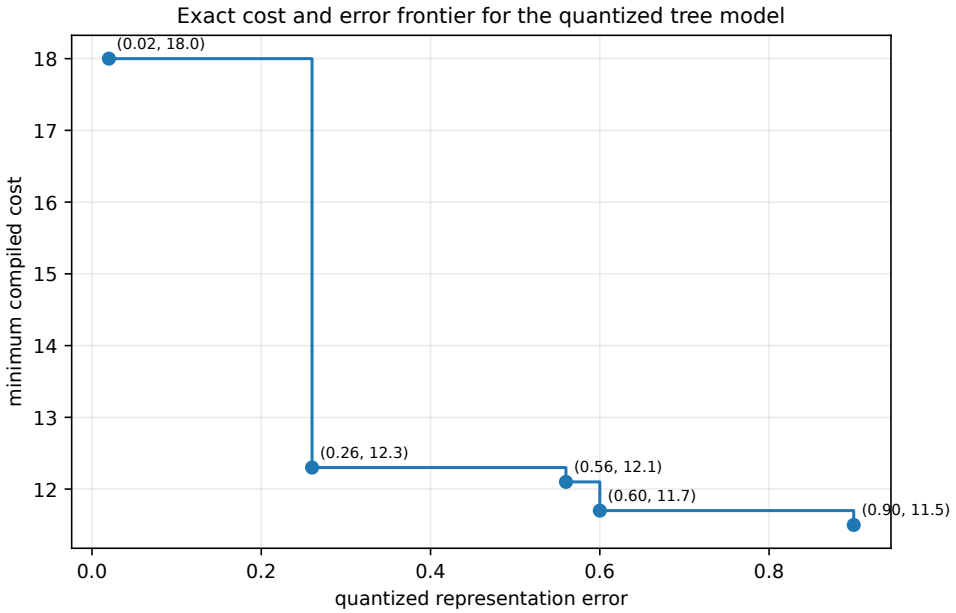


Figure 15. The quantized dynamic program produces the exact cost and error frontier for its finite model.

9.3 An exact error budget compiler

Theorem 9.8 (Quantized budget dynamic program)**.** *Assume every keep error is a nonnegative integer w_v. Let $B \in \mathbb{N}$. For each node v, define $F_v(b)$ as the minimum cost of a pruning of the subtree at v with total error exactly b. The keep option contributes*

$$F_v(w_v) \leq \kappa_v.$$

The split option is obtained by min plus convolution of the child tables, followed by addition of σ_v. The smallest value among

$$F_{\text{root}}(b), \qquad 0 \leq b \leq B,$$

is the exact minimum cost among prunings whose quantized error is at most B.

Proof. The keep case is immediate. In the split case, an admissible pruning is the union of one admissible pruning from each child. Its error is the sum of the child errors and its cost is the sum of their costs plus σ_v. Repeated min plus convolution enumerates every possible distribution of the integer budget across the child subtrees. Induction on the tree proves exactness. □

Corollary 9.9 (Pseudo polynomial complexity)**.** *If the maximum number of children is bounded and the budget is B, the quantized compiler has pseudo polynomial complexity in the number of tree nodes and B. A direct pairwise convolution implementation has quadratic dependence on B at each merge.*

Proof. Every table has at most $B+1$ entries. Merging two tables by direct convolution examines at most $(B+1)^2$ pairs. A bounded number of merges occurs at every node. □

9.4 Decision stability and local updates

Definition 9.10 (Decision gap)**.** At an internal node v, let

$$g_v = |K_\lambda(v) - S_\lambda(v)| .$$

This is the gap between the keep and split scores after the child subproblems have been solved.

Theorem 9.11 (Compiler stability)**.** *Suppose the solved keep score and split score at every internal node change in absolute value by less than $g_v/2$. These are scores after child subproblems have been solved, rather than unconstrained changes in raw local costs. Then every strict keep or split decision in the original optimal pruning remains unchanged.*

Proof. If the original keep score is lower by g_v, then after perturbation it can increase by less than $g_v/2$, while the split score can decrease by less than $g_v/2$. The order remains strict. The split case is the same. Apply the argument from the leaves upward. □

Theorem 9.12 (Local recompilation)**.** *Suppose local costs change only inside the subtree rooted at u, while ancestor and sibling local models are unchanged. Use correct cached sibling values from the old solve. Then every dynamic programming value outside that subtree and outside the ancestor chain from u to the root remains unchanged. Recomputing the subtree at u and that ancestor chain gives the exact new global optimum.*

Proof. A node value depends only on its own local data and the values of its children. Nodes outside the changed subtree and outside the ancestor chain have the same local data and the same child values. Their values remain unchanged. Every affected dependency lies inside the changed subtree or on the unique path from u to the root. □

Remark 9.13 (Boundary of the compiler theorem). The exact compiler requires a finite candidate tree and additive local models. Arbitrary overlapping decompositions, nonlocal cache interactions, global rank coupling, and unrestricted structure discovery can destroy optimal substructure. No universal approximation factor is proved for those settings.

10 Device scheduling and amortization

10.1 Shape classes

Definition 10.1 (Shape class)**.** Two active packet blocks lie in the same shape class when their input dimension, output dimension, right hand side count, scalar type, accumulation type, and kernel contract agree.

Proposition 10.2 (Launch reduction by shape)**.** *Suppose B independent active blocks fall into g shape classes, and suppose the backend accepts one batched array or pointer list for each class. Grouped execution then requires at most g launches instead of B, without changing the arithmetic or the output.*

Proof. Place all blocks in one class along a batch dimension and invoke one batched kernel for that class. Every block receives the same local computation. One launch per class is sufficient. □

CUDA provides graph capture and replay for repeated dependency graphs [23]. These mechanisms reduce repeated launch work, but they do not change the operator or its arithmetic count.

10.2 Critical path

Theorem 10.3 (Acyclic schedule depth)**.** *Let a packet task graph be acyclic. Assume every task takes one normalized stage, every edge is a dependency, and an unlimited number of independent tasks may run concurrently. For a nonempty finite graph, the minimum possible number of sequential stages equals the number of vertices on a longest directed path. The empty task graph has depth zero.*

Proof. Every path gives a lower bound because its tasks must execute in order. The topological level schedule in theorem 6.5 achieves exactly this depth. With no tasks, the empty schedule has depth zero. □

10.3 One time costs

Theorem 10.4 (Amortization threshold)**.** *Let $s \geq 0$ be a one time setup cost. Let $t_b, t_c \geq 0$ be the baseline and candidate times per use and let $R \in \mathbb{N}$ be a reuse count. If $t_b > t_c$, the candidate is faster after R uses exactly when*

$$R > \frac{s}{t_b - t_c}.$$

The smallest integer reuse count is

$$\boxed{R_{\min} = \left\lfloor \frac{s}{t_b - t_c} \right\rfloor + 1.}$$

If $t_b \leq t_c$, no positive reuse count amortizes a positive setup cost.

Proof. Compare

$$s + Rt_c$$

with

$$Rt_b.$$

Rearranging gives the stated inequality. □

Corollary 10.5 (Graph replay threshold)**.** *If graph capture costs $c \geq 0$, ordinary execution costs $t_o \geq 0$, and replay costs $0 \leq t_r < t_o$, then capture is profitable when*

$$R > \frac{c}{t_o - t_r}.$$

Proof. Apply theorem 10.4 with setup cost c. □

Corollary 10.6 (Materialization threshold)**.** *Suppose a compressed representation costs $t_c \geq 0$ per downstream use, an explicit representation costs $t_e \geq 0$ per use, and one time materialization costs $m \geq 0$, with reuse count $R \in \mathbb{N}$. Materialization is profitable exactly when*

$$R > \frac{m}{t_c - t_e}$$

and $t_c > t_e$.

Proof. Compare repeated compressed use with materialization followed by repeated explicit use. □

11 Benchmark contracts and measurement protocol

Performance data are meaningful only after the mathematical and timing contracts are fixed. The protocol follows standard concerns from numerical linear algebra, floating point analysis, and machine performance modeling [17, 14, 27].

Definition 11.1 (Case signature)**.** A case signature records the dimensions, packet layout, rank, right hand side count, random seed, source values, source type, reference computation, and output contract.

Definition 11.2 (Fair match)**.** Two measured rows are a fair match when they share one case signature, return the same declared output, use the same correctness reference, and have identical timing scopes. If one row includes an extra conversion or setup cost, it is a different scope and must be reported separately or added to both rows before comparison.

Proposition 11.3 (Fair matching is an equivalence relation)**.** *On rows with complete metadata, equality of the case signature, output contract, correctness reference, and timing scope is an equivalence relation.*

Proof. Equality of each metadata field is reflexive, symmetric, and transitive. Their conjunction has the same properties. □

Definition 11.4 (Representation gain)**.** Representation gain compares dense and structured work or storage for the same operator, independently of the processor.

Definition 11.5 (Device gain)**.** Device gain compares processors for the same representation, output, and timing scope.

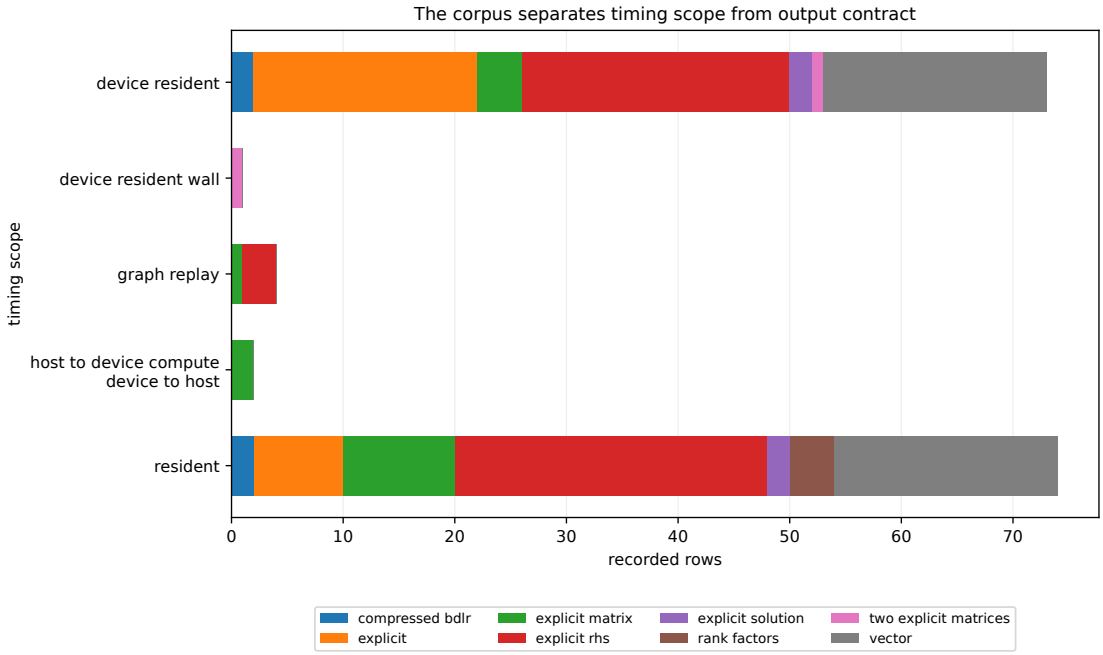


Figure 16. The corpus records timing scope and output contract as separate fields.

Definition 11.6 (Kernel gain)**.** Kernel gain compares two implementations on the same device for the same representation, output, and timing scope.

Definition 11.7 (Amortization gain)**.** Amortization gain records the effect of one time discovery, compilation, transfer, capture, or materialization at a declared reuse count.

A measured total speedup can mix these four effects. A multiplicative decomposition is only exact when matched rows isolate the factors. Otherwise the decomposition is interpretive.

Remark 11.8 (Logical work rate). A structured method may report a dense equivalent logical work rate greater than the physical dense throughput because it does not execute the dense operations. This statistic describes avoided work. It is not a claim that the device performed those operations.

12 Recorded CPU and GPU corpus

The empirical corpus contains 154 measured rows from sixteen suites. It has 74 CPU rows and 80 GPU rows. The GPU was an NVIDIA GeForce RTX 3070 Ti Laptop GPU with compute capability 8.6 and 8 GB nominal memory. The recorded software used PyTorch 2.11.0 with a CUDA 12.8 build. These facts identify the environment. They do not predict another machine.

The 154 archived rows contain recorded case identifiers, seeds, types, output contracts, and timing scopes. Source-value hashes and explicit correctness-reference fields are absent. Setup time, discovery time, and reuse count are nonnull in only two rows each. Correctness errors are present in 149 rows. Equality of available metadata cannot establish the full fair-match relation in definition 11.2. In particular, CPU-resident and GPU-device-resident scopes differ, and graph replay excludes capture while a device-resident baseline has a different raw scope. The two-stream example also compares wall time

Suite	Rows	Suite	Rows
Dense multiplication	28	BDLR application	15
Block diagonal	12	BDLR product	12
Circulant	12	Toeplitz	12
Kronecker	12	Block banded	8
Sparse plus low rank	8	HODLR	8
Strassen	6	Transfer pipeline	5
Walsh Hadamard	4	Hidden blocks	4
Randomized low rank	4	Iterative refinement	4

Table 1. Coverage of the sanitized measured corpus.

Method	Median time in ms	Logical GFLOP per second	Relative error
FP64	541.2803	253.91	2.18×10^{-15}
FP32	18.1309	7580.35	5.76×10^{-7}
TF32	8.6999	15797.75	2.94×10^{-4}
FP16	4.0264	34134.72	3.60×10^{-4}
BF16	4.1482	33132.00	2.88×10^{-3}

Table 2. Recorded dense accuracy and throughput at one dimension.

with a serial device-time scope. The values below are recorded observations with these limitations.

12.1 Dense precision frontier

At dimension 4096, the device resident dense rows gave the values in table 2.

The frontier is not a universal ordering of scalar types. It records one generated family. Packet precision assignment should use local sensitivities and supported kernels rather than one global type.

12.2 Regular packet operators

For an 8192 by 8192 equal block operator with block width 64 and 32 right hand sides, the batched packet calculation took

$$0.046032 \text{ ms}.$$

It was

$$14.8919$$

times faster than the recorded device resident dense GPU row with the same recorded case, output, and timing scope and

$$235.7404$$

times faster than the recorded CPU dense row (resident CPU scope versus device-resident GPU scope). The relative error was

$$8.17 \times 10^{-8}.$$

For BDLR application with the same dimension and right hand side count, block width 64, and rank 8, CUDA Graph replay took

$$0.079872 \text{ ms}.$$

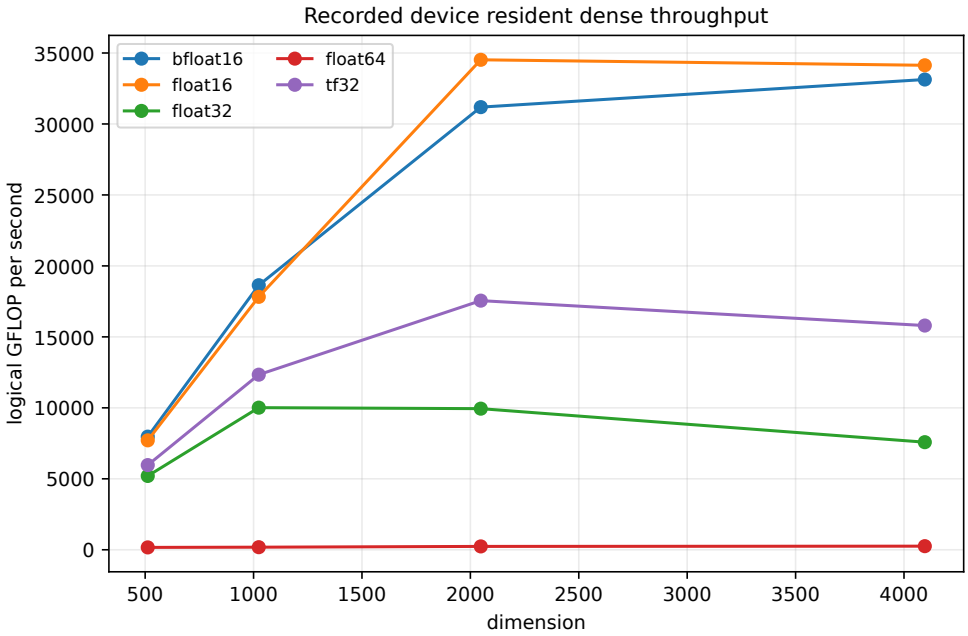


Figure 17. Recorded device resident dense throughput across dimensions and scalar types.

It was

$$10.9355$$

times faster than the recorded dense GPU row (graph-replay scope versus device-resident scope) and

$$153.6527$$

times faster than the recorded CPU dense row (resident CPU scope versus device-resident GPU scope). Capture time was excluded from the replay scope and must be amortized separately.

12.3 Compressed products

A compressed BDLR product returns block and low rank factors. Materialization returns all matrix entries. The recorded product rows illustrate the cost of changing the output contract. Compressed and materialized outputs are separate computations. The GPU materialized BDLR rows at dimensions 512 and 1024 are sums of separately measured factor and expansion medians, as the raw notes state. They are not timing medians for complete workflows.

The algebra in theorem 5.1 explains why the compressed product can remain small. The materialization cost in proposition 3.10 explains why explicit output can erase the advantage.

12.4 Discovery and transfer

For hidden permuted blocks at dimension 512, structure discovery took 2.1102 ms, dense application took 0.0969 ms, and packet application took 0.0151 ms. The exact threshold in theorem 10.4 is 26 uses. At dimension 1024, the corresponding values were 5.2726, 0.1896, and 0.0241 ms, giving 32 uses.

For one 2048 dense FP32 product, device resident graph replay took 1.9993 ms. The end to end pageable path took 7.3036 ms. The pinned path took 6.3084 ms. Two serial device resident products took 3.6557 ms, while the tested two stream path took 4.2368 ms.

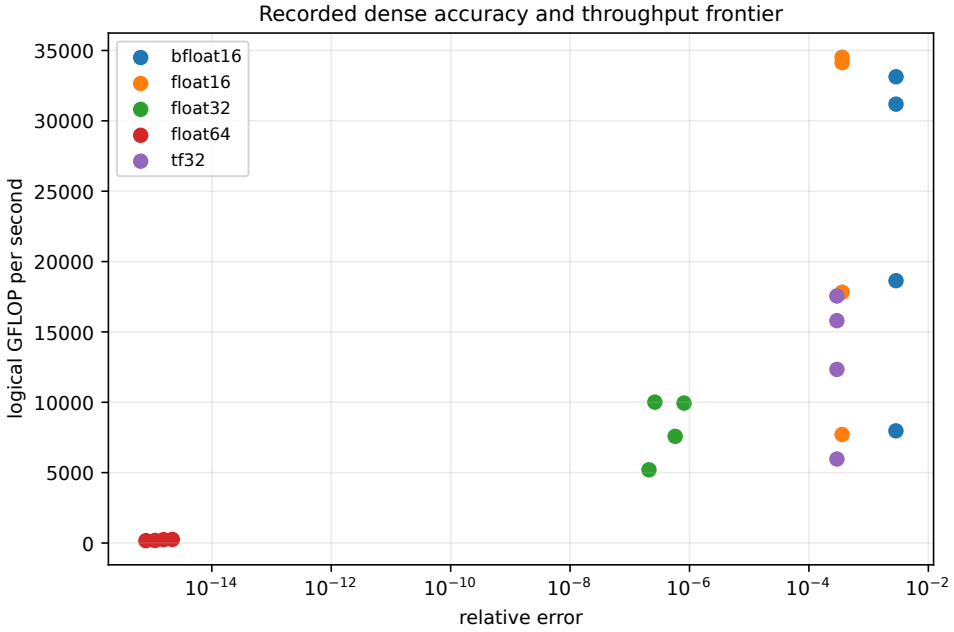


Figure 18. The speed and error tradeoff depends on both precision and problem size.

12.5 Negative controls

The corpus retains methods that lost in the recorded baseline comparisons. These are descriptive ratios. Scope and complete-metadata qualifications apply.

- At dimension 512, recursive Strassen multiplication with leaf size 128 took 7.0950 ms, while the recorded OpenBLAS dense multiplication took 1.9597 ms.
- At dimension 2048, the tested recursive HODLR GPU path took 5.6079 ms, while dense GPU multiplication took 0.0860 ms.
- The tested FP32 iterative refinement solve took 22.0848 ms, while the direct FP64 GPU solve took 12.6286 ms with a comparable final error.
- The two stream dense path was slower than the serial two product path.
- The GPU Walsh–Hadamard transform was slower than the CPU implementation at length 65536, then became faster at length 262144.

Fast multiplication can be asymptotically valuable and still lose at measured sizes because of recursion, memory traffic, and kernel granularity. Hierarchical matrix methods have similar crossover issues [1]. Transform algorithms and tensor factorizations form additional structure classes [7, 24].

13 Implementation and verification

The source corpus contains a small reference implementation of the exact operators and optimization algorithms used in the paper. The mathematical interface is deliberately narrow. This Python implementation is a numerical prototype, not extracted Lean code. Its stored-key support can retain zero-valued blocks, whereas the exact support graph uses nonzero blocks. Its reduced numerical QR is not

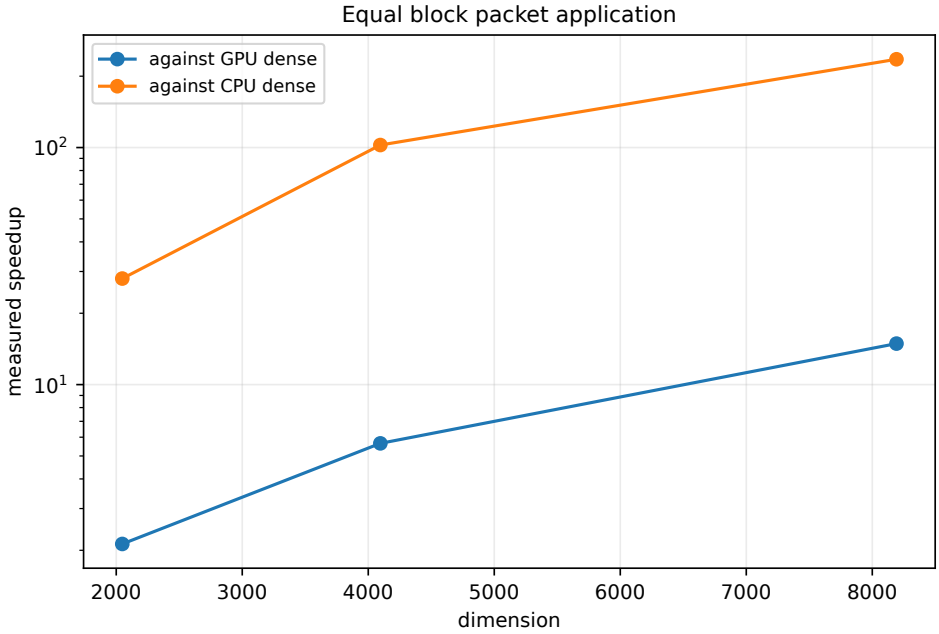


Figure 19. Regular equal shapes expose both representation reduction and batched execution.

the exact rank-compact range construction proved in Appendix F. Prototype tests and theorem proofs therefore have different targets.

1. A packet layout records dimensions and coordinate slices.
2. A packet operator stores active blocks and evaluates them without materializing inactive blocks.
3. BDLR factors support exact application, product, solve, and small core recompression.
4. Graph utilities compute support products, path closure, Schur fill inclusion, and topological levels.
5. Tree dynamic programs solve the Lagrangian and quantized budget models.
6. A multiple choice dynamic program assigns discrete precision options.
7. Benchmark utilities audit case identity, output contracts, timing scopes, and retained negative controls.

The accompanying verification record reports a deterministic verifier with 37392 checks in nine groups. It covers packet evaluation, adjoints, composition, structured primitives, recompression optimality, product chain error, graph support, Schur fill, tree compilation, precision assignment, and the benchmark corpus. It also reports eleven passing unit tests and twenty-three passing original-source tests for BDLR operations, Krylov methods, mixed precision, measured data schema, and figure generation.

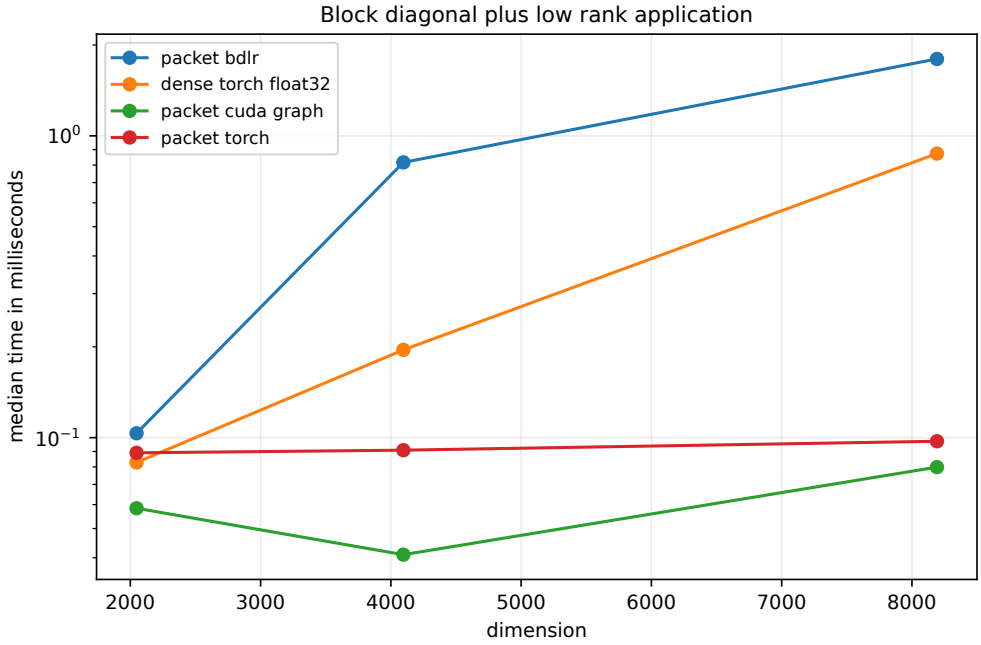


Figure 20. Representation and launch strategy both affect BDLR application time.

Layer	Verification target
Packet semantics	materialization, application, adjoint, composition
Low rank algebra	exact product, Woodbury solve, optimal core recompression
Graph algorithms	path support, topological levels, Schur fill inclusion
Optimization	Lagrangian tree compiler, budget compiler, precision assignment
Numerical bounds	product chain error, block perturbation, clipped allocation
Empirical corpus	row count, suite count, device split, status, error ceiling
Document	proofs, labels, citations, figures, font embedding, render audit

Table 3. Verification layers for the supplied implementation.

14 Hardware implications and exact boundaries

The finite results suggest a device interface with grouped block kernels, low rank contractions, packet metadata, a transfer graph scheduler, and per edge precision tags. Such an interface may reduce materialization and repeated launch work when the operator family remains closed under the chosen representation.

The exact results stop at a finite candidate tree and a declared device model. Three broader problems remain open.

1. A universal discovery algorithm would need to find a profitable packet structure from an arbitrary operator and data distribution. The simultaneous spectral theorem and randomized low rank approximation cover special cases only.

2. A universal compiler would need a controlled approximation factor when costs are nonlocal and candidate decompositions overlap. The dynamic programs in theorems 9.4 and 9.8 do not cover that setting.

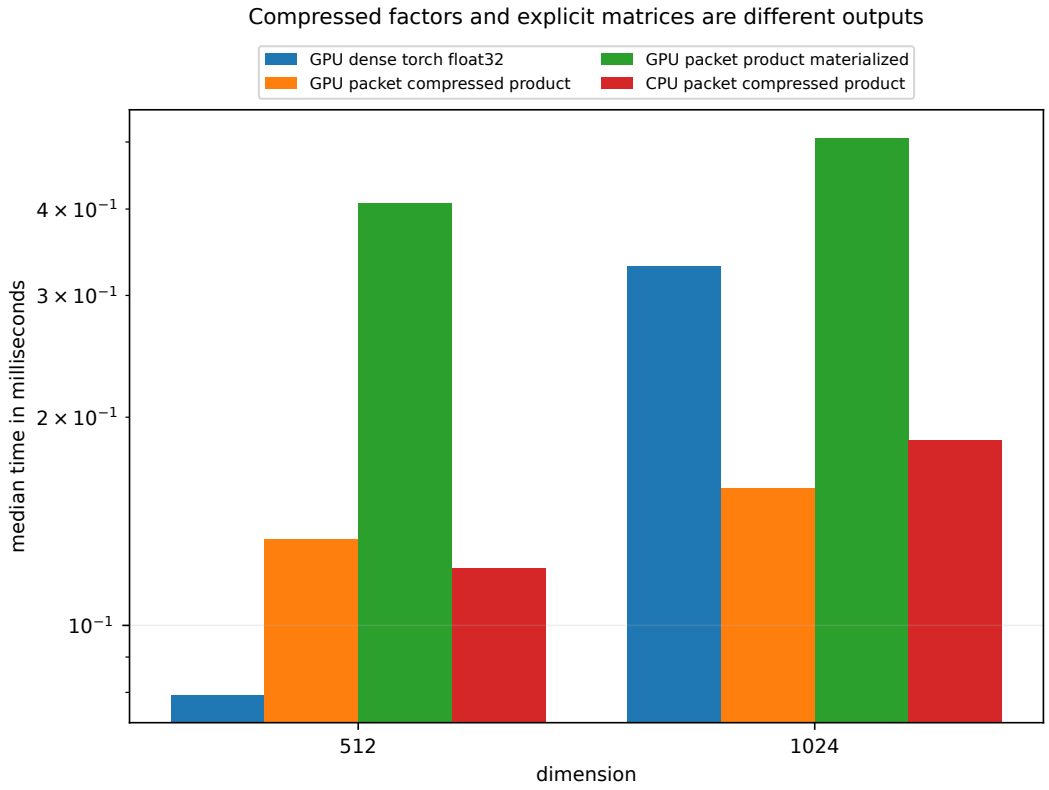


Figure 21. Compressed and explicit products answer different computational questions.

3. A packet native memory hierarchy would need a communication theorem for a family of operators and a specified architecture. The one pass byte model in proposition 3.6 is not such a lower bound.

Communication lower bounds require explicit machine models [19, 2]. CUDA Graph behavior and kernel constraints depend on the installed CUDA platform [23]. The measured results in this paper do not prove an optimal accelerator.

The article also makes no claim about the exponent of generic dense matrix multiplication. Packet algorithms answer a different question. They exploit a restricted operator class and may return a compressed output.

15 Conclusion

A packet operator representation records which block maps exist, how they are stored, and what output they denote. Exact block evaluation and composition turn the support graph into algebra. Arithmetic, storage, and one pass data movement follow from the active block dimensions. Block diagonal plus low rank operators remain closed under multiplication before recompression. Their inverse is reduced by Woodbury, and their correction can be recompressed optimally through a small singular value problem.

Sparse transfer graphs predict path support, triangular schedule depth, and possible fill under elimination. Packet diagonal structure gives local Krylov spaces and a certified block Jacobi condition bound.

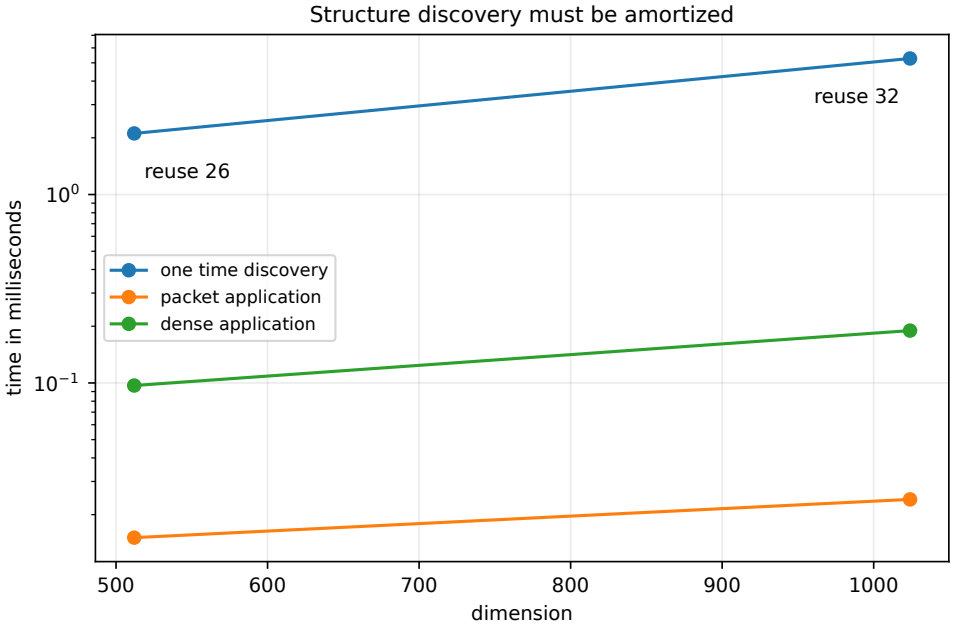


Figure 22. A profitable repeated representation can remain unprofitable for one use.

Packetwise perturbations give a direct mixed precision certificate. Continuous tolerances follow from a convex allocation, while finite data type choices lead to an exact multiple choice budget program.

A finite candidate refinement tree supports a genuine packet compiler. The Lagrangian optimum, the quantized error budget optimum, and local recompilation are exact under additive hypotheses. This is narrower than unrestricted automatic discovery, but it turns one broad compiler conjecture into a proved algorithmic statement.

The measured corpus shows the practical distinction between regular and irregular structure. Equal block and BDLR operators gained substantially on the recorded GPU. Recursive HODLR, Strassen, iterative refinement, a small transform, and an attempted overlap lost to recorded baselines. Discovery, transfer, capture, and materialization changed the break even point.

The computational principle is therefore precise. Remove work through a verified representation. Preserve the declared output. Compile only inside a model whose costs and errors are stated. Assign precision by sensitivity. Amortize one time costs. Keep negative controls. Treat the final timing as a property of the representation, device, kernel, and reuse regime together.

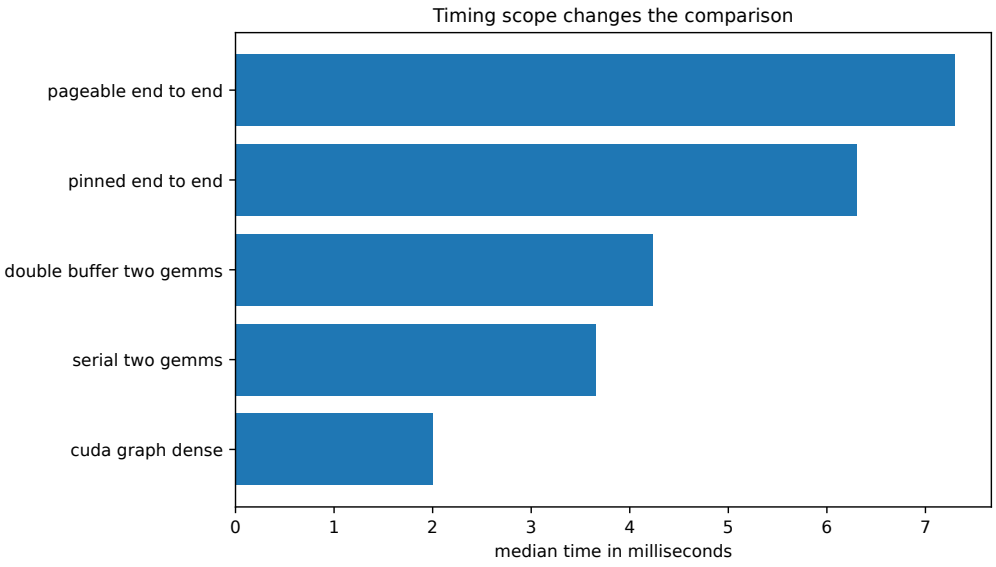


Figure 23. Device residence, transfer, graph replay, and a multioutput pipeline have different timing scopes.

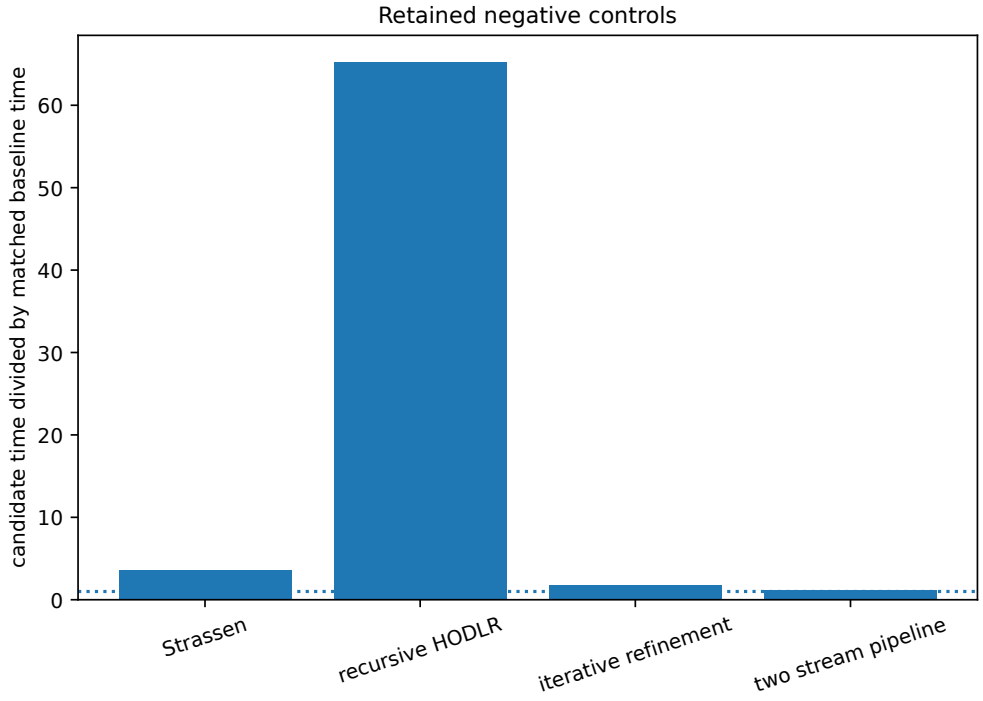


Figure 24. Values above one are slowdowns against the matched baseline.

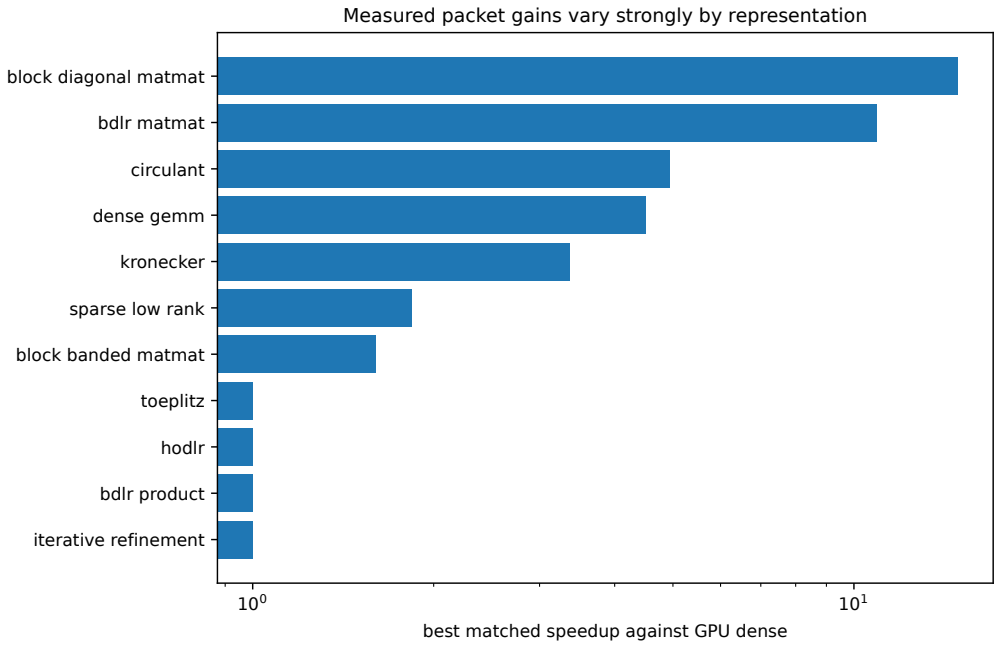


Figure 25. The best recorded gain differs strongly across suites. Regular packet structure dominates the largest measured rows.

A Exact semantics, support, and low-rank identities

This appendix develops exact packet semantics, support propagation, low-rank algebra and error models. Positive-budget conditions are isolated before the allocation theorem in Appendix B.

A.1 Exact packet semantics

Source items 2.3, 2.5, and 2.6

Let R be a ring, I a finite index type, and $(P_a)_{a\in I}$ a family satisfying

$$P_bP_a = \begin{cases} P_a & b = a, \\ 0 & b \neq a, \end{cases} \qquad \sum_a P_a = 1.$$

Write $A_{ba} = P_bAP_a$ and let $E_A = \{(b,a) : A_{ba} \neq 0\}$. These are exact algebraic zeros, not values discarded by a floating-point threshold.

Reconstruction and action. The checked finite sum is

$$\boxed{A = \sum_{(b,a)\in E_A} A_{ba}.}$$

The proof inserts the complete projector sum on both sides and removes only zero summands. Applying this identity to an arbitrary module gives exact action. A separate finite-matrix theorem establishes

$$AZ = \sum_{(b,a)\in E_A} A_{ba}Z$$

for every rectangular batch of right-hand sides Z.

Composition. Inserting the projector identity between factors yields

$$\boxed{(AB)_{ba} = \sum_c A_{bc}B_{ca}.}$$

If the left side is nonzero, at least one intermediate c has both $A_{bc} \neq 0$ and $B_{ca} \neq 0$. Consequently $\operatorname{supp}(AB) \subseteq \operatorname{supp}(A) \circ \operatorname{supp}(B)$. No equality of supports is asserted.

Adjoint and tensor products. With an involution and self-adjoint projectors, the formalization proves

$$(A_{ba})^* = (A^*)_{ab}.$$

For actual finite matrices and actual Kronecker products, it proves

$$(P_b \otimes Q_d)(A \otimes B)(P_a \otimes Q_c) = A_{ba} \otimes B_{dc}.$$

The self-adjointness hypothesis is explicit in the adjoint theorem. The tensor theorem is not a statement about an uninterpreted tensor symbol.

Concrete coordinates. Given any tag map from finite scalar coordinates to packet addresses, `coordinatePackets` constructs diagonal zero-one matrices. Their orthogonality and completeness are proved. The construction allows unequal packet sizes and unused packet addresses. Its action is checked entrywise: it retains exactly the coordinates carrying the selected tag.

The underlying ring-valued projector calculus is imported from the unchanged Chapter VII foundation, which was recompiled in the recorded formal run. Reuse is recorded separately from the new Chapter VIII declarations.

A.2 Support propagation and nilpotence

Source items 6.1, 6.2, and 6.3

A path of length n in the Lean predicate `TransferPath` is a directed walk: vertices may repeat. This is the notion required by matrix powers. At length zero the endpoints coincide. At length $n+1$, a nonzero last block is appended to a length-n walk.

Define the active contribution sum recursively by

$$F_0(b,a) = \begin{cases} P_a & b = a, \\ 0 & b \neq a, \end{cases}$$
$$F_{n+1}(b,a) = \sum_{c:\, A_{bc}\neq 0} A_{bc} F_n(c,a).$$

The checked identity is

$$\boxed{P_b A^n P_a = F_n(b,a) \qquad (n \geq 0).}$$

Its proof is induction using the exact composition theorem, not an assumed support recurrence. When no active walk exists, the corresponding contribution sum is zero, and hence the actual block of A^n vanishes.

Nilpotence certificate. If no walk of length $h+1$ exists between any pair of packets, then

$$\boxed{A^{h+1} = 0.}$$

Every block vanishes, and reconstruction forces the whole operator to vanish. A second theorem derives the required absence of walks from a supplied level map

$$\ell : I \longrightarrow \{0, \dots, h\}, \qquad A_{bc} \neq 0 \implies \ell(c) < \ell(b).$$

It proves that a length-n walk increases the level by at least n. This certifies the acyclic case with a bounded height. This block does not implement or verify an algorithm that constructs a topological ordering or computes the longest path.

Polynomial support. For the actual finite sum

$$p(A) = \sum_{j=0}^{d} c_j A^j,$$

the checked support lies in the union of the support relations of the powers. The formalization treats the $j = 0$ contribution explicitly, so the constant term is not accidentally omitted.

Cancellation control. Lean verifies the rational example

$$A = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \qquad A^2 = 0.$$

The coordinate block A_{00} is nonzero, and a length-two active walk from zero to zero exists. Nevertheless the corresponding block of A^2 is zero. This is an exact counterexample to replacing support inclusion by equality.

A.3 Low-rank algebra and inversion

Source items 5.1, 5.3, and 5.4

Let $A = D_A + U_A V_A^*$ and $B = D_B + U_B V_B^*$ have rectangular factor widths r_A and r_B. The formalization constructs

$$U_C = \begin{bmatrix} D_A U_B + U_A (V_A^* U_B) & U_A \end{bmatrix},$$
$$V_C = \begin{bmatrix} V_B & D_B^* V_A \end{bmatrix}$$

and proves

$$\boxed{AB = D_A D_B + U_C V_C^*.}$$

These are actual concatenated matrices with $r_B + r_A$ columns. Expanding the product and grouping the terms with common right factors proves the identity. The separate rank theorem establishes

$$\operatorname{rank}(AB - D_A D_B) \le r_A + r_B.$$

It bounds actual matrix rank, not just the number of stored columns. Packet diagonality of $D_A D_B$ follows from diagonality of both factors and the checked block-composition formula.

Woodbury. For D and

$$S = I + V^* D^{-1} U$$

both invertible, Lean checks the ordinary matrix inverse formula

$$\boxed{(D + UV^*)^{-1} = D^{-1} - D^{-1} U S^{-1} V^* D^{-1}.}$$

The proof specializes Mathlib's classical Woodbury identity. A separate theorem derives invertibility of $D + UV^*$ from the two stated assumptions. A further theorem verifies that multiplying the proposed inverse by any batch Z solves the updated system. Invertibility of the update is not inserted as an extra premise.

Norm bound. Over the real or complex scalars, all matrix norms in the following theorem are Hilbert-space operator norms, including the rectangular factors:

$$\boxed{\left\|(D + UV^*)^{-1}\right\| \le \left\|D^{-1}\right\| + \left\|D^{-1}U\right\| \left\|S^{-1}\right\| \left\|V^* D^{-1}\right\|.}$$

After the checked inverse identity, the proof uses the triangle inequality and two rectangular operator-norm product bounds.

Not covered here. The arbitrary product-chain rank corollary, optimal small-core recompression, singular-value approximation theorem, chain perturbation bounds, and numerical stability of a particular solve implementation are not marked complete by these results. Algebraic representability is not a GPU execution-time guarantee.

A.4 Packet perturbation certificates

Source item 8.1 and two uniform consequences

Let X be a normed space over a nontrivially normed field. The formal objects are actual continuous linear maps $X \to X$, with their operator norm. Let $a(e)$ identify the source packet of edge e, and suppose

$$\Delta_e = \Delta_e P_{a(e)}.$$

Set

$$A = \sum_e A_e, \qquad \widehat{A} = \sum_e (A_e + \Delta_e).$$

The exact operator and vector identities are proved first:

$$\widehat{A} - A = \sum_e \Delta_e P_{a(e)}, \qquad (\widehat{A} - A)x = \sum_e \Delta_e (P_{a(e)} x).$$

Applying the finite triangle inequality and the defining operator-norm estimate gives

$$\boxed{\left\|(\widehat{A} - A)x\right\| \le \sum_e \|\Delta_e\| \left\|P_{a(e)} x\right\|.}$$

If $\|\Delta_e\| \le \varepsilon_e \|A_e\|$, then the formal relative certificate is

$$\boxed{\left\|(\widehat{A} - A)x\right\| \le \sum_e \varepsilon_e \|A_e\| \left\|P_{a(e)} x\right\|.}$$

The source additionally assumes that each exact block is source supported. The proof of this bound does not need that assumption. The checked statement therefore uses weaker sufficient hypotheses without weakening the conclusion.

Uniform operator estimates. Two additional checked consequences are

$$\left\|\widehat{A} - A\right\| \le \sum_e \|\Delta_e\| \left\|P_{a(e)}\right\|$$

and, when $\|P_a\| \le 1$ for every packet,

$$\boxed{\left\|\widehat{A} - A\right\| \le \sum_e \|\Delta_e\|.}$$

These estimates are uniform in the input vector and can be used as inputs to later chain analyses. The chain results themselves are not included in this block's completed inventory.

Implementation boundary. The maps Δ_e and their bounds must be supplied or justified for a numerical implementation. This formalization does not establish IEEE rounding behavior, prevent overflow or underflow, model a CUDA kernel, or certify the recorded device speedups. Those are distinct claims and require separate evidence.

A.5 Positive budget and feasibility

Positive-budget condition for item 8.4

Positive budget is necessary. The source minimizes

$$\sum_e c_e \log \frac{1}{\varepsilon_e}, \qquad \varepsilon_e > 0, \qquad \sum_e s_e \varepsilon_e = \tau,$$

with positive sensitivities and costs. The positive-budget condition is essential. For a nonempty edge set, Lean proves

$$s_e > 0 \text{ and } \varepsilon_e > 0 \text{ for all } e \quad \Longrightarrow \quad \sum_e s_e \varepsilon_e > 0.$$

It follows that every nonpositive budget is infeasible. A one-edge, unit-sensitivity, zero-budget instance is checked as an exact boundary example.

The nonempty-index statement requires $\tau > 0$. Nonemptiness also makes the sum of positive costs in the displayed optimizer's denominator positive. The empty edge set is treated separately. Existence, global optimality and uniqueness for the positive-budget problem are proved in Section B.4.

B Exact compilation, continuous precision, and explicit costs

This appendix gives finite compiler constructions, continuous allocation and explicit storage and operation models. All optimization statements retain their independent admissibility conditions.

B.1 Exact finite-tree compilation

Source items 9.4 and 9.5

A candidate tree is either a terminal node or an internal node with a nonempty finite family of children. Arbitrary finite branching is supported. Every node carries a label from which keep cost κ_v, keep error ϵ_v, and split overhead σ_v are read.

Independent admissible objects. A legal cut is defined independently of the optimizer. It either keeps the root of its subtree or, at an internal node, chooses one legal cut in every child. A terminal node can only be kept. This is the recursive form of selecting one node on each root-to-terminal path.

The cost and error of a cut L are computed recursively. With multiplier λ, its objective is

$$J_\lambda(L) = \mathrm{cost}(L) + \lambda\,\mathrm{err}(L).$$

Lean proves that this objective equals the recursively accumulated local keep score $\kappa_v + \lambda\epsilon_v$, with σ_v charged at every split.

Recurrence. For terminal nodes,

$$F_\lambda(v) = \kappa_v + \lambda\epsilon_v.$$

For internal nodes,

$$\boxed{F_\lambda(v) = \min\left\{\kappa_v + \lambda\epsilon_v,\ \sigma_v + \sum_{w\in\mathrm{child}(v)} F_\lambda(w)\right\}.}$$

The checked result has two parts. First, $F_\lambda(v) \le J_\lambda(L)$ for every legal cut of the subtree. Second, the constructed function `compile` returns a legal cut with objective exactly $F_\lambda(v)$. Thus the minimum is attained, not merely bounded below.

The proof proceeds by induction on the actual tree and case analysis on the independently defined cut. In the split case, it sums the child lower bounds. The returned cut chooses the lower root alternative and recursively uses attaining child cuts. Ties are resolved in favor of keeping.

Shared operator families. For reuse weights r_j, the local keep model can aggregate setup and per-operator costs:

$$\kappa_v = \kappa_v^{\mathrm{setup}} + \sum_j r_j \kappa_v^{(j)}, \qquad \epsilon_v = \sum_j r_j \epsilon_v^{(j)}.$$

Lean proves the aggregation identity at the cut level, including split overhead, before applying the compiler theorem. The proof is valid over an ordered field, including $\mathbb{Q}$ and $\mathbb{R}$. The intended nonnegative model is a specialization.

No running-time theorem is asserted for the returned-cut wrapper. It can recompute subtree scores when constructing the cut. The minimum-value recurrence and the separately checked cached-update algorithm have distinct implementations and claims.

B.2 Frontiers and decision stability

Source items 9.6, 9.7, and 9.11

Let L_1, L_2 be actual optima at $0 \le \lambda_1 < \lambda_2$. Their optimality inequalities are

$$\begin{aligned} \operatorname{cost}(L_1) + \lambda_1 \operatorname{err}(L_1) &\le \operatorname{cost}(L_2) + \lambda_1 \operatorname{err}(L_2), \\ \operatorname{cost}(L_2) + \lambda_2 \operatorname{err}(L_2) &\le \operatorname{cost}(L_1) + \lambda_2 \operatorname{err}(L_1). \end{aligned}$$

Adding them and using the strict multiplier ordering gives

$$\boxed{\operatorname{err}(L_2) \le \operatorname{err}(L_1), \qquad \operatorname{cost}(L_2) \ge \operatorname{cost}(L_1).}$$

These conclusions hold for any choices among tied optima. For the cuts returned by the compiler, the required optimality premises are supplied by its constructive correctness theorem.

Supported Pareto points. For $\lambda > 0$, a scalarized optimum cannot be dominated in both cost and error with one strict improvement. Lean proves this directly from the two inequalities and positivity of λ.

The converse is not asserted. For the three cost/error points

$$(0, 2), \qquad (3/2, 1), \qquad (2, 0),$$

the middle point is undominated, but it is not selected by scalarization. A checked rational inequality gives, for every $\lambda \in \mathbb{Q}$,

$$\min\{2\lambda, 2\} < 3/2 + \lambda.$$

The quantized-budget compiler can recover unsupported points. Its construction is given in Section C.5.

Strict half-gap stability. At a solved internal node write

$$K = \kappa_v + \lambda \epsilon_v, \qquad S = \sigma_v + \sum_w F_\lambda(w), \qquad g = |K - S|.$$

If the updated scores satisfy

$$|K' - K| < g/2, \qquad |S' - S| < g/2,$$

Lean proves

$$K' \le S' \quad \Longleftrightarrow \quad K \le S.$$

Applying this condition recursively at every internal node proves equality of the entire returned cuts before and after perturbation, not only equality of one numerical minimum.

The hypotheses concern *solved* split scores, including changes propagated from descendants. A bound on raw local costs alone is not substituted for this condition. At an exact tie $g = 0$, the strict inequalities cannot hold. No robustness at ties is claimed.

B.3 Cached local recompilation

Source item 9.12

The implementation represents an edit using an explicit one-hole context. Each ancestor frame stores its parent label, the untouched sibling subtrees, the child slot containing the edit, and cached sibling optimum values. The context terminates at the root.

Let v be the new optimum of the edited subtree, and let j be the edited child slot at one frame. The update performed at that frame is

$$\boxed{\operatorname{step}(v) = \min \left\{ K,\ \sigma + \sum_i \begin{cases} v, & i = j, \\ \widehat{F}_i, & i \neq j. \end{cases} \right\}.}$$

This result is passed to the next ancestor. The update function does not recursively evaluate sibling subtrees. It reads their stored values $\widehat{F}_i$.

Cache validity. A context is valid when each untouched sibling cache equals the optimum for its unchanged subtree. The function `refresh` constructs such a context from the original tree model, and Lean proves its validity. This assumption concerns old, unchanged sibling computations. It is not an assumption about the desired new root optimum.

Correctness of the actual update. For any edited subtree T' and valid context C, Lean proves

$$\operatorname{updateValue}(C, F(T')) = F(\operatorname{fill}(C, T')).$$

The proof inducts through the context. At each frame, the updated child contributes the new value, and each other child contributes its unchanged cached optimum. The frame recurrence is therefore exactly the full-tree recurrence.

The result is connected back to legal cuts: the updated value is no larger than the objective of any cut of the edited whole tree, and some cut attains it. The latter witness follows from the independently proved compiler theorem.

Unchanged context. Separate theorems show that a nonedited sibling subtree is unchanged by the child replacement, and its minimum value is consequently unchanged. A localized change of keep/error/overhead data can be represented by replacing just the corresponding subtree while leaving the surrounding model fixed.

This formalizes the dependency-local claim: recompute the edited subtree, then update the ancestor frames using old sibling caches. It does not assert that a real compiler has no nonlocal cache, rank, allocation, or device-scheduling interactions. Those interactions are outside the additive tree model.

B.4 The continuous allocation theorem

Source item 8.4, with an explicit positive budget

Let E be a finite nonempty edge set. Assume

$$s_e > 0, \qquad c_e > 0, \qquad \tau > 0,$$

and define $C = \sum_e c_e$. The optimization problem is

$$\min_{\varepsilon_e > 0} \sum_e c_e \log \frac{1}{\varepsilon_e} \qquad \text{subject to} \qquad \sum_e s_e \varepsilon_e = \tau.$$

Lean constructs the displayed candidate

$$\boxed{\varepsilon_e^* = \frac{\tau c_e}{s_e C}}$$

and proves strict positivity and the exact budget equality. It then proves global optimality and uniqueness for the actual logarithmic objective.

Global comparison. For any other feasible ε, set $r_e = \varepsilon_e / \varepsilon_e^* > 0$. The budget identities give

$$\sum_e c_e r_e = C.$$

Using the standard analytic inequality $\log r \le r - 1$ yields

$$\sum_e c_e \log r_e \le \sum_e c_e (r_e - 1) = 0.$$

Lean separately proves the exact objective-difference identity

$$J(\varepsilon) - J(\varepsilon^*) = -\sum_e c_e \log r_e.$$

Therefore $J(\varepsilon) \ge J(\varepsilon^*)$.

If the feasible vectors differ, at least one $r_e \ne 1$. At that coordinate the logarithm inequality is strict. Since $c_e > 0$, the finite weighted sum is strict, giving

$$\varepsilon \ne \varepsilon^* \quad \Longrightarrow \quad J(\varepsilon) > J(\varepsilon^*).$$

The final theorem characterizes every feasible global minimizer as exactly the constructed formula. No unproved stationary-point, convexity, or KKT optimality assertion is supplied as a hypothesis.

Why the hypothesis matters. Section A.5 shows that positive sensitivities and positive tolerances on a nonempty edge set force a positive budget. The theorem above states that condition explicitly and proves the unique global optimum.

This is a real-valued mathematical optimizer. It is not an executable floating-point implementation. Clipped allocation is treated in Sections D.5–D.6. The discrete precision program is treated in Section C.4.

B.5 Explicit storage and operation models

Source items 3.1, 3.2, 3.4, 3.6, 3.8–3.10, and 4.2

For a finite collection of rectangular blocks, the scalar addresses are actual elements of

$$\coprod_{e \in E} (\mathrm{Fin}(m_e) \times \mathrm{Fin}(n_e)).$$

The corresponding operation addresses include a right-hand-side index and one of two scalar-operation tags. Lean proves their cardinalities rather than merely defining a formula named “cost.”

Representation	Scalar storage	Modeled work for q right sides
Dense	n^2	$2qn^2$
Packet blocks	$\sum_e m_e n_e$	$2q\sum_e m_e n_e$
Equal diagonal blocks	nb	$2qnb$
BDLR	$\sum_i n_i^2 + 2nr$	$2q(\sum_i n_i^2 + 2nr)$
Equal-block BDLR	$n(b+2r)$	$2qn(b+2r)$
Sparse plus low rank	$z + 2nr$	$2q(z+2nr)$

For equal blocks, the layout condition is $n = |I|b$. Under nonzero denominator conditions, the dense-to-block reduction is n/b and the BDLR reduction is $n/(b+2r)$. The BDLR output thin product accumulates into the diagonal result under the declared model of two operations per coefficient.

Streaming. The explicit event type counts one block read, one source read, and a target read plus write for every edge. With scalar size s, its byte count is

$$B_{\text{stream}} = s\sum_e (m_e n_e + n_e q + 2m_e q).$$

No cross-edge cache reuse is assumed. Sparse index metadata and temporary workspace remain separate.

Capacity and roofline. Positive physical capacities and the inequalities $F \le PT$, $B \le WT$ give

$$T \ge \max\{F/P, B/W\}.$$

For positive bytes and elapsed time, the rate form is $F/T \le \min\{P, WF/B\}$. The dense memory bound is the exact equivalence $n^2 s \le M \iff n \le \sqrt{M/s}$ for nonnegative n, M and positive s.

Materialization. The declared dense product followed by one addition at each output entry has exactly $2n^2 r + n^2$ modeled events and n^2 output entries. The remainder is bounded by n^2 for every n, r, giving a concrete quadratic certificate for the manuscript's $O(n^2)$ term.

These theorems count a mathematical model. They do not verify CUDA instructions, wall time, memory system behavior, or a lower bound for every implementation.

B.6 Contracts, grouping, and amortization

Source items 2.9, 10.2, 10.4–10.6, and 11.3

Output conversion. A decoded run returns the actual pair

$$(\text{decode}(\text{compressed}(x)),\ t_{\text{run}}(x) + t_{\text{decode}}(\text{compressed}(x))).$$

Under the declared output equality, Lean proves that this returns the same explicit result and charges the conversion time. It also proves the downstream composition identity when compressed output is consumed directly. Omitting a positive decode cost strictly understates the modeled explicit-output time. This formalizes the protocol. It cannot certify the truth of measured times.

Shape grouping. Active shape classes are the image of the complete shape-signature map. Their count is at most the number of active blocks. The fiberwise sum over classes equals the original block sum, both for additive outputs and for modeled work. Thus grouping neither loses nor duplicates

contributions. The ability to process one class in one batched launch remains a backend capability premise.

Exact profitability. With setup s, baseline time t_b, candidate time t_c, and $t_c < t_b$, Lean proves

$$\boxed{s + Rt_c < Rt_b \quad \Longleftrightarrow \quad R > \frac{s}{t_b - t_c}.}$$

For $s \geq 0$ and natural-number reuse counts, the smallest profitable count is

$$\boxed{R_{\text{min}} = \left\lfloor \frac{s}{t_b - t_c} \right\rfloor + 1.}$$

It is proved both profitable and minimal. At equality there is only break-even. A checked example gives setup 6, baseline 5, candidate 3: three uses break even, and four are the first profitable count. Zero setup gives a first profitable count of one when there is a per-use advantage.

When the candidate is no faster per use, nonnegative setup and reuse cannot give strict profitability. Graph capture/replay is a direct specialization. For materialization, with nonnegative conversion cost and reuse, Lean proves the full equivalence: profitability requires both a lower explicit per-use time and reuse above the conversion threshold.

Fair matching. The benchmark signature has separate case, output-contract, correctness-reference, and timing-scope fields. Equality of all four fields is proved equivalent to signature equality, hence reflexive, symmetric, and transitive. Measured time is deliberately not a field in this equivalence. Matching metadata is not proof of measurement accuracy.

B.7 Place in the finite development

The integer-budget compiler and discrete precision program follow in Appendix C. Structured transforms and clipped allocation are developed in Appendix D, graph support and schedules in Appendix E, and optimal small-core recompression in Appendix F.

The accompanying source files preserve the compiler, regression and proof-development records.

C Product chains, Krylov spaces, and integer budgets

This appendix develops product-chain bounds, packet Krylov identities and exact integer-budget programs. Cost, error and candidate-set assumptions remain explicit.

C.1 Low-rank factors through a product chain

Source item 5.2

Let

$$A_j = D_j + U_j V_j^*, \qquad U_j, V_j \in \mathbb{C}^{n \times r_j},$$

with all D_j diagonal in one packet layout. Set

$$P_m = A_m \cdots A_1, \qquad D^{(m)} = D_m \cdots D_1.$$

The formalization constructs thin matrices L_m, R_m such that

$$\boxed{P_m = D^{(m)} + L_m R_m^*.}$$

Their column index type is a recursively constructed disjoint sum. Its cardinality is proved to equal $\sum_{j=1}^{m} r_j$.

Construction. Suppose $P_{m-1} = D^{(m-1)} + L_{m-1} R_{m-1}^*$. The new factors are

$$L_m = \begin{bmatrix} D_m L_{m-1} + U_m (V_m^* L_{m-1}) & U_m \end{bmatrix},$$
$$R_m = \begin{bmatrix} R_{m-1} & (D^{(m-1)})^* V_m \end{bmatrix}.$$

Substitution into the already checked two-factor product identity proves exactness. This is not a rank-only existence argument. The recursively defined factors are themselves audited.

Applying the ordinary matrix-rank inequality gives

$$\boxed{\operatorname{rank}(P_m - D^{(m)}) \le \sum_{j=1}^{m} r_j.}$$

The bound concerns the actual correction matrix, not just the number of allocated columns. The proof also shows that $D^{(m)}$ stays packet diagonal. The empty chain has identity base, empty column type, and zero correction. Individual $r_j = 0$ are permitted.

What is not completed here. This is an unrecompressed representation. Optimal small-core singular-value recompression, item 5.6, remains separate. The construction can accumulate redundant columns and is not a claim of minimal factor width.

C.1.1 Actual noncommutative products

The error module defines the left-accumulated product recursively, with $P_0 = I$ and $P_{j+1} = A_{j+1} P_j$. For a perturbed sequence $\widehat{A}_j$, it derives the discrepancy recursion

$$\widehat{P}_{j+1} - P_{j+1} = \widehat{A}_{j+1}(\widehat{P}_j - P_j) + (\widehat{A}_{j+1} - A_{j+1}) P_j.$$

The identity follows from the actual products. It is not supplied as an assumption to a norm estimate, and it does not commute the operators. Lean uses zero-based indices internally, so index j in source code corresponds to manuscript factor $j+1$.

C.2 Error propagation and recompression

Source items 5.8, 5.9, and 8.3

For $\widehat{A}_j = A_j + E_j$, the checked product-chain bound is

$$\boxed{\left\|\widehat{A}_m \cdots \widehat{A}_1 - A_m \cdots A_1\right\| \le \sum_{j=1}^{m} \left(\prod_{i=j+1}^{m} \left\|\widehat{A}_i\right\|\right) \|E_j\| \left(\prod_{i=1}^{j-1} \|A_i\|\right).}$$

Empty products are one. The proof first solves a scalar gain recurrence as a finite sum, then bounds the actual discrepancy recursion by it. A version with certified upper bounds $\delta_j \ge \|E_j\|$ is included. The product-norm argument uses a norm-one identity, as in operators on a nonzero normed space.

Post-product recompression is different. Define the implemented iterates by

$$\widetilde{P}_0 = I, \qquad \widetilde{P}_j = A_j \widetilde{P}_{j-1} + E_j.$$

Here E_j is a residual introduced after multiplication. Lean derives

$$\widetilde{P}_j - P_j = A_j(\widetilde{P}_{j-1} - P_{j-1}) + E_j$$

and proves

$$\boxed{\left\|\widetilde{P}_m - P_m\right\| \le \sum_{j=1}^{m} \left(\prod_{i=j+1}^{m} \|A_i\|\right) \delta_j, \quad \|E_j\| \le \delta_j.}$$

This theorem works in a normed ring without the extra norm-one condition. It includes the empty iteration. There is no exact-prefix factor in the local residual term, because the residual is already measured at the post-product output.

Packetwise mixed precision. At each stage let

$$A_j = \sum_e A_{j,e}, \qquad \widehat{A}_j = \sum_e (A_{j,e} + \Delta_{j,e}), \qquad \Delta_{j,e} = \Delta_{j,e} P_{j,a(e)}.$$

The previous packet certificate supplies the uniform operator bound

$$\left\|\widehat{A}_j - A_j\right\| \le \delta_j := \sum_e \|\Delta_{j,e}\| \left\|P_{j,a(e)}\right\|.$$

The new theorem substitutes these δ_j into the actual product-chain estimate. The resulting vector error is at most the operator certificate times $\|x\|$. A vector-specific packet bound is not silently reused as an operator norm.

These are deterministic implications from stated residual bounds. They do not establish the rounding behavior of a floating-point kernel. The floating-point matrix-product model remains one of the separately tracked classical inputs.

C.3 Exact and approximate packet Krylov spaces

Source items 7.2 and 7.3

Write

$$\mathcal{K}_m(A, v) = \operatorname{span}\{v, Av, \dots, A^{m-1}v\}.$$

For a complete finite projector family P_i, suppose A commutes with every P_i. Let $A_i = P_i A P_i$. Lean proves the actual orbit identity

$$A^j v = \sum_i A_i^j P_i v$$

and therefore

$$\boxed{\mathcal{K}_m(A, v) \subseteq \bigoplus_i \mathcal{K}_m(A_i, P_i v).}$$

The case $j = 0$ uses $\sum_i P_i = I$. The zeroth local power is the ambient identity, applied to $P_i v$, rather than an incorrectly substituted local projector power.

The direct sum is justified. Each local space is proved to lie in the actual range of P_i. Projecting a finite sum of local vectors selects its ith summand. Consequently a zero sum of local vectors has every summand zero. The proof does not assert equality between the global Krylov space and the full sum of local Krylov spaces.

For approximate invariance let $A = D + K$, where D is packet diagonal, and set

$$M = \max\{\|A\|, \|D\|\}.$$

For every positive integer j, the operator estimate is

$$\boxed{\left\|A^j - D^j\right\| \le j M^{j-1} \|K\|.}$$

The proof is noncommutative and uses actual powers. Its positive-index version works even in a trivial normed ring. No division by M is used, so $M = 0$ causes no exception.

Define the comparison space

$$\mathcal{L}_{j+1} = \bigoplus_i \mathcal{K}_{j+1}(P_i D P_i, P_i v).$$

Packet diagonality is first shown to imply commutation. The exact theorem then proves $D^j v \in \mathcal{L}_{j+1}$. Using that vector as a witness in the infimum distance gives

$$\boxed{\operatorname{dist}(A^j v, \mathcal{L}_{j+1}) \le j M^{j-1} \|K\| \|v\|.}$$

This statement uses Lean's actual metric infimum distance to the subspace. It does not assume that the comparison vector lies there. The block-Jacobi condition bound and the preconditioned conjugate-gradient corollary are not completed by these Krylov results.

C.4 Integer error budgets and discrete precision

Source item 8.7 and common minimum machinery

The finite budget programs use integer weights and exact ordered-field costs. They apply to rational costs for execution and to real costs for mathematical statements. Unreachable states are represented by `none`, not by a large finite sentinel.

For a predicate S on costs, the common minimum specification means:

$$\begin{aligned} &\texttt{none}: && \text{no cost satisfies } S,\\ &\texttt{some}(v): && S(v) \text{ and } v \le z \text{ whenever } S(z). \end{aligned}$$

Both attainment and minimality are included. Optional minimum and optional addition are proved to implement union and independent sum of feasible sets.

For exact-weight tables F, G, min-plus convolution is

$$(F \star G)(b) = \min_{0 \le k \le b} \{F(k) + G(b-k)\},$$

with an infeasible sum whenever either operand is infeasible. Its correctness theorem identifies every finite answer with an actual feasible pair of component weights and costs. Repeated convolution handles arbitrary finite lists of components.

Independent precision assignments. For every edge, a finite list contains its permitted pairs (w, c) of quantized error and cost. The legal assignment type chooses exactly one listed pair per edge. It is defined independently of the dynamic program. Costs and weights are then summed over that assignment.

If D is the already solved tail table, the next edge has exact-weight recurrence

$$F(b) = \min_{(w,c)\ \text{allowed},\ w \le b} \{D(b-w) + c\}.$$

The final answer is $\min_{0 \le b \le B} F(b)$. Lean proves that this is the exact minimum over legal assignments with total error at most B, or `none` exactly when no such assignment exists.

The subtraction guard matters. Natural subtraction truncates at zero. Without the condition $w \le b$, an option of weight one could be accepted at budget zero. The implemented program checks the condition before subtraction. Dedicated regressions also cover a zero-weight option, an empty option set, and the empty collection of edges. The last case has cost and error zero.

This result does not prove that an unquantized physical error model is accurately represented by integer weights. The formal optimum concerns the declared quantized model. Clipped continuous allocation, item 8.5, remains separate.

C.5 The quantized packet-tree compiler

Source item 9.8

The new program uses the previous block's actual finite `CandidateTree` and independent legal `Cut` type. An internal node has any nonempty finite family of children. A cut either keeps its node or chooses a legal cut in every child. A terminal node can only be kept.

Keeping node v costs κ_v and incurs integer error w_v. Splitting costs overhead σ_v in addition to the selected child cuts. The exact-weight recurrence is

$$F_v(b) = \min \begin{cases} \kappa_v & \text{if } b = w_v, \\ \sigma_v + \min\limits_{\sum_i b_i = b} \sum\limits_i F_{v_i}(b_i) & \text{if } v \text{ is internal.} \end{cases}$$

Nonexistent alternatives are infeasible. The child sum is implemented by the proved finite min-plus convolution, not by assuming an optimal allocation.

Exact semantics. For every tree and every integer weight b, Lean proves that $F_v(b)$ is a minimum of costs of independent legal cuts of exact weight b. Thus each finite answer has an attaining cut and is a lower bound for every other legal cut at that weight. Infeasibility is equivalent to the absence of such a cut.

The root scan

$$\boxed{\min_{0 \le b \le B} F_{\text{root}}(b)}$$

is therefore the exact budget-constrained optimum. Increasing the budget cannot increase a finite optimum. The theorem retains feasibility hypotheses where necessary rather than treating `none` as an ordinary real cost.

Why the budget program adds something. Consider the three admissible cost/error pairs

$$(0, 4), \qquad (3, 2), \qquad (4, 0).$$

At budget two the answer is cost three. The middle pair cannot minimize cost plus λ times error: comparison with the first pair would require $\lambda \ge 3/2$, while comparison with the third would require $\lambda \le 1/2$. The executable suite checks the budget answer on an actual unary candidate tree with these cuts. It is not obtained merely by varying the previous Lagrangian compiler's multiplier.

Output boundary. This budget implementation returns the minimum cost or `none`. Attainment is proved, but executable backtracking metadata for a selected cut is not yet returned. Universal packet discovery, overlapping decompositions, nonadditive cache costs, and arbitrary device scheduling remain outside the theorem.

C.6 Cached arrays and a work certificate

Source item 9.9 and implemented forms of 8.7 and 9.8

A recursive value equation alone need not be an efficient program. Repeated calls can recompute the same subtree table. The new implementation therefore stores each table as an array-backed `Vector` with exactly $B + 1$ optional entries.

Each child table is evaluated and cached before the parent merges it. A merge reads two existing tables and tabulates their convolution. The discrete precision program likewise computes its tail table once before filling the next edge's table. The final scan reads the stored root or final-edge table.

Entrywise implementation theorem. For every $b \le B$, Lean proves

$$\text{lookup}(\text{storedTree}(B, t), b) = F_t(b).$$

There is a parallel theorem for stored discrete precision. The proof establishes first that an output entry at b depends only on input entries at most b, then proceeds through cached convolution and the tree

recursion. Consequently the optimality, attainment, and infeasibility specification applies to the actual stored result.

Counted candidates. One full convolution table examines indices

$$(b, k), \qquad 0 \le b \le B, \quad 0 \le k \le b.$$

These form a concrete finite dependent-sum type. Its cardinality is proved to be

$$\sum_{b=0}^{B} (b+1) \le (B+1)^2.$$

For a nonempty finite tree, the number of child merges plus one equals the node count N. This identity is proved for arbitrary finite branching.

Let $W_B(t)$ be the declared unit-cost count of table-cell constructions and tested min-plus candidates. The model charges the empty child-combination table as well as the output cells and merge candidates. Lean proves

$$W_B(t) \le 2(B+1)N + \big((B+1) + (B+1)^2\big)(N-1)$$

and hence

$$\boxed{W_B(t) \le N(3(B+1) + (B+1)^2).}$$

The root minimum scan adds a linear pass. Thus the declared tabulation model is pseudo-polynomial in N and the numeric budget B.

Scope of this certificate. These are table and candidate counts, not a measured execution trace or a theorem about every instruction emitted by the compiler. Rational bit complexity, allocation overhead, memory traffic, and hardware timing are not included. The certificate is not polynomial in the binary input length of B.

C.7 Place in the finite development

The structured primitives and clipped-allocation theorem follow in Appendix D. Graph schedules and preconditioning certificates are developed in Appendix E. Appendix F gives the small-core optimality argument.

The accompanying source files preserve the compiler, regression and proof-development records.

D Structured transforms and clipped precision

This appendix proves the structured-transform identities and the global clipped-allocation theorem, including the maximal-budget endpoint.

D.1 Circulant Fourier diagonalization

Source item 4.3; module CirculantFourier

Let $n > 0$, with coordinates indexed by $\mathbb{Z}/n\mathbb{Z}$. Define the actual circulant by

$$C(c)_{jk} = c_{j-k}.$$

The formal Fourier matrix is

$$F_{jk} = \frac{1}{\sqrt{n}}\,\chi(-jk), \qquad \chi(a) = \exp\left(\frac{2\pi i a}{n}\right).$$

The standard character is not left abstract: a checked theorem identifies these entries with

$$\frac{1}{\sqrt{n}}\exp\left(-\frac{2\pi i\ \mathrm{val}(j)\,\mathrm{val}(k)}{n}\right).$$

The value representatives enter only this explicit exponential formula. All modular algebra uses the residue classes themselves.

Normalization and inversion. The proof identifies matrix application with the ordinary counting-measure discrete Fourier transform:

$$Fx = n^{-1/2}\,\mathrm{DFT}(x).$$

It separately identifies F^*x with the appropriately scaled inverse transform. Mathlib's checked Fourier inversion theorem and the proved scalar normalization yield both identities

$$\boxed{F^*F = I, \qquad FF^* = I.}$$

No unitarity assumption is inserted.

Convolution and reconstruction. An explicit bijective change of index proves

$$\mathrm{DFT}(C(c)x)_k = \mathrm{DFT}(c)_k\,\mathrm{DFT}(x)_k.$$

This gives the matrix intertwining relation $FC(c) = \mathrm{diag}(\mathrm{DFT}\,c)F$. Multiplication by the proved inverse reconstructs the circulant:

$$\boxed{C(c) = F^*\,\mathrm{diag}(\sqrt{n}\,Fc)F.}$$

The two-transform application formula is also checked for every input vector. The result includes $n = 1$ and arbitrary complex coefficients. Hermiticity is not required.

Claim boundary. This verifies the exact Fourier representation, not an $O(n \log n)$ implementation of the transform. The latter is an additional algorithmic factorization of the same linear map.

D.2 Toeplitz padding and column-major vectorization

Source items 4.4 and 4.5

Toeplitz embedding. For $n > 0$, define

$$T_{ij} = \begin{cases} c_{i-j}, & j \le i, \\ r_{j-i}, & i < j. \end{cases}$$

The diagonal is supplied by c_0, and the coordinate r_0 is unused. The constructed generator of length $2n$ is

$$g = (c_0, \dots, c_{n-1}, 0, r_{n-1}, \dots, r_1),$$

and the padding map is the actual finite-index sum eliminator that appends n zeros.

Lean first proves the upper-left block identity

$$C(g)_{ij} = T_{ij} \qquad (0 \le i, j < n).$$

For $j \le i$, finite modular subtraction has its ordinary value $i - j$. For $i < j$, it has wrapped value $2n + i - j$, whose reversed generator coordinate is $j - i$. Both branches are checked with the actual finite-index arithmetic.

Splitting the product sum into the original and padded halves then gives

$$\boxed{\left(C(g)(x, 0)^T\right)_{0:n} = Tx.}$$

The generator's first half and its middle zero are checked separately. Neither symmetry nor conjugation is involved.

Kronecker application. For $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ and $X \in \mathbb{C}^{n\times m}$, Lean proves

$$\boxed{(A \otimes B)\,\mathrm{vec}(X) = \mathrm{vec}(BXA^T).}$$

The formal vectorization is explicitly column-major: its pair coordinate (j, i) stores X_{ij}. The right factor is the *ordinary transpose*, not the conjugate transpose. This distinction is essential over $\mathbb{C}$.

Finite types of multiplication/accumulation events and stored scalar addresses give

$$\boxed{2n^2m + 2nm^2 \text{ modeled operations}, \qquad n^2 + m^2 \text{ factor entries}.}$$

The explicit Kronecker matrix has n^2m^2 entries. As in the manuscript, the operation model assigns two scalar operations to each multiply-accumulate term.

D.3 Arbitrary tensor factors

Source item 4.6; module StructuredTensor

Let $A_0, \dots, A_{k-1}$ be square factors of order d. Tensor coordinates are represented by a nested product of k copies of $\{0, \dots, d-1\}$. The empty product is a singleton. Lean proves that the coordinate type has cardinality d^k.

The represented matrix is defined recursively using the actual Kronecker product:

$$M_0 = [1], \qquad M_{k+1} = A_0 \otimes M_k(A_1, \dots, A_k).$$

The algorithm is defined independently. On each slice with fixed first coordinate, it recursively contracts the trailing modes. It then applies A_0 to the resulting first-coordinate fibers:

$$y_{i,u} = \sum_{j=0}^{d-1} (A_0)_{ij} \ \mathrm{Apply}_k(A_1, \dots, A_k; x_{j,\cdot})_u.$$

Induction, the Kronecker entry formula, and finite-sum distributivity prove

$$\boxed{\mathrm{Apply}_k(A; x) = M_k(A)x.}$$

Thus correctness is not inferred merely from agreeing dimension or storage formulas.

Operation events. The event type for $k+1$ modes is a disjoint union of d copies of the trailing event type and the events for the final fiber contraction. Its cardinality obeys

$$W_0 = 0, \qquad W_{k+1} = dW_k + 2d^{k+2}.$$

Lean proves the closed form

$$\boxed{W_k = 2kd^{k+1}.}$$

The factor-address type has cardinality

$$\boxed{kd^2.}$$

These statements include $k = 0$ and $d = 0$. With no factors, the action is the identity on the singleton coordinate type and incurs zero modeled arithmetic. With $d = 0$ and at least one factor, the coordinate type is empty.

Implementation boundary. These are recursive mathematical contractions with exact complex arithmetic. The event model assumes each intermediate slice is computed once and reused. It is not an instrumented execution trace or a cost bound for every evaluator of the raw functional definition. The count excludes indexing, allocation, data movement, fused instruction effects, and the real-operation cost of a complex scalar operation. Those require a separately declared backend model.

D.4 Walsh butterfly and exact addition count

Source item 4.7; module WalshTransform

The two-point matrix is constructed as

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and the full family is defined by $H_1 = [1]$ and

$$H_{2N} = H_2 \otimes H_N = \begin{pmatrix} H_N & H_N \\ H_N & -H_N \end{pmatrix}.$$

The two-point orthogonality identity is checked explicitly. Kronecker multiplication and induction then prove

$$\boxed{H_N H_N^T = NI, \qquad N = 2^k.}$$

The ordinary transpose is sufficient because all constructed entries are real, even though the scalar field is $\mathbb{C}$.

Actual butterfly. For the two halves x_0, x_1 of the input, the independently defined procedure returns

$$\mathrm{Walsh}_{2N}(x_0, x_1) = (\mathrm{Walsh}_N(x_0 + x_1), \mathrm{Walsh}_N(x_0 - x_1)).$$

Lean proves that this vector equals $H_{2N}x$. The argument expands the two first-coordinate cases and applies finite-sum distributivity. It does not assume that the recursive algorithm already represents the matrix.

Exact arithmetic count. Each pair of slices forms one sum and one difference at every trailing coordinate. The event type therefore gives

$$A_0 = 0, \qquad A_{k+1} = 2A_k + 2^{k+1}.$$

The closed form and its equivalent logarithmic expression are both checked:

$$\boxed{A_k = k2^k = N \log_2 N.}$$

The logarithm in the formal count is the natural-number binary logarithm, with its power-of-two identity proved by Mathlib. The case $N = 1$ has zero arithmetic events.

A further action theorem gives

$$H_N(H_N^T x) = Nx,$$

which identifies the inverse after division by the nonzero scalar N. This result concerns exact arithmetic, not floating-point error accumulation through a butterfly implementation.

D.5 Clipped precision: existence and multiplier uniqueness

Source item 8.5; module ClippedPrecision

Let E be a finite nonempty set and suppose

$$s_e > 0, \qquad c_e > 0, \qquad m_e > 0, \qquad 0 < \tau \le \sum_e s_e m_e.$$

The feasible set is

$$0 < \varepsilon_e \le m_e, \qquad \sum_e s_e \varepsilon_e = \tau,$$

and the objective is the same logarithmic objective used in item 8.4:

$$J(\varepsilon) = \sum_e c_e \log(1/\varepsilon_e).$$

Strictly positive caps are stated explicitly. A zero or negative cap cannot admit a positive tolerance at that coordinate.

Constructive parameterization. Write $t = 1/\lambda$ and set

$$u_e(t) = \min\{m_e, tc_e/s_e\}, \qquad \Phi(t) = \sum_e s_e u_e(t).$$

Lean proves that Φ is continuous and nondecreasing, with $\Phi(0) = 0$. At the finite positive scale

$$T = \sum_e \frac{m_e s_e}{c_e},$$

every coordinate is saturated, so $u(T) = m$ and $\Phi(T) = \sum_e s_e m_e$. The intermediate value theorem produces $t > 0$ with $\Phi(t) = \tau$. The multiplier is not assumed to exist.

Strictness below saturation. If $t < u$ and $\Phi(t)$ is below the maximum, at least one coordinate is unsaturated at t. That coordinate strictly increases, and every other coordinate is nondecreasing. Hence $\Phi(t) < \Phi(u)$. This proves uniqueness of the scale and, by reciprocation, the exact manuscript statement

$$\boxed{\exists!\lambda > 0 : \quad \sum_e s_e \min\left\{m_e, \frac{c_e}{\lambda s_e}\right\} = \tau}$$

when $0 < \tau < \sum_e s_e m_e$.

At the maximal budget, the minimizing vector remains unique but the multiplier need not be. The one-coordinate case $s = c = m = 1$ is checked: both $\lambda = 1$ and $\lambda = 1/2$ produce the same tolerance 1.

D.6 Clipped precision: global optimality

Source item 8.5, continued

Existence of a multiplier alone does not prove optimality. The formal proof supplies a global supporting inequality, rather than invoking an assumed Karush–Kuhn–Tucker optimality oracle.

Fix the feasible vector $u = u(t)$ obtained above and any feasible competitor ε. For each coordinate, Lean proves

$$\boxed{c_e\left(\frac{\varepsilon_e}{u_e} - 1\right) \le \frac{s_e}{t}(\varepsilon_e - u_e).}$$

When $u_e = tc_e/s_e$, this is equality. When $u_e = m_e$, the capped condition implies $s_e/t \le c_e/m_e$, while $\varepsilon_e - m_e \le 0$. Multiplying by this nonpositive difference gives the required inequality in the correct direction.

The equal-budget constraint then yields

$$\sum_e c_e\left(\frac{\varepsilon_e}{u_e} - 1\right) \le \frac{1}{t}\sum_e s_e(\varepsilon_e - u_e) = 0.$$

Applying the checked inequality $\log z \le z - 1$ for $z > 0$ gives

$$\sum_e c_e \log(\varepsilon_e/u_e) \le 0.$$

Since

$$J(\varepsilon) - J(u) = -\sum_e c_e \log(\varepsilon_e/u_e),$$

we obtain $J(u) \leq J(\varepsilon)$ for every feasible competitor.

If $\varepsilon \neq u$, one coordinate has ratio different from 1. The logarithmic inequality is strict there, and its positive weight preserves strictness in the sum. Consequently,

$$\boxed{\varepsilon \neq u \quad \Longrightarrow \quad J(u) < J(\varepsilon).}$$

This proves the unique global minimizing vector for every positive feasible budget, including the maximal endpoint.

Numerical boundary. Existence uses real continuity and the intermediate value theorem. This is a mathematical construction, not an executable floating-point multiplier search with a certified stopping tolerance. The companion exact-rational water-filling routine is an independent regression mirror and is not itself the compiled Lean construction.

D.7 Place in the finite development

Appendix E develops the remaining graph, schedule, counting and preconditioning constructions. Appendix F proves the global small-core recompression theorem from its stated factorization data.

The accompanying source files preserve the compiler, regression and proof-development records.

E Constructed graph schedules, support, and preconditioning

This appendix develops support bounds, graph schedules, block discovery and preconditioning certificates. Classical algorithmic inputs remain separate from the finite certificates proved here.

E.1 Constructing the optimal level schedule

Source items 6.6 and 10.3; module GraphSchedules

Let V be finite and let $u \to v$ mean that task v depends on task u. Assume there is no nonempty directed cycle. A topological order or valid level function is *not* supplied as an additional hypothesis.

For each vertex, define its strict ancestor set

$$\mathrm{Anc}(v) = \{u : \text{there is a positive-length path } u \rightsquigarrow v\}.$$

For every edge $u \to v$, the proof establishes

$$\mathrm{Anc}(u) \subsetneq \mathrm{Anc}(v).$$

The inclusion follows by appending the edge. It is strict because u is an ancestor of v but cannot be its own strict ancestor in an acyclic graph. Cardinality therefore decreases when following predecessors. This constructs a well-founded relation on which the level recursion is defined:

$$\boxed{\ell(v) = 1 + \max_{u \to v} \ell(u), \qquad \max \emptyset = 0.}$$

Validity and attainment. Lean proves $\ell(v) > 0$ and $\ell(u) < \ell(v)$ on every edge. A path of n edges from a to b satisfies

$$\ell(a) + n \le \ell(b).$$

Conversely, predecessor-max attainment and induction produce an actual path with $\ell(v) - 1$ edges ending at each vertex v. The result is an equality with a longest path, not merely an upper bound.

A valid schedule $s : V \to \mathbb{N}$ assigns positive stages and strictly increases on each edge. Induction proves the stronger pointwise statement

$$\ell(v) \le s(v) \quad \text{for every } v.$$

Consequently the constructed schedule achieves the least possible makespan,

$$\boxed{d = \max_{v \in V} \ell(v).}$$

For a nonempty graph, d equals the number of vertices on an attained longest path. Vertices of the same level have no dependency edge in either direction.

Empty graph and interpretation. When $V = \emptyset$, the checked depth is zero. The phrase "one plus the longest edge-path length" applies to nonempty graphs. These are ideal discrete scheduling statements. Weighted task costs, a finite processor budget, launch latency and communication are not part of this model. The definitions are mathematical constructions, not a certified hardware scheduler.

E.2 Forward substitution with an actual stage evaluator

Source item 6.5; module TriangularSolve

Use the graph and levels constructed on the previous page. A vertex v carries an arbitrary vector space X_v over a field. The spaces may have different dimensions. Let

$$D_v : X_v \xrightarrow{\sim} X_v, \qquad T_{vu} : X_u \longrightarrow X_v$$

be an invertible diagonal map and linear transfer maps. The operator being solved is defined by its actual block action:

$$(Ax)_v = D_v x_v + \sum_{u \to v} T_{vu} x_u.$$

No solution, solve equation or precomputed topological certificate is assumed.

Constructed solution. Well-founded recursion defines

$$\boxed{x_v = D_v^{-1}\left(b_v - \sum_{u \to v} T_{vu} x_u\right).}$$

Applying D_v proves the original equation at v. If another vector satisfies all equations, induction over predecessors makes each previously used coordinate identical, and invertibility of D_v forces agreement at v. Thus the construction gives existence and uniqueness, and the block operator is bijective.

Parallel stages. A second, independent definition is an explicit staged evaluator. Set $x^{(0)} = 0$ and

$$x_v^{(k+1)} = \begin{cases} D_v^{-1}\left(b_v - \sum_{u \to v} T_{vu} x_u^{(k)}\right), & \ell(v) = k+1, \\ x_v^{(k)}, & \ell(v) \neq k+1. \end{cases}$$

It reads the previous-stage vector, not partially updated coordinates. The invariant is

$$\ell(v) \leq k \quad \Longrightarrow \quad x_v^{(k)} = x_v.$$

Every predecessor of a level-$k+1$ vertex has level at most k, so the recurrence uses only coordinates already proved correct. At the canonical depth d this yields

$$\boxed{Ax^{(d)} = b.}$$

The local independence theorem separately proves that changing coordinates outside the strict earlier levels cannot change a vertex's local solve.

Scope. Diagonal invertibility is expressed by genuine linear equivalences. The task graph may include declared transfers whose values happen to be zero. Optimality refers to that declared dependency graph. Exact field arithmetic and exact local inverses are used. No floating-point stability or implementation-specific parallel speed is inferred.

E.3 Support of inverses and Schur elimination

Source items 6.7 and 6.8; modules InverseSupport and SchurSupport

Let (P_a) be a complete idempotent packet family in a finite-dimensional algebra. Define $a \rightsquigarrow b$ by an actual finite path through nonzero blocks $P_b A P_a$. **Zero-length paths are included**, so every packet reaches itself.

Fix the source packet a. In the regular representation, define the submodule

$$W_a = \{x : P_b x = 0 \text{ whenever } a \not\rightsquigarrow b\}.$$

It contains P_a. Completeness of the projectors expands each component of Ax into packet transfers. If a transfer into an unreachable packet were nonzero from a reachable packet, it would extend a path, a contradiction. Thus $AW_a \subseteq W_a$.

Suppose $BA = I$. Left multiplication by A is injective, including on W_a. Finite dimensionality makes that restricted map surjective. Given $x \in W_a$, choose $y \in W_a$ with $Ay = x$. Then $Bx = BAy = y \in W_a$. Applying this to P_a proves

$$\boxed{a \not\rightsquigarrow b \quad \Longrightarrow \quad P_b B P_a = 0.}$$

The matrix specialization takes B to Mathlib's actual nonsingular matrix inverse, with an invertible determinant. No inverse-sparsity hypothesis is inserted.

Hidden elimination. For the actual Schur complement

$$S = A_{RR} - A_{RH} A_{HH}^{-1} A_{HR},$$

the hidden inverse is expanded through the hidden projectors. A nonzero retained correction block requires indices j, k with nonzero entrance and exit blocks and a hidden path $j \rightsquigarrow k$. Therefore a nonzero block S_{ba} requires either a direct retained block, or

$$a \longrightarrow j \ \rightsquigarrow_H \ k \longrightarrow b.$$

The entrance and exit matrices may be rectangular. Retained and hidden coordinate sizes and packet systems are independently typed. The hidden matrix must be invertible. A zero-length hidden path permits entrance and exit through the same hidden packet.

Inclusion is not equality. Cancellation can remove a block permitted by a graph route. For example,

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{has} \quad (A^{-1})_{13} = 0$$

despite a direct nonzero $(1, 3)$ block. The implementation proves support inclusion and the corresponding no-route zero certificate, not a converse.

E.4 Exact banded addresses and modeled work

Source item 4.1; module BandedCount

For p packets and half-bandwidth w, the actual address set is

$$\mathcal{B}_{p,w} = \{(i, j) \in \{0, \dots, p-1\}^2 : i \le j + w \text{ and } j \le i + w\}.$$

The predicate is the integer version of $|i-j| \le w$. The count is first defined as a double sum of indicators, then identified with the cardinality of this subtype. It is not defined by the target closed form.

Under $0 \le w < p$, the formal proof establishes

$$\boxed{N_{p,w} = |\mathcal{B}_{p,w}| = p(2w+1) - w(w+1).}$$

The proof uses finite prefix counting, the newly exposed boundary row and column when p increases, and the saturated initial band. An intermediate subtraction-free identity avoids relying on natural-number subtraction without the required inequality. The diagonal case $w = 0$ gives exactly p blocks, and the saturated case $w = p - 1$ gives p^2.

Stored scalars. With equal block width b, the storage-address type is

$$\mathcal{S} = \mathcal{B}_{p,w} \times \{0, \dots, b-1\}^2.$$

Its cardinality is the actual number of represented scalar entries:

$$\boxed{|\mathcal{S}| = b^2 N_{p,w}.}$$

Arithmetic events. For q right-hand sides, the declared event type adds a right-hand-side index and one of two operation roles, multiplication or accumulation:

$$\mathcal{O} = \{0,1\} \times \{0, \dots, q-1\} \times \mathcal{S}.$$

Hence

$$\boxed{|\mathcal{O}| = 2qb^2 N_{p,w}.}$$

These identities include zero right-hand-side count. The abstract storage formula also admits zero block width. To use the manuscript's substitution $p = n/b$, one must separately require $b > 0$ and $n = pb$.

Boundary. Two events per coefficient and right-hand side are a declared arithmetic convention. These counts do not identify the number of CUDA instructions, account for fused multiply-add lowering, price complex arithmetic in real operations, or include allocation, indexing, communication and launch overhead. They are also not a performance comparison with dense multiplication.

E.5 Exact residual certificates for supplied packets

Source item 9.2; module DiscoveryCertificate

Let (P_a) be a complete orthogonal projector system on a finite complex coordinate space. The projectors need not be coordinate masks. For an actual matrix A, define

$$\Pi(A) = \sum_a P_a A P_a, \qquad E = A - \Pi(A).$$

Reconstruction gives the exact off-diagonal residual

$$E = \sum_{b \ne a} P_b A P_a.$$

Hilbert–Schmidt orthogonality. For distinct ordered packet pairs $(b, a) \neq (d, c)$, adjoint multiplication and projector orthogonality give

$$\operatorname{tr}((P_b A P_a)^* P_d B P_c) = 0.$$

If $b \neq d$, the central projector product vanishes. Otherwise $a \neq c$, and cyclicity of trace brings $P_c P_a$ together. This is proved for arbitrary input matrices A, B.

Lean identifies the actual Frobenius norm with its trace expression,

$$\|M\|_{\mathrm{F}}^2 = \operatorname{Re}\operatorname{tr}(M^* M),$$

then expands the finite sum and removes each cross term. The result is

$$\boxed{\|A - \Pi(A)\|_{\mathrm{F}}^2 = \sum_{b \neq a} \|P_b A P_a\|_{\mathrm{F}}^2 .}$$

No Pythagorean equality for the target residual is assumed.

Operator norm. In a separate scoped matrix norm, the triangle inequality gives

$$\boxed{\|A - \Pi(A)\|_2 \leq \sum_{b \neq a} \|P_b A P_a\|_2 .}$$

Here $\|\cdot\|_2$ is the Euclidean induced operator norm, not the Frobenius norm. The formal declarations use distinct norm instances, preventing the two certificates from being conflated. Self-adjointness is used for Frobenius orthogonality. The sum estimate itself requires only the complete packet decomposition.

Discovery boundary. This result certifies the residual of a *supplied* decomposition. It does not construct useful projectors from an arbitrary matrix, prove that a searched decomposition is optimal, or guarantee that its application is profitable on a GPU. The simultaneous spectral-decomposition input remains separately recorded as classical.

E.6 Block-Jacobi normalization and the CG factor

Source items 7.4 and 7.6; module BlockJacobi

Work on a nonempty finite complex coordinate space with Euclidean operator norms. Assume D and $A = D + K$ are positive definite. Define

$$S = (\sqrt{D})^{-1}, \qquad B = SAS, \qquad E = SKS.$$

The square root is the actual positive matrix square root supplied by continuous functional calculus, and the inverse is the actual matrix inverse. Positive definiteness makes $\sqrt{D}$ a unit. Its square identity and inverse identities prove

$$SDS = I, \qquad B = I + E.$$

A direct inverse bound. If $C = (I + E)^{-1}$ and $\|E\|_2 \leq \eta < 1$, then $C = I - EC$. Consequently

$$(1 - \eta)\,\|C\|_2 \leq 1, \qquad \|I + E\|_2 \leq 1 + \eta.$$

The constructed normalization is invertible because S and A are. Combining the estimates gives the actual condition number bound

$$\boxed{\kappa_2(B) := \|B\|_2 \left\|B^{-1}\right\|_2 \leq \frac{1+\eta}{1-\eta}.}$$

This norm argument does not require a separately assumed Hermitian perturbation. Nonnegative η follows from the norm bound. Nonemptiness is explicit because the condition-number and norm-of-identity convention uses a nonzero coordinate space.

The preconditioning corollary. Define

$$q(\kappa) = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$$

Monotonicity of square root and positive denominators prove that q is monotone. Lean also proves $\kappa_2(B) \geq 1$, $q(\kappa) \geq 0$ for $\kappa \geq 1$, and $q(\kappa) < 1$. Thus

$$\boxed{q(\kappa_2(B)) \leq q\left(\frac{1+\eta}{1-\eta}\right).}$$

For $1 \leq \kappa$ and $e_0 \geq 0$, an explicit classical premise $e_m \leq 2q(\kappa)^m e_0$ and a certified upper bound $\kappa \leq \bar{\kappa}$ yield $e_m \leq 2q(\bar{\kappa})^m e_0$.

What is not claimed. The classical conjugate-gradient convergence theorem, item 7.5, is not newly formalized. The new corollary proves its factor comparison and bound substitution. It does not replace the original theorem's algorithm hypotheses with a new axiom or claim a floating-point CG convergence proof.

E.7 Place in the finite development

The global small-core recompression theorem is proved in Appendix F. The classical conjugate-gradient algorithm theorem, floating-point product model and simultaneous spectral decomposition remain separately supplied inputs.

The accompanying source files preserve the compiler, regression and proof-development records.

F Global small-core recompression from ordered thin SVD data

This appendix derives global small-core recompression from actual compact ranges and supplied ordered thin-SVD equations. Optimality is proved against arbitrary ambient competitors.

F.1 Compact factors from actual column spaces

Modules CompactRangeFactors and SmallCoreTransfer

Let

$$L \in \mathbb{C}^{m\times t}, \qquad R \in \mathbb{C}^{n\times t}, \qquad A = LR^*.$$

Use the ordinary Euclidean inner products on the coordinate spaces. Define

$$S_L = \operatorname{range}(L), \quad r_L = \dim S_L, \qquad S_R = \operatorname{range}(R), \quad r_R = \dim S_R.$$

The implementation constructs an orthonormal basis of each actual range subspace. Let Q_L and Q_R be the matrices of the corresponding isometric inclusions. Express the range-restricted input maps in these bases to obtain R_L, R_R.

The checked identities are

$$\boxed{Q_L^*Q_L = I, \quad L = Q_LR_L, \quad R_L = Q_L^*L,}$$

with the analogous identities on the right. The dimension theorem is not an input:

$$\boxed{r_L = \operatorname{rank} L, \qquad r_R = \operatorname{rank} R.}$$

The coordinate factors have full row rank. The construction therefore handles rank-deficient input factors without retaining artificial column directions.

Define the small core by actual multiplication,

$$G = R_LR_R^* \in \mathbb{C}^{r_L\times r_R}.$$

Finite matrix associativity and adjoint identities give

$$\boxed{A = Q_LGQ_R^*, \qquad \operatorname{rank} G = \operatorname{rank} A.}$$

The rank identity follows from both multiplication inequalities and the left inverse obtained by compression. It is not inferred from agreeing dimensions.

Zero factors and empty spaces. No positive-rank or nonempty-coordinate hypothesis is imposed. If a range is zero, its orthonormal basis has an empty index type. The factorization uses the resulting rectangular empty matrices. A separate theorem proves that the compact dimension is zero exactly when the original factor is zero.

Relation to QR. The argument only needs orthonormal columns and exact reconstruction. A compact QR factorization supplies these properties, but triangularity of its coordinate factor is irrelevant to the recompression theorem. The formal construction uses an exact orthonormal range basis. It does not claim Householder steps, pivot selection or floating-point stability. A supplied noncompact orthonormal factorization is also covered by the general transfer theorem, including extra zero singular directions.

F.2 Why a small-core optimum is a global optimum

Module SmallCoreNorms; arbitrary ambient competitors

For rectangular isometries Q, V define

$$\mathcal{L}(G) = QGV^*, \qquad \mathcal{C}(B) = Q^*BV, \qquad Q^*Q = I, \quad V^*V = I.$$

The following statements are proved for the actual matrix rank and actual matrix norms:

$$\mathcal{C}(\mathcal{L}(G)) = G, \qquad \operatorname{rank}\mathcal{L}(G) = \operatorname{rank} G, \qquad \operatorname{rank}\mathcal{C}(B) \le \operatorname{rank} B,$$

$$\boxed{\|\mathcal{L}(G)\|_\nu = \|G\|_\nu, \qquad \|\mathcal{C}(B)\|_\nu \le \|B\|_\nu, \quad \nu \in \{2, \mathrm{F}\}.}$$

For the spectral norm, the proof uses the Gram matrix identity

$$(QG)^*(QG) = G^*G$$

and the C^*-norm square identity, together with right-side adjoint symmetry. Compression follows from submultiplicativity and the isometry norm bound, including empty spaces.

For the Frobenius norm, trace cyclicity proves left and right isometric invariance. The proof also establishes the projection identity

$$\|B - QQ^*B\|_{\mathrm{F}}^2 = \|B\|_{\mathrm{F}}^2 - \|Q^*B\|_{\mathrm{F}}^2.$$

Nonnegativity of the left side gives left compression contractivity. The right version follows by adjoints. No norm-preservation or contractivity hypothesis is inserted.

Suppose G_k is a best rank-at-most-k approximation to G in either norm. Take an *arbitrary* ambient matrix B with rank $B \le k$. Compression is an admissible small-core competitor and

$$\begin{aligned}\|\mathcal{L}(G) - \mathcal{L}(G_k)\|_\nu &= \|G - G_k\|_\nu \\ &\le \|G - \mathcal{C}(B)\|_\nu \\ &= \|\mathcal{C}(\mathcal{L}(G) - B)\|_\nu \\ &\le \|\mathcal{L}(G) - B\|_\nu.\end{aligned}$$

The competitor B is not required to lie in the supplied row and column spaces. The next two pages discharge the intermediate small-core optimality obligation, rather than leaving it as a hypothesis in the final theorem.

F.3 Spectral optimality from a kernel witness

Modules DiagonalTruncation and SpectralOptimality

Let $\sigma_1 \ge \cdots \ge \sigma_d \ge 0$ and

$$D = \operatorname{diag}(\sigma_1, \dots, \sigma_d), \qquad D_k = \operatorname{diag}(\sigma_1, \dots, \sigma_{\min(k,d)}, 0, \dots, 0).$$

Lean uses indices beginning at zero, while the formulas on this page use the conventional one-based notation. The rank of D_k is proved to be at most k, even when retained singular values are zero.

For $k < d$, take any $d \times d$ matrix B of rank at most k. Restrict its domain to the first $k+1$ coordinates using the actual coordinate inclusion J. Since

$$\operatorname{rank}(BJ) \le \operatorname{rank} B \le k < k+1,$$

rank-nullity supplies a nonzero vector z with $BJz = 0$. The vector $y = Jz$ is nonzero, satisfies $By = 0$ and is supported in the first $k+1$ coordinates. The injection and the support property are checked separately.

Monotonicity of the diagonal entries gives

$$\|Dy\|^2 = \sum_{i=1}^{k+1} \sigma_i^2 |y_i|^2 \ge \sigma_{k+1}^2 \|y\|^2 .$$

Because $(D-B)y = Dy$, the operator-norm inequality implies

$$\sigma_{k+1} \|y\| \le \|(D-B)y\| \le \|D-B\|_2 \|y\| .$$

Cancellation of the strictly positive norm of y proves

$$\boxed{\sigma_{k+1} \le \|D-B\|_2 .}$$

The norm of the actual diagonal residual $D - D_k$ equals σ_{k+1}, so D_k attains the lower bound. If $k \ge d$, the residual is zero and optimality follows from norm nonnegativity.

No optimality oracle. The nonzero kernel vector, its coordinate support and the diagonal norm bound are all derived. No Eckart–Young inequality, best-rank property or singular-tail formula is assumed. The final rectangular SVD result follows by the isometric transfer proved on the preceding page.

F.4 Frobenius optimality from kernel weights

Modules KernelFrame and FrobeniusOptimality

For an arbitrary competitor B with $\operatorname{rank} B \le k$, construct an orthonormal matrix Q spanning its actual kernel. Then

$$BQ = 0, \qquad Q^*Q = I, \qquad q = \dim\ker B = d - \operatorname{rank} B.$$

Define the row weights

$$w_i = \sum_{j=1}^{q} |Q_{ij}|^2.$$

The proof derives

$$\boxed{0 \le w_i \le 1, \qquad \sum_i w_i = q \ge \max(d-k, 0).}$$

The upper bound uses contraction of Q^* on a coordinate vector. The weight sum follows from the trace of Q^*Q and rank-nullity.

For any nonnegative decreasing sequence a_i, weights with these properties satisfy

$$\sum_i a_i w_i \geq \sum_{i>k} a_i.$$

This finite weighted inequality is itself proved. For $k < d$, use $\mu = a_{k+1}$ and compare the summands to tail weights $v_i = 0$ for $i \leq k$ and $v_i = 1$ otherwise:

$$a_i v_i + \mu(w_i - v_i) \leq a_i w_i.$$

For a retained coordinate the inequality uses $a_i \geq \mu$ and $w_i \geq 0$. For a tail coordinate it uses $a_i \leq \mu$ and $w_i \leq 1$. Summation and $\sum w_i \geq \sum v_i$ give the result.

Set $a_i = \sigma_i^2$. Right compression cannot increase the Frobenius norm, and $BQ = 0$, hence

$$\begin{aligned} \|D - B\|_{\mathrm{F}}^2 &\geq \|(D - B)Q\|_{\mathrm{F}}^2 \\ &= \|DQ\|_{\mathrm{F}}^2 = \sum_i \sigma_i^2 w_i \geq \sum_{i>k} \sigma_i^2. \end{aligned}$$

Direct calculation of $D - D_k$ attains equality. Thus the best-rank Frobenius assertion is proved from kernel geometry and finite sums, independently of the spectral lower-bound proof. Repeated and zero singular values require no uniqueness claim.

F.5 Thin output factors and the final theorem

Modules CompactTruncation and SmallCoreRecompression

For the actual core $G = R_L R_R^*$, supply ordered thin SVD data

$$G = W \operatorname{diag}(\sigma) Z^*, \qquad W^*W = I, \qquad Z^*Z = I, \qquad \sigma_1 \geq \cdots \geq \sigma_d \geq 0.$$

These are precisely the fields of the formal SVD structure. It contains no rank-optimality or tail-error field. A derived theorem checks the Gram eigenvector equation

$$G^*GZ = Z \operatorname{diag}(\sigma_1^2, \ldots, \sigma_d^2).$$

Let $h = \min(d, k)$ and retain exactly the first h column indices. The rectangular square-root mask has entries

$$M_{ij} = \begin{cases} \sqrt{\sigma_j}, & i = j, \\ 0, & i \neq j, \end{cases} \qquad 1 \leq i \leq d, \quad 1 \leq j \leq h.$$

Lean proves its index cardinality and $MM^* = D_k$. Define

$$\boxed{L_k = Q_L W M, \qquad R_k = Q_R Z M.}$$

The product equals $Q_L W D_k Z^* Q_R^*$ and has rank at most k. The principal theorem proves, for every ambient B with rank $B \leq k$,

$$\boxed{\left\|LR^* - L_k R_k^*\right\|_\nu \leq \left\|LR^* - B\right\|_\nu, \quad \nu \in \{2, \mathrm{F}\}.}$$

It also gives the exact errors

$$\|LR^* - L_k R_k^*\|_2 = \begin{cases} \sigma_{k+1}, & k < d, \\ 0, & k \geq d, \end{cases} \qquad \|LR^* - L_k R_k^*\|_{\mathrm{F}} = \left(\sum_{i>k} \sigma_i^2 \right)^{1/2}.$$

Both the squared and square-root Frobenius formulas are checked. All these statements are combined in

`KomorebiPaperVIII.SmallCore.small_core_optimal_recompression`.

Endpoints. Rank zero produces the zero output, while $k \geq d$ reconstructs the original product. An empty SVD certificate is constructed for a zero core, and diagonal SVD certificates are constructed explicitly. Zero singular values are never inverted. At repeated singular values the result asserts optimality, not uniqueness of the minimizer.

F.6 Place in the finite development

The 56-item manuscript-proved register is covered under its stated hypotheses. General SVD existence, a numerical SVD algorithm, floating-point CG convergence and hardware speedups do not follow merely from this formal scope. The formal companion identifies the exact declaration map and recorded verification evidence.

The accompanying source files preserve the compiler, regression and proof-development records.

How to use this volume

Follow each small operator through its factors and formulas, then read the cited general argument in the traditional volume. The Lean companion records the separate finite formal contract.

Routes through the guide

Plates VIII.1 through VIII.32 identify worked pages independently of PDF pagination. Each source line names the corresponding general statement in the mathematical text. The integrated atlas uses the same plate labels.

Route	Mathematical question	Plates
Representations	Which exact data represent the required operator?	1–4, 27
Structured transforms	Which conventions preserve the same map?	5–8
Low-rank algebra	What remains exact after products and recompression?	9–13, 29
Support and solves	Which interactions can occur or cancel?	14–18, 30
Precision and compilation	Which finite costs and errors does the optimizer use?	19–23, 31–32
Performance contracts	Which one-time costs and outputs are compared?	24–26
Evidence	What is checked, proved, or still open?	2, 28

Keep the represented object visible

A dense array, factored operator, and execution plan may describe the same map while returning different objects. Representation savings are exact modeled counts. Measured speed requires a separate experiment with matching output and timing contracts.

Preservation and evidence

These plates illustrate the chapter's finite constructions. The accompanying models and checking code distinguish exact examples, symbolic calculations, bounded enumerations and rejected controls. Drawing geometry and model checks have separate scopes.

The finite formal correspondence contains 56 manuscript-proved items, 392 named declarations in 40 modules and 66 construction audits. Nineteen declarations are reused foundation results. The ordered-SVD hypotheses, four classical entries, recorded timing corpus and three broad research targets retain their separate scopes. These records do not claim a new Lean compiler execution.

Read this as: Choose a question, calculate its small model, and check the hypotheses, output contract, and evidence level of the corresponding general result.

Worked visual atlas

Plate VIII.1 / THREE VIEWS OF ONE COMPUTATION

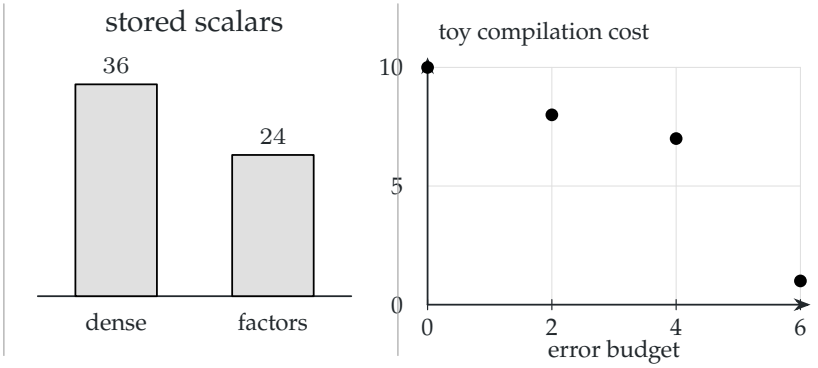
Compute the representation first

Chapter VIII: Section 2, Section 3, Appendix F.

A dense array, a factored operator, and a compiled execution plan are different objects. They may denote the same linear map, but they need not store the same data, return the same output, or incur the same costs. This guide illustrates those distinctions with calculations.

A: 36 visible entries

4	2	1	1	1	1
2	4	1	1	1	1
1	1	4	2	1	1
1	1	2	4	1	1
1	1	1	1	4	2
1	1	1	1	2	4



Three models to carry through the paper

The first is a six-dimensional operator $A = D + uv^T$. Every entry of A is nonzero, yet three 2×2 diagonal blocks and two thin vectors suffice to apply it exactly. Plate VIII.3, Plate VIII.9, Plate VIII.10, and Plate VIII.27 reuse this operator or its factors.

The second starts with redundant factors L and R. Actual range coordinates reduce a 4×3 product to a 2×3 core. Plate VIII.11–Plate VIII.12 connect rank, singular tails, and the geometric reason a small-core optimum is also an ambient optimum.

The third is a seven-node candidate tree with exactly five prunings. Its budget optimum is not selected by any Lagrangian penalty. Plate VIII.21–Plate VIII.23 let readers inspect every feasible choice and every changed subproblem.

How to use the guide

A source line beneath each title identifies the relevant result in the mathematical text. The preserved examples use rational, Gaussian-rational, symbolic, and finite-enumeration arithmetic. Portable checks use exact rational and Gaussian-rational arithmetic. The retained SymPy records list the assertions and rejected controls separately. Smooth drawing coordinates illustrate stated formulas. They are not additional proof evidence.

> **Read this as:** An exact representation can remove work before a device is chosen. It does not by itself supply a fast kernel, a good approximation, or a fair timing comparison.

Plate VIII.2 / A READER'S ATLAS

The objects and their contracts

Chapter VIII: Definition 2.1, Definition 2.2, Definition 2.7, Section 8, Section 9, Section 11.

The same letter can refer to a mathematical operator or to a particular stored representation. Keep those levels separate when reading an algorithm or a benchmark row.

Symbol	Object	Question to ask
P_α	Coordinate projector	Which input or output coordinates does it select?
$A_{\beta\alpha}$	A block map	What are its source and target dimensions?
E	Active block addresses	Is this exact support, or only a structural upper bound?
$D + UV^*$	Factored operator	Is the desired output a vector, factors, or a dense array?
Q_L, G, Q_R	Range factors and core	Are the outer columns orthonormal, with exact reconstruction?
E_j	An error operator	Is it a factor perturbation or an additive error after a product?
s_e, ϵ_e	Sensitivity and tolerance	Is the budget input-dependent or uniform over unit inputs?
$(\kappa_v, \epsilon_v, \sigma_v)$	Candidate-tree data	Do costs, errors, and split overheads really add?
$F_v(b)$	A budget table	Does b mean exact weight, or an upper bound?
R	Reuse count	Which one-time costs have been included?

Conventions used in the examples

Matrices act on column vectors. A block $A_{\beta\alpha}$ goes from source α to target β. Products act from right to left. Real examples use transpose, while complex examples use adjoint unless a vectorization identity explicitly requires transpose. Column-major vectorization stacks the columns of a matrix.

An arithmetic count uses the modeled multiply/add convention. It is not a count of executed machine instructions. Scheduling and amortization examples use declared synthetic units, rather than measured GPU times. The source files preserve the recorded benchmark corpus.

Evidence levels

The theorem ledger records its stated finite formal scope. Exact checks validate these particular examples. A plot samples a formula. A recorded timing describes its case and machine. These evidence types have different scopes.

Read this as: Before interpreting a number, identify its object, norm, output contract, and timing scope. Most apparent contradictions disappear once those four fields are explicit.

Plate VIII.3 / THE RUNNING SIX-DIMENSIONAL OPERATOR

A dense-looking matrix, one scalar coupling

Chapter VIII: Theorem 2.3, Theorem 3.4.

Let D have three identical blocks $\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$ and put $u = v = \mathbf{1} \in \mathbb{Q}^6$. Then $A = D + uv^T$ has diagonal entries four, within-block off-diagonal entries two, and every cross-block entry one. All 36 entries are nonzero.

$$x = (1, -1, 2, 0, -2, 3)^T, \qquad v^T x = 3, \qquad Ax = Dx + 3u.$$

x		Dx		$u(v^T x)$		Ax
1		2		3		5
−1		−2		3		1
2		6		3		9
0	+	2	=	3	=	5
−2		−3		3		0
3		7		3		10

$v^T x = 3$ is shared by all six output coordinates.

What was avoided?

The global part does not require 36 independent coefficients. It depends on the input through one scalar, $v^T x$, which is broadcast through u. The diagonal part is computed by three separate local matrix products. Nothing is approximated and no channel is discarded.

The sixth output is $7 + 3 = 10$, while the fifth is $-3 + 3 = 0$. A zero result can arise through cancellation even when the corresponding row contains no zero coefficient. The output vector is therefore not a support map of the input matrix.

What has not been established?

This computation proves neither that the supplied factors are the cheapest representation nor that six small tasks outperform one dense kernel. For example, the repeated diagonal blocks and all-one vectors admit further special handling. The stated BDLR model deliberately treats the blocks as dense and the factors as stored vectors so that the paper's general formulas apply literally.

The checker constructs the complete matrix independently, multiplies it by x, and compares all six entries with the factored route. The same input returns in the product calculation on Plate VIII.27.

> **Read this as:** A matrix can have full entrywise support and still admit a small exact representation. Entry sparsity and algebraic compressibility are different properties.

Plate VIII.4 / ARITHMETIC, STORAGE, AND CAPACITY

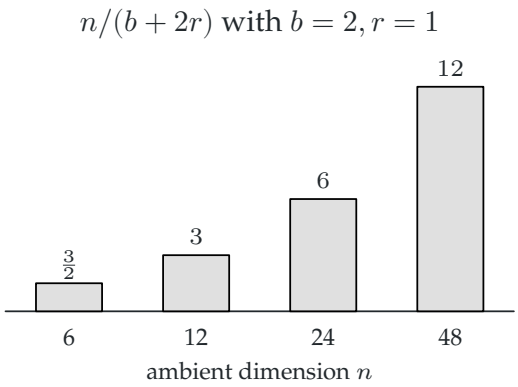
Work removed is not time measured

Chapter VIII: Theorem 3.1, Corollary 3.2, Theorem 3.4, Corollary 3.9, Proposition 4.1.

For q right-hand sides, the paper models dense application by $2n^2q$ operations. A block-diagonal-plus-rank-r representation with equal block width b uses

$$F = 2qn(b + 2r), \qquad M = n(b + 2r) \text{ stored scalars.}$$

These formulas count the dense block products and two thin products. They follow the manuscript convention rather than adding every scalar accumulation or index operation.



Six-dimensional running operator

dense: 36 scalars, $72q$ work

BDLR: 24 scalars, $48q$ work

ratio: $3/2$, not a timing result

For the running example, $n = 6, b = 2, r = 1$. The model reduces storage from 36 to 24 scalars and application work from $72q$ to $48q$. For fixed b, r, the ratio $n/(b + 2r)$ grows with n. If the rank instead grows proportionally with dimension, that apparent advantage need not grow.

Memory is another constraint

With scalar size s, the representation alone requires sM bytes. Inputs, outputs, workspace, allocator overhead, and stored indices are additional. A matrix-only capacity inequality is not a complete device-memory fit test.

For a block band of p packets and half-bandwidth w, the exact active-block count is

$$p(2w + 1) - w(w + 1).$$

The subtraction removes overcounted edge diagonals. The checker enumerates all addresses for $1 \le p \le 6$ and every admissible w, including $w = 0$ and $w = p - 1$.

> **Read this as:** An arithmetic or storage ratio explains a representation advantage. Kernel launch cost, traffic, utilization, and conversion determine whether that advantage becomes a time advantage.

Plate VIII.5 / CIRCULANT MULTIPLICATION

Four coefficients, four exact frequencies

Chapter VIII: Theorem 4.3.

Take $c = (2, 1, 0, -1)^T$ and define $C(c)_{jk} = c_{j-k \bmod 4}$. The unitary Fourier matrix is $F_{jk} = \mathrm{i}^{-jk}/2$. Its normalization is visible in the exact diagonalization:

$$C(c) = F^* \operatorname{diag}(2Fc) F, \qquad 2Fc = (2, 2 - 2\mathrm{i}, 2, 2 + 2\mathrm{i})^T.$$

$C(c)$

2	−1	0	1
1	2	−1	0
0	1	2	−1
−1	0	1	2

$\sqrt{4}\,Fc$

2
$2 - 2i$
2
$2 + 2i$

$$F_{jk} = \tfrac{1}{2}\,\mathrm{i}^{-jk}$$

$$C = F^* \operatorname{diag}(\sqrt{4}Fc) F$$

$\sqrt{4} = 2$ is essential.

A unitary transform changes the normalization of the convolution eigenvalues.

Follow an actual input

For $x = (1, 2, -1, 0)^T$, dense multiplication gives $(0, 6, 0, -2)^T$. Transforming x, multiplying by the four displayed eigenvalues, and applying F^* gives exactly the same vector. Every entry belongs to $\mathbb{Q}(\mathrm{i})$, so the check needs no approximate trigonometric values.

There is a common normalization trap. With an unnormalized discrete transform, the convolution multiplier has one convention. With the unitary transform used here, the eigenvalue vector is $\sqrt{n}\,Fc$, not Fc. Dropping the factor two reconstructs $C/2$ rather than C in this example. The invalid control deliberately tries that change and rejects it.

Where the fast algorithm enters

The identity specifies the represented operator. An FFT is an additional factorization that evaluates the transforms efficiently. The algebraic diagonalization does not certify a particular floating-point FFT, data layout, or launch policy. For a tiny four-point example, direct multiplication may be the simpler execution path even though the frequency representation is exact.

Read this as: Transform coordinates change both the storage pattern and the normalization conventions. Verify the inverse transform and convolution scale before interpreting a frequency-domain product.

Plate VIII.6 / TOEPLITZ EMBEDDING WITHOUT WRAPAROUND

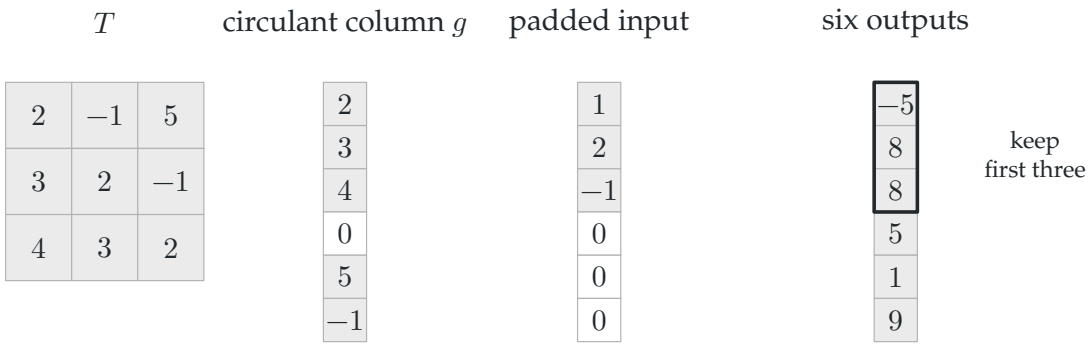
Padding is part of the theorem

Chapter VIII: Corollary 4.4.

Consider

$$T = \begin{pmatrix} 2 & -1 & 5 \\ 3 & 2 & -1 \\ 4 & 3 & 2 \end{pmatrix}, \qquad x = (1, 2, -1)^T.$$

The diagonal pattern is Toeplitz, not circulant. To use a circular convolution, embed it into dimension six with first column $g = (2, 3, 4, 0, 5, -1)^T$ and pad the input with three zeros.



Padding is part of the operator identity, not optional workspace.

Read the wrapped indices

For output index $j < 3$, an input index $k \le j$ uses the first-column coefficient c_{j-k}. An input index $k > j$ wraps to the stored first-row coefficient r_{k-j}. The padded input positions contribute nothing. Keeping the first three outputs therefore gives

$$Tx = (-5, 8, 8)^T.$$

The remaining circular outputs are workspace consequences of the embedding, not additional components of Tx.

A concrete failure mode

Using only the first column $(2, 3, 4)$ as a three-dimensional circulant changes the upper-triangular coefficients. That is a different matrix. The checker compares its output with Tx and rejects the unpadded shortcut.

The middle zero in g is not a discarded Toeplitz coefficient: it separates the lower and reversed upper diagonals in the chosen $2n$ embedding. Other embeddings require their own index identities. It is not sufficient for an array merely to have the right length or to produce a plausible plot.

> **Read this as:** Zero padding prevents the wrong circular interactions. It is an algebraic part of the operator representation, not an optional performance tweak.

Plate VIII.7 / KRONECKER ACTION AND ARRAY ORDER

A transpose, not an adjoint

Chapter VIII: Theorem 4.5, Corollary 4.6.

With column-major vectorization, the tensor identity is

$$(A \otimes B)\,\mathrm{vec}(X) = \mathrm{vec}(BXA^T).$$

Here the transpose records index order, not a Hermitian inner product. Use the explicitly complex matrix $A = \begin{pmatrix} 1 & \mathrm{i} \\ 2 & -1 \end{pmatrix}$, together with $B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$ and $X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$.

X

1	2
3	4

$\mathrm{vec}(X)$

1
3
2
4

BXA^T

$5 + 8i$	2
$9 + 12i$	6

A^T not A^*

$$(A \otimes B)\,\mathrm{vec}(X) = \mathrm{vec}(BXA^T)$$

Check the coordinates

The stacked input is $(1, 3, 2, 4)^T$. Direct tensor multiplication agrees with stacking the columns of

$$BXA^T = \begin{pmatrix} 5 + 8\mathrm{i} & 2 \\ 9 + 12\mathrm{i} & 6 \end{pmatrix}.$$

Replacing A^T with A^* changes the signs of the imaginary terms. A real-only test would conceal that error, which is why the worked model uses an imaginary coefficient.

Why modewise contraction matters

For A of order m and B of order n, the explicit tensor matrix stores m^2n^2 entries. Its factors store $m^2 + n^2$. Applying through BXA^T costs $2n^2m + 2nm^2$ modeled operations, rather than forming the full tensor matrix first.

A different flattening convention has a different coordinate formula. The statement is not independent of how an array is reshaped. In a software implementation, shape, stride, and flattening order belong to the correctness contract as well as the cost model.

> **Read this as:** Tensor structure gives exact savings only when the address convention is preserved. A silent transpose, conjugation, or reshape change implements a different map.

Plate VIII.8 / WALSH RECURSION IN EIGHT COORDINATES

See every butterfly value

Chapter VIII: Theorem 4.7.

The unnormalized Walsh matrix satisfies $H_1 = [1]$ and

$$H_{2n} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}, \qquad H_n H_n^T = nI.$$

Apply the three add/subtract stages to $(1, 2, 3, 4, 5, 6, 7, 8)^T$. The full intermediate ledger is displayed below.

input	1	2	3	4	5	6	7	8
stage 1	3	-1	7	-1	11	-1	15	-1
stage 2	10	-2	-4	0	26	-2	-4	0
stage 3	36	-4	-8	0	-16	0	0	0

$8 \log_2 8 = 24$ additions/subtractions; $H_8 H_8^T = 8I$.

Arithmetic can be counted exactly

Each stage visits eight coordinates, producing one sum or difference per coordinate. Three stages therefore use exactly 24 additions or subtractions. No multiplication by a dense 8×8 matrix is necessary.

The result $(36, -4, -8, 0, -16, 0, 0, 0)^T$ still reconstructs the input: the inverse is $H_8^T/8$. Because the displayed transform is unnormalized, its squared output norm is eight times the squared input norm. The normalized transform $H_8/\sqrt{8}$ would instead be orthogonal.

What the zeros mean

The zero coefficients arise from exact cancellation among the input values. They do not imply that the transform's defining matrix is sparse. The butterfly structure, not the final coefficient support, explains the arithmetic reduction.

The checker verifies the recursion, orthogonality, dense-versus-butterfly output, and operation count for sizes one, two, four, eight, and sixteen. These bounded tests exercise the implementation and boundary cases. The general recursion theorem remains the manuscript's mathematical statement.

> **Read this as:** Factorization can turn a dense matrix into a short exact arithmetic circuit. Keep its normalization separate from its recursive evaluation rule.

Plate VIII.9 / CLOSED LOW-RANK PRODUCT ALGEBRA

The cross term has somewhere to go

Chapter VIII: Theorem 5.1, Corollary 5.2.

Let $A = D_A + u_a v_a^T$ be the running operator. Take $D_B = \mathrm{diag}(1,2,3,4,5,6)$, $u_b = (1,0,1,0,1,0)^T$, and $v_b = (0,1,0,1,0,1)^T$. Define $B = D_B + u_b v_b^T$.

$$U_C = [D_A u_b + u_a(v_a^T u_b)\ \ u_a], \qquad V_C = [v_b\ \ D_B^T v_a].$$

U_C

6	1
4	1
6	1
4	1
6	1
4	1

V_C

0	1
1	2
0	3
1	4
0	5
1	6

$$AB = D_A D_B + U_C V_C^T$$

$$U_A(V_A^T U_B)V_B^T$$

must be retained

Two input rank-one corrections produce at most rank two, not three.

Why there are only two columns

Expanding AB gives three correction terms, but two share the right factor v_b^T:

$$D_A u_b v_b^T + u_a(v_a^T u_b)v_b^T = (D_A u_b + u_a(v_a^T u_b))v_b^T.$$

The remaining term is $u_a v_a^T D_B$. Thus $AB = D_A D_B + U_C V_C^T$ with at most two correction directions. Counting the three expanded summands as independent ranks would overestimate this representation.

The scalar $v_a^T u_b$ is three. Removing the quadratic term would therefore remove $3u_a v_b^T$, which is nonzero. The checker compares the complete product to the factor construction and rejects a calculation that omits this contribution.

Before and after recompression

A rank bound on stored factors is not always the actual matrix rank. Repeated products may introduce dependent columns. Recompression can remove those dependencies exactly, or remove small singular directions approximately under an explicit error budget.

> **Read this as:** Closure of a representation class is an algebraic statement about all product terms. It is not permission to drop an interaction merely because it is second order in the factors.

Plate VIII.10 / WOODBURY WITH BOTH INVERTIBILITY CHECKS

A large inverse, a small denominator

Chapter VIII: Theorem 5.3, Corollary 5.4.

For the running operator, each diagonal block has inverse $\frac{1}{8}\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$. Hence $D^{-1}u = u/4$ and

$$S = 1 + u^T D^{-1} u = \frac{5}{2}.$$

The Woodbury identity reduces the entire inverse to

$$A^{-1} = D^{-1} - D^{-1} u S^{-1} u^T D^{-1} = D^{-1} - \frac{1}{40} u u^T.$$

one block of D^{-1}

$\frac{3}{8}$	$-\frac{1}{8}$
$-\frac{1}{8}$	$\frac{3}{8}$

$$D^{-1}u = \tfrac{1}{4}u$$

$$S = 1 + u^T D^{-1} u = \tfrac{5}{2}$$

$$A^{-1} = D^{-1} - \tfrac{1}{40} u u^T$$

$I - e_1 e_1^T$

0	0
0	1

invertible D
singular update

Two hypotheses, not one

Invertibility of D alone does not suffice. For $D = I_2$, $u = e_1$, and $v = -e_1$, the update is $\operatorname{diag}(0, 1)$. The small matrix S is zero, exactly where the correction formula would require an invalid division.

For the running model, the checker multiplies the proposed inverse by A in both orders and obtains the identity. It also checks the determinant relation $\det A = \det D \det S$. These tests verify the actual factors and sign of the correction, not just a numerical solve residual.

Conditioning survives the reduction

A small S can still be ill-conditioned. Reducing its dimension does not bound $\|S^{-1}\|$. The manuscript's inverse estimate keeps that term explicit, along with the norms of $D^{-1}u$ and $v^T D^{-1}$.

> **Read this as:** The hidden coupling has not vanished. Its solvability and sensitivity are concentrated in a smaller operator that still has to be invertible and numerically controlled.

Plate VIII.11 / ACTUAL RANGE COORDINATES

Stored width is not intrinsic rank

Chapter VIII: Theorem 5.6, Section F.1.

Choose the isometry

$$Q_L = \begin{pmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad R_L = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad R_R = \begin{pmatrix} 3 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 0 & -1 \end{pmatrix}.$$

Set $L = Q_L R_L$ and $R = R_R$. The two equal columns of L do not create a third range direction. In contrast, R has rank three.

L: rank 2

$\frac{3}{5}$	0	$\frac{3}{5}$
$\frac{4}{5}$	0	$\frac{4}{5}$
0	1	0
0	0	0

R: rank 3

3	0	2
0	2	0
1	0	-1

G: 2×3

5	0	0
0	2	0

$$LR^T = Q_L G$$

The stored width is three; the left range has only two independent directions.

The actual core is

$$G = R_L R_R^T = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}, \qquad LR^T = Q_L G.$$

Its dimensions are the two actual input ranks, two and three. This is not inferred by merely discarding a column with a small norm. The range factor Q_L has orthonormal columns and reconstructs L exactly.

The preserved formal boundary

The small-core recompression construction in Appendix F constructs orthonormal bases of the actual column spaces. It then assumes exact ordered thin-SVD data for the core, as the manuscript instructs the reader to take an SVD. It does not assume an optimality or tail-error conclusion. Nor does it certify a floating-point QR or SVD implementation simply because those routines use similar notation.

This example has a diagonal rectangular core, so its singular directions are explicit. No iterative numerical factorization is needed to identify the two nonzero singular values, five and two.

> **Read this as:** Compactness is a statement about actual range spaces. The number of stored factor columns can exceed the dimensions that matter for recompression.

Plate VIII.12 / COMPRESSION, KERNEL WITNESSES, AND TAILS

Why a core optimum is global

Chapter VIII: Theorem 5.6, Section F.2, Section F.3, Section F.4.

The product from the previous page is

$$A_0 = LR^T = \begin{pmatrix} 3 & 0 & 0 \\ 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Keeping only its first singular direction gives a rank-one approximation with spectral and Frobenius error exactly two. Keeping none gives errors five and $\sqrt{29}$, while keeping both gives zero.

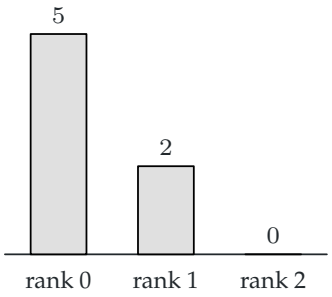
LR^T

3	0	0
4	0	0
0	2	0
0	0	0

rank-one output

3	0	0
4	0	0
0	0	0
0	0	0

spectral error



Frobenius errors: $\sqrt{29},\ 2,\ 0$. The rank-one choice loses only the second mode.

Why an outside competitor cannot do better

For isometric outer factors, compression $C(B) = Q_L^* B Q_R$ cannot increase rank or either matrix norm. Every ambient rank-k competitor therefore supplies an admissible core competitor. The lifted core residual has exactly the core residual norm. These two facts transfer a core optimum to the entire ambient matrix space, not just to competitors already lying in the supplied ranges.

For a d-dimensional diagonal core and $0 \le k < d$, the spectral lower bound has a concrete witness. A rank-at-most-k competitor has a nonzero kernel vector in the first $k+1$ coordinates. On that vector the diagonal matrix acts with norm at least σ_{k+1} times the vector norm. The competitor cannot cancel that direction.

The Frobenius proof uses an orthonormal basis of the competitor's kernel. Its row weights lie in $[0, 1]$ and sum to the kernel dimension. Weighting the decreasing squared singular values forces at least the discarded tail. The two norm arguments are distinct. One is not a graph of the other.

> **Read this as:** Optimality is not inferred from the appearance of the singular-value plot. It follows from compression and a lower bound valid against every rank-bounded competitor.

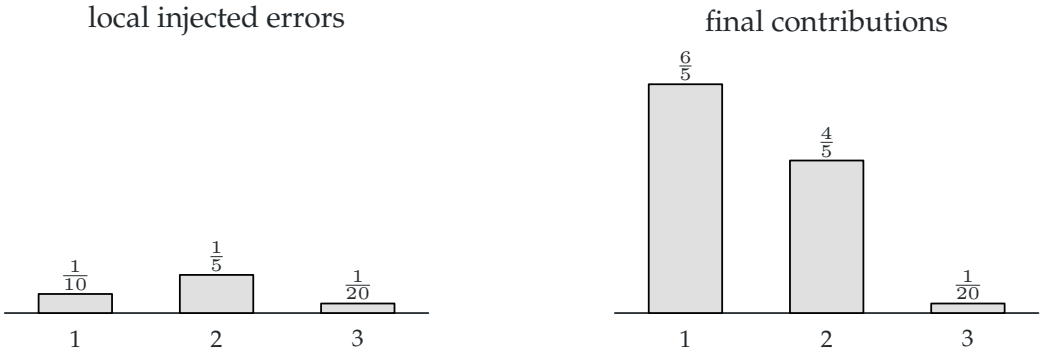
Plate VIII.13 / FACTOR PERTURBATION VERSUS RECOMPRESSION

Errors enter at different places

Chapter VIII: Theorem 5.8, Theorem 5.9.

Let the exact scalar factors be $(2, 3, 4)$ and start from $p_0 = 1$. If additive errors $(1/10, 1/5, 1/20)$ enter *after* each partial multiplication, then

$$\hat{p}_j = a_j \hat{p}_{j-1} + e_j, \qquad \hat{p}_3 - p_3 = 12e_1 + 4e_2 + e_3 = \frac{41}{20}.$$



$\hat{p}_j = a_j \hat{p}_{j-1} + e_j$, with $(a_1, a_2, a_3) = (2, 3, 4)$.

The same local numbers, a different model

If instead the factors themselves are perturbed, the computed product is $(4+e_3)(3+e_2)(2+e_1)$. Its error is $402/125$, not $41/20$. The factor-perturbation telescoping formula contains later factors that have already been perturbed, while the additive recompression formula contains the original later factors.

The visual comparison uses the additive model. The first error is multiplied by both remaining factors. The last error is not amplified at all. Summing only the local error magnitudes gives $7/20$, which is smaller than the actual final error and therefore not a valid bound here.

In a matrix chain

The corresponding formulas replace scalar magnitudes with operator norms and respect multiplication order. A bound can be conservative when contributions cancel, but it must not omit possible amplification. Singular-tail errors from a recompression step become useful only after they are inserted into the correct global error model.

> **Read this as:** An error tolerance needs an entry point. A bound for perturbing factors and a bound for compressing partial products are not interchangeable.

Plate VIII.14 / SUPPORT IS AN INCLUSION

Two routes can cancel exactly

Chapter VIII: Theorem 2.5, Theorem 6.1, Corollary 6.3.

Let K have transfers $0 \to 1$, $0 \to 2$, $1 \to 3$, and $2 \to 3$, with coefficients $1, 1, 1, -1$. There are two length-two routes from packet zero to packet three, but their contributions sum to zero.

K

0	0	0	0
1	0	0	0
1	0	0	0
0	1	-1	0

path counts at length two

0	0	0	0
0	0	0	0
0	0	0	0
2	0	0	0

$$1 + (-1) = 0$$

$$K^2 = 0$$

Structural support is an upper bound. It does not encode destructive cancellation.

$$(K^2)_{30} = K_{31}K_{10} + K_{32}K_{20} = 1 - 1 = 0, \qquad K^2 = 0.$$

The matrix of path counts sees the two possibilities and forgets the signs. It therefore predicts possible support but not actual support. Replacing the inclusion $\operatorname{supp}(AB) \subseteq \operatorname{supp}(A) \circ \operatorname{supp}(B)$ by equality would be wrong even in this four-dimensional model.

What an acyclic graph does guarantee

If the longest strict directed path has h edges, then no term of K^{h+1} can survive. Thus $K^{h+1} = 0$. This is an upper bound on the nilpotence index, not an assertion that $K^h \neq 0$. Here the graph has paths of length two, while coefficient cancellation already gives $K^2 = 0$.

A structurally permitted block may also vanish because a product of rectangular maps is zero before any sum is taken. A support graph cannot distinguish that mechanism from cancellation among separate routes. The actual block coefficients are therefore part of a proof-bearing operator representation.

> **Read this as:** Use graph reachability to exclude impossible transfers. Use the algebra to determine which permitted transfers actually survive.

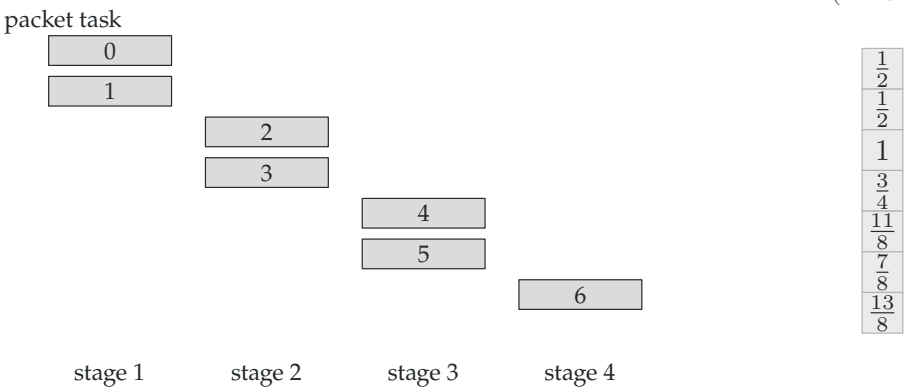
Plate VIII.15 / TOPOLOGICAL LEVELS WITH EXACT OUTPUTS

A solve has a dependency clock

Chapter VIII: Theorem 6.5, Corollary 6.6, Theorem 10.3.

For seven scalar packets, put two on the diagonal and -1 on each declared incoming edge. The edges are $0 \to 2$, $1 \to 2$, $1 \to 3$, $2 \to 4$, $3 \to 4$, $3 \to 5$, $4 \to 6$, and $5 \to 6$. Solve $(2I + K)x = \mathbf{1}$.

$$x = (2I + K)^{-1}\mathbf{1}$$



Read the stages as dependencies

Packets zero and one are ready first. Packets two and three use those values. Packets four and five then become available, followed by packet six. For instance,

$$x_4 = \frac{1 + x_2 + x_3}{2} = \frac{11}{8}, \qquad x_6 = \frac{1 + x_4 + x_5}{2} = \frac{13}{8}.$$

All entries in the displayed solution are checked by multiplying the original triangular matrix by the resulting vector.

When is four a minimum?

There are four vertices on a longest dependency path. Under the paper's unit-duration, unlimited-concurrency model, four stages are both necessary and attainable. If only one execution slot is available, seven tasks require seven slots of work. If task durations differ, the relevant critical path is weighted.

The theorem is not a universal GPU scheduling result. Transfers, synchronization, contention, and nonunit task costs change the model. The exact benefit here is that independent rows in one level need not wait for one another's mathematical results.

Read this as: A topological schedule separates dependencies from resources. Its optimal depth is conditional on the task-duration and concurrency model that defines a stage.

Plate VIII.16 / SCHUR FILL AND ITS CANCELLATION

Elimination creates new interactions

Chapter VIII: Theorem 6.7, Proposition 6.8.

Retain the first two coordinates of

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

The third coordinate is hidden. Its invertible block is $[2]$, so eliminating it produces

$$S = A_{RR} - A_{RH}A_{HH}^{-1}A_{HR} = \begin{pmatrix} 3/2 & -1/2 \\ -1/2 & 3/2 \end{pmatrix}.$$

A before elimination

2	0	1
0	2	1
1	1	2

$S = A_{RR} - A_{RH}A_{HH}^{-1}A_{HR}$

$\frac{3}{2}$	$-\frac{1}{2}$
$-\frac{1}{2}$	$\frac{3}{2}$

cancelling variant

$\frac{3}{2}$	0
0	$\frac{3}{2}$

A hidden route creates $-1/2$; adding $+1/2$ in the original block cancels it.

The hidden route has a sign

The retained coordinates had no direct coupling, yet both touch the hidden coordinate. The resulting correction couples them by $-1/2$. This is fill: a new active retained block produced by an eliminated route.

Now change both original retained off-diagonal entries to $+1/2$. The same hidden correction cancels them, leaving the diagonal matrix $\frac{3}{2}I$. Thus a structural route can create or remove visible coupling depending on the direct coefficient already present.

A stronger inverse statement

For an invertible finite packet matrix, inverse support lies in the directed transitive closure. A reachable-coordinate subspace is invariant under the matrix. Injectivity of its restriction implies surjectivity in finite dimension, so the inverse preserves it as well. This proof uses finite-dimensional invertibility, not a convergent geometric series.

The checker verifies both Schur complements and, in each invertible example, equality between S^{-1} and the retained block of A^{-1}.

> **Read this as:** Removing coordinates does not remove their influence. It changes the retained operator, with possible fill, cancellation, and new conditioning constraints.

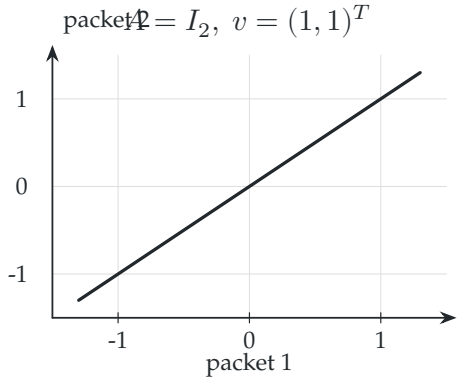
Plate VIII.17 / A STRICT KRYLOV INCLUSION

Invariant does not mean independently addressable

Chapter VIII: Proposition 7.2, Theorem 7.3.

Take $A = I_2$, split into the two coordinate packets, and start with $v = (1, 1)^T$. Every packet is invariant. Nevertheless every polynomial applies the *same* scalar to both coordinates:

$$p(A)v = p(1)(1, 1)^T, \qquad \dim \mathcal{K}_m(A, v) = 1 \quad (m \geq 1).$$



$$p(A)v = p(1)(1, 1)^T$$

$$\dim \mathcal{K}_m(A, v) = 1$$

Independent local spans have dimension 2.

The direct sum of the two local Krylov spans is all of $\mathbb{Q}^2$. It permits independent coefficients in the two packets. The global Krylov space does not. This is why the manuscript states an inclusion, not an unconditional equality.

What breaks the linkage?

With $A = \mathrm{diag}(1, 2)$, the vectors v and $Av = (1, 2)^T$ are independent. Two polynomial samples now distinguish the packet labels. Alternatively, allowing the packet projectors as extra address operators separates the coordinates even when $A = I_2$. Both changes enlarge the addressing power, but in different ways.

When invariance is only approximate

If $A = D + K$ with D packet diagonal, then

$$\|A^j - D^j\| \leq jM^{j-1}\|K\|, \qquad M = \max\{\|A\|, \|D\|\}.$$

This controls departure from the decoupled iterate. It is not a claim that the dimension or numerical conditioning of a Krylov basis varies continuously. Near dependence can remain difficult even when the operator error is small.

> **Read this as:** A single global polynomial can lock otherwise invariant packets together. Independence of subspaces is not the same as independence of the allowed actions.

Plate VIII.18 / BLOCK JACOBI AS A GEOMETRIC CHANGE

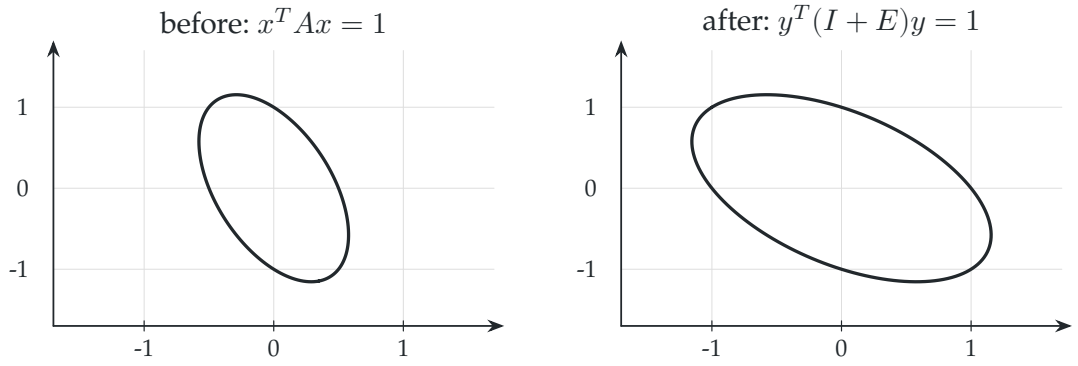
Measure coupling after normalization

Chapter VIII: Theorem 7.4, Corollary 7.6, Classical input 7.5.

Let

$$A = \begin{pmatrix} 4 & 1 \\ 1 & 1 \end{pmatrix}, \quad D = \operatorname{diag}(4, 1), \quad B = D^{-1/2} = \operatorname{diag}(1/2, 1).$$

Then $BAB = I + E = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$. Its exact eigenvalues are $1/2$ and $3/2$, giving condition number three.



Exact normalized eigenvalues: $1/2, 3/2$. Hence $\kappa_2 = 3$.

The unit circles of the energy norms

The left ellipse is $x^T Ax = 1$. Writing $x = By$ gives the right ellipse $y^T(BAB)y = 1$. The drawing samples these explicitly defined curves. The norm and eigenvalue equalities are checked symbolically.

Here $\|E\|_2 = 1/2$. The manuscript's general inequality

$$\kappa_2(D^{-1/2}AD^{-1/2}) \le \frac{1+\eta}{1-\eta}, \qquad \|E\|_2 \le \eta < 1,$$

is attained. The raw off-diagonal entry is one. Its meaningful size is one-half only after normalization by the diagonal energy scales.

Keep the convergence input separate

The condition bound can be inserted into a conjugate-gradient convergence estimate under that method's positive-definite assumptions. The CG estimate is one of the separately tracked classical inputs. The algebraic normalization theorem does not certify every stopping rule, floating-point iteration, or arbitrary preconditioner.

> **Read this as:** A small raw coupling is not scale-invariant. Normalize against the diagonal blocks before deciding whether the coupling is weak enough for a condition bound.

Plate VIII.19 / DISCOVERY AND PERTURBATION CERTIFICATES

Twelve small blocks, one coherent error

Chapter VIII: Theorem 8.1, Proposition 9.2.

Split $\mathbb{Q}^4$ into scalar packets and let

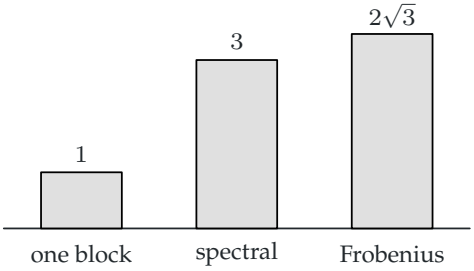
$$A = I + E, \qquad E = \frac{1}{100}(\mathbf{1}\mathbf{1}^T - I).$$

Dropping every off-diagonal block leaves I. Each discarded entry is only $1/100$, but the total error is not bounded by that single-entry size.

$100E$: twelve omitted entries

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

error norms, in units of $1/100$



$\|E\|_F^2 = 12/10^4$, while $E\mathbf{1} = (3/100)\mathbf{1}$.

Two norms answer different questions

There are twelve omitted entries, so $\|E\|_F^2 = 12/10^4$. The all-one vector is an eigenvector with eigenvalue $3/100$, while the three orthogonal directions have eigenvalue $-1/100$. Hence $\|E\|_2 = 3/100$.

For orthogonal coordinate blocks, the squared Frobenius norms add exactly. Operator norms instead require a suitable bound or further spectral information. The maximum omitted block norm is generally insufficient.

Input sensitivity is additional information

The packet perturbation estimate is

$$\|(\widehat{A} - A)x\| \le \sum_e \|E_e\| \, \|P_{\alpha(e)}x\|.$$

It can be much tighter for an input that scarcely occupies an affected source packet. To obtain a uniform operator bound, the input dependence must be bounded uniformly over unit vectors. A certificate for one input is not automatically a certificate for every right-hand side.

These are a posteriori checks for supplied projectors. They do not discover the best packet system or prove that a thresholding heuristic is globally optimal.

> **Read this as:** A proposed compression needs a residual certificate in the requested norm and input scope. Small individual blocks can align into a larger collective error.

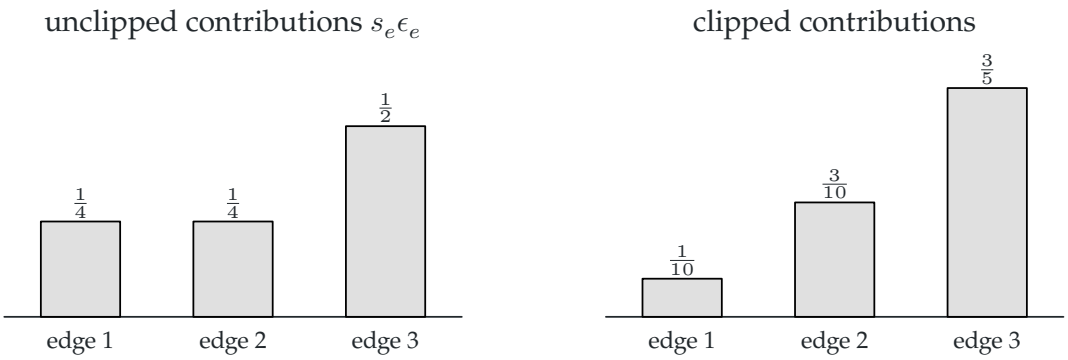
Plate VIII.20 / CONTINUOUS AND CLIPPED PRECISION

Spend error where it is cheapest

Chapter VIII: Theorem 8.4, Theorem 8.5.

Take sensitivities $s = (1, 2, 4)$, cost coefficients $c = (1, 1, 2)$, and budget $\tau = 1$. Minimizing $\sum_e c_e \log(1/\epsilon_e)$ subject to $\sum_e s_e \epsilon_e = 1$ gives

$$\epsilon_e = \frac{c_e}{4s_e}, \qquad \epsilon = (1/4, 1/8, 1/8).$$



Both allocations spend exactly one unit of error budget.

The budget is weighted

The three contributions to the error allowance are $(1/4, 1/4, 1/2)$. Equal cost coefficients give equal weighted contributions, not equal tolerances. The third edge has twice the cost coefficient of the second, which offsets its doubled sensitivity in this example.

Now impose upper tolerances $m = (1/10, 1/2, 1/2)$. The first edge must stop at its tighter cap. With multiplier $\lambda = 10/3$, the clipped solution is

$$\epsilon_e = \min\{m_e, c_e/(\lambda s_e)\}, \qquad \epsilon = (1/10, 3/20, 3/20).$$

Its weighted contributions are $(1/10, 3/10, 3/5)$ and still sum to one. The checker verifies the active cap, the multiplier equations, and the budget exactly.

From tolerance to a real data type

The continuous objective is a model. Actual kernels offer discrete precisions and may use different accumulation types. A tolerance below one does not identify a particular hardware format. The finite option program on the next page is the correct next step once admissible costs and error bounds have been supplied.

> **Read this as:** Precision allocation is an optimization over certified sensitivities and costs. It does not infer error bounds from a data-type name alone.

Plate VIII.21 / THREE EDGES, TWENTY-SEVEN ASSIGNMENTS

An exact budget is not a greedy choice

Chapter VIII: Theorem 8.7, Remark 8.8.

Each edge has three allowed $(\text{integer weight}, \text{cost})$ options:

$$1 : (0,7), (2,3), (4,1),$$
$$2 : (0,6), (1,4), (3,1),$$
$$3 : (0,9), (2,5), (5,1).$$

There are exactly $3^3 = 27$ full assignments. The table stores the minimum cost at each *exact* weight. Columns zero through eight are displayed.

	0	1	2	3	4	5	6	7	8
edges 0	0	∞	∞	∞	∞	∞	∞	∞	∞
edges 1	7	∞	3	∞	1	∞	∞	∞	∞
edges 2	13	11	9	7	7	4	∞	2	∞
edges 3	22	20	18	16	14	12	12	9	8

Columns are exact weights; the budget optimum is the minimum across allowed columns.

Read the final row correctly

At budget four, minimize across columns zero through four. The optimum costs 14: choose weights $(2, 0, 2)$ with costs $(3, 6, 5)$. The zero-weight choice on edge two is its first, more expensive option. A locally cheap choice can consume too much of the shared budget.

The checker compares the recurrence with complete enumeration for every budget from zero through twelve. Infeasible states remain $+\infty$. They are not zero-cost states.

Quantization must be conservative

Suppose real errors are $3/50$ and $2/25$, and one integer unit represents $1/10$. Rounding down gives two zero weights, even though the true sum is $7/50 > 1/10$. Rounding up gives a certified upper envelope instead. The budget method is exact for its integers. It cannot repair unsafe integers supplied by a preprocessing step.

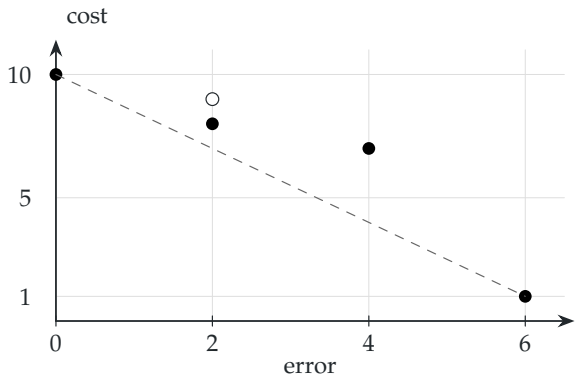
> **Read this as:** The table optimizes the declared option set exactly. Accuracy depends separately on the validity and conservativeness of the option weights.

Plate VIII.22 / FIVE PRUNINGS, ONE NONCONVEX FRONTIER

A penalty can miss the budget optimum

Chapter VIII: Theorem 9.4, Corollary 9.7, Theorem 9.8.

A root r has children a, b, each with two terminal children. Keeping r costs one with error six. Splitting it adds overhead one. Keeping either a or b costs three with error two. Splitting a costs four with zero error, and splitting b costs five with zero error.



Budget $B = 2$: cost 8

Penalty optimum: root or full split

$8 + 2\lambda \le 1 + 6\lambda$
and $8 + 2\lambda \le 10$
cannot both hold.

The five complete prunings have $(\text{cost}, \text{error})$ pairs

$$(1, 6),\ (7, 4),\ (9, 2),\ (8, 2),\ (10, 0).$$

At budget two the unique cheapest pruning is $\{a_0, a_1, b\}$, costing eight. It is Pareto optimal, but never minimizes cost $+\lambda$ err for any $\lambda \ge 0$.

See the incompatible inequalities

To beat the root, that pruning would require $8 + 2\lambda \le 1 + 6\lambda$, hence $\lambda \ge 7/4$. To beat the full split, it would require $8 + 2\lambda \le 10$, hence $\lambda \le 1$. No multiplier meets both conditions.

The exact budget recurrence retains the point because it enumerates admissible weight allocations. A sweep of Lagrangian parameters sees only supported frontier points. The two compiler formulations are therefore genuinely different, even on this tiny tree.

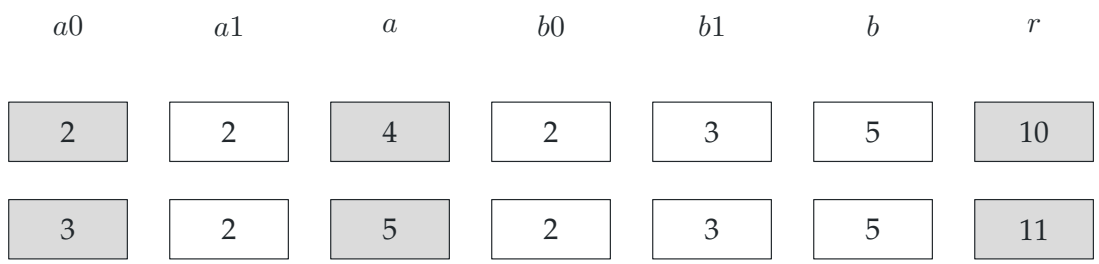
> **Read this as:** A penalty parameter is not a universal substitute for an explicit budget. Discrete, nonconvex frontiers can contain useful optima that no penalty selects.

Plate VIII.23 / LOCAL DEPENDENCIES AND STRICT MARGINS

Recompile the changed part, not everything

Chapter VIII: Theorem 9.11, Theorem 9.12.

Use the same tree with $\lambda = 2$. Increase only the leaf cost at a_0 from two to three. The exact subtree values change at a_0, its parent a, and the root r. Every value in the b subtree remains untouched.



Top: old values. Bottom: new values. Only a0, a, and r change.

Why the update is local

A dynamic-programming value depends only on the local keep/split data and its children's values. A changed leaf can therefore affect its ancestors but not a disjoint sibling subtree. Recomputing the dependency closure yields the same global optimum as a fresh traversal. In this example, the root value increases from ten to eleven.

Strict gaps protect decisions

Before the update, the root keep score is 13 and its split score is 10, giving gap three. If each score changes by less than $3/2$, their strict ordering is retained. Allowing exactly $3/2$ in both directions can produce a tie, so the strict inequality in the stability statement matters.

The bound applies to the completed scores after child subproblems have been evaluated. Bounds on raw leaf perturbations must first be propagated to those scores. Shared caches, overlapping subproblems, or global rank constraints can introduce dependencies outside a tree and invalidate this simple locality argument.

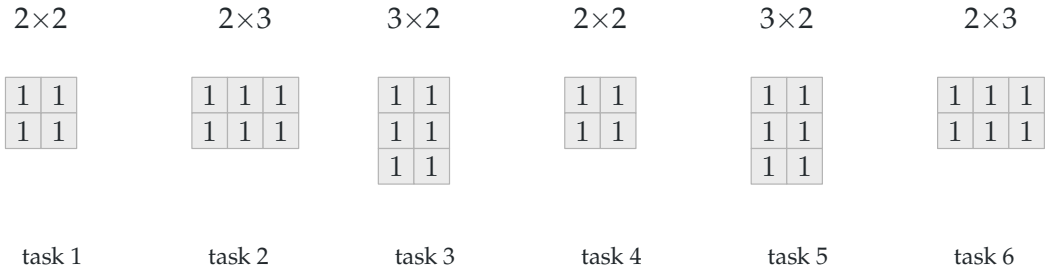
> **Read this as:** Local recompilation is exact because the model has local dependencies. The locality is a mathematical hypothesis, not an automatic property of every compiler optimization.

Plate VIII.24 / A LAUNCH MODEL IS NOT A BANDWIDTH MODEL

Batch shapes, account for traffic

Chapter VIII: Proposition 3.6, Theorem 3.8, Proposition 10.2.

Six independent blocks have the displayed dimensions. When right-hand-side count, scalar type, accumulation type, and kernel contract also agree within each class, they can be grouped into three batches. That reduces the modeled launch count from six to three without changing their arithmetic.



Three shape classes, six independent tasks.

Grouping does not remove arithmetic, data transfer, or shared-output dependencies.

Independence must be real

The batching statement assumes compatible backend support and independent block tasks. If several blocks accumulate into one target, the reduction or accumulation policy is additional work. Equal matrix dimensions alone do not remove that dependency. Different accumulation types also define different shape classes under the paper's convention.

Count the data separately

For three diagonal 2×2 blocks, eight-byte scalars, and q right-hand sides, the stated one-pass model moves

$$B_{\text{stream}} = 8 \cdot 3(4 + 2q + 4q) = 96 + 144q \text{ bytes.}$$

It gives 240 bytes at $q = 1$ and 672 bytes at $q = 4$. It deliberately assumes no reuse across edges. A cache-aware implementation may have another traffic model, but it must state one.

Given operation capacity P and bandwidth W, $T \geq \max(F/P, B/W)$ is a lower bound, not an achieved runtime. Launch overhead, imperfect overlap, and utilization can make the actual time much larger.

Read this as: Grouping can remove launches while leaving the same products and data movement. Arithmetic, traffic, and scheduling are related constraints, not one interchangeable cost.

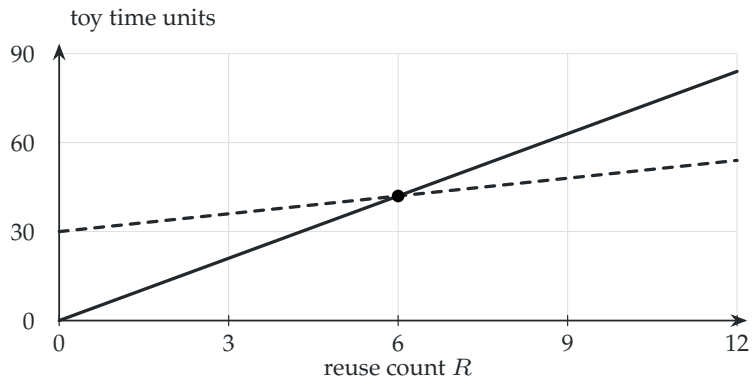
Plate VIII.25 / SETUP AND REUSE IN ONE LINEAR MODEL

The first win comes after the tie

Chapter VIII: Theorem 10.4, Corollary 10.5, Corollary 10.6.

Suppose the baseline costs seven units per use. A candidate costs 30 units to set up and two per use. These are synthetic constants chosen to expose the exact threshold:

$$T_b(R) = 7R, \qquad T_c(R) = 30 + 2R.$$



solid: $7R$

dashed: $30 + 2R$

tie at 6
first win at 7

At $R = 6$, both cost 42. The candidate becomes strictly faster at $R = 7$, where it costs 44 instead of 49. The minimum profitable integer reuse count is

$$\left\lfloor \frac{30}{7-2} \right\rfloor + 1 = 7.$$

Replacing the floor-plus-one rule with a ceiling would be wrong when the ratio is already an integer and strict improvement is required.

Which costs belong to setup?

Discovery, compilation, host/device transfer, graph capture, and factor preparation can have different reuse lifetimes. If a matrix changes but its shape does not, a captured schedule might remain reusable while its numerical factors do not. A single setup number is meaningful only after those lifetimes are fixed.

Materialization has the opposite tradeoff: pay once for an explicit representation, then use a potentially faster dense downstream operation. It pays off only when the explicit per-use cost is lower and the reuse count exceeds the corresponding conversion threshold.

> **Read this as:** A faster steady-state kernel can lose end-to-end. State the reuse regime and count the tie separately from the first strict win.

Plate VIII.26 / A BENCHMARK ROW IS A CONTRACT

Same output before comparing times

Chapter VIII: Proposition 2.9, Section 11, Section 12.

Consider four synthetic timing rows for a product. Row B returns factors, while row A returns an explicit matrix. A ratio $12/2 = 6$ does not compare two implementations of the same returned object. Adding the decoding cost changes the candidate to row C.

row	output	scope	time
A	explicit matrix	materialization included	12
B	low-rank factors	factor product only	2
C	explicit matrix	factor product + decode	14
D	explicit matrix	different input precision	1

Synthetic times: A versus C is contract-aligned. A versus B is not a sixfold speedup.

What a fair match retains

The input values, dimensions, layout, source precision, correctness reference, returned output, and timing scope must agree. Row D changes the input precision. That can be a useful accuracy tradeoff, but it is not the same case merely because the output has the same shape.

If a downstream task consumes factors directly, row B has a valid compressed-output contract. It should be compared with another method that supplies that downstream result, not forced to materialize an unused array. Fairness does not mean every algorithm must use the same internal representation.

How the original measurements are treated here

The accompanying benchmark study records the CPU/GPU corpus. Regenerating its figures redraws the saved observations. It does not rerun the device, establish current hardware performance or turn logically avoided work into physically executed floating-point operations. These figures add no new timing measurements.

Exact representation gains, device gains, kernel gains, and amortization gains can be isolated only through appropriately matched rows. A total ratio that mixes them is still an observation, but not a decomposition theorem.

> **Read this as:** A timing number without its case and scope is incomplete data. Preserve the mathematical output contract before claiming a speedup.

Plate VIII.27 / FACTORS, APPLICATION, AND AN INDEPENDENT REFERENCE

Close the loop on the running example

Chapter VIII: Theorem 2.3, Theorem 3.4, Theorem 5.1.

Return to the two operators from Plate VIII.9. The exact product representation has diagonal part $D_A D_B$ and two-column correction factors U_C, V_C. Apply it to the original $x = (1, -1, 2, 0, -2, 3)^T$.

$$V_C^T x = (2, 13)^T, \qquad (AB)x = D_A D_B x + U_C(V_C^T x).$$

$V_C^T x$	$D_A D_B x$	$U_C(V_C^T x)$	$(AB)x$
2	1	25	26
13	−5	21	16
	18	25	43
	6	21	27
	−12	25	13
	44	21	65

Two intermediate scalars and six output coordinates, with an exact dense reference.

A complete reader calculation

The local contribution is $(1, -5, 18, 6, -12, 44)^T$. The correction is $(25, 21, 25, 21, 25, 21)^T$. Their sum is

$$(26, 16, 43, 27, 13, 65)^T.$$

The checker forms the dense matrices A and B independently, multiplies them, applies the product to x, and compares all six components. The dense array is a validation reference, not a requirement of the factored application route.

Choose what happens next

For another input, reuse the factors. For a required explicit matrix, pay materialization. For a long product chain, track rank growth and decide whether recompression is useful. If approximation is introduced, transport its error through the later operators rather than reporting only a local singular tail.

This small example does not predict a speedup: at dimension six, implementation overhead can dominate. Its purpose is to make the semantics auditable. The same identity can then be used inside a larger, separately benchmarked workload.

> **Read this as:** The full route is: represent the exact operator, retain every interaction, return the requested object, and validate against an independent reference. Performance is a further experiment.

Plate VIII.28 / FORMAL, EXACT, NUMERICAL, AND EMPIRICAL EVIDENCE

What the evidence certifies

Chapter VIII: Sections 13–14; accompanying benchmark and verification records.

The formal companion covers 56 finite manuscript-proved items through 392 supporting theorem declarations in 40 modules. Nineteen declarations are reused foundation results. The 66 construction audits are counted separately. Compiler records and source identities are provided with the formal companion.

Evidence	Its precise use here
Saved Lean records	Preserve the exact hypotheses and declared finite formalization boundary.
Exact model checks	Reproduce the guide's displayed matrices, identities, integer models, and finite optima. They are not extracted Lean proofs.
Vector drawings	Display explicit arrays, schedules, and sampled geometric formulas. Pixel accuracy is not algebraic certification.
Separate software checks	The verification report records actual runs, tolerances, failures, and unavailable gates. These are separate from this guide's exact arithmetic.
Recorded GPU corpus	Remains a record of the supplied original cases and environment. It is not a new measurement.

The recompression distinction

The recompression theorem derives best-rank and singular-tail conclusions from ordered thin-SVD equations rather than an assumed optimality certificate. It does not establish general SVD existence, certify a floating-point QR/SVD algorithm or identify every result with a canonical singular-values API. These are separate statements.

The four classical entries remain visible

The manuscript separately tracks Eckart–Young–Mirsky, the conjugate-gradient energy estimate, the floating-point product model, and simultaneous spectral decomposition. The strengthened ordered-SVD proof does not silently promote all associated existence and numerical-algorithm claims.

Limits of exact compilation

The optimizer needs a finite candidate tree with additive data. Nonlocal cache effects, overlapping decompositions, global rank coupling, and unrestricted discovery can destroy its optimal substructure. No new generic dense-matrix exponent or universal accelerator claim follows from this guide.

> **Read this as:** This guide explains the chapter's finite concepts through reproducible examples and preserved source evidence. It changes exposition, not the formal theorem contract.

Plate VIII.29 / THE INNER PRODUCT IS PART OF THE CONTRACT

A complex range needs an adjoint

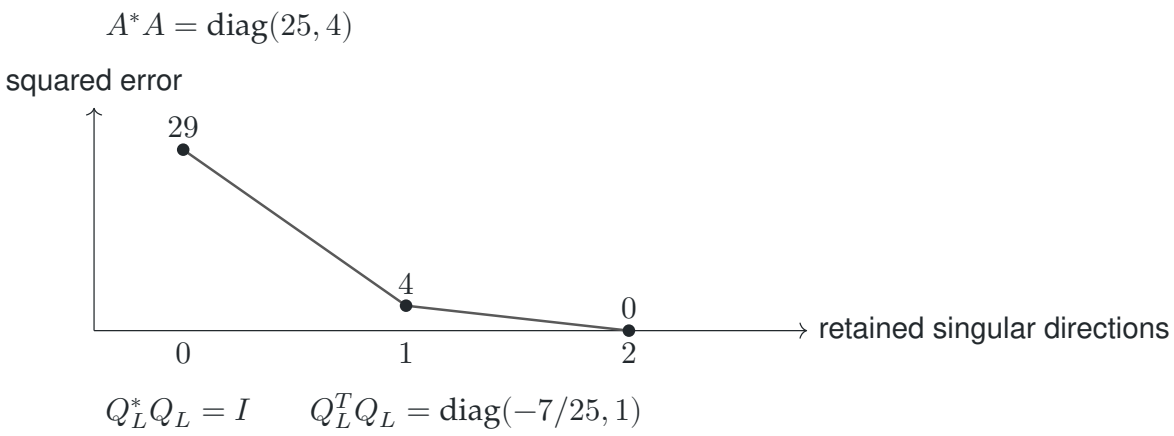
Chapter VIII: Theorem 5.6, Section F.1, Section F.2.

The earlier compact example is real. Over $\mathbb{C}$, an isometry means $Q^*Q = I$, with conjugation. Take the exact Gaussian-rational factors

$$Q_L = \begin{pmatrix} 3/5 & 0 \\ 4\mathrm{i}/5 & 0 \\ 0 & 1 \end{pmatrix}, \qquad Q_R = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \qquad G = \operatorname{diag}(5, 2).$$

Their product is a genuinely complex rectangular operator:

$$A = Q_L G Q_R^* = \begin{pmatrix} 3 & 0 \\ 4\mathrm{i} & 0 \\ 0 & -2\mathrm{i} \end{pmatrix}, \qquad A^*A = \operatorname{diag}(25, 4).$$



The small calculation exposes a wrong convention

The first column of Q_L has Hermitian squared length $9/25 + 16/25 = 1$. Its bilinear square without conjugation is $9/25 - 16/25 = -7/25$. Thus $Q_L^TQ_L$ is not the identity. The portable negative control replaces the adjoint by transpose and rejects the resulting isometry claim.

Keeping the first diagonal entry of G gives A_1 with its second column zero. The residual has squared Frobenius norm four and Gram matrix $\operatorname{diag}(0, 4)$, hence spectral norm two. Both errors agree with the discarded core singular value. Every displayed arithmetic operation occurs in $\mathbb{Q}(\mathrm{i})$.

What the finite check establishes

It checks both isometries, the reconstructed ambient matrix, its Gram matrix, and the explicit rank-one residual. Global best-rank optimality still comes from the manuscript's compression and lower-bound proof. This two-column calculation is a worked instance of that argument. No existence theorem or numerical SVD routine is executed.

> **Read this as:** Conjugation preserves the complex Euclidean norm. A transpose is correct in the vectorization identity on Plate VIII.7, but an adjoint is required for an isometric range lift.

Plate VIII.30 / A FINITE INVERSE WITH A CANCELLED ROUTE

Inverse reachability is still an inclusion

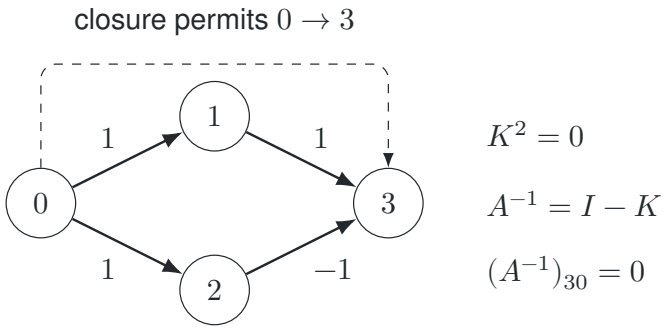
Chapter VIII: Theorem 6.7, Theorem 6.1, Corollary 6.3.

Reuse the signed diamond from Plate VIII.14, now as a complete operator:

$$A = I + K = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix}.$$

The two routes from coordinate zero to coordinate three cancel, so $K^2 = 0$. Consequently

$$A^{-1} = I - K = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 1 \end{pmatrix}.$$



What the closure permits

The directed transitive closure includes $0 \to 3$, because it records existence of a route rather than the signed sum of its coefficients. Nevertheless $(A^{-1})_{30} = 0$. Reachability is therefore an upper envelope for inverse support, just as it is for product support.

The portable checker computes the inverse independently by rational row elimination, compares it with $I - K$, constructs Boolean reachability, and checks every inverse entry outside that closure. It separately rejects the assertion that every reachable transfer is nonzero.

Why the general proof is broader

This example happens to have a terminating Neumann polynomial. The general finite-dimensional theorem does not require nilpotence or a small norm. For a fixed source coordinate, its reachable-coordinate subspace is invariant under A. If A is invertible, the restricted map is injective. Equal finite domain and codomain dimensions make it surjective, so the inverse preserves the same subspace.

The finite-dimensional and invertibility hypotheses matter. In an infinite-dimensional space, an injective restriction can fail to be onto. A singular matrix does not have an inverse whose support can be tested at all.

> **Read this as:** The inverse can never transmit outside the reachable subspace under the theorem's hypotheses. Inside that subspace, coefficient cancellation can still remove a structurally permitted interaction.

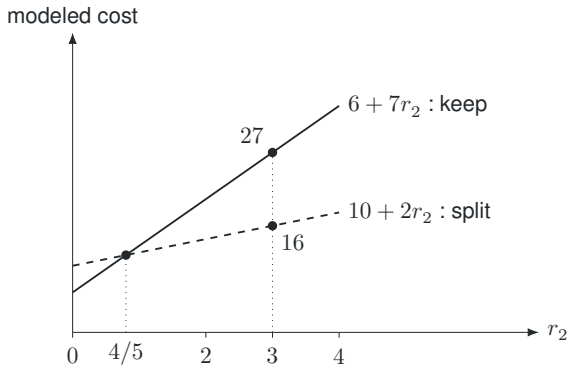
Plate VIII.31 / SHARED COMPILATION HAS ITS OWN OBJECTIVE

One tree can serve a weighted family

Chapter VIII: Corollary 9.5, Theorem 9.4.

Use a tree with one root and two terminal children. There are exactly two admissible prunings: keep the root, or split into both children. All declared errors and the split overhead are zero. Their additive total costs are:

Cost component	Keep	Split
Setup paid once	4	2
Operator 1, per use	2	8
Operator 2, per use	7	2



With reuse weights $(r_1, r_2) = (1, 3)$, the aggregated objectives are

$$J_{\text{keep}} = 4 + 2 + 3 \cdot 7 = 27, \qquad J_{\text{split}} = 2 + 8 + 3 \cdot 2 = 16.$$

The exact shared optimum is split. Operator 1 alone prefers keep on its per-use costs, while operator 2 prefers split. Choosing independently and trying to combine those decisions would not solve the declared one-pruning problem.

A hand-computable decision threshold

Fix $r_1 = 1$ and vary $r_2 = r \geq 0$. Keeping costs $6 + 7r$ and splitting costs $10 + 2r$. Splitting is strictly cheaper when $r > 4/5$. Its first integer reuse win is $r = 1$. The checker evaluates both admissible choices for integers zero through four and checks the displayed totals at three.

Where the common-tree proof applies

The family corollary aggregates local keep costs and errors, then uses the same exact subtree recurrence. In this example, each row denotes the sum of the selected local leaf terms. It does not conceal a nonlocal discount. A setup shared by distant unrelated nodes, overlapping candidates, or a global resource constraint would need an enlarged state or a different model.

Read this as: Reuse weights change the objective before compilation. Shared optimality is exact for the supplied common tree and additive family costs, rather than for an unspecified set of future workloads.

Plate VIII.32 / QUANTIZATION AND THE SIZE OF THE BUDGET TABLE

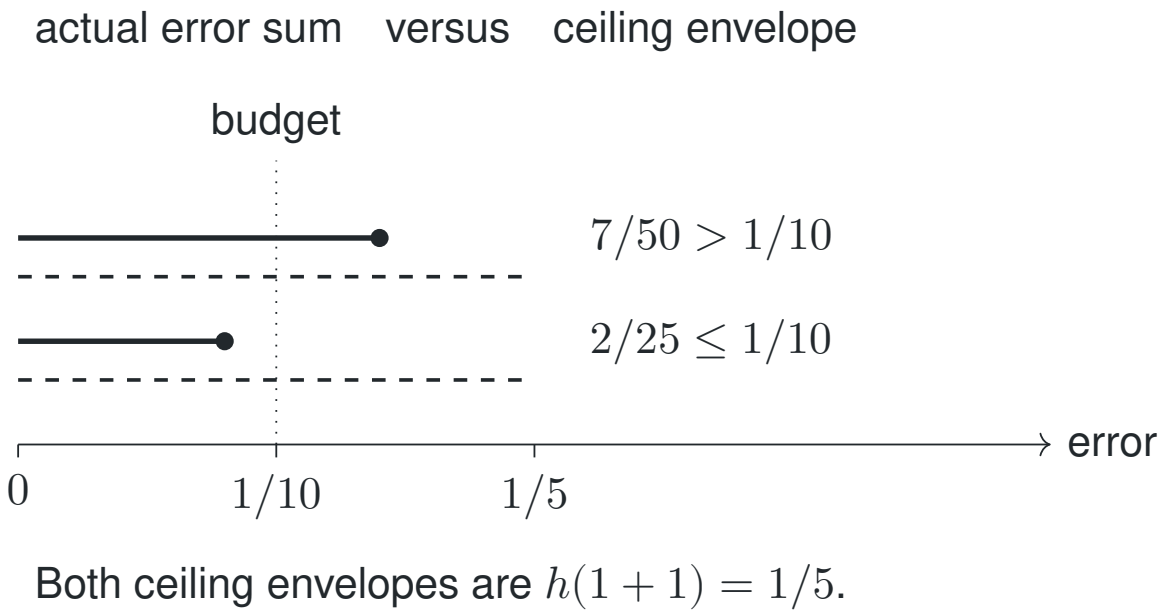
Safe integers can still be conservative

Chapter VIII: Theorem 8.7, Theorem 9.8, Corollary 9.9.

Let one integer unit represent $h = 1/10$ of certified error. For an individual nonnegative error e, choosing

$$w = \lceil e/h \rceil \qquad \text{ensures} \qquad e \le hw.$$

Thus integer feasibility $\sum w \le B$ implies the real bound $\sum e \le hB$. This implication is checked before applying the exact option or tree recurrence.



A false certificate and a conservative rejection

For errors $(3/50, 2/25)$, rounding down assigns weights $(0, 0)$ although their real sum is $7/50 > 1/10$. Rounding up gives $(1, 1)$ and correctly excludes the assignment from budget one.

Now change the errors to $(3/50, 1/50)$. Their sum is $2/25 \le 1/10$, but separate ceilings still give $(1, 1)$. The integer model rejects an assignment that is feasible for the real-valued problem. Conservative rounding protects soundness. It need not preserve every optimum of the unquantized problem.

Why the complexity is pseudo polynomial

For a declared integer budget B, a table has $B + 1$ possible exact-weight states. A direct merge checks at most $(B + 1)^2$ state pairs. At $B = 255$, this means 256 states and 65,536 candidate pairs at a merge, even though the budget fits in eight binary digits.

With a bounded number of children, the paper's direct tree implementation therefore has quadratic budget dependence at each node. Polynomial dependence on the numerical value of B does not mean polynomial dependence on the binary input length. These are counts of declared states and pairs, not measured runtimes or claims that every state is reachable.

Read this as: The dynamic program is exact for its integer data. Conservative conversion and the numerical size of the budget are separate obligations that determine its accuracy and practical scale.

CHAPTER IX

Euler Congruences

Completion, factorial strengthening, and sharp dyadic geometry

The arithmetic sequence begins with congruence identities and the hypotheses needed to extend them.

Euler congruences are developed through completion, factorial strengthening and dyadic geometry. The accompanying guides connect exact statements, bounded examples and formal coverage.

IN THIS CHAPTER

Mathematical text 931

Visual studies 1145

Formal companion 2220

Sources and references 2687

Scope

The arithmetic, dyadic and lifting results have distinct hypotheses. General Euler-tail simplicity and the broader native and polyphase extensions remain open.

The next chapter organizes residual seeds and conditional system completion from supplied arithmetic data.

Abstract

We prove Nishibuchi's generalized Euler congruence conjecture in its full stated parameter range and develop sharper arithmetic consequences. For every prime-power packet width, a charged Hurwitz differential operator transfers integral difference estimates through the reciprocal generating series. In the dyadic case this gives mixed finite-difference bounds at every row, without a starting threshold. For the four addresses of width four, a divided-power argument adds the factorial gain $v_2(k!)$, determines every repeated-step valuation in addresses zero and one, and gives a stronger address-two estimate with exact values at all power-of-two orders. The corresponding first differences yield least periods and bijective isometric continuations onto explicit dyadic residue classes. We also determine exact Taylor radii for three addresses. The manuscript retains the complete earlier arithmetic, lifting, deformation, and analytic-regularity development, with detailed mathematical appendices. A separate appendix states the precise remaining simplicity problem for ordinary Euler tails. Formal coverage is explicit: the external conjecture and several four-address results have inherited Lean verification. The new factorial, mixed-step, all-rank, and radius arguments are written proofs with recorded audits and exact computational checks, and await dedicated Lean formalization.

Principal results

The original external problem is Nishibuchi's Conjecture 3.9 for generalized Euler numbers. Its full parameter range is resolved. If p is prime, m is positive with $m \mid p-1$ or $m = 2$, $0 \le j < mp$, and $q = \operatorname{lcm}(2, p-1)$, then the original Euler coefficients are p-adically integral and

$$E^{(mp,j)}_{mpn+qp^r} \equiv E^{(mp,j)}_{mpn} \pmod{p^r} \qquad (r \ge 1,\ n \ge r).$$

The chapter states the generating function and the conversion between ordinary degree and packet row before using this formula. The normalized coefficient is $C_{N,j}(n)$ and the original coefficient is $E^{(N,j)}_{Nn} = j!C_{N,j}(n)$. The constant coefficient of the original series is $j!$.

The strongest common-step conclusion in width four is compact. For $n \ge 0$ and $k, H \ge 1$,

$$\sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} E^{(4,j)}_{4(n+iH)} \equiv 0 \pmod{2^{\,v_2(k!)+kv_2(H)+A_j(k)}},$$

where

$$A_0(k) = A_1(k) = k, \qquad A_2(k) = \max\{k+4,\ 2k + \lfloor k/2 \rfloor + 1\}, \qquad A_3(k) = k+2.$$

There is no starting-row threshold. The exponent is exact in addresses zero and one for every order and positive step. It is exact in address two at every power-of-two order. The proof chapters also give mixed-step forms, the cancellation criterion in addresses zero and one, and exact first-displacement laws.

Result	Scope	Evidence
Nishibuchi Conjecture 3.9	Every parameter in the original statement, with explicit $n \geq r$	Inherited Lean proof
Width-four all-row common-step bound, first differences, least periods and isometry	Every address, row and positive displacement	Inherited Lean proof
General prime-power inverse transfer and mixed steps	Every prime-power width, with signed steps for odd primes	Written proof and audit
Factorial gains and exact higher-order families	Four width-four addresses, with stated sharpness scopes	Written proof and audit
Taylor radii and broader dyadic first differences	Exact radii in addresses $0, 1, 2$ and stated all-rank cases	Written proof and audit
Ordinary Euler tail simplicity	Partial families and an explicit remaining discriminant problem	Open in full generality

The arithmetic development leads to polynomial lifting, deformation and analytic regularity. The appendices give the detailed continuation arguments and the recovered foundational treatment. Where an earlier formulation is retained for comparison, its hypotheses and proof status remain explicit. The formal companion supplies the exact source map and distinguishes source review from compiler execution. The visual studies supply finite examples and diagrams derived from the theorems.

Recorded proof reviews are internal reviews. Formal proof coverage and the scope of each mathematical result are stated separately.

1 Scope and conventions

The central objects are sequences and polynomial equations equipped with arithmetic coordinates. Residue projectors select addresses in a sequence, finite differences measure compatibility between nearby addresses, and valuation bounds control the passage from finite precision to a p-adic limit. These constructions provide a common language for the Euler packet results and the later lifting results. Each theorem nevertheless retains its own coefficient ring, topology and nonsingularity hypotheses.

The arithmetic framework is developed together with its completed extensions. The stated external Nishibuchi conjecture is proved in its full specified parameter range. The dyadic refinements and polynomial deformation theorems give stronger conclusions under the hypotheses stated with each result. The native conjectures retain their original identifiers, 13.2 and 13.3, when discussed as research questions, although the section numbering of this consolidated manuscript is new.

Throughout, $\mathbb{N} = \{0, 1, 2, \ldots\}$. When p is prime, $\mathbb{Z}_p$ denotes the p-adic integers and $\mathbb{Q}_p$ their fraction field. We normalize $v_p(p) = 1$. For a rational or p-adic quantity, a displayed lower valuation bound is understood with the mathematical convention $v_p(0) = +\infty$. The corresponding formal predicates are zero-safe: the quantity is zero or its valuation meets the stated bound. An exact finite valuation assertion requires a nonzero quantity, and this is stated where used. Congruence in $\mathbb{Z}_p$ means ideal divisibility. Thus

$$x \equiv y \pmod{p^r} \quad \Longleftrightarrow \quad x - y \in p^r \mathbb{Z}_p.$$

For vectors, ideal membership and congruence are coordinatewise.

Write $\Delta_u f(n) = f(n+u) - f(n)$ and use Δ_u^k for its k-fold iterate. Euler coefficients indexed by ordinary degree and by packet row are distinguished explicitly. A packet displacement by u rows is a degree displacement by Nu in a packet of width N.

The main text presents theorem statements and their arguments. Accompanying source maps identify the Lean declarations and archived files that support them. The formal companion identifies the scope of each recorded build. The 91-check polynomial record covers the polynomial development. Separate records support the arithmetic results. The inherited formal development is retained without adding Lean declarations. The strengthening chapters add written proofs and clearly mark their formalization boundary. No historical-priority claim is made. The analytic inverse-function results used in Section 11 come from Mathlib.

2 Arithmetic packets and coefficient transport

We work with sequences, formal power series, and finite recurrence states. Residue selectors isolate coefficients in a chosen congruence class, while triangular transport expresses the same object in new coordinates. A valuation filtration determines how many coordinates are needed at a given precision. Together, these operations provide a common framework for the Euler congruences and the finite lifting constructions developed later.

The original arithmetic seed notation

$$\mathcal{S} = (A, a_0, \mathcal{O}, \mathcal{R}, \{P_{N,j}\}, w, \tau_p)$$

records a coefficient ring or module, a seed, allowed operators, relations, residue selectors, a depth weight, and a topology. This notation describes the ingredients of the construction. A category of seed presentations and a notion of equivalence still require separate definitions. We therefore state each theorem for a concrete object. Section 12 discusses the additional structure needed for the original native conjectures.

The Lean declarations cited in this section have namespace prefix `KomorebiPaperIX`. Their exact modules and source hashes are recorded in the accompanying statement map. All belong to the preserved baseline formalization. This section brings their statements and proofs together.

2.1 Residue selectors and exact reconstruction

Let R be a commutative ring, $a : \mathbb{N} \to R$, $N \geq 1$, and $0 \leq j < N$. Define

$$(P_{N,j}a)(n) = \begin{cases} a(n), & n \equiv j \pmod{N}, \\ 0, & \text{otherwise.} \end{cases}$$

This definition keeps the original index set. The compressed subsequence $a(Nq + j)$, used below for recurrence packets, is a different representation of the selected coefficients.

Theorem 2.1 (Residue reconstruction)**.** *The selectors satisfy*

$$P_{N,j}^2 = P_{N,j}, \qquad P_{N,i}P_{N,j} = 0 \quad (i \neq j), \qquad \sum_{j=0}^{N-1} P_{N,j} = I.$$

If $N \mid M$ and $M > 0$, then

$$P_{N,j}a = \sum_{\substack{0 \leq r < M \\ r \equiv j \pmod{N}}} P_{M,r}a.$$

The same identities hold coefficientwise for formal power series.

Proof. At each index n, exactly one standard residue label equals n mod N. Consequently a repeated selector changes nothing, two different selectors have zero composite, and the sum retains $a(n)$ exactly once. If $N \mid M$, reducing n mod M modulo N gives n mod N. This proves the refinement identity at every coefficient. Equality of all coefficients proves the formal-series versions. □

Formal correspondence. `ResiduePackets.packet_idempotent`, `ResiduePackets.packet_orthogonal`, `ResiduePackets.packet_reconstruction`, and `ResiduePackets.packet_refinement`. The series variants have the prefix `series_packet` in the same module. The algebraic statements need no convergence hypothesis.

2.2 Formal exponential coordinates and Hurwitz multiplication

Over a characteristic-zero field K, put

$$\mathcal{E}(a) = \sum_{n\ge 0} \frac{a_n}{n!} t^n, \qquad \mathcal{D}(A)_n = n![t^n]A.$$

These inverse maps between sequences and $K[[t]]$ provide *formal* exponential coordinates. No analytic convergence is needed for the coefficient identities below.

Theorem 2.2 (Constructed unit-triangular coordinates)**.** *Let $h : \mathbb{N} \to K$ satisfy $h_0 = 1$, write $H = \mathcal{E}(h)$ and $H^{-1} = \mathcal{E}(k)$, and define*

$$U_H a = \mathcal{D}(H\mathcal{E}(a)).$$

Then U_H is an equivalence, with inverse $U_H^{-1}a = \mathcal{D}(H^{-1}\mathcal{E}(a))$. Its coefficient formulas are

$$(U_H a)_n = \sum_{r=0}^{n} \binom{n}{r} h_r a_{n-r}, \qquad (U_H^{-1} a)_n = \sum_{r=0}^{n} \binom{n}{r} k_r a_{n-r}.$$

The resulting triangular matrix has diagonal entries 1*. The transported selectors*

$$\widehat{P}_{N,j} = U_H P_{N,j} U_H^{-1}$$

are idempotent and pairwise orthogonal, and their sum is the identity.

Proof. The factorials are nonzero in K, so encoding and decoding cancel coefficientwise. The constant coefficient of H is 1. Hence its formal power-series inverse exists. Multiplication by H and H^{-1} gives the two inverse laws. The ordinary product coefficient formula, multiplied by $n!$, gives the displayed binomial sums. In row n, the coefficient of a_n is $h_0 = 1$ and coefficients of a_q with $q > n$ vanish. Conjugating the identities of Theorem 2.1 by the constructed equivalence proves the transported projector identities. □

Formal correspondence. `EGF.decode_encode`, `EGF.encode_decode`, `EGF.egf_product`, `EGF.coordinateEquiv`, `EGF.coordinate_forward`, `EGF.coordinate_inverse`, and `EGF.transported_reconstruction`, together with the idempotence and orthogonality declarations in `EGFCoordinates`.

In characteristic p, factorial division is unavailable in general, but the coefficient operation remains meaningful over every commutative ring:

$$(f \star g)_n = \sum_{r=0}^{n} \binom{n}{r} f_r g_{n-r}, \qquad (D^d f)_n = f_{n+d}, \qquad \mathbf{1}_n = \begin{cases} 1, & n = 0, \\ 0, & n > 0. \end{cases}$$

Here $\star$ is Hurwitz multiplication. The symbol D^d denotes a coefficient shift, which agrees with formal differentiation after characteristic-zero encoding.

Theorem 2.3 (Hurwitz product rules)**.** *For $d \ge 0$,*

$$D^d(f \star g) = \sum_{r=0}^{d} \binom{d}{r} (D^r f) \star (D^{d-r} g).$$

If R has prime characteristic p, then

$$D^p(f \star g) = f \star D^p g + D^p f \star g.$$

Over a field, if $g_0 = 1$, a sequence b is constructed recursively with $b \star g = \mathbf{1}$. If also $D^p g = Dg$ in characteristic p, then $D^p b = Db$.

Proof. Pascal's identity proves the first product rule, and induction proves the higher rule. The intermediate binomial coefficients $\binom{p}{r}$ vanish in characteristic p. For the inverse, solve row n using the coefficient of b_n, which is $g_0 = 1$:

$$b_n = \mathbf{1}_n - \sum_{r<n} \binom{n}{r} b_r g_{n-r}.$$

Subtract the first and pth product rules for $b \star g = \mathbf{1}$. The assumed identity for g leaves $(D^p b - Db) \star g = 0$. A second triangular induction shows that convolution with a sequence having constant coefficient 1 has trivial kernel. □

Formal correspondence. `Hurwitz.shift_convolution`, `Hurwitz.prime_shift_convolution`, `Hurwitz.convolution_left_eq_zero`, `HurwitzInverse.inverse_convolution`, and `HurwitzInverse.inverse_prime_difference`. Working directly with coefficient sequences avoids the undefined expression $1/n!$ in $\mathbb{F}_p$ when $p \mid n!$.

2.3 Triangular inversion and a finite precision filtration

Theorem 2.4 (Forward substitution and denominator clearing)**.** *Let T_{nq} be coefficients over a field with $T_{nn} \neq 0$ for every n. For any sequence b, define*

$$a_n = \frac{b_n - \sum_{q<n} T_{nq} a_q}{T_{nn}}.$$

This constructs the unique solution of $\sum_{q=0}^{n} T_{nq} a_q = b_n$ for all n. If T_{nq} and b_n are integers and the solution is taken over $\mathbb{Q}$, then

$$\left(\prod_{r=0}^{n} T_{rr} \right) a_n \in \mathbb{Z}.$$

Proof. The recursion uses only smaller indices, and substitution verifies each row. If a second solution agrees through row $n-1$, subtracting row n and cancelling T_{nn} proves equality at n. For integrality, use induction and promote every earlier clearing product to $\prod_{r<n} T_{rr}$. Multiplication by the additional factor T_{nn} cancels the only new division. □

Formal correspondence. `Triangular.solve_row`, `Triangular.solve_unique`, and `Triangular.solve_denominator_integral`. The clearing product bounds the denominator. The unscaled solution can still be nonintegral.

To isolate the filtration argument, let p be any element of a commutative ring and let $w : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$. Set $p^{[r]} = p^r$ for finite r and $p^{[\infty]} = 0$. The latter notation means the coefficient vanishes.

Theorem 2.5 (Filtered finite reconstruction)**.** *For every $n, L \in \mathbb{N}$ and sequences c, S in that ring,*

$$\sum_{r=0}^{n} \binom{n}{r} c_r p^{[w(r)]} S_{n-r} \equiv \sum_{\substack{0 \le r \le n \\ w(r) < L}} \binom{n}{r} c_r p^{[w(r)]} S_{n-r} \pmod{p^L}.$$

Here congruence means divisibility of the difference by p^L in the coefficient ring. In $\mathbb{Z}_p$, it means membership in $p^L \mathbb{Z}_p$.

Proof. The difference of the two sums consists exactly of terms with $w(r) \ge L$. Every such term is divisible by p^L, including the zero terms with infinite weight. Their finite sum has the same divisibility. □

Formal correspondence. `Filtered.filtered_triangular_reconstruction` and `Filtered.padic_filtered_reconstruction`. For weights $2r$, retaining $r \le M$ gives the bound p^{2M+2} through `Filtered.even_weight_tail`. A sharper kernel weight can improve this bound, as in the classical dyadic reconstruction later in the chapter. Each coefficient row is finite, so no hypothesis $w(r) \to \infty$ is needed here.

2.4 Finite Mahler coordinates and weighted differences

For $f : \mathbb{N} \to R$, define the step-h forward difference and its iterates by

$$\Delta_h f(n) = f(n+h) - f(n), \qquad \Delta_h^0 f = f,$$

and put $c_r(f) = \Delta_1^r f(0)$. Thus the coefficients are determined by the differences of f.

Theorem 2.6 (Finite Newton–Mahler identities)**.** *For every $n, s, h \in \mathbb{N}$,*

$$\begin{aligned} f(n) &= \sum_{r=0}^{n} c_r(f) \binom{n}{r}, \\ c_r(\Delta_1^s f) &= c_{s+r}(f), \\ \Delta_1^s f(n) &= \sum_{j=0}^{n} c_{s+j}(f) \binom{n}{j}, \\ \Delta_h^s f(n) &= \sum_{j=0}^{s} (-1)^{s-j} \binom{s}{j} f(n+jh). \end{aligned}$$

The finite interpolant $I_N f(n) = \sum_{r=0}^{N} c_r(f) \binom{n}{r}$ agrees with f for $n \le N$. It has the same order-s unit-step difference at n whenever $n + s \le N$. For arbitrary coefficients $c_0, \dots, c_N$, the corresponding binomial interpolant has exactly those coefficients and zero coefficients above N.

Proof. Write $E_h f(n) = f(n+h)$ for the shift operator. The unit shift is $I + \Delta_1$. Expanding its nth power and evaluating at zero gives reconstruction. Composing difference operators gives the coefficient shift and applying reconstruction to $\Delta_1^s f$ gives the third identity. The binomial expansion of $(E_h - I)^s$ gives the explicit stencil. Binomial coefficients vanish when their lower index exceeds n, which proves the interpolation claim. Equality of every point of the stencil proves equality of differences. Repeated finite differences of $\binom{n}{r}$ at zero give the exact coefficient extraction. □

Formal correspondence. `Mahler.finite_mahler_reconstruction`, `Mahler.coefficient_difference`, `Mahler.difference_from_coefficients`, `Mahler.difference_explicit`, `Mahler.finite_sample_difference`, and `Mahler.coefficient_interpolant`. Finite samples determine the stencils contained in the sample interval. A universal tail estimate requires additional information.

For $x \in \mathbb{Z}_p$, extend the valuation to zero by setting

$$\operatorname{ord}_p(x) = \begin{cases} v_p(x), & x \neq 0, \\ \infty, & x = 0. \end{cases}$$

Thus $L \leq \operatorname{ord}_p(x)$ is equivalent to $x \in p^L\mathbb{Z}_p$.

Theorem 2.7 (Weighted difference bound)**.** *Let $f : \mathbb{N} \to \mathbb{Z}_p$ and let $w : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ satisfy $w(s+j) \leq \operatorname{ord}_p(c_{s+j}(f))$ for $0 \leq j \leq n$. Then*

$$\operatorname{ord}_p(\Delta_1^s f(n)) \geq \min_{0 \leq j \leq n} \left\{ w(s+j) + v_p\binom{n}{j} \right\}.$$

Proof. Apply the nonarchimedean inequality to the finite coefficient formula in Theorem 2.6. For $0 \leq j \leq n$, the integer $\binom{n}{j}$ is positive. The order of its product with $c_{s+j}(f)$ is the sum of the two orders, including the case where that coefficient is zero. □

Formal correspondence. `PadicBounds.nat_le_order_iff`, `PadicBounds.weighted_difference_bound`, and `PadicBounds.weighted_difference_divisibility`. An expansion on all of $\mathbb{Z}_p$ requires a convergence theorem in addition to these finite algebraic identities and bounds.

2.5 Unit modes and the dyadic distinction

Theorem 2.8 (Differences of unit modes)**.** *Over every commutative ring,*

$$\Delta_h^s(u^n) = u^n(u^h - 1)^s.$$

For any prime p, units $u_i \in \mathbb{Z}_p^\times$, coefficients $a_i \in \mathbb{Z}_p$, and a finite index set I, the sequence $f(n) = \sum_{i \in I} a_i u_i^n$ satisfies

$$\Delta_{p-1}^s f(n) \in p^s\mathbb{Z}_p \qquad (n, s \in \mathbb{N}).$$

Proof. Induction gives the geometric identity. Reduction of a unit to $\mathbb{F}_p^\times$ gives $u_i^{p-1} = 1$ modulo p. Taking s powers supplies p^s in each summand. The difference operator is linear over finite sums. □

Formal correspondence. `UnitModes.geometric_difference`, `UnitModes.unit_fermat_divisibility`, and `UnitModes.finite_unit_modes_divisibility`. The result includes $p = 2$, where the displayed step is 1. Its scope here is finite sums of modes.

Theorem 2.9 (Finite unit coordinates)**.** *Let p be prime, $L \geq 0$, and $a \in (\mathbb{Z}/p^{L+1}\mathbb{Z})^\times$. Define*

$$\omega_L(a) = a^{p^L}, \qquad \langle a \rangle_L = a\omega_L(a)^{-1}.$$

Then

$$a = \omega_L(a)\langle a \rangle_L, \qquad \omega_L(a)^{p-1} = 1, \qquad \langle a \rangle_L \equiv 1 \pmod{p}.$$

The two factors are uniquely determined by these properties and are compatible with reduction to lower precision. For $a \in (\mathbb{Z}/2^{L+2}\mathbb{Z})^\times$, there is instead a unique decomposition

$$a = \epsilon(a)\langle a\rangle_4, \qquad \epsilon(a) \in \{1,-1\}, \qquad \langle a\rangle_4 \equiv 1 \pmod 4.$$

Proof. The unit-group exponent $p^L(p-1)$ gives the root identity for ω_L. Its residue is that of a, so the remaining factor has residue 1. If two $(p-1)$st roots have equal residue, raising them to p^L makes them equal modulo p^{L+1}, while each root is fixed by this exponent. This proves uniqueness and then compatibility under reduction. In the dyadic construction, an odd residue is exactly 1 or -1 modulo 4. Choose that sign and multiply a by it to obtain the principal factor. The two signs remain distinct modulo 4, which proves uniqueness. □

Formal correspondence. `FiniteUnits.finite_unit_decomposition`, `FiniteUnits.decomposition_unique`, `UnitRefinement.omega_compatible`, `UnitRefinement.principal_compatible`, and `DyadicUnits.dyadic_unit_decomposition` with `DyadicUnits.decomposition_unique`. The finite $p = 2$ version of the first construction has trivial root-of-unity factor. The separate modulo-4 construction retains the sign that reduction modulo 2 cannot detect.

2.6 Recurrence packets and finite arithmetic synchronization

Theorem 2.10 (Companion jumps and packet recurrences)**.** *Let C be a $d \times d$ matrix over a commutative ring and $x_n = C^n x_0$. For $N, j, k \in \mathbb{N}$, put $b_k = x_{Nk+j}$. Then*

$$b_k = C^j (C^N)^k x_0, \qquad b_{k+1} = C^N b_k.$$

If $\chi_{C^N}(X) = X^d + \sum_{r<d} q_r X^r$, then

$$b_{k+d} = -\sum_{r=0}^{d-1} q_r b_{k+r}.$$

Every coordinate and every linear scalar observable satisfies the same recurrence. No invertibility or diagonalization hypothesis is required.

Proof. The jump identity follows from the power law for one matrix. Apply the Cayley–Hamilton identity $\chi_{C^N}(C^N) = 0$ to the state b_k. The leading coefficient is 1, so the final state can be isolated. A linear observable preserves the resulting finite sum. □

Formal correspondence. `Companion.companion_jump`, `Companion.packet_charpoly_recurrence`, and `Companion.packet_observable_recurrence`. The source-sequence versions derive the packet recurrence from the relation $x_{n+1} = Cx_n$. For the Fibonacci matrix $C = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, its jump trace is the Lucas number L_N and its determinant is $(-1)^N$. Thus

$$F_{N(k+2)+j} = L_N F_{N(k+1)+j} - (-1)^N F_{Nk+j}.$$

This specialization is `FibonacciPackets.fibonacci_packet_recurrence`. It is derived from the Fibonacci state.

Theorem 2.11 (Dyadic–odd packet factorization)**.** *Let $a \geq 0$ and let M be a positive odd integer. The Chinese remainder map is a ring isomorphism*

$$\mathbb{Z}/(2^a M)\mathbb{Z} \simeq \mathbb{Z}/2^a\mathbb{Z} \times \mathbb{Z}/M\mathbb{Z}.$$

For $0 \leq r < 2^a M$ and any coefficient sequence f,

$$P_{2^a M, r} f = P_{2^a, r \bmod 2^a}\, P_{M, r \bmod M} f.$$

The same identity holds for formal-series coefficient selectors.

Proof. The two moduli are coprime. An index has the prescribed combined residue exactly when it has both component residues. Applying this equivalence to the indicator defining each selector proves the formula. □

Formal correspondence. `CRTPackets.addressEquiv`, `CRTPackets.residue_equivalence`, `CRTPackets.packet_factorization`, and `CRTPackets.series_packet_factorization`. This theorem synchronizes finite arithmetic addresses. The passage from a family of p-adic objects to a global integer object is a separate question, discussed in Section 12.3.

2.7 Polyphase notation retained from the recovered manuscript

The recovered Annals text develops a broader polyphase vocabulary. We explain it through elementary identities with explicit algebraic derivations. These identities have not been separately formalized in Lean for this consolidation. For a sequence a and $M \geq 1$, let $a_r^{[M]}(q) = a_{Mq+r}$ for $0 \leq r < M$. If $A(z) = \sum_n a_n z^n$ and $A_r(w) = \sum_q a_{Mq+r} w^q$ are ordinary generating functions, then unique quotient and remainder give

$$A(z) = \sum_{r=0}^{M-1} z^r A_r(z^M), \qquad \left(a_r^{[M]}\right)_s^{[N]} = a_{r+Ms}^{[MN]}.$$

For ordinary Cauchy multiplication, the phase-r product has a term without a carry and a term with one carry:

$$C_r(w) = \sum_{u+v=r} A_u(w)B_v(w) + w \sum_{u+v=r+M} A_u(w)B_v(w), \qquad 0 \leq u, v < M.$$

Adding the two phase addresses gives the formula. It describes ordinary Cauchy multiplication. Hurwitz convolution has its own binomial weights.

For $M \geq 1$ and $s, r, k \in \mathbb{N}$, define the affine Mahler kernel by

$$\Theta_{M,s}(r,k) = [z^r](1+z)^s\left((1+z)^M - 1\right)^k = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} \binom{Mj+s}{r}.$$

It vanishes unless $k \leq r \leq Mk + s$. For a finite binomial expansion $f(n) = \sum_{r=0}^{D} c_r \binom{n}{r}$, the kth Newton coefficient of $g(q) = f(Mq + s)$ is therefore

$$c_k(g) = \sum_{r=k}^{\min(D, Mk+s)} c_r \Theta_{M,s}(r,k).$$

Indeed, insert the finite stencil for $\Delta_1^k g(0)$ and interchange its two finite sums. Coefficient extraction of the binomial expansion of $((1+z)^M - 1)^k$ gives the kernel. In the binary zero-address case,

$$\Theta_{2,0}(k+j,k) = 2^{k-j}\binom{k}{j} \qquad (0 \leq j \leq k),$$

since $(2z + z^2)^k = z^k(2+z)^k$. Exact valuations for the general kernel remain to be determined. The recovered infinite analytic extensions and stronger historical identities lie beyond these finite derivations. Their status is discussed in Section 12.5.

3 Congruential Euler packets and their arithmetic

Two related constructions run through this section. The congruential Euler family is indexed by a modulus N and an address j. Its exceptional dyadic member has $N = 4$ and four addresses. The classical signed Euler numbers arise from $N = 2$, $j = 0$. Keeping these indices visible distinguishes the four-packet period theorems from the classical edge truncations.

3.1 Original coefficients, normalized packets, and precision

Definition 3.1 (Congruential Euler coefficients)**.** For $N \geq 1$ and $0 \leq j < N$, define rational coefficients by the formal identity

$$\sum_{d\geq 0} E_d^{(N,j)} \frac{t^d}{d!} = \left(\sum_{n\geq 0} \frac{t^{Nn}}{(Nn+j)!}\right)^{-1}. \tag{3.1}$$

The reciprocal exists because its denominator has constant coefficient $1/j! \neq 0$. Set

$$U_{N,j}(n) = E_{Nn}^{(N,j)}, \qquad C_{N,j}(n) = \frac{U_{N,j}(n)}{j!}.$$

Thus $C_{N,j}(0) = 1$ and $U_{N,j}(0) = j!$. All coefficients outside the multiples of N vanish.

The normalized source and its inverse are

$$K_{N,j}(t) = j! \sum_{n\geq 0} \frac{t^{Nn}}{(Nn+j)!}, \qquad \mathcal{P}_{N,j}(t) = \sum_{n\geq 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}. \tag{3.2}$$

Their product is one. Clearing factorials in its coefficient of t^{Nn} gives the triangular system

$$\sum_{q=0}^{n} \binom{Nn+j}{Nq} C_{N,j}(q) = \delta_{n0}. \tag{3.3}$$

In particular, for $n > 0$,

$$C_{N,j}(n) = -\binom{Nn+j}{Nn}^{-1} \sum_{q=0}^{n-1} \binom{Nn+j}{Nq} C_{N,j}(q).$$

The diagonal binomial coefficient is positive, so this recurrence determines the sequence uniquely. The Lean construction begins with this recurrence and derives the reciprocal identity from it. The resulting sequence is then identified with the degree-indexed coefficients of (3.1), so both descriptions refer to the same family.

The relevant declarations are `KomorebiPaperIX.Congruential.triangular_inverse`, `KomorebiPaperIX.Congruential.packet_is_actual_inverse`, and `KomorebiPaperIX.NishibuchiCompletion.generalizedEuler_at_packet`. The cleared identity with the original positive-address exponential section is `KomorebiPaperIX.NishibuchiCompletion.original_generating_identity`.

Proposition 3.2 (A denominator bound for every address)**.** *For $N \geq 1$, $0 \leq j < N$, and $n \in \mathbb{N}$, set*

$$D_{N,j}(n) = \prod_{r=1}^{n} \binom{Nr+j}{Nr}, \qquad D_{N,j}(0) = 1.$$

Then $D_{N,j}(n)C_{N,j}(n) \in \mathbb{Z}$.

Proof. Proceed by induction in the recurrence (3.3). For every $q < n$, the product $D_{N,j}(q)$ divides $D_{N,j}(n-1)$. Multiplying the recurrence by $D_{N,j}(n-1)$ therefore makes every previous term integral. The diagonal factor completes the product $D_{N,j}(n)$, proving the claim. At $n = 0$, both factors are one. This gives a uniform denominator bound without requiring integrality of the normalized coefficients themselves. The Lean statements are `KomorebiPaperIX.Congruential.clearing_product` and `KomorebiPaperIX.Congruential.universal_denominator_clearing`. □

Example 3.3 (The Bernoulli boundary address)**.** At $N = 2$, the address $j = 1$ gives a second classical family. Define the Bernoulli numbers by

$$B(z) = \frac{z}{e^z - 1} = \sum_{r \geq 0} B_r \frac{z^r}{r!}, \qquad B_0 = 1, \quad B_1 = -\frac{1}{2}.$$

All quotients here are regular formal series. The identity

$$2B(z) - B(2z) = \frac{z}{\sinh z} = \sum_{n \geq 0} C_{2,1}(n) \frac{z^{2n}}{(2n)!}$$

follows by putting the two fractions on a common denominator and using (3.1). Comparing even coefficients yields

$$C_{2,1}(n) = (2 - 2^{2n}) B_{2n} \qquad (n \in \mathbb{N}).$$

This classical derivation restores the boundary example from the original manuscript. No dedicated Lean bridge for this Bernoulli specialization was identified in the preserved sources.

Definition 3.4 (Differences and valuation conventions)**.** For a sequence f and $h, k, n \in \mathbb{N}$, put

$$\Delta_h f(n) = f(n+h) - f(n), \qquad \Delta_h^0 f(n) = f(n), \qquad \Delta_h^{k+1} f(n) = \Delta_h^k f(n+h) - \Delta_h^k f(n).$$

We use the extended valuation $v_p(0) = +\infty$ in mathematical statements. Thus $v_p(x) \geq L$, including negative integers L, means $x = 0$ or the ordinary valuation of the nonzero rational x is at least L. For integral coefficients and $L \geq 0$, it is equivalent to $x \in p^L \mathbb{Z}_p$. In expressions involving a step multiplier we write $\nu_p(u) = v_p(u)$ for $u > 0$ and set $\nu_p(0) = 0$. This finite convention makes the displayed exponents meaningful at step zero, where all positive-order differences vanish.

The valuation convention at zero is encoded by `KomorebiPaperIX.GlobalEulerBoundary.precision`. For p-adic congruences, divisibility is interpreted in $\mathbb{Z}_p$. Statements over other coefficient rings specify the ring explicitly. Field divisibility in $\mathbb{Q}_p$ alone does not encode precision.

3.2 The standard address window and its common reduction

The addresses below the prime have unit diagonal coefficients. In this window the normalized packets are integral, and their reductions agree at every row.

Proposition 3.5 (Lucas reduction and address independence)**.** *Let p be prime, $m > 0$, and $N = mp$. For $0 \leq j < p$ and $n, q \in \mathbb{N}$,*

$$\binom{mpn + j}{mpq} \equiv \binom{mn}{mq} \pmod{p}.$$

Every $C_{mp,j}(n)$ belongs to $\mathbb{Z}_p$. Its reduction is the unique sequence $\overline{C}_m : \mathbb{N} \to \mathbb{F}_p$ satisfying

$$\sum_{q=0}^{n} \binom{mn}{mq} \overline{C}_m(q) = \delta_{n0} \quad \text{in } \mathbb{F}_p.$$

Consequently, for any two addresses $0 \le j, k < p$,

$$C_{mp,j}(n) \equiv C_{mp,k}(n) \pmod{p} \qquad (n \in \mathbb{N}).$$

Proof. Lucas reduction separates the lowest base-p digits, which are j and zero in the upper and lower arguments. Their binomial contribution is one, leaving the displayed reduced row. On the diagonal the reduced coefficient is one. The recurrence (3.3) therefore constructs an integral packet by successive division by units in $\mathbb{Z}_p$. After reduction, every address satisfies the same triangular system over $\mathbb{F}_p$. Its diagonal entries are one, so its solution is unique. This proves both the common reduction and the congruence between addresses. The row reduction is `KomorebiPaperIX.StandardWindow.lucas_row_reduction`. The integral identification and the two reduction statements are `KomorebiPaperIX.WindowIndependence.standard_window_integral`, `KomorebiPaperIX.WindowIndependence.coeffPadic_coe`, `KomorebiPaperIX.WindowIndependence.residue_independence`, and `KomorebiPaperIX.WindowIndependence.two_window_addresses_congruent`. □

3.3 The full Nishibuchi statement and its starting bound

Theorem 3.6 (Nishibuchi congruence)**.** *Let p be prime, let $m > 0$, and suppose that $m \mid p-1$ or $m = 2$. Put $N = mp$ and $q = \operatorname{lcm}(2, p-1)$. For every $0 \le j < N$, all $U_{N,j}(n)$ belong to $\mathbb{Z}_p$. For $r \ge 1$ and $n \ge r$,*

$$E^{(N,j)}_{Nn+qp^r} \equiv E^{(N,j)}_{Nn} \pmod{p^r}. \tag{3.4}$$

Thus the positive sufficient starting index $n_0 = r$ works uniformly. If $m \mid p-1$, the stronger sufficient condition $r \le mn$ is valid.

This theorem covers the full parameter range and displacement of Conjecture 3.9 in Nishibuchi's cited statement. The corresponding Lean theorem is KomorebiPaperIX.NishibuchiCompletion.conjecture_3_9. The stronger divisor-branch threshold is `KomorebiPaperIX.NishibuchiCompletion.divisor_period`. The starting index is sufficient uniformly, although it need not be the earliest one for a given tuple of parameters.

We first prove the difference estimates on the divisor branch, then derive the period from a finite Newton expansion.

Theorem 3.7 (All-address differences on the divisor branch)**.** *Let p be prime, $m > 0$, $m \mid p-1$, $N = mp$, and $0 \le j < N$. Write $h = (p-1)/m$. For every $u, k, n \in \mathbb{N}$,*

$$U_{N,j}(n) \in \mathbb{Z}_p, \tag{3.5}$$

$$\Delta^k_{uh} U_{N,j}(n) \in p^{\min(k,mn)} \mathbb{Z}_p, \tag{3.6}$$

$$v_p\big(\Delta^k_{uh} C_{N,j}(n)\big) \ge \min(k, mn) - v_p(j!). \tag{3.7}$$

The factorial correction is part of the normalized statement, including addresses $j \ge p$.

Proof. Work in the Hurwitz integral ring of formal series

$$\mathcal{H}_p = \left\{ F \in \mathbb{Q}_p[[t]] : d![t^d]F \in \mathbb{Z}_p \text{ for every } d \right\}.$$

For $D = d/dt$, let $B_p = (D^{p^2} - D^p)/p$. Call F flat if $B_p^s F \in \mathcal{H}_p$ for every $s \geq 0$. We will prove that $j!\mathcal{P}_{N,j}$ is flat at every address. The main difficulty is to include the addresses at which the usual unit-pivot argument fails.

Set

$$H_c(t) = \sum_{n \geq 0} \frac{t^{Nn+c}}{(Nn+c)!}, \qquad A_c(t) = \sum_{n \geq 0} \frac{t^{Nn}}{(Nn+c)!}, \qquad R_c(t) = \frac{A_{c-1}(t)}{A_c(t)} \quad (1 \leq c < N).$$

The denominator of this last quotient has nonzero constant coefficient. The section identities give

$$R_c H_c = t H_{c-1}, \qquad R_c(0) = c, \qquad j!\mathcal{P}_{N,j} = \mathcal{P}_{N,0} \prod_{c=1}^{j} R_c.$$

Cancellation of consecutive source sections proves the product identity and ties the ratios directly to the family under study.

At the singular addresses $c = ap$, $1 \leq a < m$, use the weighted ring

$$\mathcal{W}_p = \left\{ \sum_{d \geq 0} a_d T^d : p^{\lfloor d/p \rfloor} a_d \in \mathbb{Z}_p \right\}.$$

The key estimate is $R_{ap}(\log(1+T)) \in \mathcal{W}_p$. Its arithmetic input comes from the cyclic group ring $\mathbb{Z}[\mathbb{Z}/N\mathbb{Z}]$. For its canonical element x and $I = (p, x^{p-1} - 1)$, the relation $(x^{p-1})^p = 1$ implies $I^p \subseteq (p)I$. Induction yields $I^q \subseteq (p^{v_p(q!)})$. Grouping the falling factorial $(x)_d = \prod_{i=0}^{d-1}(x-i)$ into consecutive blocks of length p then gives an integral residue-coordinate factor $p^{v_p(\lfloor d/p \rfloor !)}$. The identity

$$v_p(d!) = \lfloor d/p \rfloor + v_p(\lfloor d/p \rfloor !)$$

accounts for precisely the weight in $\mathcal{W}_p$ after dividing by $d!$. Comparison of the section and group-ring recurrences identifies these coordinates with the logarithmically substituted section.

To pass from the section to the singular ratio, factor out its initial zero: for $c = ap$, the section has the form

$$H_c(\log(1+T)) = \frac{T^c}{c!} V(T), \qquad V(0) = 1, \qquad V \in \mathcal{W}_p.$$

The weight of $c!$ compensates for the coefficient shift by c. The weighted ring is closed under inversion of series with constant coefficient one and under $L_T = \log(1+T)(1+T)d/dT$. With $\ell(T) = \log(1+T)/T$, interpreted by coefficient shifting, differentiation of the displayed factorization gives

$$R_c(\log(1+T)) = c(1+T)\ell(T) + (L_T V)V^{-1}.$$

Since $p\ell(T) \in \mathcal{W}_p$ and $p \mid c$, both terms are weighted. This is the singular estimate `KomorebiPaperIX.SingularRatioAmice.paper_singular_amice_estimate`.

To deduce flatness from the weighted estimate, use $\Theta = (1+T)d/dT$. Both Θ and $(\Theta^{p^2} - \Theta^p)/p$ preserve $\mathcal{W}_p$. For the first operator, its coefficient formula is $(d+1)a_{d+1} + da_d$. Whenever the floor

weight rises, the factor $d+1$ provides the missing p. For the second operator, the proof uses its full Leibniz correction and the scalar congruence $p^2 \mid b^{p^2} - b^p$. Weighted monomials factor as

$$a_d T^d = (p^{\lfloor d/p \rfloor} a_d)(T^p/p)^{\lfloor d/p \rfloor} T^{d \bmod p}.$$

The operator preserves these monomials and hence finite weighted polynomials. Each fixed output coefficient depends on only finitely many input coefficients, which extends the result to the formal series. Substitution $T = e^t - 1$ intertwines D with Θ and maps $\mathcal{W}_p$ into $\mathcal{H}_p$, because

$$d![t^d]F(e^t - 1) = [T^0]\Theta^d F.$$

It follows that the singular ratios are flat. This transfer is proved in `KomorebiPaperIX.WeightedDworkTransfer.singular_ratio_flat`.

At a nonsingular address $2 \le c < N$ with $p \nmid c$, differentiation of the section identities gives

$$tR_c' = R_c(1 + R_{c-1} - R_c).$$

Writing $q = R_{c-1} - (c-1)$ and $y = R_c - c$, this becomes

$$(tD + c)y = cq + yq - y^2.$$

On degrees divisible by p, the inverse of $tD + c$ divides the degree-d coefficient by the unit $d + c$. Starting with $y_0 = 0$, form

$$y_{s+1} = (tD + c)^{-1}(cq + y_s q - y_s^2).$$

Flatness and support are preserved. Indeed, put $E_c = tD + c$ and $C_p = pD^{p^2} - D^p$. The commutator is $[B_p, E_c] = C_p$, and iteration gives

$$B_p^r E_c y = E_c B_p^r y + r C_p B_p^{r-1} y.$$

For the inverse equation $E_c y = f$, induction on r makes the correction term Hurwitz-integral. Both operators preserve support in degrees divisible by p. On that support the degree multiplier $d + c$ is a unit, so integrality of $E_c B_p^r y$ implies integrality of $B_p^r y$. Thus the inverse preserves full flatness, as well as support. Since the difference of two right sides factors as $(z - y)(q - z - y)$ and its second factor has zero constant term, agreement with the solution gains one coefficient at each iteration. Consequently $[t^d]y_s = [t^d]y$ for $d \le s$. Every coefficient of $B_p^r y$ can therefore be read from a sufficiently long finite iterate. The argument is entirely formal. This is the content of `KomorebiPaperIX.RiccatiFlatPropagation.riccati_solution_flat`.

The standard window supplies the initial ratio R_1. The weighted estimate handles $p \mid c$, and Riccati propagation handles $p \nmid c$. Flatness is closed under products, so the exact factorization establishes flatness of $j!\mathcal{P}_{N,j}$. Extracting its supported Hurwitz coefficients and applying the proved saturation and step-multiple lemmas yields (3.5)–(3.6).

Here is the coefficient extraction explicitly. If $a_d = d![t^d]F$, then

$$d![t^d]B_p^k F = p^{-k}\Delta_{p(p-1)}^k a_{d+pk}.$$

For $F = j!\mathcal{P}_{N,j}$ and $k \le mn$, take $d = mpn - pk$. Support in multiples of mp turns this into $p^{-k}\Delta_h^k U_{N,j}(n)$, so flatness gives divisibility by p^k. For $k > mn$, expand Δ_h^{k-mn} applied to $\Delta_h^{mn} U_{N,j}(n)$. Every shifted

row is at least n, so each summand is divisible by p^{mn}. Finally $\Delta_{uh} = ((I + \Delta_h)^u - I)$, whose kth power for $u > 0$ is an integral combination of powers of Δ_h of order at least k. The case $u = 0$ gives zero for $k > 0$, and order zero is the original integral coefficient. This proves the bound for all u and k. Finally $\Delta_{uh}^k U_{N,j} = j!\Delta_{uh}^k C_{N,j}$ gives (3.7), with the zero case retained. The resulting statements are `KomorebiPaperIX.AllResidueEuler.all_residue_flat`, `KomorebiPaperIX.AllResidueEuler.global_unnormalized_bound`, and `KomorebiPaperIX.AllResidueEuler.global_normalized_precision`. □

Example 3.8 (Why the factorial loss is necessary)**.** Take $p = 3$, $m = 2$, $N = 6$, and $j = 3$. The recurrence gives

$$C_{6,3}(1) = -\frac{1}{84}, \qquad C_{6,3}(2) = \frac{703}{5460}, \qquad C_{6,3}(2) - C_{6,3}(1) = \frac{64}{455}.$$

The last difference has 3-adic valuation zero. Here $h = 1$, so at $n = 1$ the order-one bound without factorial loss would require valuation at least one. The corrected normalized bound is sharp, since

$$v_3\big(C_{6,3}(2) - C_{6,3}(1)\big) = 0 = 1 - v_3(3!).$$

Multiplication by 3! gives

$$U_{6,3}(2) - U_{6,3}(1) = \frac{384}{455}, \qquad v_3\big(U_{6,3}(2) - U_{6,3}(1)\big) = 1,$$

consistent with the unnormalized theorem. Thus the factorial correction is already necessary at a single step in the all-address parameter range. These exact values and the refutation of a universal bound without loss are formalized in `KomorebiPaperIX.Normalization.normalized_first`, `KomorebiPaperIX.Normalization.normalized_second`, `KomorebiPaperIX.Normalization.normalized_difference`, `KomorebiPaperIX.Normalization.difference_valuation`, `KomorebiPaperIX.Normalization.factorial_loss_is_sharp`, `KomorebiPaperIX.Normalization.unnormalized_difference`, `KomorebiPaperIX.Normalization.unnormalized_difference_valuation`, and `KomorebiPaperIX.Normalization.universal_no_loss_refuted`.

Corollary 3.9 (A common period in the standard window)**.** *Let p be prime, $m > 0$, $m \mid p - 1$, and $0 \le j < p$. Put $h = (p-1)/m$. For every $n \ge 1$ and $u \in \mathbb{N}$,*

$$\overline{C}_m(n + uh) = \overline{C}_m(n), \qquad C_{mp,j}(n + uh) \equiv C_{mp,j}(n) \pmod{p}.$$

Thus h is a sufficient period from row one, common to all addresses in the standard window. It need not be the least period.

Proof. Since $j < p$, the factorial $j!$ is a unit in $\mathbb{Z}_p$. Also $mn \ge 1$. The order-one case of Theorem 3.7 therefore gives valuation at least one for the normalized difference at every step uh. This proves the congruence, and Proposition 3.5 identifies its reduction with the equality for $\overline{C}_m$. The preserved Lean development also proves this period directly from the Hurwitz inverse in characteristic p: `KomorebiPaperIX.WindowPeriod.uniform_standard_window_period`, `KomorebiPaperIX.WindowPeriod.multiple_period`, and `KomorebiPaperIX.WindowPeriod.standard_window_period_divisibility`. □

Lemma 3.10 (Prime-power period from finite differences)**.** *Let R be a commutative ring, p a prime, $f : \mathbb{N} \to R$, $h, n \in \mathbb{N}$, and $r \ge 1$. If*

$$p^{\min(k,r)} \mid \Delta_h^k f(n) \quad \text{for every } k \in \mathbb{N},$$

then $p^r \mid f(n + p^{r-1}h) - f(n)$.

Proof. The finite Newton expansion gives

$$f(n+p^{r-1}h)-f(n)=\sum_{k=1}^{p^{r-1}}\binom{p^{r-1}}{k}\Delta_h^k f(n).$$

Terms with $k\geq r$ have the required factor from the difference. For $1\leq k<r$, the identity

$$k\binom{p^{r-1}}{k}=p^{r-1}\binom{p^{r-1}-1}{k-1}$$

and $v_p(k)<k$ imply $p^r\mid\binom{p^{r-1}}{k}p^k$ in $\mathbb{Z}$. Mapping that integer divisibility to R proves the claim term by term. The argument includes $r=1$. The declaration is `KomorebiPaperIX.PeriodFromDifferences.prime_power_period`. □

Proof of Theorem 3.6. On the divisor branch, $m\mid q$ and the packet step q/m is an integer multiple of $h=(p-1)/m$. If $r\leq mn$, Theorem 3.7 supplies the hypotheses of Lemma 3.10. Its packet displacement is $p^{r-1}q/m$, which converts to the original degree as follows:

$$mp\left(n+p^{r-1}\frac{q}{m}\right)=mpn+qp^r.$$

The choice $n\geq r$ implies $mn\geq r$ because $m\geq 1$.

For an odd prime, $2\mid p-1$, so the option $m=2$ introduces no further case. The sole remaining branch is $p=2,m=2,N=4$. The all-row theorem proved next gives $2^k\mid\Delta_1^k C_{4,j}(n)$ at every row and address. After multiplying by $j!$, Lemma 3.10 gives packet period 2^{r-1} and degree displacement $4\cdot 2^{r-1}=2^{r+1}=qp^r$. It also gives integrality. Hence the complete statement follows. The formal degree conversion is `KomorebiPaperIX.NishibuchiCompletion.original_degree_shift`. The all-parameter degree theorem is `KomorebiPaperIX.NishibuchiCompletion.full_period_degree`. □

3.4 All-row dyadic differences and displacement amplification

For the remainder of the four-packet discussion abbreviate $C_j=C_{4,j}$ and $U_j=U_{4,j}$. Removing the original row cap requires an unshifted coefficient operator.

Theorem 3.11 (All-row dyadic higher differences)**.** *For $0\leq j<4$ and all $u,k,n\in\mathbb{N}$,*

$$C_j(n)\in\mathbb{Z}_2, \tag{3.8}$$

$$\Delta_u^k C_j(n)\in 2^{k(1+\nu_2(u))}\mathbb{Z}_2, \tag{3.9}$$

$$\Delta_u^k U_j(n)\in 2^{k(1+\nu_2(u))+v_2(j!)}\mathbb{Z}_2. \tag{3.10}$$

There is no restriction $k\leq n$ and no minimum with the row n.

Proof. The Hurwitz coefficients of the normalized source $K_{4,j}$ at degrees $4n$ are

$$g_j(n)=\binom{4n+j}{4n}^{-1}.$$

Their explicit expressions are

$$g_0(n) = 1, \qquad g_1(n) = \frac{1}{4n+1},$$
$$g_2(n) = \frac{1}{(4n+1)(2n+1)}, \quad g_3(n) = \frac{3}{(4n+1)(2n+1)(4n+3)}.$$

Every denominator is odd. An affine sequence with even slope satisfies $2^k \mid \Delta_1^k f(n)$ for every k, n. The exact product identity

$$\Delta_1^s(fg)(n) = \sum_{i=0}^{s} \binom{s}{i} \Delta_1^i f(n) \Delta_1^{s-i} g(n+i)$$

proves closure of this property under products. Applying the formula with the factors interchanged to $fg = 1$ isolates $f(n+s)\Delta_1^s g(n)$. Multiplication by $f(n+s)^{-1}$ and induction then prove closure under pointwise unit inverses. Thus all four g_j have the property. The explicit formulas and difference bounds are formalized in `KomorebiPaperIX.DyadicFourKernel.source_coe` and `KomorebiPaperIX.DyadicFourKernel.source_kummer`.

For an integer charge c, put

$$A_c = \frac{D^4 - c}{2}.$$

A series is flat of charge c when applying any power A_c^s to it gives an element of $\mathcal{H}_2$. The required product formula is

$$A_{c+d}(FG) = (A_c F)G + F(A_d G) + 2F'G''' + 3F''G'' + 2F'''G'. \tag{3.11}$$

All middle coefficients are integral. Derivatives preserve $\mathcal{H}_2$ and commute with A_c. Induction on the number of iterations in (3.11) proves that multiplication adds charges. If $FG = 1$, the same formula yields

$$A_{-c}G = -G^2 A_c F - G(2F'G''' + 3F''G'' + 2F'''G').$$

Induction, starting from the independently established integrality of the reciprocal G, proves that inversion negates the charge. Finally $A_{c+2} = A_c - 1$, so increasing a charge by two preserves flatness. These are `KomorebiPaperIX.ChargedDworkClosure.B_mul`, `KomorebiPaperIX.ChargedDworkClosure.flat_inverse`, and `KomorebiPaperIX.ChargedDworkClosure.flat_charge_shift`.

Packing g_j into degrees divisible by four gives $K_{4,j}$. Its source property is precisely charge-one flatness. Inversion gives charge -1, and the charge shift restores charge one for the reciprocal $\mathcal{P}_{4,j}$. If $\mathrm{dec}(F)(d) = d![t^d]F$, the exact coefficient identity is

$$\mathrm{dec}(A_1^s F)(d) = 2^{-s}\Delta_4^s \,\mathrm{dec}(F)(d).$$

At $d = 4n$, packing turns the right side into $2^{-s}\Delta_1^s C_j(n)$. This extraction requires no negative degree and therefore holds at every n, including zero. It proves the unit-step bound `KomorebiPaperIX.DyadicFourAllRows.base_step_bound`.

To incorporate $v_2(u)$, expand a displacement using finite Newton coefficients. For any prime p, $u > 0$, and $1 \le a \le u$,

$$p^{v_p(u)+1} \mid \binom{u}{a} p^a,$$

as follows from $a\binom{u}{a} = u\binom{u-1}{a-1}$ and $v_p(a) < a$. Repeated displacement, retaining the remaining base-difference order at each induction step, gives a factor $p^{k(1+v_p(u))}$ from k displacements. This induction is formalized in `KomorebiPaperIX.StepAmplification.repeated_precision`. Applying it with $p = 2$ proves (3.9). Multiplication by $j!$ proves (3.10). Zero step and zero order follow from their definitions. The final statements are `KomorebiPaperIX.DyadicFourFinalStrengthening.normalized_step_bound` and `KomorebiPaperIX.DyadicFourFinalStrengthening.unnormalized_step_bound`. □

Corollary 3.12 (Step-sensitive divisor-branch bound)**.** *Under the hypotheses of Theorem 3.7, for all $u, k, n \in \mathbb{N}$,*

$$\Delta^k_{u(p-1)/m} U_{mp,j}(n) \in p^{\min(k(1+v_p(u)),mn)}\mathbb{Z}_p.$$

For $C_{mp,j}$ the corresponding valuation lower bound is $\min(k(1+v_p(u)), mn) - v_p(j!)$.

Proof. Apply the same repeated-displacement theorem with the fixed cap mn to (3.6). The argument preserves this cap and improves the contribution from the order of the difference. Division by the fixed factorial gives the rational valuation statement, with the zero case included. These are `KomorebiPaperIX.AllResidueStepAmplification.unnormalized_precision` and `KomorebiPaperIX.AllResidueStepAmplification.normalized_precision`. □

Corollary 3.13 (Dyadic periods from row zero)**.** *For $0 \le j < 4$, $r \ge 1$, and $n \ge 0$,*

$$U_j(n + 2^{r-1}) \equiv U_j(n) \pmod{2^r}.$$

For $j = 2, 3$ and $r \ge 2$, 2^{r-2} is also a sufficient packet period. The respective original degree displacements are 2^{r+1} and 2^r.

Proof. Use Lemma 3.10 and the unit-step case of Theorem 3.11. At $j = 2, 3$ the factorial contributes an additional power of two. The original degree versions are `KomorebiPaperIX.DyadicFourFinalStrengthening.all_rows_period_degree` and `KomorebiPaperIX.DyadicFourFinalStrengthening.all_rows_half_period_degree`. The next theorem sharpens these sufficient periods to a complete classification. □

3.5 Exact displacement valuations and least packet periods

Lemma 3.14 (Finite Newton dominance)**.** *Let $f : \mathbb{N} \to \mathbb{Q}$ and write $a_k = \Delta^k_1 f(0)$. Suppose that $c \in \mathbb{N}$, $a_1 \ne 0$, $v_2(a_1) = c$, and*

$$v_2(a_k) \ge c + 1 + \lfloor \log_2 k \rfloor \qquad (k \ge 2).$$

Then for every $n \ge 0$ and $T > 0$,

$$f(n+T) - f(n) \ne 0, \qquad v_2\big(f(n+T) - f(n)\big) = v_2(T) + c.$$

Proof. Finite Newton reconstruction at an arbitrary row gives

$$\Delta^k_1 f(n) = \sum_{i=0}^{n} \binom{n}{i} a_{k+i}.$$

The monotonicity of $\lfloor \log_2 k \rfloor$ therefore preserves the higher-coefficient bound after translation. In addition, $\Delta_1 f(n) - a_1 \in 2^{c+1}\mathbb{Z}_2$, so the first difference is nonzero of exact valuation c.

Now expand the displacement:

$$f(n+T)-f(n)=T\Delta_1 f(n)+\sum_{k=2}^{T}\binom{T}{k}\Delta_1^k f(n).$$

For $1\le k\le T$, the binomial denominator identity implies

$$v_2\binom{T}{k}\ge v_2(T)-v_2(k)\ge v_2(T)-\lfloor\log_2 k\rfloor.$$

Every term of the remaining sum consequently has valuation at least $v_2(T)+c+1$. Its sum cannot cancel the leading term, which has exact valuation $v_2(T)+c$. This proves nonvanishing and the claimed valuation using only finite sums. The general theorem is `KomorebiPaperIX.NewtonDominance.exact_displacement`. □

Theorem 3.15 (Exact four-packet displacement law)**.** *Set*

$$(c_0,c_1,c_2,c_3)=(1,1,4,2),\qquad (d_0,d_1,d_2,d_3)=(1,1,5,3).$$

For $0\le j<4$, $n\ge 0$, and every positive integer T,

$$\begin{aligned} C_j(n+T)-C_j(n)&\neq 0, & v_2\big(C_j(n+T)-C_j(n)\big)&=v_2(T)+c_j,\\ U_j(n+T)-U_j(n)&\neq 0, & v_2\big(U_j(n+T)-U_j(n)\big)&=v_2(T)+d_j.\end{aligned}$$

Proof. The first Newton coefficients, derived from (3.3), are

$$a_{0,1}=-2,\quad a_{1,1}=-\frac{6}{5},\quad a_{2,1}=-\frac{16}{15},\quad a_{3,1}=-\frac{36}{35}.$$

Their exact valuations are c_j. Theorem 3.11 gives $v_2(a_{j,k})\ge k$ for all orders. To obtain the stronger logarithmic envelope required by Lemma 3.14, only a finite initial segment needs improvement. The recurrence proves the following exact valuations:

Address j	Further orders k	$v_2(a_{j,k})$ in that order
0	2	3
1	2	3
2	2, 3, 4, 5, 6	6, 9, 13, 16, 22
3	2, 3, 4	8, 8, 15

For all subsequent k, the elementary inequality $c_j+1+\lfloor\log_2 k\rfloor\le k$ completes the envelope. The entries follow by exact rational computation from the defining recurrence, and their valuations are proved in Lean. Together with the inequality for all subsequent orders, they establish the bound at every order. The uniform envelope is `KomorebiPaperIX.DyadicFourPrefix.tail_precision`.

Apply Lemma 3.14 to C_j. Multiplication by the nonzero factorial $j!$ adds $v_2(j!)$ and preserves nonzeroness. Hence $d_j=c_j+v_2(j!)$, giving the four valuation gains for the original coefficients. The declarations are `KomorebiPaperIX.DyadicFourLeastPeriods.normalized_exact_displacement` and `KomorebiPaperIX.DyadicFourLeastPeriods.exact_displacement`. □

Corollary 3.16 (All positive periods and their least element)**.** *For $r \in \mathbb{N}$ and $0 \le j < 4$, a positive integer T is a packet period of U_j modulo 2^r if and only if*

$$2^{\max(0,r-d_j)} \mid T.$$

Here a packet period means $U_j(n+T) \equiv U_j(n) \pmod{2^r}$ for every $n \ge 0$. Consequently the least positive packet periods are

j	0	1	2	3
least period	$2^{\max(0,r-1)}$	$2^{\max(0,r-1)}$	$2^{\max(0,r-5)}$	$2^{\max(0,r-3)}$.

Proof. By the exact displacement law, the congruence at any row is equivalent to $v_2(T) + d_j \ge r$. This is exactly the displayed integer divisibility criterion. Its least positive solution is the indicated power of two, which also covers $r = 0$ and $r \le d_j$. The all-period and least-element statements are `KomorebiPaperIX.DyadicFourLeastPeriods.period_iff_dvd` and `KomorebiPaperIX.DyadicFourLeastPeriods.least_packet_period`. □

For example, $U_2(n) \equiv 2 \pmod{32}$ and $U_3(n) \equiv 6 \pmod{8}$ for every row. At precision 2^{10} the least packet periods are $512, 512, 32, 128$. A packet displacement T has original degree displacement $4T$, as proved by `KomorebiPaperIX.DyadicFourLeastPeriods.original_degree_exact_displacement`. Minimality here concerns packet shifts. Shifts of the full degree-indexed sequence, including its zero coefficients off the packet support, require a separate analysis. The new strengthening chapters determine all repeated-step higher differences in addresses zero and one and a power-of-two-order family in address two. The full exact classification of arbitrary higher-order differences is also a separate question from the first-displacement law.

3.6 Classical odd-length coordinates and their congruence ladder

The classical signed Euler numbers have generating function $\operatorname{sech} t = \sum_{n\ge 0} E_{2n} t^{2n}/(2n)!$. They are the $N = 2$, $j = 0$ member of (3.1). The coordinate theorem reconstructs them from finite alternating sums of odd powers.

Definition 3.17 (Odd-length source and inverse coefficients)**.** For a positive odd integer p, define

$$S_d(p) = \sum_{a=0}^{p-1} (-1)^a (2a+1)^d.$$

Let γ_m be the integral Hurwitz coefficients of sech^2:

$$\operatorname{sech}^2 t = \sum_{m\ge 0} \gamma_m \frac{t^{2m}}{(2m)!}, \qquad \gamma_m = \sum_{b=0}^{2m} \binom{2m}{b} E_b E_{2m-b},$$

where $E_b = 0$ for odd b. Then $\gamma_0 = 1$. Define

$$T^{\mathrm{odd}}_{p,M}(n) = \sum_{m=0}^{\min(M,n)} \binom{2n}{2m} \gamma_m p^{2m} S_{2(n-m)}(p).$$

Here p need only be odd. Primality is needed only to interpret the result as a prime-adic congruence ladder.

The source and inverse coefficients are `KomorebiPaperIX.OddPower.source` and `KomorebiPaperIX.EulerCoefficients.inverseEven`. The truncation is `KomorebiPaperIX.OddEuler.truncation`.

The same inverse coefficients can be written in the signed zigzag notation of the original manuscript. Define Z_r by

$$\sec t + \tan t = \sum_{r\ge 0} Z_r \frac{t^r}{r!}.$$

Then

$$\gamma_m = (-1)^m Z_{2m+1} \qquad (m \in \mathbb{N}). \tag{3.12}$$

Indeed, the odd coefficients come from $\tan t$, whose derivative is $\sec^2 t$. Changing the sign of the squared variable in the even series for $\operatorname{sech}^2 t$ gives $\sec^2 t$, so its degree-$2m$ Hurwitz coefficient is $(-1)^m\gamma_m$. Differentiation identifies this coefficient with Z_{2m+1}, proving the formula. The Lean bridge is `KomorebiPaperIX.Zigzag.inverse_is_signed_zigzag`. It uses the formal generating-function definition of the zigzag numbers.

Theorem 3.18 (Classical odd-length reconstruction)**.** *For every positive odd integer p and every $n \in \mathbb{N}$,*

$$E_{2n} = \sum_{m=0}^{n} \binom{2n}{2m} \gamma_m p^{2m} S_{2(n-m)}(p).$$

Equivalently, $E_{2n} = \sum_{q=0}^{n} V_{nq} S_{2q}(p)$ for the lower triangular matrix

$$V_{nq} = \binom{2n}{2q} \gamma_{n-q} p^{2(n-q)} \quad (q \le n), \qquad V_{nq} = 0 \quad (q > n).$$

It has $V_{nn} = 1$ and $p^{2(n-q)} \mid V_{nq}$ when $q \le n$.

Proof. The exponential generating series of the source is the finite sum

$$\mathcal{S}_p(t) = \sum_{a=0}^{p-1} (-1)^a e^{(2a+1)t}.$$

Multiplication by $\cosh t$ telescopes. Since p is odd,

$$\mathcal{S}_p(t) = \frac{1 + e^{2pt}}{2} \operatorname{sech} t.$$

Taking its even part gives

$$\sum_{n\ge 0} S_{2n}(p) \frac{t^{2n}}{(2n)!} = \cosh^2(pt) \operatorname{sech} t.$$

Multiplication by the reciprocal $\operatorname{sech}^2(pt)$ yields $\operatorname{sech} t$. The EGF product rule now gives the reconstruction coefficient by coefficient. Replacing m by $n - q$ produces the displayed lower triangular row. Its diagonal and weights follow from $\gamma_0 = 1$ and the explicit factor $p^{2(n-q)}$. The source-series identity is `KomorebiPaperIX.OddPower.odd_source_even_egf`. The reconstruction and row statement are `KomorebiPaperIX.OddEuler.odd_euler_reconstruction` and `KomorebiPaperIX.OddEuler.row_action`. □

Corollary 3.19 (Odd-prime truncation ladder)**.** *For every positive odd integer p and all $M, n \in \mathbb{N}$,*

$$p^{2M+2} \mid E_{2n} - T^{\mathrm{odd}}_{p,M}(n) \quad \text{in } \mathbb{Z}.$$

If $M \geq n$, the truncation equals E_{2n} exactly. In particular, for an odd prime p this is a congruence modulo p^{2M+2} in $\mathbb{Z}_p$.

Proof. Every omitted summand has $m \geq M+1$, hence contains the integer factor p^{2m} divisible by p^{2M+2}. All its other factors are integral. The finite sum of these omitted terms proves the assertion, with the empty-tail case giving exact equality. These are `KomorebiPaperIX.OddEuler.odd_euler_supercongruence` and `KomorebiPaperIX.OddEuler.full_depth_exact`. □

The reconstruction and its truncation bound already hold over the integers. For the dyadic construction we use a different source. Its original bound of 2^{2M+2} follows from integral reciprocal coefficients, and the next subsection sharpens that bound to the optimal uniform exponent $4M+3$.

3.7 The optimal classical dyadic edge tower

Return now to the classical signed Euler numbers

$$\operatorname{sech} t = \frac{1}{\cosh t} = \sum_{n\geq 0} E_{2n} \frac{t^{2n}}{(2n)!}.$$

Thus $E_{2n} = E_{2n}^{(2,0)}$ in the convention (3.1). In particular $E_0 = 1$, $E_2 = -1$, and $E_4 = 5$. The following construction concerns these $N = 2$ coefficients, not $C_{4,0}$.

Definition 3.20 (Dyadic truncation)**.** Define β_m by the formal unit-series identity

$$\sum_{m\geq 0} \beta_m z^{2m} = \Big(\frac{z}{\sinh z}\Big)^2, \qquad \beta_0 = 1,$$

where the quotient means the inverse of the unit series $(\sinh z/z)^2$. Define

$$A(n) = \frac{3^{2n+2}-1}{4(2n+1)(2n+2)}, \qquad S_m(n) = \beta_m 2^{2m} \frac{(2n)!}{(2n-2m)!} A(n-m) \quad (0 \leq m \leq n),$$

and

$$T_M(n) = \sum_{m=0}^{\min(M,n)} S_m(n) \qquad (M, n \in \mathbb{N}). \tag{3.13}$$

These definitions correspond to `KomorebiPaperIX.Dyadic.beta`, `KomorebiPaperIX.Dyadic.sourceEven`, and `KomorebiPaperIX.DyadicReconstruction.truncation`. The formal summand uses a falling factorial, and `KomorebiPaperIX.DyadicReconstruction.falling_factorial_fraction` proves its equality with the quotient displayed above.

Lemma 3.21 (Exact reconstruction and source integrality)**.** *For every $n \in \mathbb{N}$,*

$$E_{2n} = \sum_{m=0}^{n} S_m(n).$$

Every β_m is 2-adically integral and every $A(n)$ is a nonzero 2-adic unit. If $M \geq n$, then $T_M(n) = E_{2n}$.

Proof. The ordinary kernel coefficients of $(\sinh z/z)^2$, in z^2, are

$$\frac{2^{2m+1}}{(2m+2)!}.$$

Legendre's strict factorial bound $v_2((2m+2)!) < 2m+2$ makes them integral. Their constant coefficient is one. The coefficient recurrence for its reciprocal therefore proves integrality of every β_m. For the source factor, lifting the exponent at the even power $2n+2$ gives

$$v_2(3^{2n+2}-1) = 2 + v_2(2n+2).$$

Its denominator has the same valuation, since $2n+1$ is odd. Thus $A(n)$ is a nonzero unit.

Its exponential generating function is

$$\mathcal{A}(t) = \sum_{n\geq 0} A(n)\frac{t^{2n}}{(2n)!} = \frac{\cosh(3t)-\cosh t}{4t^2}.$$

The quotients by powers of t are well defined because the required initial coefficients vanish. The hyperbolic identity

$$\mathcal{A}(t)\cosh t = \left(\frac{\sinh(2t)}{2t}\right)^2$$

then gives the exact formal-series product

$$\left(\sum_{m\geq 0}\beta_m(2t)^{2m}\right)\mathcal{A}(t) = \operatorname{sech} t.$$

Taking the factorial-scaled coefficient of t^{2n} produces the stated finite sum. Full depth leaves no omitted terms. The relevant declarations are `KomorebiPaperIX.Dyadic.beta_integral`, `KomorebiPaperIX.Dyadic.source_valuation`, `KomorebiPaperIX.DyadicSeries.dyadic_reconstruction_series`, and `KomorebiPaperIX.DyadicReconstruction.dyadic_reconstruction`. □

Lemma 3.22 (Exact weighted inverse coefficient)**.** *Let $\kappa_m = \beta_m 2^{2m}(2m)!$. For every $m \geq 1$,*

$$\kappa_m = -2^{4m}(2m-1)B_{2m} \neq 0, \qquad v_2(\kappa_m) = 4m-1,$$

where B_{2m} is the usual even Bernoulli number.

Proof. Let $Q(z) = z\coth z$, the regular formal series with constant coefficient one. The elementary differential identity

$$\left(\frac{z}{\sinh z}\right)^2 = Q(z) - zQ'(z)$$

and the Bernoulli expansion

$$Q(z) = 1 + \sum_{m\geq 1}\frac{2^{2m}B_{2m}}{(2m)!}z^{2m}$$

give $\beta_m = -2^{2m}(2m-1)B_{2m}/(2m)!$. Multiplication by $2^{2m}(2m)!$ proves the identity for κ_m. The Bernoulli denominator theorem gives $v_2(B_{2m}) = -1$ for $m > 0$: its denominator contains each prime q with $(q-1) \mid 2m$ exactly once, including 2[3]. Since $2m-1$ is odd, the valuation is exactly $4m-1$. This also proves nonvanishing. In Lean, the hyperbolic identity is proved after clearing removable zeros.

The library Bernoulli theorem then determines the valuation of κ_m. The declarations are `KomorebiPaperIX.DyadicSharpTower.kappa_bernoulli` and `KomorebiPaperIX.DyadicSharpTower.kappa_valuation`. □

Theorem 3.23 (Optimal uniform dyadic tower). *For all $M, n \in \mathbb{N}$,*

$$E_{2n} - T_M(n) \in 2^{4M+3}\mathbb{Z}_2. \tag{3.14}$$

At the first omitted row $n = M + 1$, the error is nonzero and

$$v_2\big(E_{2M+2} - T_M(M+1)\big) = 4M + 3.$$

Thus the exponent $4M + 3$ cannot be increased uniformly in the Euler row.

Proof. Rewrite the summand exactly as

$$S_m(n) = \kappa_m \binom{2n}{2m} A(n-m).$$

For $m > M$, its first factor is divisible by 2^{4m-1}, the binomial coefficient is integral, and $A(n-m)$ is a unit. Hence every omitted summand is divisible by 2^{4M+3}. Summing the finite tail proves (3.14). If $M \geq n$, the tail is empty and the statement includes the resulting zero error.

At $n = M + 1$ there is exactly one omitted summand. Here $\binom{2n}{2n} = 1$ and $A(0) = 1$, so the error is κ_{M+1}. Lemma 3.22 gives its exact valuation $4M + 3$ and nonvanishing. The statements are `KomorebiPaperIX.DyadicSharpTower.dyadic_supercongruence_sharp`, `KomorebiPaperIX.DyadicSharpTower.first_omitted_error`, and `KomorebiPaperIX.DyadicSharpTower.first_omitted_error_valuation`. □

Proposition 3.24 (Row-dependent summand weights). *For $1 \leq m \leq n$, the summand is nonzero and*

$$v_2(S_m(n)) = 4m - 1 + v_2\binom{2n}{2m}.$$

More generally, if $L \in \mathbb{N}$ satisfies

$$L \leq 4m - 1 + v_2\binom{2n}{2m} \qquad \text{for every } M < m \leq n,$$

then $E_{2n} - T_M(n) \in 2^L\mathbb{Z}_2$.

Proof. Use the same factorization of $S_m(n)$, the nonzero binomial coefficient for $m \leq n$, the exact valuation of κ_m, and the unit property of $A(n-m)$. The tail conclusion follows by addition in $2^L\mathbb{Z}_2$. These are `KomorebiPaperIX.DyadicSharpTower.summand_valuation` and `KomorebiPaperIX.DyadicSharpTower.dyadic_supercongruence_row_weight`. □

The minimum summand weight bounds the tail valuation from below, but cancellation may raise that valuation. Appendix A gives a later exact factorization and finite-polynomial reduction for each fixed depth. A universal explicit root and simplicity classification for that factor remains open. Uniform optimality follows from the equality already established at $n = M + 1$.

3.8 Isometric continuation of the four dyadic packets

The exact displacement law also determines the images of the packets after continuous extension to the dyadic integers.

Theorem 3.25 (Isometric packet bijections)**.** *For $0 \le j < 4$, define on natural numbers*

$$f_j(n) = \frac{U_j(n) - j!}{2^{d_j}}, \qquad (d_0, d_1, d_2, d_3) = (1, 1, 5, 3).$$

There is a unique continuous map $\Phi_j : \mathbb{Z}_2 \to \mathbb{Z}_2$ whose value at every $n \in \mathbb{N}$ equals the embedded rational $f_j(n)$. It is an isometric bijection and $\Phi_j(0) = 0$:

$$|\Phi_j(x) - \Phi_j(y)|_2 = |x - y|_2 \qquad (x, y \in \mathbb{Z}_2).$$

In particular, every element of $\mathbb{Z}_2$ has exactly one preimage.

Proof. For $n > 0$, the displacement law at row zero gives $v_2(U_j(n) - j!) = v_2(n) + d_j$, so $f_j(n) \in \mathbb{Z}_2$. The value at zero is zero. Applying the same law to the smaller of two unequal natural numbers shows

$$|f_j(n) - f_j(m)|_2 = |n - m|_2.$$

The natural numbers are dense in $\mathbb{Z}_2$, and the target is complete. The map therefore extends continuously. The distance identity extends on the dense product of natural-number points, making the extension an isometry. Density also proves uniqueness among all continuous maps with these prescribed values.

To prove surjectivity, use compactness. For an isometry g of a compact metric space into itself, the sequence of iterates $g^n(x)$ has a convergent subsequence. Two sufficiently close iterates, with exponents $a < b$, yield $d(x, g^{b-a}(x))$ equally small after removing a isometric iterations. Since $b - a > 0$, the latter point is in the image of g. Thus the image is dense. It is also compact and hence closed, so it is the whole space. Apply this to Φ_j on $\mathbb{Z}_2$.

The general ingredients are `KomorebiPaperIX.DensePadicIsometry.existsUnique_extensionEquiv` and `KomorebiPaperIX.CompactSelfIsometry.surjective`. The packet specialization is `KomorebiPaperIX.DyadicFourIsometry.existsUnique_isometry`. Uniqueness among continuous maps is `KomorebiPaperIX.DyadicFourIsometry.unique_continuous`. □

Corollary 3.26 (Exact image of the original coefficients)**.** *The continuous extension of the original sequence is*

$$\widetilde{U}_j(x) = j! + 2^{d_j}\Phi_j(x).$$

It is injective, its image is exactly $j! + 2^{d_j}\mathbb{Z}_2$, and

$$|\widetilde{U}_j(x) - \widetilde{U}_j(y)|_2 = 2^{-d_j}|x - y|_2.$$

Every point of that affine image has exactly one preimage in $\mathbb{Z}_2$.

Proof. The formula agrees with U_j on every natural number. Continuity and the distance equality follow from the corresponding properties of Φ_j. Its surjectivity gives precisely the displayed affine image, and multiplication by the nonzero scalar 2^{d_j} preserves injectivity. The declarations are `KomorebiPaperIX.Dy`

`adicFourIsometry.originalExtension_nat`, `KomorebiPaperIX.DyadicFourIsometry.originalExtension_dist`, and `KomorebiPaperIX.DyadicFourIsometry.originalExtension_range`. □

The four rational packet sequences thus extend to bijections onto their exact dyadic images. The proof uses density, completeness, and the displacement law. The strengthening chapters supply the further factorial and Taylor-coefficient argument for analyticity, including exact radii in addresses zero, one and two. The analytic polynomial solution maps constructed later rest on separate hypotheses and a separate proof.

4 Threshold-free reciprocal transfer at prime-power section orders

The completion of Nishibuchi's conjecture fixes the requested eventual congruence. The stronger question is whether the sufficient starting row can be removed, and how much precision the difference order and the step actually provide. This section answers the first question at every prime-power section order. Subsequent sections give a factorial improvement and exact results at section order four.

Throughout these sections, the normalized source and its reciprocal are

$$G_{N,j}(t) = j! \sum_{n\ge 0} \frac{t^{Nn}}{(Nn+j)!}, \qquad F_{N,j}(t) = G_{N,j}(t)^{-1} = \sum_{n\ge 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}, \tag{4.1}$$

where $N \ge 1$ and $0 \le j < N$. In particular, $C_{N,j}(0) = 1$. The coefficients in the original convention satisfy

$$E^{(N,j)}_{Nn} = j! C_{N,j}(n), \qquad E^{(N,j)}_{d} = 0 \quad (N \nmid d). \tag{4.2}$$

Thus every valuation of a nonzero difference gains $v_p(j!)$ on returning to the original coefficients. This factor must be retained when stating optimal periods or the final congruence.

Write $Sf(n) = f(n+1)$ and $\Delta_H = S^H - 1$. A mixed difference $\Delta_{H_1} \cdots \Delta_{H_k}$ acts on the row variable n, rather than on the full Euler subscript. We use $v_p(0) = +\infty$. Orders and starting rows are nonnegative integers. Steps are positive integers unless stated otherwise. An empty mixed difference is the original value, and a zero ordinary step makes every nonempty product containing it zero.

4.1 The Hurwitz lattice and translated differential flatness

Definition 4.1 (Hurwitz integral series)**.** For a prime p, let

$$\mathcal{H}_p = \left\{ \sum_{d\ge 0} f_d \frac{t^d}{d!} : f_d \in \mathbb{Z}_p \right\} \subset \mathbb{Q}_p[[t]].$$

For $N = p^a$ and $\lambda \in \mathbb{Z}_p$, put

$$D = \frac{d}{dt}, \qquad L_\lambda = \frac{D^N - \lambda}{p}.$$

A series A is λ-flat if $L_\lambda^k A \in \mathcal{H}_p$ for every $k \ge 0$.

Products preserve $\mathcal{H}_p$ because the coefficient at factorial degree d is

$$(AB)_d = \sum_{i=0}^{d} \binom{d}{i} a_i b_{d-i}.$$

The derivative D acts by the coefficient shift $(DA)_d = a_{d+1}$. Consequently every nonnegative power of D preserves the lattice. If $A \in \mathcal{H}_p$ has unit constant term, its reciprocal belongs to the same lattice: with $B = A^{-1}$,

$$b_0 = a_0^{-1}, \qquad b_d = -a_0^{-1} \sum_{i=1}^{d} \binom{d}{i} a_i b_{d-i}. \tag{4.3}$$

If A is supported at degrees divisible by N, this recurrence also proves that B has the same support.

Lemma 4.2 (Translated product formula)**.** *For $N = p^a$, $a \geq 1$, and $\lambda, \mu \in \mathbb{Z}_p$,*

$$L_{\lambda+\mu}(AB) = (L_\lambda A)B + A(L_\mu B) + \sum_{i=1}^{N-1} c_i (D^i A)(D^{N-i} B), \qquad c_i = \frac{1}{p}\binom{N}{i} \in \mathbb{Z}. \tag{4.4}$$

Flatness types add under multiplication.

Proof. The identity $(1+X)^{p^a} = 1 + X^{p^a}$ over $\mathbb{F}_p$ shows that every interior binomial coefficient $\binom{N}{i}$ is divisible by p. Subtract $(\lambda+\mu)AB$ from the ordinary Nth derivative product formula and divide by p to obtain (4.4).

For completeness, the iterated formula is an integral multinomial formula, rather than a product rule with unspecified remainders. On the tensor of the two factors use the mutually commuting operators

$$L_\lambda \otimes 1, \quad 1 \otimes L_\mu, \quad c_i D^i \otimes D^{N-i} \quad (1 \leq i < N).$$

Applying multiplication after the kth power of their sum gives exactly $L^k_{\lambda+\mu}(AB)$. Expanding it yields terms indexed by $a, b, e_1, \ldots, e_{N-1} \geq 0$, with $a + b + \sum_i e_i = k$:

$$\frac{k!}{a!b!\prod_i e_i!} \prod_i c_i^{e_i} (D^\alpha L^a_\lambda A)(D^\beta L^b_\mu B), \quad \alpha = \sum_i i e_i, \quad \beta = \sum_i (N-i) e_i. \tag{4.5}$$

Every scalar is integral. If A and B have the stated flatness types, both factors in every term belong to $\mathcal{H}_p$, as does their product. □

Lemma 4.3 (Reciprocal transfer)**.** *If A is λ-flat and has unit constant term, then A^{-1} is $(-\lambda)$-flat.*

Proof. Set $B = A^{-1}$. Its order-zero integrality was established in (4.3). Assume by induction that $L^b_{-\lambda} B \in \mathcal{H}_p$ for $b < k$. Since $AB = 1$,

$$L^k_0(AB) = \left(\frac{D^N}{p}\right)^k 1 = 0 \qquad (k > 0).$$

Use (4.5) with $\mu = -\lambda$. Exactly one term has inverse order $b = k$: it has $a = 0$ and all $e_i = 0$, and is $AL^k_{-\lambda} B$. In every other term $b < k$, so its second factor is integral by induction and derivative-shift integrality. Its first factor is integral by the flatness of A, and its scalar is integral. Hence $AL^k_{-\lambda} B \in \mathcal{H}_p$. Multiplying by the already integral B proves $L^k_{-\lambda} B \in \mathcal{H}_p$. This induction takes place in the entire Hurwitz lattice. It places no restriction on the coefficient degree or the starting row. □

4.2 The actual Euler source

The packed factorial coefficient of $G_{N,j}$ is

$$g_{N,j}(n) = \frac{j!(Nn)!}{(Nn+j)!} = \prod_{i=1}^{j} \left(1 + \frac{N}{i} n\right)^{-1}. \tag{4.6}$$

The empty product is one. If $N = p^a$ and $1 \leq i < N$, then $v_p(i) < a$, so $N/i \in p\mathbb{Z}_p$. Every affine denominator in (4.6) is therefore a unit on $\mathbb{Z}_p$.

For $s \in p\mathbb{Z}_p$, an exact finite-difference identity is

$$\Delta_1^k \frac{1}{1+sn} = \frac{(-s)^k k!}{\prod_{h=0}^{k}(1+s(n+h))}. \tag{4.7}$$

It follows by induction, subtracting adjacent rational functions. Each successive subtraction adds the factor $-(k+1)s$ in the numerator. Its denominator is a unit, so it belongs to $p^k\mathbb{Z}_p$. The discrete Leibniz formula

$$\Delta_1^k(fg)(n) = \sum_{r=0}^{k} \binom{k}{r} (\Delta_1^r f)(n)(\Delta_1^{k-r} g)(n+r) \tag{4.8}$$

then proves $\Delta_1^k g_{N,j}(n) \in p^k\mathbb{Z}_p$ for every k, n. The coefficient at degree Nn of $L_1^k G_{N,j}$ is precisely $p^{-k}\Delta_1^k g_{N,j}(n)$, and the other residue classes vanish. Thus the actual source is 1-flat.

Theorem 4.4 (All-row prime-power transfer)**.** *Let p be prime, $a \geq 1$, $N = p^a$, and $0 \leq j < N$. The coefficients $C_{N,j}(n)$ belong to $\mathbb{Z}_p$ for all $n \geq 0$. Set $A_{N,j}(n) = (-1)^n C_{N,j}(n)$ and $T_H = S^H - (-1)^H$. For $k \geq 1$ and positive $H_1, \ldots, H_k$,*

$$v_p(\Delta_{H_1} \cdots \Delta_{H_k} A_{N,j}(n)) \geq k + \sum_{\nu=1}^{k} v_p(H_\nu), \tag{4.9}$$

$$v_p(T_{H_1} \cdots T_{H_k} E_{Nn}^{(N,j)}) \geq v_p(j!) + k + \sum_{\nu=1}^{k} v_p(H_\nu). \tag{4.10}$$

At $p = 2$, the ordinary differences of $C_{N,j}$ and of the original coefficients satisfy the same bounds. These dyadic statements also hold for $N = 1$.

Proof. Reciprocal transfer gives $L_{-1}^k F_{N,j} \in \mathcal{H}_p$. Extraction of the packed coefficient gives $(S+1)^k C_{N,j}(n) \in p^k\mathbb{Z}_p$. Moreover,

$$\Delta_1^k A_{N,j}(n) = (-1)^{n+k}(S+1)^k C_{N,j}(n),$$

so $A_{N,j}$ has the ordinary p^k difference property. At $p = 2$, $L_1 = L_{-1} - 1$. Its integral binomial expansion on $F_{N,j}$ gives $\Delta_1^k C_{N,j}(n) \in 2^k\mathbb{Z}_2$ directly. For $N = 1$ the interior sum in the product formula is empty, so the same argument applies without change. Equivalently, $F_{1,0}(t) = e^{-t}$.

To retain the valuation of a step, let B be any sequence satisfying $\Delta_1^m B \in p^m\mathbb{Z}_p$ at every order and row. Use the finite operator identity

$$\Delta_H = \sum_{i=1}^{H} \binom{H}{i} \Delta_1^i.$$

The relation $i\binom{H}{i} = H\binom{H-1}{i-1}$ implies

$$v_p\left(p^i \binom{H}{i}\right) \geq i + v_p(H) - v_p(i) \geq 1 + v_p(H).$$

Multiplying these identities for the k steps, every term of total order M has its scalar multiplied by p^M, and is divisible by $p^{k+\sum_\nu v_p(H_\nu)}$. This proves the mixed-step bound. Finally,

$$\Delta_H A_{N,j}(n) = (-1)^{n+H} T_H C_{N,j}(n).$$

Iteration contributes only a global sign. Multiplying by $j!$ proves (4.10). □

Corollary 4.5 (Threshold-free dyadic congruence)**.** *For $N = 2^a$, $a \geq 0$, all permitted j, and $n \geq 0$,*

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} E_{Nn}^{(N,j)}) \geq v_2(j!) + k + \sum_{\nu=1}^{k} v_2(H_\nu). \tag{4.11}$$

In particular, Δ_H^k gains $k(1 + v_2(H))$ bits before normalization. For $r > v_2(j!)$, the row displacement $H = 2^{\max(0, r - v_2(j!) - 1)}$ is a sufficient period modulo 2^r from row zero. If $r \leq v_2(j!)$, all original coefficients vanish modulo 2^r.

Remark 4.6 (Necessary boundaries). At an odd prime the ordinary step-one difference cannot replace the sign-twisted operator in the general theorem. For $j = 0$, $C_{N,0}(0) = 1$ and $C_{N,0}(1) = -1$, so their ordinary difference is -2, a unit at every odd prime. The proof also does not extend unchanged to non-prime-power N, whose interior binomial coefficients need not be divisible by the chosen prime. Neither observation is a counterexample to Nishibuchi's original eventual congruence.

5 Factorial precision at section order four

Write $C_j = C_{4,j}$ and $F_j = F_{4,j}$. The all-row theorem removes the threshold, but at $N = 4$ it leaves a further factorial gain unused. The exact algebra below proves that gain for all four addresses and identifies several infinite families where it is sharp.

Theorem 5.1 (Four-address factorial bound)**.** *For $j \in \{0, 1, 2, 3\}$, $n \geq 0$, $k \geq 1$, and positive mixed steps,*

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} C_j(n)) \geq k + v_2(k!) + \sum_{\nu=1}^{k} v_2(H_\nu) + \beta_j, \quad (\beta_0, \beta_1, \beta_2, \beta_3) = (0, 0, 3, 1). \tag{5.1}$$

Address two also satisfies

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} C_2(n)) \geq 2k + v_2(k!) + \left\lfloor \frac{k}{2} \right\rfloor + \sum_{\nu=1}^{k} v_2(H_\nu). \tag{5.2}$$

For the original coefficients, the offsets in (5.1) are $(0, 0, 4, 2)$. No positive-order offset is asserted for the empty difference.

The proof of the first two addresses uses one classical arithmetic input: Frobenius's exact congruence for the ordinary Euler numbers in the sech convention, as stated and derived by Gessel [2], Section 9. The address-two and address-three bounds below are finite Bernoulli calculations and do not require that input. A divided-power argument is necessary: ordinary integral moments alone do not justify division by $k!$.

5.1 Euler moments and an integral divided-power algebra

Let $\mathcal{L}$ be the ordinary Euler evaluation functional on a variable $\mathcal{E}$, defined by

$$\mathcal{L}(\mathcal{E}^d) = \mathrm{Euler}_d, \qquad \sum_{d \geq 0} \mathrm{Euler}_d \, \frac{t^d}{d!} = \operatorname{sech} t.$$

The functional is linear. It is not a multiplicative evaluation. Independent variables mean that moments of polynomials in separate variables factor. Odd moments vanish. Frobenius's congruence is

$$v_2\left(\sum_{r=0}^{m}(-1)^{m-r}\binom{m}{r}\operatorname{Euler}_{2n+2r}\right)=m+v_2(m!)\qquad(n,m\geq 0).\tag{5.3}$$

For $m=0$, the assertion is that every even Euler number is odd. With $P=(\mathcal{E}^2-1)/2$ and $\gamma_m(P)=P^m/m!$, the case $n=0$ says

$$\mathcal{L}(\gamma_m(P))\in\mathbb{Z}_2^{\times}\qquad(m\geq 0).\tag{5.4}$$

Introduce an independent Euler variable $\mathcal{F}$ and $Q=(\mathcal{F}^2-1)/2$.

Put $K=\mathbb{Q}_2(i)$, $R=\mathbb{Z}_2[i]$, $a=(1+i)/2$, and $b=(1-i)/2$. The exact identity

$$F_0(t)=\frac{2}{\cosh t+\cos t}=\operatorname{sech}(at)\operatorname{sech}(bt)\tag{5.5}$$

gives, for $U=a\mathcal{E}+b\mathcal{F}$, $\mathcal{L}(U^{4n})=C_0(n)$. Although a and b have negative valuation, their denominators disappear from the following exact polynomial identities:

$$\begin{aligned}U^2&=\mathcal{E}\mathcal{F}+i(P-Q),\\ H_0:=\frac{U^4-1}{2}&=P+Q-\frac{P^2}{2}-\frac{Q^2}{2}+3PQ+i\mathcal{E}\mathcal{F}(P-Q).\end{aligned}\tag{5.6}$$

Let $\Gamma_R(P,Q)$ be the R-subalgebra generated by all $\gamma_m(P)$ and $\gamma_r(Q)$. Adjoin $\mathcal{E},\mathcal{F}$ with

$$\mathcal{E}^2=1+2P,\qquad\mathcal{F}^2=1+2Q,$$

and call the resulting algebra $\mathcal{A}_0$. Its elements are finite expressions. No completion is being presumed. It is generated as a $\Gamma_R(P,Q)$-module by $1,\mathcal{E},\mathcal{F},\mathcal{E}\mathcal{F}$. The functional maps this module to R. For an even basis term this is (5.4) and independence. For a basis term with an odd Euler factor the moment is zero, since P and Q are even polynomials in their respective variables.

Every displayed component of H_0 has all its divided powers in $\mathcal{A}_0$. Besides the generators, the needed identities are

$$\begin{aligned}\gamma_m(P^2/2)&=\frac{(2m)!}{2^m m!}\gamma_{2m}(P),\\ \gamma_m(3PQ)&=3^m m!\gamma_m(P)\gamma_m(Q),\\ \gamma_m(P-Q)&=\sum_{r=0}^{m}(-1)^r\gamma_{m-r}(P)\gamma_r(Q).\end{aligned}\tag{5.7}$$

The first scalar counts partitions into pairs and is an integer. For the last term of H_0, multiply the third identity by $(i\mathcal{E}\mathcal{F})^m$ and reduce the Euler powers using the integral quadratic relations. Finally,

$$\gamma_k(X+Y)=\sum_{r=0}^{k}\gamma_r(X)\gamma_{k-r}(Y)$$

proves $\gamma_k(H_0)\in\mathcal{A}_0$. Applying the functional and putting $W=U^4-1=2H_0$ gives

$$\mathcal{L}(W^m)\in 2^m m!R\qquad(m\geq 0).\tag{5.8}$$

These moments are rational. The identity $R \cap \mathbb{Q}_2 = \mathbb{Z}_2$, obtained from the linear independence of $1, i$ over $\mathbb{Q}_2$, descends every such divisibility statement to $\mathbb{Z}_2$. No field property of $R/2R$ is needed.

For every starting row,

$$\Delta_1^k C_0(n) = \mathcal{L}(U^{4n} W^k) = \sum_{s=0}^{n} \binom{n}{s} \mathcal{L}(W^{k+s}). \tag{5.9}$$

Each summand lies in $2^k k! \mathbb{Z}_2$, proving the factorial bound for address zero from row zero onward.

5.2 A finite step-expansion lemma

The conversion from unit steps to arbitrary mixed steps is used several times, so it is useful to make it explicit. Write $\Delta = \Delta_1$. For $H > 0$,

$$\Delta_H = \sum_{m=1}^{H} \binom{H}{m} \Delta^m. \tag{5.10}$$

A mixed term with local orders $m_1, \dots, m_k \geq 1$ and $M = \sum_\nu m_\nu$ therefore has valuation at least

$$M + v_2(M!) + \beta + \sum_{\nu} \bigl(v_2(H_\nu) - v_2(m_\nu) \bigr)$$

if the unit-order estimate has offset β for every positive order. The inequalities

$$m - 1 \geq v_2(m), \qquad M \geq k, \qquad v_2(M!) \geq v_2(k!)$$

make this at least $k + v_2(k!) + \beta + \sum_\nu v_2(H_\nu)$. Every expansion is finite. This proves the desired mixed-step transfer without an exchange of limits.

5.3 An elementary Bernoulli denominator bound

Let B_m denote the Bernoulli numbers with $B_1 = -1/2$. The identity obtained by substituting $u = e^t - 1$ into $\log(1+u)/u$ is

$$B_m = \sum_{r=0}^{m} (-1)^r \frac{r!}{r+1} \left\{ {m \atop r} \right\}, \tag{5.11}$$

where the braces are Stirling numbers of the second kind. Each $2r!/(r+1)$ is dyadically integral. Indeed, if $s = v_2(r+1) > 0$, then $r \geq 2^s - 1$ and

$$v_2(r!) \geq v_2((2^s - 1)!) = 2^s - 1 - s \geq s - 1.$$

The last inequality is $2^s \geq 2s$. The case $s = 0$ is immediate. Consequently

$$2B_m \in \mathbb{Z}_2 \qquad (m \geq 0). \tag{5.12}$$

Thus the transfer factors below require no appeal to the von Staudt–Clausen theorem.

5.4 Address one and the scaled-binomial transfer

The address-one source is $G_{4,1}(t) = (\sinh t + \sin t)/(2t)$. The identities $a+b=1$, $a-b=i$, and $ab=1/2$ give

$$D_1(t) := G_{4,1}(t)F_0(t) = \frac{1}{2}\left(\frac{\tanh(at)}{at} + \frac{\tanh(bt)}{bt}\right). \tag{5.13}$$

For example, its numerator follows from

$$\sinh t + \sin t = 2b\sinh(at)\cosh(bt) + 2a\cosh(at)\sinh(bt),$$

while $\cosh t + \cos t = 2\cosh(at)\cosh(bt)$. Only degrees divisible by four survive. Its actual Hurwitz packets are

$$D_1(t) = \sum_{i\ge 0} d_i \frac{t^{4i}}{(4i)!}, \qquad d_i = \left(-\frac{1}{4}\right)^i \frac{2^{4i+2}(2^{4i+2}-1)B_{4i+2}}{(4i+2)(4i+1)}. \tag{5.14}$$

The constant is one. Since $v_2(4i+2)=1$, the elementary Bernoulli bound shows $v_2(d_i)\ge 2i$. Hurwitz inversion therefore gives

$$D_1(t)^{-1} = \sum_{i\ge 0} h_i \frac{t^{4i}}{(4i)!}, \qquad h_0 = 1, \quad v_2(h_i)\ge 2i.$$

This follows by induction from the inverse recurrence: for packet index i, every product $d_r h_{i-r}$ has valuation at least $2i$, and its binomial multiplier is integral. Because $F_1 = F_0 D_1^{-1}$,

$$C_1(n) - C_0(n) = \sum_{i\ge 1} h_i \binom{4n}{4i} C_0(n-i). \tag{5.15}$$

At a natural n this is a finite sum, with $i \le n$.

Here is the polynomial estimate that supplies one additional bit. For $i \ge 1$, put $P_i(x) = 4^i\binom{4x}{4i}$. Its constant is zero, and its degree-d coefficient is

$$[x^d]P_i(x) = \frac{4^i 4^d s(4i,d)}{(4i)!}, \tag{5.16}$$

where $s(m,d)$ is a signed Stirling number of the first kind. In each product contributing to $s(4i,d)$, at most $d-1$ of the $2i-1$ even integers in $1,\dots,4i-1$ can be omitted. Thus

$$v_2(s(4i,d)) \ge \max(0, 2i-d).$$

Writing $s_2(i)$ for the number of ones in the binary expansion of i, and using $v_2((4i)!) = 4i - s_2(i)$, gives

$$v_2([x^d]P_i) \ge 2d - 2i + s_2(i) + \max(0,2i-d) \ge d + s_2(i) \ge d+1 \quad (d\ge 1). \tag{5.17}$$

For $k\ge 1$, $\Delta_1^k x^d$ vanishes when $d<k$ and is an integer polynomial with values divisible by $k!$. To see the latter, expand $x^d = \sum_r \left\{ {d \atop r} \right\} r!\binom{x}{r}$ and apply $\Delta_1^k\binom{x}{r} = \binom{x}{r-k}$. It follows that

$$\Delta_1^k P_i(x) \in 2^{k+1}k!\mathbb{Z}_2 \qquad (k\ge 0,\ x\in\mathbb{Z}_2). \tag{5.18}$$

For $k=0$ the bound follows directly from the coefficient estimate.

To apply discrete products in (5.15) without a moving truncation, first extend C_0 to $\mathbb{Z}_2$ by its uniformly convergent Newton series. The all-row bound $v_2(\Delta_1^m C_0(0)) \geq m$ proves this convergence. Its factorial bounds extend to every dyadic starting point by continuity and approximation with natural rows. The sum in (5.15) is then uniformly convergent on $\mathbb{Z}_2$: both the binomial polynomial and $C_0(x-i)$ are integral-valued and $v_2(h_i) \geq 2i$.

For the pointwise product $P_i(x)C_0(x-i)$, the discrete Leibniz formula has scalar divisibility

$$\binom{k}{r}(2^{r+1}r!)(2^{k-r}(k-r)!) = 2^{k+1}k!.$$

Multiplication by $h_i/4^i \in \mathbb{Z}_2$ and uniform summation now give

$$v_2(\Delta_1^k(C_1 - C_0)(n)) \geq k + v_2(k!) + 1 \qquad (k, n \geq 0). \tag{5.19}$$

The same extra bit holds for positive mixed steps by (5.10). Combining this with the address-zero bound proves Theorem 5.1 for address one.

5.5 Central factorials and integral Bernoulli divided powers

Lemma 5.2 (Bernoulli divided-power moments)**.** *Let Z be the ordinary Bernoulli variable, $\mathcal{L}(Z^r) = B_r$, with exponential transform $t/(e^t-1)$. For $P = Z(Z+1)$ and $\mu_m = \mathcal{L}(P^m/m!)$,*

$$\mu_m \in \mathbb{Z}_2, \qquad v_2(\mu_m) \geq s_2(m) - 1 \quad (m \geq 1). \tag{5.20}$$

Moreover, μ_m is odd exactly when $m = 0$ or m is a positive power of two.

Proof. Define the central-factorial basis polynomials

$$P_r(X) = \prod_{a=0}^{r-1}(X - a(a+1)).$$

At $X = Z(Z+1)$, they have the exact factorization

$$P_r(Z(Z+1)) = \prod_{a=0}^{r-1}(Z-a)(Z+a+1) = (2r)!\binom{Z+r}{2r}. \tag{5.21}$$

The Bernoulli transform gives $\mathcal{L}((1+T)^Z) = \log(1+T)/T$, whence $\mathcal{L}(\binom{Z}{j}) = (-1)^j/(j+1)$. Finite Vandermonde expansion therefore gives

$$\begin{aligned}\mathcal{L}\left(\binom{Z+r}{2r}\right) &= (-1)^r \sum_{q=0}^{r}(-1)^q\binom{r}{q}\frac{1}{r+q+1} \\ &= (-1)^r\frac{(r!)^2}{(2r+1)!}.\end{aligned} \tag{5.22}$$

The last equality is the rational identity obtained by expanding $\int_0^1 t^r(1-t)^r\,dt$. This use of an ordinary finite integral makes no assertion about a positive or bounded p-adic measure. In particular,

$$\mathcal{L}(P_r(P)) = (-1)^r\frac{(r!)^2}{2r+1}.$$

Expand $X^m = \sum_{r=0}^{m} T(m,r)P_r(X)$. Since $XP_r(X) = P_{r+1}(X) + r(r+1)P_r(X)$, these integer coefficients satisfy

$$T(m+1,r) = T(m,r-1) + r(r+1)T(m,r), \quad T(0,0) = 1. \tag{5.23}$$

Every $r(r+1)$ is even. Induction, including the boundary indices, proves $2^{m-r} \mid T(m,r)$ for $0 \le r \le m$. For $m > 0$ the $r = 0$ term vanishes. Each remaining summand of $\mathcal{L}(P^m/m!)$ has valuation at least

$$m - r + 2v_2(r!) - v_2(m!) = r + s_2(m) - 2s_2(r) \ge s_2(m) - 1 \ge 0. \tag{5.24}$$

Here $r \ge 2s_2(r) - 1$ follows from $r \ge 2^{s_2(r)} - 1$ and $2^u \ge 2u$ for $u \ge 1$. This proves integrality and the stronger bound.

If m is not a power of two, $s_2(m) \ge 2$ makes μ_m even. If m is a positive power of two, the only central indices which can contribute an odd residue are $r = 1$ and $r = 3$, since $r - 2s_2(r) = -1$ only at these two positive integers. The $r = 1$ term is odd: $T(m,1) = 2^{m-1}$ and $v_2(m!) = m - 1$. For $m \ge 4$ a power of two, the $r = 3$ term is even. More explicitly,

$$\frac{T(m,3)}{2^{m-3}} = h_{m-3}(1,3,6), \qquad \sum_{d \ge 0} h_d(1,3,6)t^d = \frac{1}{(1-t)(1-3t)(1-6t)}.$$

Reduction modulo two gives $(1-t)^{-2}$, whose degree $m-3$ coefficient has parity $m - 2 = 0$. At $m = 1, 2$ the $r = 3$ term is absent, and the $r = 1$ term is again odd. Finally $\mu_0 = 1$. □

5.6 Address two and its stronger filtration

The variable $\mathcal{B} = 2Z + 1$ has transform

$$e^t \frac{2t}{e^{2t} - 1} = \frac{t}{\sinh t}.$$

Its odd moments vanish, and $P = (\mathcal{B}^2 - 1)/4 = Z(Z+1)$ has the divided-power moments of Lemma 5.2. Introduce an independent copy $\mathcal{D}$ and $Q = (\mathcal{D}^2 - 1)/4$. The exact factorization

$$F_2(t) = \frac{t^2}{\cosh t - \cos t} = \frac{at}{\sinh(at)} \frac{bt}{\sinh(bt)} \tag{5.25}$$

shows that $U = a\mathcal{B} + b\mathcal{D}$ satisfies $\mathcal{L}(U^{4n}) = C_2(n)$. Direct expansion gives

$$H_2 := \frac{U^4 - 1}{4} = P + Q - P^2 - Q^2 + 6PQ + i\mathcal{B}\mathcal{D}(P - Q). \tag{5.26}$$

Let $\mathcal{A}_2$ be the divided-power algebra over R generated by P, Q, adjoining $\mathcal{B}, \mathcal{D}$ with $\mathcal{B}^2 = 1 + 4P$, $\mathcal{D}^2 = 1 + 4Q$. The functional is integral on its module basis $1, \mathcal{B}, \mathcal{D}, \mathcal{B}\mathcal{D}$ by the Bernoulli lemma, independence, and vanishing of odd moments. Every component in (5.26) has integral divided powers, by the same finite identities as before. P^2 is even easier than $P^2/2$. Consequently

$$\mathcal{L}((U^4 - 1)^m) \in 4^m m! \mathbb{Z}_2, \qquad v_2(\Delta_1^k C_2(n)) \ge 2k + v_2(k!). \tag{5.27}$$

The starting-row conclusion follows from the finite base expansion (5.9), now with $W = 4H_2$.

The low Newton coefficients of the actual reciprocal are

$$C_2(0)=1,\quad C_2(1)=-\frac{1}{15},\quad C_2(2)=\frac{13}{45},\qquad a_1=-\frac{16}{15},\quad a_2=\frac{64}{45}, \tag{5.28}$$

where $a_m=\Delta_1^m C_2(0)$. Thus $v_2(a_1)=4$ and $v_2(a_2)=6$. For $m\ge 3$, (5.27) bounds a_m by $2m+v_2(m!)$. The finite shifted Newton formula

$$\Delta_1^k C_2(n)=\sum_{m=k}^{n+k} a_m\binom{n}{m-k} \tag{5.29}$$

therefore gives $v_2(\Delta_1^k C_2(n))\ge k+v_2(k!)+3$ for every $k\ge 1$. For $k\ge 3$ this is $2k\ge k+3$. For $k=1,2$ it uses the exact initial coefficients and the higher tail bound. The finite step expansion proves the offset three in (5.1).

There is a still stronger filtration. Put

$$X=P+Q,\qquad \tau=i\mathcal{B}\mathcal{D},\qquad \epsilon=1+\tau.$$

Then the exact identities are

$$H_2=\epsilon X+2G,\quad G=-\gamma_2(P)-\gamma_2(Q)+3PQ-\tau Q,\quad \epsilon^2=2V,\quad V=\tau-2X-8PQ. \tag{5.30}$$

Every divided power of G is integral component by component. The finite addition identity gives

$$\gamma_k(H_2)=\sum_{r=0}^{k}2^r\gamma_r(G)\epsilon^{k-r}\gamma_{k-r}(X)\in 2^{\lfloor k/2\rfloor}\mathcal{A}_2, \tag{5.31}$$

because the rth term belongs to $2^{r+\lfloor (k-r)/2\rfloor}\mathcal{A}_2$. Applying the functional, rational descent, and the base-index expansion proves the unit-step version of (5.2). For a mixed term of total order M, its lower bound exceeds the target at order k by at least

$$2(M-k)+\left\lfloor\frac{M}{2}\right\rfloor-\left\lfloor\frac{k}{2}\right\rfloor+v_2(M!)-v_2(k!)-\sum_{\nu}v_2(m_\nu)\ge 0.$$

Indeed $\sum_\nu v_2(m_\nu)\le M-k$. This proves the stronger mixed bound. The earlier offset-three bound is retained, since it is stronger at the lowest orders.

5.7 Address three and cancellation of the apparent pole

The address-three source is $G_{4,3}(t)=3(\sinh t-\sin t)/t^3$. Addition identities give

$$\sinh t-\sin t=2a\sinh(at)\cosh(bt)+2b\cosh(at)\sinh(bt),\qquad \cosh t-\cos t=2\sinh(at)\sinh(bt).$$

Consequently

$$D_3(t):=G_{4,3}(t)F_2(t)=\frac{3}{2}\left(\frac{\coth(at)}{at}+\frac{\coth(bt)}{bt}\right). \tag{5.32}$$

This expression is a power series: its two polar terms cancel because $a^{-2}+b^{-2}=0$. Its constant is one, only degrees divisible by four survive, and the Bernoulli expansion gives

$$D_3(t)=\sum_{i\ge 0} d_i\frac{t^{4i}}{(4i)!},\qquad d_i=3\left(-\frac{1}{4}\right)^i\frac{2^{4i+2}B_{4i+2}}{(4i+2)(4i+1)}. \tag{5.33}$$

Again $v_2(d_i)\ge 2i$, and the inverse coefficients h_i have the same bound. Since $F_3=F_2D_3^{-1}$,

$$C_3(n)-C_2(n)=\sum_{i\ge 1} h_i\binom{4n}{4i}C_2(n-i). \tag{5.34}$$

The scaled-binomial argument (5.18) applies unchanged. The basic bound for C_2 includes order zero. Its continuous extension is available from the threshold-free theorem. Thus the shifted sums converge uniformly on $\mathbb{Z}_2$, and the pointwise product rule proves

$$v_2(\Delta_1^k(C_3-C_2)(n))\ge k+v_2(k!)+1\qquad (k,n\ge 0). \tag{5.35}$$

At every positive order C_2 has three extra bits, so this gives the one-bit offset for C_3. Finite step expansion finishes the proof of Theorem 5.1.

6 Sharp higher-difference families

The lower bounds do not mean that every mixed stencil has exactly the stated valuation. Addresses zero and one have a complete classification of when the leading factorial precision is attained. Address two has exact mixed precision at every positive power-of-two order.

Theorem 6.1 (Exact factorial precision at addresses zero and one)**.** *Let $j\in\{0,1\}$, $k\ge 1$, $n\ge 0$, and $H_1,\dots,H_k>0$. Put*

$$L=k+v_2(k!)+\sum_{\nu=1}^{k} v_2(H_\nu),\qquad e=\#\{\nu:2\mid H_\nu\}.$$

The mixed difference has valuation exactly L if k is odd or e is even. If k is even and e is odd, it has valuation at least $L+1$. In particular, for every homogeneous positive step,

$$v_2(\Delta_H^k C_j(n))=k+v_2(k!)+kv_2(H). \tag{6.1}$$

Proof. Return to the Euler variables of address zero and put $X=P+Q$, $\tau=i\mathcal{E}\mathcal{F}$, $\epsilon=1+\tau$. Direct rearrangement of (5.6) gives

$$H_0=\gamma_2(X)+\epsilon X+2G,\qquad G=-\gamma_2(P)-\gamma_2(Q)+PQ-\tau Q,\qquad \epsilon^2=2(\tau-P-Q-2PQ). \tag{6.2}$$

The divided powers of G are integral by the same component formulas used above. Thus modulo $2\mathcal{A}_0$, $\gamma_k(H_0)$ equals the divided power of $\gamma_2(X)+\epsilon X$. The identity

$$\gamma_r(\gamma_2(X))=\frac{(2r)!}{2^r r!}\gamma_{2r}(X)$$

has an odd integer scalar. Every term involving ϵ^j for $j \geq 2$ vanishes modulo two. The surviving zero and first powers give

$$\gamma_k(H_0) \equiv \gamma_{2k}(X) + \epsilon\gamma_{2k-1}(X) \pmod{2\mathcal{A}_0}. \tag{6.3}$$

The multiplication coefficient in the second term is $2k-1$, also odd. By the odd moments (5.4) and independence,

$$\mathcal{L}(\gamma_m(P+Q)) \equiv m+1 \pmod 2.$$

The τ-weighted moment vanishes because it contains an odd power of each Euler variable. Therefore the first term of (6.3) evaluates to one modulo two and the second to zero. Hence $\mathcal{L}(\gamma_k(H_0))$ is odd at every order, including order zero. This proves

$$v_2(\Delta_1^k C_0(0)) = k + v_2(k!).$$

In the base-row expansion (5.9), every $s > 0$ term has at least one extra bit after division by $2^k k!$. Thus the same exact valuation holds at every starting row. The extra-bit transfer (5.19) proves it for address one as well.

It remains to track possible ties in the mixed-step expansion. For local orders $m_\nu \geq 1$ and total M, the excess over L is at least

$$\left(M - k - \sum_\nu v_2(m_\nu)\right) + \left(v_2(M!) - v_2(k!)\right). \tag{6.4}$$

The first bracket is nonnegative, and equality requires every m_ν to be one or two. Two or more occurrences of two make the second bracket positive, since the intervening factorial interval then contains an even number. Exactly one occurrence of two can tie only when k is even. Its binomial coefficient can have the minimally allowed valuation only if its associated step is even.

The all-one multi-index contributes one after normalization by 2^L. When k is even, each even step supplies exactly one further tied term, again with normalized residue one. For example, $\binom{H}{2} = H(H-1)/2$ attains $v_2(H) - 1$ exactly when H is even. All remaining factors and the exact normalized order-$k+1$ difference are odd. Thus the total normalized parity is one for odd k, and $1 + e$ for even k. This proves the classification. In the homogeneous case, e is zero or k, so it is even whenever k is even, proving (6.1). □

Remark 6.2 (A mixed cancellation). The exceptional class can gain more than one bit. The exact values

$$\Delta_1\Delta_2 C_0(0) = -33728, \qquad \Delta_1\Delta_2 C_1(0) = -\frac{150208}{585}$$

both have valuation six, whereas their baseline L is four. The theorem classifies attainment of the baseline. It does not give a complete deeper valuation in the cancellation class.

Theorem 6.3 (Exact address-two power orders)**.** *For every starting row $n \geq 0$ and all positive mixed steps,*

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} C_2(n)) = V_2(k) + \sum_{\nu=1}^{k} v_2(H_\nu) \tag{6.5}$$

whenever k is a positive power of two, where

$$V_2(1) = 4, \qquad V_2(2) = 6, \qquad V_2(k) = 2k + v_2(k!) + \frac{k}{2} \quad (k \geq 4).$$

For the original coefficients, the right side gains one bit.

Proof. For $k = 2^r \geq 4$, put $q = k/2$, which is even. Normalize the exact expansion (5.31) by 2^q before reducing modulo two. Its uth auxiliary term has excess $u + \lfloor (k-u)/2 \rfloor - q = \lfloor u/2 \rfloor$. Only auxiliary indices zero and one can therefore survive:

$$\frac{\gamma_k(H_2)}{2^q} \equiv V^q \gamma_k(X) + \epsilon V^{q-1} G \gamma_{k-1}(X) \pmod{2\mathcal{A}_2}. \tag{6.6}$$

Here $V \equiv \tau \pmod 2$ and $\tau^2 \equiv 1 \pmod 2$. Since q is even and $\epsilon\tau \equiv \epsilon \pmod 2$, this reduces to $\gamma_k(X) + \epsilon G \gamma_{k-1}(X)$. Moreover, $G \equiv \gamma_2(X) + \tau Q \pmod 2$. After evaluation, every term with a single τ vanishes, so the second contribution reduces to

$$\mathcal{L}\big((\gamma_2(X) + Q)\gamma_{k-1}(X)\big).$$

The $\gamma_2(X)$ product has scalar $\binom{k+1}{2}$, which is even because $4 \mid k$. The other contribution expands as

$$\mathcal{L}(Q\gamma_{k-1}(P+Q)) = \sum_{a=0}^{k-1} (k-a)\mu_a \mu_{k-a}. \tag{6.7}$$

An odd summand would require $k-a$ to be odd and μ_{k-a} to be odd. By Lemma 5.2, this requires $k-a = 1$. Then $a = k-1$ is not a power of two and μ_a is even. Hence the second contribution is zero modulo two.

For the first contribution, independence gives $\mathcal{L}(\gamma_k(P+Q)) = \sum_{a=0}^{k} \mu_a \mu_{k-a}$. The parity generating series of these convolution moments is

$$(1 + t + t^2 + t^4 + t^8 + \cdots)^2 = 1 + t^2 + t^4 + t^8 + \cdots \quad \text{in } \mathbb{F}_2[[t]].$$

Its coefficient is one at every $k = 2^r \geq 4$. Thus the normalized moment in (6.6) is odd and

$$v_2(\Delta_1^k C_2(0)) = 2k + v_2(k!) + k/2.$$

Inserting $U^{4n} = (1 + 4H_2)^n$ introduces only corrections with strictly higher valuation, by the filtered bound at orders $k + s$ for $s > 0$. The equality consequently holds at every starting row.

For mixed steps the all-one local multi-index gives the claimed leading term. A higher term of total order $M > k$ has excess at least

$$2(M-k) + \left\lfloor \frac{M}{2} \right\rfloor - \frac{k}{2} + v_2(M!) - v_2(k!) - \sum_{\nu} v_2(m_\nu) \geq M - k \geq 1.$$

Hence the leading term cannot cancel.

At order one, a_1 in (5.28) has valuation four and the Newton tail starts at valuation six. Its shifted value remains exact. In the step expansion a local order two can lose at most one bit through its binomial coefficient, so every correction is still above $4 + v_2(H_1)$. At order two, a_2 has valuation six and the next coefficient has lower bound eight from the stronger filtration. A total-order-three correction loses at most one bit, so it stays above the leading valuation six, while later orders gain still more. The same argument proves exactness for both mixed steps. □

Remark 6.4 (What the power orders do not decide). The exact orders in Theorem 6.3 form an infinite family. They do not supply a complete valuation formula at non-power-of-two orders in address two, nor an exact higher-difference formula at address three. Numerical patterns at other orders remain bounded observations unless proved separately.

7 Exact first differences, least periods, and dyadic images

The factorial estimates are stronger than needed for the optimal first-difference law. A general Newton criterion gives the exact law from the threshold-free bound and finitely many rational coefficients. This route also yields all addresses at ranks two and eight, and an infinite family at every larger dyadic rank.

Lemma 7.1 (A first-difference criterion). *Let $C : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_2$ have Newton coefficients $a_k = \Delta_1^k C(0)$. Suppose for an integer $b \geq 0$ that*

$$v_2(a_1) = b, \qquad v_2(a_k) \geq b + 1 + \lfloor \log_2 k \rfloor \quad (k \geq 2). \tag{7.1}$$

Then for every $n \geq 0$ and $H > 0$,

$$v_2(C(n+H) - C(n)) = b + v_2(H). \tag{7.2}$$

The Newton interpolation extends continuously to $\mathbb{Z}_2$ and satisfies the same identity for any two distinct dyadic arguments.

Proof. At a natural argument the exact Newton expansion is finite. Vandermonde gives

$$\binom{n+H}{k} - \binom{n}{k} = \sum_{i=1}^{k} \binom{H}{i}\binom{n}{k-i}.$$

For $1 \leq i \leq H$, $v_2(\binom{H}{i}) \geq v_2(H) - v_2(i) \geq v_2(H) - \lfloor \log_2 k \rfloor$. Terms with $i > H$ vanish. Thus each Newton summand with $k \geq 2$ has valuation at least $b + v_2(H) + 1$. The $k = 1$ summand is Ha_1, of exact valuation $b + v_2(H)$, proving (7.2).

The coefficient hypothesis implies $a_k \to 0$ dyadically, so the Newton series converges uniformly on $\mathbb{Z}_2$, where $\binom{x}{k}$ is integral-valued. For distinct $x, y \in \mathbb{Z}_2$, approximate the pair by natural integer pairs with their difference valuation eventually equal to $v_2(x-y)$. The corresponding function differences have the fixed exact valuation $b + v_2(x-y)$. A limit of elements with that fixed valuation has the same valuation: the normalized residues are units, and $\mathbb{Z}_2^\times$ is closed. □

Theorem 7.2 (Exact four-address first differences). *For every $n \geq 0$, $H > 0$, and $j \in \{0,1,2,3\}$,*

$$v_2(C_j(n+H) - C_j(n)) = v_2(H) + b_j, \qquad (b_0, b_1, b_2, b_3) = (1,1,4,2), \tag{7.3}$$

$$v_2(E^{(4,j)}_{4(n+H)} - E^{(4,j)}_{4n}) = v_2(H) + B_j, \qquad (B_0, B_1, B_2, B_3) = (1,1,5,3). \tag{7.4}$$

Consequently, the least positive row period modulo 2^r is exactly

$$P_j(r) = 2^{\max(0, r - B_j)} \qquad (r \geq 1). \tag{7.5}$$

The corresponding displacement in the original Euler subscript is $4P_j(r)$.

Proof. The actual reciprocal recurrence is

$$C_{N,j}(n) = -\sum_{q=1}^{n} \binom{Nn}{Nq} g_{N,j}(q) C_{N,j}(n-q), \qquad C_{N,j}(0) = 1. \tag{7.6}$$

For $N = 4$, the four kernels are

$$1, \qquad \frac{1}{4q+1}, \qquad \frac{1}{(4q+1)(2q+1)}, \qquad \frac{3}{(4q+1)(2q+1)(4q+3)}.$$

The threshold-free theorem gives $v_2(a_k) \geq k$ in every address. The following exact values cover all orders where that bound does not already imply (7.1):

j	k	a_k	$v_2(a_k)$
0	1	-2	1
0	2	72	3
1	1	$-6/5$	1
1	2	$184/45$	3
2	1	$-16/15$	4
2	2	$64/45$	6
2	3	$-14848/1365$	9
2	4	$71426048/69615$	13
2	5	$-25509691392/85085$	16
2	6	$149113198870528/765765$	22
3	1	$-36/35$	2
3	2	$256/231$	8
3	3	$-65792/35035$	8
3	4	$161644544/4849845$	15

They are finite rational evaluations of (7.6), followed by $a_k = \sum_{h=0}^{k} (-1)^{k-h} \binom{k}{h} C_j(h)$. For addresses zero and one, $k \geq 3$ is covered by the generic bound. For address two, $k \geq 7$ is covered. For address three, $k \geq 5$ is covered. In each range the elementary inequality is $k - \lfloor \log_2 k \rfloor \geq b_j + 1$. Lemma 7.1 proves the normalized identity. Multiplication by $j!$ proves the original identity.

A row displacement H preserves the original coefficient modulo 2^r at even one row exactly when $v_2(H) + B_j \geq r$. The least positive such integer is $2^{\max(0, r - B_j)}$, and the same displacement works at every row. This proves both optimality and the absence of a starting threshold. □

Corollary 7.3 (Exact image and residue enumeration)**.** *Let $\widetilde{C}_j$ denote the dyadic Newton interpolation. The map*

$$T_j(x) = \frac{\widetilde{C}_j(x) - 1}{2^{b_j}}$$

is a bijective isometry $\mathbb{Z}_2 \to \mathbb{Z}_2$, and

$$\widetilde{C}_j(\mathbb{Z}_2) = 1 + 2^{b_j}\mathbb{Z}_2, \qquad j!\widetilde{C}_j(\mathbb{Z}_2) = j! + 2^{B_j}\mathbb{Z}_2. \tag{7.7}$$

For $r > B_j$, one least period modulo 2^r visits every residue in $j! + 2^{B_j}\mathbb{Z}/2^r\mathbb{Z}$ exactly once. For $r \leq B_j$, there is one residue and the least row period is one.

Proof. The exact difference identity gives $v_2(T_j(x) - T_j(y)) = v_2(x - y)$ and $T_j(0) = 0$. For every $s \geq 1$, the map induces an injection of $\mathbb{Z}_2/2^s\mathbb{Z}_2$ into itself. It is a bijection because both sets have 2^s elements. These quotient bijections are compatible with reduction. Inverting them gives a compatible preimage system for every dyadic integer, proving surjectivity. Multiplication by $j!$ changes the image ideal by exactly $v_2(j!)$. The finite quotient bijection is also precisely the residue enumeration in a least row period. □

7.1 An infinite exact family at all dyadic ranks

Theorem 7.4 (Lower-half exact laws). *Let $N = 2^a \geq 4$ and $0 \leq j < N/2$. Then for all $n \geq 0$ and $H > 0$,*

$$v_2(C_{N,j}(n + H) - C_{N,j}(n)) = 1 + v_2(H). \tag{7.8}$$

Its normalized interpolation is a bijection onto $1 + 2\mathbb{Z}_2$. For the original coefficients, the least row period modulo 2^r is $2^{\max(0,r-1-v_2(j!))}$.

Proof. Every slope N/i in the source has valuation at least two, since $i \leq j < N/2$. Hence $g(1) \equiv 1 \pmod 4$ and $g(2) \equiv 1 \pmod 8$. The first two Newton coefficients from the reciprocal recurrence are

$$a_1 = -g(1) - 1, \qquad a_2 = \binom{2N}{N} g(1)^2 - g(2) + 2g(1) + 1. \tag{7.9}$$

The first has valuation exactly one. Also $\binom{2N}{N} \equiv 6 \pmod 8$ for every $N = 2^a \geq 2$. To see this, write

$$B_a = \binom{2^{a+1}}{2^a} = 2 \prod_{h=1}^{2^a-1} \left(1 + \frac{2^a}{h}\right).$$

The even-h factors give B_{a-1}. The odd-h factors are one modulo four for $a \geq 2$. Since B_{a-1} has valuation one, $B_a \equiv B_{a-1} \pmod 8$, and $B_1 = 6$. Therefore $a_2 \equiv 6 - 1 + 2 + 1 = 0 \pmod 8$. The global Mahler bound covers all $k \geq 3$, completing the criterion with $b = 1$. The period and image deductions are the same as above. The proof also covers $N = 2, j = 0$, whose kernel is identically one. □

7.2 Complete exact laws at ranks two and eight

Proposition 7.5 (Small-rank complete isometry table). *Every address at $N \in \{2, 4, 8\}$ satisfies*

$$v_2(C_{N,j}(n + H) - C_{N,j}(n)) = b_{N,j} + v_2(H).$$

The following entries are ordered by $j = 0, \ldots, N - 1$:

N	*normalized shifts* $b_{N,j}$	*original shifts* $B_{N,j} = b_{N,j} + v_2(j!)$
2	1, 2	1, 2
4	1, 1, 4, 2	1, 1, 5, 3
8	1, 1, 1, 1, 4, 3, 2, 2	1, 1, 2, 2, 7, 6, 6, 6

For each entry, the least original row period is $2^{\max(0,r-B_{N,j})}$, its Euler subscript displacement is N times that period, and the exact-image and residue-enumeration conclusions hold with the corresponding shifts.

Proof. Rank four was proved above. At ranks two and eight the finite criterion is verified by the actual rational recurrence (7.6). The following valuation lists record every initial Newton coefficient needed in addition to the generic tail $v_2(a_k) \geq k$:

N	j	$b_{N,j}$	$v_2(a_1), v_2(a_2), \ldots$ through the finite cutoff
2	0	1	1, 3
2	1	2	2, 5, 9, 11
8	0	1	1, 3
8	1	1	1, 3
8	2	1	1, 3
8	3	1	1, 3
8	4	4	4, 6, 9, 13, 16, 20
8	5	3	3, 6, 10, 13, 17
8	6	2	2, 6, 8, 13
8	7	2	2, 11, 8, 15

The first entries can also be read directly from $a_1 = -1 - \binom{N+j}{j}^{-1}$. At rank eight they are respectively

$$-2,\ -\frac{10}{9},\ -\frac{46}{45},\ -\frac{166}{165},\ -\frac{496}{495},\ -\frac{1288}{1287},\ -\frac{3004}{3003},\ -\frac{6436}{6435}.$$

All later entries in the table are finite rational arithmetic, with full numerator and denominator records retained in the accompanying exact certificate `all_rank_extensions_verified.json`. Every entry with $k \geq 2$ meets $b_{N,j} + 1 + \lfloor \log_2 k \rfloor$. Beyond the displayed cutoff, the inequality $k \geq b_{N,j} + 1 + \lfloor \log_2 k \rfloor$ holds at every order. Thus the finite table and the proved infinite tail establish the full hypothesis of Lemma 7.1. This is a finite coefficient proof followed by a universal tail argument, rather than extrapolation from sampled rows or steps. □

Remark 7.6 (Failure of a universal upper-half exact law). The all-rank lower bound remains valid in every address. The exact first-difference law does not automatically extend to all upper-half addresses. Exact rational recurrence calculations give

(N, j)	$v_2(C(1) - C(0))$	$v_2(C(2) - C(0))$	naive $b + v_2(2)$
$(16, 9)$	5	7	6
$(32, 18)$	6	6	7
$(32, 19)$	5	7	6

For example, at $(N, j) = (16, 9)$ the first difference is $-2042976/2042975$, whereas the two-row difference is $-2215603929728/2215923048625$. These exact counterexamples refute the additional universal isometry proposal. They preserve the threshold-free lower bound and the original conjecture.

8 Analytic interpolation and exact Taylor radii

The threshold-free Mahler theorem gives continuous interpolation. Analytic interpolation needs a stronger estimate, and the factorial bounds provide it directly. The ordinary Taylor radius is determined by the coefficient growth, including infinite exact subsequences.

Theorem 8.1 (Taylor coefficient bounds and radii). *For $j = 0, 1, 2, 3$, the dyadic interpolation has an ordinary power series*

$$\widetilde{C}_j(x) = 1 + \sum_{\ell \geq 1} d_{j,\ell} x^\ell, \qquad v_2(d_{j,\ell}) \geq \ell + \beta_j.$$

Its Taylor series converges for $|x|_2 < 2$. At addresses zero and one,

$$v_2(d_{j,\ell}) = \ell \quad (\ell \geq 1), \qquad R_0 = R_1 = 2. \tag{8.1}$$

At address two,

$$v_2(d_{2,\ell}) \geq \max\left(2\ell + \left\lfloor \frac{\ell}{2} \right\rfloor, \ell + 3\right), \qquad R_2 = 2^{5/2}. \tag{8.2}$$

The radius at address three is at least two. Its exact radius is not asserted here. The stated exact radii at addresses zero, one, and two hold at every center in $\mathbb{Z}_2$.

Proof. Put $a_m = \Delta_1^m C_j(0)$ and expand the falling factorial in the Newton series. The ordinary coefficients are

$$d_{j,\ell} = \sum_{m \geq \ell} \frac{a_m}{m!} s(m, \ell). \tag{8.3}$$

Since $s(m, \ell)$ is integral and $v_2(a_m/m!) \geq m + \beta_j$, the coefficient sum converges and has valuation at least $\ell + \beta_j$. The sum exchange is justified on $\mathbb{Z}_2$ by these uniform geometric coefficient bounds. The ordinary power series agrees there with the Newton interpolation and converges whenever $|x|_2 < 2$.

For $j = 0, 1$, the term $m = \ell$ in (8.3) has exact valuation ℓ, since $s(\ell, \ell) = 1$ and the Newton valuation is exact. Every later term has valuation at least $\ell + 1$. Thus $v_2(d_{j,\ell}) = \ell$, and the coefficient radius is exactly two. At a center $x_0 \in \mathbb{Z}_2$, the coefficient of y^ℓ in the translated series is

$$\sum_{m \geq \ell} d_{j,m} \binom{m}{\ell} x_0^{m-\ell}. \tag{8.4}$$

Its initial term again has exact valuation ℓ, and all later terms have at least one more bit. The exact radius is therefore two at every integral center.

For address two, the stronger factorial filtration gives the lower bound in (8.2). At each $\ell = 2^r \geq 4$, Theorem 6.3 makes the $m = \ell$ term in (8.3) have valuation $2\ell + \ell/2 = 5\ell/2$. Every $m > \ell$ term has strictly greater lower bound $2m + \lfloor m/2 \rfloor$. Hence

$$v_2(d_{2,\ell}) = 5\ell/2 \qquad (\ell = 2^r \geq 4).$$

The universal lower bound and this unbounded exact subsequence prove that the coefficient radius is exactly $2^{5/2}$. Translation by an integral center preserves the same unique leading coefficients at these degrees in (8.4), proving the exact radius at every such center. □

Remark 8.2 (Boundary interpretation). For addresses zero and one, at an argument of absolute value two every positive-degree Taylor term has valuation zero, so that ordinary Taylor series diverges. For address two, over a complete extension admitting arguments of absolute value $2^{5/2}$, the power-of-two-degree terms have valuation zero and fail to tend to zero. These statements concern the Taylor expansions just constructed. They do not prohibit a different analytic continuation. On $\mathbb{Q}_2$-valued arguments, open-disc notation must be read with its discrete value group. The coefficient radius is the precise invariant in the theorem.

9 The final congruence in the original Euler convention

The preceding bounds can be stated as one uniform congruence without suppressing the address normalization. This is the strengthened formula for the four-address family, valid from the first row.

Corollary 9.1 (Final common-step congruence). *Let $j \in \{0,1,2,3\}$, $n \geq 0$, and $k, H \geq 1$. Define*

$$A_0(k) = A_1(k) = k, \qquad A_2(k) = \max\left(k+4,\ 2k+\left\lfloor \frac{k}{2} \right\rfloor + 1\right), \qquad A_3(k) = k+2.$$

Then

$$\boxed{\sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} E^{(4,j)}_{4(n+iH)} \equiv 0 \pmod{2^{\,v_2(k!)+kv_2(H)+A_j(k)}}.} \tag{9.1}$$

For $j = 0, 1$, this displayed exponent is the exact valuation for every n, k, H in the stated range. For $j = 2$, the exact positive power-of-two-order valuations are given by Theorem 6.3, with one added bit.

Proof. The left side is $j!\Delta_H^k C_j(n)$. At addresses zero and one, $v_2(j!) = 0$ and the exact homogeneous formula gives the exponent. At address two, $v_2(2!) = 1$. The maximum of (5.1) and (5.2) yields $\max(k+4, 2k+\lfloor k/2 \rfloor + 1)$ after removing the common $v_2(k!) + kv_2(H)$. At address three, the one-bit normalized offset and $v_2(3!) = 1$ give $k+2$. □

For mixed steps the same corollary holds with $kv_2(H)$ replaced by $\sum_\nu v_2(H_\nu)$ and the binomial stencil replaced by $j!\Delta_{H_1} \cdots \Delta_{H_k} C_j(n)$. At addresses zero and one, exactness is then governed by the parity classification in Theorem 6.1. At first order these bounds are exact in every address, with original valuation offsets $(1, 1, 5, 3)$. The least periods are those in (7.5).

9.1 Proof coverage and remaining questions

These sections give detailed mathematical proofs. Formal coverage is statement-specific. The Lean sources include the rank-four homogeneous bound at every row, the sharp first-difference laws, least periods, and dyadic isometry and image conclusions.

The all-rank dyadic and general prime-power transfer arguments, factorial improvements, mixed leading-parity classification, Bernoulli power-order sharpness, all-rank lower-half exact laws, rank-eight extension, and exact Taylor-radius arguments have written proofs but no dedicated Lean formalization. Mathematical review and exact computation do not extend their formal coverage.

The completed external statement is Nishibuchi's Conjecture 3.9 in its prescribed parameter family and coefficient normalization. These stronger congruences enlarge its dyadic branch and remove its starting threshold there. They do not settle every Euler problem. In particular, the following questions remain separate:

- a complete exact higher-difference law at all non-power-of-two orders in address two;
- complete higher-difference valuations and the exact ordinary Taylor radius in address three;

- a classification of the upper-half dyadic addresses having an exact first-difference metric law at arbitrary larger ranks;
- the distinct universal simplicity or nonvanishing question for the later Euler-tail residuals, and the original native Paper IX conjectures beyond the completed external statement.

The exact counterexamples above rule out the naive universal upper-half isometry, and mixed cancellations rule out unconditional baseline sharpness for arbitrary stencils. No global historical-priority or novelty assertion is made for these strengthened statements.

10 Finite obstructions, compatible lifts, and deformation

The arithmetic results above concern particular families. We now study the underlying correction problem, from finite polynomial lifts to continuous parameter charts and quantitative deformation. Throughout, we distinguish three objects: a residual obstruction, the set of correcting digits, and a root in a completion. A vanishing obstruction at one precision gives a lift to the next. Continuing indefinitely requires additional hypotheses.

Unless a different coefficient ring is specified, a polynomial system is a finite list of integer polynomials. All divisibility conditions on vectors are coordinatewise. For a positive integer q, write $S_q = \mathbb{Z}/q\mathbb{Z}$ and use an overline for reduction modulo q. For a prime p, write $\mathbb{F}_p = S_p$, normalize $|p|_p = p^{-1}$, and put

$$B_L(x) = x + p^L\mathbb{Z}_p^d.$$

An integer center is included in this notation by its natural image in $\mathbb{Z}_p^d$. All dimensions below are finite and may be zero. A statement over S_q allows composite q and $q = 1$ when explicitly stated. A cardinality formula in terms of vector-space rank uses prime p.

10.1 The polynomial correction equation

For $F = (F_1, \dots, F_e) \in \mathbb{Z}[X_1, \dots, X_d]^e$, define its evaluated Jacobian by the formal partial derivatives:

$$DF(x)_{ij} = (\partial_j F_i)(x), \qquad J_x = DF(x) \bmod q : S_q^d \longrightarrow S_q^e.$$

The obstruction therefore uses the Jacobian of the given polynomials. The following algebraic Taylor identity relates this linear map to polynomial evaluation.

Lemma 10.1 (Polynomial Taylor remainder)**.** *Let R be a commutative ring, $P \in R[X_1, \dots, X_d]$, $a \in R^d$, and $c \in R$. There is a polynomial $Q \in R[U_1, \dots, U_d]$, with $Q(0) = 0$, such that for every $u \in R^d$,*

$$P(a + cu) = P(a) + c\sum_{j=1}^{d}(\partial_j P)(a)u_j + c^2 Q(u). \tag{10.1}$$

In particular, when $R = \mathbb{Z}$ and a, u, c are integral, subtracting the first two terms on the right from $P(a + cu)$ gives a multiple of c^2.

Proof. Apply polynomial induction using constants, addition, and multiplication by a variable. Constants have zero remainder, and remainders add. For the multiplication step, suppose the assertion is known for P and write $D_P(U) = \sum_j(\partial_j P)(a)U_j$. Multiplication of its expansion by $a_i + cU_i$ gives constant term $P(a)a_i$, linear term $c(a_i D_P(U) + P(a)U_i)$, and remainder

$$c^2\big(Q(U)(a_i + cU_i) + D_P(U)U_i\big).$$

The product rule identifies the linear term with the derivative of PX_i. The displayed remainder vanishes at $U = 0$. This constructs the polynomial remainder before evaluation at u, even over rings where division by c would be invalid. □

The general polynomial identity is formalized by `Komorebi.PolynomialTaylorRing.exists_eval_scaled_remainder`. Its integer divisibility form is `Komorebi.PolynomialTaylor.taylor_dvd`.

Theorem 10.2 (Finite polynomial obstruction)**.** *Let $q \geq 1$, $L \geq 1$, $x \in \mathbb{Z}^d$, and suppose $F(x) \in q^L\mathbb{Z}^e$. Define the exact integer residual r_x by $F(x) = q^L r_x$, and put*

$$\mathcal{O}_x = S_q^e / \operatorname{im} J_x, \qquad o_x = [\overline{r_x}].$$

Then $\mathcal{O}_x$ is finite and

$$o_x = 0 \quad \Longleftrightarrow \quad \exists y \in \mathbb{Z}^d : \quad y \equiv x \pmod{q^L}, \qquad F(y) \equiv 0 \pmod{q^{L+1}}. \tag{10.2}$$

No surjectivity or nonsingularity assumption is required.

Proof. Every point in the lower residue class has the form $y = x + q^L t$ with $t \in \mathbb{Z}^d$. By Lemma 10.1 and $2L \geq L + 1$,

$$F(x + q^L t) \equiv q^L \big(r_x + DF(x)t\big) \pmod{q^{L+1}}.$$

The integer q^L is nonzero. Cancelling it in the corresponding divisibility condition proves

$$F(x + q^L t) \equiv 0 \pmod{q^{L+1}} \quad \Longleftrightarrow \quad \overline{r_x} + J_x \bar{t} = 0. \tag{10.3}$$

Integer vectors reduce onto S_q^d. The equation on the right is solvable exactly when $\overline{r_x}$ belongs to $\operatorname{im} J_x$, since that image is closed under negation. This is precisely $o_x = 0$. Finiteness follows because $\mathcal{O}_x$ is a quotient of the finite module S_q^e. □

The formal theorem `Komorebi.PolynomialObstruction.finite_polynomial_obstruction` presents the image using the integer-domain map $\mathbb{Z}^d \xrightarrow{DF(x)} \mathbb{Z}^e \to S_q^e$. Reduction onto S_q^d identifies that image with $\operatorname{im} J_x$. The class-preserving comparison is explicit in `Komorebi.PolynomialLiftCounts.obstructionEquiv`. The change of presentation does not change the distinguished residual class.

Proposition 10.3 (Affine depth zero)**.** *Let $F(x) = Ax - b$ for an integer linear map $A : \mathbb{Z}^d \to \mathbb{Z}^e$. The criterion of Theorem 10.2 holds for every $L \geq 0$, with $J = A \bmod q$.*

Proof. The affine identity $F(x + q^L t) = F(x) + q^L A t$ has no quadratic remainder. The proof of (10.3) therefore needs no inequality involving $2L$. The residual and the finite quotient are obtained as above. □

The affine statement is formalized by `Komorebi.NativeAffine.finite_affine_obstruction`. Positive depth is essential in the nonlinear theorem: for $F(X) = X^2 - 1$, $q = 2$, $L = 0$, and $x = 0$, the Jacobian is zero and the residual class is nonzero, yet $y = 1$ is a root modulo two in the same residue modulo one.

Proposition 10.4 (Independence of the integer representative)**.** *Under the hypotheses of Theorem 10.2, let $x' = x + q^L s$ and write $F(x') = q^L r_{x'}$. Then*

$$J_{x'} = J_x, \qquad \overline{r_{x'}} = \overline{r_x} + J_x \bar{s}.$$

Consequently the two residual classes coincide in the common quotient $S_q^e / \operatorname{im} J_x$.

Proof. Polynomial evaluation respects coordinatewise congruence. Apply this first to every formal partial derivative: $x' \equiv x \pmod{q}$ gives $J_{x'} = J_x$. Next apply (10.1) to F at scale q^L. Subtract the two exact residual identities and cancel q^L to obtain the displayed relation modulo q. Its additional term belongs to the common Jacobian image, so it vanishes in the quotient. □

This comparison is formalized by `Komorebi.PolynomialRepresentative.representative_independence`. This proves independence of the integer representative within a fixed polynomial presentation. It makes no identification of the raw cokernels of different equation lists. For example, adjoining a zero equation leaves the roots unchanged while adding an S_q summand to the raw cokernel.

10.2 One correction shared by two systems

The next obstruction concerns compatibility between individually attainable corrections. Its algebraic form works over an arbitrary commutative ring. Let U, V, W be R-modules and let $J : U \to V$, $K : U \to W$ be linear. Define

$$C(J,K) = U/(\ker J + \ker K). \tag{10.4}$$

An attainable value Ju has label $[u] \in C(J,K)$. This is independent of the choice of its preimage because $\ker J$ is killed in the quotient. Similarly, Kv has label $[v]$. Denote these maps by $\lambda_J : \operatorname{im} J \to C(J,K)$ and $\lambda_K : \operatorname{im} K \to C(J,K)$.

Theorem 10.5 (Common correction and fiber product)**.** *For every $u, v \in U$,*

$$[u] = [v] \text{ in } C(J,K) \quad \Longleftrightarrow \quad \exists w \in U :\ Jw = Ju, \quad Kw = Kv. \tag{10.5}$$

Both label maps are surjective, and the image of the joint map is exactly the fiber product:

$$\operatorname{im}\big(U \to \operatorname{im} J \times \operatorname{im} K,\ u \mapsto (Ju, Ku)\big) = \operatorname{im} J \times_{C(J,K)} \operatorname{im} K. \tag{10.6}$$

If either V or W is finite, then $C(J,K)$ is finite.

Proof. If $[u] = [v]$, write $u - v = s + t$ with $s \in \ker J$ and $t \in \ker K$. Then $w = u - s = v + t$ satisfies both required equalities. Conversely, a common w gives

$$u - v = (u - w) + (w - v) \in \ker J + \ker K.$$

Every class $[u]$ is the label of Ju and of Ku, proving surjectivity. For attainable $a = Ju$ and $b = Kv$, the first equivalence says exactly that equal labels are equivalent to a simultaneous preimage. This proves (10.6). Finally, a finite target has finite image, and the associated surjective label map makes $C(J,K)$ finite. □

These statements are `Komorebi.SharedCorrection.quotient_mk_eq_iff_common` and `Komorebi.SharedCorrection.joint_range_eq_fiber`, together with the finite-target corollaries in the same module. The quotient also has the expected universal property: every linear map from U annihilating both kernels factors uniquely through $C(J,K)$. Equivalently, compatible linear observations on $\operatorname{im} J$ and $\operatorname{im} K$ factor uniquely through their label maps. This is formalized by `Komorebi.SharedCorrection.common_paired_universal`.

Corollary 10.6 (Simultaneous polynomial lifting)**.** *Let $F \in \mathbb{Z}[X_1, \dots, X_d]^e$ and $G \in \mathbb{Z}[X_1, \dots, X_d]^f$ have a common lower root x modulo q^L, where $q \geq 1$ and $L \geq 1$. Use the integer-domain reduced Jacobians $J : \mathbb{Z}^d \to S_q^e$ and $K : \mathbb{Z}^d \to S_q^f$. Suppose $u, v \in \mathbb{Z}^d$ are separately valid corrections for F and G, respectively. Then*

$$[u] = [v] \text{ in } \mathbb{Z}^d/(\ker J + \ker K)$$

if and only if a single $y \equiv x \pmod{q^L}$ solves both systems modulo q^{L+1}. Each separate label is independent of the chosen valid correction. The common quotient is finite.

Proof. For each system, (10.3) translates a valid next root into a linear correction equation. Once u corrects F, another digit w corrects F precisely when $Jw = Ju$. The analogous condition for G is $Kw = Kv$. Theorem 10.5 now gives the criterion. Two valid corrections of one block differ by a vector in its Jacobian kernel, so they have equal labels. Finiteness follows from either finite residue target. An unconditional formulation quantifies over separately valid u, v with equal labels, thereby including their existence in the criterion rather than assuming it. □

See `Komorebi.PolynomialShared.joint_lift_iff_class_eq` and `Komorebi.PolynomialShared.finite_polynomial_pair_compatibility`.

For affine integer systems $F(X) = AX - b$ and $G(X) = BX - c$, where $A : \mathbb{Z}^d \to \mathbb{Z}^e$ and $B : \mathbb{Z}^d \to \mathbb{Z}^f$ are linear, Corollary 10.6 holds for $q \geq 1$ and every $L \geq 0$, including depth zero. Here x is a common root modulo q^L, and J and K are the reductions of A and B with domain $\mathbb{Z}^d$. Given separately valid corrections u, v, a simultaneous next lift exists if and only if their classes in $\mathbb{Z}^d/(\ker J + \ker K)$ agree. The quotient is finite, and each label is independent of the chosen valid correction. Indeed, Proposition 10.3 supplies the correction equation without a positive-depth restriction, and the proof of Theorem 10.5 then applies unchanged. The formal statement is `Komorebi.AffinePair.joint_lift_iff_class_eq`. Quantifying over separately valid corrections with equal labels also includes their existence in the criterion, as in `Komorebi.AffinePair.finite_affine_pair_compatibility`. Separate liftability alone is insufficient. At $x = 0$ modulo two, $F(X) = X$ and $G(X) = X - 2$ both lift to modulus four, but their correcting digits are respectively 0 and 1 modulo two. No single residue solves both equations modulo four.

10.3 Correction torsors and exact residue counts

Integer representatives and residue classes at the next level must also be distinguished. For $q \geq 1$ and $L \geq 0$, the map

$$S_q^d \longrightarrow \{y \in (\mathbb{Z}/q^{L+1}\mathbb{Z})^d : y \equiv x \pmod{q^L}\}, \qquad t \longmapsto x + q^L\widetilde{t}$$

is a bijection, where $\widetilde{t}$ is any integer representative. Independence of the representative and injectivity both follow by cancelling q^L in integer divisibility. Surjectivity follows by expressing the lower congruence as a multiple of q^L. This is the equivalence `Komorebi.ResidueLiftDigits.digitEquivResidue`.

At positive polynomial depth, restrict it to the correction fiber

$$\mathcal{T}_x = \{t \in S_q^d : \overline{r_x} + J_x t = 0\}$$

and the lift set

$$\mathcal{L}_{x,L} = \{y \in (\mathbb{Z}/q^{L+1}\mathbb{Z})^d : y \equiv x \pmod{q^L},\ F(y) = 0\}.$$

The polynomial predicate in the second display is evaluation in the residue ring. Evaluation on canonical integer representatives gives the same predicate, because integer polynomial evaluation respects congruence.

Theorem 10.7 (Torsor and lift cardinalities)**.** *Under the hypotheses of Theorem 10.2, there is an equivalence $\mathcal{T}_x \simeq \mathcal{L}_{x,L}$. Every nonempty fiber is a torsor for* $\ker J_x$ *under addition of correction digits. If $q = p$ is prime and $\rho_x = \operatorname{rank}_{\mathbb{F}_p} J_x$, then*

$$\#\mathcal{L}_{x,L} = \begin{cases} p^{d-\rho_x}, & o_x = 0, \\ 0, & o_x \neq 0, \end{cases} \qquad \#\mathcal{O}_x = p^{e-\rho_x}. \tag{10.7}$$

Proof. Equation (10.3) identifies exactly which points of the residue fiber satisfy the equations, proving the restricted equivalence. If $u, t \in \mathcal{T}_x$, subtraction gives $J_x(t-u) = 0$. Conversely, adding any element of $\ker J_x$ preserves the correction equation. The unique kernel vector carrying u to t is $t-u$, so the action is free and transitive. Choosing one valid digit identifies the fiber with the kernel, without making that choice part of its translation action.

For prime p, rank-nullity gives $\dim \ker J_x = d - \rho_x$ and $\dim(\mathbb{F}_p^e / \operatorname{im} J_x) = e - \rho_x$. The cardinality of an n-dimensional $\mathbb{F}_p$-space is p^n. Combine these identities with the obstruction criterion and the residue equivalence. □

The cardinality formula is formalized by `Komorebi.PolynomialLiftCounts.finite_polynomial_lift_count`. The explicit torsor and chosen-point equivalence are developed in `Komorebi.FiniteFiberCounts.translate_exists_unique` and `Komorebi.PolynomialLiftCounts.liftEquivKernel`. For composite q, the torsor and kernel cardinality remain valid, while the vector-space rank formula is unavailable. Multiplication by two on $\mathbb{Z}/4\mathbb{Z}$ already has a kernel of size two.

10.4 Smooth compatible towers and exact roots

Surjectivity removes every residual obstruction and persists throughout a positive-depth residue ball. This supplies the additional hypothesis needed to iterate the finite theorem.

Theorem 10.8 (Compatible polynomial tower)**.** *Let $q \geq 1$, $L \geq 1$, $F \in \mathbb{Z}[X_1, \ldots, X_d]^e$, and $x \in \mathbb{Z}^d$ satisfy $F(x) \in q^L\mathbb{Z}^e$. Suppose $J_x : S_q^d \to S_q^e$ is surjective. There is a single sequence $z_n \in \mathbb{Z}^d$ such that*

$$z_0 = x, \qquad F(z_n) \in q^{L+n}\mathbb{Z}^e,$$
$$z_{n+1} - z_n \in q^{L+n}\mathbb{Z}^d, \qquad J_{z_n} = J_x.$$

For $a \leq b$, it also satisfies $z_b - z_a \in q^{L+a}\mathbb{Z}^d$.

Proof. At level $N = L + n$, divide the residual by q^N. Surjectivity supplies $t \in S_q^d$ with $J_{z_n} t = -\overline{F(z_n)/q^N}$. Choose an integer representative and apply (10.3) to obtain z_{n+1}. The new point is congruent to z_n modulo q. Evaluation of each formal partial derivative therefore gives $J_{z_{n+1}} = J_{z_n}$.

Construct the sequence recursively, retaining the root condition and the fixed Jacobian at each step. Surjectivity at every later stage then follows from the single initial assumption. The cross-level congruence follows by induction, adding consecutive differences whose divisibilities all imply divisibility by q^{L+a}. □

The full tower statement is `Komorebi.PolynomialIntegerTower.exists_compatible_tower_strong`. The proof constructs a compatible path using classical choice. It does not claim an optimized procedure for selecting its digits.

Theorem 10.9 (Completion and smooth branching)**.** *In Theorem 10.8, let $q = p$ be prime. The tower can be chosen with a limit $a \in \mathbb{Z}_p^d$ such that*

$$F(a) = 0, \qquad z_n \longrightarrow a, \qquad a - z_n \in p^{L+n}\mathbb{Z}_p^d \quad (n \geq 0).$$

At each stage, the number of distinct next-level residue roots over z_n is exactly p^{d-e}, and the finite obstruction module has one element. If $d = e$, there is exactly one root of F in $B_L(x)$.

Proof. Cross-level compatibility bounds the supremum distance between later terms by $p^{-(L+n)}$. Completeness gives a limit, and closed norm bounds retain the congruence to every z_n. View the coefficients of the original integer polynomials in $\mathbb{Z}_p$. Their evaluations satisfy both $F_i(z_n) \to F_i(a)$, by continuity, and $F_i(z_n) \to 0$, by their growing divisibility. Uniqueness of limits gives exact vanishing.

Surjectivity gives rank $J_x = e \leq d$. Theorem 10.7 then gives p^{d-e} next residue roots and obstruction cardinality one at every stage.

For square uniqueness, let $a, b \in B_L(x)$ be arbitrary exact roots. Their canonical integer truncations are genuine roots in the respective finite residue rings, since evaluation commutes with reduction. They agree at depth L. If they agree at depth $N \geq L$, both next residues belong to the same lift set, which has cardinality $p^{d-d} = 1$. Induction gives agreement at every depth. The residue maps separate p-adic integers, so $a = b$. Existence was already supplied by the tower. □

The endpoint construction is `Komorebi.PolynomialPadicLift.exists_padic_root_with_tower`, the branching statement is `Komorebi.PolynomialSmoothCounts.exists_regular_tower`, and uniqueness for all roots in the ball is `Komorebi.PolynomialPadicUnique.exists_unique_padic_root`.

Remark 10.10. The one-step singular criterion and the smooth infinite theorem have different hypotheses. The polynomial $X^2 - 5$ has odd roots modulo two and four but has no root modulo eight. Thus a previous zero obstruction does not force the next one to vanish. Also, rectangular surjectivity alone does not imply uniqueness: $F(X, Y) = X$ has the distinct dyadic roots $(0, 0)$ and $(0, 2)$ in the same residue ball modulo two. These examples show why the hypotheses of the tower theorem and its uniqueness clause are needed.

10.5 Coefficient stability and continuous root charts

Proposition 10.11 (Smooth coefficient stability)**.** *Let p be prime, $M \geq 1$, $F, H \in \mathbb{Z}[X_1, \ldots, X_d]^e$, and $c \in p^M\mathbb{Z}$. Suppose $DF(x) \bmod p$ is surjective for an integer $x \in \mathbb{Z}^d$. If $a \in \mathbb{Z}_p^d$ satisfies $F(a) = 0$ and $a \equiv x \pmod p$, then there is a root b of $G = F + cH$ with $b - a \in p^M\mathbb{Z}_p^d$.*

If $d = e$ and $M \geq L \geq 1$, every pair of roots $a, b \in B_L(x)$ of F and G, respectively, satisfies $b - a \in p^M\mathbb{Z}_p^d$. The perturbed root is unique in that ball whenever it exists.

Proof. Since $p \mid c$, differentiation gives $DG(x) \equiv DF(x) \pmod p$. Truncate a to an integer vector y at depth M. Then $F(y) \equiv 0 \pmod{p^M}$, hence also $G(y) \equiv 0 \pmod{p^M}$. Congruence of y and x modulo p preserves the reduced Jacobian. The smooth root theorem gives $b \equiv y \equiv a \pmod{p^M}$.

In the square case this constructed root lies in $B_L(x)$ because $M \geq L$. Square uniqueness for G identifies it with every competing root in that ball. The construction therefore gives the precision assertion for an arbitrary root there. □

See `Komorebi.PolynomialRootStability.exists_nearby_perturbed_root` and `Komorebi.PolynomialRootStability.perturbed_roots_congruent`. The scalar c may be negative or zero. The weighted theorems below allow arbitrary integral p-adic coefficients and give an explicit first-order displacement.

Theorem 10.12 (Continuous charts with a selected Jacobian minor)**.** *Let p be prime, $L \geq 1$, $F \in \mathbb{Z}[X_1, \ldots, X_{e+k}]^e$, and $x \in \mathbb{Z}^{e+k}$ satisfy $F(x) \in p^L\mathbb{Z}^e$. Suppose the full reduced Jacobian is surjective. There is a permutation σ of the coordinates such that, writing $\sigma x = (y_0, t_0)$ with e solved coordinates and k parameter coordinates, the parameter ball $T = t_0 + p^L\mathbb{Z}_p^k$ is homeomorphic to the entire root locus*

$$\mathcal{R} = \{z \in B_L(x) : F(z) = 0\}.$$

The inverse homeomorphism is the actual free-coordinate projection $z \mapsto \mathrm{free}(\sigma z)$. If $h : T \to \mathcal{R}$ is the forward map, then for every $N \geq 0$,

$$h(t) \equiv h(u) \pmod{p^N} \quad \Longleftrightarrow \quad t \equiv u \pmod{p^N}. \tag{10.8}$$

Equivalently, $\sigma h(t) = (\phi(t), t)$ for a continuous $\phi : T \to y_0 + p^L \mathbb{Z}_p^e$ whose graph is the entire reordered root locus.

Proof. First consider a square integer polynomial map P whose reduced Jacobian at x is surjective. Its Jacobian is invertible over $\mathbb{F}_p$. A digit comparison using Taylor expansion shows that equal outputs modulo p^{N+1} force equal inputs modulo p^{N+1} whenever the inputs already agree modulo p^N and belong to $B_1(x)$. Induction and integer truncation give, for arbitrary $a, b \in B_1(x)$,

$$P(a) \equiv P(b) \pmod{p^N} \quad \Longleftrightarrow \quad a \equiv b \pmod{p^N}.$$

More generally, for a rectangular map with surjective reduced Jacobian,

$$P(B_L(x)) = P(x) + p^L \mathbb{Z}_p^e. \tag{10.9}$$

One inclusion follows from polynomial congruence. For integer targets in the right-hand ball, subtract the target from the constant terms and use the smooth root theorem. For arbitrary p-adic targets, approximate by integer truncations. The input ball is compact, so its continuous image is closed. It contains all these approximants and hence their limit. This argument does not assume that independently chosen preimages form a compatible sequence.

For square P, the resulting bijection between the two compact residue balls is continuous, and its target is Hausdorff. It is therefore a homeomorphism, with the same congruence preservation for its inverse.

Now suppose a chosen set of e columns of $DF(x) \bmod p$ is independent, and put those coordinates first. Apply the square result to the augmented polynomial map

$$\Psi(Y, T) = (F(Y, T), T).$$

Its reduced Jacobian has upper blocks $(D_Y F, D_T F)$ and lower blocks $(0, I_k)$, so it is invertible. For $t \in T$, the target $(0, t)$ belongs to $\Psi(x) + p^L \mathbb{Z}_p^{e+k}$ because $F(x) \equiv 0 \pmod{p^L}$. The inverse of Ψ at that target gives the unique root with parameters t. The inverse congruence property gives one direction of (10.8). Projection onto the parameter coordinates gives the other.

Finally, surjectivity of the original rectangular Jacobian guarantees a basis among its columns. Extend the corresponding injection of coordinate indices to a permutation σ. Renaming the variables by σ^{-1} transports polynomial evaluation and every formal partial derivative. The fixed-coordinate construction therefore applies to the original system after undoing the permutation. □

The main theorem is `Komorebi.PolynomialGeneralChart.exists_parameter_homeomorph`. The square local inverse and surjectivity for arbitrary targets are `Komorebi.PolynomialLocalInverse.localHomeomorph` and `Komorebi.PolynomialLocalSurjective.exists_padic_preimage`. The center x is only an approximate root: the chart need not satisfy $\phi(t_0) = y_0$ unless $F(x) = 0$ exactly. The assertion selects suitable coordinates. It does not permit an arbitrary prescribed solved block.

10.6 Prescribed parameters and their obstruction

Suppose now that the split into solved variables and parameter variables is fixed in advance. Let

$$F \in \mathbb{Z}[Y_1, \dots, Y_m, T_1, \dots, T_k]^e, \qquad x = (y_0, t_0),$$

and assume $F(x) = q^L r_x$ with $q \geq 1$ and $L \geq 1$. Write the reduced Jacobian as

$$DF(x) \bmod q = (J_x \; K_x), \qquad J_x : S_q^m \to S_q^e, \quad K_x : S_q^k \to S_q^e.$$

For a prescribed parameter digit $v \in S_q^k$, let $\mathcal{L}_{x,L,v}$ be the set of next residue roots over x whose parameter coordinates are $t_0 + q^L \widetilde{v}$ modulo q^{L+1}.

Theorem 10.13 (Prescribed-parameter obstruction and count)**.** *Define*

$$o_x(v) = [\overline{r_x} + K_x v] \in S_q^e / \operatorname{im} J_x.$$

Then $\mathcal{L}_{x,L,v}$ is nonempty if and only if $o_x(v) = 0$. When nonempty, it is a torsor for $\ker J_x$*. If $q = p$ is prime and $\rho_x = \operatorname{rank}_{\mathbb{F}_p} J_x$, then*

$$\#\mathcal{L}_{x,L,v} = \begin{cases} p^{m-\rho_x}, & o_x(v) = 0, \\ 0, & o_x(v) \neq 0, \end{cases} \qquad \#(\mathbb{F}_p^e / \operatorname{im} J_x) = p^{e-\rho_x}. \tag{10.10}$$

The criterion holds uniformly for every prescribed v without a surjectivity hypothesis.

Proof. A candidate for the next lift has the form $x + q^L(\widetilde{u}, \widetilde{v})$, with solved digit $u \in S_q^m$. Its Taylor correction equation is

$$\overline{r_x} + J_x u + K_x v = 0.$$

Solvability is exactly vanishing of the displayed cokernel class. Restrict the residue-digit equivalence to the prescribed parameter coordinates. This identifies the lift fiber with the solved-digit fiber. Translation by $\ker J_x$ is free and transitive on any nonempty such fiber. The prime-field count follows from rank-nullity as before. □

The formal statements start from the lower root and construct its residual: `Komorebi.PolynomialPrescribedObstruction.finite_prescribed_obstruction` and `Komorebi.PolynomialPrescribedObstruction.finite_prescribed_lift_count`. For $F(Y, T) = T$ at the dyadic origin, the full Jacobian $(0\ 1)$ is surjective but $J_x = 0$. Prescribing $v = 1$ forces $T = 2 \pmod 4$, so no root exists. Prescribing $v = 0$ gives two choices of Y modulo four over the initial zero residue. This example explains why the criterion depends on the solved block, rather than merely on the full Jacobian.

Theorem 10.14 (Every compatible integer parameter path)**.** *In the preceding notation, assume J_x is surjective. For every supplied integer path $t_n \in \mathbb{Z}^k$ satisfying*

$$t_0 = \operatorname{free}(x), \qquad t_{n+1} - t_n \in q^{L+n}\mathbb{Z}^k,$$

there is a compatible polynomial tower $z_n \in \mathbb{Z}^{m+k}$ with

$$z_0 = x, \quad \operatorname{free}(z_n) = t_n, \quad F(z_n) \in q^{L+n}\mathbb{Z}^e, \quad z_{n+1} - z_n \in q^{L+n}\mathbb{Z}^{m+k}.$$

The free-coordinate equality is exact in integers. Both reduced Jacobian blocks remain fixed, and $z_b - z_a \in q^{L+a}\mathbb{Z}^{m+k}$ *whenever* $a \le b$.

Proof. At level $N = L + n$, write the supplied parameter increment exactly as $t_{n+1} - t_n = q^N w$ with $w \in \mathbb{Z}^k$. Surjectivity of the solved block gives a digit u satisfying $J_{z_n} u = -\overline{F(z_n)/q^N} - K_{z_n}\overline{w}$. Choose an integer representative $\widetilde{u}$ and set $z_{n+1} = z_n + q^N(\widetilde{u}, w)$. Its parameter coordinates equal the supplied target exactly. Taylor expansion proves the next root condition, while congruence modulo q preserves the full reduced Jacobian. Construct the sequence recursively, preserving the exact parameter coordinates and the fixed solved Jacobian. The earlier cross-level argument proves all the stated integer congruences. □

This is `Komorebi.PolynomialPrescribedIntegerTower.exists_prescribed_tower_strong`. In particular, no assumption about solvability at each later level has been added to the initial surjectivity condition.

Theorem 10.15 (Arbitrary p-adic parameter endpoints)**.** *Let* $q = p$ *be prime in Theorem 10.14. For every* $\tau \in t_0 + p^L\mathbb{Z}_p^k$*, there are an integer tower* z_n *and* $b \in \mathbb{Z}_p^{m+k}$ *such that*

$$F(b) = 0, \qquad \mathrm{free}(b) = \tau, \qquad z_n \longrightarrow b, \qquad b - z_n \in p^{L+n}\mathbb{Z}_p^{m+k}.$$

The initial value is $z_0 = x$*. At every stage and for every proposed parameter digit* $v \in \mathbb{F}_p^k$*, the corresponding next residue fiber has exactly* p^{m-e} *elements. If* $m = e$*, then for each* τ *there is exactly one such root in* $B_L(x)$*, independently of all choices of integer approximants and correcting digits.*

Proof. Set the first integer parameter approximation equal to the prescribed t_0. For $n \ge 1$, use canonical representatives of τ mod p^{L+n}. The initial congruence and truncation precision give

$$\tau - t_n \in p^{L+n}\mathbb{Z}_p^k.$$

Subtracting consecutive relations gives compatibility in $\mathbb{Z}_p$. Faithfulness of integer divisibility under $\mathbb{Z} \hookrightarrow \mathbb{Z}_p$ gives compatibility in $\mathbb{Z}$, and the same norm bounds give $t_n \to \tau$. Theorem 10.14 and completion now produce b with exact polynomial vanishing and all retained precisions. Each free coordinate has both the corresponding coordinate of b and that of τ as its limit, proving exact parameter equality.

The solved Jacobian remains surjective, so every shifted obstruction $[\overline{r} + K_{z_n} v]$ vanishes. Formula (10.10) therefore gives p^{m-e} choices for every parameter digit at every stage. When $m = e$, the fixed-coordinate chart in the proof of Theorem 10.12 contains every root in the initial ball with the specified parameters. Its uniqueness proves the endpoint assertion. This is uniqueness of the actual endpoint, not of the integer representatives used to approach it. □

The endpoint theorems are `Komorebi.PolynomialPrescribedPadicTower.exists_prescribed_padic_root_with_tower` and `Komorebi.PolynomialPrescribedPadicTower.exists_unique_prescribed_padic_root`. The separate supplied-path theorem `Komorebi.PolynomialPrescribedPadicTower.exists_root_with_prescribed_path` accepts any compatible integer parameter path with limit τ. For $m > e$, endpoint uniqueness is false in general: the system $F(Y_1, Y_2, T) = Y_1$ has two distinct roots $(0, 0, 0)$ and $(0, 2, 0)$ in the same dyadic ball with identical parameter value.

10.7 Weighted lifting for a nonunit Jacobian

The remaining theorems in this section allow arbitrary coefficients and centers in $R = \mathbb{Z}_p$. They concern square solved Jacobians invertible over $\mathbb{Q}_p$, even when their reductions modulo p are singular. The exponent controlling an integral inverse is retained explicitly.

Theorem 10.16 (Weighted nonlinear lifting). *Let p be prime, $d, s, r \in \mathbb{N}$, $F \in R[X_1, \dots, X_d]^d$, and $a \in R^d$. Set $J = DF(a)$. Suppose*

$$J \in \mathrm{GL}_d(\mathbb{Q}_p), \qquad p^s J^{-1} \in M_d(R), \qquad F(a) \in p^r R^d, \qquad r > 2s. \tag{10.11}$$

Then there is exactly one root b of F in $a + p^{s+1}R^d$, and it satisfies

$$b - a \in p^{r-s} R^d. \tag{10.12}$$

Uniqueness holds throughout the full ball of depth $s + 1$.

Proof. Let $A = p^s J^{-1} \in M_d(R)$. Inclusion into $\mathbb{Q}_p$ and its injectivity give the two integral identities

$$AJ = JA = p^s I_d.$$

These identities are also an equivalent integral certificate for the first two hypotheses of (10.11). In particular, A is injective: $Aw = 0$ implies $p^s w = JAw = 0$, hence $w = 0$.

Precondition the polynomial system by A and set $c = p^{s+1}$. Its derivative at a is $AJ = p^s I_d$. Lemma 10.1 gives a polynomial vector Q with $Q(0) = 0$ and

$$(AF)(a + cu) = (AF)(a) + p^{2s+1}u + p^{2s+2}Q(u).$$

Since A is integral, $(AF)(a) \in p^r R^d$. The strict integer inequality $r > 2s$ permits $k = r - (2s + 1) \geq 0$ and a vector $C \in p^k R^d$ such that $(AF)(a) = p^{2s+1}C$. The normalized root equation is therefore

$$u = -C - pQ(u). \tag{10.13}$$

Integral polynomial evaluation is 1-Lipschitz for the supremum p-adic metric: a common divisibility of input differences is preserved under addition and multiplication by integral elements. Consequently the right side of (10.13) has contraction constant $|p|_p < 1$ on the complete space R^d. It has a unique fixed point u.

To obtain the sharper displacement, begin with $u \in R^d$. If $u \in p^n R^d$, then $Q(0) = 0$ and polynomial congruence give $Q(u) \in p^n R^d$. For $n < k$, both C and $pQ(u)$ are divisible by p^{n+1}. Equation (10.13) therefore improves the divisibility of u by one digit. Induction gives $u \in p^k R^d$. Thus $b = a + p^{s+1}u$ satisfies (10.12), because $s + 1 + k = r - s$. Its normalized preconditioned value vanishes, and injectivity of A gives $F(b) = 0$.

Every competing root $z \in a + p^{s+1}R^d$ can be written $z = a + p^{s+1}w$. Its root equation implies the same fixed-point equation after cancellation of the nonzero scalar p^{2s+1}. Contraction uniqueness gives $w = u$, proving uniqueness on the entire stated ball. □

The existence and uniqueness statements are `Komorebi.PolynomialWeightedLift.exists_weighted_root` and `Komorebi.PolynomialWeightedLift.exists_unique_weighted_root`. The inverse hypothesis in Lean

specifies an integral matrix whose image is exactly $p^s J^{-1}$. It does not substitute a potentially larger determinant valuation for s.

Example 10.17 (Strict threshold and displacement boundary)**.** For $p = 2$, $F(X) = X^2 + 3$, and $a = 1$, one has $F(a) = 4$ and $DF(a) = 2$. Thus $s = 1$ and $r = 2$ satisfy every displayed hypothesis except the strict inequality. No root exists in $\mathbb{Z}_2$, since none exists modulo eight. Hence $r \geq 2s$ cannot replace $r > 2s$ uniformly. Also, $F(X) = p^s X + p^r$ has root $-p^{r-s}$, whose displacement is not divisible by p^{r-s+1}. The guaranteed displacement cannot uniformly gain a digit.

10.8 First-order coefficient and parameter motion

Theorem 10.18 (Weighted coefficient deformation)**.** *Let $F, H \in R[X_1, \ldots, X_d]^d$, $F(a) = 0$, and $J = DF(a)$. Suppose $J \in \mathrm{GL}_d(\mathbb{Q}_p)$, $p^s J^{-1} \in M_d(R)$, and $M > 2s$. For $G = F + p^M H$, there is exactly one root $b \in a + p^{s+1}R^d$. It satisfies $b - a \in p^{M-s}R^d$ and, in $\mathbb{Q}_p^d$,*

$$b - a = -p^M J^{-1} H(a) + \varepsilon, \qquad \varepsilon \in p^{2M-3s} R^d. \tag{10.14}$$

Only the original Jacobian $DF(a)$ is subject to an inverse hypothesis.

Proof. Retain the original certificate $A = p^s J^{-1}$ and put $c = p^{s+1}$. Taylor expansion and a polynomial difference identity yield polynomial vectors Q_F, Q_H, both zero at the origin, such that

$$(AF)(a + cu) = p^{2s+1}u + p^{2s+2}Q_F(u), \qquad (AH)(a + cu) = AH(a) + cQ_H(u).$$

Consequently

$$(AG)(a + cu) = p^{2s+1}\big(C + u + pQ(u)\big),$$

where

$$C = p^{M-(2s+1)}AH(a), \qquad Q = Q_F + p^{M-s-1}Q_H, \qquad Q(0) = 0.$$

All exponents are nonnegative. The same contraction and divisibility argument as in Theorem 10.16 gives the root, the displacement p^{M-s}, and full-ball uniqueness. This constructs the root without introducing an inverse hypothesis for $DG(a)$.

For the error estimate, let $\delta = b - a$ and $D = M - s$. Taylor's theorem for the preconditioned polynomial gives

$$E_F = (AF)(b) - (AF)(a) - p^s\delta \in p^{2D}R^d.$$

Polynomial congruence gives $E_H = (AH)(b) - (AH)(a) \in p^D R^d$. The equations $F(a) = 0$ and $G(b) = 0$ imply

$$p^s\delta + p^M AH(a) = -E_F - p^M E_H \in p^{2D}R^d,$$

because $M + D \geq 2D$. Cancel p^s in the integral domain R to obtain

$$\delta + p^{M-s}AH(a) \in p^{2M-3s}R^d.$$

This integral vector is ε. After inclusion into $\mathbb{Q}_p$, $p^{M-s}A = p^M J^{-1}$ gives (10.14). □

The complete theorem is `Komorebi.PolynomialRootMotion.exists_unique_root_motion`. The error depth exceeds the displacement depth by $M - 2s \geq 1$. It is a guaranteed divisibility, not an assertion of exact valuation for every system.

Theorem 10.19 (Joint coefficient and parameter deformation)**.**

Let $F, H \in R[Y_1, \dots, Y_e, T_1, \dots, T_k]^e$ and $a = (y_0, t_0) \in R^{e+k}$ satisfy $F(a) = 0$. Define the actual blocks

$$J = D_Y F(a) \in M_e(R), \qquad K = D_T F(a) \in M_{e\times k}(R).$$

Suppose $J \in \mathrm{GL}_e(\mathbb{Q}_p)$, $p^s J^{-1} \in M_e(R)$, and $M > 2s$. For every $v \in R^k$, set

$$\tau = t_0 + p^M v, \qquad G = F + p^M H.$$

There is exactly one $y \in y_0 + p^{s+1}R^e$ with $G(y, \tau) = 0$. It satisfies

$$\begin{aligned} &y - y_0 \in p^{M-s}R^e, \\ &y - y_0 = -p^M J^{-1}\big(H(a) + Kv\big) + \varepsilon, \qquad \varepsilon \in p^{2M-3s}R^e. \end{aligned} \tag{10.15}$$

The equality involving J^{-1} is in $\mathbb{Q}_p^e$, with an integral error vector. The parameter equality is exact.

Proof. Use the augmented polynomial systems

$$\mathcal{F}(Y, T) = (F(Y, T), T - t_0), \qquad \mathcal{H}(Y, T) = (H(Y, T), -v).$$

Then $\mathcal{F}(a) = 0$, and its derivative is

$$B = D\mathcal{F}(a) = \begin{pmatrix} J & K \\ 0 & I_k \end{pmatrix}.$$

For $A = p^s J^{-1} \in M_e(R)$ and $c = p^s$, define

$$C = \begin{pmatrix} A & -AK \\ 0 & cI_k \end{pmatrix}.$$

The identities $AJ = JA = cI_e$ give $CB = BC = cI_{e+k}$. In the second product, the upper right block is $-JAK + cK = 0$. In the first, it is $AK - AK = 0$. Thus augmentation preserves the original inverse bound s.

Theorem 10.18 gives a unique root b of $\mathcal{F} + p^M\mathcal{H}$ in $a + p^{s+1}R^{e+k}$, with displacement depth $M - s$. Its last equations say exactly $b_T = t_0 + p^M v = \tau$. Its first equations are $G(b) = 0$. The integral first-order expression uses

$$C\begin{pmatrix} H(a) \\ -v \end{pmatrix} = \begin{pmatrix} A(H(a) + Kv) \\ -p^s v \end{pmatrix}.$$

Projection onto the solved coordinates gives

$$b_Y - y_0 + p^{M-s}A(H(a) + Kv) \in p^{2M-3s}R^e.$$

The same inclusion identity as before converts this to (10.15).

Finally, if $z \in y_0 + p^{s+1}R^e$ is any competing solved root with parameters τ, then (z, τ) belongs to the full augmented ball. Indeed, $M > 2s$ implies $M \geq s + 1$. Full-point uniqueness therefore gives $(z, \tau) = b$. This comparison proves uniqueness on exactly the solved ball stated in the theorem. □

The exact solved-coordinate statement is `Komorebi.PolynomialJointDeformation.exists_joint_deformation`, with its unique-existence interface `Komorebi.PolynomialJointDeformation.exists_unique_joint_deformation`.

The triangular inverse identities hold over arbitrary commutative rings, so they apply in particular over the p-adic integers.

Example 10.20 (First-order sign and error precision)**.** For $p = 2$, $s = 0$, $M = 2$, $F(Y, T) = Y + T$, $H = 1$, $a = 0$, and $v = 1$, the prescribed parameter is $\tau = 4$ and the solved root is $y = -8$. Here $J = K = 1$, and the forcing term is $H(a) + Kv = 2$. Omitting the parameter term would predict -4, and reversing its sign would predict 0. The corresponding errors -4 and -8 are both outside $2^{2M}\mathbb{Z}_2 = 16\mathbb{Z}_2$. Thus either change would violate the stated error bound.

The error exponent cannot uniformly be increased. For $F(Y, T) = Y + Y^2 + T$, $H = 0$, $a = 0$, $p = 2$, $s = 0$, $M = 1$, and $v = 1$, there is an even root y of $y + y^2 + 2 = 0$. Its equation forces $y \equiv 2 \pmod 4$ and $y + 2 = -y^2$, so

$$4 \mid y + 2, \qquad 8 \nmid y + 2.$$

Thus the error depth $2M - 3s = 2$ is attained even for parameter-only motion. The same quadratic with a constant coefficient perturbation gives the coefficient-only boundary. These examples refute a uniform extra error digit. They do not assert optimality for every individual choice of p, s, M.

The combined results establish finite singular obstructions, smooth compatible towers, full continuous root charts after a suitable coordinate choice, and quantitative deformation for a rationally invertible solved Jacobian. Rationally rank-deficient persistence remains a separate problem. The analytic solution-map theorem in the next section supplies the regularity conclusion for these same derivative blocks.

11 Analytic regularity of the local solution map

The deformation estimates control how far an exact root moves under a small perturbation. We now show that its local dependence on the parameters is analytic, and compute the derivative from the polynomial Jacobian. The analytic argument takes place over a complete normed field. For integral p-adic data, we first include the coefficients and base point in $\mathbb{Q}_p$. $\mathbb{Z}_p$ itself is not a vector space over $\mathbb{Q}_p$.

Let $\mathbb{K}$ be a complete nontrivially normed field. Let $e, k \in \mathbb{N}$, and write the variables as $(Y, T) \in \mathbb{K}^e \times \mathbb{K}^k$. For

$$F \in \mathbb{K}[Y_1, \dots, Y_e, T_1, \dots, T_k]^e, \qquad a = (y_0, t_0),$$

define the matrices of evaluated formal partial derivatives

$$J = D_Y F(a) \in M_e(\mathbb{K}), \qquad K = D_T F(a) \in M_{e\times k}(\mathbb{K}). \tag{11.1}$$

Rows index equations and columns index variables. We first establish that these formal derivatives are the derivatives of the evaluation map.

Lemma 11.1 (Derivative of polynomial evaluation)**.** *For $P \in \mathbb{K}[X_1, \dots, X_n]$, polynomial evaluation is analytic at every point. At $a \in \mathbb{K}^n$ its strict Fréchet derivative is*

$$v \longmapsto \sum_{j=1}^{n} (\partial_j P)(a) v_j.$$

For a finite system, its strict derivative is the continuous linear map represented by the matrix of evaluated formal partial derivatives.

Proof. For the derivative statement, use polynomial induction on constants, addition, and multiplication by a variable. Constants have zero derivative. The sum rule agrees with the additivity of formal partial differentiation. For PX_j, the product rule gives

$$d(PX_j)_a(v) = a_j\, dP_a(v) + P(a)v_j,$$

which agrees with the evaluated formal product rule for every coordinate. The finite product rule for maps to coordinate spaces then assembles the system derivative. Each matrix defines a continuous linear map because the coordinate spaces are finite-dimensional over the complete field. Polynomial analyticity follows from the corresponding polynomial induction in Mathlib and its finite-coordinate assembly. □

The derivative bridge and analytic statements are `Komorebi.PolynomialAnalyticEvaluation.hasStrictFDerivAt_evalSystem`, `Komorebi.PolynomialAnalyticEvaluation.analyticAt_evalSystem`, and `Komorebi.PolynomialAnalyticEvaluation.analyticOnNhd_evalSystem`.

Lemma 11.2 (The augmented derivative)**.** *If $J \in \mathrm{GL}_e(\mathbb{K})$, the polynomial map*

$$\Phi(Y, T) = (F(Y, T), T)$$

has invertible strict derivative at a, with

$$D\Phi(a) = \begin{pmatrix} J & K \\ 0 & I_k \end{pmatrix}, \qquad D\Phi(a)^{-1} = \begin{pmatrix} J^{-1} & -J^{-1}K \\ 0 & I_k \end{pmatrix}. \tag{11.2}$$

The local inverse supplied by this derivative is analytic near $\Phi(a)$.

Proof. The lower coordinates of Φ are the parameter variables. Differentiating them gives the blocks 0 and I_k. Lemma 11.1 identifies the remaining blocks with (11.1). Multiplication of the two matrices in (11.2), in both orders, gives the identity. The resulting linear equivalence is continuous, as is its inverse. Apply the strict inverse-function theorem to Φ. The analytic inverse theorem applies to the same open partial homeomorphism, because Φ is analytic and its derivative is that continuous linear equivalence. It supplies a convergent local power-series representation of this inverse. □

The augmented Jacobian is identified in `Komorebi.PolynomialJointSystem.jacobian_augmented`. The continuous inverse block and its restriction are `Komorebi.PolynomialAnalyticBlocks.upperEquiv` and `Komorebi.PolynomialAnalyticBlocks.inverse_solved_zero`. Analyticity of the same constructed inverse is `Komorebi.PolynomialAnalyticInverse.analyticAt_inverse`.

Theorem 11.3 (Analytic local root graph)**.** *Suppose $F(a) = 0$ and $J \in \mathrm{GL}_e(\mathbb{K})$. There are open neighborhoods $t_0 \in U \subseteq \mathbb{K}^k$ and $y_0 \in V \subseteq \mathbb{K}^e$, and a map $\psi : \mathbb{K}^k \to \mathbb{K}^e$, such that*

$$\psi(t_0) = y_0, \qquad \psi(U) \subseteq V, \qquad \psi \text{ is analytic at every point of } U, \tag{11.3}$$

$$F(y, t) = 0 \iff y = \psi(t) \qquad (t \in U,\ y \in V). \tag{11.4}$$

At the base parameter, the strict Fréchet derivative is

$$\boxed{D\psi(t_0) = -J^{-1}K.} \tag{11.5}$$

In particular, ψ is differentiable on U.

Proof. Let g be the analytic local inverse of Φ supplied by Lemma 11.2. Since $F(a) = 0$, $\Phi(a) = (0, t_0)$ and $g(0, t_0) = a$. Define

$$\psi(t) = \pi_Y g(0, t),$$

where π_Y projects to the solved coordinates. This gives the base value and analyticity at t_0. The chain rule and (11.2) give

$$D\psi(t_0)v = \pi_Y \begin{pmatrix} J^{-1} & -J^{-1}K \\ 0 & I_k \end{pmatrix} \begin{pmatrix} 0 \\ v \end{pmatrix} = -J^{-1}Kv.$$

To obtain the root equivalence on a product neighborhood, we choose the neighborhoods as follows. The source of the open partial homeomorphism defined by Φ is open around a. Its inverse is g. Choose open sets V and U_0 containing y_0 and t_0 such that $V \times U_0$ lies in that source. Restrict U_0 further by the preimage of an open analytic inverse neighborhood under $t \mapsto (0, t)$, the open locus where this same ψ is analytic, and the interior of $\psi^{-1}(V)$. All of these sets contain t_0. For the last set, use the continuity of ψ at t_0 and $\psi(t_0) = y_0 \in V$. Their intersection is an open neighborhood U with $\psi(U) \subseteq V$.

If $t \in U$, $y \in V$ and $F(y, t) = 0$, then $\Phi(y, t) = (0, t)$. The left inverse identity forces $g(0, t) = (y, t)$, hence $y = \psi(t)$. Conversely, the right inverse identity $\Phi(g(0, t)) = (0, t)$ says both that its solved equations vanish and that its parameter coordinates are exactly t. Thus $g(0, t) = (\psi(t), t)$ and $F(\psi(t), t) = 0$. Analyticity at every point of U implies differentiability there. □

This is `Komorebi.PolynomialAnalyticImplicit.exists_analytic_implicit`. Differentiability is also included explicitly in the statement contract `Komorebi.PolynomialRegularityContracts.actual_padic_analytic_root_graph`.

Theorem 11.4 (Regularity of a supplied root selection)**.** *Under the hypotheses of Theorem 11.3, let $h : \mathbb{K}^k \to \mathbb{K}^e$ be continuous at t_0, with $h(t_0) = y_0$. Suppose $F(h(t), t) = 0$ for every t in some neighborhood of t_0. Then h and ψ agree in a neighborhood of t_0. Consequently h is analytic there as a germ and has strict derivative $Dh(t_0) = -J^{-1}K$.*

Proof. Continuity at t_0 places $h(t)$ inside V for all sufficiently close parameters. Intersect this neighborhood with U and the neighborhood on which the root equation holds. Equation (11.4) gives $h(t) = \psi(t)$ throughout the intersection. Analyticity and strict differentiability at a point are unchanged by agreement on a neighborhood. □

The formal result is `Komorebi.PolynomialAnalyticSelection.continuous_selection_regularity`. Its formulation derives $F(a) = 0$ from the root equation near the base parameter and the specified base value.

Corollary 11.5 (Regularity throughout an open parameter set)**.** *Let $U \subseteq \mathbb{K}^k$ be open, and let h be continuous on U. Suppose that for every $t \in U$,*

$$F(h(t), t) = 0, \qquad J_t = D_Y F(h(t), t) \in \mathrm{GL}_e(\mathbb{K}).$$

Then h is analytic at every point of U and its strict derivative is

$$Dh(t) = -J_t^{-1} D_T F(h(t), t) \qquad (t \in U).$$

Proof. At each $t \in U$, openness turns continuity on U into continuity at t and makes the root equation hold on a neighborhood of t. Apply Theorem 11.4 at $(h(t), t)$. □

This is `Komorebi.PolynomialAnalyticSelection.continuous_selection_analyticOnNhd`. The pointwise invertibility assumption is part of this corollary.

Corollary 11.6 (Integral coefficients and centers)**.** *Let p be any prime. Suppose $F \in \mathbb{Z}_p[Y,T]^e$, $a \in \mathbb{Z}_p^{e+k}$, and $F(a) = 0$. Let $\iota : \mathbb{Z}_p \hookrightarrow \mathbb{Q}_p$ be the inclusion. If $\iota(D_Y F(a)) \in \mathrm{GL}_e(\mathbb{Q}_p)$, the polynomial system obtained by including the original coefficients in $\mathbb{Q}_p$ has the analytic local root graph of Theorem 11.3 at $\iota(a)$. The strict derivative at the base is*

$$-\iota(D_Y F(a))^{-1}\iota(D_T F(a)).$$

The open neighborhoods and the output map are in the ambient spaces over $\mathbb{Q}_p$.

Proof. Coefficient inclusion commutes with evaluation and formal partial differentiation. Consequently both Jacobian blocks after inclusion are the entrywise inclusions of the original blocks, and the original exact root remains an exact root. Apply Theorem 11.3 to the system with included coefficients. □

This is `Komorebi.PolynomialRegularityIntegral.exists_analytic_integral_implicit`. It requires no invertibility modulo p. In particular, it includes solved Jacobians singular modulo p but invertible over the fraction field.

Example 11.7 (The derivative sign and a nonlinear control)**.** For $F(Y,T) = Y + T$, the exact branch is $\psi(T) = -T$ and $J = K = 1$. The derivative is -1. Uniqueness of the derivative rules out $+1$ over $\mathbb{Q}_p$, including $p = 2$. The nonlinear polynomial $F(Y,T) = Y + Y^2 + T$ has $F(0,0) = 0$ and $J = K = 1$ there. Applying the formal theorem to this polynomial constructs an analytic local root graph with derivative -1. Thus the example checks the existence conclusion as well as the derivative formula. The declarations and the deliberately rejected positive-sign statement are in the regularity control and statement-contract files.

The results cover arbitrary finite e and k, including zero. The local inverse machinery explicitly handles a zero-dimensional domain. The analytic theorem does not assign a uniform numerical radius to its neighborhoods. The earlier weighted lifting and joint deformation theorems retain their quantitative radii and error bounds, while the present statements establish regularity of the local solution germ. A direct Lean identification with each subtype-valued chart is an additional interface task. General persistence at a fraction-field singular solved block is a separate problem.

12 Open questions and further directions

The arithmetic results above resolve Nishibuchi's conjecture throughout its stated parameter range. The local analyticity and differentiability theorem is also complete under its stated invertibility hypothesis. What remains is a set of extensions and sharper classifications, together with the broader questions raised by the original native seed-calculus program.

12.1 Singular persistence beyond an invertible solved block

The weighted lifting theorem allows the solved Jacobian to be noninvertible modulo p provided that it is invertible over $\mathbb{Q}_p$ with a controlled inverse. This includes a substantial class of lifts often called singular modulo p. The next problem is to understand Jacobians that remain rank-deficient over $\mathbb{Q}_p$, where the inverse construction is no longer available.

For a fixed polynomial system $F \in \mathbb{Z}[X_1, \ldots, X_d]^e$, a precision $L \geq 1$, and a root $\bar{x}$ modulo p^L, let

$$\mathcal{L}_{F,L}(\bar{x}) = \{x' \in (\mathbb{Z}/p^{L+1}\mathbb{Z})^d : F(x') = 0,\ x' \equiv \bar{x} \pmod{p^L}\}.$$

The finite obstruction theorem decides whether this set is empty from the residual digit and reduced Jacobian. When it is nonempty, the theorem also determines its cardinality and affine kernel structure. The conclusion concerns one level: vanishing of the obstruction permits the next step, but does not guarantee that subsequent steps are possible.

For example, $X^2 - 5$ has the root 1 modulo 2, and both 1 and 3 lift it to roots modulo 4. Every odd square is 1 modulo 8, so neither branch continues to a root modulo 8. Thus even a successful first correction can lead to branches that stop at the following level.

We seek structural hypotheses and higher obstruction data that control an entire singular branch, including its behavior under coefficient and parameter deformation. One route is to propagate suitable conditions from level to level. Another is to prove existence at all levels in a form that yields compatible branches. The finite obstruction calculus supplies the tests for individual levels, but a persistence theorem needs a further argument linking them. Any general formulation must allow multiple branches, later obstructions, and changes in the rank of the Jacobian.

12.2 Native 13.2: presentation invariance

The original conjecture asks for a canonical finite obstruction packet whose vanishing is necessary and sufficient for a one-level lift in a finitely presented arithmetic seed system. For a fixed finite polynomial presentation, the chapter supplies the precise obstruction

$$o_{F,\bar{x}} = [r] \in \operatorname{coker}\left(J_{F,\bar{x}} : \mathbb{F}_p^d \longrightarrow \mathbb{F}_p^e\right),$$

where r is the residual digit. We have proved independence from the chosen integer representative. Invariance under a change of equations or presentation requires more.

The raw cokernel already reveals the difficulty. The presentations $F = (X)$ and $G = (X, 0)$ describe the same zero locus over every coefficient ring. At zero their Jacobians are, respectively,

$$\mathbb{F}_p \xrightarrow{1} \mathbb{F}_p, \qquad \mathbb{F}_p \xrightarrow{(1,0)} \mathbb{F}_p^2.$$

Their raw cokernels are 0 and $\mathbb{F}_p$. Both distinguished obstruction classes vanish, but the ambient obstruction groups are not isomorphic. The example rules out the raw cokernel as a presentation-independent object while leaving open the possibility of an intrinsic obstruction construction.

A precise form of Native 13.2 therefore needs four ingredients:

1. a category of seed objects, including how the operators, relations, projectors, weights, and topology enter an object;
2. specified changes of presentation and a proof that they preserve the actual lift problem;
3. an obstruction object that respects dependencies among the relations, together with comparison maps or a universal property;
4. a precise meaning of “finite” and “canonical”, and a proof that vanishing is equivalent to the intended one-step continuation.

For commutative algebra presentations, the relation module offers a natural way to account for dependencies among listed equations. To extend this approach to the full seed vocabulary, we would need an encoding that preserves the lift problem, together with a proof of that preservation. This is one of the links still needed between the concrete obstruction theorem and Native 13.2.

12.3 Native 13.3: finite compatibility, completion, and descent

The original proposed form is

$$X \simeq X_2^{\mathrm{wt}} \times_C {\prod_{p>2}}' X_p.$$

It leaves X, the distinguished subsets defining the restricted product, the maps to C, the ambient category, and the universal meaning of "smallest" unspecified. In particular, one must decide whether X is a global object or a completed object.

The linear compatibility theorem provides one concrete part of this program. If two local observations $J : U \to V$ and $K : U \to W$ are linear maps from the *same* correction module, define

$$C_{J,K} = U/(\ker J + \ker K).$$

The attainable observations $\operatorname{im} J$ and $\operatorname{im} K$ have canonical labels in this quotient. Equality of labels is equivalent to simultaneous realization by a single correction. The joint image is exactly the fiber product over $C_{J,K}$, with the proved linear universal property. The declarations are `Komorebi.SharedCorrection.labels_eq_iff_common`, `Komorebi.SharedCorrection.joint_range_eq_fiber`, and `Komorebi.SharedCorrection.common_universal`.

To apply this result arithmetically, one must construct the common correction module and the observation maps, then show that the resulting compatibility condition agrees with the intended arithmetic one.

Why the global/completed distinction is necessary. Consider constant sequences with integer initial value. Their global parameter space is $\mathbb{Z}$, while their local parameter spaces are $\mathbb{Z}_p$. Under the canonical local maps, the image of the global objects is the diagonal

$$D = \{(n)_p : n \in \mathbb{Z}\} \subseteq \prod_p \mathbb{Z}_p.$$

This diagonal is dense: a basic open set specifies finitely many congruences, and the Chinese remainder theorem provides an integer satisfying them. It is proper: the tuple with 2-adic coordinate zero and all odd-adic coordinates one is not the image of an integer. Indeed an integer zero in $\mathbb{Z}_2$ must be zero, contradicting its 3-adic coordinate.

Consequently this canonical global diagonal cannot be an equality locus

$$\{(a, b) : f(a) = g(b)\} \subseteq \mathbb{Z}_2 \times \prod_{p>2} \mathbb{Z}_p$$

for continuous maps to a Hausdorff space C. Such an equality locus is closed, as the inverse image of the closed diagonal in $C \times C$, whereas D is proper and dense. This topological argument, retained from the native progress note, concerns the stated topological interpretation. It has not been formalized in Lean for this manuscript.

Finite CRT reconstruction leads to the familiar completed object $\widehat{\mathbb{Z}} = \prod_p \mathbb{Z}_p$, but does not identify which local tuples come from a single global integer. A complete form of Native 13.3 must therefore specify finite-level compatibility, transition maps and completion, and any additional global descent condition. If recurrence, spectral, or Frobenius information is to supply that condition, it must be expressed through defined maps or constraints on the local data, with a precise account of how they enter C.

The accompanying `native/Native_Conjectures_Progress.md` records further results in restricted settings: an intrinsic finite Ext^1 obstruction for finitely presented commutative integer algebras, finite CRT reconstruction for polynomial solution sets, and a characterization of rectangular finite relations by equality of common labels. The obstruction result is a stated specialization of a classical external theorem. The CRT and rectangularity results have prose proofs in the note. These are preserved mathematical progress, with the restrictions and source records given there, rather than Lean completions of the general Native 13.2 and 13.3 statements.

12.4 Arithmetic refinements

Write $U_{N,j}(n) = j!C_{N,j}(n)$ and $U_j = U_{4,j}$, and use $v_p(0) = \infty$ when discussing valuation bounds. The following classification questions remain open in this program.

1. **The remaining higher-difference classification.** The new strengthening chapters determine every repeated-step valuation in addresses $j = 0, 1$, give the exact criterion for equality in the common mixed-step lower bound there, and determine address-two valuations at all power-of-two orders. What remains is a complete exact formula in address two at the other orders, a full classification in address three, and the amount of additional cancellation in mixed-step cases where the leading term vanishes. Numerical patterns through a finite range do not settle these questions.

2. **Upper-half addresses at larger dyadic rank.** The exact first-difference laws now include $N = 2, 4, 8$ and every lower-half address for $N = 2^a \geq 4$. Larger upper-half addresses need a finer classification. Explicit counterexamples show that extrapolating one low-order Newton coefficient to a universal metric law is false. A least period is measured in packet rows. Its Euler-degree displacement is NP.

3. **Exact affine Mahler-kernel valuations.** For the kernel of Section 2.7, determine exact values of

$$v_p\big(\Theta_{M,s}(r,k)\big), \qquad \Theta_{M,s}(r,k) = [z^r](1+z)^s((1+z)^M - 1)^k,$$

   beyond the explicit special families. The minimum valuation of the summands gives a lower bound. Equality requires a separate argument excluding cancellation.

4. **Classical tail root classification and simplicity.** The later tail program gives an exact factorization, finite preparation for each fixed depth, and several infinite regular-wall and simplicity families, as summarized in Appendix A. A universal explicit root classification and universal simplicity of the remaining analytic factor are still open. In the exceptional address-one quadratic reduction, the residual task is a nonvanishing discriminant condition. Empty tails remain exactly zero. This problem is distinct from the solved Nishibuchi congruence conjecture.

These refinements build on the proved estimates and exact cases, including the sharp uniform classical dyadic bound and the resolution of Nishibuchi's conjecture.

A Exact classical Euler tails and the remaining simplicity problem

This appendix states the proved reductions for classical Euler tails and the remaining simplicity problem. That problem is distinct from Nishibuchi's Conjecture 3.9, proved by the congruence results of this chapter, and from the presentation and gluing questions Native 13.2 and Native 13.3.

The tail results have mathematical proofs, supplemented in the indicated families by complete coefficient certificates and proved bounds for every omitted coefficient. Bounded sampled values support implementation checks. They do not supply the universal quantifiers. The latest continuation did not run Lean. Its mathematical claims must therefore be read with the formal-coverage distinctions in the Lean guide.

A.1 The ordinary Euler packet identity

To distinguish ordinary signed Euler numbers from the generalized numbers of the main text, write

$$\operatorname{sech} t = \sum_{n\ge 0} \mathcal{E}_n \frac{t^n}{n!}, \qquad \frac{t}{e^t-1} = \sum_{n\ge 0} B_n \frac{t^n}{n!}, \qquad B_1 = -\frac{1}{2}.$$

Throughout this appendix, $v_2(2) = 1$ and $v_2(0) = \infty$. Let $s_2(n)$ be the number of ones in the binary expansion of the nonnegative integer n. Define

$$A(N) = \frac{3^{2N+2}-1}{4(2N+1)(2N+2)}, \qquad \left(\frac{z}{\sinh z}\right)^2 = \sum_{m\ge 0} \beta_m z^{2m}, \qquad \kappa_m = 2^{2m}(2m)!\beta_m.$$

The initial values are $\beta_0 = \kappa_0 = 1$. For $m \ge 1$, differentiation of $z\coth z$ gives

$$\beta_m = -\frac{2^{2m}(2m-1)B_{2m}}{(2m)!}, \qquad \kappa_m = -2^{4m}(2m-1)B_{2m}. \tag{A.1}$$

The classical dyadic Bernoulli denominator law $v_2(B_{2m}) = -1$ and Legendre's formula yield

$$v_2(\beta_m) = s_2(m) - 1, \qquad v_2(\kappa_m) = 4m - 1. \tag{A.2}$$

The source identity

$$\sum_{N\ge 0} A(N)\frac{t^{2N}}{(2N)!} = \frac{\sinh(2t)\sinh t}{2t^2}$$

implies, after multiplication by $4t^2/\sinh^2(2t)$,

$$\mathcal{E}_{2N} = \sum_{m=0}^{N} \kappa_m \binom{2N}{2m} A(N-m). \tag{A.3}$$

For a retained depth $M \ge 0$, put

$$\mathcal{T}_M(N) = \mathcal{E}_{2N} - \sum_{m=0}^{\min(M,N)} \kappa_m \binom{2N}{2m} A(N-m).$$

An empty tail, $N \le M$, is exactly zero.

Proposition A.1 (Sharp uniform classical truncation bound)**.** *For every $M, N \ge 0$,*

$$v_2\big(\mathcal{T}_M(N)\big) \ge 4M + 3.$$

For each fixed M, equality holds at $N = M + 1$.

Proof. The source $A(n)$ is a dyadic unit for every nonnegative integer n. Each nonempty summand has $m \ge M + 1$, so (A.2) and integrality of the binomial coefficient give the asserted bound. At $N = M + 1$, only the first omitted summand remains. Its binomial coefficient and $A(0)$ both equal one. □

Thus later row-dependent cancellation can improve a particular tail, but cannot increase the uniform exponent required to hold at every row.

A.2 A finite carry minimum and an exact all-row factorization

Let $\mathbb{Z}_2\langle x\rangle$ denote the restricted power series: their integral coefficients tend to zero dyadically. Extend A by

$$\mathscr{A}(x) = \frac{9^{x+1} - 1}{8(x+1)(2x+1)}, \qquad \mathscr{A}(-1) = -\frac{\log 9}{8}.$$

The apparent singularity at $x = -1$ is removable. Since $v_2(\log 9) = 3$, its convergent expansion is

$$\mathscr{A}(x) = \frac{\log 9/8}{1 + 2x} \sum_{h \ge 0} \frac{(\log 9)^h}{(h+1)!} (x+1)^h. \tag{A.4}$$

It follows coefficientwise that $\mathscr{A} \in \mathbb{Z}_2\langle x\rangle^{\times}$, $\mathscr{A} \equiv 1 \pmod 2$, and $\mathscr{A}(N) = A(N)$.

Set $r = M + 1 \ge 1$ and define

$$P_q(x) = \prod_{i=0}^{q-1} (x - i)(2x - (2i+1)), \qquad P_0 = 1,$$

$$H_r(x) = \sum_{q \ge 0} 8^q \frac{\beta_{r+q}}{\beta_r} P_q(x) \mathscr{A}(x - q).$$

The coefficient valuations tend to infinity. At any nonnegative integer x, terms with $q > x$ vanish. Cancelling the factorials in (A.3) therefore gives the exact identity

$$\mathcal{T}_{r-1}(N) = \kappa_r \binom{2N}{2r} H_r(N - r) \qquad (N \ge r). \tag{A.5}$$

This is a finite rational identity at every original Euler row.

The scalar packet weights have exact valuations

$$\lambda_r(q) = 3q + s_2(r + q) - s_2(r).$$

Put

$$\mu_r = \min_{q \ge 0} \lambda_r(q), \qquad J_r = \{q : \lambda_r(q) = \mu_r\}, \qquad G_r = 2^{-\mu_r} H_r.$$

Since $\lambda_r(0) = 0$ and $\lambda_r(q) \geq 3q + 1 - s_2(r)$, every minimizing address satisfies

$$0 \leq q \leq \left\lfloor \frac{s_2(r) - 1}{3} \right\rfloor. \tag{A.6}$$

In particular $\mu_r \leq 0$, and the minimum is found by a finite carry calculation. Reduction of the whole series gives

$$\overline{G}_r(X) = \sum_{q \in J_r} X^{\underline{q}} \quad \text{in } \mathbb{F}_2[X]. \tag{A.7}$$

Its highest degree is $\max J_r$, with leading coefficient one. Thus the normalization is the actual Gauss normalization.

Theorem A.2 (Exact classical tail valuation)**.** *For every $r \geq 1$ and $N \geq r$,*

$$v_2\big(\mathcal{T}_{r-1}(N)\big) = 4r - 1 + v_2\binom{N}{r} + \mu_r + v_2 G_r(N - r). \tag{A.8}$$

There are unique factors

$$G_r(X) = W_r(X)U_r(X),$$

where $W_r \in \mathbb{Z}_2[X]$ is monic of degree $d_r = \max J_r$ and $U_r \in \mathbb{Z}_2\langle X\rangle^\times$ satisfies $U_r \equiv 1 \pmod 2$. Consequently $v_2 G_r(x) = v_2 W_r(x)$ for every $x \in \mathbb{Z}_2$, and

$$\deg W_r \leq \left\lfloor \frac{s_2(r) - 1}{3} \right\rfloor.$$

Proof. Equation (A.5), the exact weight in (A.2), and the digit-sum identity $v_2\binom{2N}{2r} = v_2\binom{N}{r}$ prove (A.8). The binomial coefficients themselves need not be equal.

For preparation, start from the monic polynomial $g = \overline{G}_r$ and its zero-one lift W_1, with $U_1 = 1$. Suppose $G_r - W_k U_k$ is coefficientwise divisible by 2^k. Its divided reduction is a finite polynomial because G_r is restricted. Divide it by g in $\mathbb{F}_2[X]$:

$$\overline{(G_r - W_k U_k)/2^k} = gq + s, \qquad \deg s < d_r.$$

With zero-one coefficient lifts, set $W_{k+1} = W_k + 2^k s$ and $U_{k+1} = U_k + 2^k q$. The error is now divisible by 2^{k+1}, the polynomial remains monic of degree d_r, and the unit reduces to one. The sequences converge in the Gauss norm. Uniqueness of quotient and remainder at successive binary digits proves uniqueness of the factors. For $d_r = 0$, take $W_r = 1$. The degree bound follows from (A.6). □

This theorem reduces every depth to a finite polynomial without discarding the infinite analytic tail. It does not explicitly classify every W_r, decide every repeated root from a fixed precision, or prove that all its dyadic roots are simple.

A.3 Regular walls: exact root emergence

For a wall label $j \geq 1$ and $0 \leq q < j$, define

$$e_j(q) = 3q + s_2(j - 1) - s_2(j - q - 1), \qquad E_j = \min_{q<j} e_j(q),$$

$$\Theta(j) = 3j + 1 + s_2(j-1), \qquad L_j = \Theta(j) - E_j.$$

Here E_j is a carry minimum, not an Euler number. A wall label is *regular* when $e_j(q) > 0$ for $1 \le q < j$. For $r = 2^L - j$ with $L > L_j$, put $\delta = L - L_j$. The carry identities give

$$\mu_r = E_j - \delta, \qquad J_r = \{j\}, \qquad G_r(x) = P_j(x)U(x-j) + 2^\delta V(x),$$

where $U \equiv 1 \pmod 2$. For a regular label, also $E_j = 0$ and $V \equiv 1 \pmod 2$.

For $0 \le i < j$, let

$$d_i = v_2(i!) + v_2((j-1-i)!), \qquad k_i = 1 + \lfloor \log_2 \max(i, j-1-i) \rfloor,$$

$$B_i = i + 2^{k_i}\mathbb{Z}_2, \qquad t_i = d_i + k_i.$$

At $j = 1$ use $d_0 = k_0 = t_0 = 0$. The balls B_i partition $\mathbb{Z}_2$. Indeed a residue represented twice among $0, \dots, j-1$ splits into its two children. Every other represented residue remains one ball. On each such ball,

$$P_j(i + 2^{k_i}y) = 2^{t_i} y Q_i(y), \qquad Q_i \in \mathbb{Z}_2\langle y \rangle^\times, \qquad Q_i \equiv 1 \pmod 2.$$

Theorem A.3 (Regular-wall root law)**.** *For every regular label j and every integer $\delta \ge 1$,*

$$v_2 G_r(x) = \begin{cases} \delta, & x \in B_i,\ \delta < t_i, \\ d_i + v_2(x - \alpha_i), & x \in B_i,\ \delta \ge t_i. \end{cases}$$

In the second case there is exactly one root α_i in B_i, it is simple, and $v_2(\alpha_i - i) = \delta - d_i$. Thus the number of dyadic roots is exactly

$$\#\{i : 0 \le i < j,\ t_i \le \delta\}.$$

Proof. If $\delta < t_i$, the unit perturbation has smaller valuation than the skeleton term throughout B_i, so the valuation is exactly δ. Otherwise, after the ball substitution and division by 2^{t_i}, the coefficientwise reduction is $y + c$ for a constant $c \in \mathbb{F}_2$. A restricted integral series with this reduction is a dyadic isometry: division of its difference by $y - z$ gives an odd linear coefficient plus even higher contributions. Binary lifting therefore gives its unique root and exact distance law. Evaluating at $x = i$ gives the stated displacement, since $G_r(i) = 2^\delta V(i)$ has exact valuation δ. The partition proves exhaustion of the roots. □

The largest emergence threshold is

$$\max_i t_i = v_2((j-1)!) + \lceil \log_2 j \rceil.$$

For $j = 2^a$, all thresholds equal $j - 1$. There are no roots before that distance and exactly j simple roots at and after it, within the positive-distance range. The equality and pre-wall cases are also resolved: regular labels $j \ge 2$ have no roots there. $j = 1$ has one simple odd root at equality.

For example, $j = 5$ has $(t_0, \dots, t_4) = (6, 3, 4, 3, 6)$. The exact root counts at distances $1, 2, 3, 4, 5, \ge 6$ are $0, 0, 2, 3, 3, 5$. The older sufficient stages $3, 5, 7$ are not the sharp emergence stages, which are $3, 4, 6$. The first irregular label, $j = 17$, has $V \equiv 1 + X \pmod 2$. The regular-wall theorem does not apply. Complete coefficient certificates close distances 1 through 19, and the general separation theorem gives all seventeen simple roots for $\delta \ge 20$. Those two kinds of evidence together resolve that family.

A.4 What the later simplicity reductions prove

For a general depth r, call $j_r = \max J_r$ its *true terminal*. This definition depends on the actual carry minimum. Rewriting the same integer as another wall expression does not change j_r. The universal simplicity target is

$$\text{for every integer } r \geq 1 \text{ and } x \in \mathbb{Z}_2, \qquad G_r(x) = 0 \implies G_r'(x) \neq 0. \tag{A.9}$$

The following are completed restricted results of the later continuations. Their family statements include a proof that the indicated label is the true terminal throughout the displayed ranges.

Depth r	Odd multiplier range	Outer scales
$B2^L - 129$	$0 < B < 2^{89}$	$L \geq 392$
$B2^L - 161$	$0 < B < 2^{932}$	$L \geq 487$

All dyadic roots are simple in both infinite integer families. For arbitrary larger admissible multipliers, a multiple root with true terminal 129 can occur only at $L = 515$ and in two necessary multiplier classes modulo 2^{90}. With true terminal 161, it can occur only at $L = 640$ and in one necessary multiplier class modulo 2^{935}. These classes are necessary conditions. The records do not assert existence of a singular ordinary Euler index in them.

For terminal 161, the neighboring phases are resolved uniformly: $L = 639$ has exactly one simple root and $L = 641$ exactly three simple roots, for every positive odd multiplier with that true terminal. The middle phase is reduced, before the deeper certificate is applied, to

$$B \equiv 17659 \pmod{2^{18}}, \qquad x \equiv 10202 \pmod{2^{15}}.$$

The 935-bit multiplier condition composes this whole-phase reduction with an inherited 917-bit local parameter certificate. It is neither a new 935-bit reconstruction nor an all-precision interpolation of the undepleted Euler family.

The structural argument also extends beyond lower address one. For a certified true-terminal wall, let a be its last lower minimizing address, $k = j_r - a$ the regular complement, and $t = v_2(k)$. Under the regular lower-address hypotheses of the source theorem, all roots above emerged complement balls are simple. Every root is simple at every wall distance in the band

$$\Theta(a) \leq t < \Theta(a) + v_2(a!).$$

Equality walls are simple even outside that band. This includes certified terminal-258 walls with $a = 2$, $k = 256$, and $L \geq 777$. For $a = 3$, $k = 4096$, only $L = 16385$ remains a possible critical scale. Outside the band, the theorem gives necessary local valuation conditions. It does not exclude the remaining unit cancellation. Irregular lower addresses need their own prefix analysis.

A.5 The actual stationary discriminant

The current focused obstruction is the admitted ordinary-index family

$$r = (17659 + 2^{18}T)2^{640} - 161, \qquad T \in \mathbb{Z}_{\geq 0}, \qquad j_r = 161. \tag{A.10}$$

On its surviving chart define the *actual*, undepleted function

$$K_r(Z) = 2^{-178} G_r(10202 + 2^{15} Z).$$

The complete coefficient theorem gives

$$K_r(Z) = a_r + 2b_r Z + u_r Z^2 + 256 Z^3 R_r(Z), \quad a_r, b_r \in \mathbb{Z}_2, \quad u_r \in \mathbb{Z}_2^{\times}, \quad R_r \in \mathbb{Z}_2\langle Z \rangle. \tag{A.11}$$

Its half derivative is a dyadic isometry. It therefore has a unique critical point $c_r \in \mathbb{Z}_2$. Set $\Phi_r^{\mathrm{act}} = K_r(c_r)$. The earlier independent-undepletion notation for this same scalar is $\Phi_r(2^{2r-25})$. Excluding a multiple root on the surviving chart is exactly the assertion $\Phi_r^{\mathrm{act}} \neq 0$.

Proposition A.4 (Exact quadratic compression)**.** *There is a unique preparation*

$$K_r(Z) = \mathcal{U}_r(Z)\mathcal{W}_r(Z), \qquad \mathcal{W}_r(Z) = Z^2 + 2\beta_r Z + \alpha_r, \qquad \mathcal{U}_r \equiv u_r \pmod{256}.$$

With $\delta_r = \beta_r^2 - \alpha_r$ and $\tau_r = \mathcal{U}_r'(c_r)/\mathcal{U}_r(c_r)$,

$$\delta_r = -\frac{\Phi_r^{\mathrm{act}}}{\mathcal{U}_r(c_r)} \left(1 - \frac{\tau_r^2 \Phi_r^{\mathrm{act}}}{4\mathcal{U}_r(c_r)}\right), \qquad v_2(\delta_r) = v_2(\Phi_r^{\mathrm{act}}). \tag{A.12}$$

The equality includes the zero case.

Proof. The same successive monic polynomial division used in Theorem A.2, started at precision eight, preserves the quadratic and unit reductions of (A.11). Write $W = Z^2 + 2\beta_r Z + \alpha_r$ and $w = W(c_r)$. Stationarity gives $W'(c_r) = -\tau_r w$, while $w = \Phi_r^{\mathrm{act}}/\mathcal{U}_r(c_r)$. The quadratic identity $\beta_r^2 - \alpha_r = W'(c_r)^2/4 - W(c_r)$ proves the formula. Since $\tau_r \in 256\mathbb{Z}_2$, its bracket is a unit. □

This compression retains the unit-cofactor derivative. It does not assert that c_r, α_r, or β_r is rational or algebraic over $\mathbb{Q}$. The remaining arithmetic obligation is therefore

$$\delta_r \neq 0 \quad \text{for every admitted ordinary index in (A.10).} \tag{A.13}$$

Even this assertion would still leave the terminal-129 and higher-address obligations in the universal target (A.9).

A.6 Approximation reaches every depth, but separation remains open

The latest continuation retains the exact Bernoulli undepletion factor

$$e_{r,q} = \frac{1 - 2^{2r-1}}{1 - 2^{2r-1} 4^q}, \qquad q \geq 0.$$

It is a dyadic unit, including $e_{r,0} = 1$. Multiplication by this factor preserves the all-q error of the shifted Jacobi moment model. Consequently the actual model critical values satisfy, for every $m \geq 25$,

$$v_2\big(\Phi_{r,m}^{\mathrm{act}} - \Phi_r^{\mathrm{act}}\big) \geq L_m = 8m - 4s_2(m) - 178, \qquad L_m \longrightarrow \infty. \tag{A.14}$$

The estimate includes every packet and controls motion of the critical point. There is no ceiling at the undepletion exponent $2r - 25$. For each fixed admitted r it implies the exact equivalence

$$\Phi_r^{\mathrm{act}} \neq 0 \quad \Longleftrightarrow \quad \text{there exists } m \geq 25 \text{ such that } v_2(\Phi_{r,m}^{\mathrm{act}}) < L_m. \tag{A.15}$$

Indeed a passing strict test has the same valuation as the limit. Conversely any finite limit valuation is eventually smaller than L_m. The existence of a passing depth for every ordinary index is unproved. Nonzero models at every depth are compatible with an exactly zero limit.

The degree-two preparation also reduces the old representative-height cost to a linear cost, about $2P$ bits at precision P. This remains larger than the P bits needed for a strict nonzero residue certificate. The fixed rational control $(Z - 1/3)^2$ has zero critical value while its least-residue integer discriminant representatives are nonzero at every even $P \geq 6$. Thus a nonzero integer representative alone cannot prove a nonzero dyadic discriminant.

One sufficient new arithmetic input would be an odd-denominator projective quadratic

$$AZ^2 + 2BZ + C \equiv A\mathcal{W}_r(Z) \pmod{2^P}, \qquad A \text{ odd}, \qquad 0 < |B^2 - AC| < 2^P,$$

at a proved precision for every admitted ordinary r. Then $(B^2 - AC)/A^2 \equiv \delta_r \pmod{2^P}$ cannot vanish modulo 2^P. No depth-independent theorem supplying this inequality has been established. The tested recurrence and exterior-contact identities retain balancing terms and do not force a contradiction. Finite exponential models approximate the actual target at every depth, but their order grows with precision, their clearing polynomial has zeros on the actual chart, and their proved construction-specific height cost does not give the required discriminant bound.

A.7 Status and source map

Claim	Status
Nishibuchi Conjecture 3.9 in its stated parameter range	Resolved in the main text
Sharp uniform ordinary Euler tail exponent $4M + 3$	Proved, with formal coverage
Exact tail factorization and finite polynomial at every depth	Proved by an ordinary argument
Regular-wall emergence and indicated integer simplicity families	Proved within their stated hypotheses
Actual quadratic compression and unbounded stationary approximation	Ordinary proof. Formal scope separate.
Nonzero actual stationary discriminant for every admitted index	Open
Universal simplicity of every G_r	Open
Native 13.2 presentation invariance and Native 13.3 global/completed gluing	Separate open general programs

The tail factorization, regular-wall arguments, finite preparation, coefficient cutoffs and higher-address transversality have separate proofs. The stationary discriminant identity and convergence argument leave the arithmetic separation question open. The source notes and their exact locations are collected at the back.

B Detailed continuation derivations

The following sections develop the detailed arguments behind polynomial lifting, correction torsors, compatible towers and local regularity. Each construction retains its stated hypotheses. The formal companion identifies the corresponding declarations and verification records. The main text distinguishes the completed results from the remaining questions.

B.1 Chapter IX: finite polynomial lifting and shared corrections

The nonlinear Taylor argument supplies the correction formula used in one-step polynomial lifting. The statements concern a fixed finite list of integer polynomials. They hold for every positive integer modulus, including composite moduli and modulus one. No Jacobian-surjectivity assumption is needed.

B.1.1 Statement and notation

Let

$$F = (F_1, \dots, F_e) \in \mathbb{Z}[X_1, \dots, X_d]^e, \quad q \geq 1, \quad L \geq 1, \quad x \in \mathbb{Z}^d, \quad F(x) \equiv 0 \pmod{q^L}.$$

Write $R_q = \mathbb{Z}/q\mathbb{Z}$, let ρ_q denote coordinatewise reduction, and put

$$r_x = F(x)/q^L \in \mathbb{Z}^e, \qquad A_x = DF(x) : \mathbb{Z}^d \longrightarrow \mathbb{Z}^e, \qquad J_x = \rho_q \circ A_x.$$

Here A_x is computed from the actual formal partial derivatives of the input polynomials. The Lean map J_x has domain $\mathbb{Z}^d$ and codomain R_q^e, viewed as integer modules. It factors through reduction to R_q^d. The cokernel below is presented directly using its integer-domain image.

Define the finite obstruction module and its distinguished class by

$$\mathcal{O}_x = R_q^e / \operatorname{im} J_x, \qquad o_x = [\rho_q(r_x)] \in \mathcal{O}_x.$$

The checked criterion is

$$\boxed{o_x = 0 \iff \exists y \in \mathbb{Z}^d : \quad y \equiv x \pmod{q^L}, \qquad F(y) \equiv 0 \pmod{q^{L+1}}.}$$

The residual exists by the lower congruence and is unique because $q > 0$. The quotient is finite since its ambient module R_q^e is finite. The theorem permits zero variables or zero equations as well.

B.1.2 The checked Taylor bridge

For every $P \in \mathbb{Z}[X_1, \dots, X_d]$, integer a, and $x, t \in \mathbb{Z}^d$, Lean proves

$$\boxed{a^2 \mid P(x + at) - P(x) - a \sum_{i=1}^{d} (\partial_i P)(x) t_i.}$$

The proof uses polynomial induction: constants, addition, and multiplication by a variable. The product rule for the actual partial derivatives controls the multiplication step. It does not assume a Taylor identity or a liftability condition.

Taking $a = q^L$, the inequality $2L \geq L + 1$ gives

$$F(x + q^L t) \equiv q^L (r_x + A_x t) \pmod{q^{L+1}}.$$

Since $q^L \neq 0$ in the integers, cancellation in divisibility yields

$$F(x + q^L t) \equiv 0 \pmod{q^{L+1}} \iff \rho_q(r_x) + J_x t = 0.$$

Every integer representative of the lower residue has the form $x + q^L t$, and this correction equation has a solution exactly when $o_x = 0$.

The positive-depth hypothesis is necessary for the general nonlinear theorem. For $F(X) = X^2 - 1$, $q = 2$, $L = 0$, $x = 0$, the Jacobian is zero and the residual class is nonzero, yet $y = 1$ is a solution modulo two in the same residue modulo one. Affine systems retain the previously checked $L \geq 0$ theorem.

B.1.3 Independence of the integer representative

Suppose $x' = x + q^L s$ represents the same lower residue, with exact residual $r_{x'}$. Polynomial evaluation respects coordinatewise congruences, including when applied to each partial derivative. Hence

$$J_{x'} = J_x, \qquad \rho_q(r_{x'}) = \rho_q(r_x) + J_x s.$$

The two cokernels therefore have the same ambient module and the same defining submodule, and

$$[\rho_q(r_{x'})] = [\rho_q(r_x)].$$

Lean states this using one explicit common reduced map, avoiding an implicit comparison between differently indexed quotient types. This establishes independence of the integer representative within the fixed polynomial presentation.

Changing the polynomial presentation is a separate question. Appending a zero equation can change the raw cokernel by adding a copy of R_q, even though it does not change the solutions.

B.1.4 Two polynomial blocks and one shared lift

Let $G \in \mathbb{Z}[X_1, \ldots, X_d]^f$ be a second fixed system with the same variables and the same lower root x. Let $K_x = \rho_q \circ DG(x)$, and define

$$\mathcal{C}_x = \mathbb{Z}^d / (\ker J_x + \ker K_x).$$

This quotient is finite: it is a quotient of the finite image of J_x, or equivalently of the finite image of K_x.

Suppose $u, v \in \mathbb{Z}^d$ are separately valid corrections:

$$F(x + q^L u) \equiv 0 \pmod{q^{L+1}}, \qquad G(x + q^L v) \equiv 0 \pmod{q^{L+1}}.$$

Then

$$\boxed{\begin{array}{c} [u] = [v] \text{ in } \mathcal{C}_x \iff \exists y \equiv x \pmod{q^L} : \\ F(y) \equiv 0 \pmod{q^{L+1}}, \quad G(y) \equiv 0 \pmod{q^{L+1}}. \end{array}}$$

The labels are independent of the choices of valid u and v. An unconditional version asks for separately valid corrections with equal labels. It does not assume that each block is liftable.

Taylor transport turns a common correction t into the simultaneous equations $J_x t = J_x u$ and $K_x t = K_x v$. The previously checked common-quotient theorem gives precisely the displayed equivalence. Separate solvability alone does not imply joint solvability. The controls include $F(X) = X$, $G(X) = X - 2$, starting at $x = 0$ modulo two and lifting to modulus four.

B.2 Chapter IX: correction torsors and exact polynomial lift counts

The correction-digit torsor gives exact prime-field lift counts. The counts apply to distinct next-level residue classes satisfying the actual polynomial equations. They do not count integer representatives, of which there can be infinitely many.

B.2.1 The theorem

Let p be prime, $L \geq 1$, and

$$F \in \mathbb{Z}[X_1, \ldots, X_d]^e, \qquad x \in \mathbb{Z}^d, \qquad F(x) \equiv 0 \pmod{p^L}.$$

Write

$$r_x = F(x)/p^L, \qquad b_x = r_x \bmod p, \qquad A_x = DF(x) \bmod p : \mathbb{F}_p^d \longrightarrow \mathbb{F}_p^e, \qquad \rho_x = \operatorname{rank}_{\mathbb{F}_p} A_x.$$

The actual lift set is

$$\mathcal{L}_{x,L} = \left\{ y \in (\mathbb{Z}/p^{L+1}\mathbb{Z})^d : y \equiv x \pmod{p^L}, \quad F(y) = 0 \right\},$$

where polynomial evaluation in the displayed set is modulo p^{L+1}. Define

$$\mathcal{O}_x = \mathbb{F}_p^e / \operatorname{im} A_x, \qquad o_x = [b_x].$$

Lean proves

$$\boxed{\#\mathcal{L}_{x,L} = \begin{cases} p^{d-\rho_x}, & o_x = 0, \\ 0, & o_x \neq 0, \end{cases} \qquad \#\mathcal{O}_x = p^{e-\rho_x}.}$$

These formulas permit singular Jacobians and zero variables or zero equations. Nonsingularity is not assumed. In particular, rank alone gives the size of a nonempty lift fiber. The residual obstruction determines whether it is empty.

B.2.2 Why these are actual residue-class counts

For every positive integer q, including composite moduli and modulus one, and every $L \geq 0$, the checked map

$$\Phi_{x,L} : (\mathbb{Z}/q\mathbb{Z})^d \longrightarrow \{ y \bmod q^{L+1} : y \equiv x \pmod{q^L} \}, \qquad t \longmapsto x + q^L \widetilde{t}$$

is a bijection. Here $\widetilde{t}$ is the canonical integer representative of the digit vector. The entire residue fiber therefore has q^d elements. Injectivity is proved by cancelling q^L in the corresponding integer divisibility statement. Surjectivity comes from the lower-residue congruence.

The formal type `ActualLifts` is a subtype of these genuine `ZMod (q^(L+1))` residue vectors. Its polynomial predicate is evaluated on canonical integer representatives and tested for divisibility by

q^{L+1}. The checked polynomial congruence theorem makes that predicate independent of the integer representative. Thus using canonical representatives does not discard or duplicate residue classes.

At positive depth, the preceding actual multivariable Taylor theorem gives

$$F(x + p^L\widetilde{t}) \equiv p^L(r_x + DF(x)\widetilde{t}) \pmod{p^{L+1}}.$$

The new bridge proves that the evaluated integer Jacobian reduces to the actual $\mathbb{F}_p$-linear map A_x. Restricting $\Phi_{x,L}$ therefore gives an explicit equivalence

$$\boxed{\{t \in \mathbb{F}_p^d : b_x + A_x t = 0\} \simeq \mathcal{L}_{x,L}.}$$

The proof uses the actual formal partial derivatives of the original polynomial list.

B.2.3 The torsor

Let $\mathcal{T}_x = \{t : b_x + A_x t = 0\}$. Its natural kernel action is

$$k \cdot t = k + t, \qquad k \in \ker A_x.$$

For every pair $u, t \in \mathcal{T}_x$, exactly one kernel vector sends u to t, namely $t - u$. Thus every nonempty correction fiber is a torsor for $\ker A_x$. Choosing one valid digit u gives

$$\mathcal{T}_x \simeq \ker A_x, \qquad t \longmapsto t - u.$$

Lean constructs an explicit `AddTorsor` structure for the digit fiber, proves the unique translation property, and supplies a chosen-point equivalence from the actual residue lift set to the kernel. A basepoint is required for this equivalence, but the translation action itself does not depend on choosing one.

The torsor and residue bijection remain valid over every positive modulus at positive polynomial depth. The displayed powers involving vector-space rank are restricted to prime moduli.

B.2.4 Connecting the old obstruction to the new count

The earlier obstruction used the integer-domain map

$$J_x = \rho_p \circ DF(x) : \mathbb{Z}^d \to \mathbb{F}_p^e.$$

The new count uses $A_x : \mathbb{F}_p^d \to \mathbb{F}_p^e$. Reduction of integer vectors onto finite vectors is surjective, so the two maps have the same image after restricting scalars. Lean constructs a class-preserving integer-linear equivalence

$$\mathbb{F}_p^e / \operatorname{im} J_x \simeq \mathbb{F}_p^e / \operatorname{im} A_x.$$

Consequently the cardinality theorem counts the original obstruction group, preserving its distinguished residual class. It is not merely a formula for a separately introduced comparison object.

Rank-nullity gives $\dim \ker A_x = d - \rho_x$, and the quotient dimension is $e - \rho_x$. Cardinalities of finite vector spaces complete both formulas.

B.3 Chapter IX: compatible polynomial towers and p-adic roots

The nonsingular construction passes from one-step polynomial lifting to an infinite compatible tower and an actual p-adic root. Surjectivity is required only of the reduced Jacobian at the initial root. The proof derives its persistence, constructs the tower, proves convergence and verifies the original polynomial equations at the limit.

B.3.1 Main theorem

Let p be prime, $L \geq 1$, and

$$F \in \mathbb{Z}[X_1, \ldots, X_d]^e, \qquad x \in \mathbb{Z}^d, \qquad F(x) \equiv 0 \pmod{p^L}.$$

Assume the actual reduced Jacobian

$$A_x = DF(x) \bmod p : \mathbb{F}_p^d \longrightarrow \mathbb{F}_p^e$$

is surjective. Then Lean proves that there is a point

$$\boxed{a \in \mathbb{Z}_p^d, \qquad F(a) = 0, \qquad a \equiv x \pmod{p^L}.}$$

Here the coefficients of the original integer polynomials are cast into $\mathbb{Z}_p$, and their multivariable evaluation is exactly zero. This is an existence theorem for a fixed polynomial presentation with the stated Jacobian hypothesis.

The stronger checked statement constructs $z_n \in \mathbb{Z}^d$, $n \geq 0$, with

$$\begin{aligned} z_0 &= x, \\ F(z_n) &\equiv 0 \pmod{p^{L+n}}, \\ z_{n+1} &\equiv z_n \pmod{p^{L+n}}, \\ DF(z_n) \bmod p &= A_x, \\ z_n &\longrightarrow a \quad \text{in } \mathbb{Z}_p^d, \\ a - z_n &\in p^{L+n}\mathbb{Z}_p^d. \end{aligned}$$

The last inclusion retains every prescribed precision. It is a divisibility lower bound, not an assertion that the error has exactly that valuation. An approximation can already equal the limiting root.

B.3.2 Why the single initial hypothesis suffices

The one-step Taylor theorem already supplies the exact correction equation

$$b_n + A_{z_n} t = 0, \qquad b_n = F(z_n)/p^{L+n} \bmod p.$$

Surjectivity supplies a correcting digit for every residual b_n. The chosen correction gives an actual new integer point z_{n+1}, using the proved nonlinear Taylor remainder.

Every coordinate change is divisible by p^{L+n}, hence by p. Each formal partial derivative is itself an integer polynomial, so its value is unchanged modulo p. Therefore

$$A_{z_{n+1}} = A_{z_n} = A_x.$$

The proof assumes no separate surjectivity or solvability statement at later levels. A dependent recursion chooses each next point from the proved one-step existence theorem. It produces one compatible sequence, rather than unrelated witnesses at different precisions.

The integer tower construction works more generally for every positive modulus q, including composite q and $q = 1$, with a surjective map over $\mathbb{Z}/q\mathbb{Z}$. The p-adic completion and the rank-based branching formulas in this block use prime p.

B.3.3 Convergence and exact polynomial vanishing

Compatibility implies, for $m \geq n$,

$$p^{L+n} \mid z_m - z_n.$$

After casting to $\mathbb{Z}_p$, this bounds each coordinate distance by $p^{-(L+n)}$, so the sequence is Cauchy. Completeness supplies its limit. Closed norm bounds show that the limit retains the congruence at every stage.

The generic convergence theorem is proved independently of the polynomial application. It also proves

$$p^{L+n} \mid v_n \text{ for all } n \quad \Longrightarrow \quad v_n \longrightarrow 0 \text{ in } \mathbb{Z}_p.$$

Apply this to each integer sequence $F_i(z_n)$. Continuity of multivariable polynomial evaluation also gives $F_i(z_n) \to F_i(a)$. Uniqueness of limits then yields $F_i(a) = 0$.

The cast-evaluation identity is explicitly checked, so the continuous function is the evaluation of the original polynomials. Convergence and polynomial vanishing are conclusions, not assumptions of the main existence theorem.

B.3.4 Exact number of choices at each level

Surjectivity implies

$$\operatorname{rank}_{\mathbb{F}_p} A_x = e, \qquad e \leq d.$$

Combining this with the preceding exact residue-lift count gives

$$\boxed{\#\mathcal{L}_{z_n, L+n} = p^{d-e} \quad \text{for every } n.}$$

These are distinct solutions modulo p^{L+n+1} lying above the selected point modulo p^{L+n}. The original finite obstruction group at each smooth point has cardinality one.

The theorem constructs one path through these finite choices. When $d > e$, it does not claim that all paths have the same limit. The generic uniqueness theorem concerns the limit of a fixed sequence only.

B.3.5 Boundary controls

The positive control theorems and rejected invalid proofs cover three distinctions.

1. **An initial lift does not guarantee an infinite tower without the Jacobian condition.** $X^2 - 5$ has the odd lower root modulo two and a lift modulo four, but no root modulo eight. Lean proves that no integer tower can continue through all precisions in this example.
2. **Surjective rectangular Jacobians do not give uniqueness of all roots.** The system $F(X, Y) = X$ has a surjective Jacobian and distinct dyadic roots $(0, 0)$ and $(0, 2)$, both in the same residue modulo two.

3. **The retained initial precision cannot be increased without justification.** For $F(X) = X - 2$, zero is a lower root modulo two with surjective derivative, but its exact root two does not retain zero modulo four.

The controls are theorem checks of these explicit examples. They supplement the universal proof audit and do not replace it.

For several polynomial blocks, the hypothesis applies to their combined system. Surjectivity of each block separately does not imply surjectivity of the combined Jacobian or shared liftability. The preceding shared-correction obstruction remains relevant in that situation.

B.4 Polynomial roots under coefficient perturbation

The argument proves uniqueness of roots for square systems and nearby-root stability for finite systems of integer polynomials. The polynomials and their partial derivatives are evaluated directly. No root-uniqueness or perturbation-stability premise is assumed.

B.4.1 Notation

Let p be prime, $L \geq 1$, and

$$F, H \in \mathbb{Z}[X_1, \ldots, X_d]^e, \qquad x \in \mathbb{Z}^d.$$

Write

$$A_x = DF(x) \bmod p : \mathbb{F}_p^d \longrightarrow \mathbb{F}_p^e, \qquad B_L(x) = x + p^L \mathbb{Z}_p^d.$$

For an integer c, the perturbed system is the actual polynomial list

$$G = F + cH.$$

There is no restriction on the sign of c, and $c = 0$ is allowed. Divisibility conclusions give lower bounds on precision and allow zero differences.

B.4.2 1. Unique roots for square systems

If $d = e$, A_x is surjective, and $F(x) \equiv 0 \pmod{p^L}$, then

$$\boxed{\exists!\, a \in B_L(x) : F(a) = 0.}$$

The stronger comparison lemma proves that any two exact roots of F in $B_L(x)$ coincide, without needing a separate lower-root hypothesis.

Proof implemented. Map every arbitrary p-adic root into the actual residue rings $\mathbb{Z}/p^n\mathbb{Z}$, and choose its canonical integer representative. Evaluation commutes with reduction, so these representatives are genuine finite polynomial roots. The preceding count theorem gives exactly $p^{d-d} = 1$ next-level residue lift. Inductively any two roots agree at every precision. The p-adic residue maps separate points, so the roots are equal. Existence is the preceding checked tower theorem.

This extends uniqueness from the limit of a selected sequence to **all exact roots in the prescribed ball**.

B.4.3 2. Coefficient perturbations preserve smooth liftability

If $p^L \mid c$, the actual identities give

$$G(x) = F(x) + cH(x), \qquad DG(x) \equiv DF(x) \pmod{p}.$$

Thus if $F(x) \equiv 0 \pmod{p^L}$ and A_x is surjective, then

$$\exists b \in B_L(x) : G(b) = 0.$$

This theorem allows rectangular systems. The evaluation, derivative and Jacobian identities are proved from the polynomial definition $G_i = F_i + C(c)H_i$, rather than supplied as hypotheses. Algebraic congruence helpers also permit composite moduli. Exact p-adic root existence uses prime p.

B.4.4 3. Nearby roots for rectangular systems

Let $M \geq 1$ and $p^M \mid c$. Suppose A_x is surjective and $a \in \mathbb{Z}_p^d$ is an exact root of F with $a \equiv x \pmod{p}$. Then

$$\boxed{\exists b \in \mathbb{Z}_p^d : \quad G(b) = 0, \qquad b \equiv a \pmod{p^M}.}$$

No uniqueness is claimed when $d > e$.

Proof implemented. Truncate a to an integer vector y modulo p^M. It is an actual lower root of F, and its reduced Jacobian agrees with A_x. Apply the coefficient-perturbed existence theorem at precision M, then combine the two congruences through y.

B.4.5 4. Full precision for the unique root of a square system

Suppose $d = e$, $M \geq L \geq 1$, $p^M \mid c$, and A_x is surjective. If

$$a, b \in B_L(x), \qquad F(a) = 0, \qquad G(b) = 0,$$

then

$$\boxed{b - a \in p^M \mathbb{Z}_p^d.}$$

In particular, for $G = F + p^M H$, no p-adic digits below depth M are changed in the selected root.

The formalization also proves a combined existence theorem. Starting only with $F(x) \equiv 0 \pmod{p^L}$, it constructs an original root $a \in B_L(x)$ and proves that there is exactly one $b \in B_L(x)$ satisfying $G(b) = 0$ and $b \equiv a \pmod{p^M}$. The comparison theorem shows that the precision bound holds for every root of G in that ball, so uniqueness is not obtained merely by adding a restrictive displacement condition.

Proof implemented. Section 3 produces a nearby root b'. Since $M \geq L$, it belongs to $B_L(x)$. The perturbed Jacobian remains surjective, and square uniqueness identifies b' with the arbitrary root b.

B.5 Chapter IX: continuous p-adic parameter charts

B.5.1 Main statement

Let p be prime, $L \geq 1$, $e, q \geq 0$, and

$$F \in \mathbb{Z}[X_1, \ldots, X_{e+q}]^e, \qquad x \in \mathbb{Z}^{e+q}, \qquad F(x) \equiv 0 \pmod{p^L}.$$

Assume the actual reduced Jacobian

$$A_x = DF(x) \bmod p : \mathbb{F}_p^{e+q} \longrightarrow \mathbb{F}_p^e$$

is surjective. All vector congruences are coordinatewise.

There is a coordinate permutation σ, with $(\sigma z)_i = z_{\sigma(i)}$. Write $\sigma z = (y, t)$, using e solved coordinates and q free coordinates. Define

$$(y_0, t_0) = \sigma x, \qquad B = x + p^L \mathbb{Z}_p^{e+q}, \qquad T = t_0 + p^L \mathbb{Z}_p^q, \qquad \mathcal{R} = \{z \in B : F(z) = 0\}.$$

There is a homeomorphism

$$h : T \xrightarrow{\sim} \mathcal{R}, \qquad h^{-1}(z) = \text{free}(\sigma z).$$

Equivalently, a continuous function $\phi : T \to y_0 + p^L \mathbb{Z}_p^e$ gives the entire root locus through $\sigma h(t) = (\phi(t), t)$. Every parameter has exactly one corresponding root in B. For every $N \geq 0$,

$$\boxed{h(t) \equiv h(u) \pmod{p^N} \iff t \equiv u \pmod{p^N}.}$$

The coordinate choice is a conclusion. A suitable minor and a preexisting chart are not hypotheses. The theorem covers the original root locus after undoing the coordinate reordering. Since x is an approximate root, it does not assert $\phi(t_0) = y_0$ unless $F(x) = 0$ exactly. The cases $p = 2$, $L = 1$, $e = 0$, and $q = 0$ are included.

B.5.2 Supporting theorems

For a rectangular integer polynomial system with a surjective reduced Jacobian at x, and $L \geq 1$,

$$F(x + p^L \mathbb{Z}_p^d) = F(x) + p^L \mathbb{Z}_p^e.$$

This covers arbitrary p-adic output targets. No lower-root hypothesis is needed.

When $d = e$, the actual restricted polynomial map is a homeomorphism. Moreover, for arbitrary $a, b \in x + p\mathbb{Z}_p^d$ and every $N \geq 0$,

$$F(a) \equiv F(b) \pmod{p^N} \iff a \equiv b \pmod{p^N}.$$

Its inverse therefore preserves every congruence depth.

B.5.3 Proof structure

1. The actual integer Taylor formula gives one additional input digit from one additional output digit when the reduced square Jacobian is surjective. Its finite domain and codomain have the same size, so it is injective. Induction gives every integer precision.
2. Canonical integer truncations of arbitrary p-adic vectors preserve evaluation modulo each power of p. Applying the integer result to truncations proves the p-adic congruence equivalence.
3. Subtracting an integer target from the constant terms leaves the actual Jacobian unchanged. The previous root-lifting theorem supplies a preimage. The image of the compact input ball is closed and contains the convergent truncations of every admissible p-adic target, hence contains the target. No compatibility of arbitrarily chosen preimages is assumed.

4. In the square case, the resulting continuous bijection from a compact ball to a Hausdorff ball is a homeomorphism. Its forward map is actual polynomial evaluation, and its inverse solves the output equation.
5. For a specified surjective solved-variable block, apply this inverse to the actual augmented polynomial map
$$\Psi(y,t) = (F(y,t),t).$$
Its derivative has an identity free-coordinate block. The inverse at $(0,t)$ gives the root. The lower-root congruence places this target in the output ball. Injectivity proves uniqueness and full graph coverage. Projection identifies the inverse chart and reflects congruences.
6. The columns of a surjective finite-dimensional linear map contain a basis. Select those columns and extend their injection to a coordinate permutation. Polynomial renaming by σ^{-1} transports actual evaluations and partial derivatives. The fixed-coordinate chart then gives the main theorem on the entire original root locus.

B.5.4 Scope

Coefficients are integers. Roots and parameters are arbitrary p-adic points. This supplies continuous polynomial parameter charts. Analyticity, differentiability, arbitrary p-adic coefficient interfaces, weighted lifting for nonunit Jacobians, and first-order root-motion estimates are separate.

Full Jacobian surjectivity need not allow an arbitrary prescribed parameter choice. At the dyadic origin, $F(X,T) = X^2 - T$ has a surjective full Jacobian but zero X-derivative. Prescribing $T = 2$ gives no root even modulo 4. Suitable coordinates instead solve for T.

B.5.5 Verification and provenance

The eight new mathematical modules contain 75 declarations: 60 theorems and 15 definitions. The active polynomial dependency set contains 25 mathematical modules. Counts including inherited support and theorem contracts are recorded separately in the final report.

The exact universal statements are Lean theorems, not conclusions from sampled numerical tests. The fresh gate rebuilds project sources with warnings treated as errors, checks expanded theorem contracts, audits explicit declarations, and checks rejected invalid extensions. Only propext, Classical.choice, and Quot.sound are permitted as axiom dependencies.

B.6 Chapter IX: weighted nonlinear lifting

B.6.1 The theorem

Let p be prime, $R = \mathbb{Z}_p$, $K = \mathbb{Q}_p$, and $d, s, r \in \mathbb{N}$. Let $F = (F_1, \ldots, F_d) \in R[X_1, \ldots, X_d]^d$, $a \in R^d$, and let
$$J = DF(a) = \left(\partial_j F_i(a)\right)_{i,j}.$$

Assume

$$J \in \mathrm{GL}_d(K), \qquad p^s J^{-1} \in M_d(R), \qquad F(a) \in p^r R^d, \qquad r > 2s.$$

Then

$$\boxed{\exists!\, b \in a + p^{s+1}R^d :\ F(b) = 0,} \qquad \boxed{b - a \in p^{r-s}R^d.}$$

All vector divisibility assertions are coordinatewise. The uniqueness predicate is the entire ball of depth $s+1$: competitors need not be assumed to have depth $r-s$. The theorem permits arbitrary integral p-adic coefficients and centers, every prime including two, and dimension zero. It retains the supplied inverse-denominator exponent s. There is no replacement by a determinant valuation. Integrality of $p^s J^{-1}$ is expressed in Lean by existence of an integral matrix with exactly that image under the coefficient inclusion $R \hookrightarrow K$.

B.6.2 Proof

Write $A = p^s J^{-1} \in M_d(R)$. Then

$$AJ = JA = p^s I.$$

The formal matrix bridge proves the equivalence with the stated fraction-field hypotheses, including nonsingularity. In particular, multiplication by A is injective, since multiplication by J sends $Av = 0$ to $p^s v = 0$.

Put $G = AF$, $t = p^{s+1}$, and $k = r - (2s+1)$. Taylor expansion of the actual preconditioned polynomials gives one polynomial system Q such that

$$G(a + tu) = G(a) + p^{2s+1}u + p^{2s+2}Q(u), \qquad Q(0) = 0.$$

This expansion is proved first over every commutative ring. Its remainder is a polynomial valid for every u, not a pointwise quotient depending on u. Since $G(a) \in p^r R^d$, it can be written $p^{2s+1}C$ with $C \in p^k R^d$. Thus

$$G(a + tu) = p^{2s+1}\big(C + u + pQ(u)\big).$$

Integral polynomial evaluation is 1-Lipschitz in the supremum p-adic metric. Consequently $T(u) = -C - pQ(u)$ has contraction constant $|p|_p < 1$ on the complete space R^d, and has exactly one fixed point u.

For completeness, the sharp precision is additional to this fixed-point existence. Inductively, if $u \in p^n R^d$, then polynomial congruence and $Q(0) = 0$ imply $Q(u) \in p^n R^d$. For $n < k$, the equation $u = -C - pQ(u)$ therefore gives $u \in p^{n+1}R^d$. Starting at $n = 0$, this proves $u \in p^k R^d$. Hence $b = a + tu$ satisfies

$$b - a \in p^{s+1+k}R^d = p^{r-s}R^d.$$

The normalization and injectivity of A give $F(b) = 0$.

Every other $z \in a + p^{s+1}R^d$ has the form $a + tw$. If $F(z) = 0$, the same normalization, with cancellation of the nonzero scalar p^{2s+1}, gives $w = T(w)$. Fixed-point uniqueness implies $w = u$, hence $z = b$.

B.6.3 Boundary and sharpness

The strict inequality cannot in general be weakened to $r \geq 2s$: for $p = 2$, $F(X) = X^2 + 3$, and $a = 1$, we have $F(a) = 4$, $DF(a) = 2$, and may take $s = 1, r = 2$. Yet no root exists in $\mathbb{Z}_2$, since there is none modulo eight. The auxiliary Lean controls prove this obstruction and the exact residual depth.

The displacement exponent cannot uniformly be increased by one. For $F(X) = p^s X + p^r$ at $a = 0$, with $r > 2s$, its root is $b = -p^{r-s}$, which is not divisible by p^{r-s+1}. The controls prove the linear root and the absence of that extra digit for all primes.

B.7 Chapter IX: weighted first-order root motion

B.7.1 The theorem

Let p be prime, $R = \mathbb{Z}_p$, $K = \mathbb{Q}_p$, and $d, s, M \in \mathbb{N}$. Let $F, H \in R[X_1, \dots, X_d]^d$, let $a \in R^d$ satisfy $F(a) = 0$, and set

$$J = DF(a), \qquad G = F + p^M H.$$

Assume

$$J \in \mathrm{GL}_d(K), \qquad p^s J^{-1} \in M_d(R), \qquad M > 2s.$$

Then there is exactly one root b of G in $a + p^{s+1}R^d$. It satisfies

$$b - a \in p^{M-s}R^d,$$

and there is $\varepsilon \in p^{2M-3s}R^d$ such that, in K^d,

$$\boxed{b - a = -p^M J^{-1} H(a) + \varepsilon.}$$

The inverse in this formula belongs to the original Jacobian $DF(a)$. The theorem constructs the perturbed root from these hypotheses and proves uniqueness throughout the full ball of depth $s + 1$. It does not require an additional Jacobian hypothesis on G. Coefficients and centers are arbitrary integral p-adic elements. Every prime and finite square dimension are included, including $p = 2$ and dimension zero.

The error depth exceeds the displacement depth by

$$(2M - 3s) - (M - s) = M - 2s \geq 1.$$

Thus the first-order term determines strictly more digits than the displacement bound alone. These are guaranteed divisibility bounds. Equality of valuations is not asserted for every system.

B.7.2 Root construction with the original inverse

Write $A = p^s J^{-1} \in M_d(R)$, so $AJ = JA = p^s I$. Let $t = p^{s+1}$ and $k = M - (2s + 1) \geq 0$. The actual polynomial Taylor and difference identities give polynomial systems Q_F, Q_H, both vanishing at zero, such that

$$(AF)(a + tu) = p^{2s+1}u + p^{2s+2}Q_F(u),$$
$$(AH)(a + tu) = (AH)(a) + tQ_H(u).$$

Consequently

$$(AG)(a + tu) = p^{2s+1}\big(C + u + pQ(u)\big),$$

where

$$C = p^k AH(a), \qquad Q = Q_F + p^{M-s-1}Q_H, \qquad Q(0) = 0.$$

All exponents are nonnegative under $M > 2s$. Actual integral polynomial evaluation is 1-Lipschitz, so $u \mapsto -C - pQ(u)$ is a strict contraction on the complete space R^d. Its unique fixed point inherits the depth k of C, because $Q(0) = 0$. Thus $b = a + tu$ has displacement depth $s + 1 + k = M - s$.

The certificate $JA = p^s I$ makes multiplication by A injective, so the normalized zero is an actual root of G. Every root in $a + tR^d$ has the form $a + tw$ and satisfies the same fixed-point equation. This proves uniqueness on the entire stated ball using the original matrix certificate.

B.7.3 The first-order error

Put $\delta = b - a$ and $L = M - s$. The displacement form of Taylor's theorem, proved over any commutative ring, gives

$$E_F = (AF)(b) - (AF)(a) - p^s\delta \in p^{2L}R^d.$$

Polynomial congruence gives

$$E_H = (AH)(b) - (AH)(a) \in p^L R^d.$$

Since $F(a) = 0$ and $G(b) = 0$,

$$p^s\delta + p^M AH(a) = -E_F - p^M E_H.$$

We have $M + L \geq 2L$, so the right side lies in $p^{2L}R^d$. Cancelling the nonzero scalar p^s proves the integral estimate

$$\delta + p^{M-s}AH(a) \in p^{2M-3s}R^d.$$

Define this integral vector to be ε. Under $R \hookrightarrow K$, the exact identity $p^{M-s}A = p^M J^{-1}$ gives the displayed first-order formula. The Lean bridge checks matrix evaluation and all coefficient inclusions explicitly.

B.7.4 Sharpness and the strict threshold

For $p = 2$, take $F(X) = X + X^2$, $H(X) = 1$, $a = 0$, $s = 0$, $M = 1$. There is a unique root $b \in 2\mathbb{Z}_2$ of $X + X^2 + 2$. Its equation gives $b + 2 = -b^2$. It also forces $b \equiv 2 \pmod 4$. Therefore

$$v_2(b + 2) = 2 = 2M - 3s.$$

The accompanying Lean controls show that the error is divisible by four and is not divisible by eight. This refutes a uniform replacement of $2M - 3s$ by $2M - 3s + 1$, even for a one-variable unit Jacobian.

The prior strict-threshold example also applies to perturbations: take $F(X) = X^2 - 1$, $H(X) = 1$, $a = 1$, $p = 2$, $s = 1$, $M = 2$. Then $F(a) = 0$, $DF(a) = 2$, and $F + 2^2H = X^2 + 3$ has no root modulo eight, hence none in $\mathbb{Z}_2$. The new explanatory identification uses the previously checked no-root polynomial. The strict inequality $M > 2s$ cannot uniformly be replaced by $M \geq 2s$.

B.8 Chapter IX: prescribed-parameter polynomial lifting

B.8.1 Setup and exact lift set

Let $q > 0$, $L \geq 1$, and $m, k, e \geq 0$. Write $S = \mathbb{Z}/q\mathbb{Z}$. Let

$$F \in \mathbb{Z}[Y_1, \ldots, Y_m, T_1, \ldots, T_k]^e, \qquad x = (y_0, t_0) \in \mathbb{Z}^{m+k}, \qquad F(x) \equiv 0 \pmod{q^L}.$$

The actual lower root supplies the unique integral residual

$$r = q^{-L}F(x) \in \mathbb{Z}^e, \qquad \bar{r} = r \bmod q \in S^e.$$

Let the actual reduced Jacobian blocks be

$$J = (D_Y F)(x) \bmod q : S^m \to S^e, \qquad K = (D_T F)(x) \bmod q : S^k \to S^e.$$

For a prescribed parameter digit $v \in S^k$, choose its canonical integral representative $\widehat{v}$. Define $\mathcal{L}_v$ to be the set of actual residue vectors $z \in (\mathbb{Z}/q^{L+1}\mathbb{Z})^{m+k}$ satisfying

$$z \equiv x \pmod{q^L}, \qquad F(z) \equiv 0 \pmod{q^{L+1}}, \qquad z_T \equiv t_0 + q^L \widehat{v} \pmod{q^{L+1}}.$$

The formal predicate uses canonical integer representatives for evaluation and the lower congruence. The previously checked polynomial congruence theorem and digit-to-residue equivalence establish the actual residue-class interpretation. Changing an integral representative of a parameter digit by a multiple of q does not change its prescribed residue modulo q^{L+1}.

B.8.2 The obstruction theorem

Define the finite solved-block cokernel and its parameter-dependent class by

$$\mathcal{C}_x = S^e / \operatorname{im} J, \qquad o_x(v) = [\bar{r} + Kv] \in \mathcal{C}_x.$$

Then

$$\boxed{\mathcal{L}_v \neq \varnothing \iff o_x(v) = 0.}$$

More precisely, the checked digit-to-lift equivalence is

$$\boxed{\mathcal{L}_v \cong \{u \in S^m : Ju = -\bar{r} - Kv\}.}$$

It sends a solved digit u to the actual next residue represented by $x + q^L(\widehat{u}, \widehat{v})$. The obstruction depends on the solved block J, not on the range of the full map $(J\ K)$.

Every nonempty $\mathcal{L}_v$ has a checked free and transitive action of $\ker J$. The action is defined without choosing a base lift. Choosing one valid solved digit then gives an explicit bijection $\mathcal{L}_v \cong \ker J$. In particular, for every positive modulus,

$$o_x(v) = 0 \quad \Longrightarrow \quad \#\mathcal{L}_v = \#\ker J.$$

These results permit composite q, the degenerate modulus $q = 1$, and zero dimensions. Neither J nor the full reduced Jacobian is assumed surjective.

B.8.3 Exact prime-field counts

When $q = p$ is prime, put $\rho = \operatorname{rank}_{\mathbb{F}_p} J$. Then

$$\boxed{\#\mathcal{L}_v = \begin{cases} p^{m-\rho}, & o_x(v) = 0, \\ 0, & o_x(v) \neq 0, \end{cases}} \qquad \boxed{\#\mathcal{C}_x = p^{e-\rho}.}$$

The count is of distinct actual residue classes at level $L+1$. The solved dimension m, rather than the total number $m+k$ of variables, appears in the exponent because the parameter digit is fixed. These rank formulas are asserted only over prime fields.

The main wrapper assumes only the original lower polynomial congruence. It constructs one residual r and supplies the criterion and count uniformly for every v, together with the exact obstruction cardinality.

B.8.4 Proof and dependency chain

The actual derivative calculation gives

$$(DF(x) \bmod q)(u,v) = Ju + Kv.$$

Taylor expansion at depth $L \geq 1$ makes the quadratic remainder divisible by q^{L+1}. Therefore the digit (u,v) is a polynomial lifting correction exactly when

$$\bar{r} + Ju + Kv = 0.$$

The quotient class $[\bar{r} + Kv]$ vanishes exactly when this equation admits u. This uses additive inverses in the image of J, so the sign agrees with the actual Taylor equation.

The earlier unrestricted correction-digit equivalence is then restricted to the predicate that the parameter digit equals v. Taking free coordinates commutes with forming digit residues, and digit residues are injective in the digit. Consequently the restricted equivalence enforces exactly the actual parameter residue condition on $\mathcal{L}_v$.

Translation by $\ker J$ acts freely and transitively on the solved-digit fiber. Transporting this explicit action through the residue equivalence yields the torsor on $\mathcal{L}_v$. Rank-nullity over $\mathbb{F}_p$ then gives the two displayed cardinalities. No numerical enumeration is used to establish the general theorem.

B.8.5 Why the solved block matters

Take $F(Y,T) = T$, $q = 2$, $L = 1$, and $x = (0,0)$. The full Jacobian $(0\ 1)$ is surjective, but $J = 0$. Prescribing $v = 1$ forces $T \equiv 2 \pmod 4$, which cannot satisfy $F = 0 \pmod 4$. Prescribing $v = 0$ gives exactly two lifts, corresponding to $Y \equiv 0, 2 \pmod 4$.

The Lean controls check this actual polynomial example. Thus full-Jacobian surjectivity alone does not make every prescribed coordinate choice admissible. The automatic coordinate selection in the earlier chart theorem is compatible with this obstruction.

B.9 Chapter IX: compatible towers with prescribed parameters

The construction gives actual compatible polynomial towers from initial solved-block surjectivity, for every compatible integer parameter path. For a prime base, it constructs such a path for every p-adic parameter in the initial ball and proves an exact polynomial endpoint with all retained precisions. The formal companion identifies the six supporting modules and their recorded compiler and axiom-audit results.

B.9.1 Statement and notation

Let $q > 0$, $L \geq 1$, and

$$F \in \mathbb{Z}[Y_1, \ldots, Y_m, T_1, \ldots, T_k]^e, \qquad x = (y_0, t_0) \in \mathbb{Z}^{m+k}, \qquad F(x) \in q^L \mathbb{Z}^e.$$

Write $S_q = \mathbb{Z}/q\mathbb{Z}$, and let the actual reduced Jacobian blocks be

$$J_x = D_Y F(x) \bmod q : S_q^m \longrightarrow S_q^e, \qquad K_x = D_T F(x) \bmod q : S_q^k \longrightarrow S_q^e.$$

Assume J_x is surjective. A vector divisibility condition below is coordinatewise.

Integer prescribed-path theorem. For every supplied path $t : \mathbb{N} \to \mathbb{Z}^k$ satisfying

$$t(0) = t_0, \qquad t(n+1) - t(n) \in q^{L+n} \mathbb{Z}^k,$$

there is a single tower $z : \mathbb{N} \to \mathbb{Z}^{m+k}$ with

$$z(0) = x, \qquad \text{free}(z(n)) = t(n),$$

$$F(z(n)) \in q^{L+n} \mathbb{Z}^e, \qquad z(n+1) - z(n) \in q^{L+n} \mathbb{Z}^{m+k}.$$

The parameter equality is exact in integers. For $a \leq b$,

$$z(b) - z(a) \in q^{L+a} \mathbb{Z}^{m+k}, \qquad DF(z(n)) \bmod q = DF(x) \bmod q.$$

Consequently $J_{z(n)} = J_x$ at every level. No later surjectivity or per-step solvability is assumed. This theorem includes positive composite bases and $q = 1$.

Arbitrary p-adic target theorem. Now let $q = p$ be prime. For every

$$\tau \in t_0 + p^L \mathbb{Z}_p^k,$$

there are a compatible integer tower z as above and $b \in \mathbb{Z}_p^{m+k}$ such that

$$z(n) \longrightarrow b, \qquad F(b) = 0, \qquad \text{free}(b) = \tau, \qquad b - z(n) \in p^{L+n} \mathbb{Z}_p^{m+k} \quad (n \geq 0).$$

A compatible integer parameter path converging to τ is constructed in the proof, with its initial representative fixed exactly. A separate theorem accepts every supplied compatible integer path with limit τ and retains its exact coordinates in the constructed tower.

Exact branching. At any stage n, put $N = L + n$. For every parameter digit $v \in \mathbb{F}_p^k$, let

$$\mathcal{L}_{n,v} = \left\{ \bar{w} \in (\mathbb{Z}/p^{N+1}\mathbb{Z})^{m+k} : \begin{array}{l} \bar{w} \equiv z(n) \pmod{p^N}, \\ F(\bar{w}) = 0, \\ \text{free}(\bar{w}) = t(n) + p^N v \pmod{p^{N+1}} \end{array} \right\}.$$

Here $p^N v$ means any integer representative of v, multiplied by p^N. The residue class does not depend on the representative. Then

$$\#\mathcal{L}_{n,v} = p^{m-e} \qquad \text{for every } n \text{ and every } v.$$

Surjectivity implies $e \le m$. This is a count of distinct actual next-level residue classes. It does not count integer representatives or entire infinite towers.

Square solved-block uniqueness. If $m = e$, every prescribed τ has exactly one root in the original ball:

$$\exists!\, b \in x + p^L \mathbb{Z}_p^{e+k} : \quad F(b) = 0, \qquad \mathrm{free}(b) = \tau.$$

Thus all valid integer parameter approximations and all lifting choices with the same target have the same endpoint. This does not assert uniqueness of the integer representatives in the tower. All prime statements include $p = 2$.

B.9.2 Proof

At depth $N \ge 1$, suppose $F(a) = q^N r$ and a desired next integer parameter target satisfies $t' - \mathrm{free}(a) = q^N w$, with $w \in \mathbb{Z}^k$. Surjectivity supplies a solved digit $u \in S_q^m$ satisfying

$$J_a u = -\bar{r} - K_a \bar{w}.$$

Choose an integer representative $\widetilde{u}$, and set

$$a' = a + q^N(\widetilde{u}, w).$$

The parameter quotient is the supplied integer w. Therefore $\mathrm{free}(a') = t'$ exactly. The checked polynomial Taylor identity gives

$$F(a') \equiv F(a) + q^N DF(a)(\widetilde{u}, w) \equiv 0 \pmod{q^{N+1}}.$$

Its quadratic remainder is divisible by q^{2N}, and $2N \ge N + 1$. Also $a' \equiv a \pmod{q}$, so evaluation of every polynomial partial derivative gives the same reduced Jacobian. Induction on the level, carrying the root, exact parameters, and fixed solved Jacobian as its state, constructs the entire integer tower. The existence proof uses ordinary classical choice. It does not supply an optimized computational lifting algorithm.

For a p-adic target τ, use $t(0) = t_0$ and, for $n \ge 1$, the canonical integer representatives of $\tau \bmod p^{L+n}$. The initial congruence hypothesis and the truncation precision imply

$$\tau - t(n) \in p^{L+n} \mathbb{Z}_p^k.$$

Subtracting successive precision relations gives compatibility in $\mathbb{Z}_p$. The checked faithful integer-divisibility theorem gives compatibility in $\mathbb{Z}$. These precision relations also give convergence to τ.

The compatible vector-tower completion theorem constructs b, proves convergence, and retains every stage precision. Continuous polynomial evaluation and the increasing root precision give $F(b) = 0$. Each parameter coordinate has both the corresponding coordinate of b and τ as its limit, so uniqueness of limits gives exact parameter equality.

For every proposed next parameter digit, the previously checked obstruction is $[\bar{r} + K_a v] \in S_p^e / \operatorname{im} J_a$. The fixed solved Jacobian is surjective, so this class vanishes for every digit at every stage. The actual lift fiber is a torsor for $\ker J_a$. Rank-nullity gives its cardinality p^{m-e}.

When $m = e$, the new solved-Jacobian definition is definitionally the existing square block used by the checked continuous parameter chart. Every root in the initial ball with the given parameters is that

chart point. Applying this to any two roots proves equality, including endpoints obtained by different constructions.

B.9.3 Scope controls

The inherited checked example $F(Y,T)=T$, $p=2$, $x=0$, $L=1$ has a surjective full Jacobian but zero solved block. Prescribing parameter digit 1 forces $T=2 \pmod 4$, so no next root exists. Full-Jacobian surjectivity cannot replace the hypothesis on the prescribed solved block.

The new checked example $F(Y_1,Y_2,T)=Y_1$ has a surjective solved block $(1,0)$. Nevertheless $(0,0,0)$ and $(0,2,0)$ are distinct exact roots in the same modulo-two ball with identical parameters. Uniqueness therefore requires the square solved case. The corresponding negative control deliberately attempts the stronger false conclusion and must be rejected by Lean.

Depth $L \geq 1$ is essential for the Taylor step used here. The general singular case can have a nonzero next obstruction after an earlier zero obstruction. Neither the finite one-step criterion nor this smooth theorem establishes unrestricted singular persistence.

B.10 Chapter IX: simultaneous coefficient and parameter deformation

The construction combines coefficient perturbation with prescribed parameter motion. The original solved Jacobian controls existence, local uniqueness, displacement and the first-order error. The formal companion identifies the four supporting modules and their recorded verification scope.

B.10.1 The theorem

Let p be any prime, $R=\mathbb{Z}_p$, and $Q=\mathbb{Q}_p$. Let

$$F,H \in R[Y_1,\dots,Y_e,T_1,\dots,T_k]^e, \qquad a=(y_0,t_0)\in R^{e+k}, \qquad F(a)=0.$$

Here $e,k\geq 0$ are arbitrary finite dimensions. There are as many solved coordinates as equations. Define the actual derivative blocks at the original root:

$$J=D_Y F(a)\in M_e(R), \qquad K=D_T F(a)\in M_{e\times k}(R).$$

Suppose, for natural numbers s,M, that

$$J\in \mathrm{GL}_e(Q), \qquad p^s J^{-1}\in M_e(R), \qquad M>2s.$$

For every $v\in R^k$, set

$$\tau=t_0+p^M v, \qquad G=F+p^M H.$$

Then there is exactly one $y\in y_0+p^{s+1}R^e$ with

$$G(y,\tau)=0.$$

It satisfies

$$y-y_0\in p^{M-s}R^e$$

and there is an integral vector $\varepsilon\in p^{2M-3s}R^e$ such that, in Q^e,

$$\boxed{y-y_0=-p^M J^{-1}\big(H(a)+Kv\big)+\varepsilon.}$$

The same original exponent s appears throughout. No inverse or rank assumption on the perturbed Jacobian is imposed. The statement includes arbitrary integral p-adic coefficients and centers, the dyadic prime, and zero dimensions.

The full-point theorem constructs $b = (y, \tau)$ and also proves

$$b - a \in p^{M-s} R^{e+k} \cap p^{s+1} R^{e+k}.$$

Parameter equality $b_T = \tau$ is exact. Uniqueness concerns every root in the stated solved-coordinate ball with these parameters, not only roots obtained by the construction. No global uniqueness is claimed.

B.10.2 Actual augmented system

Define polynomial systems with $e + k$ equations and variables by

$$\mathcal{F}(Y, T) = \big(F(Y, T), T - t_0\big), \qquad \mathcal{H}(Y, T) = \big(H(Y, T), -v\big).$$

Both are actual polynomials over R. Direct evaluation proves

$$\mathcal{F}(a) = 0, \qquad (\mathcal{F} + p^M \mathcal{H})(b) = 0 \iff F(b) + p^M H(b) = 0 \text{ and } b_T = t_0 + p^M v.$$

Actual polynomial partial differentiation gives

$$D\mathcal{F}(a) = B = \begin{pmatrix} J & K \\ 0 & I_k \end{pmatrix}.$$

These identities are proved from evaluation and partial derivatives, rather than supplied as additional hypotheses.

Choose the integral matrix $A = p^s J^{-1}$. The existing inverse-certificate equivalence gives

$$AJ = JA = p^s I_e.$$

Put $c = p^s$ and define an integral block matrix

$$C = \begin{pmatrix} A & -AK \\ 0 & cI_k \end{pmatrix}.$$

Then

$$CB = \begin{pmatrix} AJ & AK - AK \\ 0 & cI_k \end{pmatrix} = cI_{e+k},$$

$$BC = \begin{pmatrix} JA & -JAK + cK \\ 0 & cI_k \end{pmatrix} = cI_{e+k}.$$

The second equality uses $JA = cI_e$. The generic block proof works over any commutative ring. The p-adic specialization also explicitly proves that B is invertible over Q and that $p^s B^{-1}$ is integral. There is no denominator loss from appending the parameter equations.

B.10.3 Construction, error and uniqueness

Apply the previously checked weighted coefficient-perturbation construction to $\mathcal{F}, \mathcal{H}, a, C$. Its hypotheses follow from the actual augmented identities and the two-sided certificate. It constructs b, proves the

full displacement bound p^{M-s}, and proves uniqueness throughout $a + p^{s+1}R^{e+k}$. The augmented root equivalence gives the required original equations and exact parameter values.

The checked integral first-order estimate gives

$$b - a + p^{M-s}C\mathcal{H}(a) \in p^{2M-3s}R^{e+k}.$$

The decisive block action is

$$C\begin{pmatrix} H(a) \\ -v \end{pmatrix} = \begin{pmatrix} A\big(H(a) + Kv\big) \\ -p^s v \end{pmatrix}.$$

Projecting the solved coordinates therefore gives

$$y - y_0 + p^{M-s}A\big(H(a) + Kv\big) \in p^{2M-3s}R^e.$$

The existing rational-conversion theorem turns $p^{M-s}A$ into $p^M J^{-1}$ and retains an integral error vector. This proves the displayed first-order formula with the original derivative blocks.

For any competing solved-coordinate root $z \in y_0 + p^{s+1}R^e$, the joined point (z, τ) lies in the augmented uniqueness ball: $M > 2s$ implies $M \geq s + 1$, so $\tau - t_0 = p^M v \in p^{s+1}R^k$. Full-point uniqueness gives $(z, \tau) = b$, hence $z = y$. This is the additional comparison needed to obtain uniqueness in exactly the ball stated in the solved-coordinate theorem.

B.10.4 Checked controls and precision boundary

For $p = 2$, $s = 0$, $F(Y, T) = Y + T$, $H = 1$, $a = 0$, and $v = 1$, both original blocks satisfy $J = K = 1$. At $M = 1$, the exact parameter is $\tau = 2$, and the unique solved value is $y = -4$. The combined correction is $-2(1 + 1) = -4$, with zero error.

The sign controls use $M = 2$: the actual augmented root is $(y, \tau) = (-8, 4)$, and the required error ideal is $16\mathbb{Z}_2$. Omitting Kv predicts -4, with error $-4 \notin 16\mathbb{Z}_2$. Reversing its sign predicts 0, with error $-8 \notin 16\mathbb{Z}_2$. Both failures are checked. At binary depth $M = 1$, the reversed-sign difference would lie in the allowed error ideal, so that shallower example alone would not distinguish the sign.

For parameter-only motion, use

$$F(Y, T) = Y + Y^2 + T, \quad H = 0, \quad a = 0, \quad v = 1, \quad p = 2,\ s = 0,\ M = 1.$$

A checked transfer from the prior scalar quadratic root constructs an even y with

$$y + y^2 + 2 = 0, \qquad 4 \mid y + 2, \qquad 8 \nmid y + 2.$$

Thus the error exponent $2M - 3s = 2$ is attained even when only parameters move. No uniform extra error digit can be added to the general theorem. This example establishes that limitation. It does not assert optimality at every choice of p, s, M.

The inherited controls continue to record the necessity of the strict weighted threshold and other earlier scope boundaries. General rationally rank-deficient systems are outside the inverse hypothesis.

B.11 Chapter IX: analytic regularity of the local parameter solution

The argument establishes both analyticity and differentiability of the actual local polynomial solution map at a root with an invertible solved Jacobian. The statements apply to every prime, including $p = 2$. They also hold over any complete nontrivially normed field.

Let $\mathbb{K}$ be such a field, let $e, k \in \mathbb{N}$, and let

$$F \in \mathbb{K}[Y_1, \ldots, Y_e, T_1, \ldots, T_k]^e, \qquad a = (y_0, t_0) \in \mathbb{K}^e \times \mathbb{K}^k.$$

Evaluation of F means coordinatewise evaluation of its actual polynomials. Define the actual derivative blocks

$$J = D_Y F(a) \in M_e(\mathbb{K}), \qquad K = D_T F(a) \in M_{e\times k}(\mathbb{K}),$$

where each matrix entry is the corresponding formal partial derivative evaluated at a. The hypotheses are exactly

$$F(a) = 0, \qquad J \in \mathrm{GL}_e(\mathbb{K}).$$

Neither analyticity of a proposed solution nor a derivative formula is assumed.

Local analytic graph theorem. There exist a map

$$\psi : \mathbb{K}^k \longrightarrow \mathbb{K}^e$$

and open sets $U \subseteq \mathbb{K}^k$, $V \subseteq \mathbb{K}^e$ such that

$$t_0 \in U, \qquad y_0 \in V, \qquad \psi(t_0) = y_0, \qquad \psi(U) \subseteq V.$$

The map is analytic at every point of U, and its graph describes all roots in the product neighborhood:

$$\forall t \in U\ \forall y \in V, \qquad F(y, t) = 0 \iff y = \psi(t).$$

At the base parameter it has the strict Fréchet derivative

$$\boxed{D\psi(t_0) = -J^{-1}K.}$$

Consequently it is differentiable on U. The map is defined on the ambient parameter space, but its root, analyticity and uniqueness conclusions are restricted to the specified neighborhood. Uniqueness is among solved coordinates in V.

The main Lean theorem is `Komorebi.PolynomialAnalyticImplicit.exists_analytic_implicit`. The matrix in its `HasStrictFDerivAt` conclusion is the continuous linear map represented by the displayed matrix. `differentiableOn_of_analytic_graph` records the differentiability consequence.

Proof chain. The scalar polynomial derivative bridge is proved by induction on actual multivariate polynomials: constants, sums, and multiplication by a variable. For a scalar polynomial P, its strict derivative is

$$v \longmapsto \sum_j \frac{\partial P}{\partial x_j}(a) v_j.$$

Assembling the coordinates gives the strict derivative of the evaluated system as its actual Jacobian matrix. Polynomial analyticity is supplied by the pinned Mathlib theorem for multivariate polynomial evaluation, followed by finite coordinate assembly.

Consider the actual augmented polynomial map

$$\Phi(y,t) = (F(y,t),t).$$

Its derivative and inverse derivative at a are

$$D\Phi(a) = \begin{pmatrix} J & K \\ 0 & I_k \end{pmatrix}, \qquad D\Phi(a)^{-1} = \begin{pmatrix} J^{-1} & -J^{-1}K \\ 0 & I_k \end{pmatrix}.$$

The proof constructs the continuous linear equivalence from these explicit matrices and verifies both inverse identities. The strict inverse theorem gives a local inverse g of Φ, and the analytic inverse result shows that this same inverse is analytic near $\Phi(a) = (0,t_0)$. Set

$$\psi(t) = \pi_Y g(0,t).$$

The two inverse identities and a sufficiently small product neighborhood give the exact local graph statement. Composing the derivative of g with parameter insertion and solved-coordinate projection yields $-J^{-1}K$. The chosen parameter neighborhood is also restricted so that the inverse identities, analyticity and inclusion $\psi(U) \subseteq V$ hold there simultaneously.

Transfer to a supplied continuous solution. Suppose $g : \mathbb{K}^k \to \mathbb{K}^e$ is continuous at t_0, $g(t_0) = y_0$, and

$$F(g(t),t) = 0$$

for all t in some neighborhood of t_0. Under the same invertibility hypothesis at a, continuity places $g(t)$ in the uniqueness neighborhood V for all sufficiently close parameters. Hence g and ψ agree on a neighborhood of t_0. The theorem `continuous_selection_regularity` transfers both analyticity and the strict derivative formula to g.

There is also an explicit open-set version. Let U be open, let g be continuous on U, and suppose for every $t \in U$ that

$$F(g(t),t) = 0, \qquad J_t := D_Y F(g(t),t) \in \mathrm{GL}_e(\mathbb{K}).$$

Writing $K_t = D_T F(g(t),t)$, the theorem `continuous_selection_analyticOnNhd` proves

$$g \text{ is analytic at every } t \in U, \qquad Dg(t) = -J_t^{-1}K_t \quad (t \in U),$$

with a strict Fréchet derivative at each point. The invertibility hypothesis at each point is part of this theorem. This is a transfer statement for continuous local root selections satisfying its hypotheses. It does not assert equality of arbitrary global branch choices.

Integral p-adic specialization. For any prime p, let

$$F \in \mathbb{Z}_p[Y,T]^e, \qquad a \in \mathbb{Z}_p^{e+k}, \qquad F(a) = 0.$$

Let $\iota : \mathbb{Z}_p \hookrightarrow \mathbb{Q}_p$ be the coefficient inclusion, applied entrywise to points and matrices. Assume only that

$$\iota(D_Y F(a)) \in \mathrm{GL}_e(\mathbb{Q}_p).$$

The theorem `PolynomialRegularityIntegral.exists_analytic_integral_implicit` constructs the preceding analytic graph in the ambient spaces $\mathbb{Q}_p^k$ and $\mathbb{Q}_p^e$. Its root equation evaluates the original integral polynomial coefficients using ι, and its base derivative is

$$-\iota(D_Y F(a))^{-1}\,\iota(D_T F(a)).$$

The source proves that polynomial evaluation, formal partial differentiation and both Jacobian blocks commute with this inclusion. No invertibility modulo p or integral inverse is required. This permits a solved Jacobian that is singular modulo p while remaining invertible over $\mathbb{Q}_p$. The new specialization is an ambient-field theorem. It does not treat $\mathbb{Z}_p$ as a vector space over $\mathbb{Q}_p$ or assert a new explicit integral radius.

Source module	Mathematical role
`PolynomialAnalyticEvaluation.lean`	Actual polynomial analyticity and strict derivative equals the evaluated formal Jacobian.
`PolynomialAnalyticInverse.lean`	Analyticity of the same local inverse supplied by an invertible strict derivative.
`PolynomialAnalyticBlocks.lean`	Explicit augmented derivative equivalence and its solved-coordinate restriction.
`PolynomialAnalyticImplicit.lean`	Open local root graph, analyticity, base strict derivative and differentiability.
`PolynomialAnalyticSelection.lean`	Transfer to continuous root selections, including the per-point open-set theorem.
`PolynomialRegularityIntegral.lean`	Integral coefficients and centers included in the ambient p-adic field, with actual Jacobian compatibility.

This block closes the local analytic regularity obligation stated in `TARGET_CONTRACT.json`. It supplies no new uniform numerical radius. The previous weighted lifting and joint deformation results retain their own explicit radii and error estimates. It does not establish a global analytic branch or a result at a point whose solved Jacobian is singular over the fraction field. The proof uses established analytic inverse machinery and makes no novelty claim for the general implicit-function theorem.

The remaining questions include singular persistence with higher obstructions, native presentation invariance (13.2), global or completed compatibility and descent (13.3), and sharper arithmetic statements. The local regularity theorem does not settle those questions.

C Restricted native progress and complete prose proofs

This appendix presents restricted polynomial obstruction, finite reconstruction, rectangularity and topological arguments as ordinary proofs. The intrinsic commutative-algebra obstruction uses the cited Stacks specialization and its finiteness argument. These results do not complete unrestricted Native 13.2 or Native 13.3 and have no new Lean certification.

C.1 2. A complete polynomial obstruction theorem

Scope. Fix a prime p, an integer $L \geq 1$, finite natural numbers d, r, and polynomials $F_1, \ldots, F_r \in \mathbb{Z}[X_1, \ldots, X_d]$. Let $\bar{x} \in (\mathbb{Z}/p^L\mathbb{Z})^d$ satisfy every equation $F_i(\bar{x}) = 0$. Choose any integer representative $x \in \mathbb{Z}^d$. Set

$$e_i = F_i(x)/p^L \pmod{p}, \qquad J_{ij} = \frac{\partial F_i}{\partial X_j}(x) \pmod{p}.$$

The divisions are exact integer divisions. Regard J as a linear map $\mathbb{F}_p^d \to \mathbb{F}_p^r$. Define

$$O_{F,\bar{x}} = \mathbb{F}_p^r / \operatorname{im} J, \qquad o_{F,\bar{x}} = [e].$$

Theorem. For this fixed polynomial presentation, the group and distinguished class are independent of the chosen integer representative. The class vanishes if and only if $\bar{x}$ lifts to a solution modulo p^{L+1}. The obstruction group has $p^{r-\operatorname{rank} J}$ elements. When lifts exist, their set is a torsor for $\ker J$ and contains exactly $p^{d-\operatorname{rank} J}$ elements.

Proof. For every integer correction vector t, expansion of each monomial gives

$$F(x + p^L t) \equiv F(x) + p^L J(x) t \pmod{p^{L+1}}. \tag{1}$$

Indeed, each term involving at least two correction factors is divisible by p^{2L}, and $2L \geq L + 1$. Constant and linear correction terms give the two displayed summands. Adding the monomial expansions proves (1) for every polynomial, with no denominator or analytic convergence issue.

Every element reducing to $\bar{x}$ is uniquely expressible modulo p^{L+1} as $x + p^L t$ with $t \in \mathbb{F}_p^d$. By (1), it solves the equations exactly when

$$e + Jt = 0 \quad \text{in } \mathbb{F}_p^r. \tag{2}$$

Equation (2) is soluble exactly when $e \in \operatorname{im} J$: the sign makes no difference because the image is a subspace. This is equivalent to $[e] = 0$.

Another representative has the form $x' = x + p^L s$. Since $L \geq 1$, its Jacobian is the same modulo p. Dividing (1) by p^L shows that its residual digit is $e' = e + Js$, so it gives exactly the same quotient class. If t_0 solves (2), all its solutions are $t_0 + \ker J$, and no others. The finite cardinalities follow from rank-nullity. This also covers no variables, no equations, zero Jacobian and singular solutions. □

The proof is a universal mathematical argument. The accompanying Python checks are independently bounded evidence about an implementation, not a substitute for that argument. The standalone Lean specialization does not formalize multivariable polynomial expansion (1), unless its own checked source explicitly says so.

C.1.1 Necessary boundaries and exact examples

- **A failed singular lift:** $p = 3$, $L = 1$, $F(X) = X^2 - 3$, $x = 0$. Here $e = -1 \neq 0$ in $\mathbb{F}_3$ and $J = 0$. There are no lifts of zero modulo 3 to a root modulo 9.
- **Representative independence is a quotient assertion:** for $F(X) = X^2 - 1$, $p = 3$, $L = 1$, representatives 1 and 4 have residual digits 0 and 2. Their common Jacobian is 2, with full image, and both give zero obstruction. Their corrected representatives 1 and 10 are the same root modulo 9.
- **A successful step need not give indefinite continuation:** for $F(X) = X^2 - 5$, $p = 2$, the root 1 modulo 2 has zero obstruction and lifts to $1, 3$ modulo 4. Both have nonzero obstruction at the next step. There is no root modulo 8 and no compatible infinite lift.
- **The raw Jacobian cokernel need not be presentation independent:** the presentations $F = (X)$ and $G = (X, 0)$ define the same solution system. Their raw obstruction groups at zero are respectively zero and $\mathbb{F}_p$. Both actual distinguished obstructions are zero. Thus one cannot claim a canonical isomorphism of these two raw groups merely because the equations define the same seed.

C.2 3. An intrinsic finite obstruction under an algebraic restriction

Let B be a finitely presented commutative $\mathbb{Z}$-algebra, p prime and $L \geq 1$. For a ring map $f : B \to \mathbb{Z}/p^L\mathbb{Z}$, use its reduction to make $\mathbb{F}_p$ a B-module. There is a canonical class

$$o(f) \in \mathrm{Ext}^1_B(\mathrm{NL}_{B/\mathbb{Z}}, \mathbb{F}_p)$$

whose vanishing is equivalent to a lift $B \to \mathbb{Z}/p^{L+1}\mathbb{Z}$. This follows from the ring-map lifting theorem in Stacks Project, Lemma 91.2.1, tag 08S5. Specialize the source extension to $0 \to 0 \to B \to B$ and the target extension to the kernel $p^L/p^{L+1} \cong \mathbb{F}_p$. The base maps are identities of $\mathbb{Z}$. The kernel is square zero because $L \geq 1$.

The comparison in Stacks Project, Lemma 10.134.2, tag 00S1 identifies the complex from a polynomial presentation with the canonical naive cotangent complex up to homotopy. Thus this construction is intrinsic, up to the stated canonical comparison.

The group is finite. Choose a finite presentation $P = \mathbb{Z}[X_1, \ldots, X_d] \twoheadrightarrow B$ with finitely generated kernel I. The presentation description in Stacks Project, Remark 91.2.4, tag 0GPU identifies the group as

$$\mathrm{coker}\left(\mathrm{Hom}_B(\Omega_{P/\mathbb{Z}} \otimes_P B, \mathbb{F}_p) \longrightarrow \mathrm{Hom}_B(I/I^2, \mathbb{F}_p)\right).$$

If I has r chosen generators, evaluation on their classes injects the latter Hom module into $\mathbb{F}_p^r$. Hence its quotient is finite. The relation module I/I^2 retains dependencies between equations, including redundant zero equations, which the raw target $\mathbb{F}_p^r$ does not.

Status: externally proved after the exact stated specialization, with the finiteness argument above. This theorem is not independently formalized in the accompanying Lean file. Extending it to all operator/recurrence/module systems admitted by manuscript Definition 1.1 requires a specified encoding and a proof that its notion of lift is preserved. That is the missing implication for applying it to native Conjecture 13.2 without restriction.

C.3 4. A topological obstruction to the proposed spine formula

Consider the recurrence $a_{n+1} = a_n$ with integer initial value. Its global parameter space is $X = \mathbb{Z}$. Its local parameter spaces are $X_p = \mathbb{Z}_p$, and every local realization has the same prime-independent recurrence. Give the local factors their usual p-adic topologies and the product its product topology. For these integral local factors, the restricted product relative to all $\mathbb{Z}_p$ is the ordinary product. Write

$$A = \mathbb{Z}_2, \qquad B = \prod_{p>2} \mathbb{Z}_p, \qquad D = \{(n, (n)_{p>2}) : n \in \mathbb{Z}\} \subset A \times B.$$

Theorem. There are no Hausdorff space C and continuous maps $f : A \to C$, $g : B \to C$ for which the equality locus

$$\{(a, b) : f(a) = g(b)\}$$

is exactly D. Thus the canonical local realizations of global integer constant sequences cannot satisfy the manuscript's fiber-product formula in this Hausdorff topological interpretation.

Proof. The equality locus is closed: it is the inverse image of the diagonal of $C \times C$ under the continuous map $f \times g$, and a Hausdorff space has closed diagonal.

The set D is dense in $A \times B$. A nonempty basic open set prescribes congruences at only finitely many primes and finite precision at each. The moduli are pairwise coprime. The finite Chinese remainder theorem supplies an integer meeting all those congruences, so every such open set meets D.

However, D is not the whole product. The element with 2-adic coordinate zero and every odd-adic coordinate one is not diagonal. If an integer had zero 2-adic coordinate, it would be divisible by every power of two, forcing it to be zero. Its 3-adic coordinate would then be zero, not one. Consequently D is a proper dense subset and is not closed. It cannot be the equality locus above. □

This is not asserted to refute the original, underspecified sentence under every possible reading. It refutes the explicit and natural interpretation just stated. The assumptions of continuity, Hausdorffness and identification through the canonical local maps are essential to this claim.

C.3.1 A precise reconstruction that does hold

For a finite set S of primes, positive precisions L_p, and a fixed integer polynomial system F, put

$$N = \prod_{p \in S} p^{L_p}, \qquad \mathrm{Sol}_F(R) = \{x \in R^d : F(x) = 0\}.$$

Then coefficientwise CRT gives a canonical bijection

$$\mathrm{Sol}_F(\mathbb{Z}/N\mathbb{Z}) \cong \prod_{p \in S} \mathrm{Sol}_F(\mathbb{Z}/p^{L_p}\mathbb{Z}).$$

The reason is exact: the CRT ring isomorphism preserves polynomial evaluation, and zero in a finite product is equivalent to zero in every component. For F empty, passing to all compatible finite congruences identifies the profinite completion $\widehat{\mathbb{Z}}$ with $\prod_p \mathbb{Z}_p$. The same finite reconstruction

is compatible with coefficient projections and integer recurrence matrices because these operations commute with reduction and CRT.

Here the polynomial equations or recurrence matrices are fixed common data. The theorem reconstructs finite or profinite objects. It does not identify a "smallest" spine, impose additional spectral descent, or recover global $\mathbb{Z}$ from arbitrary tuples of local points. A full form of 13.3 must state exactly which object is completed and specify its universal property.

C.3.2 A finite compatibility criterion with a precise universal property

Fix finite sets A, B of local packets and an explicitly given relation $R \subseteq A \times B$. Say that R is rectangular when

$$R(a,b),\ R(a',b),\ R(a',b') \quad \Longrightarrow \quad R(a,b')$$

for all $a, a' \in A$ and $b, b' \in B$.

Theorem. The relation R can be represented exactly as

$$R(a,b) \quad \Longleftrightarrow \quad f(a) = g(b)$$

for maps from A and B to a common set if and only if R is rectangular. In that case a canonical finite choice of common set is the connected-component set C_R of the bipartite graph with vertex set $A \sqcup B$ and edges given by R.

The component maps $f_R : A \to C_R$ and $g_R : B \to C_R$ have the following universal property. For every set Y and maps $h : A \to Y$, $k : B \to Y$ satisfying $h(a) = k(b)$ whenever $R(a,b)$, there is a unique map $u : C_R \to Y$ with $u \circ f_R = h$ and $u \circ g_R = k$.

Proof. Any equality representation is rectangular, since the three assumed equalities imply $f(a) = g(b) = f(a') = g(b')$.

Conversely, suppose R is rectangular. An edge always gives equal component labels. If $a \in A$ and $b \in B$ have equal component labels, there is a finite path between them. Its length is odd. A path of length one is already an R-edge. A longer path begins with three edges $a - b_1 - a_1 - b_2$. Rectangularity gives the edge $a - b_2$. Replacing these three edges by that edge shortens the path by two. Induction therefore yields $R(a,b)$. This proves the exact equality representation. Finiteness follows because the component set is a quotient of the finite set $A \sqcup B$.

For the universal property, combine h and k into a function on $A \sqcup B$. It is constant across every edge by hypothesis, hence along every finite path. It therefore descends to a well-defined function on components. The two required factorization identities hold by construction. They determine the value on every component, proving uniqueness. □

For example, the three pairs $(a_0, b_0), (a_1, b_0), (a_1, b_1)$ without (a_0, b_1) fail rectangularity. Every equality-label representation agreeing on those three pairs also admits the fourth pair, so it cannot represent that relation exactly.

This is a precise finite gluing obligation: future arithmetic applications must define A, B and R and prove rectangularity. The universal property holds for the component quotient even when rectangularity fails, but then the equality locus strictly enlarges R. The theorem establishes neither a minimum number of labels nor a smallest spine in the manuscript's unspecified category. It carries no automatic topology, spectral information or Frobenius compatibility. **Evidence: prose proof in this note, not a Lean formalization.**

D Local polyphase constructions

The following derivations retain their stated parameter ranges. The Amice analytic pullback, depleted branches, finite-field modal minimality and unrestricted infinite polyphase claims have only the evidence identified in their source records. Their inclusion does not give them a general Lean certificate.

The figures are explanatory schematics and exact finite examples. The modulo-seven inverse plot uses the tail 6, 5, 2, 6, 5, 2, ... The divisor-branch finite-difference theorem is Theorem 3.7, with its normalization loss. Higher kernel valuations and unrestricted local-to-global questions remain open in their stated scopes.

D.1 Introduction

Euler numbers carry several kinds of local information at once. Their indices have residue addresses. Their coefficient transforms have triangular depth. Their local expansions have Mahler degree. Their congruences depend on the valuation of the transfer kernel and on the valuation of the source value in the row being reconstructed. These pieces should not be merged into one undifferentiated congruence calculation.

This article begins with the Hurwitz product

$$(a \star b)_n = \sum_{r=0}^{n} \binom{n}{r} a_r b_{n-r}.$$

Over a field of characteristic zero it is multiplication of exponential generating functions. Over a finite field it realizes the additive group algebra through the exponential modes $\epsilon_u = (u^n)_{n\geq 0}$. This finite algebra gives an exact proof of the standard window period. Hurwitz series rings are treated in [15, 7]. The finite difference and umbral background may be found in [21, 10, 13].

Fix an odd prime p, a divisor $m \mid p-1$, and put

$$h = \frac{p-1}{m}.$$

The reduced standard window inverse is defined by

$$\sum_{q=0}^{n} \binom{mn}{mq} e_q^{(m,p)} = \delta_{n0} \quad \text{in } \mathbb{F}_p.$$

We prove that

$$e_{n+h}^{(m,p)} = e_n^{(m,p)} \qquad (n \geq 1).$$

The result first follows from subgroup symmetry in the finite Hurwitz algebra. A second proof shows that the standard window operator preserves the finite space of sequences that are h periodic after their constant term. The two arguments expose different parts of the same finite character structure.

The odd and dyadic Euler formulas share a common reconstruction framework. For each prime p we define an even source series H_p and an inverse kernel K_p satisfying

$$K_p(t)H_p(t) = \operatorname{sech} t.$$

At an odd prime,

$$H_p(t) = \cosh^2(pt)\operatorname{sech} t, \qquad K_p(t) = \operatorname{sech}^2(pt).$$

At the dyadic prime,

$$H_2(t) = \frac{\sinh(2t)\sinh t}{2t^2}, \qquad K_2(t) = \frac{4t^2}{\sinh^2(2t)}.$$

The coefficient convolution is identical in form, but the kernels have different integral weights. This is the correct sense in which the local theories unify.

The dyadic kernel is substantially sharper than the separate coefficient bound alone suggests. If

$$\frac{z^2}{\sinh^2 z} = \sum_{r\geq 0} \beta_r z^{2r}$$

and

$$\kappa_{2,r} = \beta_r 2^{2r}(2r)!,$$

then

$$v_2(\kappa_{2,r}) = 4r - 1 \qquad (r \geq 1).$$

Consequently, truncation after depth M is uniformly valid modulo

$$2^{4M+3},$$

and this exponent is optimal. The rowwise estimate records the additional binary carries in the binomial coefficient. Classical congruence theory for Euler families is extensive [8, 26, 27, 24, 9]. The formulas here are stated as exact coordinate identities. Historical priority requires a separate comparison of the individual theorems.

The second expansion concerns local interpolation. The exponent of the finite unit group moves every unit mode into a principal unit neighborhood. Binomial expansion then gives a Mahler series whose depth r coefficient gains at least Lr digits when the chosen jump is the identity modulo p^L. The same argument applies to polynomial exponential modes and to invertible matrix state systems. Mahler interpolation and the Amice transform are classical [19, 6, 20]. The local unit structure and valuation tools used here are standard [16, 22, 25, 12].

The odd Euler source contains one nonunit base. Removing that base gives an analytic unit source and an interpolation of a depleted Euler sequence on every even character branch. The dyadic source is

coordinate layer	what is retained	coordinate
$n = Nk + j$	residue address	j
$u = \omega(u)\langle u\rangle$	finite character	$\omega(u)$
$f(x) = \sum c_r \binom{x}{r}$	Mahler depth	r
$KH = \operatorname{sech}$	weighted reconstruction	$v_p(\kappa_r)$

Figure 1. New reconstruction. The arithmetic coordinate chain. Residue address, local character, Mahler depth, transfer weight, and reconstruction are recorded separately.

already analytic on $\mathbb{Z}_p$ after a removable singularity is filled. It yields a direct analytic interpolation of E_{2n} on $\mathbb{Z}_2$. These branches are then refined by the substitution

$$x = Mq + s.$$

The induced Mahler coefficient transform is finite and explicit. Its kernels compose exactly. At a prime power radix, they become sparse modulo the residue prime. Every monotone valuation ladder survives the refinement.

The remaining sections treat arithmetic lift obstructions, recurrence packets, and Chinese remainder reconstruction. The lifting obstruction is the residual defect modulo the image of the declared correction map. A recurrence packet is governed by a jump matrix C^N. If one integral recurrence is fixed, all of its local realizations have one common source. These statements do not prove that arbitrary unrelated local objects have a global origin.

Four boundaries remain explicit. A period modulo p does not by itself lift to all powers of p. The full finite difference estimate for every congruential reciprocal packet still requires control of the triangular inverse. Exact higher valuations of the affine polyphase kernel are not yet known in general. Chinese remainder reconstruction and finite adelic synchronization do not prove a global source for arbitrary local systems.

D.2 Hurwitz series and finite field subgroup inversion

For the divisor-branch family, Theorem 3.7 proves the bound with the j! normalization and factorial valuation loss. No unrestricted family is asserted.

D.2.1 The Hurwitz ring

definition D.1 (Hurwitz series ring). Let R be a commutative ring with identity. The Hurwitz series ring $\mathcal{H}(R)$ is the set $R^{\mathbb{N}}$ with componentwise addition and product

$$(a \star b)_n = \sum_{r=0}^{n} \binom{n}{r} a_r b_{n-r}.$$

Its identity is

$$\delta = (1, 0, 0, \ldots).$$

When the factorials are invertible, the sequence a may be written as the exponential generating series

$$A(t) = \sum_{n \geq 0} a_n \frac{t^n}{n!},$$

and the Hurwitz product becomes ordinary multiplication. The sequence definition remains valid in every characteristic.

definition D.2 (Exponential mode). For $u \in R$, define

$$\epsilon_u = (u^n)_{n \geq 0},$$

where $0^0 = 1$.

proposition D.3 (Addition law for modes). For every $u, v \in R$,

$$\boxed{\epsilon_u \star \epsilon_v = \epsilon_{u+v}.}$$

Proof. The nth coefficient is

$$\sum_{r=0}^{n} \binom{n}{r} u^r v^{n-r} = (u + v)^n$$

by the binomial theorem. □

D.2.2 The finite additive group algebra

Let p be prime. Write $\mathbb{F}_p^+$ for the additive group of the finite field.

theorem D.4 (Finite field mode algebra). The map

$$\Theta : \mathbb{F}_p[\mathbb{F}_p^+] \longrightarrow \mathcal{H}(\mathbb{F}_p), \qquad [u] \longmapsto \epsilon_u,$$

is an injective algebra homomorphism. Its image is the p dimensional subalgebra spanned by the modes ϵ_u.

Proof. The product law follows from proposition D.3. To prove injectivity, suppose

$$\sum_{u \in \mathbb{F}_p} c_u \epsilon_u = 0.$$

Looking at the coordinates $n = 0, 1, \ldots, p-1$ gives a Vandermonde system with matrix

$$(u^n)_{0 \leq n \leq p-1,\ u \in \mathbb{F}_p}.$$

Its determinant is the product of all differences between distinct field elements, so it is nonzero. Hence every c_u vanishes. □

proposition D.5 (Augmentation criterion). Let

$$x = \sum_{u \in \mathbb{F}_p} c_u [u] \in \mathbb{F}_p[\mathbb{F}_p^+].$$

Then x is a unit if and only if

$$\operatorname{aug}(x) = \sum_u c_u$$

is nonzero. If J is the augmentation ideal, then

$$J^p = 0.$$

Proof. Choose a generator g of the cyclic additive group. Then

$$\mathbb{F}_p[\mathbb{F}_p^+] \cong \mathbb{F}_p[y]/(y^p - 1) = \mathbb{F}_p[y]/(y-1)^p.$$

The augmentation ideal is generated by $y - 1$, so its pth power is zero. Every element has the form $a + n$ with $a \in \mathbb{F}_p$ and $n \in J$. It is invertible exactly when $a \neq 0$, which is the augmentation condition. □

corollary D.6 (Finite inverse formula). If x has augmentation one and $n = x - 1$, then

$$\boxed{x^{-1} = \sum_{r=0}^{p-1} (-n)^r.}$$

Proof. By proposition D.5, $n^p = 0$. The finite geometric sum gives the inverse. □

D.2.3 Subgroup selectors

Fix a divisor m of $p - 1$. Let

$$H_m = \{\zeta \in \mathbb{F}_p^\times : \zeta^m = 1\}.$$

This group has order m.

definition D.7 (Multiple selector). Define $a^{(m)} \in \mathcal{H}(\mathbb{F}_p)$ by

$$a_n^{(m)} = \begin{cases} 1, & m \mid n, \\ 0, & m \nmid n. \end{cases}$$

lemma D.8 (Subgroup average). The multiple selector has the mode expansion

$$\boxed{a^{(m)} = \frac{1}{m} \sum_{\zeta \in H_m} \epsilon_\zeta.}$$

Proof. The nth coefficient on the right is

$$\frac{1}{m} \sum_{\zeta \in H_m} \zeta^n.$$

The character sum is one when $m \mid n$ and zero otherwise. □

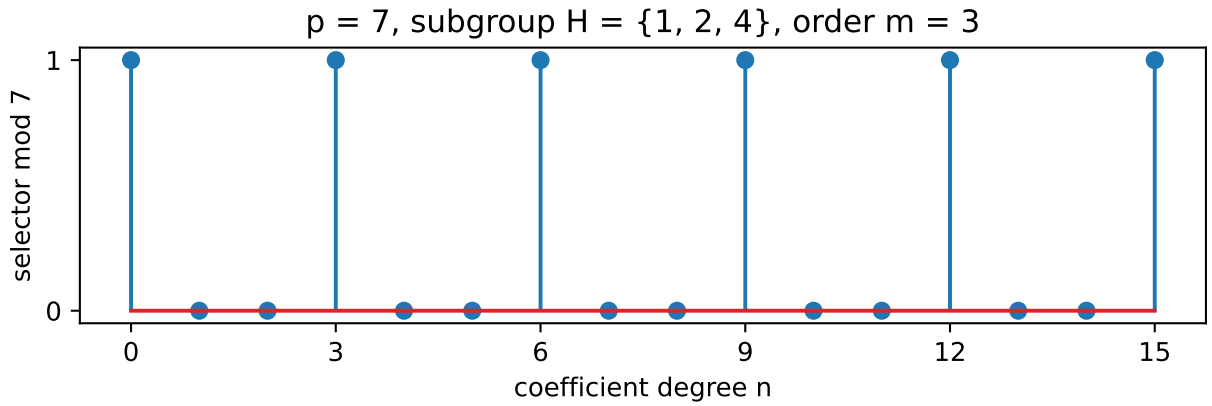


Figure 2. New reconstruction. A subgroup average in the additive group algebra. The highlighted modes select coefficient degrees divisible by the subgroup order.

Since the subgroup average has augmentation one, it has a unique Hurwitz inverse. Write

$$b^{(m,p)} = (a^{(m)})^{-1}.$$

theorem D.9 (Support and positive degree period). The inverse $b^{(m,p)}$ has the following properties.

1. $b_n^{(m,p)} = 0$ unless $m \mid n$.
2. For every positive n,

$$\boxed{b_{n+p-1}^{(m,p)} = b_n^{(m,p)}.}$$

Proof. Multiplication by an element $\eta \in H_m$ gives an automorphism of the additive group and therefore an automorphism

$$\epsilon_u \longmapsto \epsilon_{\eta u}$$

of the mode algebra. The selector $a^{(m)}$ is fixed by every such automorphism, so its unique inverse is fixed as well.

Write

$$b^{(m,p)} = \sum_{u \in \mathbb{F}_p} c_u \epsilon_u.$$

Invariance gives $c_{\eta u} = c_u$. After the change of variable $u \mapsto \eta u$,

$$b_n^{(m,p)} = \eta^n b_n^{(m,p)}$$

for every $\eta \in H_m$. If $m \nmid n$, some η satisfies $\eta^n \neq 1$, so the coefficient vanishes.

For positive n, the zero mode contributes nothing. Every nonzero mode satisfies $u^{p-1} = 1$. Hence

$$b_{n+p-1}^{(m,p)} = \sum_{u \neq 0} c_u u^{n+p-1} = \sum_{u \neq 0} c_u u^n = b_n^{(m,p)}.$$

□

definition D.10 (Compressed inverse). Set

$$e_n^{(m,p)} = b_{mn}^{(m,p)}.$$

proposition D.11 (Compressed inverse recurrence). The compressed coefficients satisfy

$$\boxed{\sum_{q=0}^{n} \binom{mn}{mq} e_q^{(m,p)} = \delta_{n0}.}$$

In particular, $e_0^{(m,p)} = 1$, and the positive coefficients are determined recursively because the diagonal is one.

Proof. Evaluate the Hurwitz identity

$$a^{(m)} \star b^{(m,p)} = \delta$$

at degree mn. Both factors vanish outside degrees divisible by m. Reindexing the surviving terms gives the displayed recurrence. □

corollary D.12 (Uniform compressed period). Let

$$h = \frac{p-1}{m}.$$

Then

$$\boxed{e_{n+h}^{(m,p)} = e_n^{(m,p)} \qquad (n \geq 1).}$$

Proof. Since $m(n+h) = mn + p - 1$, the conclusion follows from theorem D.9. □

corollary D.13 (Rational ordinary generating function). In $\mathbb{F}_p[[z]]$,

$$\boxed{\sum_{n \geq 0} e_n^{(m,p)} z^n = 1 + \frac{\sum_{r=1}^{h} e_r^{(m,p)} z^r}{1 - z^h}.}$$

Proof. Separate the constant coefficient and group the positive indices into residue classes modulo h. □

theorem D.14 (Exponential form and minimal period). Let

$$G_m = (\mathbb{F}_p^{\times})^m.$$

This cyclic group has order h. There are uniquely determined coefficients $d_v \in \mathbb{F}_p$, indexed by $v \in G_m$, such that

$$\boxed{e_n^{(m,p)} = \sum_{v \in G_m} d_v v^n \qquad (n \geq 1).}$$

The minimal positive period is the least common multiple of the orders of those v for which $d_v \neq 0$. In particular, it divides h.

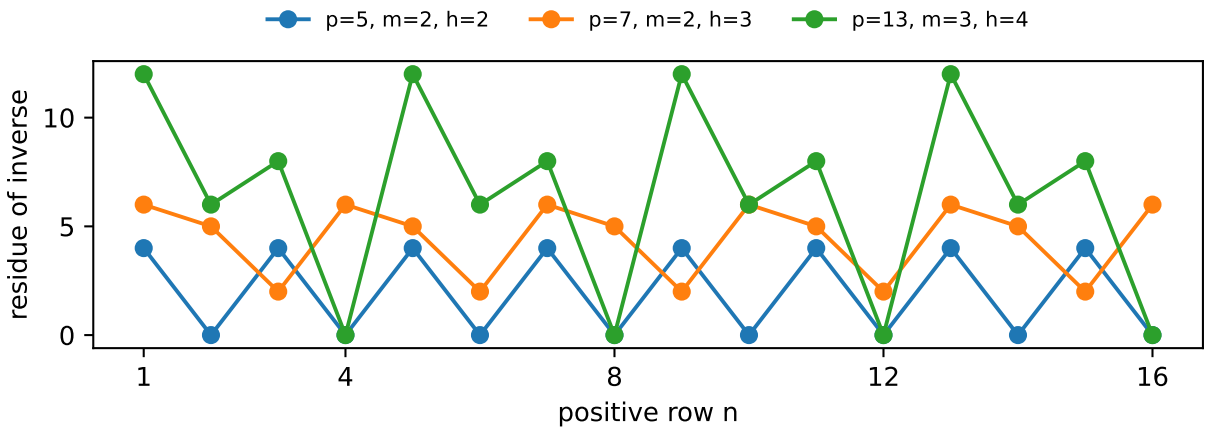


Figure 3. New reconstruction. Several compressed inverses. Every row is periodic after the constant term, as proved in corollary D.12.

Proof. The coefficients c_u in the proof of theorem D.9 are constant on the cosets of H_m. For $u \neq 0$, the value u^m depends only on the coset, and every element of G_m arises. Grouping the modes by cosets gives the displayed expansion.

Distinct elements of G_m define linearly independent functions on one full period, by the Vandermonde determinant. Therefore a positive integer r is a period exactly when $v^r = 1$ for every active mode. The least such r is the stated least common multiple. □

example D.15 (A compressed inverse modulo seven). **Editorial numerical correction.** The positive-index standard-window tail printed below is erroneous. The defining recurrence gives $e_1 = -1 \equiv 6$, $e_2 = 5$ and $e_3 = -61 \equiv 2 \pmod 7$, so the correct positive tail is $6, 5, 2, 6, 5, 2, \ldots$. The period-three conclusion remains correct. Use the recurrence-derived tail for this example.

Take $p = 7$ and $m = 2$, so $h = 3$. The recurrence gives

$$e_n^{(2,7)} = 1, 6, 4, 1, 6, 4, \ldots \qquad (n \geq 1).$$

The positive tail has exact period three.

D.3 Unit triangular coefficient transport

The examples later in the chapter use different kernels, but their coefficient algebra has one common form. This section records that form once. Let R be a commutative algebra over $\mathbb{Q}$, and take all coefficients in R.

Let

$$H(t) = \sum_{r \geq 0} h_r \frac{t^r}{r!}, \qquad h_0 = 1.$$

There is a unique inverse

$$K(t) = H(t)^{-1} = \sum_{r \geq 0} k_r \frac{t^r}{r!}.$$

theorem D.16 (Exact coefficient transport). Let

$$A(t) = \sum_{n\geq 0} A_n \frac{t^n}{n!}, \qquad S(t) = H(t)A(t).$$

Then

$$\boxed{S_n = \sum_{r=0}^{n} \binom{n}{r} h_r A_{n-r}.}$$

The inverse coordinate formula is

$$\boxed{A_n = \sum_{r=0}^{n} \binom{n}{r} k_r S_{n-r}.}$$

Both transforms are lower triangular with diagonal one.

Proof. The first identity is the coefficient rule for the product of exponential generating functions. Multiply by $K(t)$ and use $K(t)H(t) = 1$ for the inverse identity. The diagonal coefficient in each transform is the constant coefficient of its kernel, which is one. □

corollary D.17 (Composition of coefficient transports). If U_H and U_G are the triangular transforms associated with unit kernels H and G, then

$$\boxed{U_H U_G = U_{HG}.}$$

In particular, the inverse of U_H is $U_{H^{-1}}$.

Proof. Multiplication by G, followed by multiplication by H, is multiplication by HG. □

theorem D.18 (Filtered triangular reconstruction). Let p be prime. Suppose

$$A_n = \sum_{r=0}^{n} \binom{n}{r} c_r p^{w(r)} S_{n-r},$$

where $c_r \in \mathbb{Z}_p$, $S_n \in \mathbb{Z}_p$, and

$$w(r) \in \mathbb{N}.$$

For every precision $L \geq 1$,

$$\boxed{A_n \equiv \sum_{\substack{0\leq r\leq n \\ w(r)<L}} \binom{n}{r} c_r p^{w(r)} S_{n-r} \pmod{p^L}.}$$

When the omitted set is nonempty, the tail has valuation at least

$$\boxed{\min_{\substack{0\leq r\leq n \\ w(r)\geq L}} \left(w(r) + v_p\binom{n}{r} + v_p(S_{n-r}) \right).}$$

Proof. Every omitted summand belongs to $p^L\mathbb{Z}_p$. The sharper statement follows by taking the minimum of the exact lower bounds for the summands and applying the nonarchimedean triangle inequality. □

remark D.19. An arbitrary invertible triangular matrix can encode an arbitrary sequence. The transforms retained here have an explicit generating kernel and a visible filtration. Those two features make exact inversion and uniform truncation possible.

D.4 Congruential reciprocal packets

D.4.1 Residue sections of the exponential series

For integers $N \geq 1$ and $0 \leq j < N$, define

$$\mathcal{E}_{N,j}(t) = \sum_{n \geq 0} \frac{t^{Nn+j}}{(Nn+j)!}.$$

The forced factor t^j may be removed without changing the residue address.

definition D.20 (Congruential reciprocal packet). Define

$$\boxed{\mathcal{C}_{N,j}(t) = \frac{t^j/j!}{\mathcal{E}_{N,j}(t)} = \sum_{n \geq 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}.}$$

theorem D.21 (Triangular inverse system). The coefficients satisfy $C_{N,j}(0) = 1$ and

$$\boxed{\sum_{q=0}^{n} \binom{Nn+j}{Nq} C_{N,j}(q) = \delta_{n0}.}$$

For $n \geq 1$,

$$\boxed{C_{N,j}(n) = -\frac{\sum_{q=0}^{n-1} \binom{Nn+j}{Nq} C_{N,j}(q)}{\binom{Nn+j}{Nn}}.}$$

Proof. Multiply $\mathcal{C}_{N,j}(t)$ by $\mathcal{E}_{N,j}(t)$ and compare the coefficient of $t^{Nn+j}/(Nn+j)!$. The last summand is the diagonal term, which is nonzero over $\mathbb{Q}$. □

proposition D.22 (A denominator bound). Let

$$D_{N,j}(n) = \prod_{r=1}^{n} \binom{Nr+j}{Nr}, \qquad D_{N,j}(0) = 1.$$

Then

$$\boxed{D_{N,j}(n) C_{N,j}(n) \in \mathbb{Z}.}$$

For every prime p,

$$v_p(C_{N,j}(n)) \geq -\sum_{r=1}^{n} v_p \binom{Nr+j}{Nr}.$$

Proof. Use induction on n in the recurrence of theorem D.21. Every earlier denominator divides $D_{N,j}(n-1)$, while the new diagonal denominator is the final factor of $D_{N,j}(n)$. The valuation statement follows. □

example D.23 (Euler and Bernoulli boundaries). For $(N, j) = (2, 0)$,

$$\mathcal{C}_{2,0}(t) = \operatorname{sech} t,$$

so

$$C_{2,0}(n) = E_{2n}.$$

For $(N, j) = (2, 1)$,

$$\mathcal{C}_{2,1}(t) = \frac{t}{\sinh t},$$

and

$$C_{2,1}(n) = 2\left(1 - 2^{2n-1}\right) B_{2n}.$$

The usual convention $B_1 = -1/2$ is used. Euler and Bernoulli number conventions are reviewed in [10, 23].

D.4.2 Standard windows

Fix a prime p, an integer $m \geq 1$, and an address $0 \leq j < p$. Put

$$N = mp.$$

For arbitrary m, let $\widetilde{e}_n^{(m,p)} \in \mathbb{F}_p$ be the unique sequence satisfying

$$\boxed{\sum_{q=0}^{n} \binom{mn}{mq} \widetilde{e}_q^{(m,p)} = \delta_{n0}.}$$

When $m \mid p - 1$, this sequence agrees with the compressed Hurwitz inverse in proposition D.11.

standard fact D.24 (Lucas congruence). If $a = a_0 + a_1 p + \cdots$ and $b = b_0 + b_1 p + \cdots$ are base p expansions, then

$$\binom{a}{b} \equiv \prod_r \binom{a_r}{b_r} \pmod{p}.$$

This is the classical theorem of Lucas [18].

theorem D.25 (Standard window reduction). For $0 \leq q \leq n$,

$$\boxed{\binom{mpn + j}{mpq} \equiv \binom{mn}{mq} \pmod{p}.}$$

The diagonal coefficient is a unit modulo p:

$$\binom{mpn + j}{mpn} \equiv 1 \pmod{p}.$$

Proof. Write

$$mpn + j = p(mn) + j, \qquad mpq = p(mq).$$

Lucas reduction separates the lowest base p digit and gives

$$\binom{mpn+j}{mpq} \equiv \binom{j}{0}\binom{mn}{mq} \pmod{p}.$$

The diagonal case is obtained by setting $q = n$. □

corollary D.26 (Standard window integrality). Every coefficient $C_{mp,j}(n)$ belongs to $\mathbb{Z}_p$.

Proof. The recurrence in theorem D.21 starts with a p integral constant term. The binomial coefficients are integers, and every diagonal term is a p adic unit by theorem D.25. Induction gives the result. □

corollary D.27 (Address independence modulo the prime). Let $\widetilde{e}_n^{(m,p)}$ be the reduced standard inverse above. Then

$$\boxed{C_{mp,j}(n) \equiv \widetilde{e}_n^{(m,p)} \pmod{p}}$$

for every $0 \le j < p$.

Proof. Modulo p, the recurrence in theorem D.21 becomes

$$\sum_{q=0}^{n} \binom{mn}{mq} C_{mp,j}(q) = \delta_{n0}.$$

This is the defining recurrence for $\widetilde{e}_n^{(m,p)}$. The diagonal is one, so the solution is unique. □

theorem D.28 (Uniform standard window period). Assume that p is odd and $m \mid p-1$. Put

$$h = \frac{p-1}{m}.$$

Then for every standard window address $0 \le j < p$,

$$\boxed{C_{mp,j}(n+h) \equiv C_{mp,j}(n) \pmod{p} \qquad (n \ge 1).}$$

Proof. By corollary D.27, every address reduces to the same reduced standard inverse. Under the divisibility assumption, that inverse agrees with the compressed Hurwitz inverse by proposition D.11. Its positive tail has period h by corollary D.12. □

corollary D.29 (Rational residual generating function). Under the hypotheses of theorem D.28, for every standard address j,

$$\sum_{n\ge 0} \left(C_{mp,j}(n) \bmod p\right) z^n = 1 + \frac{\sum_{r=1}^{h} \widetilde{e}_r^{(m,p)} z^r}{1 - z^h}.$$

Proof. Combine corollaries D.13 and D.27. □

theorem D.30 (Fourier invariance of the standard window operator). Let p be odd, let $m \mid p-1$, and let $h = (p-1)/m$. Define

$$(T_{m,p} f)_n = \sum_{q=0}^{n} \binom{mn}{mq} f_q \quad \text{in } \mathbb{F}_p.$$

Let V_h be the vector space of sequences satisfying

$$f_{n+h} = f_n \qquad (n \geq 1).$$

Then

$$\boxed{T_{m,p}(V_h) = V_h.}$$

The restriction of $T_{m,p}$ to V_h is an automorphism.

Proof. Choose $d_0 = f_h$ and $d_n = f_n$ for $n \geq 1$. Then d is fully h periodic and $f = d + c\delta_0$ for $c = f_0 - f_h$. Since $T_{m,p}\delta_0$ is the constant sequence one, it is enough to treat d. Expand

$$d_q = \sum_{\chi \in \mu_h} c_\chi \chi^q.$$

The mth power map on $\mathbb{F}_p^\times$ has image μ_h. Choose y with $y^m = \chi$. The root filter gives

$$T_{m,p}(\chi^\bullet)_n = \frac{1}{m} \sum_{\zeta \in \mu_m} (1 + \zeta y)^{mn}.$$

If $1 + \zeta y = 0$, the corresponding term vanishes for positive n. Otherwise

$$((1 + \zeta y)^m)^h = (1 + \zeta y)^{p-1} = 1.$$

Every summand is therefore h periodic for positive n. Thus $T_{m,p}$ preserves V_h. It is unit lower triangular on every finite initial segment. Its restriction to the finite dimensional space determined by $f_0, f_1, \ldots, f_h$ is injective and hence bijective. □

corollary D.31 (A second proof of the standard window period). The reduced inverse $e^{(m,p)}$ and every standard address satisfy the period in theorem D.28.

Proof. The sequence $\delta = (1, 0, 0, \ldots)$ belongs to V_h. Its inverse image under the automorphism in theorem D.30 also belongs to V_h. Address independence modulo p gives the statement for every standard address. □

remark D.32. The theorem gives a period modulo p. A period modulo higher powers requires additional control of the triangular inverse and cannot be inferred from the finite field argument alone.

D.5 One local Euler source and kernel calculus

The odd and dyadic reconstructions have one convolutional form. Their sources, kernels, and valuation profiles are different. We first construct each chart and then state the common theorem.

D.5.1 The odd source and its inverse

Let q be a positive odd integer. Define

$$S_n(q) = \sum_{a=0}^{q-1} (-1)^a (2a + 1)^n.$$

p = 7, m = 2, h = 3	root filter	mode at row n
$\chi = 2,\ y = 3$	input character	2^{n}
$(1+y)^{2} = 2$	first output mode	$4\,2^{n}$
$(1-y)^{2} = 4$	second output mode	$4\,4^{n}$
$T(2^{\bullet})_{n}$	sum modulo 7	$4(2^{n}+4^{n})$

Figure 4. New reconstruction. Finite field Fourier transfer for a standard window. Each character becomes a finite sum of positive index modes with period dividing h.

Let the signed Euler numbers and Euler zigzag numbers be defined by

$$\operatorname{sech} t = \sum_{n \geq 0} E_{n} \frac{t^{n}}{n!}$$

and

$$\sec t + \tan t = \sum_{n \geq 0} Z_{n} \frac{t^{n}}{n!}.$$

lemma D.33 (Odd source generating function). For every positive odd q,

$$\boxed{\sum_{n \geq 0} S_{n}(q) \frac{t^{n}}{n!} = \mathrm{e}^{qt} \cosh(qt) \operatorname{sech} t.}$$

Its even part is

$$\boxed{\sum_{n \geq 0} S_{2n}(q) \frac{t^{2n}}{(2n)!} = \cosh^{2}(qt) \operatorname{sech} t.}$$

Proof. The finite geometric sum gives

$$\sum_{a=0}^{q-1} (-1)^{a} \mathrm{e}^{(2a+1)t} = \mathrm{e}^{t} \frac{1+\mathrm{e}^{2qt}}{1+\mathrm{e}^{2t}} = \mathrm{e}^{qt} \cosh(qt) \operatorname{sech} t.$$

Averaging at t and $-t$ gives the even part. □

lemma D.34 (Zigzag inverse kernel).

$$\boxed{\operatorname{sech}^{2} t = \sum_{r \geq 0} (-1)^{r} Z_{2r+1} \frac{t^{2r}}{(2r)!}.}$$

Proof. The identity $\tanh t = -i \tan(it)$ gives the odd zigzag expansion of $\tanh t$. Differentiate. □

n / q	0	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
2	1	6	1	0	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0
4	1	0	0	0	1	0	0	0	0
5	1	3	0	0	3	1	0	0	0
6	1	3	5	0	5	3	1	0	0
7	1	0	0	0	0	0	0	1	0
8	1	1	0	0	4	0	0	1	1

Figure 5. New reconstruction. The standard window matrix for $p = 7$, $m = 2$, and $0 \le j < 7$. Reduction removes the address and retains diagonal one.

theorem D.35 (Odd modulus Euler coordinate pair). For every positive odd q and every even $n \ge 0$,

$$\boxed{E_n = \sum_{r=0}^{n/2} (-1)^r Z_{2r+1} \binom{n}{2r} q^{2r} S_{n-2r}(q).}$$

The inverse transform is

$$\boxed{S_n(q) = E_n + \sum_{r=1}^{n/2} 2^{2r-1} \binom{n}{2r} q^{2r} E_{n-2r}.}$$

Both transforms are lower triangular with diagonal one.

Proof. The even source identity gives

$$\operatorname{sech} t = \operatorname{sech}^2(qt) \sum_{n \ge 0} S_{2n}(q) \frac{t^{2n}}{(2n)!}.$$

Insert lemma D.34 and compare coefficients. For the inverse, multiply $\operatorname{sech} t$ by

$$\cosh^2(qt) = 1 + \sum_{r \ge 1} 2^{2r-1} q^{2r} \frac{t^{2r}}{(2r)!}.$$

□

corollary D.36 (Odd modulus truncation). For every $M \ge 0$, every positive odd q, and every even n,

$$\boxed{E_n \equiv \sum_{r=0}^{\min(M, n/2)} (-1)^r Z_{2r+1} \binom{n}{2r} q^{2r} S_{n-2r}(q) \pmod{q^{2M+2}}.}$$

If $p \mid q$ and $a = v_p(q)$, the same truncation is valid modulo $p^{2a(M+1)}$.

Proof. Every omitted term contains q^{2r} with $r \geq M+1$. The local statement follows by taking the p adic valuation. □

proposition D.37 (Odd row weight). For an odd prime p, the rth nonzero term in the reconstruction has valuation

$$\boxed{W_p(n,r) = 2r + v_p(Z_{2r+1}) + v_p\binom{n}{2r} + v_p\big(S_{n-2r}(p)\big).}$$

The binomial contribution is

$$\boxed{v_p\binom{n}{2r} = \frac{s_p(2r) + s_p(n-2r) - s_p(n)}{p-1},}$$

where s_p is the base p digit sum.

Proof. The first identity separates the valuation of every factor. The second is Legendre's formula, equivalently Kummer's carry theorem [14]. □

D.5.2 The dyadic source and its inverse

For even $n \geq 0$, define

$$\boxed{R_n = \frac{3^{n+2} - 1}{4(n+1)(n+2)}.}$$

Its even exponential generating function is

$$\boxed{\sum_{n \text{ even}} R_n \frac{t^n}{n!} = \frac{\sinh(2t)\sinh t}{2t^2}.}$$

Define β_r by

$$\boxed{\frac{z^2}{\sinh^2 z} = \sum_{r \geq 0} \beta_r z^{2r}.}$$

proposition D.38 (Dyadic inverse coefficients). For $r \geq 1$,

$$\boxed{\beta_r = -\frac{2^{2r}(2r-1)B_{2r}}{(2r)!}.}$$

Consequently,

$$\boxed{v_2(\beta_r) = s_2(r) - 1.}$$

Every R_n is a dyadic unit.

Proof. Differentiate the Bernoulli expansion of $\coth z$ and use $-(\coth z)' = \sinh(z)^{-2}$. The von Staudt and Clausen theorem gives $v_2(B_{2r}) = -1$. Legendre's formula gives

$$v_2((2r)!) = 2r - s_2(r).$$

For R_n, put $k = n + 2$. Since k is even,

$$v_2(3^k - 1) = 2 + v_2(k).$$

The denominator has the same valuation because $n + 1$ is odd. □

theorem D.39 (Dyadic Euler coordinate pair). For every even $n \geq 0$,

$$\boxed{E_n = \sum_{r=0}^{n/2} \beta_r 2^{2r} (n)_{2r} R_{n-2r}.}$$

The inverse transform is

$$\boxed{R_n = \sum_{r=0}^{n/2} \frac{2^{4r+1}}{(2r+2)!} (n)_{2r} E_{n-2r}.}$$

Proof. Substitute $z = 2t$ in the inverse series and multiply

$$\frac{4t^2}{\sinh^2(2t)} \frac{\sinh(2t)\sinh t}{2t^2} = \operatorname{sech} t.$$

Coefficient extraction gives the forward transform. The inverse follows from

$$\left(\frac{\sinh(2t)}{2t}\right)^2 = \sum_{r \geq 0} \frac{2^{4r+1}}{(2r+2)!} t^{2r}.$$

□

D.5.3 The common source and kernel law

For a prime p, define

$$A_{p,n} = \begin{cases} S_n(p), & p \text{ odd}, \\ R_n, & p = 2 \text{ and } n \text{ even}, \\ 0, & p = 2 \text{ and } n \text{ odd}, \end{cases}$$

and

$$\kappa_{p,r} = \begin{cases} (-1)^r Z_{2r+1} p^{2r}, & p \text{ odd}, \\ \beta_r 2^{2r} (2r)!, & p = 2. \end{cases}$$

theorem D.40 (Unified local Euler transfer). For every prime p and every even $n \geq 0$,

$$\boxed{E_n = \sum_{r=0}^{n/2} \binom{n}{2r} \kappa_{p,r} A_{p,n-2r}.}$$

If

$$H_p(t) = \sum_{m \geq 0} A_{p,2m} \frac{t^{2m}}{(2m)!}$$

and

$$K_p(t) = \sum_{r \geq 0} \kappa_{p,r} \frac{t^{2r}}{(2r)!},$$

then

$$\boxed{K_p(t) H_p(t) = \operatorname{sech} t.}$$

For odd p,

$$H_p(t) = \cosh^2(pt) \operatorname{sech} t, \qquad K_p(t) = \operatorname{sech}^2(pt).$$

For $p = 2$,

$$H_2(t) = \frac{\sinh(2t) \sinh t}{2t^2}, \qquad K_2(t) = \frac{4t^2}{\sinh^2(2t)}.$$

Proof. The two source and kernel products were proved in theorems D.35 and D.39. Their divided power coefficient laws have the displayed common form. □

theorem D.41 (Unified inverse local transfer). Define

$$\lambda_{p,r} = \begin{cases} 1, & p \text{ odd and } r = 0, \\ 2^{2r-1} p^{2r}, & p \text{ odd and } r \geq 1, \\ \dfrac{2^{4r+1}}{(2r+1)(2r+2)}, & p = 2. \end{cases}$$

Then for every prime p and every even $n \geq 0$,

$$\boxed{A_{p,n} = \sum_{r=0}^{n/2} \binom{n}{2r} \lambda_{p,r} E_{n-2r}.}$$

Equivalently, if

$$L_p(t) = K_p(t)^{-1},$$

then

$$\boxed{H_p(t) = L_p(t) \operatorname{sech} t,}$$

where

$$L_p(t) = \cosh^2(pt)$$

for odd p, and

$$L_2(t) = \left(\frac{\sinh(2t)}{2t} \right)^2$$

for $p = 2$.

Proof. For odd p, this is the inverse identity in theorem D.35. For $p = 2$, rewrite the inverse identity in theorem D.39 by using

$$(n)_{2r} = (2r)! \binom{n}{2r}.$$

Table 3. The common local Euler architecture and the parts that remain prime dependent.

Feature	Odd prime chart	Dyadic chart
Finite torsion address	Modulo $p-1$	Sign address modulo two
Principal units	$1+p\mathbb{Z}_p$	$1+4\mathbb{Z}_2$
Source	Alternating odd power sum $S_n(p)$	Weighted source R_n
Kernel	$\operatorname{sech}^2(pt)$	$4t^2/\sinh^2(2t)$
Positive depth weight	At least $2r$, with zigzag and row factors	Exactly $4r-1$, before the binomial carry
Analytic branch	Depleted unit source	Direct integral source
Natural refined step	$(p-1)p^a$	2^{a+1}

The coefficient becomes

$$\frac{2^{4r+1}(2r)!}{(2r+2)!}=\frac{2^{4r+1}}{(2r+1)(2r+2)}.$$

The generating functions are the inverses of the kernels in theorem D.40. □

corollary D.42 (Exact change between local source charts). **Editorial range correction.** This comparison holds for the even source subsequences, equivalently the evenized generating functions H_p. It does not identify their full odd-degree source values. For $p=3$, the original odd source has $A_{3,1}=S_1(3)=3$, whereas every odd dyadic source coefficient is zero. The even kernel L_3K_2 cannot change that odd coefficient. Read the coefficient consequence below only at even indices. The displayed identity $H_p=L_pK_\ell H_\ell$ and its proof remain valid.

For any two primes p and ℓ, the source sequences $A_{p,n}$ and $A_{\ell,n}$ are related by an exact unit lower triangular transformation obtained by composing the inverse kernel L_p with the reconstruction kernel K_ℓ.

Proof. Both source series reconstruct $\operatorname{sech} t$. Hence

$$H_p(t)=L_p(t)K_\ell(t)H_\ell(t).$$

The multiplier has constant coefficient one, so theorem D.16 gives an exact unit lower triangular coordinate change. □

example D.43 (The first odd reconstruction). For $p=3$ and $n=2$,

$$S_2(3)=1-9+25=17.$$

Since $Z_1=1$, $Z_3=2$, and $S_0(3)=1$, the odd coordinate formula gives

$$E_2=17-2\cdot 9=-1.$$

The large source value and the depth one correction are both necessary.

example D.44 (The first nontrivial dyadic row). For $n=4$,

$$R_4=\frac{91}{15},\qquad R_2=\frac{5}{3},\qquad R_0=1.$$

The exact kernel coefficients are

$$\kappa_{2,1} = -\frac{8}{3}, \qquad \kappa_{2,2} = \frac{128}{5}.$$

Therefore

$$E_4 = \frac{91}{15} - \frac{8}{3}\binom{4}{2}\frac{5}{3} + \frac{128}{5} = 5.$$

The rational source values are dyadic units, while the transfer depth is carried by the kernel.

remark D.45. The common convolution does not identify the integral charts. At an odd prime the finite character coordinate has order $p-1$. At the dyadic prime it begins with the sign coordinate, and the principal unit subgroup begins at $1+4\mathbb{Z}_2$.

D.5.4 Exact dyadic transfer depth

For $r \geq 1$, write

$$\kappa_{2,r} = \beta_r 2^{2r}(2r)! = -2^{4r}(2r-1)B_{2r}.$$

theorem D.46 (Exact dyadic kernel weight). **Editorial index conventions.** The standalone kernel valuation holds for all $r \geq 1$. Its reconstruction-term formula requires even $n \geq 0$ and $1 \leq r \leq n/2$. In the following truncation formulas only terms with $r \leq n/2$ occur. Terms that would use R_{n-2r} at a negative index are omitted. An empty tail is exactly zero, and the minimum over an empty set of term valuations is $+\infty$. These conventions include $M \geq n/2$.

For every $r \geq 1$,

$$\boxed{v_2(\kappa_{2,r}) = 4r - 1.}$$

The rth reconstruction term has exact valuation

$$\boxed{v_2\left(\kappa_{2,r}\binom{n}{2r}R_{n-2r}\right) = 4r - 1 + v_2\binom{n}{2r}.}$$

Equivalently,

$$\boxed{W_2(n,r) = 4r - 1 + s_2(r) + s_2(n-2r) - s_2(n).}$$

Proof. The Bernoulli formula gives $v_2(B_{2r}) = -1$. The source value is a unit. Kummer's binary carry formula gives the last expression. □

theorem D.47 (Optimal dyadic truncation). Let

$$T_{2,M}(n) = \sum_{r=0}^{\min(M,n/2)} \kappa_{2,r}\binom{n}{2r}R_{n-2r}.$$

Then

$$\boxed{E_n \equiv T_{2,M}(n) \pmod{2^{4M+3}}.}$$

More precisely,

$$\boxed{v_2\big(E_n - T_{2,M}(n)\big) \geq \min_{M+1\leq r\leq n/2}\left(4r - 1 + v_2\binom{n}{2r}\right).}$$

The uniform exponent $4M+3$ is best possible.

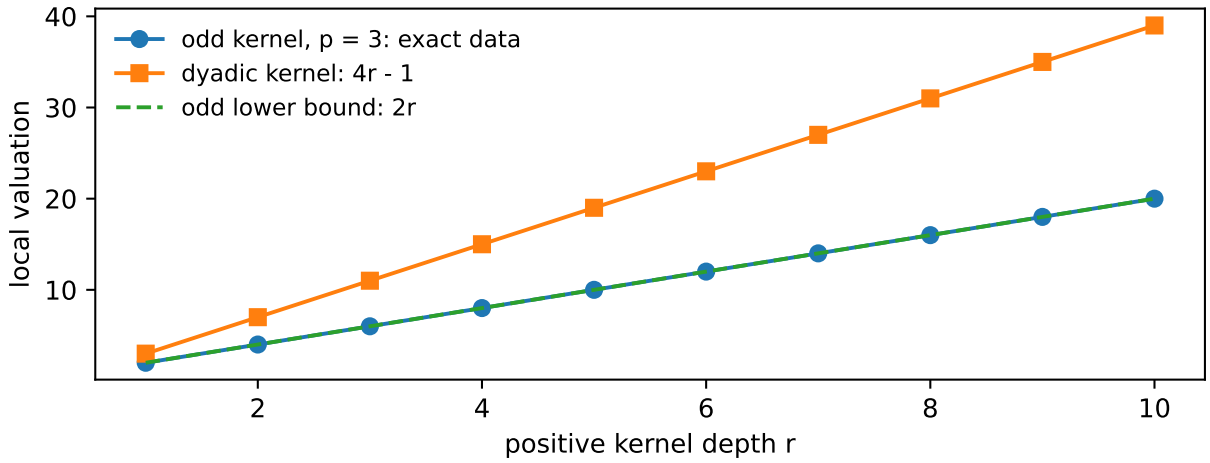


Figure 6. New reconstruction. Odd and dyadic kernels share one convolution law but have different valuation profiles. The positive depth dyadic kernel has exact slope four and offset minus one.

local coordinate	odd p	p = 2
finite torsion	modulo p - 1	sign modulo 2
principal units	$1 + p\mathbb{Z}_p$	$1 + 4\mathbb{Z}_2$
source choice	depleted unit branch	direct dyadic source
kernel floor	$2r$	$4r - 1$

Figure 7. New reconstruction. The local Euler charts share source and kernel reconstruction. Their torsion addresses, principal unit depths, source corrections, and precision growth remain different.

Proof. Every omitted term has valuation at least $4(M+1)-1$. For optimality, take $n = 2M+2$. There is one omitted term. Its binomial coefficient and source value are one, so its valuation is exactly $4M+3$. □

The first three sharp truncations are

$$E_n \equiv R_n \pmod 8,$$

$$E_n \equiv R_n - \frac{8}{3}\binom{n}{2} R_{n-2} \pmod{128},$$

$$E_n \equiv R_n - \frac{8}{3}\binom{n}{2} R_{n-2} + \frac{128}{5}\binom{n}{4} R_{n-4} \pmod{2048}.$$

D.6 Mahler transfer and local analytic branches

The finite character coordinate resolves the torsion part of a local unit. A sufficiently long arithmetic jump moves the remaining unit into a principal unit neighborhood. The coarse variable then has a convergent Mahler expansion with an explicit valuation slope.

D.6.1 Finite Mahler coordinates

For a function f on the nonnegative integers, define

$$\Delta f(n) = f(n+1) - f(n).$$

theorem D.48 (Finite Mahler reconstruction). Given $f(0), \dots, f(N)$, put $c_r = \Delta^r f(0)$. Then

$$\boxed{f(n) = \sum_{r=0}^{n} c_r \binom{n}{r} \qquad (0 \le n \le N).}$$

For every s with $0 \le s \le N$,

$$\boxed{\Delta^s f(n) = \sum_{r=s}^{N} c_r \binom{n}{r-s} \qquad (0 \le n \le N-s).}$$

Proof. Let

$$P(x) = \sum_{r=0}^{N} c_r \binom{x}{r}.$$

Newton interpolation gives $P(n) = f(n)$ for $0 \le n \le N$. Iterating

$$\Delta \binom{x}{r} = \binom{x}{r-1}$$

gives

$$\Delta^s P(n) = \sum_{r=s}^{N} c_r \binom{n}{r-s}.$$

When $0 \le n \le N-s$, every value used in $\Delta^s f(n)$ lies among the prescribed data, so $\Delta^s f(n) = \Delta^s P(n)$. □

corollary D.49 (Mahler valuation ledger). If $c_r \in \mathbb{Z}_p$ and $v_p(c_r) \ge w(r)$, then

$$\boxed{v_p(\Delta^s f(n)) \ge \min_{r \ge s} \left(w(r) + v_p \binom{n}{r-s} \right).}$$

Proof. Apply the nonarchimedean triangle inequality to theorem D.48. □

The infinite interpolation theorem states that every continuous function on $\mathbb{Z}_p$ has a unique convergent Mahler expansion [19, 20]. We use that theorem as a standard input when the local variable is allowed to range over all of $\mathbb{Z}_p$.

D.6.2 Carmichael and Mahler transfer for unit modes

standard fact D.50 (Local unit decomposition). For odd p, every unit has a unique factorization

$$u = \omega(u)\langle u\rangle, \qquad \omega(u)^{p-1} = 1, \qquad \langle u\rangle \in 1 + p\mathbb{Z}_p.$$

Every dyadic unit has a unique factorization

$$u = (-1)^{\epsilon(u)}\langle u\rangle, \qquad \epsilon(u) \in \{0,1\}, \qquad \langle u\rangle \in 1 + 4\mathbb{Z}_2.$$

The first factor gives the finite character address. The second factor admits logarithmic and Mahler coordinates.

For odd p and $L \geq 1$, put

$$\Lambda_{p,L} = (p-1)p^{L-1}.$$

For $p = 2$ and $L \geq 3$, put

$$\Lambda_{2,L} = 2^{L-2}.$$

These are exponents of the unit groups modulo the declared prime powers.

theorem D.51 (One unit mode). Let $u \in \mathbb{Z}_p^\times$, let $H = \Lambda_{p,L}$, and put

$$q_{u,L} = u^H - 1 \in p^L\mathbb{Z}_p.$$

For every integer address j and every $x \in \mathbb{Z}_p$,

$$\boxed{u^{Hx+j} = u^j \sum_{r\geq 0} q_{u,L}^r \binom{x}{r}.}$$

The rth Mahler coefficient has valuation at least Lr.

Proof. Write $u^{Hx+j} = u^j(1+q_{u,L})^x$ and use the convergent binomial series on $\mathbb{Z}_p$. □

example D.52 (Paired principal unit expansions). At the odd prime three,

$$2^{2x+j} = 2^j 4^x = 2^j \sum_{r\geq 0} 3^r \binom{x}{r}.$$

At the dyadic prime,

$$3^{2x+j} = 3^j 9^x = 3^j \sum_{r\geq 0} 8^r \binom{x}{r}.$$

The formal expansions are identical. Their first local slopes are one and three because $4 - 1 = 3$ while $9 - 1 = 8$.

corollary D.53 (Basic odd and dyadic difference gains). For odd p,

$$\boxed{\Delta^r_{(p-1)p^a} u^n = u^n \left(u^{(p-1)p^a} - 1\right)^r, \qquad v_p\left(\Delta^r_{(p-1)p^a} u^n\right) \geq (a+1)r.}$$

For odd $u \in \mathbb{Z}_2^\times$,

$$\boxed{\Delta^r_{2^{a+1}} u^n = u^n \left(u^{2^{a+1}} - 1\right)^r, \qquad v_2\left(\Delta^r_{2^{a+1}} u^n\right) \geq (a+3)r.}$$

Proof. One step difference multiplies a unit mode by $u^H - 1$. Fermat reduction and lifting the exponent give the odd estimate. For odd u,

$$v_2\left(u^{2^{a+1}} - 1\right) = v_2(u^2 - 1) + a \geq a + 3.$$

□

D.6.3 Polynomial exponential modes

theorem D.54 (Polynomial exponential Mahler transfer). Let

$$f(n) = P(n)u^n$$

with $P \in \mathbb{Z}_p[n]$ of degree d. Let $H = \Lambda_{p,L}$, fix an address j, and write

$$Q_j(x) = P(Hx + j), \qquad y = u^H, \qquad q = y - 1.$$

Then

$$\boxed{f(Hx + j) = \sum_{r \geq 0} c_{j,r} \binom{x}{r},}$$

where

$$\boxed{c_{j,r} = u^j \sum_{k=0}^{\min(d,r)} \binom{r}{k} \Delta^k Q_j(0) y^k q^{r-k}.}$$

Consequently,

$$\boxed{v_p(c_{j,r}) \geq L \max(r - d, 0).}$$

For $R \geq d$, truncation at depth R is valid modulo $p^{L(R+1-d)}$ uniformly on $\mathbb{Z}_p$.

Proof. On the polynomial factor the difference operator acts as $q + y\Delta$. The two terms commute. Expand $(q + y\Delta)^r$ and evaluate at zero. Since $\Delta^k Q_j = 0$ for $k > d$, every surviving term contains at least q^{r-d}. □

D.6.4 Matrix state systems and finite adelic synchronization

Let

$$v_{n+1} = Cv_n, \qquad a_n = w^T C^n v_0,$$

with $C \in M_d(\mathbb{Z}_p)$ invertible modulo p.

theorem D.55 (Matrix Carmichael and Mahler transfer). **Editorial hypothesis correction.** The original displayed statement leaves the vectors untyped. Its integrality conclusion requires $v_0, w \in \mathbb{Z}_p^d$, $L \geq 1$, and $H, j, n \in \mathbb{N}$ with $H > 0$. The value at $Hx + j$ for $x \in \mathbb{Z}_p$ means the matrix-binomial interpolation. With $p = 3$, $C = 4$, $L = 1$, $w = 1$ and $v_0 = 1/3$, one has $\Delta_1 a_0 = 1$, which is not in $3\mathbb{Z}_3$. The printed claim without these restrictions is therefore not a certificate as written. The full original proof follows under the corrected hypotheses.

Assume

$$C^H = I + p^L B$$

with $B \in M_d(\mathbb{Z}_p)$. For every address j and every $x \in \mathbb{Z}_p$,

$$\boxed{a_{Hx+j} = \sum_{r \geq 0} p^{Lr} \binom{x}{r} w^T C^j B^r v_0.}$$

The series converges on $\mathbb{Z}_p$, and

$$\boxed{\Delta_H^r a_n \equiv 0 \pmod{p^{Lr}}.}$$

Proof. Use

$$C^{Hx+j} = C^j (I + p^L B)^x$$

and the matrix binomial series. One difference in the coarse variable multiplies the series by $p^L B$. □

theorem D.56 (Finite adelic transfer). **Editorial hypothesis correction.** Read this statement with $v_0, w \in \mathbb{Z}^d$, $H > 0$ and $j, n, r \in \mathbb{N}$, and positive integer L_p. These restrictions make the integral congruences and powers well typed. Allowing $v_0 = 1/3$ in the one-prime example above loses the advertised digit. Invertibility of C alone does not restore it. The original proof is retained under these hypotheses.

Let

$$Q = \prod_{p \in S} p^{L_p}$$

for a finite set of primes. Let $C \in M_d(\mathbb{Z})$ be invertible modulo Q, and choose H such that

$$C^H = I + QB$$

with $B \in M_d(\mathbb{Z})$. Then for every nonnegative integer x,

$$\boxed{a_{Hx+j} = \sum_{r=0}^{x} Q^r \binom{x}{r} w^T C^j B^r v_0.}$$

For every r,

$$\boxed{\Delta_H^r a_n \equiv 0 \pmod{Q^r}.}$$

Proof. Apply the integer matrix binomial theorem and reduce at every selected prime. □

example D.57 (A simultaneous Fibonacci jump). Let

$$C = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

The matrix C is invertible modulo twenty four, and

$$C^{24} - I = 24 \begin{pmatrix} 3126 & 1932 \\ 1932 & 1194 \end{pmatrix}.$$

For every scalar coordinate of the state sequence $C^n v_0$,

$$\boxed{\Delta_{24}^r a_n \equiv 0 \pmod{24^r}.}$$

One integral jump therefore gives simultaneous congruence ladders at the primes two and three.

D.6.5 Amice pullback

definition D.58 (Amice pullback). For a formal exponential generating series

$$F(z) = \sum_{n \geq 0} a_n \frac{z^n}{n!},$$

define

$$\widehat{F}(T) = F(\log(1+T)).$$

theorem D.59 (Derivative conjugacy).

$$\boxed{\widehat{F}'(T) = (1+T)\frac{d}{dT}\widehat{F}(T).}$$

Proof. Differentiate the composition $F(\log(1+T))$. □

proposition D.60 (Weighted convergence). If

$$\widehat{F}(T) = \sum_{r \geq 0} c_r T^r$$

and

$$v_p(c_r) + r\alpha \longrightarrow +\infty,$$

then the series converges whenever $v_p(T) \geq \alpha$.

Proof. The valuation of the rth term is $v_p(c_r) + r v_p(T)$, which tends to positive infinity. □

D.6.6 Odd prime depletion

For an odd prime p, the source $S_n(p)$ contains one nonunit base. Put

$$\sigma_p = (-1)^{(p-1)/2}$$

and

$$S_n^{\times}(p) = S_n(p) - \sigma_p p^n.$$

theorem D.61 (Depleted odd prime reconstruction). For every even n,

$$\boxed{\left(1 - \sigma_p p^n\right) E_n = \sum_{r=0}^{n/2} (-1)^r Z_{2r+1} \binom{n}{2r} p^{2r} S_{n-2r}^{\times}(p).}$$

Proof. The removed source mode contributes $\sigma_p \cosh(pt)$. Multiplication by $\operatorname{sech}^2(pt)$ gives $\sigma_p \operatorname{sech}(pt)$. Subtract it from the full source and kernel product. □

Fix an even address $j \in \{0, 2, \dots, p-3\}$. For $x \in \mathbb{Z}_p$, define

$$\mathscr{E}^{\times}_{p,j}(x) = \sum_{r \geq 0} (-1)^r Z_{2r+1} p^{2r} \binom{(p-1)x + j}{2r} \times \sum_{\substack{0 \leq a < p \\ 2a+1 \neq p}} (-1)^a (2a+1)^{j-2r} \left((2a+1)^{p-1}\right)^x.$$

theorem D.62 (Odd prime analytic Euler branch). The series defining $\mathscr{E}^{\times}_{p,j}$ converges in the Tate algebra on $\mathbb{Z}_p$. For every nonnegative integer x,

$$\boxed{\mathscr{E}^{\times}_{p,j}(x) = \left(1 - \sigma_p p^{(p-1)x+j}\right) E_{(p-1)x+j}.}$$

Proof. Every retained base is a unit, and its $(p-1)$st power lies in $1 + p\mathbb{Z}_p$. The polynomial

$$p^{2r} \binom{(p-1)x + j}{2r}$$

has Gauss norm at most

$$p^{-2r+v_p((2r)!)}.$$

This tends to zero because p is odd. The remaining factors have bounded Gauss norm. Thus the series converges in the Tate algebra. At a nonnegative integer x, the binomial factor vanishes for large r, and theorem D.61 gives the value. □

remark D.63. The analytic function is defined by its unit source series. The expression $p^{(p-1)x+j}$ in the interpolation identity is evaluated only at nonnegative integers. No analytic function $x \mapsto p^x$ on all of $\mathbb{Z}_p$ is used.

D.6.7 Direct dyadic interpolation

For $x \in \mathbb{Z}_2$, define

$$\boxed{\mathscr{R}(x) = \frac{9^{x+1} - 1}{8(x+1)(2x+1)}.}$$

The apparent singularity at $x = -1$ is removable, with value $-\log(9)/8$.

lemma D.64 (Analytic dyadic source). The function $\mathscr{R}$ belongs to the dyadic Tate algebra and takes values in $\mathbb{Z}_2$. For every nonnegative integer n,

$$\boxed{\mathscr{R}(n) = R_{2n}.}$$

Proof. Put $y = x + 1$ and let $\ell = \log 9$. Since $v_2(\ell) = 3$,

$$\frac{9^y - 1}{8y} = \frac{\ell}{8} \sum_{m \geq 0} \frac{\ell^m}{(m+1)!} y^m.$$

The coefficient of y^m has valuation

$$3m - v_2((m+1)!),$$

which is nonnegative and tends to positive infinity. The quotient therefore belongs to $\mathbb{Z}_2\langle y\rangle$. The factor $2x+1 = 1+2x$ is a unit in $\mathbb{Z}_2\langle x\rangle$, with inverse $\sum_{m\geq 0}(-2x)^m$. Hence $\mathscr{R}$ is integral and analytic. The integer values follow from the definition of R_{2n}. □

Define

$$\boxed{\mathscr{E}_2(x) = \sum_{r\geq 0} \kappa_{2,r}\binom{2x}{2r}\mathscr{R}(x-r).}$$

theorem D.65 (Direct dyadic Euler interpolation). The series defining $\mathscr{E}_2$ converges in the dyadic Tate algebra. For every nonnegative integer n,

$$\boxed{\mathscr{E}_2(n) = E_{2n}.}$$

Truncation after depth M is uniformly valid modulo 2^{4M+3} on $\mathbb{Z}_2$.

Proof. The monomial Gauss norm of $\binom{2x}{2r}$ is at most $2^{v_2((2r)!)}$. Since

$$v_2(\kappa_{2,r}) = 4r-1,$$

the product tends to zero in Gauss norm. Translation by the integer r preserves the unit disc, so $\mathscr{R}(x-r)$ remains an integral analytic function with uniformly bounded supremum norm. The series therefore converges in the Tate algebra.

At a nonnegative integer n, every term with $r > n$ vanishes and theorem D.39 remains. For arbitrary $x \in \mathbb{Z}_2$, the integer valued polynomial $\binom{2x}{2r}$ and the source value $\mathscr{R}(x-r)$ lie in $\mathbb{Z}_2$. Every omitted term after depth M is therefore divisible by 2^{4M+3}. This proves the uniform truncation statement. □

theorem D.66 (Unified local interpolation statement). The odd branch $\mathscr{E}^{\circ\times}_{p,j}$ and the dyadic branch $\mathscr{E}_2$ are analytic local realizations of the source and kernel law in theorem D.40. At integer arguments they recover, respectively,

$$\left(1-\sigma_p p^n\right) E_n$$

on each even odd prime character branch and

$$E_{2n}$$

on the dyadic even branch. The two realizations have distinct source corrections and distinct kernel valuation slopes.

Proof. Combine theorems D.62 and D.65 with the source and kernel definitions in theorem D.40. □

D.7 Polyphase refinement of local Mahler coordinates

Polyphase refinement replaces one local branch by finitely many digit branches while preserving exact synthesis. For sequences it is residue analysis. For local functions it is the digit tree of $\mathbb{Z}_p$. In Mahler coordinates it becomes a finite triangular kernel.

D.7.1 Discrete analysis and exact synthesis

For a sequence $a = (a_n)_{n\geq 0}$ and $M \geq 2$, define

$$a_r^{[M]}(q) = a_{Mq+r} \qquad (0 \leq r < M).$$

The analysis map is

$$\mathcal{P}_M a = \left(a_0^{[M]}, \ldots, a_{M-1}^{[M]}\right).$$

Synthesis interleaves the branches by the rule

$$a_{Mq+r} = a_r^{[M]}(q).$$

theorem D.67 (Exact discrete polyphase reconstruction). The analysis map $\mathcal{P}_M$ is a bijection between sequences and ordered families of M sequences. If

$$A(z) = \sum_{n\geq 0} a_n z^n$$

and

$$A_r^{[M]}(w) = \sum_{q\geq 0} a_{Mq+r} w^q,$$

then

$$\boxed{A(z) = \sum_{r=0}^{M-1} z^r A_r^{[M]}(z^M).}$$

Proof. Every nonnegative integer has a unique expression $Mq + r$ with $0 \leq r < M$. Group the terms of the ordinary generating function by this residue. □

theorem D.68 (Nested discrete refinement). For positive integers M, N,

$$\boxed{\left(a_r^{[M]}\right)_s^{[N]} = a_{r+Ms}^{[MN]}.}$$

For a mixed radix path $(M_1, \ldots, M_\ell)$ with digits $0 \leq s_i < M_i$, the final address is

$$\boxed{r(\boldsymbol{s}) = s_1 + M_1 s_2 + \cdots + \left(\prod_{i=1}^{\ell-1} M_i\right) s_\ell.}$$

Only the product radix and accumulated address determine the final branch.

Proof. Direct substitution gives

$$\left(a_r^{[M]}\right)_s^{[N]}(q) = a_{M(Nq+s)+r}.$$

Iterate this identity along the mixed radix path. □

D.7.2 Carry convolution and the shift matrix

Let $c = a * b$ be the ordinary Cauchy convolution. Write A_r, B_r, and C_r for the phase ordinary generating functions.

proposition D.69 (One carry convolution law). For every residue r,

$$\boxed{C_r(w) = \sum_{u+v=r} A_u(w)B_v(w) + w \sum_{u+v=r+M} A_u(w)B_v(w).}$$

The first sum has no phase carry. The second has one coarse index carry.

Proof. Write the input indices as $Mq+u$ and $Mq'+v$. Since $0 \le u, v < M$, their sum has residue $u+v$ when $u+v < M$, and residue $u+v-M$ with one additional coarse index when $u+v \ge M$. □

theorem D.70 (Polyphase shift matrix). Let E be the scalar shift $(Ea)_n = a_{n+1}$, and let E_q shift the coarse index. Then

$$\boxed{\mathcal{P}_M(Ea) = \mathsf{S}_M(E_q)\mathcal{P}_M a,}$$

where

$$\mathsf{S}_M(E_q) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ E_q & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

It also satisfies

$$\boxed{\mathsf{S}_M(E_q)^M = E_q I_M.}$$

Proof. For $r < M-1$,

$$(Ea)_{Mq+r} = a_{Mq+r+1} = a_{r+1}^{[M]}(q).$$

For $r = M-1$,

$$(Ea)_{Mq+M-1} = a_{M(q+1)} = E_q a_0^{[M]}(q).$$

After M shifts every phase has advanced by one coarse index. □

D.7.3 Local branches and digit reconstruction

For $a \ge 1$ and $0 \le s < p^a$, define

$$f_{a,s}(x) = f(p^a x + s).$$

theorem D.71 (Exact local branch reconstruction). The family $\{f_{a,s}\}_{0 \le s < p^a}$ determines f uniquely on $\mathbb{Z}_p$. More precisely, if $y \in s + p^a\mathbb{Z}_p$, then

$$\boxed{f(y) = f_{a,s}\left(\frac{y-s}{p^a}\right).}$$

For $a, b \ge 1$,

$$\boxed{(f_{a,s})_{b,t} = f_{a+b,s+p^a t}.}$$

Thus local refinement is exactly the p adic digit tree.

Proof. The cosets $s + p^a\mathbb{Z}_p$ form a disjoint partition of $\mathbb{Z}_p$. The first formula is the inverse affine change of variables on one coset. The second follows from

$$p^a(p^b x + t) + s = p^{a+b}x + p^a t + s.$$

□

D.7.4 The affine Mahler kernel

Let

$$\Phi_M(z) = (1+z)^M - 1.$$

For $M \geq 1$, $s \geq 0$, and $r, k \geq 0$, define

$$\boxed{\Theta_{M,s}(r,k) = [z^r](1+z)^s \Phi_M(z)^k.}$$

Equivalently,

$$\boxed{\Theta_{M,s}(r,k) = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} \binom{Mj+s}{r}.}$$

It vanishes unless

$$k \leq r \leq Mk + s.$$

proposition D.72 (Edge coefficients of the affine kernel). For every $M \geq 1$, $s \geq 0$, and $k \geq 0$,

$$\boxed{\Theta_{M,s}(k,k) = M^k, \qquad \Theta_{M,s}(Mk+s,k) = 1.}$$

For binary radix and zero address,

$$\boxed{\Theta_{2,0}(k+j,k) = 2^{k-j}\binom{k}{j} \qquad (0 \leq j \leq k).}$$

Hence

$$\boxed{v_2\left(\Theta_{2,0}(k+j,k)\right) = k - j + v_2\binom{k}{j}.}$$

Proof. The lowest degree term of $\Phi_M(z)$ is Mz, and its highest degree term is z^M. Multiplication by $(1+z)^s$ gives the two edge coefficients. When $M = 2$ and $s = 0$,

$$\Phi_2(z)^k = (2z + z^2)^k = z^k(2+z)^k.$$

Taking the coefficient of z^{k+j} gives the binary formula. □

theorem D.73 (Mahler polyphase transfer). Let

$$f(x) = \sum_{r \geq 0} c_r \binom{x}{r}$$

be a convergent Mahler expansion. For

$$g_s(q) = f(Mq + s),$$

one has

$$\boxed{g_s(q) = \sum_{k\geq 0} d_{s,k} \binom{q}{k},}$$

where

$$\boxed{d_{s,k} = \sum_{r=k}^{Mk+s} c_r \Theta_{M,s}(r,k).}$$

Every refined coefficient uses a finite window of the original Mahler coefficients.

Proof. Use

$$(1+z)^{Mq+s} = (1+z)^s \, (1+\Phi_M(z))^q = \sum_{k\geq 0} \binom{q}{k} (1+z)^s \Phi_M(z)^k.$$

Taking the coefficient of z^r gives the affine expansion of $\binom{Mq+s}{r}$. Since $c_r \to 0$ and every $d_{s,k}$ uses indices $r \geq k$, the transformed coefficients also tend to zero. □

theorem D.74 (Kernel composition). For positive integers M, N,

$$\boxed{\sum_{k\geq 0} \Theta_{M,s}(r,k)\Theta_{N,t}(k,\ell) = \Theta_{MN,Mt+s}(r,\ell).}$$

The sum is finite.

Proof. Apply theorem D.73 first to $x = Mq + s$ and then to $q = Ny + t$. The composite substitution is $x = MNy + Mt + s$. Uniqueness of Mahler coefficients gives the identity. □

D.7.5 Valuation transport

theorem D.75 (Polyphase valuation ledger). For every refined coefficient,

$$\boxed{v_p(d_{s,k}) \geq \min_{k\leq r\leq Mk+s} \left(v_p(c_r) + v_p(\Theta_{M,s}(r,k))\right).}$$

If $v_p(c_r) \geq w(r)$ for a nondecreasing function w, then

$$\boxed{v_p(d_{s,k}) \geq w(k).}$$

Thus every monotone Mahler precision ladder survives arbitrary polyphase refinement.

Proof. Apply the nonarchimedean triangle inequality to the finite coefficient formula. Since $r \geq k$, monotonicity gives $w(r) \geq w(k)$, and the affine kernel is integral. □

theorem D.76 (Prime power sparsification). **Editorial hypothesis correction.** The affine-kernel congruence below needs no change, but its consequence for transformed Mahler coefficients requires $c_r \in \mathbb{Z}_p$ for every r. Convergence of a $\mathbb{Q}_p$-valued Mahler expansion alone is insufficient. For $p = 2$,

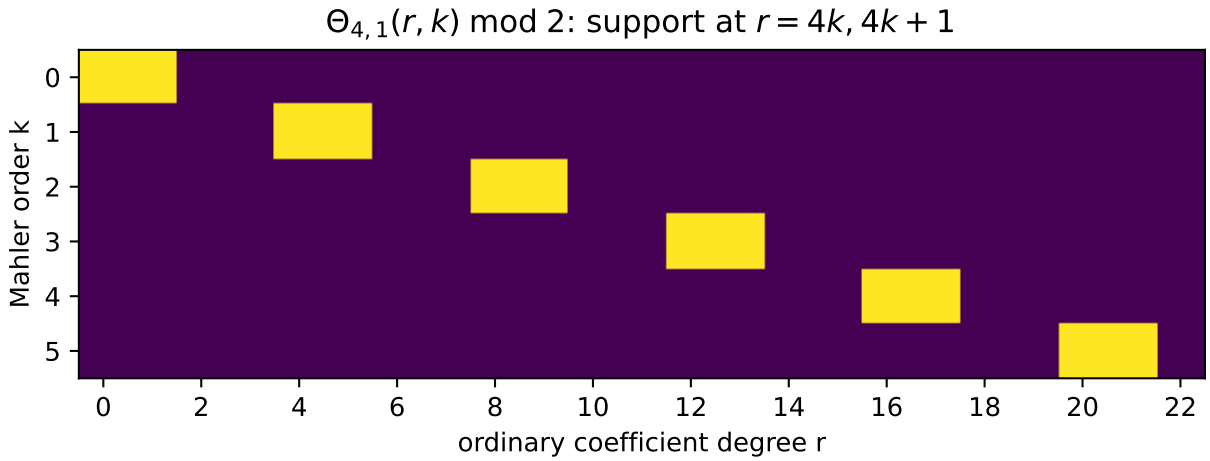


Figure 8. New reconstruction. The affine Mahler kernel has finite triangular support. At a prime power radix, its reduction is confined to a short Frobenius window.

$M = 2$, $s = 0$ and $f(x) = x/2$, one has $c_1 = 1/2$, $c_2 = 0$, while $g(q) = q$ has $d_1 = 1$. Thus $d_1 \not\equiv c_2 \pmod 2$. The coefficient consequence is read with integral original coefficients.

Let $M = p^a$. Then

$$\boxed{\Theta_{p^a,s}(r,k) \equiv \binom{s}{r - p^a k} \pmod p,}$$

where the binomial coefficient is zero outside its natural range. Consequently,

$$\boxed{d_{s,k} \equiv \sum_{u=0}^{s} \binom{s}{u} c_{p^a k+u} \pmod p.}$$

In particular,

$$\boxed{d_{0,k} \equiv c_{p^a k} \pmod p.}$$

Proof. In characteristic p,

$$(1+z)^{p^a} - 1 \equiv z^{p^a}.$$

Substitute this congruence in the definition of $\Theta_{p^a,s}$. □

D.7.6 Precision amplification

theorem D.77 (Scalar polyphase precision amplification). Let $u \in \mathbb{Z}_p^\times$, let $a, r \geq 0$, and put

$$H_a = \begin{cases} (p-1)p^a, & p \text{ odd}, \\ 2^{a+1}, & p = 2. \end{cases}$$

Then

$$\boxed{\Delta_{H_a}^r u^n = u^n \left(u^{H_a} - 1\right)^r.}$$

For odd p,

$$\boxed{v_p\left(\Delta_{H_a}^r u^n\right) \geq (a+1)r.}$$

For $p = 2$ and odd u,

$$\boxed{v_2\left(\Delta_{H_a}^r u^n\right) \geq (a+3)r.}$$

Equivalently, every additional prime power digit increases the certified Mahler slope by one.

Proof. One difference with step H_a multiplies the unit mode by $u^{H_a} - 1$. Iterate. The valuation estimates are those of corollary D.53. □

lemma D.78 (Matrix power amplification). **Editorial hypothesis correction.** Require $L \geq 1$. At $L = 0$ the standalone claim is false: for $p = 3$, $d = 1$ and $X = 2 = 1 + B$, one has $X^3 - 1 = 7 \notin 3\mathbb{Z}_3$. With $L \geq 1$, every higher binomial term in the retained proof has the required valuation, including $p = 2$.

If

$$X = I + p^L B$$

with $B \in M_d(\mathbb{Z}_p)$, then for every $a \geq 0$ there is $B_a \in M_d(\mathbb{Z}_p)$ such that

$$\boxed{X^{p^a} = I + p^{L+a} B_a.}$$

Proof. Induct on a. Expand the pth power of $I + p^{L+a} B_a$. The linear term has valuation at least $L + a + 1$, and every higher term has at least that valuation. □

theorem D.79 (Matrix polyphase precision amplification). **Editorial hypothesis correction.** Require $L \geq 1$, $C, B \in M_d(\mathbb{Z}_p)$, $v_0, w \in \mathbb{Z}_p^d$, $H > 0$, and $a, j, n, r \in \mathbb{N}$. These are the hypotheses of the two corrected matrix statements used by the retained proof. Without integral observation vectors or positive L, the preceding counterexamples invalidate the claimed precision.

Assume

$$C^H = I + p^L B.$$

Put $H_a = Hp^a$. Then

$$C^{H_a} = I + p^{L+a} B_a$$

for an integral matrix B_a. Every refined packet

$$b_{a,j}(x) = a_{H_a x + j}$$

has Mahler coefficients satisfying

$$\boxed{v_p(c_{a,j,r}) \geq (L+a)r.}$$

Equivalently,

$$\boxed{\Delta_{H_a}^r a_n \equiv 0 \pmod{p^{(L+a)r}}.}$$

Proof. Combine lemma D.78 and theorem D.55. □

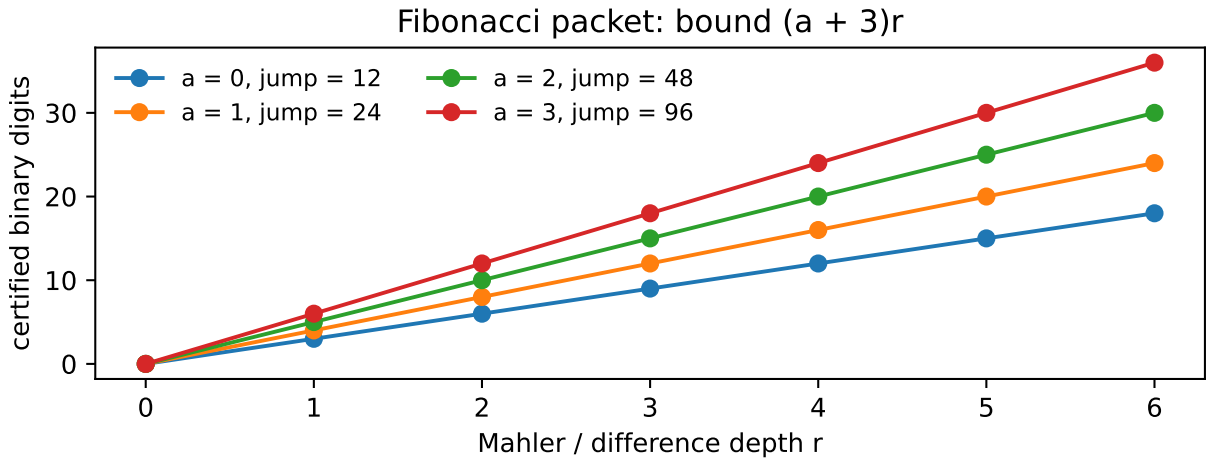


Figure 9. New reconstruction. Every additional prime power refinement digit increases the certified Mahler slope of a principal unit matrix packet by one.

example D.80 (Refined dyadic Fibonacci ladder). For the Fibonacci companion matrix

$$C = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix},$$

one has

$$C^{12} = I + 8 \begin{pmatrix} 29 & 18 \\ 18 & 11 \end{pmatrix}.$$

Therefore

$$\boxed{\Delta^{r}_{12\cdot 2^{a}} F_n \equiv 0 \pmod{2^{(a+3)r}}.}$$

Every dyadic refinement doubles the step and adds one guaranteed binary digit at each Mahler depth.

D.7.7 Character and polyphase coordinates

For odd p, write

$$\boxed{n = j + (p-1)(s + p^{a} q),}$$

with $0 \le j < p-1$ and $0 \le s < p^{a}$. The address j records the finite character. The address s records the principal unit digit. The variable q carries the remaining Mahler expansion.

At the dyadic prime, write

$$\boxed{n = j + 2(s + 2^{a} q),}$$

with $j \in \{0, 1\}$ and $0 \le s < 2^{a}$. The sign coordinate replaces the odd prime character coordinate.

proposition D.81 (Local phase lattice). For a unit mode u^{n}, the odd prime coordinate gives

$$\boxed{u^{n} = u^{j+(p-1)s} \left(u^{(p-1)p^{a}}\right)^{q}.}$$

The dyadic coordinate gives

$$\boxed{u^{n} = u^{j+2s} \left(u^{2^{a+1}}\right)^{q}.}$$

Thus finite character phase, local digit address, and Mahler depth are separate coordinates.

Proof. Substitute the two displayed decompositions of n. □

corollary D.82 (Polyphase refinement of the Euler branches). The analytic branches in theorems D.62 and D.65 admit exact refinements under $x = p^a q + s$. Their refined Mahler coefficients are obtained from $\Theta_{p^a,s}$. The original monotone valuation ladders are preserved, and unit mode or invertible matrix representations gain one certified digit per refinement level.

Proof. Apply theorems D.73, D.75 and D.79 and corollary D.53. □

remark D.83. Exact refinement does not imply that every branch is simpler. A simplification theorem requires an additional invariant, such as Frobenius sparsity, valuation gain, spectral collision, or a vanishing transfer block.

D.8 Arithmetic lift obstructions

A congruence class may solve an equation to one precision and fail at the next. The correct obstruction is not an arbitrary residual vector. It is the residual vector modulo the image of the declared correction map.

D.8.1 Linear lifting

Let p be a prime. Let

$$M \in M_{r,d}(\mathbb{Z}_p), \qquad b \in \mathbb{Z}_p^r.$$

Suppose that $x_L \in \mathbb{Z}_p^d$ satisfies

$$Mx_L \equiv b \pmod{p^L}$$

for some $L \geq 1$. Define the reduced residual

$$\rho_L(x_L) = \frac{Mx_L - b}{p^L} \pmod{p} \in \mathbb{F}_p^r.$$

Write $\overline{M}$ for the reduction of M modulo p.

definition D.84 (Linear lift obstruction). The linear obstruction at depth L is

$$\boxed{\mathrm{Ob}_L^{\mathrm{lin}}(x_L) = [\rho_L(x_L)] \in \mathrm{coker}(\overline{M}).}$$

theorem D.85 (One level linear lifting). A class represented by x_L lifts to a solution modulo p^{L+1} if and only if

$$\boxed{\mathrm{Ob}_L^{\mathrm{lin}}(x_L) = 0.}$$

More precisely, a correction $y \in \mathbb{F}_p^d$ gives a lift

$$x_{L+1} = x_L + p^L y \pmod{p^{L+1}}$$

if and only if

$$\boxed{\overline{M}y = -\rho_L(x_L).}$$

When one correction exists, the set of all correction classes is a coset of $\ker \overline{M}$.

Proof. Modulo p^{L+1},

$$M(x_L + p^L y) - b = Mx_L - b + p^L My.$$

After division by p^L and reduction modulo p, the lift condition becomes

$$\rho_L(x_L) + \overline{M}y = 0.$$

This equation is solvable exactly when the residual lies in the image of $\overline{M}$. If y_0 is one solution, every solution is $y_0 + z$ with $z \in \ker \overline{M}$. □

proposition D.86 (Representative independence). The class $\mathrm{Ob}_L^{\mathrm{lin}}(x_L)$ depends only on the solution class of x_L modulo p^L.

Proof. Replace x_L by $x_L + p^L z$. The reduced residual changes by $\overline{M}z$. Its class in the cokernel is unchanged. □

corollary D.87 (Exact number of linear children). If $\rho = \mathrm{rank}_{\mathbb{F}_p} \overline{M}$, then a solvable depth L class has exactly

$$\boxed{p^{d-\rho}}$$

children modulo p^{L+1}.

Proof. The correction set is a coset of a vector space of dimension $d - \rho$. □

example D.88 (A visible linear obstruction). Take $p = 3$, $L = 1$, and

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

The class $x_1 = (0, 0)$ solves

$$Mx_1 \equiv b \pmod 3$$

for both

$$b_1 = (-3, 0) \qquad \text{and} \qquad b_2 = (0, -3).$$

For b_1, the reduced residual is $(1, 0)$, which belongs to the image of $\overline{M}$. There are three correction classes, given by

$$y_1 + y_2 = 2 \pmod 3.$$

For b_2, the residual is $(0, 1)$, whose cokernel class is nonzero. No lift modulo nine exists because the second row of M remains zero.

D.8.2 Polynomial lifting

Let

$$F = (F_1, \dots, F_r) : \mathbb{Z}_p^d \longrightarrow \mathbb{Z}_p^r$$

be a polynomial map.

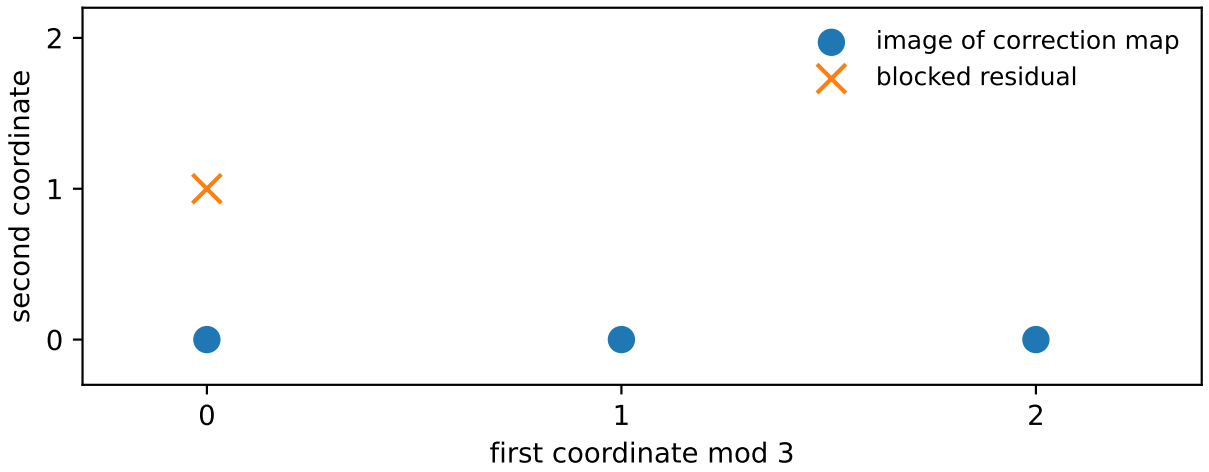


Figure 10. New reconstruction. A residual vector lifts precisely when its class vanishes in the cokernel of the reduced correction map.

Editorial coefficient clarification. Here polynomial means $F_i \in \mathbb{Z}_p[X_1, \dots, X_d]$, not merely an integer-valued polynomial with coefficients in $\mathbb{Q}_p$. This makes the Jacobian and Taylor remainder integral and their reduction well typed.

Suppose

$$F(x_L) \in p^L \mathbb{Z}_p^r.$$

Define

$$\rho_L^F(x_L) = \frac{F(x_L)}{p^L} \pmod{p}.$$

Let $J_F(x_L)$ denote the Jacobian matrix and write $\overline{J}_F(x_L)$ for its reduction modulo p.

definition D.89 (Polynomial lift obstruction). The polynomial obstruction at depth L is

$$\boxed{\mathrm{Ob}_L^F(x_L) = [\rho_L^F(x_L)] \in \operatorname{coker}(\overline{J}_F(x_L)).}$$

theorem D.90 (One level polynomial lifting). Let $L \geq 1$. A class represented by x_L lifts to a zero of F modulo p^{L+1} if and only if

$$\boxed{\mathrm{Ob}_L^F(x_L) = 0.}$$

A correction $y \in \mathbb{F}_p^d$ gives a lift if and only if

$$\boxed{\overline{J}_F(x_L) y = -\rho_L^F(x_L).}$$

When a correction exists, the correction classes form a coset of

$$\ker \overline{J}_F(x_L).$$

Proof. The polynomial Taylor expansion gives

$$F(x_L + p^L y) \equiv F(x_L) + p^L J_F(x_L) y \pmod{p^{L+1}}.$$

Every term of degree at least two in the increment contains p^{2L}, which is divisible by p^{L+1}. Divide by p^L, reduce modulo p, and apply the same cokernel criterion as in theorem D.85. □

proposition D.91 (Polynomial representative independence). The class $\mathrm{Ob}_L^F(x_L)$ depends only on the class of x_L modulo p^L.

Proof. If x_L is replaced by $x_L + p^L z$, Taylor expansion shows that the reduced residual changes by an element of the image of $\bar{J}_F(x_L)$. The reduced Jacobian itself is unchanged because $L \geq 1$. □

corollary D.92 (Simple root lifting). Assume $r = d$ and

$$\det J_F(x_L) \not\equiv 0 \pmod{p}.$$

Then the depth L solution has exactly one child modulo p^{L+1}. Iterating gives a unique p adic zero above the initial class.

Proof. The reduced Jacobian is invertible, so every residual has one correction. The resulting compatible sequence is Cauchy in $\mathbb{Z}_p^d$, and continuity of F gives a zero in the limit. Uniqueness holds at every depth. □

example D.93 (A singular polynomial obstruction). Let $p = 3$ and consider the class $x_1 = 0$ modulo three. For

$$F_0(x) = x^2,$$

the residual and the reduced derivative both vanish. Every correction $y \in \mathbb{F}_3$ gives

$$F_0(3y) \equiv 0 \pmod{9},$$

so the class has three children. For

$$F_1(x) = x^2 + 3,$$

the reduced derivative is still zero, but the reduced residual is one. Its class in the cokernel is nonzero, so no lift modulo nine exists.

This is the usual simple root mechanism behind Hensel lifting. The obstruction formulation also covers singular Jacobians, where some residual classes lift and others do not [22, 12, 20].

D.8.3 Packetwise obstruction splitting

theorem D.94 (Packet decomposition of a lift obstruction). Suppose

$$\mathbb{F}_p^d = \bigoplus_{a \in A} U_a, \qquad \mathbb{F}_p^r = \bigoplus_{a \in A} V_a,$$

and suppose a linear correction map $D : \mathbb{F}_p^d \to \mathbb{F}_p^r$ satisfies

$$D(U_a) \subseteq V_a$$

for every a. Write $D_a = D|_{U_a}$. Then

$$\boxed{\operatorname{coker} D \cong \bigoplus_{a \in A} \operatorname{coker} D_a.}$$

address j	k = 0	k = 1	k = 2	k = 3	k = 4
0	0	2	8	34	144
1	1	3	13	55	233
2	1	5	21	89	377

Figure 11. New reconstruction. Fibonacci residues F_{3k+j}, with $F_0 = 0$ and $F_1 = 1$. Each packet evolves by the same jump matrix C^3 and has a different initial address.

A residual lifts if and only if every packet obstruction vanishes.

Proof. The image of D is the direct sum of the images of the maps D_a. Taking the quotient of the codomain packet decomposition gives the stated cokernel decomposition. □

remark D.95. The obstruction theorems solve one declared local lifting step. They do not prove that a compatible branch exists across unrelated primes, that a global Diophantine point exists, or that an arithmetic representation is automorphic.

D.9 Recurrence packets and a common integral spine

Linear recurrences provide a setting where residue subsequences, local completions, and a common global source can all be described exactly. Background on recurrence sequences may be found in [11, 23].

D.9.1 Companion jumps

Let R be a commutative ring. Let

$$v_{n+1} = Cv_n, \qquad C \in M_d(R).$$

For a positive integer N and an address $j \in \{0, \dots, N-1\}$, define the residue state packet

$$u_k^{(j)} = v_{Nk+j}.$$

theorem D.96 (Companion jump recurrence). Every residue state packet satisfies

$$\boxed{u_{k+1}^{(j)} = C^N u_k^{(j)}.}$$

More explicitly,

$$\boxed{u_k^{(j)} = C^j (C^N)^k v_0.}$$

Proof. The state relation gives $v_n = C^n v_0$. Substitute $n = Nk + j$. □

Fibonacci state matrix	characteristic polynomial	roots
C = [[1, 1], [1, 0]]	$x^2 - x - 1$	α, β
C^3 = [[3, 2], [2, 1]]	$x^2 - 4x - 1$	α^3, β^3

Figure 12. New reconstruction. Passing to every Nth term raises each characteristic root to its Nth power.

We use the resultant convention

$$\operatorname{Res}_y(f, g) = \prod_{f(\alpha)=0} g(\alpha)$$

when f is monic, with roots counted with multiplicity.

theorem D.97 (Characteristic polynomial of the jump). Let

$$\chi_C(y) = \det(yI - C).$$

Then

$$\boxed{\chi_{C^N}(x) = \operatorname{Res}_y\bigl(\chi_C(y), x - y^N\bigr).}$$

The identity holds over the coefficient ring after interpreting the resultant universally.

Proof. After passage to an algebraic closure, let $\alpha_1, \dots, \alpha_d$ be the eigenvalues of C, counted with algebraic multiplicity. The resultant is

$$\prod_{i=1}^{d} (x - \alpha_i^N),$$

which is the characteristic polynomial of C^N. Both sides are polynomial expressions with integer coefficients in the entries of C, so the identity descends to every commutative coefficient ring. □

D.9.2 Second order recurrences

theorem D.98 (Residue subsequences of a second order recurrence). Suppose

$$x_{n+2} = P x_{n+1} - Q x_n$$

over a commutative ring. Define $V_0 = 2$, $V_1 = P$, and

$$V_{n+2} = P V_{n+1} - Q V_n.$$

For every positive integer N and every address j, the subsequence

$$u_k = x_{Nk+j}$$

satisfies

$$\boxed{u_{k+2} = V_N u_{k+1} - Q^N u_k.}$$

Proof. The companion matrix

$$C = \begin{pmatrix} P & -Q \\ 1 & 0 \end{pmatrix}$$

has trace P and determinant Q. The trace of C^N is V_N, while its determinant is Q^N. Cayley Hamilton applied to C^N gives

$$(C^N)^2 - V_N C^N + Q^N I = 0.$$

Apply this identity to the residue state packet and take its scalar coordinate. □

corollary D.99 (Fibonacci residue packets). Let $F_0 = 0$, $F_1 = 1$, and $F_{n+2} = F_{n+1} + F_n$. Let L_N denote the Lucas number. For every $N \geq 1$ and every address j,

$$\boxed{F_{N(k+2)+j} = L_N F_{N(k+1)+j} - (-1)^N F_{Nk+j}.}$$

Proof. Apply theorem D.98 with $P = 1$ and $Q = -1$. Then $V_N = L_N$. □

D.9.3 Finite and local periodicity

theorem D.100 (Periodicity of an invertible recurrence modulo a prime power). Let

$$C \in M_d(\mathbb{Z}), \qquad p \nmid \det C.$$

For every $L \geq 1$, every state orbit modulo p^L is purely periodic. Its period divides the order of C in

$$\mathrm{GL}_d(\mathbb{Z}/p^L\mathbb{Z}).$$

If $p \mid \det C$, every orbit is still eventually periodic.

Proof. When $p \nmid \det C$, multiplication by C is a permutation of the finite state set $(\mathbb{Z}/p^L\mathbb{Z})^d$. Every orbit of a permutation is periodic, and applying the group order relation for C gives the divisibility statement. Without invertibility, repeated application of a map on a finite set eventually repeats a state, after which the orbit is periodic. □

corollary D.101 (Residue packet periods). Under the invertibility hypothesis of theorem D.100, every residue packet with jump N has period dividing the order of C^N modulo p^L.

Proof. Use the jump matrix from theorem D.96. □

D.9.4 The common integral spine

theorem D.102 (Base change of an integral recurrence). Let

$$C \in M_d(\mathbb{Z}), \qquad v_0 \in \mathbb{Z}^d.$$

The integral orbit

$$v_n = C^n v_0$$

determines, by base change, compatible orbits in every ring R equipped with a homomorphism $\mathbb{Z} \to R$. In particular, the reductions modulo every positive integer and the orbits in every $\mathbb{Z}_p^d$ share the same companion matrix, characteristic polynomial, and integer recurrence relations.

Proof. Every coordinate of $C^n v_0$ is an integer polynomial in the entries of C and v_0. Applying a ring homomorphism commutes with matrix multiplication and preserves every polynomial relation with integer coefficients. □

corollary D.103 (Prime independent recurrence spine). An integral companion system gives a canonical common source for all of its local realizations. No choice of coefficient prime is required to define the recurrence or its residue packets.

Proof. Apply theorem D.102 to the companion matrix and its initial state. □

remark D.104. This theorem supplies a prime independent spine for integral linear recurrences. It does not prove that every unrelated family of local arithmetic packets arises from one global object.

D.10 Chinese remainder packet refinement

The Chinese remainder theorem gives an exact factorization of residue addresses. We state it at the level of sequence projectors so that it can be combined directly with the earlier coordinate systems.

definition D.105 (Residue projector). For a sequence $a = (a_n)_{n \geq 0}$, a positive integer N, and an address r modulo N, define

$$(P_{N,r}a)_n = \begin{cases} a_n, & n \equiv r \pmod{N}, \\ 0, & n \not\equiv r \pmod{N}. \end{cases}$$

proposition D.106 (Complete residue packet system). For fixed N, the projectors $P_{N,r}$ satisfy

$$P_{N,r}P_{N,s} = \delta_{rs}P_{N,r}$$

and

$$\boxed{\sum_{r \bmod N} P_{N,r} = I.}$$

Proof. Every nonnegative integer belongs to exactly one residue class modulo N. □

theorem D.107 (Coprime packet factorization). Let A and B be coprime positive integers. Let r_A be an address modulo A and r_B an address modulo B. Let r be the unique address modulo AB satisfying

$$r \equiv r_A \pmod{A}, \qquad r \equiv r_B \pmod{B}.$$

Then

$$\boxed{P_{AB,r} = P_{A,r_A}P_{B,r_B} = P_{B,r_B}P_{A,r_A}.}$$

Proof. The product retains exactly the indices satisfying both congruences. By the Chinese remainder theorem, those indices form one residue class modulo AB [14]. □

corollary D.108 (Several coprime moduli). Let $N_1, \dots, N_s$ be pairwise coprime and put

$$N = \prod_{j=1}^{s} N_j.$$

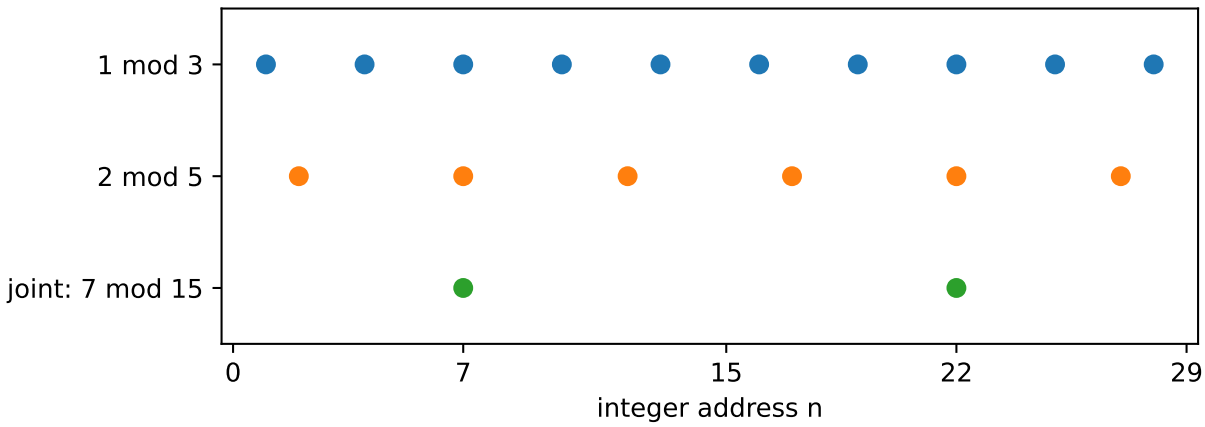


Figure 13. New reconstruction. A residue modulo a product of coprime moduli is the exact intersection of its local residue packets.

For compatible addresses r_j,

$$\boxed{P_{N,r} = \prod_{j=1}^{s} P_{N_j,r_j},}$$

where r is the unique combined address modulo N.

Proof. Iterate theorem D.107. □

standard fact D.109 (Finite local reconstruction). Let

$$N = \prod_{j=1}^{s} p_j^{a_j}$$

be the prime power factorization of a positive integer. The map

$$\mathbb{Z}/N\mathbb{Z} \longrightarrow \prod_{j=1}^{s} \mathbb{Z}/p_j^{a_j}\mathbb{Z}$$

is a ring isomorphism. Thus a compatible collection of prime power residues reconstructs one residue modulo N [14].

proposition D.110 (Dyadic and odd residue refinement). Let $N = 2^a m$ with m odd. Every residue packet modulo N factors uniquely as the product of one dyadic packet modulo 2^a and one odd packet modulo m.

Proof. The moduli 2^a and m are coprime. Apply theorem D.107. □

remark D.111. The residue address factors exactly, but valuation weights do not become identical across primes. The odd coordinate system of appendix D.5 and the dyadic coordinate system of appendix D.5 remain genuinely different integral charts.

Table 4. Status of the principal local arithmetic blocks.

Block	Status	Main boundary
Standard window period	Proved modulo the residue prime	Higher prime powers require inverse estimates
Unified local Euler transfer	Proved for every prime	Odd and dyadic integral charts remain different
Dyadic kernel and truncation	Exact weight and optimal uniform modulus proved	Row cancellation may give stronger values
Carmichael and Mahler transfer	Proved for scalar, polynomial exponential, matrix, and finite adelic systems	A jump congruence must first be available
Local Euler interpolation	Proved for depleted odd branches and the direct dyadic branch	The odd nonunit factor is only evaluated at integer interpolation points
Affine polyphase transfer	Exact analysis, synthesis, carry law, kernel composition, residual sparsity, and valuation preservation proved	General exact kernel valuations remain open
Lift obstruction and recurrence spine	Proved for the declared local problems and integral recurrences	Unrestricted global gluing is not proved

D.11 Verification and exact boundaries

The proofs in the article are symbolic. Exact computation tests formulas, finite ranges, and the source implementation. It does not replace a uniform proof.

D.11.1 The remaining congruential difference problem

Let $C_{mp,j}(n)$ be a congruential reciprocal coefficient and let

$$h = \frac{p-1}{m}.$$

The unit mode calculations suggest bounds for

$$v_p\left(\Delta_h^s C_{mp,j}(n)\right).$$

The theorems of appendices D.6 and D.7 determine the gain before the congruential triangular inverse is applied. They do not yet determine every valuation loss introduced by that inverse. The full family bound therefore remains open.

scope	conclusion or boundary
standard windows	finite periods and inverse coordinates
declared correction map	one-step cokernel obstruction
fixed integral recurrence	common source for its local images
arbitrary local families	global source NOT supplied
affine kernel valuations	general sharp higher laws remain open

Figure 14. New reconstruction. The proved local coordinate theory and the remaining inverse valuation, affine kernel, and global gluing problems.

D.11.2 Higher valuations of the affine kernel

The residual formula

$$\Theta_{p^a,s}(r,k) \equiv \binom{s}{r-p^a k} \pmod{p}$$

is exact. A closed formula for

$$v_p\left(\Theta_{p^a,s}(r,k)\right)$$

in terms of base p carries and the three digit addresses is not proved here. Such a formula would sharpen the min plus valuation ledger and could distinguish optimal refinement trees.

D.11.3 The remaining global gluing problem

The finite adelic theorem synchronizes a finite set of primes when one integral state system is already fixed. The recurrence theorem gives a common local source under the same assumption. Chinese remainder projectors reconstruct finite residue addresses. None of these results proves that an arbitrary collection of local analytic branches has a common global arithmetic origin.

D.11.4 Scope

The article uses classical work on Euler congruences, finite differences, local fields, Hurwitz series, Mahler interpolation, Lucas congruences, and recurrence sequences [8, 26, 27, 24, 9, 18, 17, 11]. The exact formulas and local transfer statements should be compared individually with that literature before any historical priority claim is made.

D.12 Conclusion

The local arithmetic theory now has one common architecture. The finite character coordinate identifies the torsion phase. The source and kernel product reconstructs the Euler coefficient. The kernel valuation records certified precision. Carmichael and Mahler coordinates describe the remaining principal unit

motion. Polyphase analysis resolves additional local digits, records carries and shifts exactly, and transports Mahler coefficients through a finite affine kernel.

The odd and dyadic cases agree at the level of convolution and differ at the level of integral geometry. At odd primes the source must be depleted before it becomes a unit mode interpolation. At the dyadic prime the source extends directly to an analytic integral function. The dyadic kernel has exact weight $4r - 1$, giving the optimal uniform modulus 2^{4M+3} after depth M.

Three tasks remain outside the proved region. The first is a complete finite difference estimate after the congruential triangular inverse. The second is an exact higher valuation theory for the affine Mahler kernel. The third is a global existence theorem for local systems that do not already arise from one integral source.

For the divisor-branch family, Theorem 3.7 proves the bound with the j! normalization and factorial valuation loss. No unrestricted family is asserted.

E Supplementary derivations

These supplements give the arithmetic derivations, examples and formal interfaces. The scope of each result is stated in the main text and the formal companion. A restricted intermediate construction is not the full conclusion of the chapter.

E.1 Coefficient coordinates

Scope and result

The following arguments develop the initial formal interfaces for the arithmetic constructions.

Seven of 24 named items proved in the manuscript now have Lean coverage in the stated scopes. Seventeen remain pending. Three classical inputs and two conjectures are registered separately. The development contains **89 explicit theorem declarations and 22 definitions across eight modules**. Supporting lemmas and concrete instances are included. These are not 89 independent discoveries.

Item	Covered mathematical content
1.2	Residue reconstruction and divisor refinement
2.1	Actual unit-EGF coordinate equivalence and transported projectors
2.2	Filtered finite coefficient reconstruction, including $\mathbb{Z}_p$
3.2	Constructed unique triangular inverse and original section identity
3.3	Universal denominator clearing
8.2	Lucas standard-window reduction and the diagonal unit
13.1	Exact normalization counterexample and sharp factorial loss

The correction is preserved

The normalized packet is not the same object as the packet multiplied by $j!$. The checked recurrence gives

$$C_{6,3}(1) = -\frac{1}{84}, \qquad C_{6,3}(2) = \frac{703}{5460},$$

$$C_{6,3}(2) - C_{6,3}(1) = \frac{64}{455}, \qquad v_3\left(\frac{64}{455}\right) = 0.$$

Thus the unrestricted normalized no-loss claim fails already at a valid one-step parameter instance. Multiplication by $3!$ restores one factor of three in this instance. The general corrected finite-difference theorem is **not** inferred from this calculation and is not yet formalized here.

1. Residue packets and actual EGF coordinates

Source items 1.2 and 2.1

For a sequence $a : \mathbb{N} \to R$, define the actual selector

$$(P_{N,j}a)(n) = \begin{cases} a(n), & n \bmod N = j, \\ 0, & \text{otherwise.} \end{cases}$$

The source proves idempotence for every selector, orthogonality for distinct standard residue labels, and, when $N > 0$,

$$\sum_{j=0}^{N-1} P_{N,j}a = a.$$

If $N \mid M$ and $M > 0$, it also proves

$$P_{N,j}a = \sum_{\substack{0 \le r < M \\ r \bmod N = j}} P_{M,r}a.$$

The formal power-series versions are proved coefficientwise for actual `PowerSeries` objects. There is no finite truncation of the sequence domain in these theorems.

A constructed inverse, not an inverse hypothesis

Let K be a characteristic-zero field. Define

$$\mathcal{E}(a) = \sum_{n \ge 0} \frac{a_n}{n!} t^n, \qquad \mathcal{D}(A)_n = n![t^n]A.$$

Both inverse laws $\mathcal{D}\mathcal{E} = I$ and $\mathcal{E}\mathcal{D} = I$ are proved. The ordinary power-series product then yields the binomial convolution

$$\mathcal{D}\big(\mathcal{E}(h)\mathcal{E}(a)\big)_n = \sum_{r=0}^{n} \binom{n}{r} h_r a_{n-r}.$$

For $h_0 = 1$, the formal inverse of $H = \mathcal{E}(h)$ is used to construct a sequence equivalence

$$U_H a = \mathcal{D}(H\mathcal{E}(a)), \qquad U_H^{-1} a = \mathcal{D}(H^{-1}\mathcal{E}(a)).$$

The equivalence has proved left and right inverse laws. The inverse's constant coefficient is proved to be one. The associated lower-triangular entries have diagonal one and vanish above the diagonal.

For the actual conjugated selectors

$$\widehat{P}_{N,j} = U_H P_{N,j} U_H^{-1},$$

idempotence, orthogonality, and finite reconstruction are proved from the constructed maps. No projector identity for the conjugates is inserted as an assumption.

Formal boundary. These are formal series and coefficient identities. They do not establish convergence for an analytic EGF. The characteristic-zero field assumption is explicit. Unrestricted factorial division in an arbitrary ring is not asserted.

Principal targets.

`ResiduePackets.packet_reconstruction`

`ResiduePackets.packet_refinement`

`EGF.coordinateEquiv,EGF.coordinate_forward,EGF.coordinate_inverse`

`EGF.transported_reconstruction`

2. Triangular inversion and denominator control

Source items 3.2 and 3.3

For a general lower-triangular coefficient array T over a field with nonzero diagonal, the code defines forward substitution by well-founded recursion:

$$a_n = \frac{b_n - \sum_{q<n} T_{nq} a_q}{T_{nn}}.$$

It proves that this constructed sequence solves every row and that every sequence solving those rows equals it. No existence or uniqueness premise is used.

The congruential coefficients

Over $\mathbb{Q}$, specialize to

$$T_{nq} = \binom{Nn+j}{Nq}, \qquad b_n = \delta_{n0}.$$

The diagonal is positive as an integer, hence nonzero. The resulting sequence $C_{N,j}(n)$ satisfies

$$C_{N,j}(0) = 1, \qquad \sum_{q=0}^{n} \binom{Nn+j}{Nq} C_{N,j}(q) = \delta_{n0}.$$

The displayed recurrence in the manuscript and the unique-solution theorem are consequences of this definition.

The formalization does not stop at a recurrence. In the variable $z = t^N$, construct

$$K_{N,j}(z) = \sum_{n\geq 0} \frac{j!}{(Nn+j)!} z^n, \qquad F_{N,j}(z) = \sum_{n\geq 0} \frac{C_{N,j}(n)}{(Nn)!} z^n.$$

It proves the factorial coefficient identity, then $F_{N,j} K_{N,j} = 1$ and therefore $F_{N,j} = K_{N,j}^{-1}$.

For $N > 0$ and $0 \leq j < N$, the independently defined original exponential section is

$$E_{N,j}(t) = \sum_{d \equiv j \pmod N} \frac{t^d}{d!}.$$

Using the actual power-series expansion homomorphism $z \mapsto t^N$, the code proves

$$E_{N,j}(t) = \frac{t^j}{j!} K_{N,j}(t^N), \qquad \boxed{F_{N,j}(t^N) E_{N,j}(t) = \frac{t^j}{j!}.}$$

Thus the recurrence is connected back to the original normalized EGF definition. The section vanishes at zero when $j > 0$. No inverse of that nonunit section is assumed.

Universal clearing

Define

$$D_{N,j}(n) = \prod_{r=1}^{n} \binom{Nr+j}{Nr}, \qquad D_{N,j}(0) = 1.$$

For every n, the code proves $D_{N,j}(n)C_{N,j}(n) \in \mathbb{Z}$. The general proof promotes each earlier clearing denominator to the common product, then cancels the new nonzero diagonal. This is a denominator bound, not a claim that every normalized coefficient is integral.

3. Filtered tails and the standard window

Source items 2.2 and 8.2

Filtration at a finite coefficient row

Let R be any commutative ring. For $w \in \mathbb{N} \cup \{\infty\}$ define

$$p^{[w]} = \begin{cases} p^w, & w < \infty, \\ 0, & w = \infty. \end{cases}$$

The infinity convention expresses a vanishing coefficient, not an infinite exponent operation. For any finite index set, every term with weight at least L is divisible by p^L. A finite-sum proof yields

$$\boxed{p^L \mid \sum_{r=0}^{n} \binom{n}{r} c_r p^{[w(r)]} S_{n-r} - \sum_{\substack{0 \le r \le n \\ w(r) < L}} \binom{n}{r} c_r p^{[w(r)]} S_{n-r}.}$$

The result is specialized to actual p-adic integers $R = \mathbb{Z}_p$ for prime p. Divisibility there is precisely the asserted membership in $p^L\mathbb{Z}_p$.

For an individual row the sum is finite, so the proof does not require $w(r) \to \infty$. This is a stronger coefficient-row scope, not a theorem about an infinite limiting process. A supporting lemma also proves a p^{2M+2} tail bound for a sum with visible factors p^{2m}. It is not counted as either Euler supercongruence theorem: their generating-function reconstruction identities remain pending.

Actual Lucas reduction

For prime p, all $m, n, q \in \mathbb{N}$, and $0 \le j < p$, the checked binomial congruence is

$$\boxed{\binom{mpn+j}{mpq} \equiv \binom{mn}{mq} \pmod{p}.}$$

The proof applies Mathlib's Lucas low-digit reduction to the actual natural binomial coefficients. It does not assume a pre-reduced row. The statement also permits $q > n$. If $m > 0$, both binomial coefficients then vanish.

Taking $q = n$ gives

$$\binom{mpn+j}{mpn} \equiv 1 \pmod{p}.$$

Nondivisibility by p and invertibility of that diagonal in $\mathbb{Z}/p\mathbb{Z}$ are proved. The separate supporting lemma $p \nmid j!$ for $j < p$ establishes the standard-window factorial-unit condition.

Dependency boundary. Lucas' theorem is a named classical input in the manuscript and is reused from the pinned Mathlib library. This block does not independently reconstruct Lucas' full digit theorem. The residue-independence result (8.3), uniform standard-window period (8.5), and the global finite-difference theorem (7.5) remain pending.

Principal targets.

`Filtered.padic_filtered_reconstruction`

`StandardWindow.lucas_row_reduction`

`StandardWindow.standard_diagonal_unit`

`StandardWindow.factorial_valuation_zero`

4. Sharp normalization boundary

Source item 13.1

The construction fixes the two coefficient conventions:

$$C_{N,j}(n), \qquad \widetilde{C}_{N,j}(n) = j!\, C_{N,j}(n).$$

The witness uses $p = 3$, $m = 2$, $N = mp = 6$, $j = 3$, $n = 1$ and difference step $h = (p-1)/m = 1$. The parameter conditions are checked explicitly, including odd primality, $m \mid p - 1$ and $j < N$.

The first row gives $84C_{6,3}(1) + 1 = 0$. The second row, evaluated by the same recursive definition, gives $C_{6,3}(2) = 703/5460$. Therefore

$$\Delta_1 C_{6,3}(1) = \frac{64}{455} \neq 0.$$

Neither numerator nor denominator is divisible by three, so its actual rational 3-adic valuation is zero. A further theorem negates the universal one-step no-loss statement by applying it to this valid tuple and deriving $1 \leq 0$.

For the unnormalized convention,

$$\Delta_1 \widetilde{C}_{6,3}(1) = 3!\,\frac{64}{455} = \frac{384}{455}, \qquad v_3\left(\frac{384}{455}\right) = 1.$$

The factorial's valuation is also proved:

$$v_3(3!) = 1, \qquad v_3\big(\Delta_1 C_{6,3}(1)\big) = 1 - v_3(3!) = 0.$$

This proves sharpness at the displayed normalization boundary. It does not establish the corrected global inequality for all parameters, and does not assert that every nonunit-factorial parameter tuple must violate every no-loss specialization. All valuation calculations here concern nonzero rationals.

Work remaining in the named proved register

Items	Pending mathematical content
4.1–4.3	Odd-power generating function, Euler reconstruction, odd-prime ladder
5.1–5.4	Dyadic inverse and source integrality, reconstruction, weighted ladder
6.2	Finite unit-decomposition certificates
7.2–7.5	Finite Mahler reconstruction, weighted differences, unit modes, global odd-prime differences
8.3, 8.5	Standard-window residue independence and uniform period
9.1, 9.2	Companion jumps and Fibonacci packet recurrence
10.1	Arithmetic CRT packet factorization

The two conjectures, 13.2 and 13.3, remain unproved and outside the 24-item register. The 17 pending entries are manuscript proof claims awaiting this project's Lean verification, not conjectures newly created by the formalization process.

5. Verification and reproducibility

Exact source pins, observed checks and retained failures

Module	Theorems	Definitions
ResiduePackets	8	2
Triangular	14	3
EGFCoordinates	21	6
FilteredReconstruction	7	1
CongruentialEuler	16	7
ExponentialSection	5	3
StandardWindow	6	0
NormalizationBoundary	12	0
Total	89	22

E.2 Odd Euler reconstruction

Scope and result

This supplement gives the formal constructions for Lemma 4.1, Theorem 4.2 and Corollary 4.3.

Ten of 24 named items proved in the manuscript are now covered in their recorded scopes. Fourteen remain pending. Three classical inputs and two conjectures are tracked separately. The cumulative corpus has **163 named theorem declarations and 37 explicit definitions in 13 modules**. The 74 new theorems include auxiliary identities and concrete examples. They are not 74 independent discoveries.

Item	New formal coverage	Theorems
4.1	Literal odd-power EGF and its even part	18
4.2	Integral Euler and signed zigzag coefficient bridges	29
4.2, 4.3	Exact reconstruction, matrix rows and truncation	16
Example	Exact scalar anchors and sharpness controls	11
	Total added across five modules	74

The five added modules are:

`OddPowerSeries`, `EulerCoefficients`, `ZigzagCoefficients`, `OddEulerReconstruction` and `OddEulerExamples`.

A useful scope strengthening

The algebraic proof needs *oddness*, not primality. The exact reconstruction and the integer divisibility statement are therefore checked for every positive odd length p, including $p = 1$ and odd composites. The manuscript's odd-prime statement follows as a specialization. This is a sharpening of assumptions in this development, not a historical-priority claim. Even lengths are not included. A checked row-zero counterexample shows why.

What is not completed

The weighted dyadic theory and general odd-prime finite-difference theorem lie outside this argument. No continuum convergence or analytic summability theorem follows from the formal power-series identities.

1. The actual odd-power generating function

Lemma 4.1; manuscript page 10

Work in the rational formal power-series ring $\mathbb{Q}[[t]]$. Define

$$S_n(p) = \sum_{a=0}^{p-1} (-1)^a (2a+1)^n \in \mathbb{Z}, \qquad F_p(t) = \sum_{n \geq 0} S_n(p) \frac{t^n}{n!}.$$

The formal exponential is Mathlib's `PowerSeries.exp`. Scalar substitution is its actual `rescale` homomorphism. We define

$$\exp_a(t) = e^{at}, \qquad \cosh_a(t) = \frac{e^{at} + e^{-at}}{2}, \qquad \operatorname{sech}(t) = \cosh_1(t)^{-1}.$$

The constant term of $\cosh_a$ is proved to be one, so the reciprocal used here is a unit inverse.

From finite sums to a series identity

Coefficient extraction proves

$$F_p(t) = \sum_{a=0}^{p-1} (-1)^a e^{(2a+1)t}.$$

An induction on the actual finite summation length establishes the telescoping identity

$$F_p(t) \cosh t = \frac{1 - (-1)^p e^{2pt}}{2}.$$

For odd p this gives, without a geometric-series convergence assumption,

$$\boxed{F_p(t) = \frac{1 + e^{2pt}}{2} \operatorname{sech} t.}$$

This is an equality of complete formal series, not a fixed-degree polynomial check.

Even projection is identified coefficientwise

For any formal series f, let $\mathrm{EvenPart}(f) = (f(t) + f(-t))/2$. Its nth coefficient is proved to be the coefficient of f when n is even and zero otherwise. Rescaling the inverse law proves $\operatorname{sech}(-t) = \operatorname{sech}(t)$. The exponential multiplication identity then gives

$$\cosh^2(pt) = \frac{2 + e^{2pt} + e^{-2pt}}{4},$$

and hence

$$\boxed{\sum_{r \geq 0} S_{2r}(p) \frac{t^{2r}}{(2r)!} = \cosh^2(pt) \operatorname{sech} t.}$$

Both the full and even EGF formulas are derived from the original integer sum. Neither is supplied as a hypothesis.

Principal targets.

`OddPower.source_as_exponentials`

`OddPower.source_times_cosh`

`OddPower.odd_source_egf`, `OddPower.odd_source_even_egf`

2. Integral Euler coefficients and the zigzag bridge

Coefficient foundations for Theorem 4.2

The first block already constructed $C_{N,j}(r)$ by the manuscript's triangular recurrence. At $(N, j) = (2, 0)$, every diagonal binomial coefficient equals one. Its denominator-clearing product is therefore one, and $C_{2,0}(r)$ is an integer.

The source defines E_{2r} as the integer numerator of that rational coefficient and proves that its rational cast equals $C_{2,0}(r)$. Set $E_{2r+1} = 0$. The actual exponential-section identity from the first block proves

$$\boxed{\sum_{n\geq 0} E_n \frac{t^n}{n!} = \operatorname{sech} t.}$$

Thus the integer sequence is connected to the intended signed Euler family, rather than merely assigned the name Euler.

The inverse coefficients are integral by construction

Define the integer binomial convolution

$$B_n = \sum_{r=0}^{n} \binom{n}{r} E_r E_{n-r}, \qquad b_m = B_{2m}.$$

The product formula for actual EGFs gives

$$\sum_{n\geq 0} B_n \frac{t^n}{n!} = \operatorname{sech}^2 t.$$

Odd coefficients vanish by parity. In particular, every b_m is an integer. This is the integrality required by the later congruence.

Connecting to the chapter's Z_n convention

The chapter defines Z_n through $\sec t + \tan t$. The Lean construction uses Mathlib's actual formal sine and cosine series and defines

$$\sec t = (\cos t)^{-1}, \quad \tan t = (\sin t)\sec t, \quad Z_n = n![t^n](\sec t + \tan t).$$

It proves the sine and cosine derivative identities, $\sin^2 t + \cos^2 t = 1$, and

$$(\tan t)' = \sec^2 t.$$

Sign substitution in the squared variable carries the hyperbolic cosine kernel to the actual cosine kernel. The same homomorphism carries their inverses and their squares. Extracting the even coefficient in the derivative of tangent yields

$$\boxed{b_m = (-1)^m Z_{2m+1}.}$$

This formula is *proved*, not assumed as an input to reconstruction. The formalization follows the generating-function definition of zigzag numbers. It does not additionally formalize their interpretation as counts of alternating permutations.

Principal targets.

`EulerCoefficients.euler_egf`, `EulerCoefficients.inverse_coeff_cast`

`Zigzag.derivative_tan,Zigzag.inverse_is_signed_zigzag`

3. Exact reconstruction and the triangular matrix

Theorem 4.2; manuscript page 11

For odd p, rescaling the already proved inverse law gives

$$\operatorname{sech}(pt)\cosh(pt) = 1.$$

Multiplying the even-source identity by $\operatorname{sech}^2(pt)$ therefore proves

$$\operatorname{sech}^2(pt)\left(\sum_{r\geq 0} S_{2r}(p)\frac{t^{2r}}{(2r)!}\right) = \operatorname{sech} t.$$

Actual EGF multiplication now gives the integer coefficient formula

$$\boxed{E_{2r} = \sum_{m=0}^{r}\binom{2r}{2m} b_m p^{2m} S_{2(r-m)}(p).}$$

The formal proof first extracts the sum over all indices and then proves that the odd summands vanish. A checked even-index summation identity supplies the displayed change of variables.

Substituting the independently proved signed-zigzag identity recovers the manuscript's notation. For every even degree n,

$$\boxed{E_n = \sum_{m=0}^{n/2}(-1)^m Z_{2m+1}\binom{n}{2m} p^{2m} S_{n-2m}(p).}$$

The Lean version uses $n = 2r$ to avoid ambiguous natural-number division. Every even natural degree has that form. Primality is not used in this equality.

The row action is connected to the matrix entries

Reversing the finite sum and using binomial symmetry gives the actual source-index row action

$$E_{2r} = \sum_{q=0}^{r} U_{r,q}(p) S_{2q}(p),$$

where

$$U_{r,q}(p) = \begin{cases}\binom{2r}{2q} b_{r-q} p^{2(r-q)}, & q \leq r,\\ 0, & q > r.\end{cases}$$

The row-action equality, the diagonal identity $U_{r,r} = 1$, and vanishing above the diagonal are all checked. The arithmetic weight is also explicit:

$$p^{2(r-q)} \mid U_{r,q}(p) \quad \text{in } \mathbb{Z}, \qquad 0 \leq q \leq r.$$

No inverse-coordinate identity, unit diagonal or weight bound is assumed for these rows.

Principal targets.

`OddEuler.odd_reconstruction_series`

`OddEuler.odd_euler_reconstruction`

`OddEuler.odd_euler_reconstruction_zigzag`

`OddEuler.row_action,OddEuler.row_diagonal,OddEuler.row_weight`

4. Uniform truncation and a sharp exact example

Corollary 4.3 and Example 4.4

For row $2r$ and depth M, the actual truncation is

$$T_M(p,2r) = \sum_{m=0}^{\min(M,r)} \binom{2r}{2m} b_m p^{2m} S_{2(r-m)}(p).$$

Every omitted term has $m \geq M+1$. Its other factors are integers, so the first-block omitted-tail theorem, now applied to the *proved Euler identity*, gives

$$\boxed{p^{2M+2} \mid E_{2r} - T_M(p,2r) \quad \text{in } \mathbb{Z}.}$$

This is not divisibility in $\mathbb{Q}$, where a nonzero divisor would make the assertion vacuous. The statement holds for every row and depth. For an odd prime it is precisely the claimed supercongruence. If $M \geq r$, complete retention is proved exact.

The displayed manuscript row is kernel-checked

The scalar anchors are derived from the same general coefficient recurrence:

$$(E_0, E_2, E_4, E_6, E_8) = (1, -1, 5, -61, 1385),$$

$$(b_0, b_1, b_2, b_3, b_4) = (1, -2, 16, -272, 7936).$$

The literal source sums at $p = 5$ give

$$(S_0, S_2, S_4, S_6, S_8) = (1, 49, 4705, 428689, 37665985).$$

Consequently,

$$T_2(5,8) = 2\,731\,001\,385, \qquad E_8 - T_2(5,8) = -2\,731\,000\,000.$$

The source proves both

$$5^6 \mid E_8 - T_2(5,8), \qquad 5^7 \nmid E_8 - T_2(5,8).$$

Thus the guaranteed power is attained in this example. No stronger uniform exponent is inferred.

Oddness cannot be silently dropped

For length two, $S_0(2) = 1 - 1 = 0$, whereas $E_0 = 1$. The row-zero reconstruction already fails. The source proves the failure of
the even-length identity.

Principal targets.

`OddEuler.odd_euler_supercongruence`

E.3 Dyadic reconstruction

Scope and result

This supplement develops the stated forms of Lemmas 5.1 and 5.2, Theorem 5.3 and Corollary 5.4.

Fourteen of 24 named items proved in the manuscript now have formal coverage. Ten remain pending. Three classical inputs and two conjectures are tracked separately. The cumulative corpus has **219 named theorem declarations and 53 explicit definitions in 17 modules**. The 56 added theorems include supporting identities and exact examples, not 56 independent discoveries.

New module	Scope	Theorems	Defs.
`DyadicIntegrality`	5.1, 5.2	21	8
`DyadicSeries`	5.1, 5.3	11	4
`DyadicReconstruction`	5.3, 5.4	10	4
`DyadicExamples`	Exact controls	14	0
Total added	Four named items	56	16

The completed mathematical chain

The inverse coefficients are defined from a genuine unit power series and connected to $z^2/\sinh^2 z$. The source is the chapter's explicit rational formula, not an abstract integral input. Its valuation calculation proves the stronger statement that *every even source coefficient is a* 2*-adic unit*. Formal-series multiplication and coefficient extraction then yield the Euler identity with the correct falling factorial. The omitted-tail theorem gives divisibility in the actual completed integer ring $\mathbb{Z}_2$.

Completion boundary

The Bernoulli closed form for the inverse coefficients has no dedicated universal formalization here. Formal-series identities do not establish analytic convergence. The global odd-prime finite-difference theorem is developed separately.

1. Integral coefficients and an exact unit source

Lemmas 5.1 and 5.2; manuscript page 14

Define the rational subring

$$\mathcal{O}_2 = \{q \in \mathbb{Q} : |q|_2 \le 1\}.$$

Its ring laws follow from the actual nonarchimedean norm. For every $q \in \mathcal{O}_2$, the code constructs its element in Mathlib's completed ring $\mathbb{Z}_2$, and proves that the underlying value in $\mathbb{Q}_2$ is the rational cast of q. No integrality claim is hidden in a coercion.

The unit kernel and its reciprocal

Use a squared formal variable u and define

$$k_m = \frac{2^{2m+1}}{(2m+2)!}, \qquad K(u) = \sum_{m\geq 0} k_m u^m, \qquad \beta_m = [u^m]K(u)^{-1}.$$

The constant term is proved to be one. The factorial valuation bound gives

$$v_2(k_m) = 2m + 1 - v_2((2m+2)!) \geq 0.$$

Thus every k_m belongs to $\mathcal{O}_2$. A general coefficientwise subring theorem is proved by strong induction using the actual unit-inverse coefficient formula. Applied to K, it establishes

$$\boxed{\beta_m \in \mathcal{O}_2 \quad \text{for every } m \geq 0.}$$

The reciprocal's coefficients are therefore proved integral. The proof does not assume them integral in order to obtain the final congruence. Their connection to the intended hyperbolic function is proved on the next page.

The source has valuation exactly zero

At even degree $2r$, define the literal rational source

$$R_{2r} = \frac{3^{2r+2} - 1}{4(2r+1)(2r+2)}.$$

Both numerator and denominator are nonzero. The pinned Mathlib lifting-the-exponent theorem gives

$$v_2(3^{2r+2} - 1) = 2 + v_2(2r+2).$$

Because $2r+1$ is odd, the denominator has the same valuation. Consequently,

$$\boxed{v_2(R_{2r}) = 0, \qquad |R_{2r}|_2 = 1.}$$

The code proves actual `IsUnit` status for the constructed $\mathbb{Z}_2$ element. This strengthens the manuscript's integrality conclusion without changing its source. The argument applies at $r = 0$, where $R_0 = 1$, as well as at every later row.

Principal targets.

`Dyadic.kernel_coefficient_integral`, `Dyadic.beta_integral`

`Dyadic.source_valuation`, `Dyadic.sourcePadic_isUnit`

2. Connecting the definitions to actual formal series

Generating-function foundations for Theorem 5.3

Work in $\mathbb{Q}[[t]]$. The formal exponential and rescaling homomorphism are the same checked objects used in the second block. Define

$$\sinh_a(t) = \frac{e^{at} - e^{-at}}{2}, \qquad \cosh_a(t) = \frac{e^{at} + e^{-at}}{2}.$$

Let $B(t) = K(t^2)^{-1} = \sum_{m\geq 0} \beta_m t^{2m}$ and let

$$F(t) = \sum_{r\geq 0} R_{2r} \frac{t^{2r}}{(2r)!}.$$

These series are constructed independently from the definitions on the preceding page.

No inverse of a zero-constant series is assumed

Coefficient extraction proves the cleared identity

$$t^2 K(t^2) = \frac{\cosh(2t) - 1}{2}.$$

A separate identity from the formal exponential gives

$$\sinh^2 t = \frac{\cosh(2t) - 1}{2}.$$

Since $K(t^2)$ has constant term one, its reciprocal exists as a unit. Multiplication then proves

$$\boxed{B(t)\sinh^2 t = t^2.}$$

This is the precise formal meaning of $B(t) = t^2/\sinh^2 t$. The forced quadratic zero is removed first. The proof does not claim that $\sinh t$ itself is an invertible formal power series.

The explicit rational source supplies its own identity

From the formula for R_{2r}, coefficient arithmetic proves

$$\boxed{t^2 F(t) = \frac{\cosh(3t) - \cosh t}{4}.}$$

The product relation for the formal hyperbolic cosines yields

$$F(t)\cosh t = K(4t^2).$$

This equality is proved after multiplication by t^2 and cancellation of that monomial in the formal-series ring. It is not an assumed source-transform relation.

Rescale the reciprocal by $t \mapsto 2t$. The unit inverse law and the preceding identity now give

$$\boxed{B(2t)F(t) = \operatorname{sech} t.}$$

The right side uses the same integral signed Euler sequence. These are equalities of full formal series, not fixed-degree experiments.

Principal targets.

`DyadicSeries.beta_original_generating_function`

`DyadicSeries.source_shift_identity,DyadicSeries.source_times_cosh`

`DyadicSeries.dyadic_reconstruction_series`

3. Exact reconstruction and genuine dyadic congruence

Theorem 5.3 and Corollary 5.4; manuscript pages 14–15

The reciprocal coefficients β_m use an *ordinary* power expansion. In contrast, the source and Euler series use exponential coefficients. Extracting degree $2r$ from $B(2t)F(t) = \operatorname{sech} t$ therefore gives

$$\boxed{E_{2r} = \sum_{m=0}^{r} \beta_m 2^{2m} (2r)_{2m} R_{2(r-m)},} \qquad (2r)_{2m} = \frac{(2r)!}{(2r-2m)!}.$$

The formal proof uses the actual natural-number descending factorial and proves its factorial-quotient identity for $2m \le 2r$. The odd coefficient terms vanish by a proved parity argument. The binomial EGF product then supplies the falling factorial, rather than treating it as an extra assumed factor.

Both row indexing by r and the literal even-degree version with hypothesis $\mathrm{Even}(n)$ are checked. All rows include degree zero. The exact identity is a rational equality with the Euler coefficient's integer-to-rational cast made explicit.

The finite truncation

For $M \ge 0$, define

$$T_M(2r) = \sum_{m=0}^{\min(M,r)} \beta_m 2^{2m} (2r)_{2m} R_{2(r-m)}.$$

The integral rational factors are embedded into $\mathbb{Z}_2$ using their proved witnesses. The cast of this $\mathbb{Z}_2$ truncation into $\mathbb{Q}_2$ is proved to equal the cast of the literal rational truncation.

Each omitted term has $m \ge M+1$. Its remaining factors lie in $\mathbb{Z}_2$, so the checked weighted-tail theorem proves

$$\boxed{2^{2M+2} \mid E_{2r} - T_M(2r) \quad \text{in } \mathbb{Z}_2.}$$

This is not divisibility in $\mathbb{Q}$ or $\mathbb{Q}_2$, where division by any nonzero scalar would make the claim vacuous. A second theorem supplies the equivalent rational statement

$$\boxed{\frac{E_{2r} - T_M(2r)}{2^{2M+2}} \in \mathcal{O}_2.}$$

It follows from actual $\mathbb{Z}_2$ divisibility and the proved rational/completed-field bridge. No precision assertion is passed in as a hypothesis.

Endpoints and scope

When $M \ge r$, the sum retains every term and $T_M(2r) = E_{2r}$ exactly, in both rational and dyadic coordinates. The zero-error case is included. No finite valuation of zero is used. This is a separate dyadic construction, not substitution of $p = 2$ into the odd-length theorem.

Principal targets.

`DyadicReconstruction.dyadic_reconstruction`

`DyadicReconstruction.dyadic_supercongruence`

`DyadicReconstruction.normalized_error_integral`

4. Exact examples and adversarial checks

Example 5.5 and controls against incorrect normalizations

The actual reciprocal recurrence gives

$$(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) = \left(1, -\frac{1}{3}, \frac{1}{15}, -\frac{2}{189}, \frac{1}{675}\right).$$

These values are obtained from the general formal-series definition. They are not substituted as independent input data for the reconstruction theorem. The source formula gives

$$(R_0, R_2, R_4, R_6, R_8) = \left(1, \frac{5}{3}, \frac{91}{15}, \frac{205}{7}, \frac{7381}{45}\right).$$

Using the retained Euler anchor $E_8 = 1385$, Lean checks the literal decomposition

$$E_8 = \frac{7381}{45} - \frac{6560}{3} + \frac{163072}{15} - \frac{204800}{9} + \frac{229376}{15} = 1385.$$

Truncations and actual valuations

Depth M	$T_M(8)$	Guaranteed exponent $2M+2$
0	7381/45	2
1	−91019/45	4
2	398197/45	6
3	−208601/15	8
4	1385	10

The checked nonzero error valuations are

$$E_8 - T_0(8) = \frac{54944}{45}, \qquad v_2(E_8 - T_0(8)) = 5,$$

$$E_8 - T_3(8) = \frac{229376}{15}, \qquad v_2(E_8 - T_3(8)) = 15.$$

They exceed the uniform guarantees. These two examples do not establish a universally stronger exponent.

What the negative controls rule out

Omitting the rescaling 2^{2m} already breaks the degree-two identity: the altered sum is 1, whereas $E_2 = -1$. Confusing ordinary beta coefficients with EGF coefficients would replace $\beta_1 = -1/3$ by $-2/3$. A checked inequality excludes that normalization. A deliberately false claim that the depth-zero degree-eight error has valuation at least six is rejected.

These three negative proof files are separate from the active mathematical modules. The verifier also retains the five earlier invalid controls, including the normalized all-residue counterexample. Rejection of an attempted proof is an engineering control. The actual counterexample theorems establish the mathematical failures.

Principal targets.

E.4 Mahler interpolation

Scope and result

This supplement develops the stated forms of Theorem 7.2, Corollary 7.3 and Theorem 7.4.

Seventeen of 24 named items proved in the manuscript now have formal coverage. Seven remain pending. Three classical inputs and two conjectures are tracked separately. The cumulative project contains **272 named theorem declarations and 60 explicit definitions across 21 modules**.

New module	Scope	Theorems	Defs.
`FiniteMahler`	7.2	19	3
`PadicDifferenceBounds`	7.3	12	2
`UnitModeDifferences`	7.4	9	0
`MahlerExamples`	Exact controls	13	2
Total added	Three items	53	7

The 53 additions include supporting lemmas and examples, not 53 independent discoveries. The classical Gregory–Newton and binomial-difference results are reused from pinned Mathlib. The new modules connect those results to the actual Paper IX coefficient definitions, interpolation scopes, valuation bounds and unit modes.

Two explicit specification boundaries

A finite sample through N determines $\Delta^s f(n)$ only when $n + s \le N$. For arbitrary orders, the finite-coefficient identity describes the constructed interpolant, not every unknown continuation of the sample. All three versions are separately checked.

The additive valuation is extended to $\mathrm{ord}_p(0) = \infty$. This avoids incorrectly applying Mathlib's finite-valued default at zero to an inequality that must also cover cancellation and identically zero differences.

1. Reconstruction and finite-data scope

Theorem 7.2; manuscript page 17

Let R be any commutative ring. The implementation uses the actual forward-difference operator

$$\Delta_h f(n) = f(n+h) - f(n), \qquad c_r = \Delta_1^r f(0).$$

For a full sequence $f : \mathbb{N} \to R$, pinned Mathlib's Gregory–Newton formula gives

$$\boxed{f(n) = \sum_{r=0}^{n} c_r \binom{n}{r}.}$$

The source applies the theorem to these exact definitions. A separate explicit formula relates an s-th difference to its original sampled values, so the data dependence is inspectable:

$$\Delta_h^s f(n) = \sum_{j=0}^{s} (-1)^{s-j} \binom{s}{j} f(n + jh).$$

Applying Newton reconstruction to $\Delta_1^s f$ and composing the iterated operators proves

$$\boxed{\Delta_1^s f(n) = \sum_{j=0}^{n} c_{s+j} \binom{n}{j}.}$$

The apparently infinite coefficient sum is finite at each natural argument. Neither continuity nor convergence is needed for these identities.

An independently defined interpolant

For a finite coefficient list, define

$$I_N(c)(n) = \sum_{r=0}^{N} c_r \binom{n}{r}.$$

The source proves exact coefficient extraction:

$$\Delta_1^s I_N(c)(0) = \begin{cases} c_s, & s \le N, \\ 0, & s > N. \end{cases}$$

It follows that $I_N(c)$ has the stated shifted-coefficient formula at every order, with zero coefficients beyond N. Every difference of order greater than N vanishes. Its coefficients are uniquely determined by its values at $0, \dots, N$.

What the finite sample does and does not determine

For $c_r = \Delta_1^r f(0)$ with $r \le N$, the source proves

$$I_N(c)(n) = f(n) \quad (n \le N), \qquad \Delta_1^s I_N(c)(n) = \Delta_1^s f(n) \quad (n + s \le N).$$

The condition is essential. The sequences $f(n) = n$ and $g(n) = 0$ agree on the sample $\{0\}$, but $\Delta f(0) = 1$ and $\Delta g(0) = 0$. This is a checked counterexample to unrestricted inference outside the observed stencil.

Principal targets.

`Mahler.finite_mahler_reconstruction`, `Mahler.coefficient_interpolant`

`Mahler.difference_from_coefficients`, `Mahler.finite_sample_difference`

2. Weighted bounds in the actual ring $\mathbb{Z}_p$

Corollary 7.3; manuscript page 18

Let p be prime. For $x \in \mathbb{Z}_p$, define

$$\operatorname{ord}_p(x) = \begin{cases} \infty, & x = 0, \\ v_p(x), & x \neq 0. \end{cases}$$

The nonzero branch uses Mathlib's actual p-adic valuation. The implementation proves its connection to genuine integral divisibility:

$$\boxed{L \leq \operatorname{ord}_p(x) \quad \Longleftrightarrow \quad p^L \mid x \text{ in } \mathbb{Z}_p, \qquad L \in \mathbb{N}.}$$

This is not divisibility in $\mathbb{Q}_p$, where a nonzero scalar can divide any field element. The zero case is proved separately, so zero does not spuriously violate a positive lower bound.

The source also proves multiplicativity of this extended order, the nonarchimedean lower bound for addition, and its finite-sum consequence. The natural-number cast is connected to the actual integer binomial valuation.

The binomial valuation is retained

Let $f : \mathbb{N} \to \mathbb{Z}_p$, let $c_r = \Delta_1^r f(0)$, and suppose

$$w(s+j) \leq \operatorname{ord}_p(c_{s+j}) \quad (0 \leq j \leq n), \qquad w : \mathbb{N} \to \mathbb{N} \cup \{\infty\}.$$

Set

$$B_p(w; n, s) = \min_{0 \leq j \leq n} \left\{ w(s+j) + v_p\binom{n}{j} \right\}.$$

Every binomial in this range is nonzero. The code proves

$$\boxed{B_p(w; n, s) \leq \operatorname{ord}_p(\Delta_1^s f(n)).}$$

This is the manuscript minimum after the substitution $r = s + j$. Coefficients outside this range have zero binomial multiplier. Weights equal to infinity are allowed, and a completely vanishing difference has order infinity.

For a uniform coefficient hypothesis, the simpler conclusion follows:

$$\boxed{\inf_{j \geq 0} w(s+j) \leq \operatorname{ord}_p(\Delta_1^s f(n)).}$$

For each finite $L \leq B_p(w; n, s)$, a separate theorem yields $p^L \mid \Delta_1^s f(n)$ inside $\mathbb{Z}_p$. Another specialization handles arbitrary finite interpolants with infinite weights beyond their degree.

Principal targets.

`PadicBounds.nat_le_order_iff`, `PadicBounds.weighted_difference_bound`

`PadicBounds.weighted_difference_divisibility`

`PadicBounds.coefficient_tail_bound`, `PadicBounds.interpolant_weighted_bound`

3. Unit modes and residue-field reduction

Theorem 7.4; manuscript page 18

For u in any commutative ring, and every natural step h, iteration proves

$$\boxed{\Delta_h^s(u^n) = u^n(u^h - 1)^s.}$$

The proof unfolds the actual forward-difference operator and uses the power multiplication law. No valuation premise or geometric-difference identity is assumed. More generally, a proved divisibility $p^k \mid u^h - 1$ implies

$$p^{ks} \mid \Delta_h^s(u^n).$$

Why a p-adic unit supplies one factor of p

Now let $u \in \mathbb{Z}_p^\times$. The actual ring homomorphism

$$\pi : \mathbb{Z}_p \longrightarrow \mathbb{F}_p$$

sends u to a nonzero unit. Fermat's finite-field theorem therefore gives

$$\pi(u)^{p-1} = 1, \qquad \pi(u^{p-1} - 1) = 0.$$

Using the proved kernel identity $\ker \pi = p\mathbb{Z}_p$, the source derives

$$\boxed{p \mid u^{p-1} - 1 \quad \text{inside } \mathbb{Z}_p.}$$

The conclusion is not supplied as an abstract certificate. It follows from the unit hypothesis and the standard residue-field construction.

Combining it with the geometric identity establishes

$$\boxed{p^s \mid \Delta_{p-1}^s(u^n), \qquad s \le \operatorname{ord}_p(\Delta_{p-1}^s(u^n)).}$$

These statements hold for every prime, including $p = 2$ with step one. Zero differences and order zero are included. Every multiple of the Fermat step satisfies the same lower bound.

Finite integral linear combinations

For any finite index set S, coefficients $a_i \in \mathbb{Z}_p$ and units $u_i \in \mathbb{Z}_p^\times$, linearity of the actual iterated operator gives

$$\Delta_h^s\left(\sum_{i\in S} a_i u_i^n\right) = \sum_{i\in S} a_i u_i^n (u_i^h - 1)^s.$$

At $h = p - 1$, the result is divisible by p^s. The empty sum is permitted. This does not prove that the general congruential Euler packets of Theorem 7.5 admit such a representation.

Principal targets.

`UnitModes.geometric_difference`, `UnitModes.unit_fermat_divisibility`

`UnitModes.unit_mode_valuation_gain`

`UnitModes.finite_unit_modes_valuation_gain`

4. Exact examples and adversarial checks

The examples use the same definitions as the general results

A polynomial anchor

The actual difference table for $f(n) = n^2$ yields

$$(c_0, c_1, c_2, c_3) = (0, 1, 2, 0).$$

The finite interpolant therefore reconstructs $I_2(c)(7) = 49$. This checks the sign and indexing conventions, not just the names of the defined objects.

A sharp binomial-weight improvement

Let

$$F(n) = 81\binom{n}{1} + 3\binom{n}{2} + 3\binom{n}{3} + 81\binom{n}{4}.$$

At $n = 3$ and $s = 1$, the four coefficient valuations are $(4, 1, 1, 4)$, while the binomial valuations are $(0, 1, 1, 0)$. Thus the refined bound is

$$B_3(w; 3, 1) = \min(4, 2, 2, 4) = 2,$$

stronger than the coefficient-only lower bound one. The source checks both the minimum and the actual result:

$$\boxed{\Delta F(3) = 180, \qquad \operatorname{ord}_3(180) = 2.}$$

The valuation equality is proved in the actual completed ring $\mathbb{Z}_3$, with an independent exact integer divisibility check as well.

A sharp unit-mode gain

For $p = 5$ and $u = 2$,

$$\Delta_4^2(u^n)\big|_{n=0} = (2^4 - 1)^2 = 225.$$

The source proves $5^2 \mid 225$ and $5^3 \nmid 225$. This example attains the guaranteed gain. No additional uniform power of p is claimed.

What fails when hypotheses or conventions are dropped

The nonunit $u = 3$ at $p = 3$ gives $\Delta_2(u^n)|_{n=0} = 8$, with no factor of three. A constant unit has every positive-order difference zero, hence infinite extended valuation. The finite-sample counterexample on page 2 distinguishes a true interpolant identity from an unsupported claim about unknown continuation.

Source. The module `MahlerExamples` contains these exact certificates. Principal target names are listed in the fourth-block ledger.

5. Verification record and remaining work

Evidence is retained separately from conclusions

E.5 Finite units and CRT

Scope and result

This supplement develops Proposition 6.2, Corollary 8.3 and Theorem 10.1 in the scopes stated below.

Twenty of 24 named items proved in the manuscript now have formal coverage. Four remain pending. Three classical inputs and two conjectures are tracked separately. The cumulative project contains **333 named theorem declarations and 77 explicit definitions across 28 modules**.

New module	Scope	Theorems	Defs.
`UnitTriangular`	Integral solver	5	1
`WindowIndependence`	8.3	11	3
`FiniteUnits`	6.2, prime powers	13	4
`DyadicUnits`	6.2, sign branch	10	4
`UnitRefinement`	Finite compatibility	4	2
`CRTPackets`	10.1	5	1
`FiniteArithmeticExamples`	Exact controls	13	2
Total added	Three items	61	17

The 61 new theorems include supporting lemmas and examples, not 61 independent discoveries. Pinned Mathlib supplies finite-field, unit-group, ideal-power, p-adic residue and Chinese-remainder results. The new modules connect these to the actual packet coefficients, explicit finite lifts and existing selectors.

The specification matters

Rational coefficients are not reduced through an imaginary map $\mathbb{Q} \to \mathbb{F}_p$. The proof first constructs their integral lifts in $\mathbb{Z}_p$ and establishes equality in $\mathbb{Q}_p$. Root representatives are actual units at the requested precision, not unchanged residues that need not satisfy the root equation modulo p^ℓ.

1. Integral reduction of the original coefficients

Corollary 8.3; manuscript pages 19–20

Fix a prime p, an integer $m \geq 1$ and a window address $0 \leq j < p$. The earlier project defines the rational packet coefficients by the actual recurrence

$$\sum_{q=0}^{n} \binom{mpn+j}{mpq} C_{mp,j}(q) = \delta_{n0}.$$

The previous Lucas theorem proves

$$\binom{mpn+j}{mpq} \equiv \binom{mn}{mq} \pmod{p}, \qquad \binom{mpn+j}{mpn} \equiv 1 \pmod{p}.$$

The last congruence makes each diagonal entry a unit in the actual completed ring $\mathbb{Z}_p$.

First construct the integral solution

For a ring R, a triangular row system T, right-hand side b, and proved unit pivots $d_n = T_{nn}$, define

$$s_n = d_n^{-1} \left(b_n - \sum_{q<n} T_{nq} s_q \right).$$

The new ring-level solver uses well-founded recursion. Its row identities and unique-solution property are proved. The source also proves compatibility of this solver with ring homomorphisms into fields.

Specializing to $R = \mathbb{Z}_p$ constructs $c_{m,j}^{(p)}(n)$. Mapping to $\mathbb{Q}_p$ and comparing with the original rational forward substitution gives

$$\boxed{c_{m,j}^{(p)}(n) = C_{mp,j}(n) \quad \text{in } \mathbb{Q}_p.}$$

Thus every original standard-window coefficient is p-integral, for every row. Integrality is a conclusion, not an additional hypothesis on the coefficients.

Then apply the residue map

Independently define $e^{(m,p)}$ over $\mathbb{F}_p$ by the unit-diagonal system

$$\sum_{q=0}^{n} \binom{mn}{mq} e_q^{(m,p)} = \delta_{n0}.$$

Its recurrence and uniqueness are checked. Applying the actual homomorphism $\pi : \mathbb{Z}_p \to \mathbb{F}_p$ to the integral row, and then Lucas reduction, proves

$$\boxed{\pi(c_{m,j}^{(p)}(n)) = e_n^{(m,p)}.}$$

For $j, k < p$, a separate ideal statement gives

$$p \mid c_{m,j}^{(p)}(n) - c_{m,k}^{(p)}(n) \quad \text{inside } \mathbb{Z}_p.$$

The formal solver also covers $m = 0$. The manuscript uses $m \geq 1$. No global finite-difference theorem or period assertion enters this proof.

Principal targets.

`WindowIndependence.coeffPadic_coe,.standard_window_integral`

`WindowIndependence.residue_independence,.reduced_unique`

`WindowIndependence.two_window_addresses_congruent`

2. Explicit finite prime-power decomposition

Proposition 6.2; odd-prime branch

Let p be prime, $\ell \geq 1$, and

$$a \in (\mathbb{Z}/p^{\ell}\mathbb{Z})^{\times}.$$

Define actual units in the same ring by

$$\boxed{\omega_{\ell}(a) = a^{p^{\ell-1}}, \qquad \langle a \rangle_{\ell} = a\,\omega_{\ell}(a)^{-1}.}$$

The Lean source writes $\ell = L + 1$, so zero precision is excluded without truncated natural-number subtraction.

The root equation and residue are derived

Euler's unit-group exponent gives

$$\omega_{\ell}(a)^{p-1} = a^{p^{\ell-1}(p-1)} = a^{\varphi(p^{\ell})} = 1.$$

Reduction to the actual field $\mathbb{F}_p$ commutes with powers. Its Frobenius identity gives

$$\overline{\omega_{\ell}(a)} = \bar{a}^{p^{\ell-1}} = \bar{a}.$$

Consequently,

$$\boxed{a = \omega_{\ell}(a)\langle a \rangle_{\ell}, \quad \omega_{\ell}(a)^{p-1} = 1, \quad \overline{\langle a \rangle_{\ell}} = 1.}$$

The manuscript's finite existence conclusion is obtained directly. The proof does not assume a Hensel lift or an already supplied root representative.

Uniqueness requires a lifting argument

The source proves the actual kernel statement

$$\bar{x} = 0 \quad \Longleftrightarrow \quad p \mid x \quad \text{in } \mathbb{Z}/p^{\ell}\mathbb{Z}.$$

For $x \equiv y \pmod{p}$, the checked ideal-power lifting theorem gives

$$x^{p^{\ell-1}} - y^{p^{\ell-1}} \in (p)^{\ell} = (0).$$

If $w^{p-1} = 1$, then $w^{p} = w$ and hence $w^{p^{k}} = w$ for every k. Therefore two roots with the same residue are equal. The decomposition is unique whenever its second factor reduces to one. Multiplicativity of both factors is also proved.

For $p = 3, \ell = 2, a = 2$, the checked values are

$$\omega_2(2) = 8, \qquad \langle 2 \rangle_2 = 7, \qquad 8^2 = 1, \quad 8 \cdot 7 = 2 \quad \text{in } \mathbb{Z}/9\mathbb{Z}.$$

The unlifted representative 2 is not a square root of one modulo nine.

Principal targets.

`FiniteUnits.finite_unit_decomposition,.power_lift_congruence`

`FiniteUnits.omega_unique,.decomposition_unique`

`FiniteUnits.omega_mul,.principal_mul`

3. Precision compatibility and the dyadic sign

Finite refinements and Proposition 6.2's dyadic branch

For $1 \le k \le \ell$, let

$$\rho_{\ell,k} : \mathbb{Z}/p^{\ell}\mathbb{Z} \longrightarrow \mathbb{Z}/p^{k}\mathbb{Z}$$

be the actual reduction homomorphism. It induces a map on units. The explicit construction satisfies

$$\boxed{\rho_{\ell,k}(\omega_\ell(a)) = \omega_k(\rho_{\ell,k}(a)),}$$

$$\boxed{\rho_{\ell,k}(\langle a\rangle_\ell) = \langle \rho_{\ell,k}(a)\rangle_k.}$$

The proof uses the exponent factorization $p^{\ell-1} = p^{k-1}p^{\ell-k}$ and the already proved Frobenius invariance of a lifted root. Principal factors also satisfy

$$\langle a\rangle_\ell^{\,p^{\ell-1}} = 1.$$

At precision one the principal factor is exactly one. These are coherent finite certificates. No infinite inverse limit or topological unit-group equivalence is constructed here.

Why the mod-four branch is separate

Now let $\ell \ge 2$ and $a \in (\mathbb{Z}/2^{\ell}\mathbb{Z})^{\times}$. The actual units of $\mathbb{Z}/4\mathbb{Z}$ are exhausted by 1 and -1. Define

$$\varepsilon_\ell(a) = \begin{cases} 1, & a \equiv 1 \pmod 4, \\ -1, & a \equiv 3 \pmod 4, \end{cases} \qquad \langle a\rangle_{2,\ell} = \varepsilon_\ell(a)a.$$

The source uses $\ell = L + 2$. It proves

$$\boxed{a = \varepsilon_\ell(a)\langle a\rangle_{2,\ell}, \quad \varepsilon_\ell(a) \in \{1,-1\}, \quad \langle a\rangle_{2,\ell} \equiv 1 \pmod 4.}$$

The sign is uniquely determined by the residue modulo four, and cancellation gives uniqueness of the principal factor.

The checked example $a = 3$ modulo eight has sign $7 = -1$ and principal factor 5:

$$7 \cdot 5 = 3 \pmod 8, \qquad 5 = 1 \pmod 4.$$

The code separately proves $1 = -1$ in $\mathbb{Z}/2\mathbb{Z}$ but $1 \ne -1$ in $\mathbb{Z}/4\mathbb{Z}$. Modulo two alone cannot identify the sign. The generic prime-power construction at $p = 2$ only asks for a principal residue modulo two and is not silently substituted for this stronger mod-four factorization.

Principal targets.

`UnitRefinement.omega_compatible,.principal_compatible,.principal_power`

`DyadicUnits.dyadic_unit_decomposition,.decomposition_unique`

4. CRT addresses and the actual packet selectors

Theorem 10.1; manuscript page 22

Let $a \geq 0$, let M be odd, and put $N = 2^a M$. Coprimality is derived from oddness:

$$\gcd(2^a, M) = 1.$$

The construction uses the actual Chinese-remainder ring equivalence

$$\boxed{\mathbb{Z}/N\mathbb{Z} \;\simeq\; \mathbb{Z}/2^a\mathbb{Z} \times \mathbb{Z}/M\mathbb{Z}.}$$

Its bijectivity and component formula are checked. On a natural representative r, the components are precisely its reductions modulo the two factors, not arbitrary coordinate labels.

The algebraic bridge to projectors

For natural indices n, r, the source proves

$$n \equiv r \pmod{2^a M} \quad\Longleftrightarrow\quad \big(n \equiv r \pmod{2^a} \text{ and } n \equiv r \pmod{M}\big).$$

For the original coefficient selector

$$(P_{N,r} f)(n) = \begin{cases} f(n), & n \bmod N = r, \\ 0, & \text{otherwise}, \end{cases}$$

take the canonical label $0 \leq r < N$. The logical equivalence yields the actual equality of sequence operators

$$\boxed{P_{N,r} f = P_{2^a,\, r \bmod 2^a}\big(P_{M,\, r \bmod M} f\big).}$$

This statement holds for coefficients in any type with zero. A separate theorem lifts the equality coefficientwise to the existing formal power-series selectors over any semiring.

The cases $a = 0$ and $M = 1$ are included. The corresponding one-element residue ring is not excluded. Oddness of M already implies positivity.

What is not asserted

CRT factors addresses, not the visibility weights or valuation filtrations associated with those addresses. Nothing in this proof gives an equal-weight decomposition at every prime or an additional analytic convergence theorem.

Coprimality cannot be discarded. The distinct classes 0 and 2 modulo four both map to $(0, 0)$ when both components are reduced modulo two. The source checks this counterexample. It also checks a composite-modulus failure of the Fermat exponent and a coefficient outside the standard window that is not 3-adically integral.

Principal targets.

`CRTPackets.address_bijective,.address_components,.residue_equivalence`

`CRTPackets.packet_factorization,.series_packet_factorization`

E.6 Recurrence packets

Scope and result

This supplement develops Theorem 9.1 and Corollary 9.2.

Twenty-two of 24 named items proved in the manuscript now have formal coverage. Two remain pending. Three classical inputs and two conjectures are tracked separately. The cumulative corpus has **382 named theorem declarations and 84 explicit definitions across 31 modules**.

New module	Scope	Theorems	Defs.
`CompanionPackets`	General jump and recurrence	17	2
`FibonacciPackets`	Actual Fibonacci specialization	18	4
`RecurrenceExamples`	Examples and boundary cases	14	1
Total added	Two named items	49	7

These counts include supporting lemmas and examples, not 49 independent mathematical discoveries. The principal statements are universal over their explicit inputs, not finite numerical observations.

The main mathematical connection

A sampled state $u_k = v_{Nk+j}$ evolves by the actual jump matrix C^N. Its recurrence is derived from the characteristic polynomial of that matrix. No scalar recurrence or spectral decomposition is supplied as a substitute for the desired conclusion.

For Fibonacci numbers, the state is independently defined from Mathlib's actual sequence. The Lucas sequence is independently defined by its initial values and recurrence. The trace and determinant of C^N are then proved to supply the sampled recurrence coefficients.

Important qualification

A d-dimensional state system supplies a recurrence of **order at most** d. Its minimal recurrence need not have order d. The original statement is read as the characteristic-polynomial recurrence, not a minimal-order claim. The formal equations also handle singular matrices, zero-dimensional spaces and zero step.

1. Actual matrix orbits and arithmetic jumps

Theorem 9.1; manuscript page 22

Let R be a commutative ring, I a finite coordinate type and $d = |I|$. For $C \in M_I(R)$ and $v_0 \in R^I$, define

$$\operatorname{orbit}(C, v_0, n) = C^n v_0, \qquad u_k = \operatorname{packet}(C, v_0, N, j, k) = C^{Nk+j} v_0.$$

The implementation uses actual matrix powers and matrix–vector multiplication. A separately supplied sequence v with

$$v_{n+1} = Cv_n$$

is proved by induction to equal this power orbit. Thus the source-sequence formulation is connected to the construction.

The jump formula

Associativity and the laws for powers of one matrix give

$$\boxed{u_k = C^j (C^N)^k v_0 = (C^N)^k (C^j v_0).}$$

Only powers of the same matrix are interchanged. The argument does not assume that arbitrary matrices commute. In particular,

$$u_{k+1} = C^N u_k, \qquad u_{k+a} = (C^N)^a u_k.$$

The step matrix is C^N, not C. A checked example with the Fibonacci matrix and $N = 2$ rejects replacing the jump with a single original step.

Refinement of an arithmetic address

Taking another subsequence with step M and offset s yields

$$u_{Mk+s} = v_{(NM)k+(Ns+j)}.$$

The formal proof establishes this identity for the actual packet function by checking the index arithmetic. It is not inferred from a diagram or a naming convention.

Endpoints and interpretation

The algebraic identities allow $N = 0$, in which case $u_k = v_j$ is constant. Positive-modulus residue packets are the special case $N > 0$ and $0 \le j < N$. The formula also permits offsets beyond that standard range. No inverse matrix, diagonalization, eigenvalue, or characteristic-zero assumption is needed.

Principal targets.

`Companion.orbit_of_recurrence`

`Companion.companion_jump`, `Companion.source_companion_jump`

`Companion.packet_step`, `Companion.packet_advance`

`Companion.packet_refinement`, `Companion.packet_zero_step`

2. From the characteristic polynomial to a recurrence

Theorem 9.1, continued

Put $D = C^N$ and let

$$\chi_D(X) = \det(XI - D) = X^d + \sum_{a=0}^{d-1} c_a X^a$$

be its actual characteristic polynomial. The implementation reuses the pinned Mathlib Cayley–Hamilton theorem, expands polynomial evaluation into a finite sum and proves

$$\sum_{a=0}^{d} \chi_D[a] D^a = 0, \qquad \chi_D[d] = 1.$$

Here $\chi_D[a]$ denotes the coefficient at index a. The identities hold over any commutative coefficient ring. The zero-ring case is handled explicitly by the proof's subsingleton branch.

The complete vector recurrence

Apply the matrix identity to the actual state u_k. The already proved advance identity identifies each $D^a u_k$ with u_{k+a}. Consequently,

$$\sum_{a=0}^{d} \chi_D[a] u_{k+a} = 0, \qquad \boxed{u_{k+d} = -\sum_{a=0}^{d-1} c_a u_{k+a}.}$$

This is the desired recurrence, with its monic leading coefficient and all earlier coefficients supplied by χ_D. It is not an assumption on the packet sequence.

Every coordinate and every R-linear scalar observation $\lambda : R^I \to R$ satisfies the corresponding equation:

$$\lambda(u_{k+d}) = -\sum_{a=0}^{d-1} c_a \lambda(u_{k+a}).$$

A separate source-sequence theorem transports the recurrence back to the original v_{Nk+j} notation.

Order versus minimal order

The characteristic polynomial gives a recurrence of order at most d. For example, $C = I_2$ gives a degree-two characteristic polynomial, but every orbit is constant and obeys $u_{k+1} = u_k$. Both statements are checked. No minimal-polynomial classification is claimed.

For $d = 0$, the unique empty vector is zero and the formula remains valid. For singular matrices, Cayley–Hamilton still applies: invertibility is irrelevant. The nilpotent example $C = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ has $C^2 = 0$ and $\chi_C = X^2$.

Principal targets.

`Companion.charpoly_sum_powers`, `Companion.charpoly_coeff_dimension`

`Companion.packet_charpoly_annihilation`

`Companion.packet_charpoly_recurrence`

`Companion.packet_coordinate_recurrence`

`Companion.packet_observable_recurrence`

3. Fibonacci and Lucas coefficients are derived

Corollary 9.2; manuscript page 22

Define the actual integer matrix and state

$$C = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \qquad v_n = \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}, \qquad v_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The values F_n are Mathlib's `Nat.fib`, cast into $\mathbb{Z}$. Its basic recurrence proves $v_{n+1} = Cv_n$. The second coordinate of the packet state is therefore proved to be the intended F_{Nk+j}, not a newly assumed sequence.

An independent Lucas sequence

Define

$$L_0 = 2, \quad L_1 = 1, \quad L_{n+2} = L_n + L_{n+1}.$$

Direct finite matrix arithmetic proves $C^2 = C + I$. Multiplication by C^n gives $C^{n+2} = C^n + C^{n+1}$. Taking traces shows that $\operatorname{tr}(C^n)$ satisfies the same recurrence and initial values as L_n. Induction yields

$$\boxed{\operatorname{tr}(C^N) = L_N.}$$

The implementation separately proves

$$\det C = -1, \qquad \det(C^N) = (-1)^N, \qquad L_N = 2F_{N+1} - F_N.$$

No Binet roots or diagonalizability argument is needed.

The sampled Fibonacci recurrence

The actual two-by-two characteristic-polynomial formula gives

$$\chi_{C^N}(X) = X^2 - L_N X + (-1)^N.$$

Insert these coefficients into the general packet recurrence and take the second coordinate:

$$\boxed{F_{N(k+2)+j} = L_N F_{N(k+1)+j} - (-1)^N F_{Nk+j}.}$$

The full two-coordinate state satisfies the same equation. All subtraction is in $\mathbb{Z}$, not truncated natural-number subtraction. The result holds for all natural N, j, k, including zero step and zero initial values.

Principal targets.

`FibonacciPackets.state_is_orbit`, `FibonacciPackets.packet_is_fibonacci`

`FibonacciPackets.trace_power_is_lucas`, `FibonacciPackets.det_power`

`FibonacciPackets.jump_charpoly`

`FibonacciPackets.fibonacci_packet_recurrence`

4. Exact examples and meaningful boundary checks

Example 9.3 and supporting results

The independently defined Lucas recurrence gives

$$(L_0, L_1, L_2, L_3, L_4, L_5) = (2, 1, 3, 4, 7, 11).$$

For $N = 3$, the general theorem therefore specializes to

$$u_{k+2} = 4u_{k+1} + u_k.$$

The implementation proves this for every offset j and every k, not only the displayed sample. At $j = 2$, direct evaluation of the actual Fibonacci sequence gives

$$(F_2, F_5, F_8, F_{11}, F_{14}, F_{17}) = (1, 5, 21, 89, 377, 1597).$$

This is the exact manuscript example.

The parity sign cannot be removed

For $N = 2$, the recurrence is

$$u_{k+2} = 3u_{k+1} - u_k.$$

At $j = 1, k = 0$, the correct equation is $5 = 3 \cdot 2 - 1$. Replacing the minus sign with a plus would assert $5 = 7$. The source proves the inequality, and a separate invalid-proof control attempts and rejects the false equation.

At $N = 1$ the ordinary Fibonacci recurrence is recovered. At $N = 0$, $L_0 = 2$ and the packet is constant, so the same formula becomes $u = 2u - u$.

Uniqueness from the first two packet values

For a fixed N, j, any integer sequence satisfying the proved sampled recurrence and starting at F_j, F_{N+j} agrees with F_{Nk+j} for all k. This is proved by two-step induction. It is a consequence of the established coefficients and recurrence, not a premise used to identify Fibonacci values initially.

Independent finite mirrors

A pure-Python exact-arithmetic implementation computes characteristic polynomials by the permutation determinant and states by repeated matrix application. It checks 280 matrix systems over integers, rationals and several modular rings, including composite moduli and the zero ring. Separately it computes sampled Fibonacci sequences recursively and compares them with direct Fibonacci evaluation. These are useful regression checks, not Lean extraction or universal proofs.

Principal targets.

`RecurrenceExamples.manuscript_sequence`

`RecurrenceExamples.manuscript_step_three,RecurrenceExamples.step_two`

`FibonacciPackets.fibonacci_packet_unique`

`RecurrenceExamples.even_step_wrong_sign`

`RecurrenceExamples.wrong_jump_matrix`

5. Verification, provenance and remaining work

Cumulative checks and exact scope

`aeb72f1e9981b6d35b426b16dca7759be3e7ba06.`

E.7 Periods and difference bounds

Scope and result

This block completes Corollary 8.5 of the unchanged status-corrected manuscript by an independent finite-field argument. It does **not** assume the still-pending global finite-difference theorem, item 7.5.

Twenty-three of 24 named items proved in the manuscript now have formal coverage. The cumulative corpus contains **439 named theorem declarations and 94 explicit definitions across 38 modules**. Three classical inputs and two conjectures remain separately tracked. All 31 upstream proof modules and the six earlier supplements remain unchanged.

New module	Theorems	Definitions
`HurwitzConvolution`	13	3
`HurwitzInverse`	6	1
`StandardWindowPeriod`	8	1
`DifferenceFiltration`	4	0
`GlobalEulerBoundary`	7	1
`DworkCriterion`	7	4
`PeriodExamples`	12	0
Total added	57	10

The counts include supporting lemmas, conditional implications and exact examples. They are not counts of independent discoveries. The conditional implications do not promote their antecedents to proved facts.

The independent route

In characteristic p, both the first and pth coefficient shifts obey a Leibniz rule for Hurwitz convolution. A section supported at multiples of m, with $m \mid p-1$, has equal first and pth shifts. Its constructed reciprocal inherits this equality. Identifying that reciprocal with the existing reduced Euler sequence gives its period on the positive-index tail.

What remains

Theorem 7.5 asks for powers as high as $p^{\min(k,mn)}$ and covers nonstandard residue classes. A period modulo p does not establish that stronger result. This proves a genuine standard-window mod-p fragment and precise conditional interfaces for the remaining argument, while keeping item 7.5 pending.

1. Coefficient products and reciprocal differentiation

A characteristic-p argument without factorial division

For sequences $f, g : \mathbb{N} \to K$, where K is a field, define

$$(f \star g)(d) = \sum_{i=0}^{d} \binom{d}{i} f(i) g(d-i), \qquad (U^a f)(d) = f(d+a).$$

These are Hurwitz coefficients. The notation does not divide by $d!$ in $\mathbb{F}_p$. In particular, the implementation is not ordinary power-series multiplication with undefined inverse factorials in positive characteristic.

The identity sequence is $\mathbf{1}(d) = \delta_{d0}$. The new source proves the actual higher product rule

$$U^a(f \star g) = \sum_{i=0}^{a} \binom{a}{i} (U^i f) \star (U^{a-i} g).$$

The argument starts with Pascal's identity for the first shift, then inducts on a. All sums are finite at each coefficient.

The prime-shift cancellation

For prime p and characteristic p, every interior binomial coefficient in row p vanishes. Hence

$$U^p(f \star g) = (U^p f) \star g + f \star (U^p g).$$

The first-shift rule has exactly the same form. Both are checked for the literal coefficient product.

Suppose $g(0) = 1$. The source constructs f by the full triangular recurrence

$$f(d) = \delta_{d0} - \sum_{i<d} \binom{d}{i} f(i) g(d-i)$$

and proves $f \star g = \mathbf{1}$. It also proves cancellation: if $a \star g = 0$, then $a = 0$, by induction on the first nonzero coefficient.

If $U^p g = Ug$, differentiate the inverse equation with both operators and subtract. The terms involving $f \star (Ug)$ cancel, leaving

$$(U^p f - Uf) \star g = 0.$$

The proved cancellation law yields

$$\boxed{U^p f = Uf.}$$

The section identity is an explicit premise of this general reciprocal lemma. The next page proves that premise for the specific section used by the Euler sequence.

Principal targets.

`Hurwitz.shift_convolution`, `Hurwitz.prime_shift_convolution`

`Hurwitz.convolution_left_eq_zero`

`HurwitzInverse.inverse_convolution`

`HurwitzInverse.inverse_prime_difference`

2. Identification with the reduced Euler sequence

Corollary 8.5; manuscript page 21

Work over $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$. For positive $m \mid p-1$, take the literal section

$$g_m(d) = \begin{cases} 1, & m \mid d, \\ 0, & m \nmid d. \end{cases}$$

It satisfies $g_m(0) = 1$. Since $p \equiv 1 \pmod{m}$, the conditions $m \mid d + p$ and $m \mid d + 1$ are equivalent. The source therefore proves $U^p g_m = U g_m$.

Let f_m be the reciprocal constructed on the previous page. A separate strong-induction proof establishes its actual support:

$$m \nmid d \quad \Longrightarrow \quad f_m(d) = 0.$$

This support is not included as an assumption of the construction.

The same residual sequence, not a renamed substitute

Evaluate $f_m \star g_m = \mathbf{1}$ at $d = mn$. Reindex the surviving summation indices by $i = mq$ using the proved support. The result is

$$\sum_{q=0}^{n} \binom{mn}{mq} f_m(mq) = \delta_{n0}.$$

The fifth block independently constructed the unique sequence $e_n^{(m,p)}$ satisfying this triangular system. Applying that uniqueness theorem gives

$$\boxed{f_m(mn) = e_n^{(m,p)}.}$$

Thus the reciprocal really is connected to the manuscript's Lucas-reduced inverse.

The positive-tail period

Write $h = (p-1)/m$. For $n \geq 1$, evaluate $U^p f_m = U f_m$ at the nonnegative degree $mn - 1$:

$$f_m(mn + p - 1) = f_m(mn).$$

Since $mn + p - 1 = m(n + h)$, the identified sequence satisfies

$$\boxed{e_{n+h}^{(m,p)} = e_n^{(m,p)} \quad (n \geq 1).}$$

Induction gives $e_{n+uh}^{(m,p)} = e_n^{(m,p)}$ for every $u \geq 0$. This proves a period, not necessarily the least positive period. The proof also includes $p = 2$, where $m = 1$ and $h = 1$.

Principal targets.

`WindowPeriod.section_prime_shift`

`WindowPeriod.inverse_off_support`, `WindowPeriod.inverse_on_support`

`WindowPeriod.uniform_standard_window_period`

`WindowPeriod.multiple_period`

3. Actual packet congruences and sharp boundaries

Consequences for the standard window; not the full Theorem 7.5

The fifth block constructed integral lifts $\widehat{C}_{mp,j}(n) \in \mathbb{Z}_p$ of the original rational coefficients when $0 \leq j < p$, and proved that their residues equal $e_n^{(m,p)}$.

Applying the actual residue map and its kernel ideal to the new period gives

$$\boxed{p \mid \widehat{C}_{mp,j}(n+uh) - \widehat{C}_{mp,j}(n) \quad \text{in } \mathbb{Z}_p}$$

for $n \geq 1$ and $u \geq 0$. Every higher positive difference is a difference of terms already divisible by p. Consequently,

$$p \mid \Delta_{uh}^{k} \widehat{C}_{mp,j}(n) \qquad (k \geq 1,\ n \geq 1).$$

A ring-homomorphism lemma identifies the image of these differences in $\mathbb{Q}_p$ with the differences of the original rational coefficient function. This is not vacuous field divisibility. The valuation version uses order infinity at zero.

Three checked boundaries

The initial row. For $(p, m) = (3, 2)$, the same reduced recurrence gives $e_0 = 1$ and $e_1 = -1 = 2$ in $\mathbb{F}_3$. Its proposed period is one, so the positive-tail result cannot in general include row zero.

The divisor hypothesis. At $(p, m) = (5, 3)$, where $3 \nmid 4$, the independently computed reduced row is

$$(e_0, e_1, e_2, e_3) = (1, 4, 4, 2).$$

Replacing h by the integer quotient $\lfloor 4/3 \rfloor = 1$ does not give a period: $e_3 \neq e_2$.

No automatic extra power. The original rational triangular coefficients satisfy

$$C_{6,0}(1) = -1, \quad C_{6,0}(2) = 923, \quad v_3\big(C_{6,0}(2) - C_{6,0}(1)\big) = v_3(924) = 1.$$

A period modulo 3 therefore cannot simply be reported as a period modulo 9. This is consistent with the global theorem's proposed first-difference bound. It is not a counterexample to Theorem 7.5.

Principal targets.

`WindowPeriod.standard_window_period_divisibility`

`GlobalEulerBoundary.standard_window_all_orders_mod_p`

`GlobalEulerBoundary.lifted_difference_coe`

`PeriodExamples.period_can_fail_at_zero`

`PeriodExamples.first_difference_precision_is_one`

4. Exact interfaces for the remaining global argument

Conditional results are not counted as completion of item 7.5

Let p be prime and m a positive divisor of $p - 1$. For a sequence e in a commutative coefficient ring, define its actual packed Hurwitz sequence

$$X(d) = \begin{cases} e(d/N), & N \mid d, \\ 0, & N \nmid d, \end{cases} \qquad N = mp.$$

Let $L = p(p-1)$ and $\delta = U^{p+L} - U^p = U^{p^2} - U^p$. The source proves the operator dictionary

$$(\delta^s X)(d) = \Delta_L^s X(d + ps).$$

At the valid nonnegative coefficient degree $d = mpn - ps$, with $s \le mn$, this becomes

$$(\delta^s X)(mpn - ps) = \Delta_h^s e(n), \qquad h = (p-1)/m.$$

The packed sequence and difference operations are definitions. Their coefficient correspondence is proved before any divisibility assumption is introduced.

A sufficient condition, explicitly conditional

Define flatness by the actual coefficientwise divisibility assertion

$$\delta^s X(d) \in p^s R \qquad (s, d \ge 0).$$

The new theorem proves that this hypothesis implies

$$\boxed{p^{\min(k,mn)} \mid \Delta_h^k e(n).}$$

First, coefficient extraction proves the triangular region $k \le mn$. A separate induction on difference order proves that this region is equivalent to the complete saturated estimate. This implication is useful over $\mathbb{Z}_p$. Interpreting it over a field would not supply the intended integral content.

Normalization is also an explicit implication

Using the existing definition $E_{Nn}^{(N,j)} = j! C_{N,j}(n)$, the source proves that multiplication by $j!$ commutes with every iterated difference. If the unnormalized difference has precision A, the normalized difference has precision $A - v_p(j!)$. Precision is defined as

$$x = 0 \quad \text{or} \quad v_p(x) \ge A,$$

so zero and negative factorial-loss thresholds are handled correctly. For $j < p$, the loss is zero.

Not established here. The general unnormalized Euler family has not been proved to possess the required flat integral lift, especially across singular residue intervals. The retrieved 18-page supporting prose proof uses weighted Amice/Mahler bounds, singular ratios and nonsingular propagation. Those obligations remain, and the full arbitrary-multiple global estimate is not promoted by these interfaces.

5. Verification record and reproducibility boundary

Evidence accompanies the source, not only the summary

aeb72f1e9981b6d35b426b16dca7759be3e7ba06.

Nine dependency heads are compared with their recorded manifest pins. The compiled Mathlib cache is reused, not independently rebuilt. The source checker rejects admitted proofs and selected trust-bypass constructs. The transitive axiom audit is a separate check.

E.8 Standard-window higher differences

Scope and result

The all-residue Theorem 7.5 is treated separately. This argument proves its full saturated higher-power **estimate throughout the standard window**, rather than only a period modulo p. For every prime p, positive $m \mid p-1$, $0 \le j < p$, and natural n, k, u, put $h = (p-1)/m$. Then

$$\boxed{\Delta_{uh}^{k} C_{mp,j}(n) = 0 \quad \text{or} \quad v_p\Big(\Delta_{uh}^{k} C_{mp,j}(n)\Big) \ge \min\{k, mn\}.}$$

The unnormalized family $j!C_{mp,j}(n)$ satisfies the same bound. Both statements concern the original rational coefficient function, identified with its actual integral p-adic lift. They include $p = 2$, $k = 0$, $n = 0$, and the additional harmless case $u = 0$.

Coverage remains 23 of 24 named items proved in the manuscript. Three classical inputs and two conjectures remain separate. The new result covers every permissible residue when $m = 1$, but not the range $p \le j < mp$ when $m > 1$.

New module	Theorems	Definitions
`DworkAlgebra`	7	3
`DworkClosure`	20	2
`HurwitzDworkClosure`	15	2
`ResidueZeroDwork`	9	2
`DifferenceStepMultiples`	5	1
`EulerDefect`	4	2
`StandardWindowDwork`	14	1
`StandardWindowPrecision`	5	0
`DworkWindowExamples`	8	0
Total added	87	13

The cumulative corpus has **526 explicit theorem declarations and 107 explicit definitions in 47 modules**. Counts include support lemmas and examples, not independent discoveries. All 38 upstream proof modules and seven earlier supplements are preserved.

1. The characteristic-zero product rule

The normalized defect is not an ordinary derivation

Let A be a commutative algebra over a characteristic-zero field K, and let U be an actual K-derivation. The source defines

$$B = \frac{U^{p^2} - U^{p}}{p}, \qquad M_q(f,g) = \sum_{a=1}^{q-1} \frac{1}{p}\binom{q}{a}(U^a f)(U^{q-a} g), \quad q = p, p^2.$$

The coefficients of each M_q are proved integral: the middle entries of a prime-power binomial row are divisible by p. In Lean they are implemented as natural quotients, with exact division proved before the product identity is used. Induction on derivative order establishes the full Leibniz formula and hence

$$\boxed{B(fg) = B(f)g + fB(g) + M_{p^2}(f,g) - M_p(f,g).}$$

The middle terms are essential. At $p = 3$, $f = t$ and $g = t^2$, the constant coefficient of $B(fg)$ is -2, whereas that of $B(f)g + fB(g)$ is zero. This inequality is itself Lean-checked.

All orders, without a circular closure assumption

Let $I \subset A$ be a subring stable under U. Define

$$\mathrm{Bounded}_r(f) \iff B^s f \in I \ (0 \le s \le r), \qquad \mathrm{Flat}(f) \iff B^s f \in I \ (s \ge 0).$$

The definitions contain no product or inverse closure requirement. Since B commutes with U, derivatives of a bounded element remain bounded. Induction on r in the displayed product rule then proves closure under multiplication. The finite middle sums use the lower-order induction hypothesis, not the desired next-order conclusion.

For an actual inverse $fg = 1$, applying the product rule and multiplying by g gives

$$B(g) = -g^2 B(f) - gM_{p^2}(f,g) + gM_p(f,g).$$

If $g \in I$, induction on r proves boundedness of g from boundedness of f. At each successor stage the preceding stage supplies boundedness of g needed on the right. Passing to all finite orders proves reciprocal closure for an integral inverse.

Principal targets.

`DworkAlgebra.B_mul`

`DworkClosure.bounded_mul`, `DworkClosure.bounded_inverse`

`DworkClosure.flat_mul`, `DworkClosure.flat_inverse`

2. The actual integral Hurwitz lattice

Connecting differential expressions to coefficient divisibility

The specialization works inside $\mathbb{Q}_p[[t]]$ and uses factorial-scaled coefficients

$$a_f(d) = d![t^d]f, \qquad I_p = \{f \in \mathbb{Q}_p[[t]] : a_f(d) \in \mathbb{Z}_p \text{ for every } d\}.$$

The source constructs I_p as a subring. Actual exponential-series multiplication gives

$$a_{fg}(d) = \sum_{r=0}^{d} \binom{d}{r} a_f(r) a_g(d-r), \qquad a_{Uf}(d) = a_f(d+1).$$

These identities prove multiplication and derivative stability of the lattice. Membership does not require ordinary power-series coefficients to lie in $\mathbb{Z}_p$.

The reciprocal is integral

For $f \in I_p$ with constant coefficient one, the usual power-series reciprocal is used. Its inverse equation yields the exact coefficient recurrence

$$a_{f^{-1}}(d) = \delta_{d0} - \sum_{r=0}^{d-1} \binom{d}{r} a_{f^{-1}}(r) a_f(d-r).$$

Strong induction proves that all of these coefficients are integral. Combining this construction with the preceding all-order theorem proves flatness of the *actual reciprocal*. No flatness or integrality of that reciprocal is assumed in the normalized-series specialization.

The earlier coefficient criterion is recovered exactly

Define the literal coefficient defect

$$(\delta a)(d) = a(d + p^2) - a(d + p).$$

The new source proves, for every s, d,

$$a_{B^s f}(d) = p^{-s}(\delta^s a_f)(d).$$

For an integral coefficient x, it proves the bridge

$$\boxed{p^{-s}x \in \mathbb{Z}_p \quad \Longleftrightarrow \quad p^s \mid x \text{ in } \mathbb{Z}_p.}$$

Thus the differential definition is equivalent to the existing coefficient-level Dwork criterion. Division occurs in $\mathbb{Q}_p$, while the assertion after division is membership in $\mathbb{Z}_p$. It is not divisibility in the field. Zero coefficients satisfy every finite precision condition.

Principal targets.

`HurwitzDworkClosure.integral_inverse`

`HurwitzDworkClosure.flat_product`, `HurwitzDworkClosure.flat_reciprocal`

`HurwitzDworkClosure.decode_B_iterate`

`HurwitzDworkClosure.normalized_mem_iff`, `HurwitzDworkClosure.flat_encode_iff`

3. A nonsingular Euler-operator argument

The standard interval without an analytic power-series-in-the-index premise

On Hurwitz coefficients, the actual Euler operator plus c is

$$(\Lambda_c a)(d) = (d + c)a(d).$$

Direct induction proves its commutator with every defect iterate:

$$\delta^s \Lambda_c a = \Lambda_c \delta^s a + sp\big(pU^{p^2} - U^p\big)\delta^{s-1} a, \qquad s \geq 1.$$

Suppose a is integral and supported at degrees divisible by p, and $p \nmid c$. Every supported pivot $d + c$ is proved to be a unit of $\mathbb{Z}_p$. The defect preserves this support. Induction in the commutator identity therefore proves

$$\mathrm{Flat}(\Lambda_c a) \Longrightarrow \mathrm{Flat}(a).$$

This is a reflection theorem on the integral supported domain. It does not assume flatness of the proposed preimage, and does not assert invertibility at a singular pivot $p \mid c$.

Apply it to the explicit normalized sections

Set $N = mp$ and, for $0 \le j < p$, define integral Hurwitz coefficients

$$g_j(d) = \begin{cases} \binom{d+j}{d}^{-1}, & N \mid d, \\ 0, & N \nmid d. \end{cases}$$

The inverse binomial is constructed using the actual unit pivots already proved by Lucas reduction. Its image in $\mathbb{Q}_p$ is proved equal to the displayed reciprocal.

For $j = 0$, g_0 is the literal N-divisibility indicator. Because $N \mid p(p-1)$, the two shifts in δg_0 agree and $\delta g_0 = 0$. It is flat at every order. The actual binomial identity gives

$$\Lambda_{j+1} g_{j+1} = (j+1) g_j.$$

As $1 \le j + 1 < p$, the nonsingular reflection theorem inductively proves flatness of every g_j in the standard window. There is no uniform-convergence assumption in this argument.

Principal targets.

`EulerDefect.iterated_commutator`, `EulerDefect.flat_of_euler_flat`

`StandardWindowDwork.kernel_coe`, `StandardWindowDwork.kernel_euler_step`

`StandardWindowDwork.kernel_flat`

4. From these sections to the original Euler packets

Actual reciprocal, actual rational sequence, every order and step multiple

The factorial-ratio identity is proved coefficient by coefficient:

$$\sum_{d \ge 0} g_j(d) \frac{t^d}{d!} = j! \sum_{r \ge 0} \frac{t^{Nr}}{(Nr + j)!}.$$

The right side is precisely the normalized source kernel from the first block, not a replacement family. The earlier rational identity for its reciprocal is transported to $\mathbb{Q}_p[[t]]$. A separate encoding theorem identifies the reciprocal's coefficients with the already constructed $\mathbb{Z}_p$ lifts of the original $C_{N,j}(r)$.

All-order reciprocal closure therefore gives flatness of the actual packed Euler coefficient sequence. The earlier coefficient-extraction theorem now has a *proved antecedent* for every $j < p$, and yields

$$p^{\min(k, mn)} \mid \Delta_h^k C_{mp,j}(n) \quad \text{in } \mathbb{Z}_p, \qquad h = (p-1)/m.$$

A new finite telescoping argument proves the extension to arbitrary natural multiples of h. Its first-step identity is

$$f(n+uh) - f(n) = \sum_{a=0}^{u-1} \Delta_h f(n+ah).$$

Commutation of differences and induction on their order transfer any fixed ideal bound valid on a forward tail. The required precision $\min(k, mn)$ is nondecreasing as the starting row advances. Hence

$$\boxed{p^{\min(k,mn)} \mid \Delta_{uh}^{k} C_{mp,j}(n) \quad (0 \le j < p).}$$

The scalar-extension bridge identifies these differences with the original rational coefficient function. The rational theorem explicitly allows zero differences, avoiding the library's finite valuation convention at zero. Since $j!$ is a unit for $j < p$, both normalized and unnormalized versions have the same precision.

Scope. The proof includes all primes, all rows and orders, and all natural step multipliers. For $m = 1$, the standard window is the complete allowed residue range. For larger m, residues $p \le j < mp$ are not covered by this theorem. The earlier one-power period is not the source of the new higher-power estimate.

Principal targets.

`StandardWindowDwork.window_is_reciprocal`, `StandardWindowDwork.standard_window_flat`

`StandardWindowDwork.standard_window_global_bound`

`StandardWindowPrecision.normalized_window_precision`

`StandardWindowPrecision.unnormalized_window_precision`

`StandardWindowPrecision.modulus_p_complete_family`

5. Examples, verification and the remaining gap

The coverage count is unchanged; the unresolved theorem now has a stronger proved part

The original rational recurrence gives

$$C_{6,2}(1) = -\frac{1}{28}, \quad C_{6,2}(2) = \frac{425}{364}, \quad C_{6,2}(3) = -\frac{2039759}{2660}.$$

Consequently, the new source proves

$$\Delta_1^2 C_{6,2}(1) = -\frac{949959}{1235}, \qquad v_3\big(\Delta_1^2 C_{6,2}(1)\big) = 2.$$

This attains the bound $\min(2, 2 \cdot 1) = 2$. A claim of a third power is rejected. The separate cubic example proves that the normalized differential defect is not an ordinary derivation. Controls also reject an asserted unit pivot at $c = p$ and the prime-power binomial-divisibility formula at a composite base.

Recorded checks

The cumulative verifier rebuilds all 47 modules and audits all 633 explicit declarations: 526 theorems and 107 definitions. Warnings are errors, unexpected axiom dependencies are rejected, and 27 deliberately invalid proof controls are tested. The allowed transitive axioms remain `propext`, `Classical.choice` and `Quot.sound`. Exact results, compiler commands and source hashes are in the accompanying machine-readable reports.

What is still unproved

The all-residue Theorem 7.5 requires the additional range $p \le j < mp$. Its supporting prose proof treats singular ratios at $c = p, 2p, \dots, (m-1)p$ through weighted Amice estimates and a dual Mahler pairing, then propagates across the intervening indices. Those actual-family estimates and the full remaining propagation are not supplied by this part . A unit argument at $p \nmid c$ cannot be used at $p \mid c$.

The standard-window theorem is an infinite family. Its width gives no numerical proportion of the work needed for the all-residue theorem. The two native conjectures remain unproved.

E.9 Singular Amice Supplement

Scope and result

Let p be prime, let $m \mid p-1$, and put $N = mp$. For $1 \le a < m$, put $c = ap$ and define the original exponential section and ratio by

$$H_{N,c}(t) = \sum_{n \ge 0} \frac{t^{Nn+c}}{(Nn+c)!}, \qquad R_c(t) = \frac{tH_{N,c-1}(t)}{H_{N,c}(t)}.$$

The quotient is interpreted after removing its forced zero. This block proves, for the actual rational coefficients α_d,

$$R_{ap}(\log(1+T)) = \sum_{d \ge 0} \alpha_d T^d, \qquad \boxed{\alpha_d = 0 \text{ or } v_p(\alpha_d) \ge -\lfloor d/p \rfloor .}$$

Equivalently, every $p^{\lfloor d/p \rfloor}\alpha_d$ belongs to the actual completed ring $\mathbb{Z}_p$. The zero-safe alternative is explicit in the formal rational statement.

This is the weighted estimate of Proposition 4.7 in the supporting argument. It is **not yet the singular higher finite-difference theorem**. The full all-residue Paper IX Theorem 7.5 remains pending, and full manuscript coverage stays at **23 of 24**.

No hidden estimate. The final weighted-ratio theorem assumes primality, the modulus divisibility and the specified residue bounds. It does not assume weighted section membership, reciprocal closure, the ratio identity, a Mahler pairing or the desired higher-difference bound. Pages 2–5 describe the constructed objects and proved bridges.

1. Integral cyclic coefficients

Work first in an arbitrary commutative ring R. If $z^p = 1$, set

$$I = (p, z-1), \qquad P = (p).$$

Expanding $((z-1)+1)^p = 1$ and using divisibility of every interior binomial coefficient in prime row p gives

$$(z-1)^p \in PI.$$

Every other degree-p product of the two generators of I also belongs to PI. The formal ideal proof therefore establishes

$$\boxed{I^p \subseteq PI.}$$

For $q = p\lfloor q/p \rfloor + (q \bmod p)$, strong induction and the factorial valuation identity give

$$\boxed{I^q \subseteq P^{v_p(q!)}}.$$

In detail, write $b = \lfloor q/p \rfloor$. Then

$$I^q \subseteq P^b I^b \subseteq P^{b+v_p(b!)} = P^{v_p(q!)}.$$

No integral-domain hypothesis or division in a field is used here.

Now take the integral cyclic group ring

$$R_N = \mathbb{Z}[\mathbb{Z}/N\mathbb{Z}], \qquad x = [1], \qquad x^N = 1,$$

and assume $N \mid p(p-1)$. Thus $(x^{p-1})^p = 1$. Each genuine consecutive block

$$\prod_{i=0}^{p-1} \bigl(x - (pb+i)\bigr)$$

belongs to $(p, x^{p-1} - 1)$. This follows from the actual polynomial identity

$$\prod_{i \in \mathbb{F}_p} (X - i) = X^p - X$$

modulo p, not from a hypothesized valuation of the product.

For the falling factorial $(x)_d = \prod_{i=0}^{d-1}(x-i)$, grouping into consecutive blocks gives

$$(x)_d \in I^{\lfloor d/p \rfloor} \subseteq \bigl(p^{v_p(\lfloor d/p \rfloor !)}\bigr).$$

Every integral residue coordinate therefore satisfies

$$\boxed{p^{v_p(\lfloor d/p \rfloor !)} \mid [(r)](x)_d \quad \text{in } \mathbb{Z}.}$$

Dividing a coordinate by $d!$ is done only afterward, using $v_p(d!) = \lfloor d/p \rfloor + v_p(\lfloor d/p \rfloor !)$. This establishes the weighted denominator estimate before any field-valued divisibility could make it vacuous.

Modules. `CyclotomicFiltration`, `PrimeBlockPolynomial`, and `CyclicAmiceCoefficients`.

2. Connection to the actual section

An integral coordinate estimate becomes relevant only after connecting it to the manuscript's generating functions. For a residue $r \in \mathbb{Z}/N\mathbb{Z}$, define

$$H_r(t) = \sum_{d \geq 0} \mathbf{1}_{d \equiv r \ (N)} \frac{t^d}{d!}, \qquad F_r(T) = H_r(\log(1+T)).$$

The implementation uses the actual formal logarithm and substitution in $\mathbb{Q}[[T]]$. It proves

$$H_r' = H_{r-1}, \qquad (1+T)F_r' = F_{r-1},$$

and hence the coefficient recurrence

$$(d+1)[T^{d+1}]F_r = [T^d]F_{r-1} - d[T^d]F_r.$$

The initial values are $F_r(0) = \mathbf{1}_{r=0}$.

Independently, let $b_{d,r}$ be the residue-r coordinate of $(x)_d$ in R_N. Actual group-ring multiplication gives

$$b_{0,r} = \mathbf{1}_{r=0}, \qquad b_{d+1,r} = b_{d,r-1} - d\, b_{d,r}.$$

Induction comparing these two recurrences proves

$$\boxed{d![T^d]H_r(\log(1+T)) = b_{d,r}.}$$

Thus the cyclic coefficient is not an unrelated stand-in for a section coefficient. For the canonical label $0 \le c < N$, the formal section is also proved equal to the original `ExponentialSection.originalSection` used in the existing Euler project.

Weighted membership

The class used in the formalization is

$$\mathcal{W}_p = \left\{ F \in \mathbb{Q}_p[[T]] : p^{\lfloor d/p \rfloor}[T^d]F \in \mathbb{Z}_p \text{ for every } d \right\}.$$

The integral-coordinate estimate from page 2 now proves

$$\boxed{H_r(\log(1+T)) \in \mathcal{W}_p}$$

for every residue r, including the singular ones. Rational coefficients are mapped into $\mathbb{Q}_p$ by the actual coefficient ring homomorphism.

The same source package retains the earlier independent formal identity

$$d![T^d]\exp(x\log(1+T)) = (x)_d$$

over a commutative $\mathbb{Q}$-algebra. The section bridge above does not rely on selecting or averaging roots in a cyclotomic extension.

Principal targets.

`factorial_coeff_section`; `section_weighted_integral`;

`weighted_section`.

3. Weighted closure and the normalized denominator

Let $w(d) = \lfloor d/p \rfloor$. Since $w(i) + w(d-i) \le w(d)$, the coefficient formula for a product proves closure of $\mathcal{W}_p$ under multiplication. Additive closure is immediate. For $U \in \mathcal{W}_p$ with $U(0) = 1$, strong induction in the actual reciprocal coefficient recurrence proves

$$\boxed{U^{-1} \in \mathcal{W}_p.}$$

This inverse property is a theorem, not an additional field in a structure encoding weighted series.

Differentiation and the Euler field

The derivative coefficient is $(d+1)a_{d+1}$. If $p \nmid d+1$, the adjacent weights agree. If $p \mid d+1$, the integer factor $d+1$ compensates for the one-unit weight increase. Therefore ordinary differentiation preserves $\mathcal{W}_p$.

For the actual logarithm, its coefficient denominator is d, and the proved inequality

$$v_p(d) \le \lfloor d/p \rfloor \qquad (d > 0)$$

implies $\log(1+T) \in \mathcal{W}_p$. Consequently

$$L_T F = \log(1+T)(1+T)F'$$

also preserves $\mathcal{W}_p$.

Removing a singular forced zero

Put $c = ap < N$ and define the actual normalized kernel

$$K_c(t) = c! \sum_{n \ge 0} \frac{t^{Nn}}{(Nn+c)!}, \qquad K_c(0) = 1.$$

Let $\ell(T) = \log(1+T)/T$, defined by shifting logarithm coefficients, not by inverting T. Set

$$U(T) = \ell(T)^c K_c(\log(1+T)).$$

The formal proof establishes $U(0) = 1$ and the exact factorization

$$H_{N,c}(\log(1+T)) = \frac{T^c}{c!} U(T).$$

It follows coefficientwise that

$$[T^d]U = c![T^{d+c}]H_{N,c}(\log(1+T)).$$

Because $p^a \mid (ap)!$ and $\lfloor (d+ap)/p \rfloor = a + \lfloor d/p \rfloor$, the already proved section estimate gives $U \in \mathcal{W}_p$. Thus $U^{-1} \in \mathcal{W}_p$ as well. The factorial divisibility is derived from the pinned Legendre valuation theorem.

Boundary. The original positive-address section has zero constant coefficient. It is not the series to which unit inverse closure is applied.

4. The singular ratio estimate

For a positive address c, the rational ratio is independently constructed from the original kernels:

$$R_c(t) = c\,K_{c-1}(t)K_c(t)^{-1}.$$

The scalar factor c is necessary. The source proves the cleared identity

$$\boxed{R_c(t)H_{N,c}(t) = tH_{N,c-1}(t).}$$

This connects the constructed object to the original unnormalized ratio A_{c-1}/A_c. It does not assign the desired transformed formula as its definition.

After actual logarithmic substitution and the chain rule,

$$R_c(\log(1+T))H_{N,c}(\log(1+T)) = L_T H_{N,c}(\log(1+T)).$$

Insert the proved factorization $H_{N,c}(\log(1+T)) = T^c U/c!$. Leibniz differentiation and cancellation of that nonzero section give

$$\boxed{R_c(\log(1+T)) = c(1+T)\ell(T) + (L_T U)U^{-1}.}$$

All differentiation, substitution, nonzeroness and cancellation obligations are proved in the source.

For $c = ap$, the second summand belongs to $\mathcal{W}_p$ by the closure results on page 4. The first does too: shifting the logarithm coefficient loses at most one unit of the floor weight, so $p\ell(T) \in \mathcal{W}_p$, and the integer a supplies the remaining factor. Therefore

$$\boxed{R_{ap}(\log(1+T)) \in \mathcal{W}_p.}$$

The final rational coefficient theorem states the bound with an explicit zero alternative. No finite-valued default valuation at zero is used as the meaning of unlimited precision.

Checks that prevent a normalization mistake

The source proves that the actual ratio has constant coefficient c, not one. In the concrete case $N = 6, c = 3$, the universal theorem gives weighted membership and the constant coefficient is proved to be 3.

It also derives $[T^3]H_{3,0}(\log(1+T)) = 1/6$ from the actual differential recurrence and checks its valuation $v_3(1/6) = -1$. Thus weighted membership does not mean ordinary coefficient integrality. This value attains the allowed floor weight at degree three.

Principal target. `SingularRatioAmice.paper_singular_amice_estimate`. The theorem's assumptions contain only the declared arithmetic parameter conditions.

5. Verification and the remaining theorem

The verifier retains the previous 27 invalid-proof controls and adds two: incorrectly promoting a weighted coefficient to an integral one, and incorrectly normalizing the singular ratio constant to one. Missing imports, panics and timeouts do not qualify as successful negative controls. The preserved source checker rejects proof placeholders, added axioms and selected kernel-bypass constructs.

Independent finite regressions

The new exact-rational regression program is a separate implementation, not Lean extraction. Its checked scope is:

Finite check	Count
Integral cyclic-coordinate numerator tests	17,775
Corresponding section-weight tests	17,775
Direct logarithmic composition comparisons	24
Singular ratio families	13
Singular ratio coefficient comparisons	597
Weighted product, inverse and derivative cases	120

For $p = 3, N = 6, c = 3$, this independent computation obtains the degree-six ratio coefficient $1/10080$, with valuation -2, attaining the allowed weight. This particular coefficient calculation is recorded as a finite regression, not as a new Lean theorem. A violating modulus $p = 3, N = 4$ yields a section coefficient of valuation -4 at degree nine, showing that the modulus divisibility hypothesis cannot simply be discarded.

What has not been proved by this block

To obtain the singular higher-difference theorem, the weighted coefficient estimate must still be connected to the dual Mahler polynomial filtration and its integral pairing. The argument must then establish the required ratio flatness, propagate between singular indices, and assemble the original all-residue Euler family. None of these obligations is replaced by the finite regression tests.

Neither native conjecture is resolved by these arguments. Compiler evidence and source records are supplied separately.

E.10 All Residue Completion Supplement

Scope and final statement

Let p be a prime, let $m > 0$ divide $p-1$, put $N = mp$ and $h = (p-1)/m$, and let $0 \le j < N$. The original rational packet is defined by

$$\sum_{q=0}^{n} \binom{Nn+j}{Nq} C_{N,j}(q) = \delta_{n0}, \qquad E_j(n) = j!\, C_{N,j}(n).$$

For every $n, k, u \in \mathbb{N}$, the new formal result establishes

$$\boxed{E_j(n) \in \mathbb{Z}_p, \qquad \Delta_{uh}^k E_j(n) \in p^{\min(k,mn)}\mathbb{Z}_p.}$$

Here $\Delta_h f(n) = f(n+h) - f(n)$. The corresponding rational valuation statement is zero-safe:

$$\Delta_{uh}^k E_j(n) = 0 \quad \text{or} \quad v_p\big(\Delta_{uh}^k E_j(n)\big) \ge \min(k, mn).$$

For the normalized coefficients, the exact correction is retained:

$$\boxed{\Delta_{uh}^k C_{N,j}(n) = 0 \quad \text{or} \quad v_p\big(\Delta_{uh}^k C_{N,j}(n)\big) \ge \min(k, mn) - v_p(j!).}$$

The manuscript's odd-prime, positive-u theorem is a specialization. The formal statement also includes zero row, zero order, zero multiplier, and the $p = 2, m = 1$ subcase. This does not assert a different or broader external dyadic conjecture.

Principal declaration. `KomorebiPaperIX.Theorem75.global_finite_difference` states the integrality and both precision bounds together, for the original coefficient functions. Its hypotheses are primality, $m > 0$, $m \mid p-1$ and $j < mp$, not an assumed flatness or weightedness certificate.

1. Stability of the weighted differential ring

The construction uses the actual weighted subring

$$\mathcal{W}_p = \left\{ F(T) = \sum_{d \ge 0} a_d T^d : \; p^{\lfloor d/p \rfloor} a_d \in \mathbb{Z}_p \text{ for every } d \right\}.$$

The coefficient condition agrees with the singular-Amice construction. The formal source checks the complete formal-series statement below.

Put

$$\Theta = (1+T)\frac{d}{dT}, \qquad B_\Theta = \frac{\Theta^{p^2} - \Theta^p}{p}.$$

The coefficient formula

$$[T^d]\Theta F = (d+1)a_{d+1} + d a_d$$

proves $\Theta \mathcal{W}_p \subseteq \mathcal{W}_p$. If the floor weight increases between d and $d+1$, then $p \mid d+1$, which compensates for that increase.

The normalized defect is not assumed to be a derivation. Its proved product rule contains the two middle Leibniz sums:

$$B_\Theta(FG) = (B_\Theta F)G + F(B_\Theta G) + \mathcal{B}_p(F,G).$$

The coefficients of $\mathcal{B}_p$ are the actual integral quotients $\binom{p^2}{a}/p$ and $\binom{p}{a}/p$. Derivative stability and this correction formula show that the elements $F \in \mathcal{W}_p$ satisfying $B_\Theta F \in \mathcal{W}_p$ are closed under the finite ring operations.

The scalar input is proved for every natural b, including nonunits:

$$\boxed{p^2 \mid b^{p^2} - b^p \quad \text{in } \mathbb{Z}.}$$

Because $\Theta(1+T)^b = b(1+T)^b$, the binomial expansion of $T^p = ((1+T)-1)^p$ then proves stability of the divided generator T^p/p. For $d = pq + r$, $0 \le r < p$, every weighted monomial has the exact factorization

$$a_d T^d = (p^q a_d)\left(\frac{T^p}{p}\right)^q T^r.$$

The scalar $p^q a_d$ is integral by the actual weight condition. Thus all weighted monomials, and all finite weighted polynomials, satisfy the defect condition.

Finally, $[T^d]B_\Theta F$ depends only on coefficients through $d + p^2$. Replacing F by this finite prefix proves the result for the entire formal series, without an analytic convergence assumption:

$$\boxed{B_\Theta \mathcal{W}_p \subseteq \mathcal{W}_p, \qquad B_\Theta^s \mathcal{W}_p \subseteq \mathcal{W}_p \ (s \ge 0).}$$

Modules. `WeightedAmiceRing`, `FrobeniusSquare`, `ThetaWeighted`.

2. Actual substitution and singular higher differences

Let

$$\mathcal{E}(F)(t) = F(e^t - 1), \qquad D = \frac{d}{dt}, \qquad B_D = \frac{D^{p^2} - D^p}{p}.$$

All substitutions use Mathlib's actual formal exponential, logarithm and substitution maps. The chain rule gives

$$D\mathcal{E}(F) = \mathcal{E}(\Theta F), \qquad B_D^s \mathcal{E}(F) = \mathcal{E}(B_\Theta^s F).$$

The implementation also proves the inverse-substitution identity

$$\mathcal{E}\big(f(\log(1+T))\big) = f(t).$$

The relevant integral subring is the *Hurwitz* subring

$$\mathcal{H}(\mathbb{Z}_p) = \{f \in \mathbb{Q}_p[[t]] : \ n![t^n]f \in \mathbb{Z}_p \text{ for every } n\}.$$

It is not coefficientwise integrality of ordinary power-series coefficients. The precise connection is

$$n![t^n]\mathcal{E}(F) = [T^0]\Theta^n F.$$

Since every iterate of Θ preserves $\mathcal{W}_p$, the right side is integral. Therefore $\mathcal{E}(\mathcal{W}_p) \subseteq \mathcal{H}(\mathbb{Z}_p)$. Applying the same statement to every $B^s_\Theta F$ proves

$$\boxed{f(\log(1+T)) \in \mathcal{W}_p \quad \Longrightarrow \quad B^s_D f \in \mathcal{H}(\mathbb{Z}_p) \text{ for every } s.}$$

This is the proved transfer that was missing in the earlier attempt. No weighted Mahler pairing is assumed. The formal differential route supplies an alternative bridge.

Application to the actual singular ratios

The singular-ratio construction independently gives

$$H_{N,c}(t) = \sum_{n\geq 0} \frac{t^{Nn+c}}{(Nn+c)!}, \quad A_c(t) = t^{-c}H_{N,c}(t), \quad R_c = \frac{A_{c-1}}{A_c}.$$

The notation $t^{-c}H_{N,c}$ means removing its proved initial zero coefficients. For each $c = ap$, $1 \leq a < m$, that source proves $R_{ap}(\log(1+T)) \in \mathcal{W}_p$. The new transfer consequently proves flatness of the *same* rational ratio after scalar extension to $\mathbb{Q}_p$.

The actual ratio is also proved to be an expansion in t^N, so its coefficients vanish outside multiples of N. Constructed $\mathbb{Z}_p$ lifts and the established coefficient criterion now give, for its literal Hurwitz coefficients $r_a(n)$,

$$\boxed{\Delta^k_{uh} r_a(n) \in p^{\min(k,mn)}\mathbb{Z}_p.}$$

The separate rational precision theorem is obtained through the coefficient-lift equality. Divisibility in the field $\mathbb{Q}_p$ is never substituted for this assertion.

Modules. `WeightedDworkTransfer`, `SeriesDifferenceBridge`, `RatioSeriesSupport`.

3. Nonsingular propagation without an assumed limit

For consecutive original sections, the already defined ratios satisfy

$$R_c H_{N,c} = tH_{N,c-1}, \qquad H'_{N,c} = H_{N,c-1}.$$

Differentiating the first relation, using the relation for R_{c-1}, and cancelling a section whose leading coefficient is proved nonzero gives the actual Riccati equation

$$\boxed{tR_c' = R_c(1 + R_{c-1} - R_c), \qquad 2 \le c < N.}$$

It is not an equation imposed on a replacement ratio. Write

$$q = R_{c-1} - (c-1), \qquad y = R_c - c.$$

Both have zero constant coefficient. With $\Lambda_c = tD + c$, the equation becomes

$$\boxed{\Lambda_c y = cq + yq - y^2.}$$

When $p \nmid c$, define the actual inverse coefficientwise:

$$[t^d]\Lambda_c^{-1} f = \frac{[t^d]f}{d+c}.$$

On p-supported degrees, $d + c$ is a genuine p-adic unit. The proof establishes Hurwitz integrality of this solution, its support, both inverse identities, and preservation of full flatness using the earlier all-order Euler commutator theorem. The singular pivots $c = ap$ are never declared to be units.

Now define explicit iterations

$$y_0 = 0, \qquad y_{r+1} = \Lambda_c^{-1}(cq + y_r q - y_r^2).$$

If q is flat and p-supported, every iterate has those properties. The crucial link to the actual ratio is an exact finite-prefix theorem:

$$\boxed{[t^d]y_r = [t^d]y \qquad (d \le r).}$$

Indeed,

$$(cq + zq - z^2) - (cq + yq - y^2) = (z - y)(q - z - y).$$

The second factor has zero constant term. Multiplication therefore raises the established agreement prefix by one. Coefficientwise division by $d + c$ preserves that prefix.

For fixed s, d, the coefficient $d![t^d]B_D^s y$ depends only on the coefficients of y through $d + p^2 s$. Choosing the matching finite iterate proves its integrality. This establishes flatness of the actual y, and hence of R_c, without postulating flatness of the solution or invoking an unproved analytic fixed-point theorem.

Modules. `EulerSeriesInverse`, `RiccatiFlatPropagation`, `RatioRiccati`.

4. All-residue assembly and original coefficients

The standard-window source and inverse results supply R_1 and $F_{N,0}$. Induction over $1 \le c < N$ now has two proved cases. At $p \mid c$, the singular weighted estimate and the new substitution theorem give flatness. At $p \nmid c$, the preceding Riccati propagation applies. Thus every actual ratio R_c is flat.

Let $K_j(t)$ be the normalized section kernel with constant coefficient one. The constructed unnormalized Euler series is

$$F_{N,j}(t) = j!\, K_j(t)^{-1}.$$

The source/inverse identities and the factorial successor identity prove

$$F_{N,j+1} = F_{N,j} R_{j+1}, \qquad F_{N,j} = F_{N,0} \prod_{c=1}^{j} R_c.$$

The product formula is not an assembly hypothesis. Proved flatness of all factors and the original product-closure theorem establish flatness of the full family.

Coefficient extraction identifies, literally,

$$(Nn)! [t^{Nn}] F_{N,j} = j!\, C_{N,j}(n).$$

The support outside multiples of N is proved separately. The constructed integral lift therefore represents the original rational coefficient function. The existing saturation criterion and the proved step-multiple argument give the ideal bound at every order, row and multiplier. Multiplication by $j!$ commutes with finite differences, giving precisely the displayed normalization loss.

Exact examples outside the standard window

For $(p, m, j) = (3, 2, 3)$, the source derives

$$C_{6,3}(1) = -\frac{1}{84}, \quad C_{6,3}(2) = \frac{703}{5460}, \quad C_{6,3}(3) = -\frac{223199}{7980}.$$

Consequently,

$$\Delta_1^2\big(3! C_{6,3}\big)(1) = -\frac{1464768}{8645}, \qquad v_3\left(-\frac{1464768}{8645}\right) = 2.$$

This attains $\min(2, 2 \cdot 1) = 2$. At the regular address $j = 4$ beyond the singular boundary, the source likewise proves

$$\Delta_1^2\big(4! C_{6,4}\big)(1) = -\frac{102745512}{2377375},$$

and applies the universal exponent-two bound to this actual coefficient function.

The earlier no-loss counterexample is retained: $v_3(C_{6,3}(2) - C_{6,3}(1)) = 0$. Proving the corrected theorem does not make that false normalized claim true.

Modules. `AllRatiosDwork`, `AllResidueEuler`, `Theorem75`, `AllResidueExamples`.

5. Verification record and claim boundary

Each local module is rebuilt in a fresh output directory, with warnings treated as errors. A live `#printaxioms` audit checks every registered declaration, permitting only `propext`, `Classical.choice` and `Quot.sound`, or subsets.

E.11 Nishibuchi conjecture completion

1. The complete external statement

For $N > 0$ and $0 \le j < N$, use the original unnormalized convention

$$\sum_{d\ge 0} \mathcal{E}_d^{(N,j)} \frac{t^d}{d!} = \left(\sum_{n\ge 0} \frac{t^{Nn}}{(Nn+j)!} \right)^{-1}.$$

The existing packet convention satisfies $\mathcal{E}_{Nn}^{(N,j)} = j! C_{N,j}(n)$. The formalization defines the degree-indexed coefficients through the actual factorial-scaled series coefficients, proves their packet identification, and checks the cleared generating identity with the original exponential section.

Theorem E.1. *Let p be prime, $m > 0$, and assume $m \mid p-1$ or $m = 2$. Put $N = mp$ and $q = \operatorname{lcm}(2, p-1)$. Every $\mathcal{E}_{Nn}^{(N,j)}$ is p-adically integral. For every $r \ge 1$,*

$$\boxed{\mathcal{E}_{Nn+qp^r}^{(N,j)} \equiv \mathcal{E}_{Nn}^{(N,j)} \pmod{p^r} \qquad (0 \le j < N,\ n \ge r).}$$

In particular, the positive sufficient threshold $n_0 = r$ works uniformly in all allowed parameters.

This is the parameter range and displacement of Nishibuchi's Conjecture 3.9 in arXiv:2605.10677v2. The explicit threshold also establishes the suggested bound $n_0 \le r$. The original paper already proves the $p = 2, m = 2, j = 0$ period as a special case, with all starting rows. This extension does not claim that special case as a new discovery.

Principal declaration

`KomorebiPaperIX.NishibuchiCompletion.conjecture_3_9`

The theorem type includes only the stated arithmetic hypotheses. There is no assumed congruence, weighted estimate, or flatness certificate. Precision is the zero-safe predicate $x = 0$ or $v_p(x) \ge r$. Its connection to genuine ideal divisibility in $\mathbb{Z}_p$ is used in the proof.

Claim boundary. This establishes a sufficient threshold and a period, not the earliest starting index, the least period, or a pointwise optimal valuation formula. The correspondence with the externally written statement is a mathematical review of definitions and quantifiers, not an automated semantic-equivalence theorem. The two native conjectures in Paper IX remain unchanged and unproved.

2. The missing dyadic branch

For odd primes, the extra option $m = 2$ already divides $p-1$. Therefore the only allowed family outside the preceding all-residue theorem is $p = 2, m = 2$, with $N = 4$.

The new proof establishes a higher-difference estimate for every one of its four addresses:

$$\boxed{v_2\Big(\Delta_u^k C_{4,j}(n)\Big) \ge \min(k, n) \qquad (0 \le j < 4).}$$

All natural u, k, n are allowed, including zero step, zero order and zero differences. The same bound holds for $j!C_{4,j}(n)$. Both a genuine $\mathbb{Z}_2$ divisibility theorem and a valuation theorem for the original rational sequence are included.

An explicit source instead of an assumed regularity condition

Let $g_j(n) = \binom{4n+j}{4n}^{-1}$. Its four formulas are

$$\begin{aligned} g_0(n) &= 1, \\ g_1(n) &= (4n+1)^{-1}, \\ g_2(n) &= ((4n+1)(2n+1))^{-1}, \\ g_3(n) &= 3((4n+1)(2n+1)(4n+3))^{-1}. \end{aligned}$$

Every denominator is odd. The code constructs the actual unit inverses in $\mathbb{Z}_2$ and proves these identities against the original binomial coefficients, including $n = 0$.

Define a Kummer sequence over a commutative ring by

$$\mathcal{P}_a(f) \quad \Longleftrightarrow \quad a^k \mid \Delta_1^k f(n) \text{ for all } k, n.$$

The exact discrete product rule

$$\Delta_1^s(fg)(n) = \sum_{i=0}^{s} \binom{s}{i} \Delta_1^i f(n)\, \Delta_1^{s-i} g(n+i)$$

is proved by induction. It proves closure under products. Applied to $fg = 1$, the same formula isolates $\Delta_1^s g(n) f(n+s)$. Induction and multiplication by $g(n+s)$ prove closure under actual pointwise inverses.

An affine sequence whose slope is divisible by a satisfies $\mathcal{P}_a$. Consequently the four displayed source sequences satisfy $\mathcal{P}_2$. No numerical interpolation or finite set of samples is used in that deduction.

Source modules: `KummerSequences`, `DyadicFourKernel`.

3. The dyadic differential operator and reciprocity

The operator required here is

$$\boxed{B_4 = \frac{D^8 - D^4}{2}, \qquad D = \frac{d}{dt}.}$$

It is not the preceding $p = 2$ specialization $(D^4 - D^2)/2$. It is also not an ordinary derivation. The new algebra module proves the actual product formula, including the middle derivative sums whose coefficients are $\binom{8}{i}/2$ and $\binom{4}{i}/2$.

Write $\mathcal{H}_{\mathrm{int}}$ for the subring of $\mathbb{Q}_2[[t]]$ whose factorial-scaled coefficients belong to $\mathbb{Z}_2$. Flatness means $B_4^s F \in \mathcal{H}_{\mathrm{int}}$ for every s. The proof of multiplicative and reciprocal closure is induction on s, using the full product correction. It separately proves that the actual reciprocal has integral Hurwitz coefficients when $F(0) = 1$.

Pack g_j into degrees divisible by four and form its actual EGF. The formal coefficient identity is

$$G_j(t) = \sum_{n\ge 0} g_j(n)\frac{t^{4n}}{(4n)!} = j!\sum_{n\ge 0}\frac{t^{4n}}{(4n+j)!}.$$

Its inverse is therefore the original normalized packet EGF. The source Kummer property gives flatness of G_j. The proved reciprocal closure gives flatness of G_j^{-1}.

For a packed coefficient sequence e, the exact extraction formula is

$$\left(U^8 - U^4\right)^k \operatorname{pack}_4(e)(4n-4k) = \Delta_1^k e(n), \qquad k \le n.$$

Here U is the coefficient shift. The nonnegative extraction index is why the first estimate covers $k \le n$. The existing saturation lemma extends it to all k with exponent $\min(k, n)$, and the step-multiple lemma extends step one to every natural u.

Identity with the original rational coefficients

The integral lift is constructed from the decoded coefficients of the actual reciprocal. The code proves

$$\operatorname{coe}(e_j(n)) = C_{4,j}(n) \quad \text{in } \mathbb{Q}_2,$$

and proves that the coefficient map commutes with every difference stencil. Multiplying by $j!$ yields the unnormalized family. The valuation result is not divisibility in a field.

Source modules: `CubicDworkAlgebra`, `CubicDworkClosure`; `DyadicFourDwork` and `DyadicFourCongruences`.

4. From differences to the exact external period

Let f take values in a commutative ring and suppose

$$p^{\min(k,r)} \mid \Delta_h^k f(n) \qquad \text{for every } k.$$

The proved finite Newton expansion is

$$f(n + p^{r-1}h) - f(n) = \sum_{k=1}^{p^{r-1}} \binom{p^{r-1}}{k} \Delta_h^k f(n).$$

For $k \ge r$, the difference already supplies p^r. For $1 \le k < r$, the code proves the integer divisibility

$$p^r \mid \binom{p^{r-1}}{k} p^k.$$

It follows from $k\binom{p^{r-1}}{k} = p^{r-1}\binom{p^{r-1}-1}{k-1}$ and $v_p(k) < k$. Thus every summand is divisible by p^r, including the endpoint $r = 1$.

Divisor branch

For $m \mid p-1$, the preceding all-residue theorem provides the needed differences as soon as $r \le mn$. With $q = \operatorname{lcm}(2, p-1)$, the required packet step q/m is proved to be an integer multiple of $(p-1)/m$. The degree conversion is checked exactly:

$$mp\left(n + p^{r-1}\frac{q}{m}\right) = mpn + qp^r.$$

This retains the stronger sufficient condition $r \le mn$ on the divisor branch. The uniform choice $n \ge r$ implies it for every positive m.

Exceptional branch

For $p = 2, m = 2$, $q = 2$ and the packet displacement is 2^{r-1}. The new dyadic bound supplies the hypothesis when $n \ge r$, and

$$4\big(n + 2^{r-1}\big) = 4n + 2^{r+1}.$$

The $p = 2, m = 1$ branch is also checked using $q = 2$, not an incorrectly substituted $q = p - 1 = 1$.

An exact arithmetic check, not an optimality claim

The independent rational recurrence gives $C_{4,1}(1) = -1/5$ and $C_{4,1}(2) = 121/45$. Their difference is $26/9$, of 2-adic valuation one. This agrees with the new bound at $n = k = u = 1$. The source also contains universal instantiations of the degree-indexed theorem at all four dyadic addresses and at an odd singular address.

Source modules: `PeriodFromDifferences`, `NishibuchiCompletion`, `ConjectureExamples`.

5. Verification, provenance and boundary

Reproduction

```
bash Run-Portable.sh /path/to/Komorebi_Portable_Lean_4.34.0-rc2
```

The external Conjecture 3.9 is distinct from Paper IX's native Conjectures 13.2 and 13.3. Those two remain unproved. The present result does not assert least periods, earliest thresholds, or an optimal exponent for every individual parameter tuple. It makes no historical-priority claim relative to other unpublished work.

F Worked arithmetic and lifting atlas

These examples connect the coefficient calculus with the correction machinery. They instantiate the statements in the main text rather than add new universal claims. Each displayed rational value, residue, and finite reconstruction is checked by the accompanying exact script.

F.1 Compute the reciprocal before estimating it

For width N and address j, the normalized source and its reciprocal are

$$G_{N,j}(t) = j! \sum_{n\ge 0} \frac{t^{Nn}}{(Nn+j)!}, \qquad G_{N,j}(t)^{-1} = \sum_{n\ge 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}.$$

Multiplying the two series and clearing the factorials gives an exact triangular recurrence:

$$C_{N,j}(0) = 1, \qquad C_{N,j}(n) = -\frac{\sum_{r=0}^{n-1} \binom{Nn+j}{Nr} C_{N,j}(r)}{\binom{Nn+j}{Nn}}.$$

address j	C(0)	C(1)	C(2)
0	1	-1	69
1	1	-1/5	121/45
2	1	-1/15	13/45
3	1	-1/35	59/1155

The table is for $N = 4$. The first two address-zero rows illustrate the computation without any valuation estimate:

$$C_{4,0}(1) = -1, \qquad C_{4,0}(2) = -\left(1 + \binom{8}{4}(-1)\right) = 69.$$

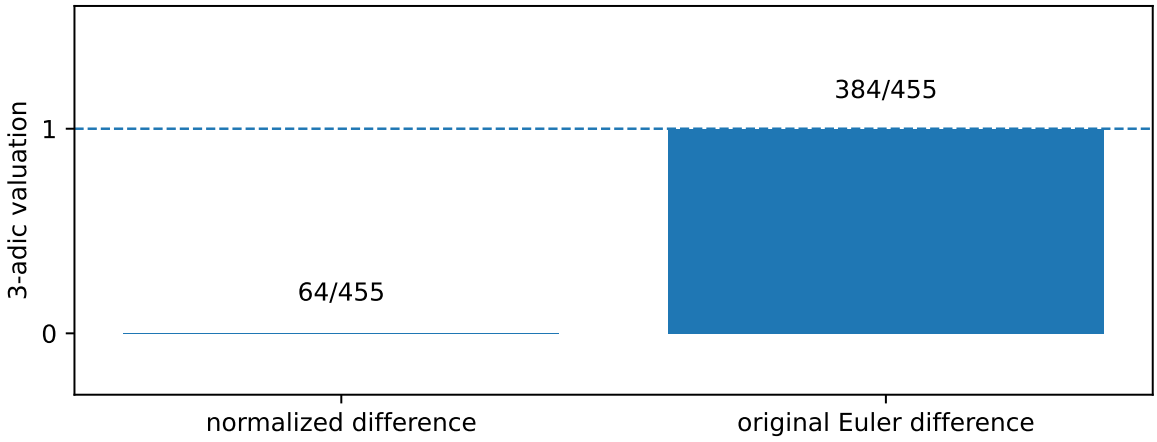
At address one, the same recurrence yields $C_{4,1}(1) = -1/5$ and $C_{4,1}(2) = 121/45$. These are exact rational coefficients, not rounded approximations. The second implementation in the supplied script inverts the ordinary power series first, then multiplies by $(Nn)!$. Agreement between the two constructions checks the factorial placement.

F.2 The normalization loss is a real boundary

The original Euler coefficient is $E_{Nn}^{(N,j)} = j!C_{N,j}(n)$. Multiplication by $j!$ changes the available p-adic precision. That change cannot be discarded outside the unit-factorial window.

Take $p = 3$, $N = 6$, $j = 3$, and compare rows one and two. Exact inversion gives

$$C_{6,3}(1) = -\frac{1}{84}, \quad C_{6,3}(2) = \frac{703}{5460}, \quad C_{6,3}(2) - C_{6,3}(1) = \frac{64}{455}.$$



Neither 64 nor 455 is divisible by three, so the normalized difference has valuation zero. The original coefficient difference is $3! \cdot 64/455 = 384/455$, of valuation one.

This is the counterexample behind the corrected factorial-loss statement. A bound for the original Euler normalization must lose $v_p(j!)$ when translated to $C_{N,j}$. For $0 \leq j < p$, that loss is zero. The distinction concerns which sequence the theorem measures, not a numerical weakness of the calculation.

F.3 Odd primes require the sign twist

The prime-power transfer controls a sign-twisted reciprocal sequence at odd primes. It must not be read as an ordinary-difference theorem for the untwisted coefficients.

At $N = p = 3$ and $j = 0$, put $A(n) = (-1)^n C_{3,0}(n)$. The first three rows are

row n	C at width 3, address 0	(-1)^n C(n)
0	1	1
1	-1	1
2	19	19

The ordinary first difference at row zero is $-1 - 1 = -2$. It is a 3-adic unit, so an asserted lower bound of one for that difference would be false. The twisted first difference is zero, whose mathematical valuation is $+\infty$. At the next row, the twisted difference is $19 - 1 = 18$, of valuation two.

Equivalently, the untwisted sequence is acted on by $T_H = S^H - (-1)^H$, where S advances one row. At odd step one this is $S + 1$, not $S - 1$. The zero in the example is handled by a zero-safe divisibility statement rather than a finite valuation equality. The explicit sign and the zero convention are both part of the theorem's interface.

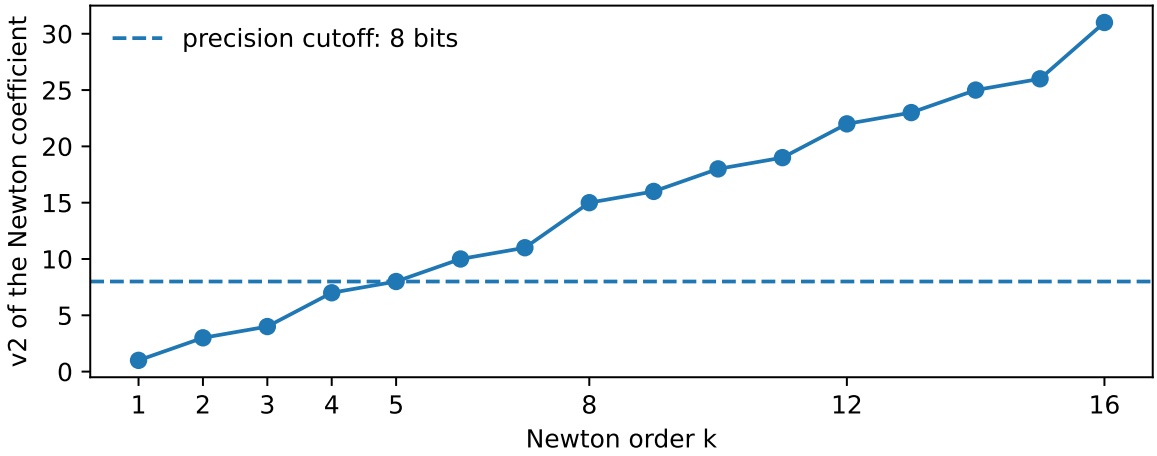
F.4 Newton coordinates turn precision into a finite computation

Let $b_k = \Delta_1^k C_{4,0}(0)$. The first coefficients obtained from the difference triangle are

$$(b_0, b_1, b_2, b_3, b_4) = (1, -2, 72, -33872, 60511872).$$

The exact Newton identity recovers each nonnegative integer row:

$$C_{4,0}(n) = \sum_{k=0}^{n} b_k \binom{n}{k}.$$



For this address the proved law is $v_2(b_k) = 2k - s_2(k)$ for $k \geq 1$. Every $k \geq 5$ therefore has $v_2(b_k) \geq 8$. Since $\binom{n}{k}$ is integral, those terms vanish modulo 256. Thus, for every $n \geq 0$,

$$C_{4,0}(n) \equiv 1 + 254\binom{n}{1} + 72\binom{n}{2} + 176\binom{n}{3} + 128\binom{n}{4} \pmod{256}.$$

The uniform truncation follows from the coefficient theorem. The script separately checks the exact Newton reconstruction and this reduction for rows 0 through 32. Those finite checks validate the worked arithmetic without replacing the theorem.

F.5 A compatible root is built one digit at a time

Take $F(x) = x^2 - 7$ over the 3-adic integers and start with $x_1 = 1$ modulo three. If $F(x_L) = 3^L r_L$, the next representative has the form $x_{L+1} = x_L + 3^L t_L$. Its correcting digit satisfies

$$r_L + 2x_L t_L \equiv 0 \pmod 3.$$

L	3^L	current x	next digit	next x
1	3	1	1	4
2	9	4	1	13
3	27	13	0	13
4	81	13	2	175
5	243	175	0	175

For example, $F(1)/3 = -2$ and $2x_1 = 2$, so the next digit is one and $x_2 = 4$. Then $F(4)/9 = 1$, again giving digit one and $x_3 = 13$. At the next stage the residual is divisible by three, so digit zero is correct. A zero digit is not a stalled construction.

Every center remains congruent to one modulo three. Hence $2x_L$ stays a unit, and the correction equation has a unique digit at every stage. The representatives are compatible and their increments are divisible by 3^L. Completeness gives a limit, and continuity of the polynomial gives a root of F. The finite table exhibits the construction. The unit derivative is what supports indefinite continuation.

F.6 A singular residual can stop a lift

Now take $F(x) = x^2 - 3$ at the residue $x = 0$ modulo three. The residual is $r = F(0)/3 = -1$ and the Jacobian is zero modulo three. The next-digit equation would require

$$-1 + 0 \cdot t \equiv 0 \pmod 3,$$

which has no solution.

digit t	candidate y	y^2 - 3	residue mod 9
0	0	-3	6
1	3	6	6
2	6	33	6

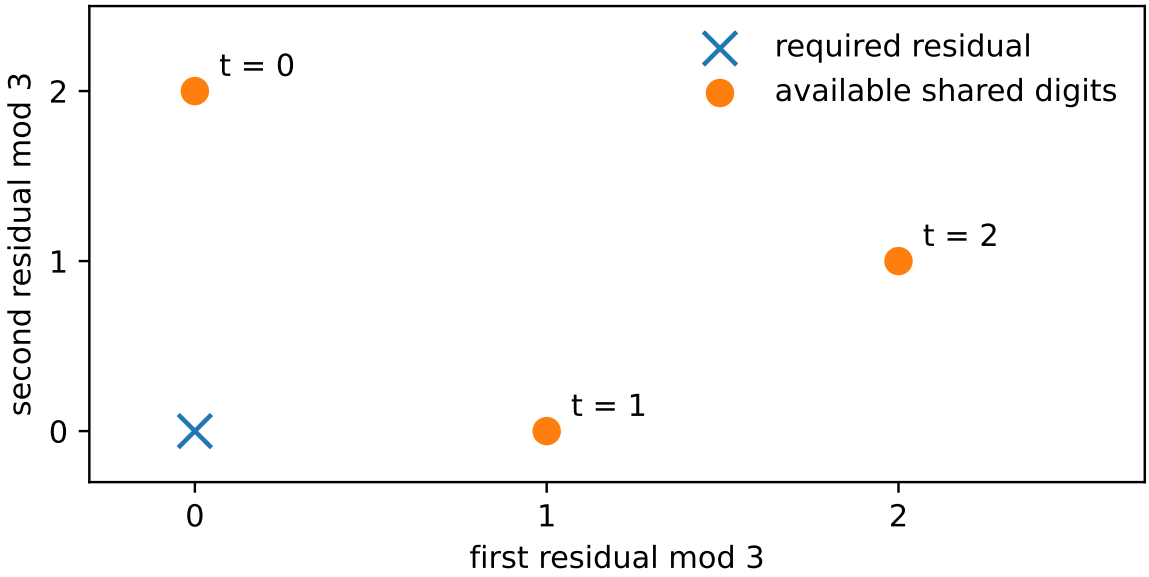
The three representatives of the old root class modulo nine all leave residual six modulo nine. Thus the root class exists at precision three but does not lift to precision nine. This is an exact obstruction, not failure of an iterative search.

A singular Jacobian alone does not imply failure. For $F(x) = x^2$ at the same center, the residual is zero and all three candidate digits lift to roots modulo nine. The obstruction is the residual class in the cokernel of the actual Jacobian. It is this class, rather than the rank by itself, that decides whether a next lift exists.

F.7 Separate corrections need not be one shared correction

Consider the two-component system $F(x) = (x, x-3)$ at $x = 0$, with $q = 3$ and $L = 1$. Both components vanish modulo three. After writing a candidate as $y = 3t$, its residual vector modulo three is

$$\bar{r} + J\bar{t} = (0,2) + (t,t) = (t, t-1).$$



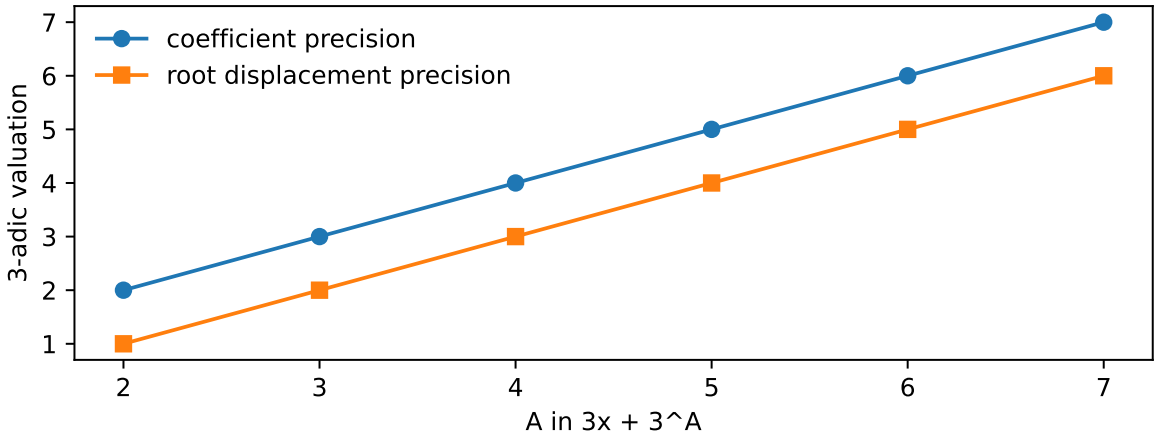
The first coordinate can be corrected by $t = 0$. The second can be corrected by $t = 1$. No digit corrects both. The available residuals are $(0,2)$, $(1,0)$, and $(2,1)$, none of which is $(0,0)$.

The combined Jacobian maps t to the diagonal vector (t,t). Its image is smaller than the product of its coordinate images. Checking the two projected obstructions independently therefore loses the compatibility condition. The common-correction theorem retains the full residual vector and the shared digit space, which is exactly the information needed to detect this failure.

F.8 A derivative weight can cost one digit of root precision

Compare the linear equations $F(x) = 3x = 0$ and $G_A(x) = 3x + 3^A = 0$, where $A \geq 2$. Their coefficient difference has 3-adic valuation A, but their roots differ by

$$x_A - 0 = -3^{A-1}, \qquad v_3(x_A) = A - 1.$$



The derivative has valuation one. Solving the correction equation divides by three, so one digit of coefficient precision is lost in root displacement. The calculation is exact for every displayed A and also follows immediately from the two linear equations.

This example explains why weighted lifting and deformation estimates track a Jacobian loss rather than silently treating every invertible derivative over $\mathbb{Q}_p$ as a unit over $\mathbb{Z}_p$. A valuation-zero derivative would not incur this particular loss. Nonlinear systems additionally require their stated neighborhood, residual, and right-inverse hypotheses.

The eight examples use exact rational and integer arithmetic. The computation scripts and tested ranges accompany the sources. This atlas adds no Lean endpoint and changes no formal claim in the companion.

Reading scope

This guide explains the indexing, valuation gains, exact periods, and remaining boundaries through exact arithmetic and theorem-derived figures. The main text supplies the proofs. The formal companion supplies the formal source and evidence map.

1 The object: rows, degrees, and normalization

The visual guide accompanies *Euler Congruences: Completion, Factorial Strengthening, and Sharp Dyadic Geometry*. It makes the indexing, precision gains, and boundaries of the results visible. The main text contains the proofs. The separate Lean guide records which statements are already formalized and which belong to the new mathematical extension.

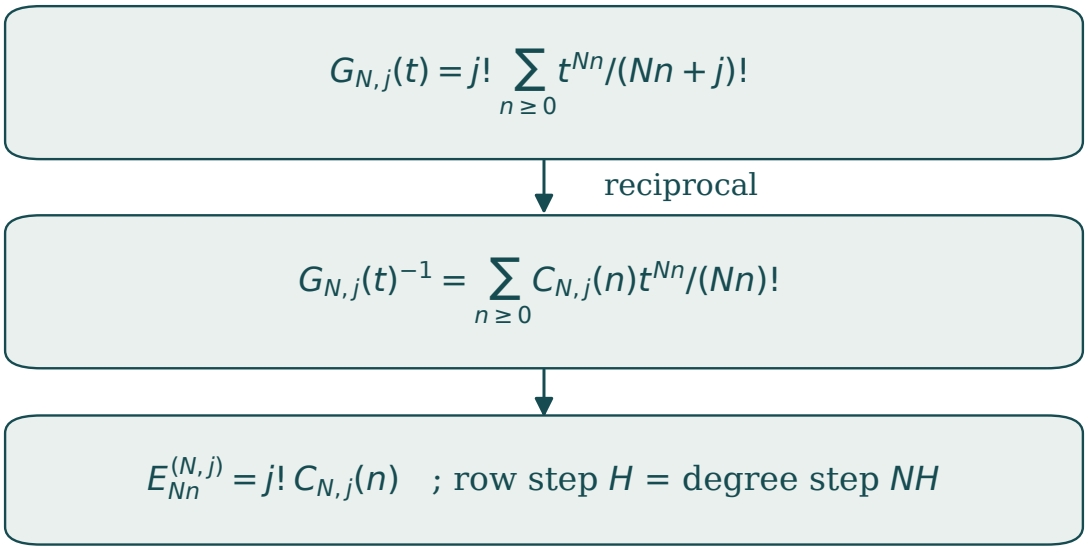


Figure 1. The actual reciprocal series is the source of every coefficient in this guide. The factorial normalization changes valuations as well as the constant term.

There are two coefficient conventions, and they must stay distinct:

$$C_{N,j}(0) = 1, \qquad E_0^{(N,j)} = j!, \qquad v_2(E_{Nn}^{(N,j)}) = v_2(C_{N,j}(n)) + v_2(j!).$$

The last equality uses $v_2(0) = +\infty$. Congruences of dyadic rational numbers mean membership in the corresponding ideal of $\mathbb{Z}_2$. A displayed rational denominator is invertible modulo 2^r when it is odd.

Symbol	Meaning	At rank four
n	packed row index	Euler degree $4n$
H	positive row step	Euler degree step $4H$
k	number of differences	order k, independent of n
j	address within a section	$0, 1, 2, 3$

For example, $C_{4,1}(1) = -1/5$ and $C_{4,1}(2) = 121/45$. Their difference is $26/9$, of valuation one. The row difference joins Euler degrees 4 and 8.

2 What the original conjecture asked

Nishibuchi's Conjecture 3.9 concerns a prime p, a positive integer m with $m \mid p-1$ or $m = 2$, and every address $0 \le j < mp$. Put

$$N = mp, \qquad q = \operatorname{lcm}(2, p-1).$$

It asserts p-adic integrality of the actual generalized Euler coefficients and eventual periodicity at every precision:

$$\text{for every } r \ge 1, \quad E^{(mp,j)}_{mpn+qp^r} \equiv E^{(mp,j)}_{mpn} \pmod{p^r} \quad \text{for all sufficiently large } n.$$

The completed theorem gives the explicit sufficient threshold $n \ge r$ in every permitted branch. Its formal capstone is the theorem named `NishibuchiCompletion.conjecture_3_9`. No conjectural flatness premise is supplied to that theorem.

Prime branch	Allowed m	q	Row step for the original shift
$p = 2$	1	2	2^r
$p = 2$	2	2	2^{r-1}
p odd	$m \mid p-1$	$p-1$	$\dfrac{p-1}{m}p^{r-1}$

For odd primes, $m = 2$ is already in the divisor branch. The conversion is exact:

$$N\left(\frac{q}{m}p^{r-1}\right) = qp^r.$$

At $p = 2, m = 2$, the original conjectured Euler-degree shift 2^{r+1} is the row step 2^{r-1} for the rank-four sequence.

The source already contains special cases. In particular, Nishibuchi's Proposition 3.10 treats the rank-four address-zero dyadic period from row zero and an eventual cubic case. The completion covers the full quantified conjecture. The new dyadic work then removes starting-row restrictions from higher differences and determines substantially more precise laws.

The conjecture is cited in the source list at the back. The statement uses its coefficient normalization and actual Euler degrees.

3 Removing the starting-row threshold

At rank four the earlier completed estimate was $v_2(\Delta_H^k C_j(n)) \geq \min(k, n)$. The strengthened theorem reaches every starting row, including $n = 0$, and every order, including $k > n$.

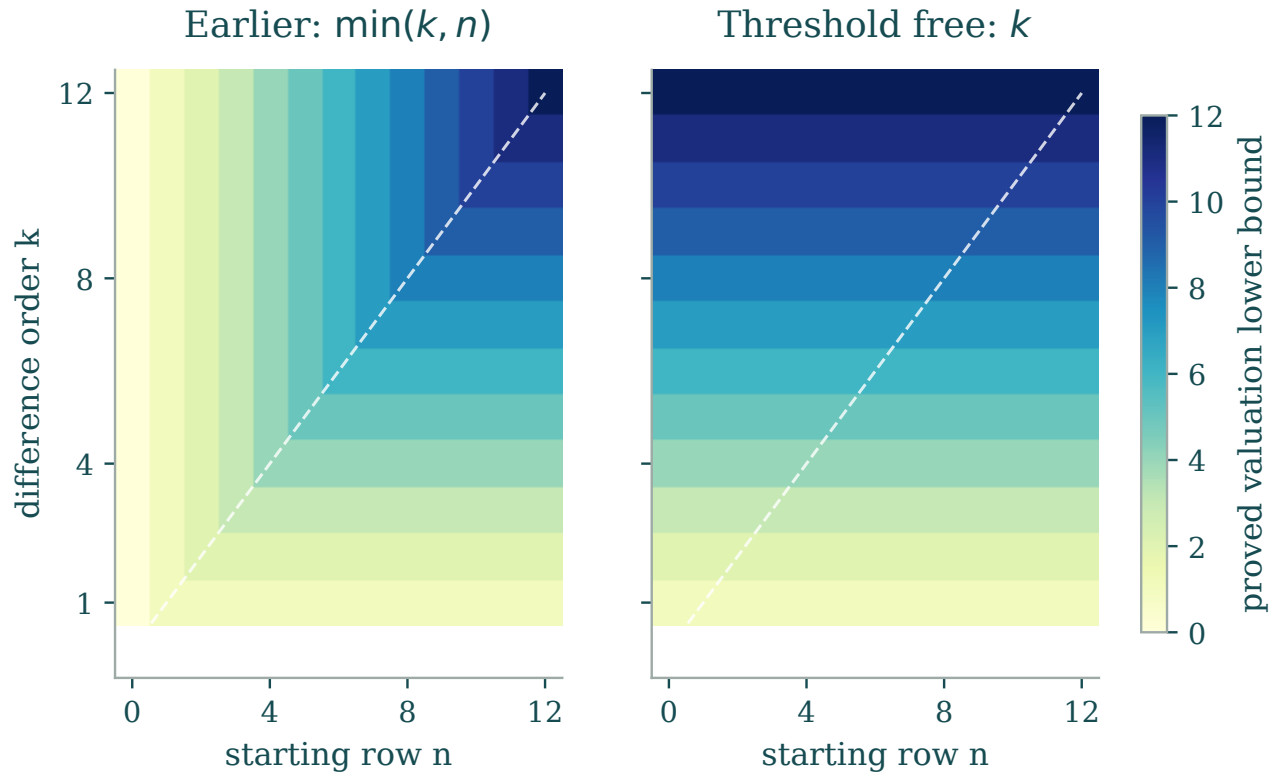


Figure 2. Unit-step lower bounds on a finite display window. Each colored cell is obtained from a proved formula, rather than from sampled coefficients. The dashed line indicates $k = n$.

For *every* dyadic section size $N = 2^a$, $a \geq 0$, every permitted address, and all positive steps,

$$v_2\big(\Delta_{H_1} \cdots \Delta_{H_k} C_{N,j}(n)\big) \geq k + \sum_{\nu=1}^{k} v_2(H_\nu), \qquad n \geq 0.$$

The corresponding statement for $E_{Nn}^{(N,j)}$ adds $v_2(j!)$.

The proof acts on the full factorial-scaled power series before it extracts a packed row. This avoids the old extraction constraint $k \leq n$. Reciprocal induction controls the unique term with the highest inverse order in the commuting product expansion. All other terms are covered by the preceding induction stages.

The figure displays the gain at unit step. An even step contributes additional precision, and the theorem permits different steps in the same difference.

4 Each step contributes its own precision

Write $Sf(n) = f(n+1)$ and $\Delta_H = S^H - 1$. Mixed differences commute. Their exact finite expression is

$$\Delta_{H_1} \cdots \Delta_{H_k} f(n) = \sum_{A \subseteq \{1,\dots,k\}} (-1)^{k-|A|} f\left(n + \sum_{i \in A} H_i\right).$$

Repeated subset offsets are combined with their signed multiplicities.

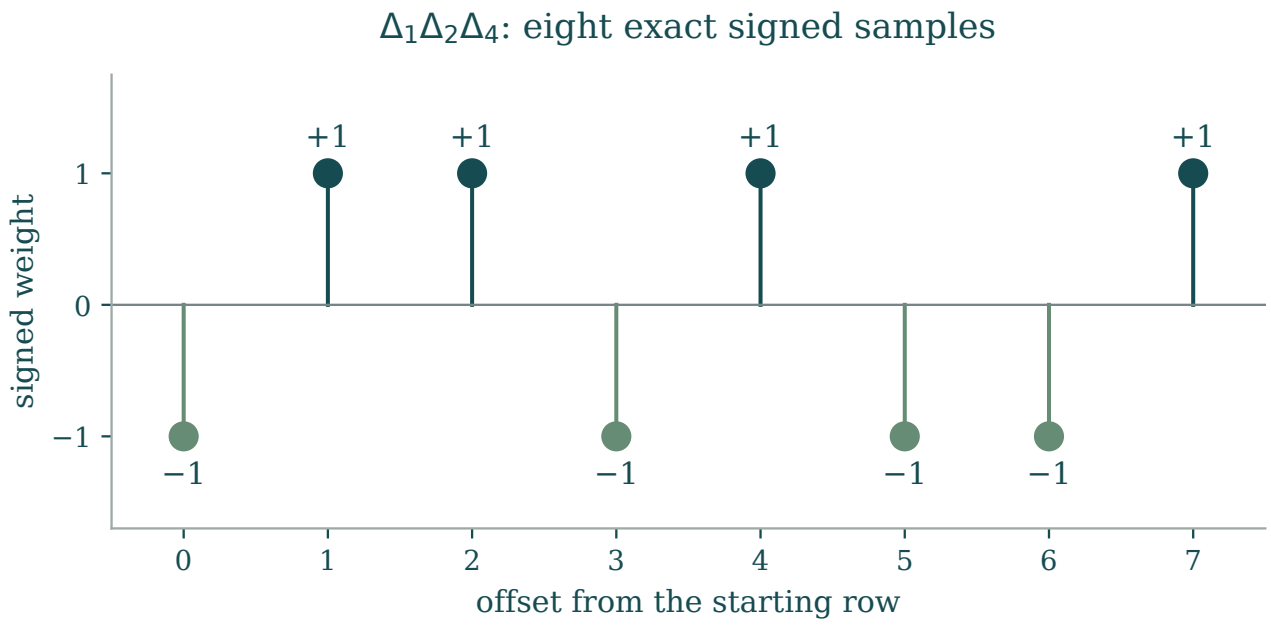


Figure 3. For steps $(1, 2, 4)$ all eight offsets are distinct. This is an exact signed stencil, rather than an approximate derivative.

For this stencil, the general dyadic bound is

$$3 + v_2(1) + v_2(2) + v_2(4) = 6.$$

At rank four the factorial theorem improves it to seven for addresses zero and one. Exact rational evaluation gives $v_2(\Delta_1\Delta_2\Delta_4 C_0(0)) = 7$. The order is odd, so the mixed-step equality criterion also proves this valuation at *every* starting row.

There is a prime-power extension. For $N = p^a$, $a \geq 1$, the sign-corrected operator is

$$T_H = S^H - (-1)^H, \qquad v_p\Big(T_{H_1} \cdots T_{H_k} E^{(N,j)}(n)\Big) \geq v_p(j!) + \sum_{i=1}^{k} \big(1 + v_p(H_i)\big).$$

Here $E^{(N,j)}(n) = E^{(N,j)}_{Nn}$. For an odd prime, an ordinary first difference need not have positive valuation: at address zero its first two normalized values are 1 and -1, with difference -2. The sign correction is therefore essential to this prime-power theorem.

5 The final rank-four congruence

For $n \ge 0$, $k \ge 1$, and $H \ge 1$, the actual Euler coefficients satisfy

$$\sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} E^{(4,j)}_{4(n+iH)} \equiv 0 \pmod{2^{\,v_2(k!)+k v_2(H)+A_j(k)}},$$

where

$$\begin{aligned} A_0(k) = A_1(k) &= k, \\ A_2(k) &= \max\{k+4,\ 2k+\lfloor k/2 \rfloor + 1\}, \\ A_3(k) &= k+2. \end{aligned}$$

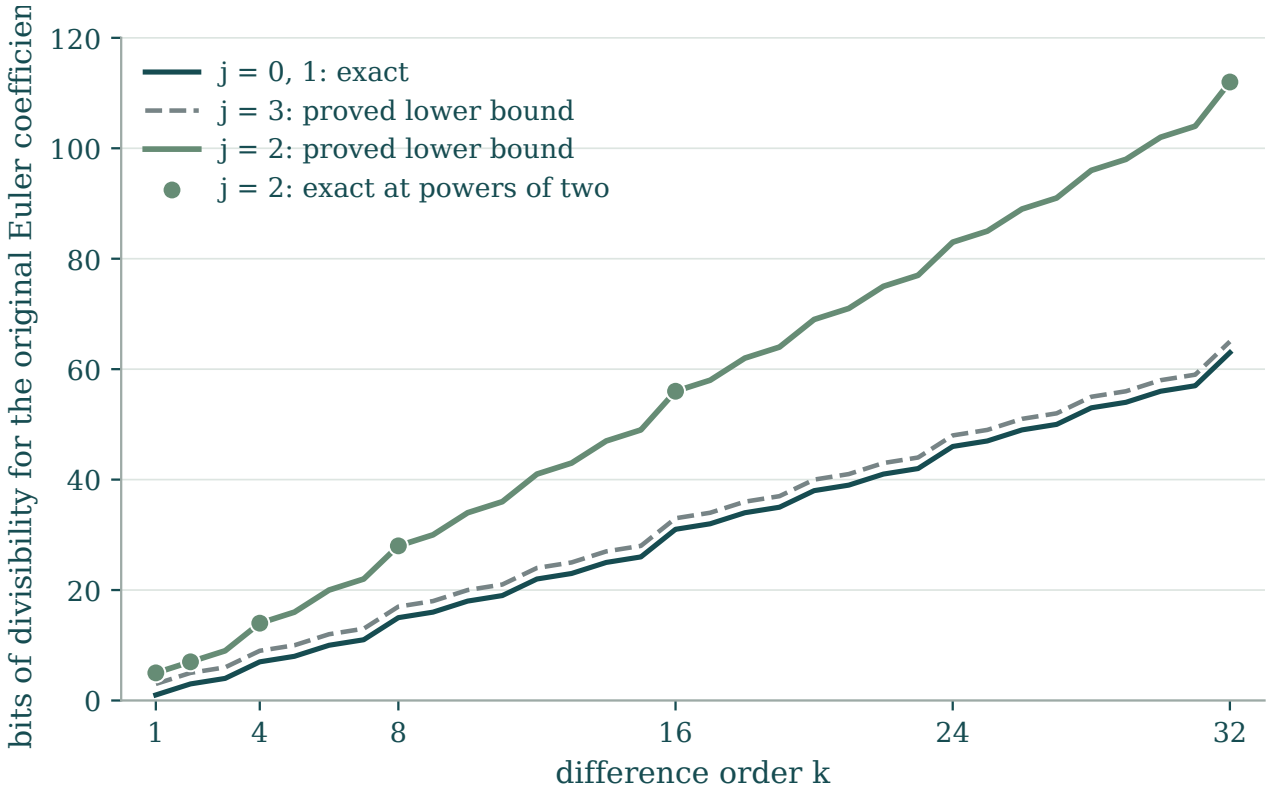


Figure 4. The exponent of the final Euler congruence at unit step. The address zero and one curve is exact at every order. Marked address-two orders are universally exact. The other displayed address-two and address-three values are proved lower bounds.

At a common step with $v_2(H) = s$, add ks to every displayed exponent. Thus a doubling of the step gains one additional bit *per difference*, rather than one bit for the whole expression.

This statement combines two different proved bounds at address two. A lower bound is not automatically an exact valuation: divisibility may be stronger at unmarked orders or at particular starting rows.

6 Why address two has a different slope

The normalized address-two sequence obeys both

$$v_2(\Delta_{H_1}\cdots\Delta_{H_k}C_2(n)) \geq k + v_2(k!) + 3 + \sum_i v_2(H_i),$$
$$v_2(\Delta_{H_1}\cdots\Delta_{H_k}C_2(n)) \geq 2k + v_2(k!) + \lfloor k/2\rfloor + \sum_i v_2(H_i).$$

Their maximum supplies the useful envelope.

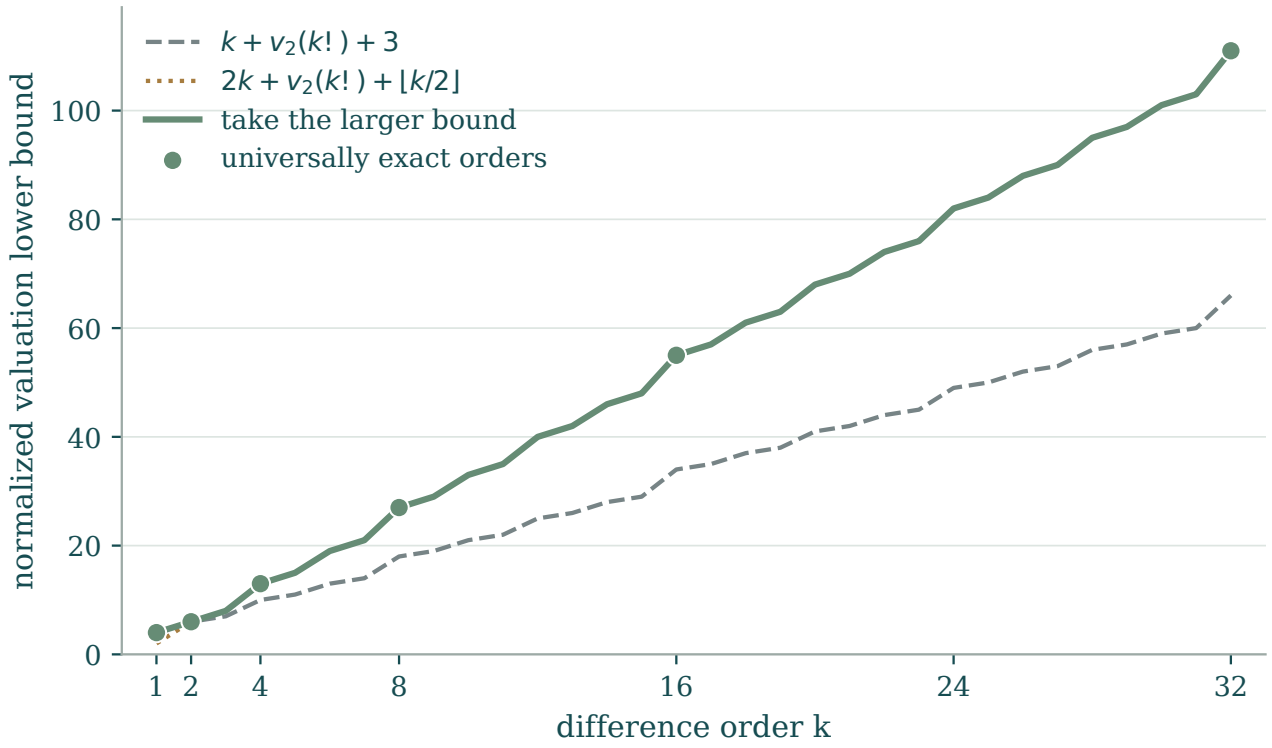


Figure 5. Normalized address-two precision. The paired filtration controls the large-order slope. At order one the constant-surplus bound is stronger.

For every base row and every positive step list, equality is proved when k is a power of two:

$$v_2(\Delta_{H_1}\cdots\Delta_{H_k}C_2(n)) = V_2(k) + \sum_i v_2(H_i),$$

with $V_2(1) = 4$, $V_2(2) = 6$, and $V_2(k) = 2k + v_2(k!) + k/2$ for $k = 2^r \geq 4$. The infinite marked family establishes sharpness at unbounded orders.

For original Euler coefficients add one, since $v_2(2!) = 1$. The extra $\lfloor k/2\rfloor$ comes from a paired divided-power filtration in the Bernoulli transformation. It does not arise by fitting the plotted points.

7 When mixed steps attain the baseline

For addresses $j = 0, 1$, put

$$V = k + v_2(k!) + \sum_i v_2(H_i), \qquad q = \#\{i : H_i \text{ is even}\}.$$

The valuation equals V exactly when k is odd or q is even.

Order parity	Number of even steps	Conclusion
k odd	either parity	exact valuation V
k even	q even	exact valuation V
k even	q odd	valuation at least $V + 1$

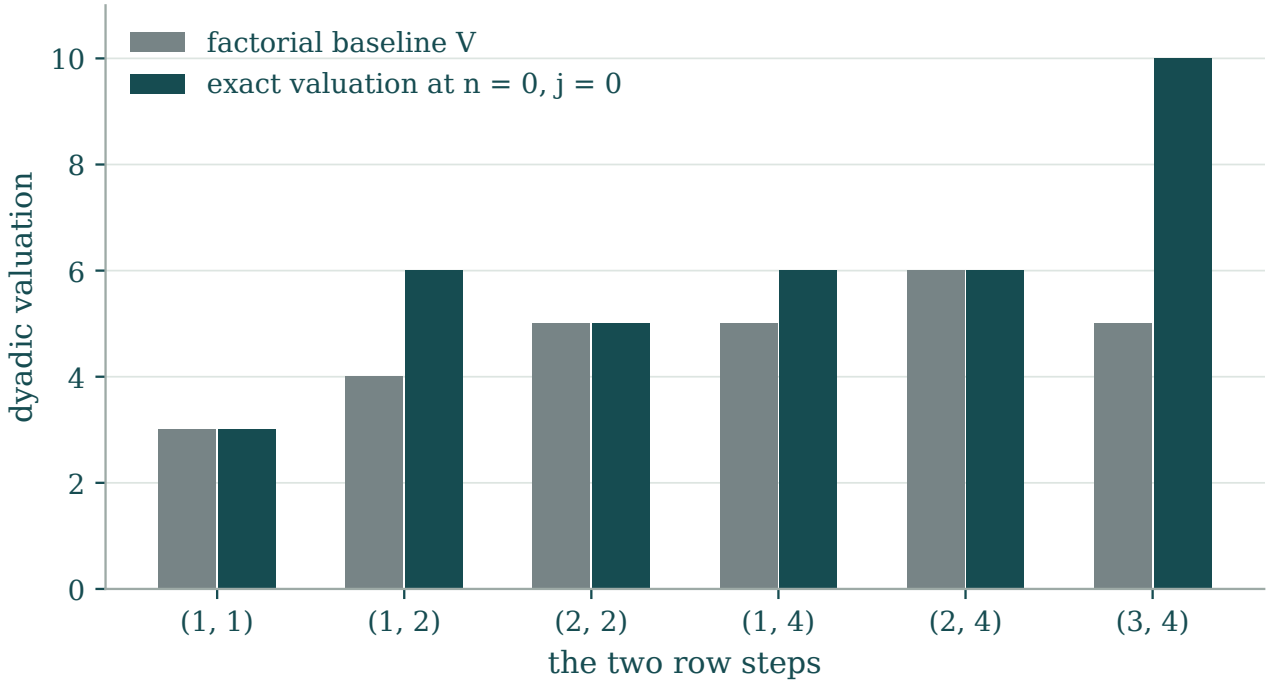


Figure 6. Exact rational examples at order two and base zero. Strict cases show why the factorial baseline cannot always be called an equality for mixed steps. The universal parity criterion is proved separately.

For instance,

$$\Delta_1\Delta_2 C_0(0) = -33728, \qquad v_2(-33728) = 6, \qquad V = 4.$$

Cancellation removes the leading term, and here it removes two bits beyond the baseline. The parity theorem proves strictness in this case. It does not give a complete formula for the further depth.

With a common step, q is either zero or k. Consequently the exceptional parity never occurs, and the homogeneous address-zero/one formula is always exact.

8 First differences determine an exact metric

All four rank-four sequences satisfy the sharp first-difference law

$$v_2(C_j(n+H) - C_j(n)) = v_2(H) + b_j, \qquad b = (1,1,4,2).$$

For the actual Euler coefficients this becomes

$$v_2(E^{(4,j)}_{4(n+H)} - E^{(4,j)}_{4n}) = v_2(H) + B_j, \qquad B = (1,1,5,3).$$

Both laws hold at every starting row and every positive integer step.

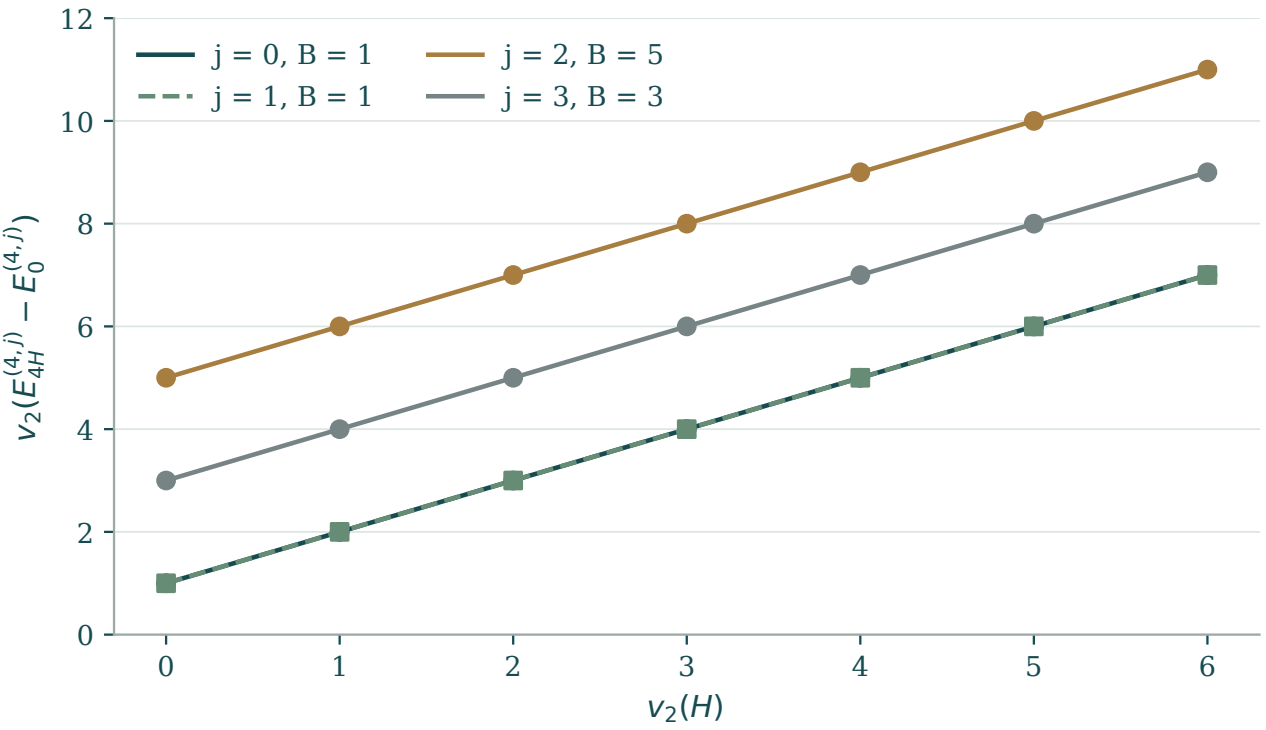


Figure 7. The proved Euler metric laws, with exact modular sample points at base zero and steps $H = 2^s$. Addresses zero and one have the same scale. Their circle and square marks overlap.

In dyadic distance, $|x|_2 = 2^{-v_2(x)}$. After continuous interpolation to $\mathbb{Z}_2$, the law reads

$$|E^{(4,j)}(x) - E^{(4,j)}(y)|_2 = 2^{-B_j}|x-y|_2.$$

It says exactly how much the map shrinks distances. It also prevents any hidden collision at finer precision: distinct inputs remain distinct after the known scale is removed.

The scale of address two is 2^{-5} in the original normalization and 2^{-4} in the normalized sequence. Omitting its factor 2! would change the period claimed for the original coefficients.

9 Least periods, rather than sufficient periods

At precision 2^r the least positive *row* period is

$$P_j(r) = 2^{\max(0,r-B_j)}, \qquad B = (1,1,5,3).$$

The exact valuation law proves both periodicity and minimality. A step H is a period precisely when $v_2(H) + B_j \geq r$.

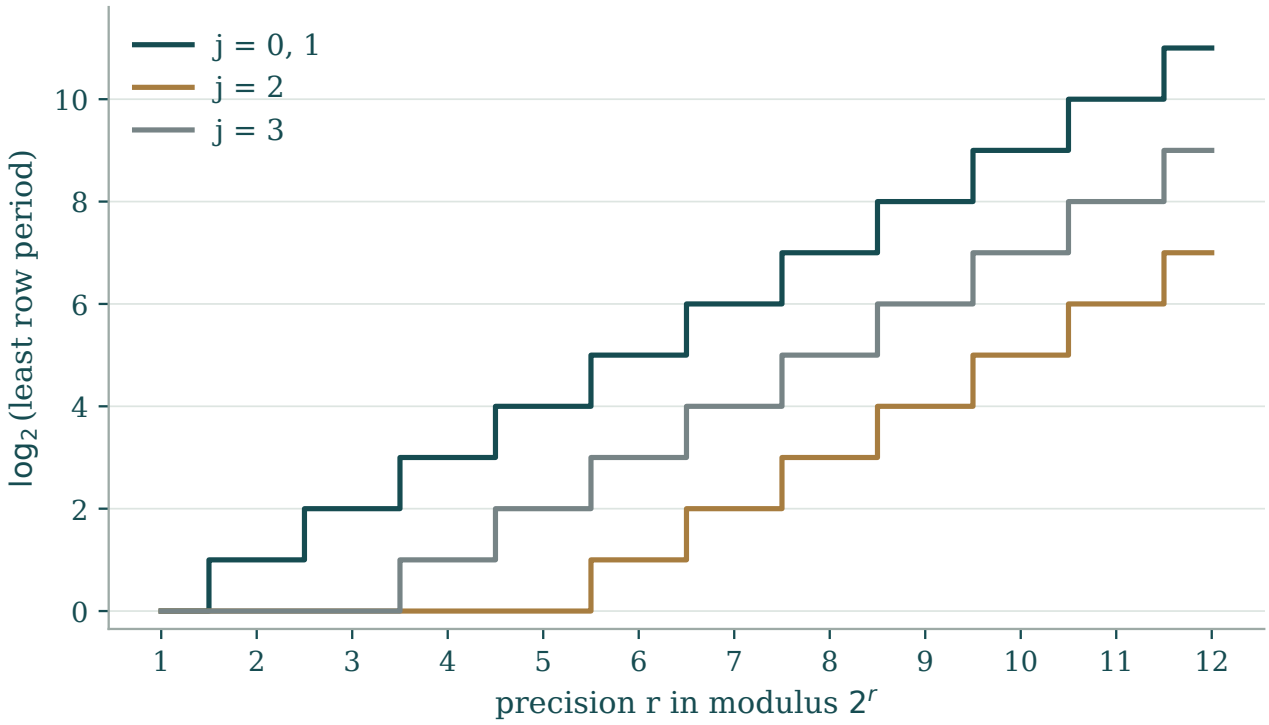


Figure 8. Binary logarithms of the least row periods. A height of zero means period one. These are theorem-derived values, valid at every precision.

Modulo $256 = 2^8$, the least periods are

$$(P_0(8), P_1(8), P_2(8), P_3(8)) = (128, 128, 8, 32).$$

In actual Euler degree, multiply by four:

$$(512, 512, 32, 128).$$

For address two, precision at most 2^5 already has row period one. Its Euler coefficients all lie in $2 + 32\mathbb{Z}_2$. Further precision resolves additional dyadic digits within that residue class.

The original rank-four conjectural shift has row step 2^{r-1}. It remains a valid period for every address, but addresses two and three usually admit shorter periods. The sharp theorem identifies the shortest one.

10 A full period visits every allowed residue

For each address the exact interpolation image is

$$E^{(4,j)}(\mathbb{Z}_2) = j! + 2^{B_j}\mathbb{Z}_2.$$

When $r > B_j$, a least period lists every residue in this coset modulo 2^r exactly once. The following figure checks this assertion by exact arithmetic at precision 256.

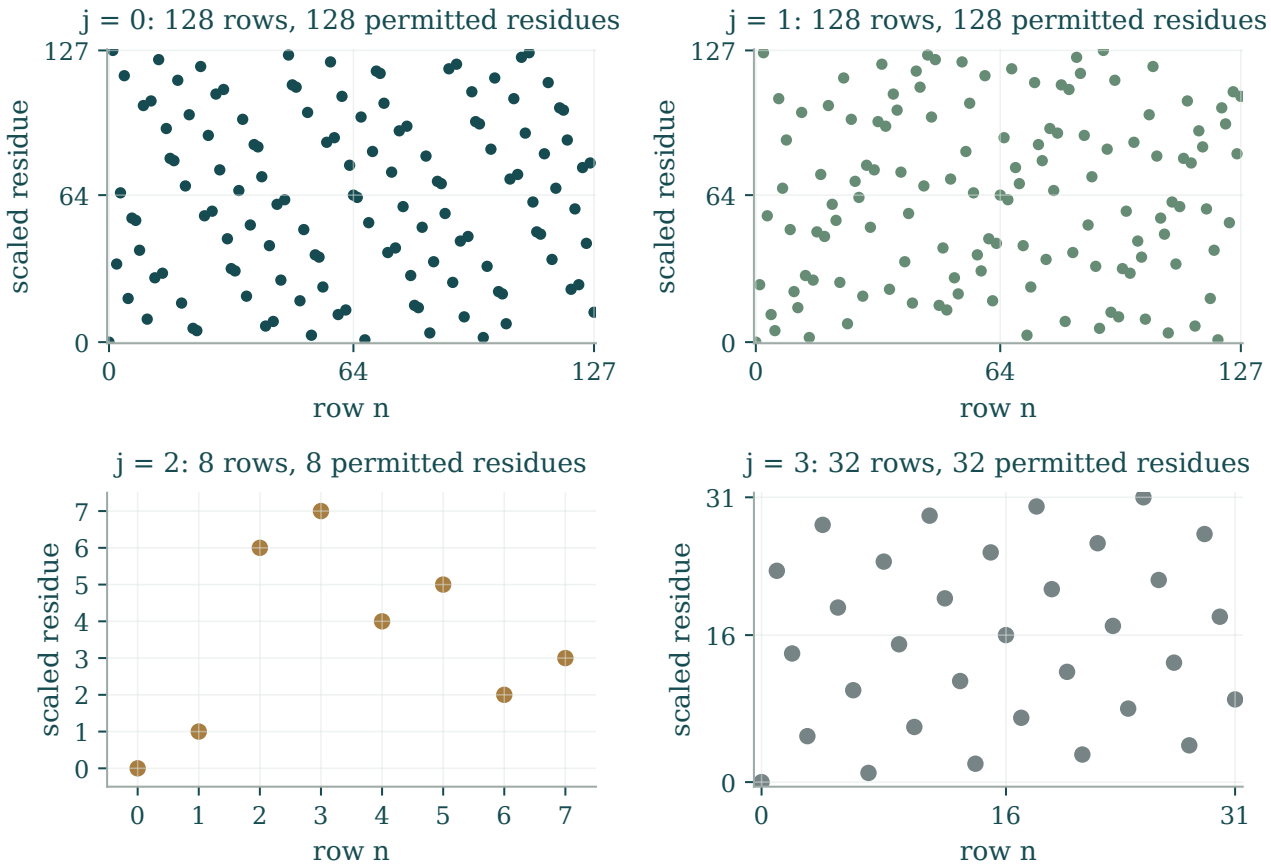


Figure 9. Exact residue permutations modulo 256. The vertical coordinate is $y_n = (E_{4n}^{(4,j)} - j!)/2^{B_j}$ modulo $P_j(8)$. Each panel contains one full least period. Every vertical residue occurs once.

No decimal approximation is used to form these residues. The generating series is inverted modulo 2^{256} using odd denominators, then the exact result is reduced modulo 256. An independent rational inversion checks the initial coefficients.

The shapes are permutations, not monotone real functions. Their organizing property is dyadic distance: agreement in input low bits forces the same agreement in the scaled output low bits.

11 Seeing the scaled isometry

The map

$$x \longmapsto \frac{E^{(4,j)}(x) - j!}{2^{B_j}}$$

is a bijective isometry of $\mathbb{Z}_2$. Its finite reductions preserve the number of common low binary digits.

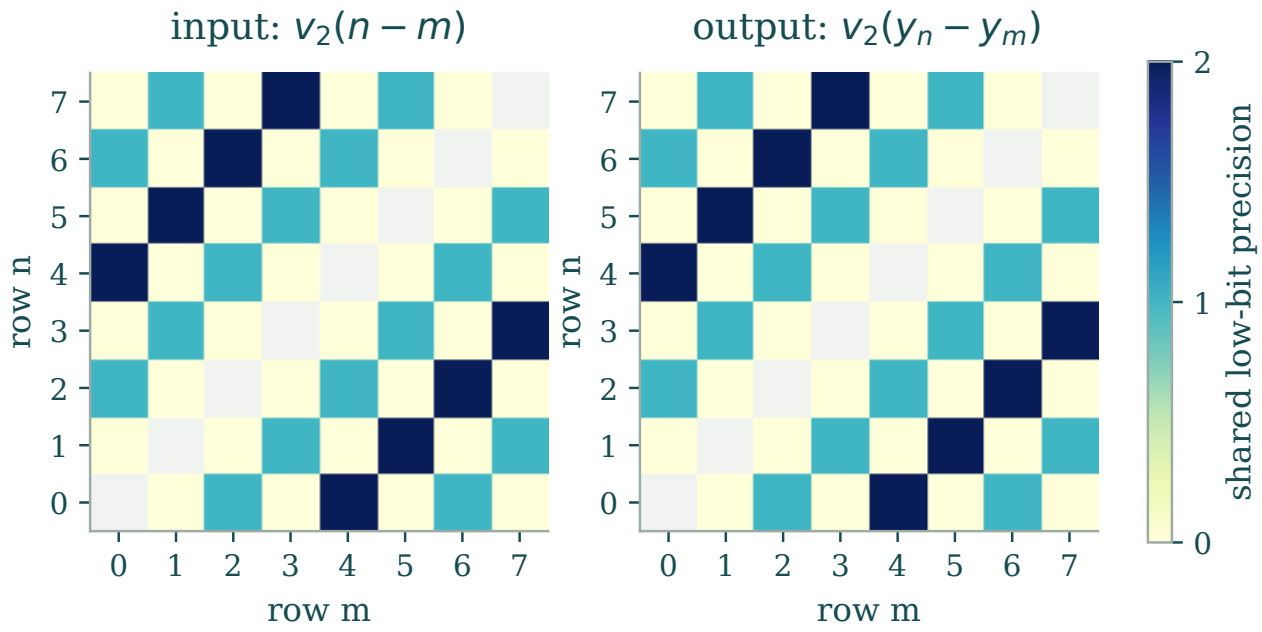


Figure 10. Address two modulo 256 gives an eight-element quotient. The two exact matrices coincide: every pair of distinct rows has the same dyadic valuation before and after the scale is removed. The diagonal represents $v_2(0) = +\infty$ and is left blank.

If $n - m$ is odd, the rows disagree in their first binary digit and the matrix entry is zero. If it is divisible by two but not four, they share exactly one low digit. An entry two means agreement through two low digits.

This structure explains the residue permutation in the preceding section. At each new precision, every input residue has exactly two lifts. An isometry distinguishes those lifts at the corresponding output precision. Starting from the single residue at the scale threshold, induction gives surjectivity and a unique inverse residue at every finite level.

The finite matrix is an illustration of that proof. Its eight rows alone would not establish a statement about the whole of $\mathbb{Z}_2$.

12 How far the simple metric law extends

Sharp first-difference laws hold for every address at ranks 2, 4, and 8. The arrays below give their normalized scales $b_{N,j}$ and original Euler scales $B_{N,j} = b_{N,j} + v_2(j!)$.

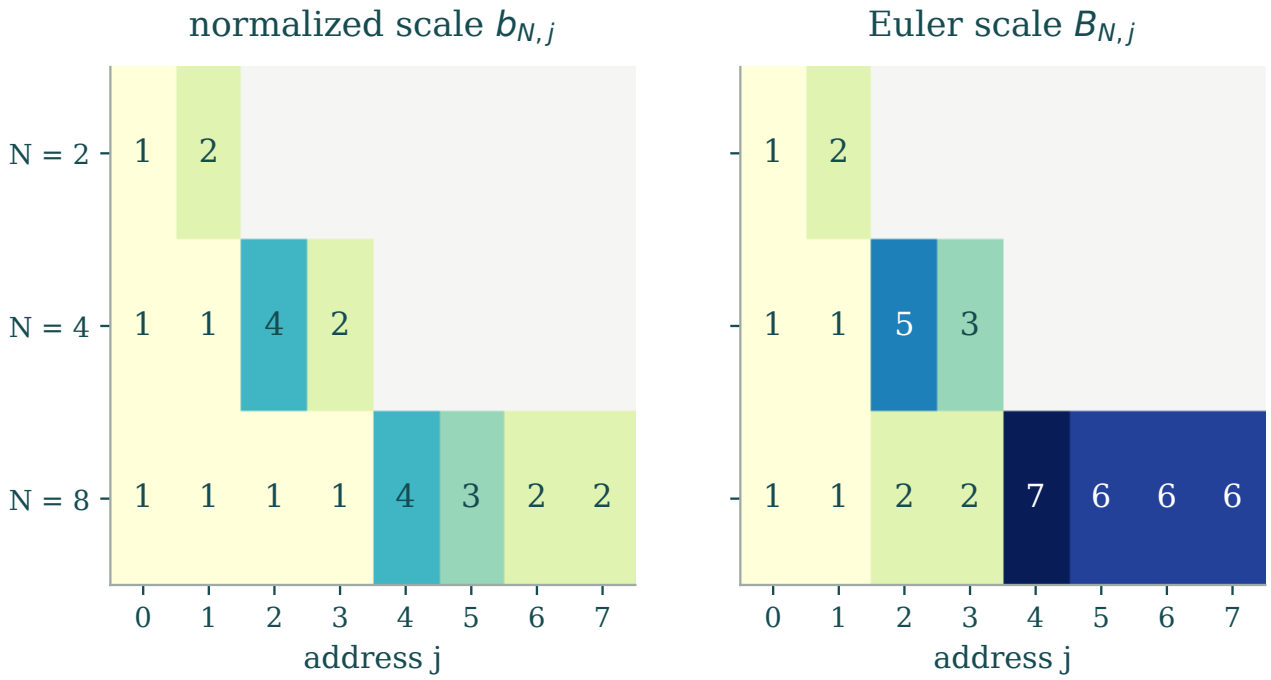


Figure 11. Proved scale arrays for the three small ranks. Blank cells are outside the address range. The Euler normalization contributes the factorial valuation, which is already substantial at the larger addresses of rank eight.

There is also an infinite family: at every $N = 2^a \geq 4$, all lower-half addresses $j < N/2$ have normalized scale one. Their Euler scale is $1 + v_2(j!)$.

The same constant-scale proposal fails at some upper-half addresses:

(N, j)	$v_2(C(1) - C(0))$	Step-two prediction	Actual step-two valuation
$(16, 9)$	5	6	7
$(32, 18)$	6	7	6
$(32, 19)$	5	6	7

These are exact rational counterexamples to the additional metric proposal. They do not contradict the universal higher-difference *lower* bound, which continues to hold at every address of every dyadic rank.

13 From factorial gain to analytic radius

At every center $x_0 \in \mathbb{Z}_2$, the interpolations admit ordinary Taylor series $C_j(x_0+z) = \sum_{d\ge 0} a_{j,d}(x_0) z^d$. For positive degrees,

$$\begin{aligned} v_2(a_{j,d}(x_0)) &= d && (j=0,1),\\ v_2(a_{2,d}(x_0)) &\ge \max\{2d + \lfloor d/2 \rfloor,\ d+3\},\\ v_2(a_{3,d}(x_0)) &\ge d. \end{aligned}$$

At address two equality with $5d/2$ holds along every power-of-two degree $d \ge 4$.

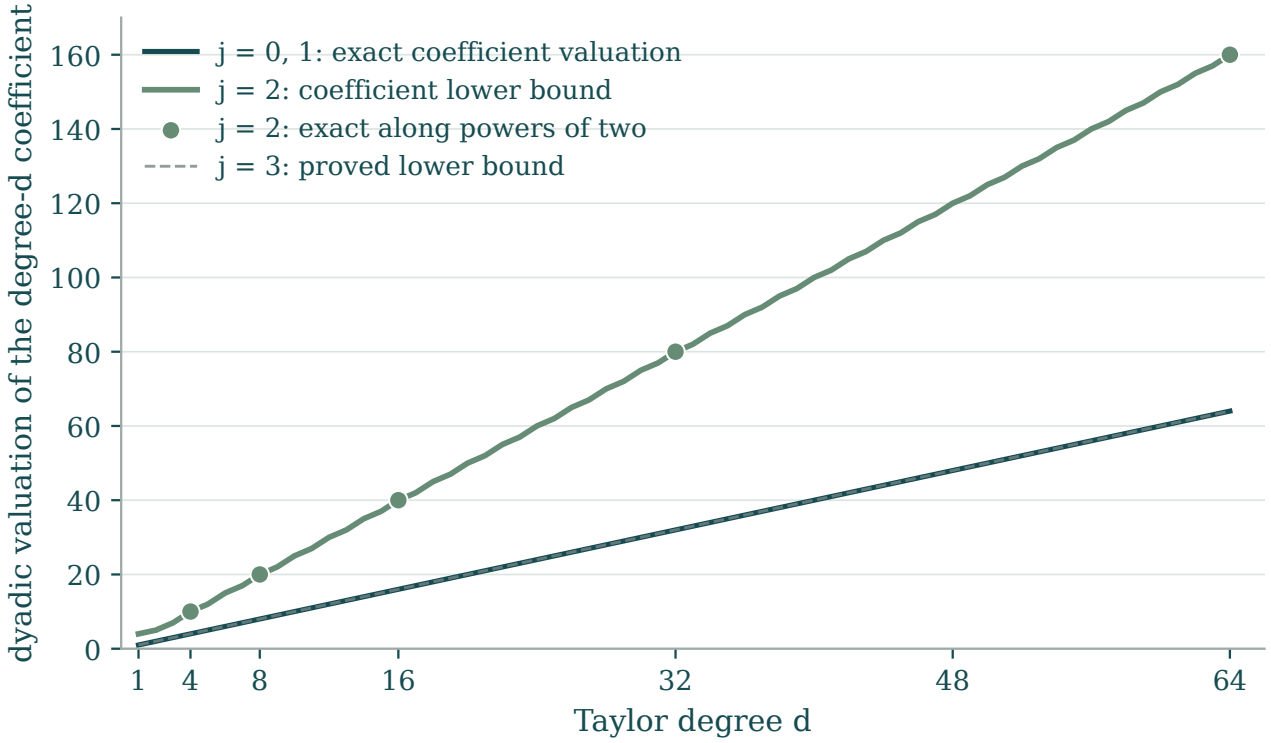


Figure 12. The Taylor coefficient precision slopes. The degree-zero coefficient is excluded. Marked address-two degrees belong to an unbounded proved equality family, rather than a finite observed pattern.

The radius is $2^{\liminf_d v_2(a_d)/d}$ in the normalized dyadic absolute value. Hence

$$R_0 = R_1 = 2, \qquad R_2 = 2^{5/2}, \qquad R_3 \ge 2.$$

The exact radius of address three is not asserted. For address two, the lower bound gives convergence inside the claimed radius and the unbounded equality family prevents a larger one.

The proof converts the factorial Newton estimates into ordinary Taylor coefficients with the Stirling relations. Generic continuity or first-order Lipschitz bounds alone do not supply this analytic conclusion.

14 Which evidence does which job

The package has three complementary levels of evidence. They answer different questions, and the final claims retain those distinctions.

Evidence	What it establishes	Verification scope
Lean theorem	The precise formal statement follows from its recorded dependencies and the accepted foundational axioms.	The completed Nishibuchi conjecture, earlier Chapter IX theory, and the inherited rank-four ordinary-step, sharp-period, and isometry results. See the Lean guide for build provenance.
Algebraic proof and audit	A universal mathematical result, subject to the written proof and its stated inputs.	The new general prime-power transfer, mixed-step and factorial gains, exact higher-order families, broader dyadic metric laws, and radius results.
Exact computation	The implemented assertion holds in its recorded finite domain, or an exact counterexample refutes an extension.	Coefficient regressions, period illustrations, and sharpness or cancellation checks.

The generic theorem uses the integral Hurwitz-series product and reciprocal induction. The factorial theorem uses divided powers and explicit transforms of the ordinary Euler and Bernoulli series. The ordinary Euler arithmetic input is the classical Frobenius congruence, with normalization checked against Gessel's primary exposition. Address two has a self-contained central-factorial Bernoulli argument. Addresses one and three use the proved tangent and cotangent transfers.

The exact regressions include more than 1.8 million bounded difference checks through row 512. Smaller runs test the stronger address-two filtration and mixed parity law. Their records specify precision, coefficient definitions, inputs and tested ranges.

The figures in this guide have a separate reproducibility record: `sources/generate_figures.py`, `figures/exact_figure_data.json`, and `figures/figure_manifest.json`. Its exact assertions validate the illustration data. Theorem-derived curves are explicitly recorded as such.

The new general transfer, mixed-step, factorial, and analytic extension has not been newly formalized in Lean. The rank-four ordinary-step and sharp-period results have inherited Lean coverage, as mapped in the companion. Mathematical audits in this package are internal reviews, not claims of external peer review.

15 Reading the conclusion at the right strength

The original quantified Nishibuchi conjecture is completed. The stronger dyadic result applies without a starting-row threshold, and the rank-four theory now has factorial precision, exact common-step laws for addresses zero and one, an infinite exact address-two family, optimal periods, and exact analytic radii for three addresses.

For a common step, the final congruence to use is

$$\Delta_H^k E^{(4,j)}(n) \equiv 0 \pmod{2^{v_2(k!)+kv_2(H)+A_j(k)}}, \quad n \geq 0,\ k, H \geq 1,$$

with

$$(A_0, A_1, A_2, A_3) = \big(k, k, \max\{k+4, 2k+\lfloor k/2 \rfloor + 1\}, k+2\big).$$

For different steps, replace $kv_2(H)$ by $\sum_i v_2(H_i)$ in the lower bound. The address-zero/one equality rule then depends on order parity and the number of even steps.

Statement	Final status
Nishibuchi Conjecture 3.9, all allowed branches	completed as the formal capstone
All-row dyadic mixed-difference bound	proved and audited
Rank-four factorial and paired bounds	proved and audited
All common-step exact valuations, $j = 0, 1$	proved at all rows and orders
Address-two exact valuations at power-of-two orders	proved for all positive step lists
Complete exact address-two formula at other orders	open
Complete exact higher-order address-three formula	open
Exact address-three analytic radius	open
Simple constant metric at every upper-half address	false, with exact witnesses retained
Separate Euler-tail universal simplicity problem	open as a distinct research program

The remaining questions do not weaken the completed conjecture. They specify the next boundary of the stronger theory. The main text gives the proofs and precise dependency statements, while the Lean guide supplies the formal coverage map and a reproducible route to its recorded checks.

16 Compute the reciprocal before estimating it

For width N and address j, the normalized source and its reciprocal are

$$G_{N,j}(t) = j! \sum_{n \ge 0} \frac{t^{Nn}}{(Nn+j)!}, \qquad G_{N,j}(t)^{-1} = \sum_{n \ge 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}.$$

Multiplying the two series and clearing the factorials gives an exact triangular recurrence:

$$C_{N,j}(0) = 1, \qquad C_{N,j}(n) = -\frac{\sum_{r=0}^{n-1} \binom{Nn+j}{Nr} C_{N,j}(r)}{\binom{Nn+j}{Nn}}.$$

address j	C(0)	C(1)	C(2)
0	1	-1	69
1	1	-1/5	121/45
2	1	-1/15	13/45
3	1	-1/35	59/1155

The table is for $N = 4$. The first two address-zero rows illustrate the computation without any valuation estimate:

$$C_{4,0}(1) = -1, \qquad C_{4,0}(2) = -\left(1 + \binom{8}{4}(-1)\right) = 69.$$

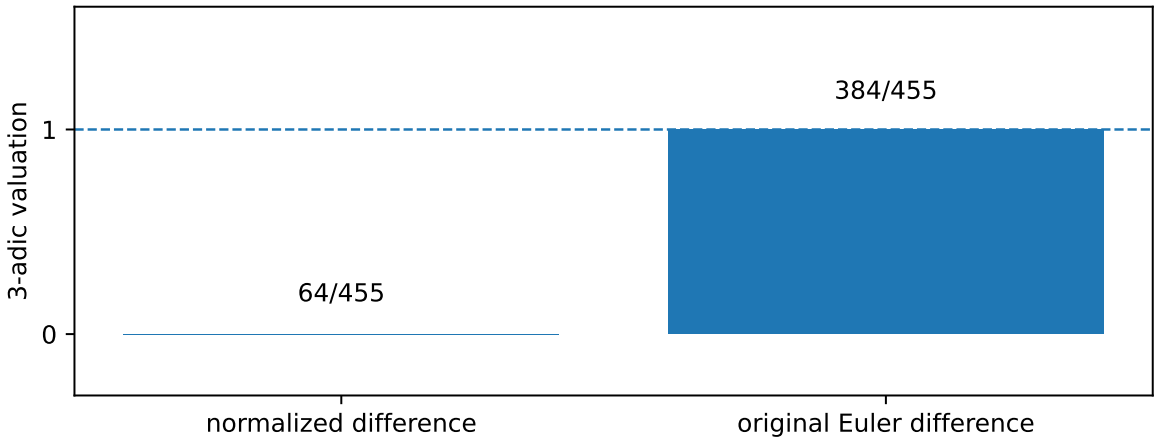
At address one, the same recurrence yields $C_{4,1}(1) = -1/5$ and $C_{4,1}(2) = 121/45$. These are exact rational coefficients, not rounded approximations. The second implementation in the supplied script inverts the ordinary power series first, then multiplies by $(Nn)!$. Agreement between the two constructions checks the factorial placement.

17 The normalization loss is a real boundary

The original Euler coefficient is $E_{Nn}^{(N,j)} = j!C_{N,j}(n)$. Multiplication by $j!$ changes the available p-adic precision. That change cannot be discarded outside the unit-factorial window.

Take $p = 3$, $N = 6$, $j = 3$, and compare rows one and two. Exact inversion gives

$$C_{6,3}(1) = -\frac{1}{84}, \quad C_{6,3}(2) = \frac{703}{5460}, \quad C_{6,3}(2) - C_{6,3}(1) = \frac{64}{455}.$$



Neither 64 nor 455 is divisible by three, so the normalized difference has valuation zero. The original coefficient difference is $3! \cdot 64/455 = 384/455$, of valuation one.

This is the counterexample behind the corrected factorial-loss statement. A bound for the original Euler normalization must lose $v_p(j!)$ when translated to $C_{N,j}$. For $0 \leq j < p$, that loss is zero. The distinction concerns which sequence the theorem measures, not a numerical weakness of the calculation.

18 Odd primes require the sign twist

The prime-power transfer controls a sign-twisted reciprocal sequence at odd primes. It must not be read as an ordinary-difference theorem for the untwisted coefficients.

At $N = p = 3$ and $j = 0$, put $A(n) = (-1)^n C_{3,0}(n)$. The first three rows are

row n	C at width 3, address 0	(-1)^n C(n)
0	1	1
1	-1	1
2	19	19

The ordinary first difference at row zero is $-1 - 1 = -2$. It is a 3-adic unit, so an asserted lower bound of one for that difference would be false. The twisted first difference is zero, whose mathematical valuation is $+\infty$. At the next row, the twisted difference is $19 - 1 = 18$, of valuation two.

Equivalently, the untwisted sequence is acted on by $T_H = S^H - (-1)^H$, where S advances one row. At odd step one this is $S + 1$, not $S - 1$. The zero in the example is handled by a zero-safe divisibility statement rather than a finite valuation equality. The explicit sign and the zero convention are both part of the theorem's interface.

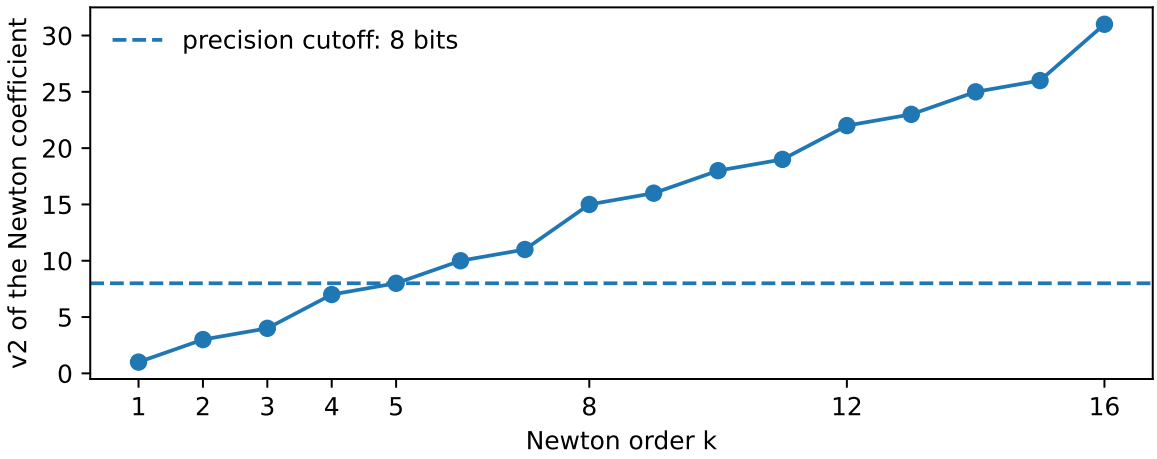
19 Newton coordinates turn precision into a finite computation

Let $b_k = \Delta_1^k C_{4,0}(0)$. The first coefficients obtained from the difference triangle are

$$(b_0, b_1, b_2, b_3, b_4) = (1, -2, 72, -33872, 60511872).$$

The exact Newton identity recovers each nonnegative integer row:

$$C_{4,0}(n) = \sum_{k=0}^{n} b_k \binom{n}{k}.$$



For this address the proved law is $v_2(b_k) = 2k - s_2(k)$ for $k \geq 1$. Every $k \geq 5$ therefore has $v_2(b_k) \geq 8$. Since $\binom{n}{k}$ is integral, those terms vanish modulo 256. Thus, for every $n \geq 0$,

$$C_{4,0}(n) \equiv 1 + 254\binom{n}{1} + 72\binom{n}{2} + 176\binom{n}{3} + 128\binom{n}{4} \pmod{256}.$$

The uniform truncation follows from the coefficient theorem. The script separately checks the exact Newton reconstruction and this reduction for rows 0 through 32. Those finite checks validate the worked arithmetic without replacing the theorem.

20 A compatible root is built one digit at a time

Take $F(x) = x^2 - 7$ over the 3-adic integers and start with $x_1 = 1$ modulo three. If $F(x_L) = 3^L r_L$, the next representative has the form $x_{L+1} = x_L + 3^L t_L$. Its correcting digit satisfies

$$r_L + 2x_L t_L \equiv 0 \pmod 3.$$

L	3^L	current x	next digit	next x
1	3	1	1	4
2	9	4	1	13
3	27	13	0	13
4	81	13	2	175
5	243	175	0	175

For example, $F(1)/3 = -2$ and $2x_1 = 2$, so the next digit is one and $x_2 = 4$. Then $F(4)/9 = 1$, again giving digit one and $x_3 = 13$. At the next stage the residual is divisible by three, so digit zero is correct. A zero digit is not a stalled construction.

Every center remains congruent to one modulo three. Hence $2x_L$ stays a unit, and the correction equation has a unique digit at every stage. The representatives are compatible and their increments are divisible by 3^L. Completeness gives a limit, and continuity of the polynomial gives a root of F. The finite table exhibits the construction. The unit derivative is what supports indefinite continuation.

21 A singular residual can stop a lift

Now take $F(x) = x^2 - 3$ at the residue $x = 0$ modulo three. The residual is $r = F(0)/3 = -1$ and the Jacobian is zero modulo three. The next-digit equation would require

$$-1 + 0 \cdot t \equiv 0 \pmod 3,$$

which has no solution.

digit t	candidate y	y^2 - 3	residue mod 9
0	0	-3	6
1	3	6	6
2	6	33	6

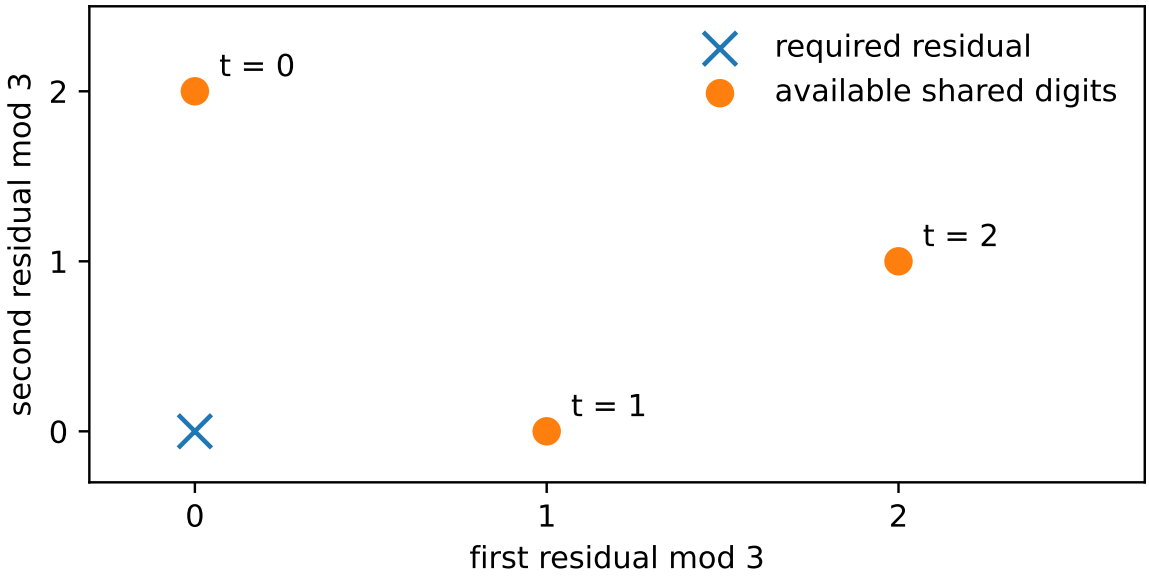
The three representatives of the old root class modulo nine all leave residual six modulo nine. Thus the root class exists at precision three but does not lift to precision nine. This is an exact obstruction, not failure of an iterative search.

A singular Jacobian alone does not imply failure. For $F(x) = x^2$ at the same center, the residual is zero and all three candidate digits lift to roots modulo nine. The obstruction is the residual class in the cokernel of the actual Jacobian. It is this class, rather than the rank by itself, that decides whether a next lift exists.

22 Separate corrections need not be one shared correction

Consider the two-component system $F(x) = (x, x-3)$ at $x = 0$, with $q = 3$ and $L = 1$. Both components vanish modulo three. After writing a candidate as $y = 3t$, its residual vector modulo three is

$$\bar{r} + J\bar{t} = (0,2) + (t,t) = (t, t-1).$$



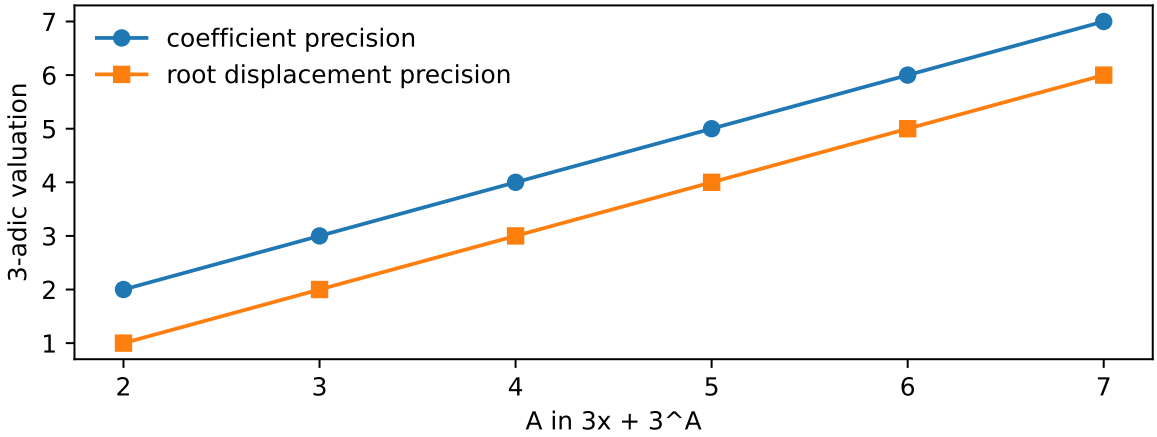
The first coordinate can be corrected by $t = 0$. The second can be corrected by $t = 1$. No digit corrects both. The available residuals are $(0,2)$, $(1,0)$, and $(2,1)$, none of which is $(0,0)$.

The combined Jacobian maps t to the diagonal vector (t,t). Its image is smaller than the product of its coordinate images. Checking the two projected obstructions independently therefore loses the compatibility condition. The common-correction theorem retains the full residual vector and the shared digit space, which is exactly the information needed to detect this failure.

23 A derivative weight can cost one digit of root precision

Compare the linear equations $F(x) = 3x = 0$ and $G_A(x) = 3x + 3^A = 0$, where $A \geq 2$. Their coefficient difference has 3-adic valuation A, but their roots differ by

$$x_A - 0 = -3^{A-1}, \qquad v_3(x_A) = A - 1.$$



The derivative has valuation one. Solving the correction equation divides by three, so one digit of coefficient precision is lost in root displacement. The calculation is exact for every displayed A and also follows immediately from the two linear equations.

This example explains why weighted lifting and deformation estimates track a Jacobian loss rather than silently treating every invertible derivative over $\mathbb{Q}_p$ as a unit over $\mathbb{Z}_p$. A valuation-zero derivative would not incur this particular loss. Nonlinear systems additionally require their stated neighborhood, residual, and right-inverse hypotheses.

The eight examples use exact rational and integer arithmetic. Their recorded finite ranges and computation scripts are supplied with the sources. The atlas adds no Lean endpoint and changes no formal claim in the companion.

CHAPTER X

Universal Residual Seeds

Deformation forests, transport categories, and conditional system completion

The congruence study is followed by a framework for residual seeds and compatible arithmetic descriptions.

Residual seeds, deformation forests and transport categories organize conditional system completion. Supplied arithmetic and carrier data remain explicit inputs.

IN THIS CHAPTER

Mathematical text 1170

Visual studies 1227

Formal companion 2350

Sources and references 2690

Scope

The target family and recovered carrier are independent existential outputs. Identifying them and proving nonzero occupancy require additional premises. Source families, carrier presentations and local comparisons remain supplied.

The next chapter studies local packet geometry and the remaining obligations of arithmetic realization.

Reading scope

Residual forests describe deformation above a fixed residual root. The later constructions compare compatible families and recover summands inside supplied carriers. Universal companion existence, carrier existence, and nonzero occupancy require additional inputs. Section D assembles the conditional result and states those inputs explicitly.

Abstract

A residual representation is a zeroth-order coordinate. All positive-depth perturbations above it may fill its formal deformation neighborhood, but they cannot move into a different residual class. This chapter develops the corresponding universal object: a residual forest whose roots are residual representations, whose finite levels are lifts modulo powers of a uniformizer, and whose infinite branches are complete characteristic-zero objects. We prove the residual partition, no-crossing, inverse-limit reconstruction, finite-branch compactness, exact smooth and singular branch counts, packet decomposition of tangent and obstruction cohomology, a directed minimum-seed theorem, rational seed-zeta formulas, and finite companion-forest completion. The coefficient prime two is represented by a weighted forest in which characteristic-zero packets may merge modulo two and become visible only at later depth. We then define a prime-independent system seed through a common Frobenius spine, prove a compatible-carrier summand theorem under finite Frobenius separation, and reduce finite occupancy in a fixed semisimple carrier to the nonnegative integer trace of an exact joint projector. The chapter does not prove that every admissible residual companion exists or that every finite Hecke packet is occupied. Those two existence statements remain the exact horizontal boundary beyond vertical deformation theory.

A route through the chapter

The central distinction is between a residual root, a coherent branch and an occupied packet in a carrier. Sections 2–7 develop the finite and formal machinery. Sections 8–11 assemble its arithmetic uses under stated hypotheses. The residue-counting arguments appear in Appendix C. Appendix D states the conditional occupied-system completion theorem. The visual studies provide worked examples of the same constructions.

Read for	Follow this route
The mathematical core	Residual partition, reconstruction, smooth and singular counts, cohomology packets and directed seed rank.
The arithmetic assembly	Common coefficient data, finite Frobenius separation, explicit carrier summands and inverse-limit-closed profiles.
The formal contract	The formal companion explains the terminal theorem and its three direct dependency modules, with a statement register and a guide to each hypothesis.
The finite examples	Plates X.1–X.30 show residue tables, projector tests, countermodels and norm-character examples.

Two outputs require two sets of inputs. The terminal theorem proves that a transported target family exists and that the declared summand can be recovered inside a supplied compatible carrier. It does not identify those two objects. An application that needs such an identification must supply its comparison. Likewise, zero multiplicities are permitted by the formal assembly, so a nonzero conclusion retains a separate nonzero premise.

Mathematical and formal scope. The final source is over the rational absolute Galois group. Carrier data, constituent comparisons and a completed source family are explicit inputs. The formal companion distinguishes the amended conditional statements from stronger existence questions and identifies their recorded verification. Earlier source contracts remain available in the accompanying research records.

1 The universal seed at residual depth

The earlier chapters treat a seed as a primitive object together with an operator grammar, a packet system, a topology, and an exact reconstruction law. In arithmetic deformation problems, the primitive coordinate has two layers:

$$\boxed{\text{residual root} \quad + \quad \text{positive-depth deformation coordinates.}}$$

The distinction is structural. A correction divisible by the uniformizer changes higher digits but leaves the residual digit fixed. Therefore the correct universal arithmetic seed cannot be one residual representation unless the admissible residual space has only one point. It must contain every residual root and the complete formal neighborhood above each root.

Let $\mathcal{O}$ be a complete discrete valuation ring with uniformizer ϖ, maximal ideal (ϖ), residue field

$$k = \mathcal{O}/(\varpi), \qquad |k| = q,$$

and fraction field K. When a characteristic-zero conclusion is intended, assume additionally that $\operatorname{char} K = 0$. The general counting statements do not require that restriction. Let $\mathcal{M}$ be a moduli functor on complete local Noetherian $\mathcal{O}$-algebras with residue field k, equipped with a well-defined reduction map

$$\operatorname{red}_A : \mathcal{M}(A) \longrightarrow \mathcal{M}(k).$$

For a Galois representation this means either that a framed integral model has been fixed or that the residual semisimplified class is used.

Definition 1.1 (Residual seed system)**.** For $\bar{x} \in \mathcal{M}(k)$, define the residual fiber

$$\mathcal{M}_{\bar{x}}(A) = \{x \in \mathcal{M}(A) : \operatorname{red}_A(x) = \bar{x}\}.$$

The point $\bar{x}$ is the residual root. The family of all nonempty fibers is the residual special fiber, and the formal deformation problems above them form the residual forest.

For a Galois problem the principal example is

$$\bar{\rho} : G_{F,S} \longrightarrow \mathrm{GL}_n(k),$$

with a deformation functor $\mathrm{Def}_{\bar{\rho}}$ cut out by a determinant and selected local conditions.

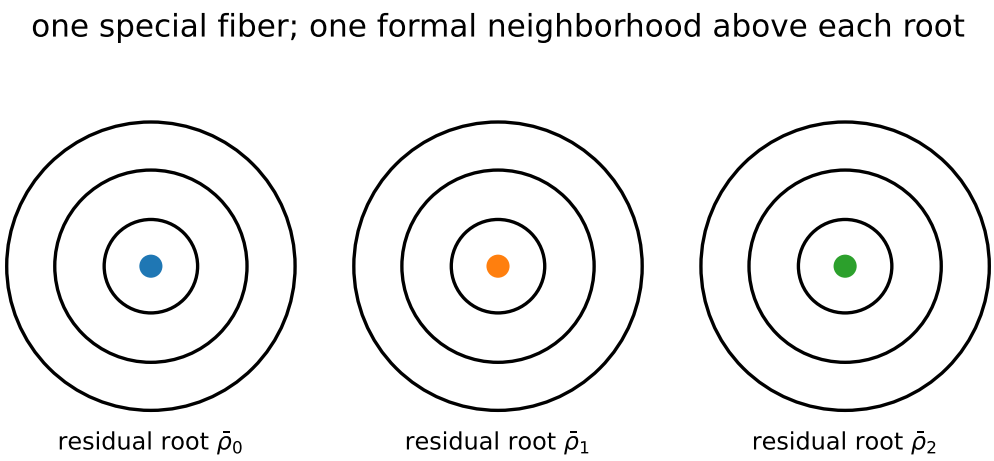


Figure 1. The universal residual seed is not one point. It is the special fiber together with one complete formal neighborhood above each admissible root.

1.1 What the chapter is designed to separate

Three statements are often compressed into one sentence even though they have different logical content.

1. **Vertical completeness:** every compatible perturbation stream above a fixed root reconstructs a point of its complete deformation space.
2. **Horizontal companion existence:** every required residual root, including roots at other coefficient primes, actually exists.
3. **Occupancy:** the resulting arithmetic packet occurs in a nonzero global carrier of the required type.

The first belongs to deformation and inverse-limit theory. The second is a residual companion problem. The third is an arithmetic or automorphic nonvanishing problem. Neither of the last two follows from arbitrarily deep perturbation above one already occupied root.

2 Residual partition and no crossing

Theorem 2.1 (Residual partition)**.** *For every coefficient algebra A,*

$$\boxed{\mathcal{M}(A) = \bigsqcup_{\bar{x} \in \mathcal{M}(k)} \mathcal{M}_{\bar{x}}(A).}$$

Proof. Every A-valued object has one reduced object $\mathrm{red}_A(x) \in \mathcal{M}(k)$, so the union is exhaustive. Two distinct fibers cannot intersect because an object cannot have two different reductions. Hence the union is disjoint. □

Theorem 2.2 (No crossing under positive-depth perturbation)**.** *Suppose $x, y \in \mathcal{M}(\mathcal{O})$ are related by a deformation operation that is congruent to the identity modulo ϖ. Then*

$$\mathrm{red}(x) = \mathrm{red}(y).$$

Every finite composition remains in one residual fiber. If an infinite perturbation stream converges in the ϖ-adic topology, its limit remains in the same fiber. The existence of that limit is an explicit hypothesis, not a consequence of positive depth alone.

Proof. A correction in $\varpi\mathcal{O}$ vanishes after reduction modulo ϖ. Thus one operation preserves the residual root, and induction proves the finite statement. For a convergent stream, convergence at the first finite precision makes its reduction eventually equal to that of its limit. Every term already has the initial reduction, so the limit has that reduction too. Completeness applies to Cauchy streams. It does not make every root-preserving stream Cauchy. For example, $0, 3, 0, 3, \ldots$ has constant reduction modulo 3 but no limit modulo 9. □

Corollary 2.3 (Zeroth-order obstruction)**.** *If two admissible objects have different residual roots, no sequence of genuinely positive-depth perturbations beginning at one can reach the other. A horizontal transport between their roots is a zeroth-order operation, not a deeper lift.*

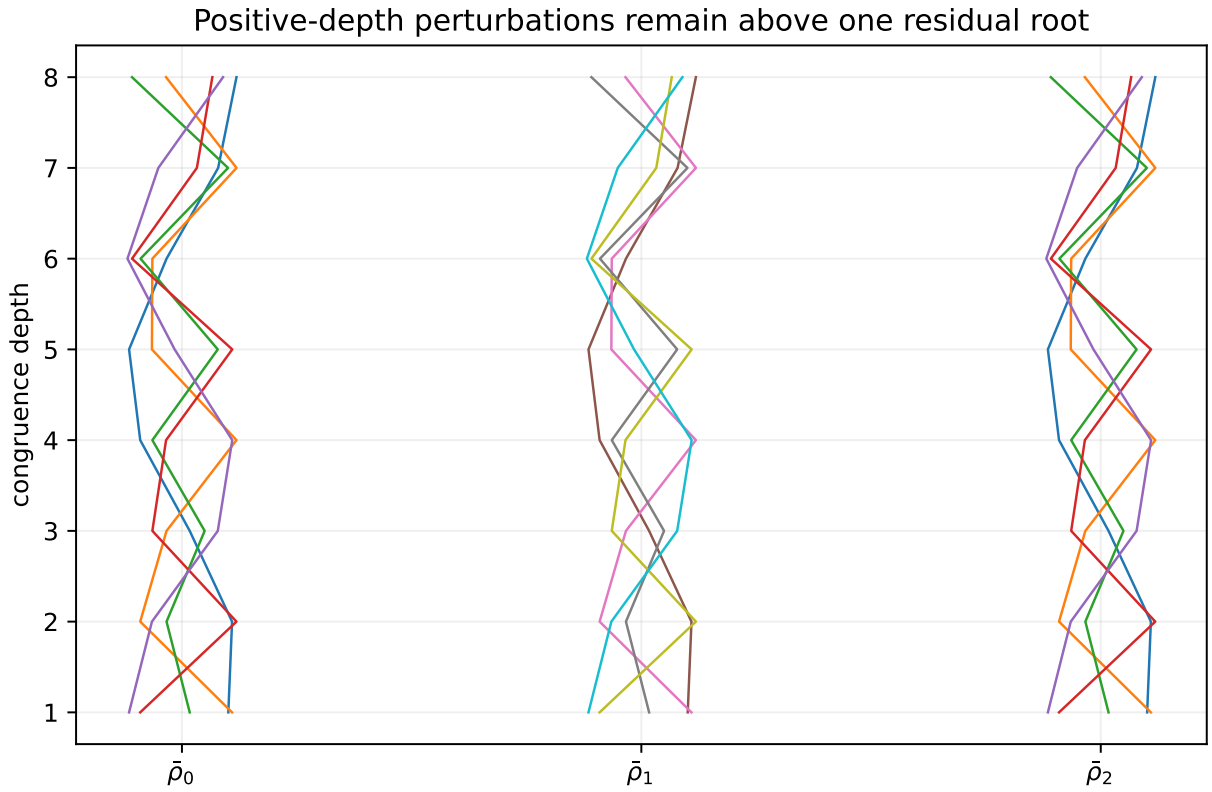


Figure 2. Each perturbation stream remains above its initial residual root. Refinement adds digits vertically. It does not move a branch horizontally.

Numerical Certificate 2.4 (Exact residual partition)**.** The finite implementation partitions the integers $0, \dots, 8$ into the three classes modulo 3:

$$\{0,3,6\}, \qquad \{1,4,7\}, \qquad \{2,5,8\}.$$

Every tested correction of the form $x \mapsto x + ph$ preserves the residual class exactly.

3 Lift forests and inverse-limit reconstruction

Fix $\bar{x} \in \mathcal{M}(k)$. For $m \geq 1$, set

$$X_m(\bar{x}) = \mathcal{M}_{\bar{x}}(\mathcal{O}/\varpi^m).$$

Reduction gives transition maps

$$r_{m+1,m} : X_{m+1}(\bar{x}) \longrightarrow X_m(\bar{x}).$$

A compatible branch is a sequence $(x_m)_{m\geq 1}$ satisfying

$$r_{m+1,m}(x_{m+1}) = x_m.$$

Classical Input 3.1 (Represented complete deformation problems)**.** When a deformation functor is represented by a complete local ring $R_{\bar{x}}$, its $\mathcal{O}$-valued points are continuous local maps $R_{\bar{x}} \to \mathcal{O}$. Completeness identifies such maps with compatible maps to every quotient $\mathcal{O}/\varpi^m$.

Theorem 3.2 (Formal inverse-limit reconstruction)**.** *Assume $\mathcal{M}_{\bar{x}}$ is represented by a complete local $\mathcal{O}$-algebra $R_{\bar{x}}$. Then reduction induces a natural bijection*

$$\boxed{\mathcal{M}_{\bar{x}}(\mathcal{O}) \cong \varprojlim_m \mathcal{M}_{\bar{x}}(\mathcal{O}/\varpi^m).}$$

Proof. A continuous local homomorphism $R_{\bar{x}} \to \mathcal{O}$ reduces to a compatible family of homomorphisms into $\mathcal{O}/\varpi^m$. Conversely, compatible finite-level maps determine a map

$$R_{\bar{x}} \longrightarrow \varprojlim_m \mathcal{O}/\varpi^m \cong \mathcal{O}.$$

The two constructions are inverse because equality in a separated complete ring is detected modulo every power of its maximal ideal. □

Theorem 3.3 (Finite-branch compactness)**.** *Suppose every $X_m(\bar{x})$ is finite and nonempty and every node has finitely many children. Then the lift tree above $\bar{x}$ contains an infinite compatible branch.*

Proof. The tree is infinite because it has a nonempty level at every depth, and it is finitely branching by hypothesis. König's lemma gives an infinite path. The edge relation is reduction, so that path is a compatible branch. □

Surjectivity of every transition map is sufficient but not necessary. Finite branching and nonempty arbitrarily deep levels are the compactness mechanism.

Proposition 3.4 (Smooth formal branch count)**.** *If*

$$R_{\bar{x}} \cong \mathcal{O}[[t_1, \dots, t_d]] \quad \textit{as complete local } \mathcal{O}\textit{-algebras},$$

then

$$\boxed{\#X_m(\bar{x}) = q^{d(m-1)}.}$$

Fix a uniformizer and a section $s : k \to \mathcal{O}$ of reduction, with $s(0) = 0$. Every $\mathcal{O}$-valued branch is encoded by d positive-depth digit streams

$$t_i = s(a_{i,1})\varpi + s(a_{i,2})\varpi^2 + \cdots, \qquad a_{i,j} \in k,$$

Proof. A local map sends each parameter into the image of (ϖ) in $\mathcal{O}/\varpi^m$. Conversely, every tuple in that image is nilpotent and defines a coefficient-preserving evaluation homomorphism. This homomorphism is local. To prove uniqueness without assuming continuity of an arbitrary competing homomorphism, observe in one variable that a homomorphism killing X^N only sees the finite polynomial truncation modulo X^N. Induction over the finite number of variables gives uniqueness in all dimensions. The actual local maps are therefore in bijection with the full parameter tuple set. Its cardinality is $q^{d(m-1)}$, including the zero-dimensional chart.

For integral maps, reduce an arbitrary tuple in $(\varpi)\mathcal{O}$ at every precision, evaluate there, and use the represented-map reconstruction theorem. The finite evaluations are compatible, the reconstructed algebra map is local, and every competing local map has the same finite reductions. Separatedness gives uniqueness. Local maps are continuous in the maximal-ideal adic topologies. This establishes the integral parameter classification rather than assuming it from the finite count.

For the digit claim, subtract the chosen representative of the residue and divide the remainder by the uniformizer. Iterating gives unique digits. Conversely, the partial sums of any digit stream form a compatible Cauchy family and reconstruct uniquely by completeness and separatedness. More precisely, for each parameter there is a unique stream $(a_{i,j})_{j\geq 1}$ such that

$$\varpi^{n+1} \mid t_i - \sum_{j=1}^{n} s(a_{i,j})\varpi^j \qquad (n \geq 0).$$

The section is a set-theoretic section of reduction, not an asserted embedding of residue fields into the coefficient ring. Finally, a supplied local $\mathcal{O}$-algebra isomorphism from $R_{\bar{x}}$ transports both the map classification and the count. □

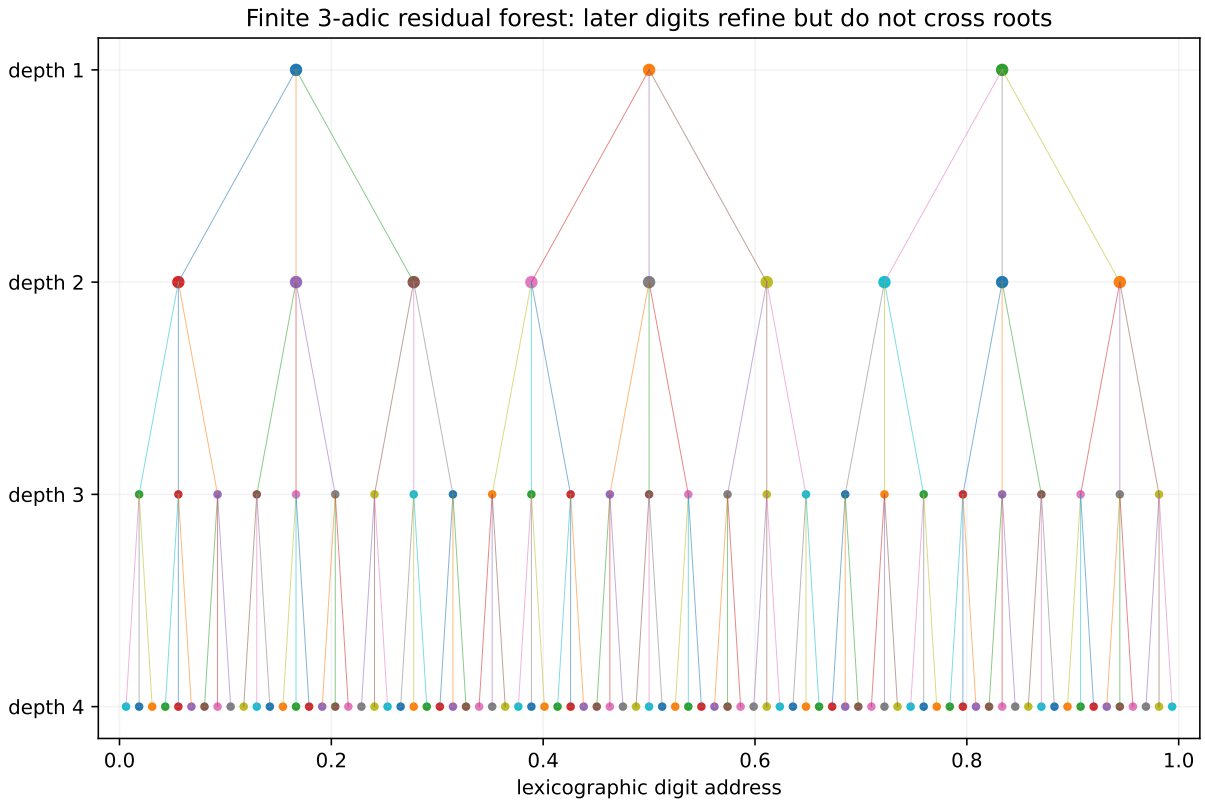


Figure 3. The first four levels of a full 3-adic digit forest. A path is a compatible finite-depth perturbation stream.

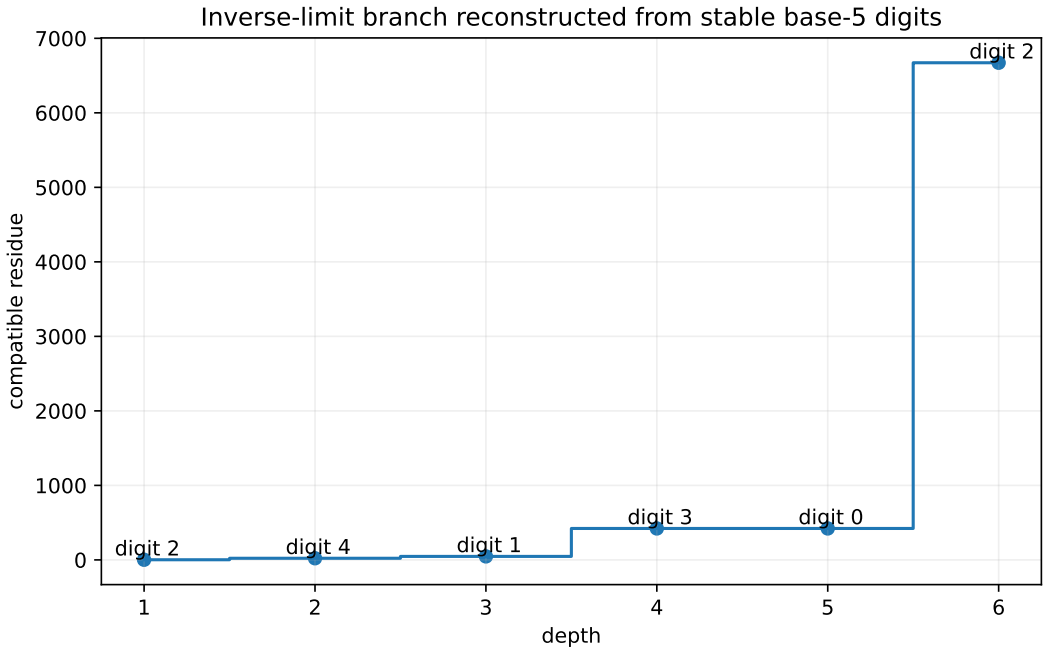


Figure 4. A compatible base-five residue tower reconstructs the digit stream $(2, 4, 1, 3, 0, 2)$.

4 Tangent and obstruction packets

For a residual representation $\bar{\rho}$, first-order deformations and lifting obstructions are classically controlled by cohomology of the adjoint module. The projector calculus does not replace that theory. It resolves the controlling complex into invariant packets when suitable projectors exist.

Classical Input 4.1 (Tangent and obstruction spaces)**.** For a framed representation problem, first-order lifts are described by $Z^1(G, \operatorname{ad}\bar{\rho})$. For an unframed Schur problem, tangent directions are described by $H^1(G, \operatorname{ad}\bar{\rho})$. Relations in a presentation of a deformation ring map into $H^2(G, \operatorname{ad}\bar{\rho})$, which contains the obstruction classes.

Let

$$C^0 \xrightarrow{d_0} C^1 \xrightarrow{d_1} C^2 \to \cdots$$

be a cochain complex over a field. Suppose a finite family of projectors $P_j^r : C^r \to C^r$ satisfies

$$\sum_j P_j^r = I, \qquad P_i^r P_j^r = \delta_{ij} P_j^r, \qquad d_r P_j^r = P_j^{r+1} d_r.$$

Theorem 4.2 (Cohomology packet decomposition)**.** *Under these hypotheses, each packet is a subcomplex and*

$$\boxed{H^r(C^\bullet) \cong \bigoplus_j H^r(P_j C^\bullet).}$$

In particular,

$$H^1 = \bigoplus_j H^1_j, \qquad H^2 = \bigoplus_j H^2_j,$$

so tangent and obstruction spaces reconstruct exactly from their packet components.

Proof. For each packet, use the actual image submodules $C_j^r = \operatorname{im} P_j^r$. Commutation with the differential makes these a subcomplex. Write

$$Z^r = \ker d_r, \qquad B^r = \operatorname{im} d_{r-1} \subseteq Z^r, \qquad H^r = Z^r / B^r,$$

and define Z_j^r, B_j^r, H_j^r using the restricted differentials of $C_j^\bullet$. The inclusion $i_j : C_j^\bullet \to C^\bullet$ and the image-valued projection $p_j : C^\bullet \to C_j^\bullet$ are cochain maps, so they induce maps on these genuine quotients.

For $i \neq j$, orthogonality gives $H(p_i)H(i_j) = 0$. Idempotence gives $H(p_j)H(i_j) = I$, and completeness gives $\sum_j H(i_j)H(p_j) = I$. Thus the maps

$$[z] \longmapsto ([P_j^r z])_j, \qquad ([z_j])_j \longmapsto \left[\sum_j i_j z_j\right]$$

are inverse linear maps. The finite product equals the direct sum, proving the assertion. In the formalization, both sides are also identified with Mathlib's standard homology objects of the genuine graded complexes. □

Scope of the splitting. The algebraic proof works over any commutative ring and does not divide by 2, so it applies in characteristic two when the stated integral chain projectors exist. It does not construct

those projectors. The identification of a particular Galois tangent or obstruction problem with this cohomology remains the preceding classical input. The splitting detects vanishing:

$$H^r(C^\bullet) = 0 \quad \Longleftrightarrow \quad H^r(C_j^\bullet) = 0 \text{ for every } j.$$

It does not force either side to vanish or decouple nonlinear deformation equations. Moreover, commutation with the differential is essential: complementary idempotents on the underlying vector spaces need not preserve cocycles or coboundaries.

Naturality. If a cochain map $F : C^\bullet \to D^\bullet$ intertwines the supplied projectors, it restricts to maps $F_j : C_j^\bullet \to D_j^\bullet$. The direct-sum decomposition commutes with $H^r(F)$ and the family $H^r(F_j)$. This is checked using representatives in the actual quotients, independently of any dimension computation.

Remark 4.3 (Odd and dyadic nonlinear equations). At odd residue characteristic, a differential graded Lie model may be written in Maurer–Cartan form

$$d\eta + \frac{1}{2}[\eta, \eta] = 0,$$

and a cyclic grading converts this into exact packet convolutions. At residue characteristic two, division by 2 is not integral. A quadratic or divided-square refinement must be retained. This is one reason the dyadic residual forest is weighted rather than obtained by formal substitution into the odd-prime formulas.

Numerical Certificate 4.4 (Finite packet complex)**.** The exact modular test uses two invariant subcomplexes over $\mathbf{F}_5$. Their cohomology dimensions are

$$(0, 0, 0), \qquad (1, 2, 1),$$

and the reconstructed total is

$$\dim(H^0, H^1, H^2) = (1, 2, 1).$$

The packet-sum residual is zero.

An explicit realization of the finite pattern. A separate finite-complex implementation uses

$$\mathbf{F}_5^2 \xrightarrow{d_0} \mathbf{F}_5^4 \xrightarrow{d_1} \mathbf{F}_5^2, \quad d_0(a, b) = (a, 0, 0, 0), \quad d_1(a, b, c, d) = (b, 0).$$

The first packet is the exact complex with dimensions $(1, 2, 1)$, and the second packet has zero differentials and dimensions $(1, 2, 1)$. The actual middle cohomology is linearly isomorphic to $\mathbf{F}_5^2$, which is checked in Lean. Independent exhaustive coset tests check all three cohomology degrees after changes of basis. This realizes the displayed dimension pattern. Its formal check is distinct from the original companion implementation.

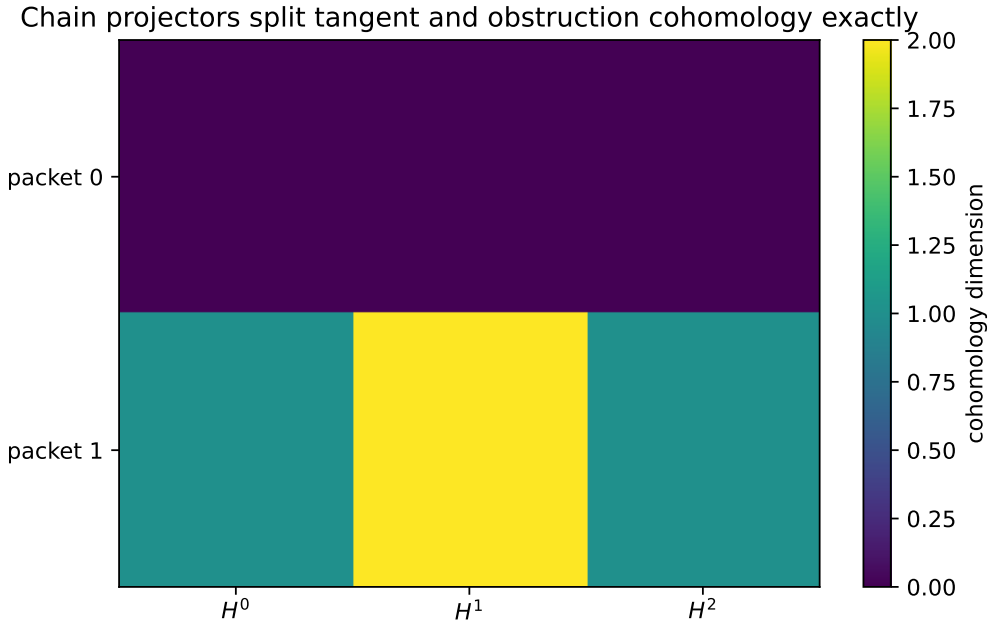


Figure 5. A finite chain-projector model. Tangent and obstruction dimensions are carried by one packet and reconstruct exactly.

5 The weighted dyadic residual forest

At an odd prime, distinct sign characters remain distinct in the residue field. At the coefficient prime two,

$$1 \equiv -1 \pmod 2.$$

Thus distinct characteristic-zero packets can share one residual root and separate only after more precision is retained.

Definition 5.1 (Visibility weight)**.** For two integral packet coordinates $a, b \in \mathcal{O}$, define their separation depth

$$w(a,b) = \min\{m \geq 1 : a \not\equiv b \pmod{\varpi^m}\},$$

when such an m exists. A dyadic packet decomposition is weighted by these first visibility depths and by the valuations of any denominators required by its rational projectors.

Example 5.2 (The two sign branches)**.** The characters $\chi_+(g) = 1$ and $\chi_-(g) = -1$ have the same reduction modulo 2, but

$$1 \not\equiv -1 \pmod 4.$$

Their first separation depth is therefore 2. One mod-two root contains two branches that become distinct at the next level.

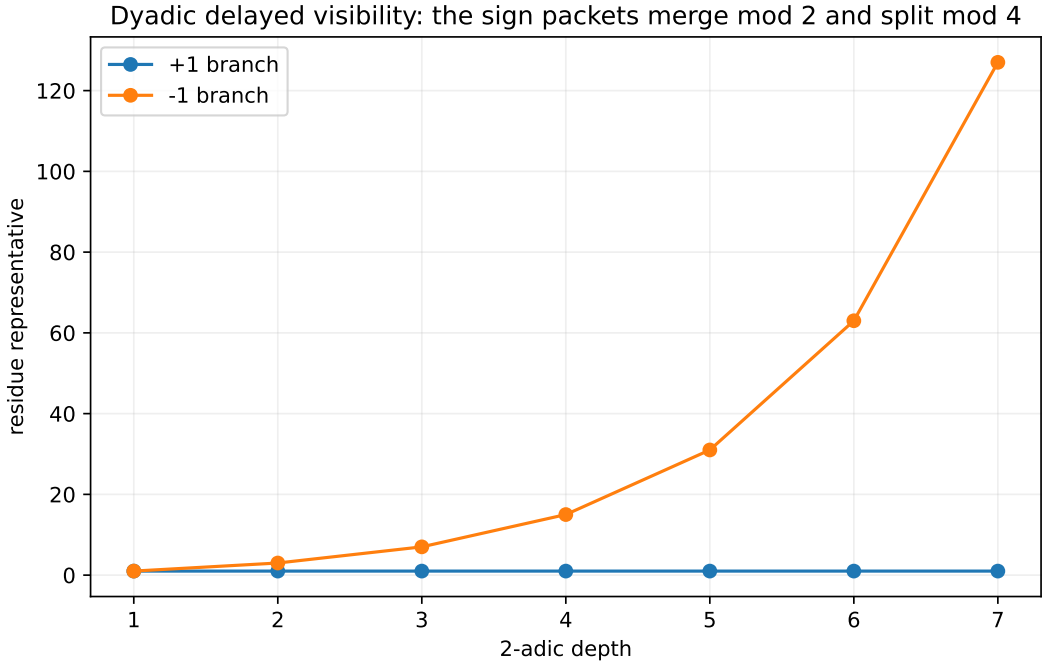


Figure 6. The sign packets merge at residual depth and separate at depth two. This is a genuine change in integral coordinates, not a numerical plotting effect.

A simple weighted formal model already captures delayed visibility. Let $w_i \geq 1$ and constrain

$$t_i \in (\varpi^{w_i}\mathcal{O} + \varpi^m\mathcal{O})/\varpi^m\mathcal{O}.$$

Then coordinate i is invisible until $m > w_i$.

Proposition 5.3 (Weighted smooth-chart count). *For weights $\boldsymbol{w} = (w_1, \ldots, w_d)$,*

$$\boxed{N_{\boldsymbol{w}}(m) = q^{\sum_{i=1}^{d} \max(m-w_i,0)}.}$$

Proof. The image of $\varpi^{w_i}\mathcal{O}$ in $\mathcal{O}/\varpi^m$ has one element when $m \leq w_i$, and q^{m-w_i} elements when $m > w_i$. Multiplying the independent coordinate counts gives the formula. □

This model does not assert that every dyadic Galois deformation ring is a weighted formal ball. It gives the exact local counting law once such weighted coordinates or a Smith-normal-form chart have been proved.

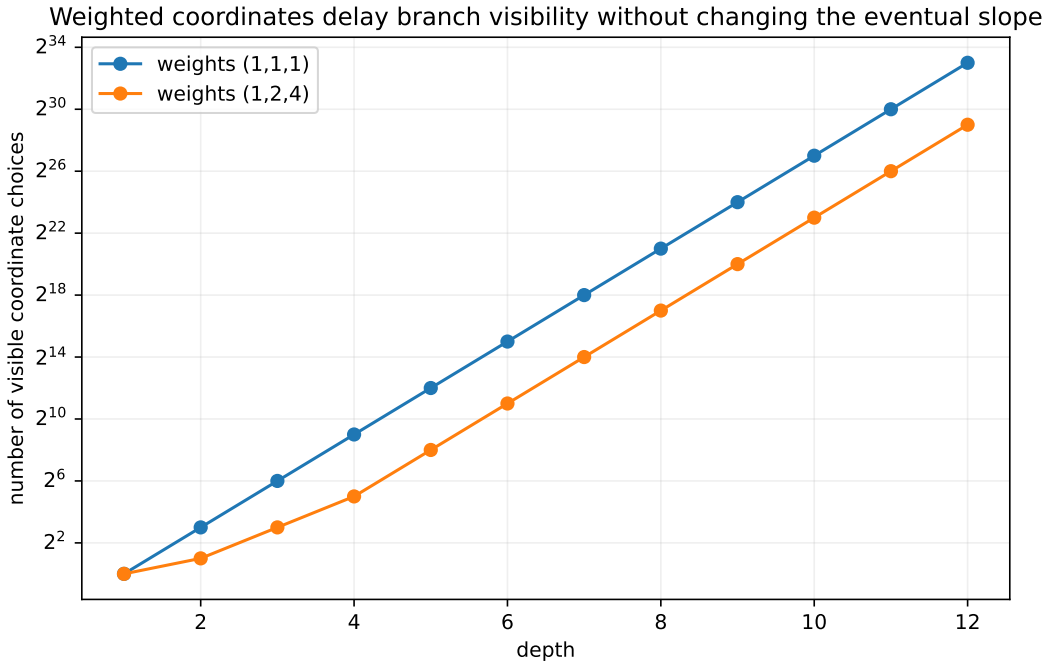


Figure 7. Weights delay finite-depth visibility. Once all coordinates are active, the asymptotic slope equals the ordinary formal dimension.

6 Seed zeta functions and branching dimensions

Finite-depth counts record more than the existence of a branch. They distinguish smooth, weighted, and singular local geometry.

Definition 6.1 (Seed zeta function)**.** For a residual root $\bar{x}$ with finite levels, define

$$\boxed{Z_{\bar{x}}(T) = \sum_{m \geq 1} \#X_m(\bar{x}) T^m.}$$

When a weighted chart is specified, write $Z_{\boldsymbol{w}}(T)$.

Theorem 6.2 (Smooth seed zeta function)**.** *For a smooth formal component of dimension d,*

$$\boxed{Z_d(T) = \frac{T}{1 - q^d T}.}$$

Proof. By the proved formal-model map-to-parameter equivalence, $\#X_m = q^{d(m-1)}$ counts all the actual local coefficient-algebra maps. Summing the geometric series gives the identity. The checked statement is an equality of infinite formal power series, with zero constant coefficient. □

Theorem 6.3 (Weighted seed zeta function)**.** *Let $1 \leq w_1 \leq \cdots \leq w_d$, put $W = \sum_i w_i$, and let*

$$a_m = q^{\sum_i \max(m - w_i, 0)}.$$

Then

$$\boxed{Z_{\boldsymbol{w}}(T) = \sum_{m=1}^{w_d} a_m T^m + q^{-W} \frac{(q^d T)^{w_d+1}}{1 - q^d T}.}$$

In particular, every weighted smooth chart has a rational seed zeta function.

Proof. The finite sum records the depths before all coordinates are active. For $m \geq w_d + 1$,

$$a_m = q^{dm-W},$$

so the remaining terms form one geometric series. □

Theorem 6.4 (Singular power seed)**.** *For integers $e \geq 1$ and $m \geq 1$, in $\mathcal{O}/\varpi^m$ the equation*

$$x^e = 0$$

has exactly

$$\boxed{q^{m-\lceil m/e \rceil}}$$

solutions. Its seed zeta function is

$$\boxed{Z_e(T) = \frac{\sum_{r=1}^{e} q^{r-1} T^r}{1 - q^{e-1} T^e}.}$$

Proof. The congruence is equivalent to

$$e\, v_\varpi(x) \geq m,$$

so x must lie in $\varpi^{\lceil m/e \rceil}\mathcal{O}/\varpi^m$, which has the stated cardinality. Write $m = ek + r$, $1 \leq r \leq e$. The exponent becomes $(e-1)k + r - 1$. Summing over $k \geq 0$ and $1 \leq r \leq e$ gives the rational expression. □

Definition 6.5 (Branching dimension)**.** When the limit exists, define the finite-level counting exponent

$$\dim_{\mathrm{seed}}(\bar{x}) = \lim_{m \to \infty} \frac{\log_q \#X_m(\bar{x})}{m}.$$

A limsup may be used in general.

Corollary 6.6 (Fractional singular dimension)**.** *For $x^e = 0$,*

$$\boxed{\dim_{\mathrm{seed}} = 1 - \frac{1}{e}.}$$

In particular, $x^2 = 0$ has branching dimension $1/2$.

Counting exponent versus compatible branches. *The preceding exponent counts all finite-level solutions, including nodes which cannot lift indefinitely. It is not the cardinality or dimension of the space of infinite branches. In fact every compatible tower of solutions to $x^e = 0$, for $e \geq 1$, is the zero tower. For each m, a solution at depth em lies in $\varpi^m\mathcal{O}/\varpi^{em}$. Its reduction at depth m is therefore zero. Compatibility proves the assertion at every depth. The formal companion identifies proofs of both this observation and the real logarithmic limit.*

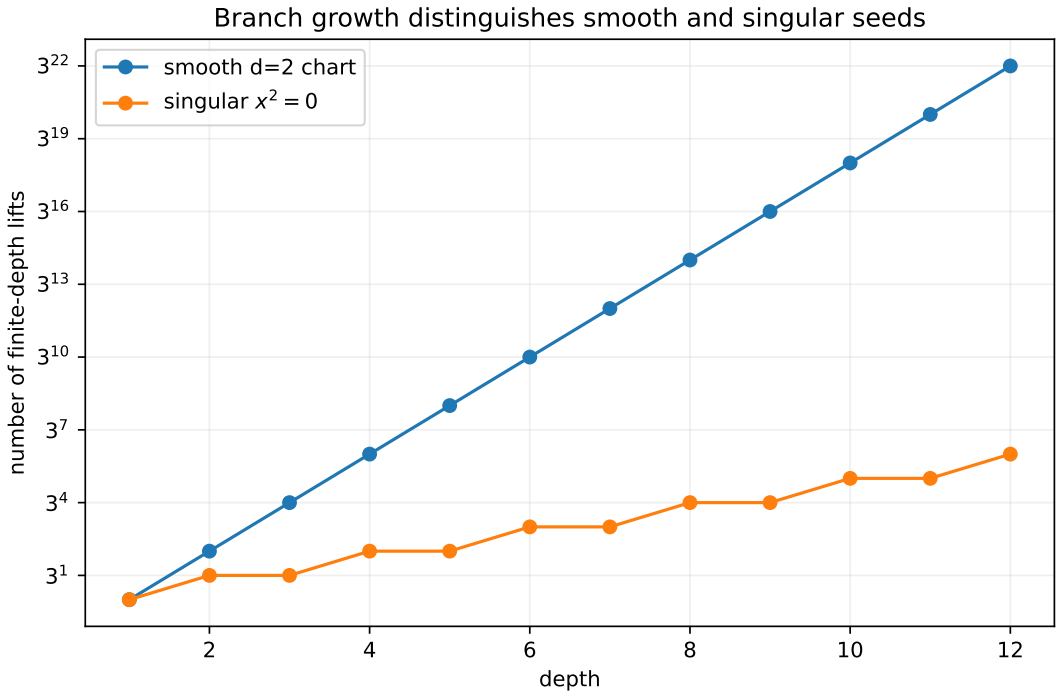


Figure 8. A smooth two-dimensional formal chart and the singular equation $x^2 = 0$ have visibly different exponential growth.

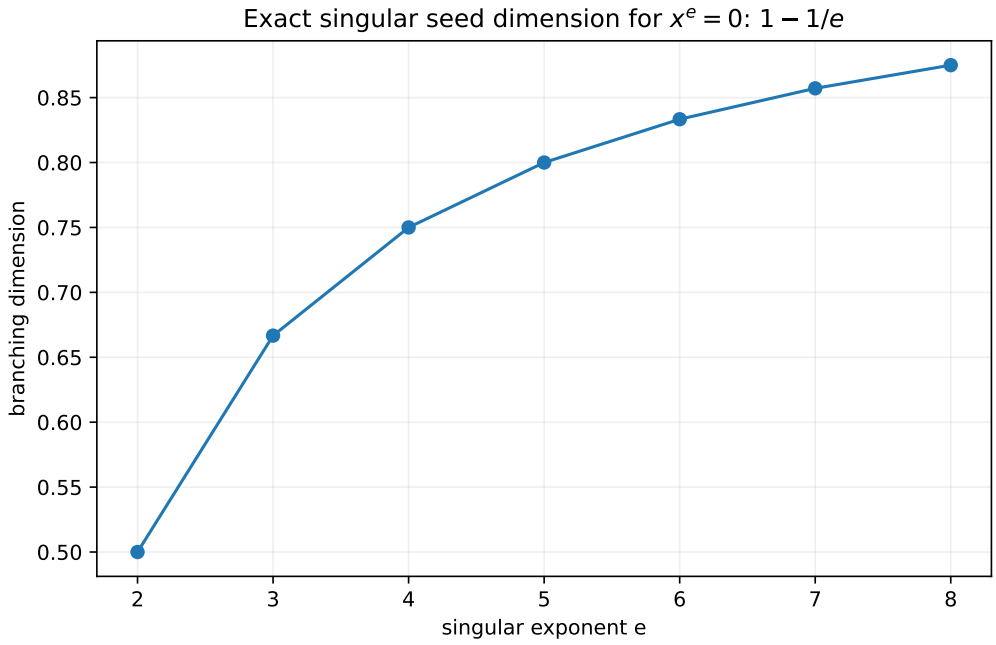


Figure 9. The finite-depth branching dimension of $x^e = 0$ is $1 - 1/e$, giving a family of fractional local growth exponents.

7 Residual transport categories

Vertical lifting never changes a residual root. Horizontal coverage requires operations whose residual effect is not congruent to the identity. Examples that may define such transports include twisting, duality, restriction, induction, tensor product, symmetric powers, exterior powers, and functorial transfer. The word "may" is essential: each operation must preserve the deformation conditions and the realization statement being transported.

Definition 7.1 (Admissible residual transport)**.** An admissible transport from $\bar{\rho}$ to $\bar{\rho}'$ consists of

$$\mathcal{T} : \bar{\rho} \longmapsto \bar{\rho}'$$

together with natural maps

$$\mathcal{T}_A : \mathrm{Def}_{\bar{\rho}}(A) \longrightarrow \mathrm{Def}_{\bar{\rho}'}(A)$$

for every coefficient algebra A, compatible with base change and reduction. When the transport is reversible, it is an isomorphism in the residual transport groupoid. One-way operations give a directed category.

For a finite collection of roots, record a directed edge $u \to v$ whenever occupancy or the relevant property is known to propagate from u to v.

Definition 7.2 (Directed seed rank)**.** For a directed transport graph G, define

$$\mathrm{srank}_G = \min\{|S| : \mathrm{Cl}_G(S) = V(G)\},$$

where $\mathrm{Cl}_G(S)$ is the set of vertices reachable from S.

Theorem 7.3 (Directed minimum-seed theorem)**.** *Let G be a finite directed graph. Collapse every strongly connected component to one vertex, obtaining the condensation DAG. Then*

$$\boxed{\mathrm{srank}_G = \#\{\textit{source strongly connected components of } G\}.}$$

One representative from each source component is a minimum seed set.

Proof. Every vertex of a finite DAG is reachable from a source, so one seed in each source component reaches the entire condensation graph. Within a strongly connected component, one vertex reaches every other. Hence the source representatives are sufficient.

Conversely, a source component has no incoming edge from another component. It cannot be reached from a seed outside itself, so every generating seed set must meet it. Therefore the number of source components is also a lower bound. □

For reversible transports the condensation has one vertex per connected component, recovering the undirected minimum-seed theorem.

Formal scope and propagation. Reachability is the reflexive-transitive closure of the supplied directed edge relation, and a strongly connected component is the actual quotient by mutual reachability. The minimum seed count is defined independently as the least cardinality of a set that reaches all vertices. Every covering set must meet every source component. Conversely, meeting every source component suffices in a finite graph. Thus any choice of one representative from each source is optimal. Empty

graphs have rank zero, and self-loops are permitted. Adding admissible edges cannot increase the minimum seed count.

If a property P propagates along every supplied edge and holds at all selected seeds, it holds at every reached vertex. These are explicit hypotheses. The minimum-seed theorem does not construct valid arithmetic edges or supply initial occupancy.

Why the finite hypothesis cannot be dropped. Take vertices $\{*\} \sqcup \{v_n : n \in \mathbf{N}\}$, with $*$ isolated and edges $v_{n+1} \to v_n$. Every strongly connected component is a singleton, and $\{*\}$ is the unique source component. No finite seed set covers all vertices: it can reach only indices no larger than the largest chosen index. Both the unique-source statement and the failure of every finite cover are checked in Lean. Hence one source component in an infinite condensation does not imply that one seed reaches all objects.

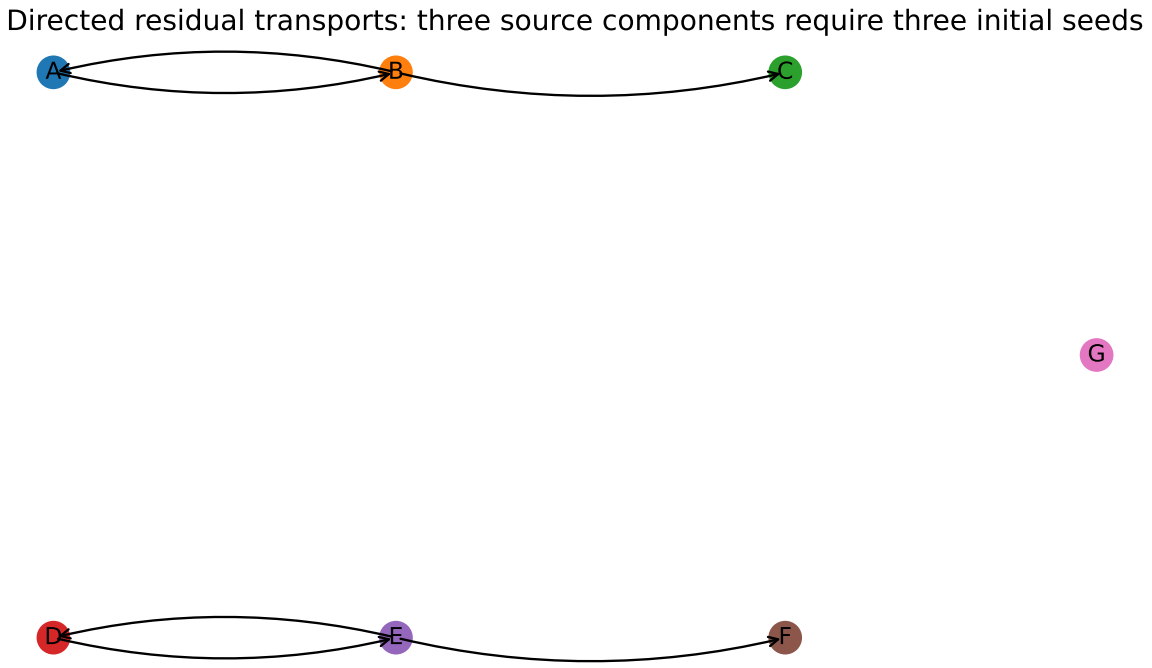


Figure 10. A finite directed transport model. The components $\{A, B\}$, $\{D, E\}$, and $\{G\}$ are sources. C and F are reached downstream.

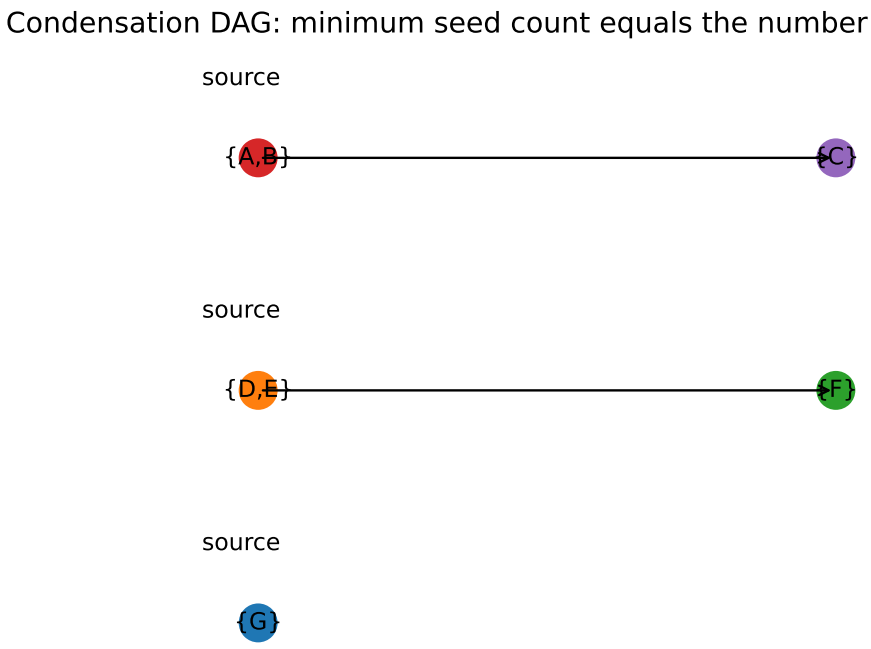


Figure 11. The condensation DAG makes the minimum seed count visible without searching every subset of roots.

Numerical Certificate 7.4 (Directed seed count)**.** The test graph has five strongly connected components and three source components. The theorem gives seed rank three. Exhaustive search over every subset of vertices independently returns three, and the selected representatives reach all seven roots.

An explicit realization of the directed graph. An independently specified graph has vertices A, B, C, D, E, F, G and the eight edges

$$A \to B,\ B \to A,\ B \to C,\ C \to F,\ D \to E,\ E \to D,\ E \to F,\ G \to F.$$

The source components are $\{A, B\}$, $\{D, E\}$, and $\{G\}$. The other components are $\{C\}$ and $\{F\}$. A bounded path search to length three is proved closed under every original edge, which certifies equality with unbounded reachability. The resulting label certificate is identified with the actual quotient by mutual reachability. Lean checks seed rank three and exactly the four minimum sets

$$\{A, D, G\}, \quad \{A, E, G\}, \quad \{B, D, G\}, \quad \{B, E, G\}.$$

Independent exact tests enumerate every directed graph on zero through four vertices, including loops, and every seed subset. These finite tests audit the implementation. The universal theorem is a separate formal proof.

8 Prime-independent system seeds

A compatible system should not be treated as an unrelated collection of λ-adic roots. Its members share algebraic Frobenius polynomials, Hodge–Tate data, determinant information, and local Weil–Deligne packets.

Definition 8.1 (Common arithmetic spine)**.** Let M be a number field. A common spine $\mathfrak{C}$ consists of:

- characteristic polynomials $P_v(X) \in M[X]$ at almost all places;
- coefficient-prime independent weight and determinant data;
- prescribed local Weil–Deligne or inertial packet labels at the ramified places.

For each finite place λ of M, let

$$X_{\lambda,m}(\mathfrak{C})$$

be the finite set of admissible objects modulo ϖ_λ^m realizing the reduction of this spine.

The system seed is the fibered object

$$\boxed{\mathscr{S}_{\mathrm{sys}} = \mathscr{S}_2^{\mathrm{wt}} \times_{\mathfrak{C}} {\prod_{\lambda\nmid 2}}' \mathscr{S}_\lambda.}$$

The fiber product notation states a compatibility requirement. It does not prove that every fiber is nonempty.

Theorem 8.2 (Adelic finite-depth completion)**.** *Assume:*

1. *an algebraic common spine $\mathfrak{C}$ over a number field M is fixed;*
2. *for every finite place λ of M and every $m \geq 1$, the companion level $X_{\lambda,m}(\mathfrak{C})$ is finite and nonempty;*
3. *the reduction forests are finitely branching;*
4. *the selected local conditions are closed under inverse limits.*

Then for every λ there is an infinite compatible branch

$$\rho_\lambda \in \varprojlim_m X_{\lambda,m}(\mathfrak{C}),$$

and the collection $\{\rho_\lambda\}_\lambda$ realizes the prescribed strictly compatible spine.

Proof. Finite-branch compactness gives an infinite branch in every coefficient-prime forest. Inverse-limit closure gives an admissible λ-adic object. By definition each finite level has the prescribed reductions of the common Frobenius and local packets. Equality at every finite precision implies equality in the complete coefficient ring. Thus all members realize the same spine. □

Corollary 8.3 (Residual companion obstruction)**.** *If*

$$X_{\lambda,1}(\mathfrak{C}) = \varnothing$$

for one coefficient prime λ, no compatible completion with spine $\mathfrak{C}$ exists. Deeper perturbations at other primes cannot create the missing residual companion.

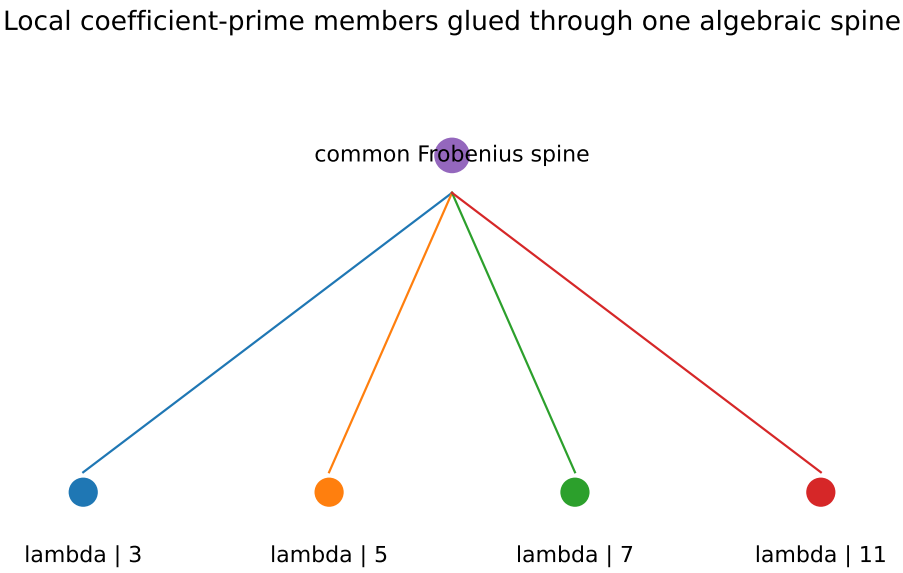


Figure 12. The local members are coordinates of one system only after their finite data descend to a common algebraic spine.

Example 8.4 (A scalar companion failure)**.** For

$$f(x) = x^2 + 1,$$

there are two simple branches at $p = 5$ and $p = 13$, no residual roots at $p = 3$ or $p = 7$, and one root modulo 2 that has no child modulo 4. A complete 5-adic branch therefore does not imply an adelic solution.

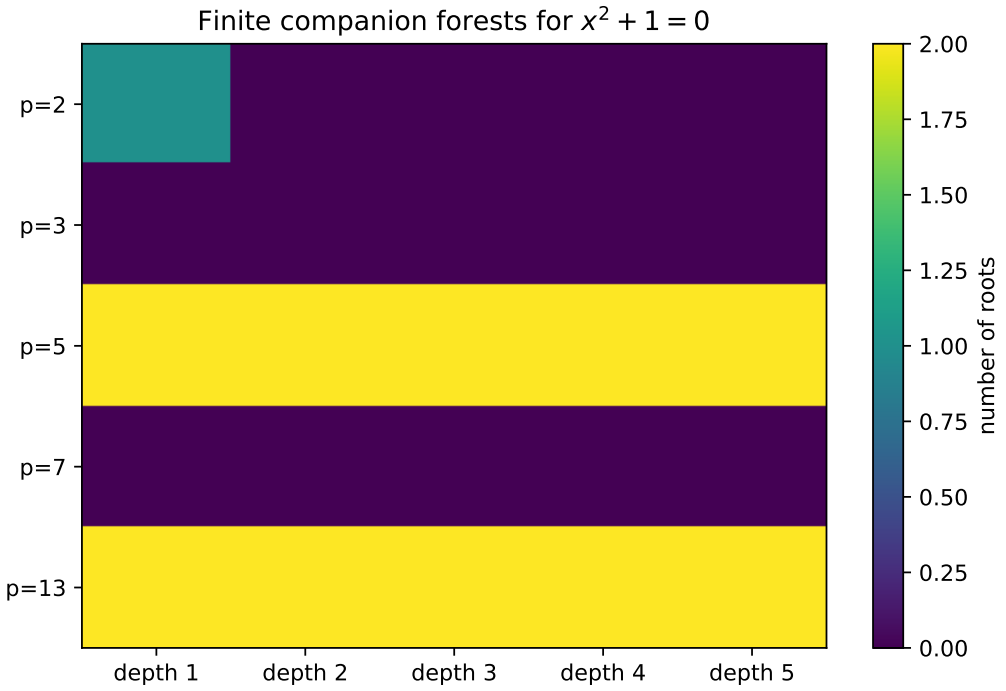


Figure 13. Finite companion levels for $x^2 + 1 = 0$. The obstruction at $2, 3, 7$ is visible before any deep lifting argument begins.

Conditional scope and complete assembly. The theorem above is conditional on the chosen spine and the stated local inverse-limit closure. In the fixed-rank integral representation setting, "closed under inverse limits" means that every compatible family of admissible finite representation classes, after coherent matrix framing and reconstruction, satisfies the full selected local realization relations. It is not merely closedness of an unrelated label space. Weight and packet relations retain their supplied arithmetic meaning. They are parameters of the universal theorem. For a general moduli functor, effectivity requires its own reconstruction hypothesis or the corresponding full meaning of admissible inverse-limit closure.

The complete Lean assembly now includes the common polynomials, good-place sets with finite complements, coefficient-specific prescribed determinants, and every selected weight and packet relation. Determinants are recovered from their actual finite-class determinants. A single common field-valued character on the whole Galois group is not assumed. The returned representations are continuous, and their reductions form the actual compatible branches. Their common polynomial identities hold at every declared good place, hence almost everywhere for each coefficient prime.

This checks Claim 8.2 as the universal conditional source theorem. The previous partial status also demanded proofs that particular intended arithmetic packets satisfy its premises. That stronger application target is not a conclusion of the theorem. Concrete Hodge–Tate and Weil–Deligne attachments, torsion interpretations, closure instances, and companion nonemptiness remain obligations of arithmetic applications. Inertial-type labels are not upgraded to full Weil–Deligne data.

Checked representation-level completion and its arithmetic boundary. The representation-level completion construction works with actual matrix representations of a fixed group. For a complete separated ring R and ideal I, compatible maps $\rho_m : G \to \mathrm{GL}_d(R/I^m)$ reconstruct a unique integral

representation for that fixed compatible family. Multiplication and inverses are verified after every reduction. If all finite representations are continuous for the discrete quotient topology, the integral representation is continuous for the ideal-adic topology. This does not assert uniqueness among all possible branches of the allowed forest.

The forest has as its level the actual subtype of representations satisfying the declared finite conditions. Those conditions must be preserved by reduction. Finiteness and nonemptiness at every level give an infinite branch and hence an actual continuous representation. Finiteness of the coefficient quotient alone is not a substitute for finiteness of the allowed representation set for an arbitrary source group. Surjectivity of the reductions is not required.

All coefficients of the characteristic polynomial commute with reduction. Consequently, prescribed integral characteristic polynomials at every precision hold exactly in the reconstructed representation. For a common coefficient field E, local integral polynomials lie in $\mathcal{O}_\lambda[X]$ and are compared with $P_v \in E[X]$ only after mapping both into the local coefficient field. No field embedding $E \to \mathcal{O}_\lambda$ is assumed. The allowed-place predicate is explicit and may depend on λ. Trivial action on a supplied subgroup is likewise detected at every precision. Identifying that subgroup with arithmetic inertia is a separate input.

The eleventh block assembles continuous coefficient-field representations, exact common polynomials and the prescribed local predicate from positive finite depths. Depth-zero finiteness and nonemptiness are derived. Claims 8.2 and 11.1 already assume inverse-limit closure of the selected local conditions. An unconditional closure theorem is not a missing hypothesis of their conditional proofs. Their conditional matrix core is checked. The remaining full arithmetic interpretation must identify the admissible finite objects and the actual local invariants with these data. The coherent-framing obligation is resolved by the completion from conjugacy classes below. An arbitrary local predicate is not being renamed Hodge–Tate or Weil–Deligne compatibility.

Completion from conjugacy classes. Over an I-adically complete separated commutative coefficient ring R, for a fixed finite rank, the formalization now takes finite-level objects to be actual matrix representations modulo conjugation by an invertible matrix. Assume the coefficient quotients R/I^k are finite. Conjugation commutes with coefficient reduction, and each conjugacy class has finitely many frames over a finite coefficient ring. Thus finite allowed class levels give finite full preimages of allowed matrix representations. Downward admissibility and positive-depth nonemptiness yield an exactly compatible matrix branch and an actual integral representation. More strongly, every prescribed compatible class branch has such representatives, using finite nonempty fibers over that branch. No surjectivity of reductions or lifting of prescribed change-of-basis matrices is assumed. This does not assert uniqueness up to integral conjugacy.

For finitely generated I with finite R/I, finiteness of R/I^k is derived. One continuous representative per allowed class suffices, because continuity is proved invariant under conjugation. Characteristic polynomials, determinants and trivial action on a supplied subgroup are likewise actual invariants. The unframed common-spine theorem constructs continuous coefficient-field representations with the prescribed whole polynomials and the explicitly supplied inverse-limit-closed local predicate.

An actual $\mathbb{Z}_3$ example applies the class-level completion theorem for the cyclic group of order two and polynomial $X^2 - 1$, producing a nonidentity generator. Another checked example has compatible classes represented by $\mathrm{diag}(1, -1)$ modulo 3 and $\mathrm{diag}(-1, 1)$ modulo 9: these chosen matrices disagree under reduction, and an actual swap of basis repairs them. Arbitrary independent representatives therefore do not replace the framing argument.

The framing obligation is finished. Identifying the manuscript's arithmetic admissible objects, actual local invariants and finite-level packet reductions with this interface remains separate. An uninterpreted local predicate is not a construction of Hodge–Tate or Weil–Deligne compatibility. The source's local closure hypothesis remains explicit. The conditional completion construction completes the universal conditional implication. These particular arithmetic applications remain unproved.

Actual local comparisons from fixed-depth certificates. Let R be an I-adically complete separated commutative ring with finite coefficient quotients and an injective map into a field K. Fix a finite matrix rank. For finitely many local group maps $\phi_s : H_s \to G$ and integral template representations τ_s, fix detecting depths d_s. At precision m retain matrices $P_{s,m}$ as part of the admissible object and require

$$\rho_m(\phi_s(h))P_{s,m} = P_{s,m}\tau_{s,m}(h).$$

For $m \geq d_s$ require their determinants to remain nonzero after reduction to R/I^{d_s}. Earlier levels omit that test. Finite occupied augmented levels give a compatible branch of both representations and comparisons. The integral limits satisfy the actual equations and have nonzero determinants, so the comparisons are units over K. A converse supplies all finite nodes from any integral realization with these detectors. Finite-level invertibility and compatible prior choices of comparisons are not assumed.

The unframed positive-depth version also retains continuity and whole common polynomials. Its actual local isomorphisms are derived from the certificates. There is no uninterpreted local-stability predicate for this particular problem. The fixed depths and augmented-level occupancy remain explicit inputs.

The formalization specializes to the actual maps $G_{\mathbb{Q}_p} \to G_{\mathbb{Q}}$ induced by the field embeddings and choices of algebraic closures. A nontrivial continuous quadratic character of $G_{\mathbb{Q}}$, constructed from its action on $i^2 = -1$, gives the integral 2-adic actions

$$\tau(\sigma) = \operatorname{diag}(1,-1), \quad \rho(\sigma) = \begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$$

at a nonidentity quadratic element. The determinant 2 is detected modulo 4. The representations are conjugate over $\mathbb{Q}_2$ but not integrally conjugate. A separate $\mathbb{Z}_3$ example invokes simultaneous completion and reconstructs a comparison which is an integral nonunit and a field unit.

The bound is substantive: for the unipotent actions $\rho_a(g) = \begin{pmatrix} 1 & ag \\ 0 & 1 \end{pmatrix}$, every integral intertwiner with ρ_1 has determinant aP_{22}^2. Taking $a = \pi^n$ for a nonzero $\pi \in I$ rules out every fixed detecting depth, even though all the field-valued approximants are isomorphic and their integral limit is trivial. This does not refute Claim 8.2, which already assumes inverse-limit closure of its chosen local conditions.

Exact prescribed local Galois action is stronger than an inertial-type label. The common arithmetic packet interpretation, its coefficient comparisons and the actual Hodge–Tate/Weil–Deligne attachment remain separate. The new theorem does not supply a uniform bound for all intended arithmetic packets. Those arithmetic applications remain unfinished. The universal conditional implication is checked in the conditional completion construction.

Derived comparison bounds. For a finite group Γ whose order is a unit in a commutative local coefficient ring R, any prescribed equivariant residual isomorphism lifts to an integral isomorphism. Lift its matrix to A and define

$$P = |\Gamma|^{-1} \sum_{g \in \Gamma} \rho(g) A \tau(g^{-1}).$$

Reindexing proves equivariance, and reduction recovers the prescribed matrix. Its determinant is a unit. Complete local coefficients with finite quotients, occupied compatible finite conditions and this residual type therefore give actual integral completion with detector depth one. A fixed continuous map $G \to \Gamma$ permits pullback to infinite source groups. The finite-level representations must factor through the same map, and the residual type is imposed on their Γ-actions. No uniform determinant bound or augmented certificate occupancy is assumed in this sector.

For a second sector let R be a DVR with uniformizer π, fraction field K and matrix rank n. If a fixed template τ has full K-linear action span, finite generation and denominator clearing give a fixed c such that

$$\pi^c M_n(R) \subseteq \operatorname{span}_R\{\tau(g) : g \in G\}.$$

Normalize a field comparison to an integral P with a unit entry. The displayed span constructs an integral B with $BP = \pi^c I$, so

$$\det P \mid \pi^{cn}, \qquad \det P \not\equiv 0 \pmod{\pi^{cn+1}}.$$

One c works for all integral representations in the fixed field-isomorphism class. Jacobson density derives full span from actual irreducibility and a scalar commutant. The formalization states these hypotheses directly and does not assume the DVR fraction field is algebraically closed.

If at every precision there is an integral representation of this rational type satisfying the finite conditions, the derived bound supplies the simultaneous certificates of the fixed-depth comparison-certificate construction and preserves the type after completion. The integral approximants are not required to be coherent, but their existence is a substantive premise. Arbitrary torsion nodes are not asserted to lift.

Over $\mathbb{Z}_3$, the actual average corrects $A = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ to $P = \left(\begin{smallmatrix} 1 & -1/2 \\ 0 & 1 \end{smallmatrix}\right)$ for the involutions $\rho(g) = \left(\begin{smallmatrix} 1 & 1 \\ 0 & -1 \end{smallmatrix}\right)$ and $\tau(g) = \operatorname{diag}(1, -1)$. It preserves the residual comparison and has determinant one. Pullback through the quadratic character of the actual $G_{\mathbb{Q}}$ gives continuous arithmetic examples. In contrast, the identity and $\operatorname{diag}(1, \operatorname{sign})$ over $\mathbb{Z}/4$ agree modulo two but are not integrally conjugate. Their group order is not a unit. These restricted results do not supply general arithmetic packet attachment, all finite-level existence statements, or cross-coefficient compatibility.

9 Compatible carriers and finite Frobenius separation

One useful completion mechanism is available when the desired packet already occurs as a semisimple summand of a known compatible carrier.

Theorem 9.1 (Finite Frobenius separation). *Let $\mathscr{W}_1, \dots, \mathscr{W}_r$ be pairwise nonisomorphic strictly compatible systems whose members are absolutely irreducible, with common algebraic unramified trace values $t_i(v)$ in a number field E. There are r distinct permitted unramified places $v_1, \dots, v_r$ for which*

$$C = (t_i(v_j))_{j,i} \in M_r(E), \qquad \det C \neq 0.$$

Every coefficient-field embedding $\sigma_\lambda : E \hookrightarrow E_\lambda$ therefore sends C to an invertible matrix. Its entries agree with the ordinary representation traces $\mathrm{Tr}(W_{i,\lambda}(\mathrm{Frob}_{v_j}))$ when the selected places satisfy the unramified compatibility convention for that coefficient prime. In particular, places above the residue characteristic of λ are excluded from this ordinary trace comparison.

Proof. Fix one coefficient prime. The classical independence of the characters of pairwise nonisomorphic absolutely irreducible representations and the density of unramified Frobenius conjugacy classes give independent restricted trace functions. Density here concerns the union of the conjugacy classes, not an unsupported assertion about a set of chosen representatives.

For r independent functions, their evaluation linear forms span the r-dimensional coefficient dual: a proper span would leave a nonzero coefficient vector annihilated by every evaluation, contradicting independence. Choose a basis from the evaluations. Its r distinct sample points give an invertible square matrix. The common algebraic traces put that matrix over E. An injective field embedding reflects the nonvanishing of its determinant. All further field embeddings preserve the same nonvanishing. □

Formalization boundary. Characteristic-zero independence is now Lean-checked for actual finite-dimensional irreducible representations of arbitrary monoids. Finiteness of the group and algebraic closedness of the field are not needed for this result. The proof uses the actual monoid algebra and Jacobson density to select constituent actions. Jointly continuous coordinate actions have continuous characters. Coverage in every open normal quotient of a compact totally disconnected group implies density of the conjugacy-saturated samples and hence the finite separator. General arithmetic Frobenius coverage remains unformalized. The system-isomorphism and finite-cyclotomic comparison now proves the bridge from componentwise nonisomorphic systems to nonisomorphic members at the selected coefficient prime, under density on the required good-place overlaps. It also proves the complete restricted separator for a fixed finite cyclotomic Galois group over $\mathbf{Q}$, with arithmetic coverage derived from the proved Dirichlet theorem. The full arithmetic theorem remains partial. Selection is existential and gives no effective upper bound on the selected primes. A common algebraic matrix is still distinct from an ordinary unramified trace comparison at all coefficient primes simultaneously.

System isomorphism and good-place overlaps. The system equivalence used here is componentwise representation isomorphism:

$$\mathscr{W}_i \simeq \mathscr{W}_j \quad \Longleftrightarrow \quad \forall \lambda,\ W_{i,\lambda} \simeq W_{j,\lambda}.$$

For finite-dimensional irreducible representations of any monoid over a characteristic-zero field, the new character-rigidity theorem proves that equality of characters is equivalent to an actual equivariant linear isomorphism. Neither a finite group nor an algebraically closed coefficient field is needed.

Suppose two members are isomorphic at λ_0. Injectivity of the common coefficient embedding gives $t_i(v) = t_j(v)$ at every place good for λ_0. At another coefficient prime λ, this equality compares actual member traces on the intersection of the two good-place sets. Conjugacy density of that intersection and continuity extend the equality to the whole group. Character rigidity then gives an actual isomorphism at λ. Thus the nonisomorphism of systems supplies the required nonisomorphism of members. Equality of optional labels or record fields is not used as a substitute for representation equivalence.

The generic separator now derives this bridge and permits any additional finite set of excluded places. Its coverage premise retains every finite deletion: for each finite excluded set and each open normal quotient, every quotient class contains a conjugate of an allowed sample. This implies density

on each pair of cofinite good-place sets. No all-coefficient-prime common ordinary-trace domain is assumed.

Arithmetic coverage for a fixed cyclotomic extension. Let $n > 0$ and let $L = \mathbf{Q}(\zeta_n)$ be a cyclotomic number field. The canonical isomorphism

$$\mathrm{Gal}(L/\mathbf{Q}) \simeq (\mathbb{Z}/n\mathbb{Z})^\times$$

defines σ_p by $\sigma_p(\zeta_n) = \zeta_n^p$ when $p \nmid n$. For every automorphism σ, finite set $B \subset \mathbb{N}$ and bound b, Lean proves that there is a rational prime $p > b$, with $p \notin B$ and $p \nmid n$, such that $\sigma_p = \sigma$. The prime-existence step uses the proved Dirichlet theorem in the pinned Mathlib dependency.

The arithmetic attachment is checked in this finite cyclotomic case. There is an actual maximal prime $\mathfrak{P}$ of $\mathcal{O}_L$ above (p), its ramification index is one, and σ_p belongs to its decomposition group. For every integral $x \in \mathcal{O}_L$,

$$\overline{\sigma_p(x)} = \overline{x}^{\,p} \quad \text{in } \mathcal{O}_L/\mathfrak{P}.$$

The residue identity is proved on all integral elements using $\mathcal{O}_L = \mathbb{Z}[\zeta_n]$. Existence of $\mathfrak{P}$ is obtained by lying over, rather than assumed. This specifies arithmetic Frobenius, with p-power residue action. The action and unramifiedness statements also hold for every maximal prime above the selected p.

Consequently, for pairwise componentwise nonisomorphic compatible irreducible representation families of this fixed finite Galois group, the common algebraic trace matrix can be selected at distinct rational primes outside any prescribed finite set, with nonzero determinant. Every coefficient embedding preserves its nonvanishing. Actual member trace comparisons retain that member's good-prime condition. No coverage or character-continuity hypothesis is added in this cyclotomic specialization: the prime coverage is proved, and the finite extension's Krull topology is discrete. A concrete conductor-four instance uses the actual field $\mathbf{Q}(\zeta_4)$.

This is a complete restricted arithmetic case of Claim 9.1. It does not establish coverage in arbitrary finite quotients of the absolute Galois group, nor the Hodge–Tate and Weil–Deligne comparisons of Claim 9.2. The source-claim ledger therefore remains seventeen checked and three partial.

Absolute Galois restriction and arithmetic witnesses. Fix a cyclotomic number field $L = \mathbf{Q}(\zeta_n)$ with $n > 0$ and an embedding $j : L \hookrightarrow \overline{\mathbf{Q}}$. The new restriction map

$$q_L : G_{\mathbf{Q}} \longrightarrow \mathrm{Gal}(L/\mathbf{Q}), \qquad j(q_L(\gamma)x) = \gamma(j(x)),$$

is an actual continuous surjective homomorphism with open kernel. Surjectivity is derived by extending automorphisms to the algebraic closure. Pullback along a surjective homomorphism preserves and reflects equivariant linear isomorphisms, invariant subspaces, and irreducibility. These assertions keep the underlying linear maps and submodules and do not assume a finite acting group. In particular, the equivalence relation for systems is preserved and reflected componentwise.

The arithmetic witnesses use the integral closure $\mathcal{A}$ of $\mathbb{Z}$ in $\overline{\mathbf{Q}}$. For each rational prime p, Lean constructs a maximal ideal $\mathfrak{Q}$ of $\mathcal{A}$ above (p) and an actual element $\gamma_p \in G_{\mathbf{Q}}$ such that

$$\gamma_p \in \mathrm{Stab}_{G_{\mathbf{Q}}}(\mathfrak{Q}), \qquad \overline{\gamma_p(a)} = \overline{a}^{\,p} \quad (a \in \mathcal{A}).$$

The existence argument applies Mathlib's proved profinite stabilizer-surjectivity theorem to this actual invariant ring. The residue p-power map is proved bijective by taking p-th roots in $\overline{\mathbf{Q}}$ and showing

that they remain integral when their p-th powers are integral. This constructs an arithmetic Frobenius witness in the prime-ideal decomposition group. It does not assert that an arbitrary global lift belongs to that group.

When $p \nmid n$, the comparison with the finite cyclotomic sample is proved:

$$q_L(\gamma_p) = \sigma_p, \qquad \sigma_p(\zeta_n) = \zeta_n^p.$$

Reduction distinguishes n-th roots of unity at such a prime. Indeed, if $x^n = y^n = 1$ and their reductions agree, the factorization of $x^n - y^n$ gives

$$(x - y) \sum_{i=0}^{n-1} x^i y^{n-1-i} = 0.$$

The second factor reduces to $n\overline{y}^{n-1} \neq 0$, so the domain property gives $x = y$. This proves both Frobenius recognition on the cyclotomic generator and triviality of its inertia action. Consequently, every representation pulled back through q_L is unramified there.

The valuation formulation is checked separately. For any valuation subring $A \subset \overline{\mathbf{Q}}$ with residue characteristic $p \nmid n$, its actual residue-action kernel fixes every n-th root of unity. A supplied element of its decomposition subgroup with p-power residue action restricts to σ_p. These statements use Mathlib's valuation inertia and decomposition subgroups. No identification of the valuation and integral-closure prime models is assumed in the formal comparison.

For a compatible family $\rho_{i,\lambda}$ of representations of the fixed finite group, set $V_{i,\lambda} = \rho_{i,\lambda} \circ q_L$. Suppose the actual absolute-Galois members are irreducible and the systems are pairwise componentwise nonisomorphic. The finite prime separator now yields distinct permitted rational primes p_j, actual prime ideals $\mathfrak{Q}_j$ and arithmetic Frobenius witnesses γ_j as above, outside any prescribed finite set, such that the common algebraic trace matrix has nonzero determinant. Every coefficient embedding preserves this nonvanishing, and

$$\iota_\lambda(t_i(p_j)) = \operatorname{tr}(V_{i,\lambda}(\gamma_j))$$

whenever p_j is good for that coefficient member. The operator and trace values agree for every global lift with the same cyclotomic restriction. Neither Frobenius-witness existence nor quotient coverage is a supplied hypothesis in this restricted arithmetic theorem. The common cyclotomic factorization and the compatible trace data remain explicit hypotheses.

This completes the passage to actual absolute-Galois Frobenius witnesses for the fixed cyclotomic class. It does not establish prime coverage for arbitrary finite Galois extensions. Claim 9.1 therefore remains partial, and the source ledger remains seventeen checked and three partial.

The boundaries are formalized as well. Pullback along a nonsurjective homomorphism can make inequivalent irreducible representations isomorphic. Samples can cover one quotient while missing an open quotient of the full group. Finally, in $\mathbb{Z}_2$, the distinct square roots 1 and -1 have the same residue, so the condition that the root order be a unit cannot be discarded.

General finite-field Frobenius comparison. Let $L/\mathbf{Q}$ now be any finite normal number field, with a chosen embedding $j : L \hookrightarrow \overline{\mathbf{Q}}$. Write $\mathcal{A}$ for the integral closure of $\mathbb{Z}$ in $\overline{\mathbf{Q}}$. The embedding restricts to an injective map $\mathcal{O}_L \to \mathcal{A}$ which intertwines the actual Galois actions. If $\mathfrak{Q} \subset \mathcal{A}$ is maximal above (p), its contraction $\mathfrak{P} \subset \mathcal{O}_L$ is a nonzero maximal ideal above (p), and the induced residue map is injective.

Restriction carries the actual decomposition group at $\mathfrak{Q}$ into that at $\mathfrak{P}$, carries ideal inertia into ideal inertia, and preserves the arithmetic p-power residue action. Thus the absolute Frobenius constructed in the absolute-Galois restriction construction gives a simultaneous finite Frobenius at the contracted prime. This assertion applies to arbitrary finite normal number fields and does not require a cyclotomic generator.

The exceptional primes are controlled by the actual field discriminant:

$$S_L = \{p \text{ rational prime} : p \mid \operatorname{disc}(L)\}.$$

This set is finite. For $p \notin S_L$, every prime of $\mathcal{O}_L$ above (p) is unramified. Lean proves that the actual inertia cardinality is the ramification index. It follows that inertia is trivial and the arithmetic Frobenius at a fixed finite prime is unique as a field automorphism. Neither trivial inertia nor uniqueness is assumed in place of the arithmetic unramifiedness condition.

The choice of absolute prime and Frobenius lift is also controlled. Let $(\mathfrak{Q}_u, \gamma_u)$ and $(\mathfrak{Q}_v, \gamma_v)$ be any two actual arithmetic witnesses at the same rational prime. Transitivity on primes of $\mathcal{A}$ above (p) gives $a \in G_{\mathbf{Q}}$ with

$$\mathfrak{Q}_v = a\mathfrak{Q}_u, \qquad \gamma_v^{-1} a \gamma_u a^{-1} \in I_{\mathfrak{Q}_v}.$$

Consequently, any representation trivial on $I_{\mathfrak{Q}_v}$ has the same trace on the two witnesses. For representations factoring through L, triviality on this actual absolute inertia is derived whenever $p \notin S_L$. The restrictions of all absolute witnesses at such a prime therefore form a single conjugacy class. This conclusion requires only the indicated finite factorization and the discriminant exclusion.

There is a final compatibility between embeddings. For a normal subfield $L \subset \overline{\mathbf{Q}}$, restriction defined through a chosen embedding is conjugate to canonical restriction by one fixed automorphism of L. This identifies their conjugacy classes and lets the arithmetic comparison apply directly to the fixed fields of open normal subgroups of $G_{\mathbf{Q}}$. Those fixed fields are proved to be finite Galois number fields, with the original open normal subgroup as the kernel of restriction.

The remaining prime-existence obligation now has a precise formulation. For every finite Galois subfield L, every finite set B of rational primes, and every $\sigma \in \operatorname{Gal}(L/\mathbf{Q})$, require a prime $p \notin B \cup S_L$ and an actual absolute arithmetic witness γ_p such that

$$q_L(\gamma_p) \sim \sigma.$$

Lean proves that this assertion is equivalent to the cofinite quotient coverage required by the general separator, for any fixed choice of actual Frobenius witnesses. The equivalence derives independence from the witness choices. It does not supply the displayed prime-existence assertion. Arbitrary finite exclusions absorb S_L, and the resulting conditional density theorem preserves the separate good-prime sets of coefficient members. General prime coverage remains unproved, so Claim 9.1 remains partial. The source ledger is still seventeen checked and three partial. This new arithmetic interface has base $\mathbf{Q}$. For the source's general-base instances involving $G_{F,S}$, the analogous actual-prime and norm-power Frobenius attachment and coverage over F, or a proved transfer to the rational-base result, also remain necessary.

Finite deletion can destroy density. In the actual discrete group $C_2 = \{1, a\}$, set $s(0) = a$ and $s(m+1) = 1$. The original samples are surjective and satisfy quotient coverage. Removing index zero leaves only the identity, so both density and quotient coverage fail. The actual rational sign and trivial

characters agree on the remaining samples but differ globally. This checked example explains why ordinary coverage alone cannot justify deleting the bad places in the system comparison.

Theorem 9.2 (Compatible-carrier summand completion)**.** *Suppose a semisimple strictly compatible carrier decomposes as*

$$\mathscr{W} \cong \bigoplus_{i=1}^{r} \mathscr{W}_i^{\oplus c_i},$$

with the $\mathscr{W}_i$ as above. At one coefficient prime λ_0, suppose a semisimple summand has the form

$$V_{\lambda_0} \cong \bigoplus_{i=1}^{r} W_{i,\lambda_0}^{\oplus m_i}, \qquad 0 \le m_i \le c_i.$$

Then

$$\boxed{V_\lambda = \bigoplus_{i=1}^{r} W_{i,\lambda}^{\oplus m_i}}$$

defines a strictly compatible summand at every coefficient prime. The vector (m_i) is uniquely recoverable from finitely many Frobenius traces supplied by the separation theorem.

Proof. Use the same multiplicities in every member of the compatible carrier. Direct sums preserve strict compatibility, and $m_i \le c_i$ makes V_λ a summand of W_λ. At the separating places, the target trace vector is $C\boldsymbol{m}$. Full column rank makes $\boldsymbol{m}$ the unique solution. □

Recovery and realization are different steps. For a certified square separator the coefficient vector is $C^{-1}t$. This is a valid carrier multiplicity vector only when every entry is a nonnegative integer and satisfies its capacity bound $m_i \le c_i$. Those tests and uniqueness of natural multiplicities in characteristic zero are separately Lean-checked, as is transport through a field embedding. Reduction modulo a residue characteristic is not a field embedding and may lose rank. The representation construction and common-polynomial part are now Lean-checked as described below. The full strictly compatible arithmetic statement, including its local conditions, remains partially formalized.

Scope of the supplied-carrier hypothesis. Claim 9.2 supplies the compatible carrier, its decomposition and the compatible constituent systems. Constructing unknown such objects is not required by that conditional theorem. The remaining formal obligation is that finite sums preserve the intended actual local arithmetic structures, together with the arithmetic separator input. The algebraic monodromy results below do not by themselves identify attached Weil–Deligne representations or Hodge–Tate comparison data.

Explicit summands inside a supplied decomposition. Let K be a field, let G be a monoid, and let $\rho_i : G \to \mathrm{End}_K(W_i)$ be actual representations on finite-dimensional spaces, indexed by a finite set I. For natural numbers $m_i \le c_i$, set

$$U_m = \prod_{i \in I} \prod_{k < m_i} W_i, \qquad U_c = \prod_{i \in I} \prod_{k < c_i} W_i.$$

These finite products realize the corresponding finite direct sums. The representation on U_m acts componentwise. Define $j : U_m \to U_c$ by retaining the first m_i copies and filling every remaining coordinate with zero. Define $r : U_c \to U_m$ by restriction to the selected copies. Lean proves

$$rj = 1_{U_m}, \qquad j\rho_m(g) = \rho_c(g)j, \qquad r\rho_c(g) = \rho_m(g)r.$$

The image of j is exactly the subspace whose unused coordinates vanish. Consequently jr is an equivariant idempotent. A supplied equivariant isomorphism $D : W \xrightarrow{\sim} U_c$ gives the split maps $D^{-1}j$ and rD inside the original carrier. The decomposition D is an input. These statements do not construct a decomposition of an unknown arithmetic carrier.

The same formalization proves, for each $g \in G$,

$$\det\big(XI - \rho_m(g)\big) = \prod_{i\in I} \det\big(XI - \rho_i(g)\big)^{m_i}, \qquad \operatorname{Tr}\big(\rho_m(g)\big) = \sum_{i\in I} (m_i : K)\operatorname{Tr}\big(\rho_i(g)\big).$$

Constituents may have different dimensions. Zero multiplicities, zero-dimensional constituents, and the empty index set are included. If at a permitted sample g the constituent polynomials are the images of $P_{i,g} \in E[X]$ under a specified map $E \to K$, the summand polynomial is the image of $\prod_i P_{i,g}^{m_i}$. Applying this independently at each coefficient field preserves a supplied common polynomial family. This does not establish the Hodge–Tate or Weil–Deligne conditions in strict compatibility.

Joint continuity of the constituent actions implies joint continuity of the constructed action. The inclusion and retraction are continuous for the product topologies, and triviality on a supplied subgroup is preserved. Neither continuity nor triviality is inferred for a carrier whose corresponding hypothesis has not been supplied. The maps transported by D are proved algebraically. Their topological interpretation additionally requires the appropriate continuity of D and its inverse.

For the three-character example below, the construction produces an actual three-dimensional summand inside the carrier of dimension seven. At the sample $(0, 1)$ its characteristic polynomial is $(X-1)^2(X+1)$. A separate checked counterexample uses the coordinate inclusion $\mathbf{Q} \to \mathbf{Q}^2$, its coordinate retraction, and the swap involution: the split law holds, but compression sends the swap to zero and therefore fails to preserve its square. Thus selecting whole constituent copies, with the proved equivariance, is essential to this construction.

Algebraic monodromy data on the selected copies. Fix a field K and a nonzero scalar $q \in K^\times$. On each constituent W_i, supply an automorphism F_i and a nilpotent endomorphism N_i with $F_i N_i F_i^{-1} = qN_i$. The scalar q is common to this finite family. The componentwise operators F_m, N_m on U_m satisfy the same relation. F_m is invertible and N_m is nilpotent. The split maps constructed above intertwine both operators. The proof first identifies the independently defined kernel through an equivariant linear isomorphism

$$\ker N_m \simeq \prod_{i\in I} \prod_{k<m_i} \ker N_i.$$

The relation, with $q \neq 0$, proves that both F_i and its inverse preserve $\ker N_i$. Thus $F_i|_{\ker N_i}$ is an actual automorphism, and Lean proves

$$\det(XI - F_m|_{\ker N_m}) = \prod_{i\in I} \det(XI - F_i|_{\ker N_i})^{m_i}.$$

The corresponding kernel dimensions and traces are additive with multiplicity. Common constituent kernel polynomials, when supplied through a coefficient-ring map, give the mapped product polynomial for the summand. Whole-space Frobenius polynomials are not substituted for these kernel polynomials.

The exact special chain has basis $e_0, \dots, e_{n-1}$ and operators

$$Ne_0 = 0, \qquad Ne_j = e_{j-1}\ (j > 0), \qquad Fe_j = aq^{-j}e_j, \qquad a, q \in K^\times.$$

Lean proves $N^n = 0$, and for $n > 0$ also $N^{n-1} \neq 0$ and $\ker N = Ke_0$. Consequently its kernel polynomial is $X - a$, whereas its whole-space polynomial is $\prod_{j=0}^{n-1}(X - aq^{-j})$. The zero-length chain is included as a zero-dimensional space. The assertion $\ker N = Ke_0$ is made only for positive length.

For $n > 1$, replacing N by zero retains exactly the same F and the same whole-space characteristic polynomial, while changing the degree of the kernel polynomial from one to n. Both pairs still satisfy the algebraic monodromy relation. This Lean-checked example shows that whole Frobenius data alone do not determine the monodromy kernel polynomial. It is not asserted to be a counterexample among the intended strictly compatible arithmetic systems.

These statements concern supplied algebraic pairs. No Weil-group or inertia action, Hodge–Tate object, arithmetic comparison, or realization of the pair from a Galois representation is constructed here. The polynomial convention is $\det(XI - F|_{\ker N})$. No identification with an inertia-dependent local Euler factor is asserted. Claim 9.2 remains partial at its full arithmetic scope.

Checked group, inertia and monodromy finite sums. Let K be a field, G a group, and $\nu : G \to K^\times$ a character. The new datum consists of an actual representation r and nilpotent N with $r(g)N = \nu(g)Nr(g)$ for every g. For a supplied subgroup H, define

$$E(D, H) = \ker N \cap V^H.$$

This is the intersection of independently defined kernel and actual invariant subspaces. For normal H, the restricted whole-group representation and its inverse automorphisms are constructed. A change from φ to φh, with $h \in H$, leaves the restricted operator itself unchanged.

For a finite family of finite-dimensional constituents and natural multiplicities, the actual local space of the constructed sum is linearly equivalent to the sum of the constituent local spaces, equivariantly for every group element. Its characteristic polynomials multiply, its dimensions and traces add, and maps intertwining both the group and N restrict to the local spaces. The explicit bounded-copy inclusion and retraction still split there. Finite inertia image and an open inertia kernel are preserved.

Writing $Q_D(T)$ for the reversed local characteristic polynomial, Lean proves

$$Q_D(T) = \det(1 - Tr(\varphi) \mid E(D, H)), \qquad Q_D(0) = 1, \qquad Q_{\bigoplus_i D_i^{\oplus m_i}}(T) = \prod_i Q_{D_i}(T)^{m_i}.$$

The abstract formulas do not need characteristic zero. Normality and finite dimension appear explicitly where used. Arithmetic identification of G, H, ν and φ remains separate. In the geometric-Frobenius convention, $\nu(\varphi)$ is the inverse residue cardinality. It is this unit, rather than the residue cardinality, that enters the earlier special-chain parameter.

An explicit action of $C_2 \times \mathbb{Z}$ has $\nu(\varphi) = 1/3$, $F = \operatorname{diag}(2, 6, 18)$ and $Ne_1 = e_0$, $Ne_2 = e_1$, $Ne_0 = 0$. Trivial inertia gives local dimension one and denominator $1 - 2T$. Scalar sign inertia gives dimension zero and denominator one. The entire F and N are equal in these two examples, and $N^2 \neq 0$. Thus

inertia cannot be omitted. Conversely, unramified chains of lengths one and three have equal Euler denominators while only the first has $N = 0$. Euler factors therefore do not identify complete monodromy data.

The remaining arithmetic comparison is $\mathrm{WD}(\bigoplus_i V_i^{\oplus m_i}) \cong \bigoplus_i \mathrm{WD}(V_i)^{\oplus m_i}$ for the actual attached functor, intertwining the Weil action and monodromy, together with the intended coefficient and Hodge–Tate comparisons. These are not supplied by the abstract datum. Claim 9.2 remains partial at its full arithmetic scope.

Recovered multiplicities determine the semisimple isomorphism type relative to the specified constituents, not a canonical embedding or projector inside a repeated isotypic component. For example, the two projectors $\mathrm{diag}(1,0)$ and $\mathrm{diag}(0,1)$ on a two-dimensional trivial representation are distinct equivariant idempotents with trace one.

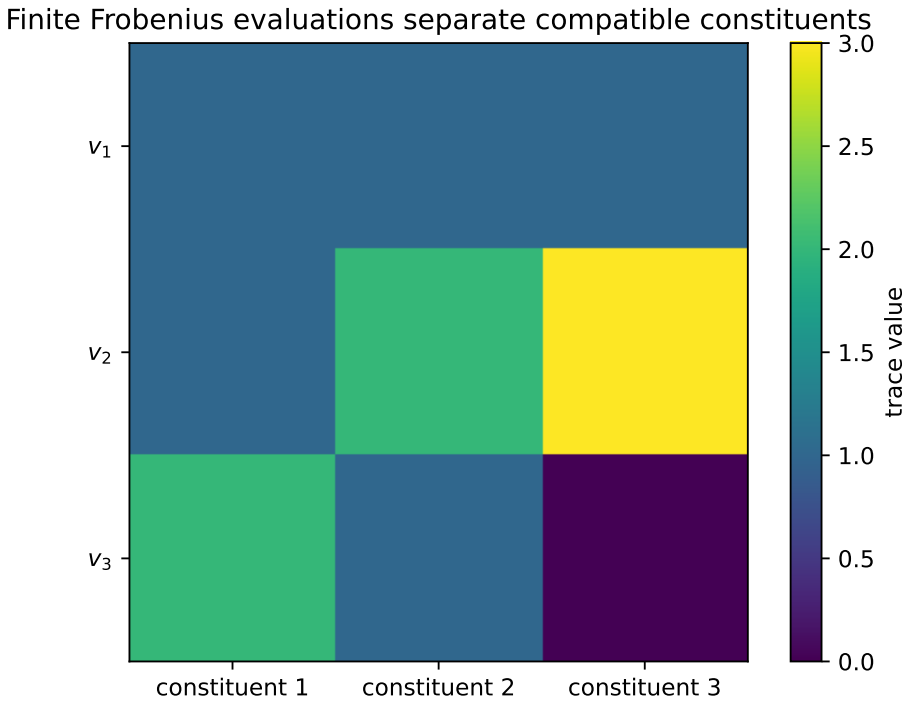


Figure 14. A finite set of trace evaluations separates three compatible constituents and recovers their multiplicities.

Numerical Certificate 9.3 (Multiplicity recovery)**.** The original companion matrix was not retrieved or rerun. A new exact realization uses the trivial character and two independent sign characters of $(\mathbf{Z}/2\mathbf{Z})^2$, evaluated at $(0,0), (1,0), (0,1)$:

$$C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}, \qquad \det C = 4, \qquad C \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}.$$

Lean checks that these are the traces of actual one-dimensional representations, recovers $(2,0,1)$ uniquely, rejects a target whose unique solution is $(1/2,0,0)$, and rejects $(4,0,1)$ under capacities $(3,2,2)$. The separator becomes singular modulo two. These are finite-image representation checks, not fabricated Frobenius data for an arithmetic carrier.

The theorem is substantial only when a suitable compatible carrier and its irreducible compatible constituents are already known. It does not assert that every geometric representation embeds into such a carrier.

Finite-exponential reconstruction and a cubic inverse. Let K be a characteristic-zero field and let (r, N) be supplied group data with nilpotent N and $r(g)N = \nu(g)Nr(g)$. For a supplied scalar map satisfying

$$t(gh) = t(g) + \nu(g)t(h),$$

the formalization constructs

$$\rho(g) = \exp(t(g)N)r(g), \qquad \exp(aN) = \sum_{j=0}^{d-1} \frac{a^j}{j!}N^j \quad (N^d = 0).$$

Lean proves the representation laws, actual finite-copy and split-map compatibility, and joint continuity from continuity of N, r and t with the stated continuous vector-space operations. The monodromy fixed kernel intersected with subgroup invariants is unchanged by this twist. For normal inertia, its restricted action and finite-dimensional Euler denominator agree. Twisting again by $-t$ recovers the datum with N still supplied.

The finite-exponential reconstruction also established an extracted inverse on the explicit class $N^3 = 0$:

$$\log_2 U = (U - I) - \tfrac{1}{2}(U - I)^2, \qquad a^{-1}\log_2(\exp(aN)) = N \quad (a \neq 0).$$

Both exponential/logarithmic inverse identities are checked on the cubic nilpotent and unipotent loci. If a supplied common sample u has $r_D(u) = r_E(u) = I$ and $t(u) \neq 0$, an intertwiner of the reconstructed cubic actions recovers both original group equivariance and monodromy equivariance. The general nilpotence exponential theorem is not restricted to this cubic class, but the extracted inverse in the finite-exponential reconstruction was. The finite logarithmic recovery argument below removes that restriction by using the full finite logarithm.

The rational affine-group example has $u = (1, 1)$, $d = (0, 1/3)$ and

$$\rho(u) = \begin{pmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \rho(d) = \operatorname{diag}(1, 3, 9).$$

Reversing the factor order or dropping the quadratic term gives checked failures. For the length-four chain, the quadratic logarithm has erroneous e_3-to-e_0 coefficient $-1/3$, so its cubic restriction is substantive. A zero-parameter sample also fails to recover N.

No arithmetic group, tame cocycle or required sample is constructed by these statements. The attached comparison for every intended representation, coefficient comparisons and Hodge–Tate/period data remain separate. Finite inertia image and an open inertia kernel of r are not asserted to persist under this twist to ρ. Claim 9.2 remains partial.

10 Finite spectral occupancy

System completion produces arithmetic members. It does not prove that a desired packet occurs in a global carrier. In a fixed finite-dimensional semisimple carrier, however, occupancy is an exact projector question.

Let E be a field of characteristic zero and let $T_1, \dots, T_N$ be commuting operators on a finite-dimensional E-vector space A, each diagonalizable over E. Thus the selected spectra split over the coefficient field. Let $a_i \in E$ be a selected target scalar for T_i, possibly absent from its spectrum. Define

$$e_i = \prod_{\mu \in \mathrm{Spec}(T_i) \setminus \{a_i\}} \frac{T_i - \mu I}{a_i - \mu}, \qquad e_T = \prod_{i=1}^{N} e_i,$$

with $e_i = 0$ if a_i is absent.

Theorem 10.1 (Exact finite occupancy trace)**.** *The operator e_T is the projector onto the joint eigenspace*

$$A_T = \{v : T_i v = a_i v \text{ for every } i\}.$$

Moreover,

$$\boxed{\mathrm{Tr}(e_T) = \mathrm{rank}(e_T) = \dim A_T \in \mathbf{Z}_{\geq 0}.}$$

The trace is identified with its unique nonnegative integer preimage under $\mathbf{Z} \hookrightarrow E$. Thus $\mathrm{Tr}(e_T) > 0$ refers to that integer, not to an ordering of an arbitrary field E. The finite packet is occupied exactly when this integer is positive. In positive characteristic the field-valued trace can vanish for a nonzero projector. The rank test, not a positive field trace, must then be used.

Proof. For each operator, its finite set of actual eigenvalues is used without multiplicity. The polynomial

$$p_{i,a}(X) = \prod_{\mu \in \mathrm{Spec}(T_i) \setminus \{a\}} (a - \mu)^{-1}(X - \mu)$$

is used when a occurs, and the zero polynomial otherwise. For an eigenvector $T_i v = bv$, evaluation gives

$$p_{i,a}(T_i)v = \begin{cases} v & b = a, \\ 0 & b \neq a. \end{cases}$$

Because the eigenspaces span A, this proves that $e_i = p_{i,a_i}(T_i)$ maps all of A into the selected eigenspace and fixes that eigenspace pointwise. It therefore proves the individual projector and image assertions, including an absent target.

Polynomials in commuting operators commute. If projections P, Q onto U, W commute, then PQ maps into both U and W, and fixes every vector in $U \cap W$. Induction proves that the actual ordered product e_T is a projection onto A_T. Commutation also makes the result independent of the ordering. An empty list of conditions gives the identity.

For a projection onto a subspace, the direct sum of its image and kernel identifies it with $I \oplus 0$. Thus its trace in E is the natural dimension of its image cast into E. In characteristic zero this cast is injective, so there is a unique natural integer represented by the trace. That integer is positive exactly when the joint eigenspace contains a nonzero vector. □

Naturality and finite hypotheses. If $FT_i = U_i F$ and the source operators split, then $Fe_{T,a} = e_{U,a}F$. This follows first on source eigenvectors and then on their span. In particular, a change of basis conjugates the constructed projector and preserves its trace. Adding conditions decreases the actual joint eigenspace and cannot increase its dimension. These statements do not assume an orthonormal basis or a positive definite inner product.

A specified finite realization. On $\mathbf{Q}^4$, take the diagonal operators with diagonal lists

$$(1,1,1,0), \quad (2,2,0,0), \quad (3,0,0,0), \quad (0,4,4,4)$$

and target scalars $(1,2,3,4)$. All four targets occur individually. Imposing these conditions successively gives actual projector ranks $3,2,1,0$. The last packet is empty despite individual feasibility. The formal code checks these operators, projector formulas, traces and dimensions. This is a newly specified realization, not a claim to have recovered the original companion implementation.

Why the hypotheses cannot be omitted. For

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad Q = \tfrac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

each operator is idempotent, but PQ is not idempotent and has trace $1/2$. For the Jordan block $J = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, the singleton-spectrum Lagrange polynomial is 1, so the formula produces the identity although the eigenspace at 1 is only a line. Both failures are exact formal counterexamples to dropping the corresponding hypotheses, not counterexamples to the theorem with those hypotheses retained.

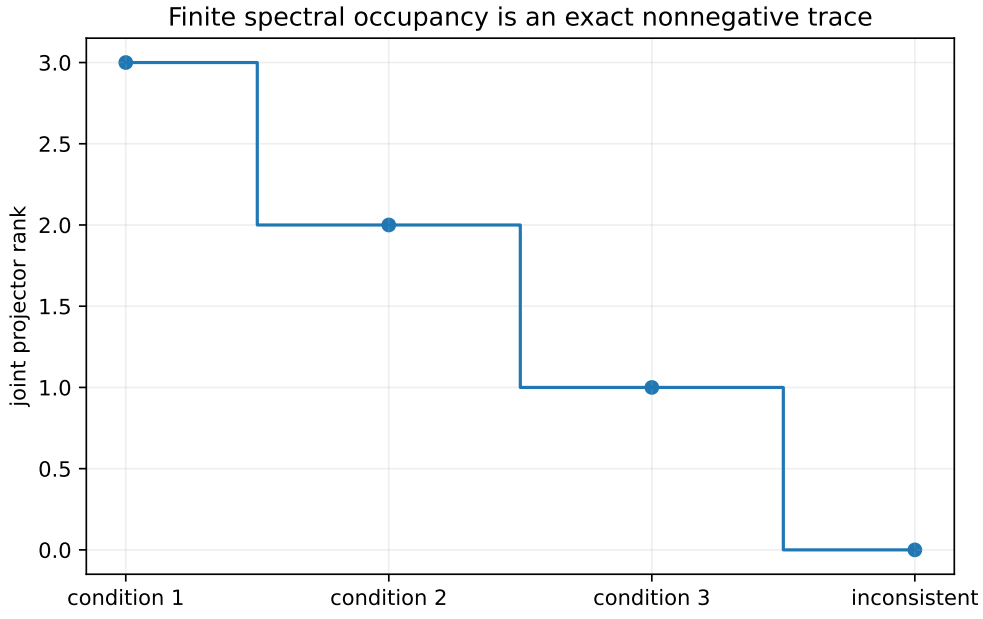


Figure 15. A consistent sequence of finite conditions narrows the carrier from ranks 3 to 2 to 1. One inconsistent condition forces the joint projector to zero.

Conjecture 10.2 (Universal occupancy)**.** For every admissible finite arithmetic packet in the intended global class, there exists a finite-dimensional carrier in which the associated joint projector has positive trace.

This conjecture is not a consequence of the projector formula. The formula measures occupancy exactly once the carrier is fixed. It cannot force the trace to be positive.

11 A conditional system-completion criterion

The preceding results can be assembled without promoting their missing hypotheses to theorems.

Theorem 11.1 (Residual-system completion criterion)**.** *Fix an admissible arithmetic class. Suppose:*

1. *every object has a unique residual root in an exhaustive residual moduli space;*
2. *a common algebraic spine is defined for the desired system;*
3. *every coefficient-prime residual companion exists;*
4. *every companion forest has nonempty finite levels, finite branching, and inverse-limit closed local conditions;*
5. *the required horizontal transports preserve the common spine and the selected local deformation conditions.*

Then the desired strictly compatible arithmetic system exists. If, in addition, the system is a summand of a known compatible carrier, the summand is recovered at every coefficient prime by finitely many Frobenius traces.

Proof. The first four hypotheses give the system by adelic finite-depth completion. The transport hypothesis identifies any roots that are meant to be generated horizontally without changing the prescribed spine. The final assertion is the compatible-carrier summand theorem. □

Corollary 11.2 (Exact location of the remaining existence problem)**.** *Inverse-limit reconstruction alone is not a nonemptiness theorem. The existence requirements must retain both residual companions and liftability through every finite depth:*

$$\boxed{\begin{array}{l}\text{companion existence: } X_{\lambda,1}(\mathfrak{C}) \neq \varnothing \text{ for every } \lambda;\\ \text{finite-depth liftability: } X_{\lambda,m}(\mathfrak{C}) \neq \varnothing \text{ for every } \lambda, m \geq 2;\\ \text{finite occupancy: } \operatorname{Tr}(e_T) > 0 \text{ for every required global packet.}\end{array}}$$

Formal reconstruction does not create a missing residual root, repair an empty higher level, or force global occupancy. Nonempty finite levels with finite branching supply a branch, after which inverse-limit closure reconstructs an admissible object.

An occupied node can have no child. The actual order-two representation with generator

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \pmod 4$$

satisfies $A^2 = I$ and $\det(XI - A) = X^2 - 1$. It has no lift modulo 8 with the same characteristic polynomial. Indeed, every lift has entries congruent to $1, 2, 2, 3$ modulo 4. Trace zero then forces its determinant to be 3 modulo 8, whereas $X^2 - 1$ requires determinant $-1 = 7$ modulo 8. Both

the occupied node and the absence of any next-level matrix or representation with the prescribed polynomial are Lean-checked. Other branches of the same unrestricted involution forest survive, so this example does not contradict finite-branch compactness.

Local properties need their own limit argument. Over a domain, for a nonzero $\pi \in I$, the representations of the additive integer group given by

$$\rho_n(g) = \begin{pmatrix} 1 & \pi^n g \\ 0 & 1 \end{pmatrix}$$

are all nontrivial and converge pointwise in the ideal-adic topology to the trivial representation. The convergence and the failure of limit preservation for nontriviality are formally checked. Thus not every property of approximating integral representations can be asserted for a limit. This example is not claimed to disprove stability for any particular intended Galois local condition.

Complete conditional existence boundary. The eleventh block established the corresponding conditional matrix criterion. The conditional existence-boundary argument now uses the complete common-spine conclusion of Claim 8.2: the returned coefficient-field representations are continuous, have the prescribed common polynomials and coefficient-specific determinants, and satisfy every selected local realization relation under the stated inverse-limit closure hypothesis. All finite reductions belong to the admissible classes.

For each supplied required packet p, let A_p be its finite-dimensional carrier over a characteristic-zero field E_p, and let e_p be the actual joint spectral projector for its finite list of split commuting conditions. No order on E_p is assumed. The source notation $\mathrm{Tr}(e_p) > 0$ means that the unique natural number n_p with $\mathrm{Tr}(e_p) = (n_p : E_p)$ is positive. The checked projector formula identifies n_p with the dimension of the actual joint eigenspace.

An admissible realization here includes continuous coefficient-field members and membership of every finite reduction in its designated admissible class. Under the completion hypotheses and these supplied carrier hypotheses, Lean proves

$$\begin{aligned} &\exists\, \rho,\ (x_p)_p : \quad \rho \text{ is an admissible realization of the full chosen spine}, \quad 0 \neq x_p \in \mathrm{im}(e_p) \text{ for every } p \\ &\quad \Longleftrightarrow \quad \big(\forall \lambda,\ X_{\lambda,1}(\mathfrak{C}) \neq \varnothing\big) \ \wedge\ \big(\forall \lambda,\ \forall m \geq 2, X_{\lambda,m}(\mathfrak{C}) \neq \varnothing\big) \ \wedge\ \big(\forall p,\ n_p > 0\big). \end{aligned}$$

The packet index may be infinite. Each x_p belongs to its own supplied carrier. No common vector across distinct carriers or embedding of the reconstructed system into those carriers is asserted. Neither an unknown carrier nor positive trace is constructed without the corresponding hypotheses.

This completes Claim 11.2 at its conditional source scope, on the same counting basis used for Claim 8.2 in the conditional completion construction. The full arithmetic application premises remain separate. In particular, the criterion does not prove universal occupancy, Hodge–Tate or Weil–Deligne attachment, or the Frobenius recovery clause of Claim 11.1. The latter still depends on the partial Claims 9.1 and 9.2.

Every required packet is necessary. For each omitted index $N \in \mathbb{N}$, consider one-dimensional rational carriers with the identity operator, requiring eigenvalue 0 at N and eigenvalue 1 at every other index. The actual spectral projector has trace 0 at N and trace 1 elsewhere. Lean proves that any predetermined finite sample of packet indices can miss an empty required packet: choose N outside

that sample. The family depends on the omitted index. This does not assert that one fixed family passes every finite prefix while failing global occupancy.

Conjecture 11.3 (Natural finite seed rank)**.** For the admissible residual transport category attached to a fixed arithmetic class, the directed seed rank is finite. A one-seed theorem additionally requires a source component from which every required object is reachable and transports preserving deformation and occupancy data. For a finite condensation DAG, having exactly one source implies that reachability condition. For an infinite category, the existence of a unique source alone does not establish it.

12 Worked finite models

12.1 Smooth branch

For $q = 3$ and formal dimension $d = 2$, the finite levels have cardinalities

$$1, 9, 81, 729, 6561, \dots.$$

The seed zeta function is

$$Z(T) = \frac{T}{1 - 9T}.$$

12.2 Weighted dyadic chart

For weights $(1, 2, 4)$, the counts are

$$1, 2, 8, 32, 256, 2048, \dots.$$

The first coordinate is active immediately, the second one level later, and the third at depth five. The exact rational function generated by the code is

$$Z_{(1,2,4)}(T) = \frac{T(1 - 6T - 8T^2 - 32T^3)}{1 - 8T}.$$

The negative signs in this presentation arise from moving the denominator to $1 - 8T$. Every power-series coefficient is positive.

12.3 Singular branch

For $x^2 = 0$ over a residue field of size three, the counts are

$$1, 3, 3, 9, 9, 27, 27, 81, \dots,$$

and

$$Z(T) = \frac{T + 3T^2}{1 - 3T^2}.$$

The exact branching dimension is $1/2$.

12.4 Directed transport

The finite graph in figure 10 has three source components. The source-component theorem and brute-force enumeration both return the minimum seed count three.

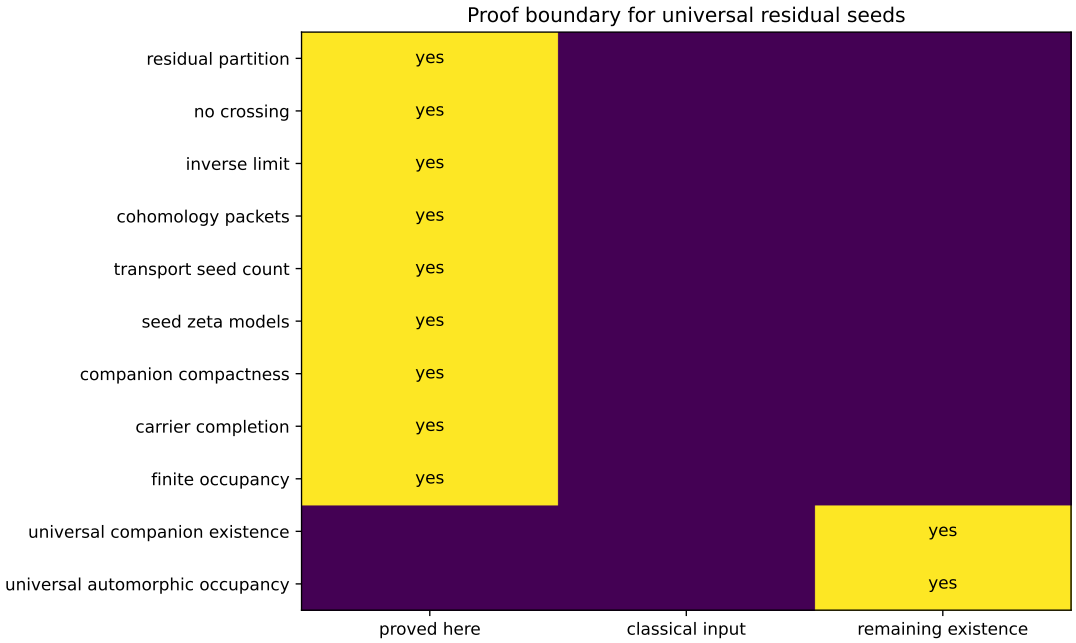


Figure 16. The finite and formal statements proved in this chapter are separated from the two remaining universal existence assertions.

13 Computational certificates

The code uses exact integer, rational, symbolic, and modular arithmetic for all mathematical identities. Floating-point arithmetic is used only to render the figures and to report a regression estimate alongside an exact branching dimension.

Table 1. Selected exact certificates.

Certificate	Result
Residual packet count in the mod-three model	3
Compatible base-five inverse-limit branch	yes
First dyadic sign separation depth	2
Source strongly connected components	3
Brute-force minimum directed seed count	3
Cohomology packet reconstruction residual	0
Compatible common-spine test	passed
Inconsistent spine detected	yes
Recovered carrier multiplicities	$(2, 0, 1)$
Finite occupancy ranks	$(3, 2, 1, 0)$

The complete automated suite contains 28 tests. They cover residual partition, no crossing, inverse-limit digits, weighted and singular zeta coefficients, directed source components, cohomology packets,

companion forests, common-spine consistency, finite Frobenius separation, multiplicity recovery, and exact occupancy projectors.

14 Claim ledger

Statement	Status	Scope or boundary
Residual partition	proved	any moduli functor with well-defined reduction
No crossing	proved	operations trivial at residual depth
Inverse-limit reconstruction	proved	represented complete deformation problem
Finite-branch compactness	proved	finite nonempty levels and finite branching
Smooth and weighted seed zeta formulas	proved	stated formal coordinate models
Singular power zeta and dimension	proved	scalar equation $x^e = 0$
Cohomology packet decomposition	proved	chain projectors commuting with the differential
Directed minimum-seed theorem	proved	finite directed transport graph
Adelic finite-depth completion	proved	common spine, all finite levels occupied, closed local conditions
Compatible-carrier summand completion	proved	known semisimple compatible carrier and finite separation
Finite occupancy trace	Lean checked	supplied commuting split operators and a characteristic-zero trace criterion
Universal residual companions	not proved	requires nonempty root at every coefficient prime
Universal compatible-system completion	not proved	algebraic spine and companion occupancy remain necessary
Universal automorphic occupancy	not proved	requires positive global packet traces
Unrestricted reciprocity	not claimed	belongs to later local and global chapters

15 Conclusion

The residual seed theory becomes precise once vertical and horizontal generation are separated. Positive-depth perturbations can be complete within one formal component. They cannot change the residual root. The universal arithmetic seed is therefore a forest rather than one branch:

$$\boxed{\text{residual roots} \longrightarrow \text{finite deformation levels} \longrightarrow \text{inverse-limit branches.}}$$

Projectors add three useful structures to this forest. They split tangent and obstruction complexes into exact subproblems, they measure delayed dyadic visibility through weights, and they turn finite carrier occupancy into an integer trace. Directed transport adds a fourth: it converts the informal question "how many seeds are needed?" into the source-component count of a condensation graph.

The system-level boundary is now narrow enough to state without ambiguity. One must construct a common algebraic spine, populate every residual companion forest, lift those companions through all depths, and prove positive occupancy of the required global packets. The present chapter proves the formal reconstruction, counting, transport, compactness, carrier, and finite-projector theorems used in that reduction. It does not assert the missing universal existence statements.

A Algorithms implemented in the companion package

A.1 Residual towers

The residual module partitions coordinates modulo p, verifies compatibility

$$x_{m+1} \equiv x_m \pmod{p^m},$$

and reconstructs the digit a_{m-1} by

$$a_{m-1} = \left\lfloor \frac{x_m}{p^{m-1}} \right\rfloor \pmod{p}.$$

For canonical representatives $0 \le x_m < p^m$, set $x_0 = 0$. Compatibility also gives $a_{m-1} = (x_m - x_{m-1})/p^{m-1} \in \{0, \dots, p-1\}$. The floor in the first formula is essential. For example, the compatible base-five residues $4, 19$ give the second digit 3, not the rational number $19/5$. The executable companion uses the same integer floor division.

A.2 Transport condensation

The transport module computes strongly connected components by two depth-first passes, forms the condensation graph, finds all source components, and selects one representative from each. A brute-force subset search is included only as an independent finite certificate.

A.3 Companion forests

For an integral polynomial f, the finite root forest is obtained by exact enumeration of

$$f(x) \equiv 0 \pmod{p^m}$$

and by connecting a child x_{m+1} to its reduction modulo p^m. This is intended for small exact examples, not large deformation rings.

A.4 Cohomology packets

Modular Gaussian elimination computes the ranks of the differentials. For

$$C^{i-1} \xrightarrow{d_{i-1}} C^i \xrightarrow{d_i} C^{i+1},$$

the finite dimension is evaluated as

$$\dim H^i = \dim C^i - \operatorname{rank} d_i - \operatorname{rank} d_{i-1}.$$

Packet subcomplexes are computed separately and checked against the total.

A.5 Finite occupancy

For each commuting semisimple operator, a Lagrange polynomial constructs the exact spectral idempotent. Their product gives the joint packet projector. Symbolic rational arithmetic verifies idempotence and the equality of trace and rank.

B Reproducibility boundary

The companion code is a finite certificate library. It does not construct universal Galois deformation rings, compute general continuous group cohomology, prove representability, discover compatible systems, or evaluate automorphic trace formulas. Those are mathematical inputs or later existence problems. The source package makes the finite examples and displayed counts reproducible. Their computational checks remain distinct from universal proofs.

Further construction details

The following constructions explain the representation, comparison and local arithmetic steps used in the chapter. Appendix D assembles their conditional endpoint. The formal companion keeps the distinct source contracts and verification records separate.

The formal model and its hypotheses. The smooth formal-model classification, its finite point count, its integral digit streams, and its rational seed zeta identity are now checked in Lean. The actual representing-ring algebra isomorphism remains a supplied hypothesis. Neither smoothness nor Galois representability is inferred from numerical counts. All local maps are included in the classification, and locality and actual maximal-ideal-adic continuity are separately proved. Arbitrary choices of parameter representatives do not add phantom maps.

General prescribed-conjugacy prime coverage remains an input. Claims 8.2 and 11.2 retain their conditional hypotheses, including the required residual levels, higher depths and supplied packets. Their nonemptiness and positive trace are not unconditional conclusions. The finite cyclotomic comparison and the fixed quadratic packet give restricted arithmetic realizations. In the latter, the packet characters are trivial on actual valuation inertia at odd primes. Identifying the decomposition group with the full local Galois group remains a separate obligation, and ramification at two is not claimed. The constructions do not identify every packet in the common arithmetic spine or attach all Hodge–Tate and Weil–Deligne invariants. Universal occupancy and natural finite seed rank remain conjectural.

Finite-group comparison with precision loss. Let R be a commutative domain, Γ a finite group, and ρ, τ actual integral matrix representations. Suppose $a = |\Gamma| \neq 0$ in R and $a \mid \pi^e$. An equivariant matrix modulo $(\pi)^{e+k}$ lifts to an integral equivariant matrix with the prescribed reduction modulo $(\pi)^k$. Indeed, lift its entries to A and form

$$S = \sum_{g \in \Gamma} \rho(g) A \tau(g^{-1}).$$

The finite equations imply $S - aA = \pi^{e+k}E$ with E integral. Choose b with $\pi^e = ab$ and put $P = A + b\pi^k E$. Then $aP = S$, so cancellation of $a \neq 0$ proves equivariance. Over a local ring, if π is a nonunit, $k > 0$ and the supplied comparison is invertible, P is an integral unit. Neither completeness nor a finite residue field is needed for this step.

For a DVR, $e = v_\pi(|\Gamma|)$ is derived from the actual group order. The single precision $e + 1$, independent of rank, detects integral isomorphism. It makes reduction injective on integral isomorphism classes, and finite coefficient quotients give finitely many integral classes in fixed rank. This is a formalization of classical finite-group rigidity (Maranda, 1953, Theorem 2), with an explicit proof of the comparison's precision loss.

For complete local coefficients with finite quotients, downward-compatible allowed representations occupied at every precision, and the required finite-group type at depth $e+1$, reconstruction followed by this comparison produces the required integral representation. Continuity is proved for a finite discrete source and for pullback through a fixed continuous map to that source. Occupancy at only one precision is insufficient. The result lifts comparisons between actual integral representations. It does not lift arbitrary quotient representations or supply the missing arithmetic packet interpretation.

For C_2 over $\mathbb{Z}_2$, take

$$\tau(g) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \rho(g) = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

The matrix A intertwines modulo 4 and its average is $P = \left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right)$. No exact intertwiner can retain A modulo 4, although P retains it modulo 2. The actual trivial and sign representations also agree modulo 2 but are not integrally isomorphic. These checked examples show why residual-only detection and full preservation of the input precision cannot be asserted here.

Lifting a summand inside a supplied finite-group carrier. Let R be a commutative integral domain complete and separated for the (π)-adic filtration. Let Γ be a finite group, and let $\rho : \Gamma \to \mathrm{GL}_n(R)$ be an actual integral representation. Assume that $a = |\Gamma| \neq 0$ in R and $a \mid \pi^e$. For $k > 0$, suppose

$$\bar{E} \in M_n\big(R/(\pi)^{e+k}\big), \qquad \bar{E}^2 = \bar{E}, \qquad \bar{\rho}(g)\bar{E} = \bar{E}\bar{\rho}(g) \quad (g \in \Gamma).$$

There is an actual integral matrix E satisfying

$$E^2 = E, \qquad \rho(g)E = E\rho(g), \qquad E \bmod (\pi)^k = \bar{E} \bmod (\pi)^k.$$

No integral summand or coherent family of quotient projectors is assumed. No local-ring assumption or finiteness of the residue field is needed.

To prove this, apply the preceding averaging correction with both representations equal to ρ. This produces an exact equivariant matrix A with the prescribed reduction at depth k, so that $A^2 - A \in (\pi)^k M_n(R)$. Averaging alone need not preserve idempotence. Use the integral polynomial

$$f(A) = 3A^2 - 2A^3.$$

The identities

$$f(A) - A = (1-2A)(A^2-A), \qquad f(A)^2 - f(A) = (A^2-A)^2(4A^2-4A-3)$$

show that successive corrections preserve the retained reduction and improve the idempotence defect. In these formulas scalar constants denote scalar matrices. The Lean proof establishes at least one additional ideal digit at each step, constructs the resulting compatible family of finite-level matrices, and uses completeness and separatedness to recover E. Every matrix commuting with A commutes with E. There is no division by 2, so the correction also works in residue characteristic 2.

The actual submodules $U = \operatorname{im} E$ and $W = \ker E$ are invariant, and addition gives an equivariant isomorphism

$$U \oplus W \simeq R^n.$$

The inclusion $i : U \hookrightarrow R^n$ and retraction $r(v) = Ev$ are constructed as linear maps and satisfy $ri = 1_U$ and $ir = E$. Their equivariance is proved. Coordinatewise reduction maps U onto $\mathrm{im}(\bar{E} \bmod (\pi)^k)$ and maps W onto $\ker(\bar{E} \bmod (\pi)^k)$. Both equalities are surjectivity statements onto the actual submodules over the finite-level quotient, not merely containment. The same projector defines a summand after pullback through any group homomorphism to Γ.

For $R = \mathbb{Z}_2$ and the sign carrier $\rho(s) = \mathrm{diag}(1, -1)$, the projector

$$\bar{E} = \begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix} \pmod 4$$

is equivariant and the theorem constructs a nonzero proper equivariant projector retaining its reduction modulo 2. No exact equivariant endomorphism retains this prescribed matrix modulo 4. At the insufficient input depth 1, the projector $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right) \pmod 2$ has no exact equivariant lift retaining its reduction. These are statements about the supplied carrier and prescribed projectors, not about nonexistence of all integral summands.

This construction gives one summand and its complement inside a supplied finite-group carrier. It advances the carrier step in Claim 9.2. It does not identify the intended arithmetic packets, construct the arithmetic Weil–Deligne attachment, establish the Hodge–Tate comparisons, or prove universal occupancy.

Simultaneous projector-family lifting and rigidity. Keep the complete domain, supplied finite-group carrier and group-order hypotheses of the supplied-carrier summand-lifting construction. Let J be any finite label set and $k > 0$. Suppose $(\bar{E}_i)_{i \in J}$ is a complete orthogonal family of equivariant idempotents modulo $(\pi)^{e+k}$. There exists an integral equivariant family $(E_i)_{i \in J}$ satisfying

$$E_i^2 = E_i, \qquad E_i E_j = 0 \ (i \neq j), \qquad \sum_{i \in J} E_i = 1, \qquad E_i \bmod (\pi)^k = \bar{E}_i \bmod (\pi)^k.$$

The loss is at most e, independent of $|J|$ and the ambient matrix rank. There is no local-ring or finite-residue-field hypothesis for existence. The formal theorem also treats empty label sets whenever their input equations can hold.

Average every supplied matrix once. Inductively let C be the complement of the previously lifted projectors and correct the next averaged matrix inside the corner defined by C. The integral polynomial correction preserves the retained reduction and commutation with C and the carrier. Multiplication by C gives the required supported projector. This consumes no further precision. Once every label has been lifted, the complementary idempotent reduces to zero at positive depth. Adic separatedness forces it to vanish, proving the sum-to-identity equation.

The actual invariant images $U_i = \mathrm{im}\, E_i$ give an equivariant linear isomorphism

$$\bigoplus_{i \in J} U_i \simeq R^n, \qquad (v_i)_i \longmapsto \sum_i v_i, \qquad v \longmapsto (E_i v)_i.$$

Lean represents the finite direct sum by the finite product of the image submodules. The existing coordinatewise reduction theorem maps every U_i onto the corresponding reduced projector image. No chosen bases or unproved freeness of the images are used.

For rigidity alone, let R be a commutative local ring, let π be a nonunit, and let E, F be complete equivariant families on the same carrier with $E_i \equiv F_i \pmod{(\pi)^k}$ for every label and $k > 0$. Then

$$U = \sum_{i \in J} F_i E_i, \qquad U \equiv 1 \pmod{(\pi)^k}, \qquad UE_i = F_i U.$$

Reduction reflects invertibility in this setting, so U is an integral unit and $UE_iU^{-1} = F_i$. It commutes with the carrier and induces actual equivariant isomorphisms $\operatorname{im} E_i \simeq \operatorname{im} F_i$. This rigidity theorem requires neither a finite acting group, a domain, nor completeness. More generally a unit-reflecting coefficient map suffices. The combined result proves existence and uniqueness up to this labeled conjugacy. Exact uniqueness of the matrices is false.

For the continuous $\mathbb{Z}_2$ carrier $\rho(s) = \operatorname{diag}(1, -1, 1)$, conjugate the three coordinate projectors by $1 + 2e_{01}$ and reduce modulo 4. They form a complete equivariant family. The general theorem constructs three nonzero integral summands retaining this family modulo 2. No exact equivariant family retains the whole prescribed family modulo 4. Separate checked examples show that independent equivariant idempotent lifts can lose orthogonality, and that distinct complete families with the same reduction can be related by the asserted identity-congruent conjugacy.

The simultaneous projector-family lifting construction completes this finite-group projector-family step. The actual arithmetic packets, Galois-to-Weil–Deligne attachment, coefficient and Hodge–Tate/period comparisons, and arithmetic Frobenius coverage remain outside this result. Universal occupancy remains conjectural.

General finite logarithmic recovery. Let A be a rational algebra, without any commutativity assumption on A. For a unipotent element U and a positive integer d satisfying $(U - 1)^d = 0$, put

$$L_d(U) = \sum_{j=1}^{d-1} \frac{(-1)^{j+1}}{j} (U - 1)^j.$$

The value is independent of the chosen annihilating exponent. The finite logarithm and the nilpotent exponential satisfy

$$L(\exp N) = N, \qquad \exp(L(U)) = U.$$

Both maps preserve every annihilating exponent, and hence preserve the exact nilpotency class of N and $\exp(N) - 1$, or of $U - 1$ and $L(U)$. The Lean finite sum includes its zero-index term with coefficient zero. The displayed mathematical sum starts at one. The formal statements also handle the zero algebra and the exponent-zero case.

The proof evaluates Mathlib's formal power-series inverse identities at constant nilpotent series. Such evaluation is exactly a finite polynomial. For an arbitrary noncommutative ambient algebra, perform the calculation inside the commutative subalgebra generated by the element and then apply its inclusion. No analytic convergence or topology is assumed. The finite logarithm commutes with ring homomorphisms between rational algebras.

Let K be a characteristic-zero field and N an endomorphism with $N^d = 0$. For $t \neq 0$ the actual recovery formula is

$$N = \frac{1}{t} \sum_{j=1}^{d-1} \frac{(-1)^{j+1}}{j} \big(\exp(tN) - 1\big)^j.$$

All coefficients lie in K. The vector spaces need not be finite dimensional, but nilpotence still means that some finite power of the whole operator is zero. A linear map intertwines two nilpotent operators exactly when it intertwines their exponential samples at a common nonzero parameter. Also

$$\ker\bigl(\exp(tN) - 1\bigr) = \ker N.$$

For supplied covariance data and the twisted cocycle of the finite-exponential reconstruction, let $\rho_D(g) = \exp(t(g)N_D)r_D(g)$. If a common element u satisfies $r_D(u) = r_E(u) = 1$ and $t(u) \neq 0$, every intertwiner of ρ_D and ρ_E recovers both the original r-intertwiner and the N-intertwiner. There is no cubic nilpotence restriction. This recovers equivariant projectors for both original components. Reconstructed-action isomorphisms induce actual isomorphisms of the inertia-fixed monodromy kernels. Their local characteristic polynomials agree under the existing finite-dimensional and normal-subgroup hypotheses.

For actual finite original inertia images the common kernel sample is derived. Suppose $u \in H$ satisfies $v(u) = 1$ and $t(u) \neq 0$ and both $r_D(H)$ and $r_E(H)$ are finite. Their sampled actions have positive finite orders a, b. At u^{ab} both original actions are trivial, and

$$t(u^{ab}) = (ab)t(u) \neq 0$$

by the cocycle identity and characteristic zero. Thus the same full intertwiner equivalence holds using the actual finite-image hypotheses, without a separately supplied common trivial-action sample. The initial nonzero cocycle sample and the arithmetic interpretation of the data remain requirements.

Actual affine representations with monodromy chains of every finite length invoke these theorems. In particular a length-six example has $N^5 \neq 0$ and $N^6 = 0$ and is recovered from its actual reconstructed action. The length-four example still defeats the old quadratic logarithm, while the full finite logarithm recovers it.

The sample hypotheses retain substantive content. A zero parameter loses N. For unrestricted supplied algebraic data, one can also absorb $\exp(tN)$ into r and set the new monodromy to zero. The reconstructed representation is unchanged. The absorbed original action fails the required trivial-action sample condition. This is a counterexample in the algebraic datum class, not a failure of uniqueness of arithmetic Weil–Deligne attachment with finite original inertia action.

The finite logarithmic recovery argument removes the general logarithmic-extraction gap from the local algebraic comparison. It does not construct the actual arithmetic attachment, identify every intended packet, prove the coefficient and Hodge–Tate/period comparisons, or prove arithmetic Frobenius coverage.

Interpreted quadratic arithmetic completion. Fix the continuous quotient

$$q : G_{\mathbf{Q}} \longrightarrow C_2$$

obtained by the action on a chosen square root i of -1 in the algebraic closure. Its surjectivity and the equivalence

$$q(\sigma) = 1 \quad \Longleftrightarrow \quad \sigma(i) = i$$

are proved using the existing actual quadratic Galois construction. For a commutative ring B, define the arithmetic packet

$$\mathcal{P}_q(B) = \{\chi : G_{\mathbf{Q}} \longrightarrow B^{\times} \mid \ker q \subseteq \ker \chi\},$$

where the maps are group homomorphisms. Unique descent through the actual surjective map q is derived, rather than supplied. Every member has finite image and is continuous for the coefficient topology, since its descent has the finite discrete domain C_2.

Let R be any commutative ring and B any commutative R-algebra. The coordinate ring $A_R = R[X]/(X^2 - 1)$ represents precisely this specified packet:

$$\mathrm{Hom}_{R\text{-alg}}(A_R, B) \simeq \{x \in B : x^2 = 1\} \simeq \mathrm{Hom}(C_2, B^\times) \simeq \mathcal{P}_q(B).$$

The first map evaluates the coordinate. Its inverse is the quotient universal property. A root is a unit with itself as inverse and determines a character by its value on the generator of C_2. Finally pull back along q. All inverse identities and naturality under coefficient homomorphisms are proved. In particular, the point reductions and actual arithmetic character reductions coincide. There is no invertibility-of-two, reducedness or nontriviality hypothesis on B. No smoothness or étaleness statement at two is used or asserted.

Let $I \subseteq R$ and suppose that R is I-adically complete. Using the checked ring-map reconstruction on the quadratic coordinate ring, Lean proves

$$\mathcal{P}_q(R) \simeq \varprojlim_{m \geq 0} \mathcal{P}_q(R/I^m).$$

The zero-precision quotient is included only for the formal indexing. The inverse constructs an actual arithmetic character, with exactly the prescribed reductions. It is unique, and its continuity is derived from its factorization through the same finite quotient.

There is also a completion theorem without chosen compatible characters. Suppose that R/I^m is finite for every $m \geq 1$, and that $C_m \subseteq \mathcal{P}_q(R/I^m)$ are allowed subsets such that every reduction of an allowed member is allowed. Then

$$\begin{aligned} &\exists \chi \in \mathcal{P}_q(R), \quad \chi \text{ is continuous and } \chi \bmod I^m \in C_m \text{ for all } m \geq 1 \\ &\qquad \Longleftrightarrow \quad C_m \neq \varnothing \text{ for all } m \geq 1. \end{aligned}$$

The root interpretation makes the allowed finite levels genuinely finite. Finite-branch compactness chooses the compatible roots, and the ring reconstruction gives their integral root. Naturality turns this into the required arithmetic character. Surjectivity of transitions and individual liftability of every finite node are not assumed. The root-level theorem needs only the allowed root subsets to be finite.

For $R = \mathbf{Z}_2$, Lean derives finiteness of all positive maximal-ideal quotients and applies this completion theorem. Since $\mathbf{Z}_2$ is a domain, its only roots of $X^2 - 1$ are 1 and -1. Thus the integral packet has exactly two distinct characters: the trivial character and the sign character describing the action on i. They have the same reduction modulo two and distinct reductions modulo four.

The distinction between finite occupancy and continuation is witnessed by an actual finite arithmetic character. The roots modulo eight are $1, 3, 5, 7$. The root 3 defines a member of $\mathcal{P}_q(\mathbf{Z}/8\mathbf{Z})$, but no member of $\mathcal{P}_q(\mathbf{Z}/16\mathbf{Z})$ reduces to it. Indeed the only scalar lifts of 3 are 3 and 11, and both square to 9 modulo sixteen. The formal proof also classifies all roots modulo sixteen and the reductions with children. Consequently a nonempty finite node need not lie on an infinite branch, even in this explicitly interpreted arithmetic packet.

The fixed-quadratic-packet interpretation completes this restricted arithmetic interpretation and its conditional completion. The packet is defined by one fixed quotient. It is not the collection of all quadratic characters of $G_{\mathbf{Q}}$. No local ramification criterion, Hodge–Tate weight, Weil–Deligne attach-

ment, or cross-coefficient local invariant was established by the fixed-quadratic-packet interpretation. The full manuscript still requires its remaining packet identifications and arithmetic comparisons. Universal positive occupancy remains conjectural.

Actual quadratic local conditions and completion. Continue with the fixed packet $\mathcal{P}_q$ above. For a characteristic-zero field K, let

$$j_K : G_K \longrightarrow G_{\mathbf{Q}}$$

be the actual field-induced homomorphism between absolute Galois groups. It includes a chosen embedding of algebraic closures. Let i_K be the image of the chosen square root i under that embedding. Lean proves compatibility with the Galois actions and hence

$$(q \circ j_K)(\sigma) = 1 \quad \Longleftrightarrow \quad \sigma(i_K) = i_K.$$

The fixed-field theorem for the algebraic closure then gives

$$q \circ j_K = 1 \quad \Longleftrightarrow \quad \exists x \in K,\ x^2 = -1.$$

If no such root exists, the local quotient is surjective onto C_2. These are statements about the actual local Galois map, not a supplied arithmetic interpretation of a scalar label.

For every commutative coefficient ring B and $\chi \in \mathcal{P}_q(B)$, define $\chi_K = \chi \circ j_K$. The local character is continuous and commutes with coefficient change. The precise local admissibility condition is

$$\mathrm{LocallyTrivial}_K(\chi) \quad \Longleftrightarrow \quad \chi_K = 1.$$

Descent through the fixed quadratic quotient proves

$$\mathrm{LocallyTrivial}_K(\chi) \quad \Longleftrightarrow \quad \big(\exists x \in K,\ x^2 = -1\big) \ \vee\ \chi = 1.$$

No reducedness, characteristic-zero or domain hypothesis is imposed on B. In particular, the formula applies to the finite coefficient quotients used in completion.

The p-adic square-root criterion is proved for every prime:

$$\big(\exists x \in \mathbf{Q}_p,\ x^2 = -1\big) \quad \Longleftrightarrow \quad p \equiv 1 \pmod 4.$$

A root satisfies $|x|_p^2 = 1$ and is therefore integral. Reduction modulo three rules out a root in $\mathbf{Q}_3$, and reduction modulo four rules out a root in $\mathbf{Q}_2$. At an odd prime, any residue root of $X^2 + 1$ has nonzero derivative and lifts by the pinned Mathlib Hensel theorem. The finite-field square criterion then gives the stated congruence. An explicit Hensel application at the residue root 2 modulo 5 supplies a root in $\mathbf{Z}_5$. Consequently

$$\boxed{\mathrm{LocallyTrivial}_{\mathbf{Q}_p}(\chi) \quad \Longleftrightarrow \quad p \equiv 1 \pmod 4 \ \vee\ \chi = 1.}$$

The arithmetic prime p is independent of the coefficient prime. For the two integral $\mathbf{Z}_2$-valued characters, the sign character is locally trivial exactly at primes congruent to one modulo four. Both distinct global characters are locally trivial at five, while the sign character is nontrivial on the full local Galois groups at three and two.

The inertia theorem uses actual residue actions. For a valuation subring A of an extension field L/K, let D_A be its decomposition subgroup in $\mathrm{Aut}_K(L)$ and put

$$I_A = \ker\bigl(D_A \longrightarrow \mathrm{Aut}(k_A)\bigr),$$

where k_A is the residue field. If 2 is a unit in A, every member of I_A fixes every $y \in L$ with $y^2 = -1$. Indeed $y^{-1} = -y$, so y belongs to A and is a unit there. An automorphism sends y to y or $-y$. Their residues are distinct because 2 and y are units, whereas inertia fixes residues. The second possibility is excluded.

For every prime p, Lean constructs a valuation subring $A \subseteq \overline{\mathbf{Q}_p}$ containing the integral image of $\mathbf{Z}_p$ such that the induced map $\mathbf{Z}_p \to A$ is local. For odd p, two is a unit. The preceding argument applies to every valuation subring containing that integral image, and hence every member of $\mathcal{P}_q(B)$ is trivial on its inertia subgroup after actual local restriction. The existence theorem supplies a locally dominating choice, so this assertion is not vacuous.

This inertia is the actual residue-action kernel inside D_A. No equality $D_A = G_{\mathbf{Q}_p}$ is established in this block. Nor is ramification at two proved: the nontrivial full local character there is a separate result. At three the $\mathbf{Z}_2$-valued sign character is nontrivial on the full local Galois group but trivial on every inertia subgroup covered by the theorem. Thus local triviality and the stated inertia triviality are distinguished by checked arithmetic examples.

The local conditions survive coefficient completion by a proved theorem. For an I-adically separated commutative ring R,

$$\mathrm{LocallyTrivial}_K(\chi) \quad\Longleftrightarrow\quad \mathrm{LocallyTrivial}_K(\chi \bmod I^m) \text{ for every } m \geq 1.$$

If -1 has a square root in K, every packet member satisfies the condition. Otherwise the condition says that the character is globally trivial, and separatedness detects that equality through all reductions. This proves the closure statement for the independently defined arithmetic property.

Suppose now that R is I-adically complete and that every positive quotient R/I^m is finite. Let $C_m \subseteq \mathcal{P}_q(R/I^m)$ be allowed subsets preserved by reduction, and let $(p_s)_{s\in S}$ be any indexed family of arithmetic primes. The index set S need not be finite. Lean proves the simultaneous arithmetic completion criterion

$$\begin{aligned}&\exists \chi \in \mathcal{P}_q(R),\quad \chi \text{ continuous},\quad \chi \bmod I^m \in C_m\ (m \geq 1),\quad \chi_{\mathbf{Q}_{p_s}} = 1\ (s \in S)\\ &\quad\Longleftrightarrow\quad \forall m \geq 1,\ \exists \psi \in C_m,\quad \forall s \in S,\ p_s \equiv 1 \pmod 4 \ \vee\ \psi = 1.\end{aligned}$$

The constructed character also satisfies every odd-prime valuation-inertia condition stated above. Finite-branch compactness selects all finite characters together. No coherent branch or surjective transition maps are assumed. The new local-closure implication is proved, not supplied. Every-level occupancy remains a substantive hypothesis. One nonsplit selected prime forces the completed global character to be trivial.

For the integral dyadic packet, a single detecting precision is also proved: reduction modulo four detects local triviality uniformly in the arithmetic prime. Modulo two does not suffice. The sign character is nontrivial locally at three, while its reduction modulo two is trivial. This is a coefficient-depth failure, distinct from splitting of the arithmetic field.

The quadratic local-condition construction finishes this actual local admissibility and completion step for the fixed quadratic packet. It does not identify all packets in the manuscript's common

arithmetic spine, attach Weil–Deligne objects, or prove coefficient and Hodge–Tate/period comparisons. Identifying the decomposition group with the full local Galois group remains a separate inertia obligation, and ramification at two is not claimed. Universal occupancy and natural finite seed rank remain conjectures.

C Retained residue-counting continuation

The following arguments develop residue counting and the associated norm-image conditions. Their formulas and hypotheses remain distinct from the amended conditional completion in Section D. The formal companion distinguishes the source contracts and their verification status.

Scope and notation. Let K be a number field of degree $d \geq 1$, with ring of integers $\mathcal{O} = \mathcal{O}_K$. Write $\mathrm{N}\, I = \#(\mathcal{O}/I)$ for the absolute norm of a nonzero integral ideal and $n(x) = \mathrm{N}_{K/\mathbb{Q}}(x) \in \mathbb{Z}$ for $x \in \mathcal{O}$. Thus $\mathrm{N}(x) = |n(x)|$. Let ι be the mixed embedding in its Euclidean coordinates, $L(J) = \iota(J)$, and D the actual unit-norm fundamental cone used in the preceding counting block. Its norm function scales by t^d under a positive real dilation t. The full-paper ledger remains **17 checked claims and 3 partial claims**.

Actual additive residues. Fix nonzero integral ideals $M \subseteq J$, and put $q = [J : M] = \#(J/M)$. The formal quotient of ideal lattices is identified with the additive congruence relation $x - a \in M$ on actual elements of J. In particular, the residue label is not an independent parameter attached to a count.

Proposition C.1. *There is a constant $B \geq 0$, depending on K, M, J, such that, for every $a \in J/M$ and every real $t \geq 1$,*

$$\left| \#\{x \in J : x \bmod M = a,\ \iota(x) \in tD\} - \frac{\mathrm{vol}(D)}{q\ \mathrm{covol}(L(J))} t^d \right| \leq B(t^{d-1} + 1).$$

The same constant works for all residues a.

The estimate is proved from the actual boundary charts and the fixed-lattice discrepancy theorem. The index factor comes from the covolume identity $\mathrm{covol}(L(M)) = q\ \mathrm{covol}(L(J))$. For a complex weight $w : J/M \to \mathbb{C}$, summing the estimates gives the leading term

$$\frac{\mathrm{vol}(D) t^d}{q\ \mathrm{covol}(L(J))} \sum_{a \in J/M} w(a),$$

with error at most $B \sum_a |w(a)|(t^{d-1} + 1)$. If the finite sum of weights is zero, only that error remains. Writing $Y = t^d$ gives an error $O(Y^{1-1/d} + 1)$ for the actual norm threshold. These assertions include $d = 1$.

Lean attachment: `ResidueLatticeIdeal`, `ResidueLatticeIdealPoints`, `ResidueLatticeIdealWeights`, and `ResidueLatticeNormCounts`.

The residue norm and its sign. For $m \in \mathbb{N}$, reducing the multiplication matrices of $\mathcal{O}$ modulo m and taking their determinants constructs a monoid homomorphism

$$\overline{n} : \mathcal{O}/m\mathcal{O} \longrightarrow \mathbb{Z}/m\mathbb{Z}, \qquad \overline{n}([x]) = n(x) \bmod m.$$

The construction is proved independent of the integral basis. For $m \neq 0$, its source is finite of cardinality m^d. Consequently $x - y \in m\mathcal{O}$ implies $n(x) \equiv n(y) \pmod{m}$. No surjectivity of $\overline{n}$ is asserted.

The relation to the geometric sign is also proved. If σ ranges over real embeddings and τ over one embedding from each complex pair, then

$$n(x) = \prod_{\sigma} \sigma(x) \prod_{\tau} |\tau(x)|^2.$$

For $x \neq 0$, the second factor is positive. Thus the sign of $n(x)$ is exactly the sign of the real-coordinate product. Cutting D into its positive and negative portions D_+, D_- adds no boundary outside the existing frontier, and both portions retain the proved boundary atlas. They are disjoint and their union is D.

For $\varepsilon \in \{1, -1\}$ define $\overline{n}_\varepsilon(a) = \varepsilon \overline{n}(a)$. On the corresponding sign portion,

$$\mathrm{N}(x) \bmod m = \overline{n}_\varepsilon([x]).$$

The sign cannot be dropped: $-1 \equiv 2 \pmod{3}$, whereas $\mathrm{N}(-1) = 1$ and $\mathrm{N}(2) = 2$ for principal ideals of $\mathbb{Z}$.

The exact ideal filter. Suppose $IJ = (x)$, $n(x)$ has the indicated sign, and $\gcd(\mathrm{N}\, J, m) = 1$. For $c \in \mathbb{Z}/m\mathbb{Z}$,

$$\mathrm{N}\, I \equiv c \pmod{m} \quad \Longleftrightarrow \quad \overline{n}_\varepsilon([x]) = c(\mathrm{N}\, J \bmod m).$$

This gives an actual finite residue target, which is allowed to be empty. The product identity before cancellation does not require coprimality.

Let $[J] = C^{-1}$ in $\mathrm{Cl}(\mathcal{O})$, and let $\mathcal{G}(J)$ denote the actual fundamental-cone generator set. For every predicate $P : \mathbb{R} \to \mathrm{Prop}$ and every real X, the norm-preserving correspondence gives

$$\begin{aligned} &\#\{x \in \mathcal{G}(J) : \mathrm{N}(x) \leq X\, \mathrm{N}\, J,\ P(\mathrm{N}(x))\} \\ &\qquad = \#\mu(K)\, \#\{I \neq 0 : [I] = C,\ \mathrm{N}\, I \leq X,\ P(\mathrm{N}\, I\ \mathrm{N}\, J)\}. \end{aligned}$$

The same norm-preserving equivalence carries arbitrary complex weights. In particular, write $G_{J,\chi}(Y)$ for the sum of $\chi(\mathrm{N}(x) \bmod m)$ over these generators with $\mathrm{N}(x) \leq Y$, and $T_{C,\chi}(X)$ for the corresponding sum over ideals in class C. For multiplicative χ preserving 0 and 1, the exact identity is

$$G_{J,\chi}(X\, \mathrm{N}\, J) = \#\mu(K)\, \chi(\mathrm{N}\, J \bmod m)\, T_{C,\chi}(X). \tag{1}$$

Both signs are included before applying this full roots-of-unity factor. A root of unity can interchange the sign portions, so that factor is not automatically valid for either portion separately.

Lean attachment: `NormResidueArithmetic`, `NormResidueIdealFilter`, `NormSignArithmetic`, `IdealNormFilterAttachment`, and `IdealNormWeightedAttachment`.

Ordinary class characters: an unconditional power saving. The preceding block proved, for every ordinary ideal class C,

$$A_C(X) = c_K X + O_{K,C}(X^{1-1/d}), \qquad c_K = \frac{2^{r_1}(2\pi)^{r_2} R_K}{\#\mu(K)\sqrt{|\Delta_K|}}.$$

Here A_C counts actual nonzero integral ideals, r_1, r_2 count the real and complex places, R_K is the regulator, and Δ_K is the discriminant. For a weight $w : \mathrm{Cl}(\mathcal{O}) \to \mathbb{C}$, define the actual finite ideal sum

$$S_w(X) = \sum_{\substack{I \neq 0 \\ \mathrm{N}\, I \le X}} w([I]).$$

Exact grouping by ideal class gives $S_w(X) = \sum_C w(C) A_C(X)$. If $\sum_C w(C) = 0$, the common leading coefficient cancels, so there is $B_w \ge 0$ such that

$$|S_w(X)| \le B_w X^{1-1/d} \qquad (X \ge 1).$$

For every nontrivial character $\chi : \mathrm{Cl}(\mathcal{O}) \to \mathbb{C}^\times$, the finite character sum is zero. Thus the displayed estimate applies without an additional arithmetic cancellation hypothesis. This conclusion concerns ordinary class characters. It does not identify them with the further norm-residue characters needed for Frobenius separation.

An actual Mellin endpoint. The counting functions are constant on the integer norm steps. Their measurability, and hence that of S_w, is proved from this fact. For zero-sum w, define

$$F_w(h) = (1+h)\int_1^\infty x^{-2-h} S_w(x)\, dx \qquad (h \ge 0).$$

Since $1 - 1/d < 1$, the logarithmic first moment

$$M_w = \int_1^\infty (1 + \log x)\, |x^{-2} S_w(x)|\, dx$$

is finite. Lean proves the explicit estimate $|F_w(h) - F_w(0)| \le h M_w$ for every $h \ge 0$, and therefore $F_w(h) = F_w(0) + O(h)$ as $h \downarrow 0$. The function here is the displayed integral of the actual ideal sum. Its identification with the intended arithmetic Euler product is a separate step.

For a finite family of nontrivial ordinary class characters, suppose functions p, Z satisfy $h\, p(h) \to P \neq 0$, $h\, Z(h) \to Q \neq 0$, and, for all sufficiently small $h > 0$,

$$Z(h) = \left(p(h) \prod_i F_{\chi_i}(h) \right)^e, \qquad e \in \mathbb{N}.$$

The checked real-axis criterion then gives $e = 1$ and $F_{\chi_i}(0) \neq 0$ for every i. The pole limits and product identity remain hypotheses of this application.

Lean attachment: `IdealClassCharacterBound`, `IdealCharacterMellinMeasurable`, `IdealCharacterMellin`, and `IdealCharacterMellinPole`.

An actual unit action on ideal residues. Fix $m > 0$ and a nonzero integral ideal J. Multiplication by $u \in (\mathcal{O}/m\mathcal{O})^\times$ gives a proved permutation of J/mJ. Its inverse follows from $(ab-1)x \in mJ$ when $ab - 1 \in m\mathcal{O}$ and $x \in J$. No surjectivity of $J/mJ \to \mathcal{O}/m\mathcal{O}$ or coprimality condition on J is used. The well-defined residue norm on this quotient satisfies

$$\overline{n}_J(u \cdot a) = \overline{n}(u)\,\overline{n}_J(a).$$

For a multiplicative $\chi : \mathbb{Z}/m\mathbb{Z} \to \mathbb{C}$ preserving 1, an exhibited unit with

$$u \in (\mathcal{O}/m\mathcal{O})^\times, \qquad \chi(\overline{n}(u)) \neq 1 \tag{2}$$

therefore forces

$$\sum_{a \in J/mJ} \chi\big(s\,\overline{n}_J(a)\big) = 0 \qquad (s \in \mathbb{Z}/m\mathbb{Z}).$$

Taking $s = 1, -1$ gives the two finite means needed by the sign-corrected norm weights. Their vanishing follows from the actual permutation.

The assembled ideal norm-character estimate.

Proposition C.2. *Let $C \in \mathrm{Cl}(\mathcal{O})$, let $J \neq 0$ be an integral ideal with $[J] = C^{-1}$, and let $m > 0$. Let $\chi : \mathbb{Z}/m\mathbb{Z} \to \mathbb{C}$ preserve multiplication, 0, and 1. Assume a residue-ring unit satisfying (2), and assume $\chi(\mathrm{N}\,J \bmod m) \neq 0$. Then*

$$T_{C,\chi}(X) = \sum_{\substack{I \neq 0,\ [I]=C \\ \mathrm{N}\,I \leq X}} \chi(\mathrm{N}\,I \bmod m) = O_{K,J,m,\chi}(X^{1-1/d}).$$

More precisely, some $B \geq 0$ bounds its absolute value by $BX^{1-1/d}$ for every real $X \geq 1$.

The assembly identifies the residue sum with the sum over actual generators, recombines the sign portions, and uses $Y = t^d$ to obtain $|G_{J,\chi}(Y)| \leq B_0(Y^{1-1/d} + 1)$. Equation (1) then permits division by the nonzero factor $\#\mu(K)\chi(\mathrm{N}\,J \bmod m)$. The final theorem supplies its finite means and counting estimates internally. For $d = 1$ the exponent is zero and the conclusion is a bounded sum, so the argument does not require $d > 1$.

What remains open. Nontriviality of χ on the larger residue ring does not supply (2): its restriction to the image of the norm on units may be trivial. Witness existence for the intended characters, their remaining norm-image cases, and suitable inverse-class representatives with $\chi(\mathrm{N}\,J) \neq 0$ are still arithmetic obligations. Measurability and the Mellin attachment for these norm-character sums remain to be checked. The compiled Mellin results above concern ordinary class characters. The resulting estimates must also be attached to the required Euler-product identities and prime-occurrence argument. The conditional pole criterion does not supply those identities. Claims 9.1, 9.2, and 11.1 remain partial, including the separate local compatibility obligations. The ledger stays **17 checked / 3 partial**.

Lean attachment: `IdealResidueUnitActionMean`, `NormResidueUnitCounting`, `NormResidueGeneratorAttachment`, `IdealNormCharacterBound`, and `NormResidueCharacterCounting`.

D Conditional occupied-system completion: final checked scope

The current endpoint is `PaperX.ConditionalOccupiedSystemCompletion.conditional_occupied_system_completion`. The twenty internal claims are checked at their amended and conditional source scopes. The assembly theorem uses explicit inputs. Unrestricted companion existence and carrier existence remain separate questions.

D.1 Two conclusions with separate inputs

The first conclusion starts with actual completed source members at every coefficient place, admissibility at every positive depth, reduction-natural horizontal transport, and an inverse-limit-closed target profile. The predecessor theorem `exists_transported_system_of_inverseLimitClosed` constructs a coefficient-indexed target family. Each target member satisfies every declared finite-depth admissibility condition, is related to the source by the exact all-depth relation `TransportedAlong`, and has the declared complete target profile.

The second conclusion starts with a supplied finite compatible family of pairwise nonisomorphic irreducible constituents, a supplied carrier and its explicit constituent-copy decomposition, and a supplied initial subrepresentation with its constituent-copy presentation. Claim 9.1's arithmetic occurrence result supplies a finite set of distinct good primes outside any prescribed finite forbidden set. Its actual Frobenius trace matrix has nonzero determinant. Claim 9.2 then uses this separator to recover the selected multiplicity vector and its summand inside the carrier.

Write ρ_i for the supplied constituents, c_i for the carrier multiplicities and m_i for the selected multiplicities, with $m_i \le c_i$. At a distinguished coefficient place λ, the input presentations have the form

$$V_{\mathrm{car},\lambda} \simeq \bigoplus_i \rho_{i,\lambda}^{\oplus c_i}, \qquad U_{0,\lambda} \simeq \bigoplus_i \rho_{i,\lambda}^{\oplus m_i}.$$

These are actual equivariant presentations. They are not inferred from an unexplained equality of numerical traces. The selected family $\bigoplus_i \rho_{i,\lambda}^{\oplus m_i}$ has a split equivariant inclusion in the carrier. The finite trace sample uniquely determines the multiplicity vector in the declared constituent packet.

The package `CarrierRecoveryAt` also supplies continuity of the selected copies, the common Frobenius characteristic-polynomial product, finite-copy Weil–Deligne and smooth-inertia data, local characteristic and Euler polynomial products, monodromy-kernel dimension and inertia-fixed trace formulas, and genuine semilinear period comparisons with weight-multiplicity sums. The constituent local packets and comparisons are semantic inputs. The conclusion proves their stated finite-copy closure.

D.2 Why the assembly closes

The final proof is a conjunction. The first branch applies the checked inverse-limit-closed horizontal completion theorem to the completed source family. The second branch constructs the finite arithmetic separator and applies the checked compatible-carrier closure theorem. Claim 9.1 constructs its separator independently of the carrier recovery. Claim 9.2 accepts the separator and the compatible-carrier packet. The final source composes these results. It does not use its conclusion to establish either premise.

The two branches do not identify the recovered carrier's supplied local Weil or period structures with restrictions of the transported target family. They remain logically distinct outputs. A simultaneous

all-place carrier statement is obtained by applying the pointwise result at every place when the corresponding carrier presentations are supplied.

D.3 Occupancy and inherited support

Zero multiplicities are permitted by the formal endpoint. A nonzero occupied conclusion additionally requires the manuscript's declared nonzero-multiplicity premise and nonzero constituents. The module-support/Fitting-ideal alternative is logically separate: its checked support formalization is inherited and is not proved again by this final module. No unrestricted companion existence, automorphy, carrier existence, or universal p-adic Hodge comparison follows from this conditional assembly.

D.4 Evidence tied to this exact source

The recorded verification includes warnings-as-errors builds, matching proof objects and an axiom audit allowing only propext, Classical.choice and Quot.sound. False proofs, placeholders and omitted closure or separator premises were rejected. An injected custom axiom compiled and was detected by the policy audit. The external dependency cache is a trusted input.

D.5 A later attempt with a different scope

The retained stronger residual-system attempt reports 19/20 and `FAIL_CLAIM111_NOT_PROMOTED`, with terminal error `NO_PINNED_LEAN`. No endpoint was promoted by that attempt. It is not a compiler rejection or retraction of the earlier checked conditional theorem. Its criterion begins with finite nonempty companion levels, while the checked conditional endpoint assumes actual completed members. These are different source obligations.

E Worked visual atlas

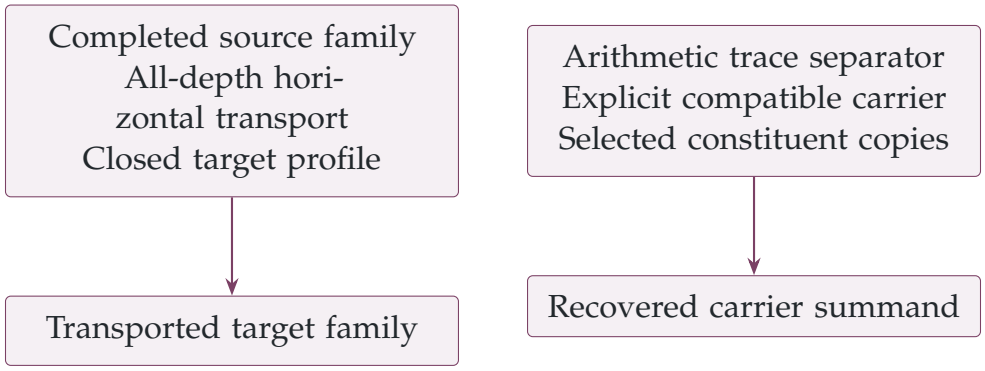
Plates X.1–X.30 form the worked visual atlas. Their labels, qualified by chapter, identify the same examples throughout the collection.

The atlas develops exact finite models, not substitutes for universal proofs. Its first part separates residual roots, coherent branches, and occupied carrier packets. Its second part studies residue norms, cancellation, Mellin estimates, and the boundary between a finite calculation and arithmetic occurrence. The final plate distinguishes the three source contracts and their mathematical boundaries.

The visual studies connect these examples to the mathematical arguments and to the formal companion.

Two routes to realization

The worked examples connect residual forests, finite occupancy and scalar/Mellin estimates. The conditional occupied-system theorem uses actual completed source members, inverse-limit profile closure, compatible carrier presentations and constituent local comparisons as explicit inputs.



The endpoint proves both outputs. It does not assert that their supplied local structures have been identified. Zero multiplicity is allowed. Nonzero occupancy needs its own input. All-place carrier recovery needs the pointwise presentations at each place.

Reading the examples

The mathematical text gives the arguments and the conditional assembly. The formal companion explains the terminal source, its dependencies and recorded verification. Each plate states the finite data used by its example and the conclusion that those data support.

The target-family and carrier-recovery constructions have different inputs. Their outputs need not be identified. A stronger residual-system contract, retained in the research record, has no promoted endpoint. Its existence requirements must not be substituted for the conditional theorem proved here.

Plate X.1 / READER'S GUIDE

One root is not the whole problem

Main text: §§1–3, 8–11. Conditional contract: Appendix D.

Chapter X separates three questions that can look deceptively similar. Does a residual object exist? Does it extend coherently through every precision? Does the resulting object occur in the required carrier? A successful answer to one question does not supply the next.

A residual root

$x^2 + 1 = 0$ over $\mathbb{F}_2$

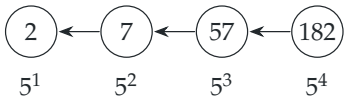


one point mod 2

no point mod 4

A coherent branch

$x^2 + 1 = 0$ over $\mathbb{Z}_5$



reductions agree

An occupied packet

1	0
0	0

rank $E = 1$

positivity needs a carrier

Three models to keep in view

The scalar equation $x^2 + 1 = 0$ has the root 1 modulo two, but no root modulo four. Over powers of five, the residues $2, 7, 57, 182$ form the beginning of a compatible branch. Neither fact constructs an automorphic carrier. In a supplied finite carrier, an idempotent such as $\mathrm{diag}(1, 0)$ detects an occupied subspace. It does not explain where that carrier came from.

Read the appendix in three passes. Pages Plates X.3–X.11 explain residual digits, finite counts, and obstruction spaces. Pages Plates X.12–X.20 explain directed transport, common data, and finite occupancy. Pages Plates X.21–X.29 give small exact models for the newer arithmetic continuation and its remaining limits.

> The foundational scalar/Mellin contract covers 17 of 20 records. Claims 9.1, 9.2 and 11.1 need their completed interfaces. The amended conditional assembly covers all twenty under the inputs stated in Appendix D. The finite character examples below do not prove prescribed Frobenius occurrence.

The figures illustrate exact models. Their finite Python checks and formal verification have separate records and scopes.

Plate X.2 / NOTATION BEFORE PICTURES

An atlas of the objects

Main text: Definitions 1.1, 5.1, 6.1, 7.1–7.2, and 8.1.

Keep the coefficient ring, its reduction, and the carrier separate. A residue class is a finite-precision address. A vector in a carrier is a different kind of object, even when both are selected by something called a packet.

Object	What it records	What it does not supply
$\mathcal{O}, \varpi, k$	Complete valuation ring, uniformizer, and residue field k of size q.	A represented deformation functor.
$\bar{x}$	One residual object in the declared moduli problem.	Other residual roots or their transports.
$X_m(\bar{x})$	Admissible objects modulo ϖ^m reducing to $\bar{x}$.	A lift of every individual member.
$(x_m)_m$	A branch satisfying $x_{m+1} \bmod \varpi^m = x_m$.	A global object without the stated reconstruction hypotheses.
w_i	A coordinate constrained to begin in $\varpi^{w_i}\mathcal{O}$.	A classification of all singular deformation rings.
$Z(T)$	The formal series $\sum_{m\geq 1} \#X_m T^m$.	The cardinality of the inverse limit.
$C^\bullet, d$	Cochains and a differential with $d^2 = 0$.	Its arithmetic identification with a Galois problem.
G	The graph of supplied admissible residual transports.	A proof that a proposed arithmetic edge exists.
$\mathscr{C}$	Common algebraic polynomials and declared local data.	A member realizing those data at every prime.
$C\boldsymbol{m} = \boldsymbol{b}$	A finite trace system for carrier multiplicities.	Integrality, positivity, or available capacity automatically.
$e_T V$	The joint eigenspace inside an existing finite carrier V.	Nonzero occupancy for every desired joint label.

Three indices that must not be conflated

The depth m counts retained digits. A coordinate index i selects a parameter or a packet. A coefficient place λ selects a different local field. Increasing m at one λ neither changes i nor constructs the member at another λ.

> **Read this as:** Specify the ambient object before applying a projector, a reduction, a rank, or a counting formula. The same symbol can refer to quite different spaces.

Plate X.3 / RESIDUAL PARTITION

Higher digits cannot change the root

Main text: Theorems 2.1–2.2 and Corollary 2.3.

Work in $\mathbb{Z}/27\mathbb{Z}$. Every element has a unique form $x = 3q + r$ with $r \in \{0,1,2\}$ and $q \in \{0,\dots,8\}$. The row below is the residual root. The column contains all the remaining information at this precision.

$x = 3q + r \pmod{27}$: columns are higher-digit addresses q

	0	1	2	3	4	5	6	7	8
$r = 0$	0	3	6	9	12	15	18	21	24
$r = 1$	1	4	7	10	13	16	19	22	25
$r = 2$	2	5	8	11	14	17	20	23	26

Adding 3 moves within one row. Adding 1 can change the row.

Read the operation, not just the arrow

An operation $x \mapsto x + 3h$ preserves r, because its correction vanishes modulo three. Repeated corrections of this form can visit every column in a row, but can never enter another row. By contrast, adding one changes the residual class. It is therefore not a positive-depth perturbation in this model.

For example, the addresses $1, 4, 7, 10, \dots, 25$ may be deeply connected by higher-digit changes. They remain disjoint from the addresses reducing to zero. In a genuine deformation problem, the same conclusion applies to operations whose reduction is the identity, not to every operation that happens to be called a deformation.

$$\operatorname{red}(x + 3h) = \operatorname{red}(x), \qquad \mathcal{M}(A) = \bigsqcup_{\bar{x}} \mathcal{M}_{\bar{x}}(A).$$

> **Read this as:** Vertical completeness can fill one entire residual fiber without reaching a single point in a different fiber.

All 27 addresses and their finite corrections are checked by integer reduction. Passing to an infinite perturbation stream additionally requires the convergence condition recorded in the formal no-crossing statement.

Plate X.4 / INVERSE-LIMIT RECONSTRUCTION

A branch fixes digits permanently

Main text: Theorem 3.2 and the compatible-digit example in §3.

The rational number $1/4$ is a five-adic integer. Its normalized residues are obtained by solving $4x_m \equiv 1 \pmod{5^m}$, not by taking decimal approximations. At the first six depths they are $4, 19, 94, 469, 2344, 11719$.

	5^0	5^1	5^2	5^3	5^4	5^5	
$x_1 = 4$	4	—	—	—	—	—	
$x_2 = 19$	4	3	—	—	—	—	$4x_m - 1 = 3 \cdot 5^m$
$x_3 = 94$	4	3	3	—	—	—	
$x_4 = 469$	4	3	3	3	—	—	
$x_5 = 2344$	4	3	3	3	3	—	digits already fixed never change
$x_6 = 11719$	4	3	3	3	3	3	

Compatibility is stronger than independent solvability

The residues satisfy both

$$x_{m+1} \equiv x_m \pmod{5^m}, \qquad 4x_m - 1 = 3 \cdot 5^m.$$

The latter identity shows that the residual error tends to zero in the five-adic norm. The stabilized digit sequence is

$$\frac{1}{4} = 4 + 3 \cdot 5 + 3 \cdot 5^2 + 3 \cdot 5^3 + \cdots \quad \text{in } \mathbb{Z}_5.$$

This series is not convergent in the ordinary real norm. The topology is part of the statement. To view the branch as a positive-depth coordinate above the residual root 4, use $t = x - 4 \in 5\mathbb{Z}_5$.

An inverse-limit point is the entire compatible family, not its last displayed approximation. The six rows are a finite window. The displayed exact formula supplies a branch at every depth in this particular example. In the represented deformation theorem, compatibility reconstructs maps on all ring elements, with multiplication, continuity, and locality retained.

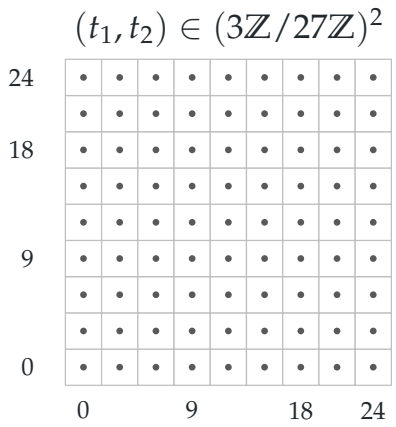
> **Read this as:** A later approximation may append a digit. It may not revise a digit already certified by reduction.

Plate X.5 / ACTUAL FINITE POINTS

A smooth chart has independent choices

Main text: Proposition 3.4 and Theorem 6.2.

Take $R = \mathbb{Z}_3[[t_1, t_2]]$ and coefficient-preserving local maps to $\mathbb{Z}/27\mathbb{Z}$. Locality forces both parameters into the maximal ideal. Each therefore has nine possible values, $0, 3, 6, \ldots, 24$.



$$9 \times 9 = 81 \text{ points}$$

$$(t_1, t_2) \mapsto (t_1, t_2) \bmod 9$$

Each parent has $3^2 = 9$ children.

The chart hypothesis supplies independent digits.

Why the exponent is $d(m-1)$

At depth m, the image of the maximal ideal contains 3^{m-1} elements. Two independent parameters give $3^{2(m-1)}$ points. The grid displays all 81 points at depth three, not a random sample. Reduction to depth two maps each coordinate to one of $0, 3, 6$, and every parent has nine children.

$$\#X_m = 3^{2(m-1)}, \qquad Z(T) = \sum_{m \geq 1} \#X_m T^m = \frac{T}{1 - 9T}.$$

Counting all 27^2 parameter assignments at depth three would ignore locality. Counting only tangent directions would describe the first layer but not all later layers. The power-series presentation explains why these choices stay independent through every precision.

This is a coordinate model with a proved presentation. A general deformation ring can impose equations among the parameters, changing the counts and causing branches to die. The next pages retain the same depth convention while changing the allowed coordinates or adding a singular equation.

> **Read this as:** The smooth count is not a universal rule for deformation spaces. Its independence comes from the actual formal power-series chart.

Plate X.6 / WEIGHTED DYADIC CHARTS

Weights delay visibility

Main text: Proposition 5.3, Theorem 6.3, and §12.2.

Over $\mathbb{Z}_2$, require three coordinates to lie in $2\mathbb{Z}_2$, $2^2\mathbb{Z}_2$, and $2^4\mathbb{Z}_2$. At depth m, coordinate i has $\max(m - w_i, 0)$ free binary digits. A zero in the table means only one allowed residue: zero.

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
$w_1 = 1$	0	1	2	3	4	5	6
$w_2 = 2$	0	0	1	2	3	4	5
$w_3 = 4$	0	0	0	0	1	2	3
count	1	2	8	32	256	2048	16384

Cell = number of free binary digits, not first separation depth.

The two different thresholds

A coordinate of weight w_i first becomes nontrivial when $m = w_i + 1$. By contrast, the first separation depth of two distinct fixed values a, b is $v_2(a - b) + 1$. Thus 1 and -1 separate at depth two, because their difference has valuation one. Neither threshold is simply the printed coordinate label.

For weights $(1, 2, 4)$,

$$N(m) = 2^{\max(m-1,0)+\max(m-2,0)+\max(m-4,0)}.$$

The first counts are $1, 2, 8, 32, 256, 2048$. Once all three coordinates are active, every additional precision digit multiplies the count by eight. Separating the initial levels from that geometric tail gives

$$Z(T) = \frac{T(1 - 6T - 8T^2 - 32T^3)}{1 - 8T}.$$

Negative coefficients in the numerator do not mean negative numbers of points. Expanding the whole rational expression gives positive coefficients.

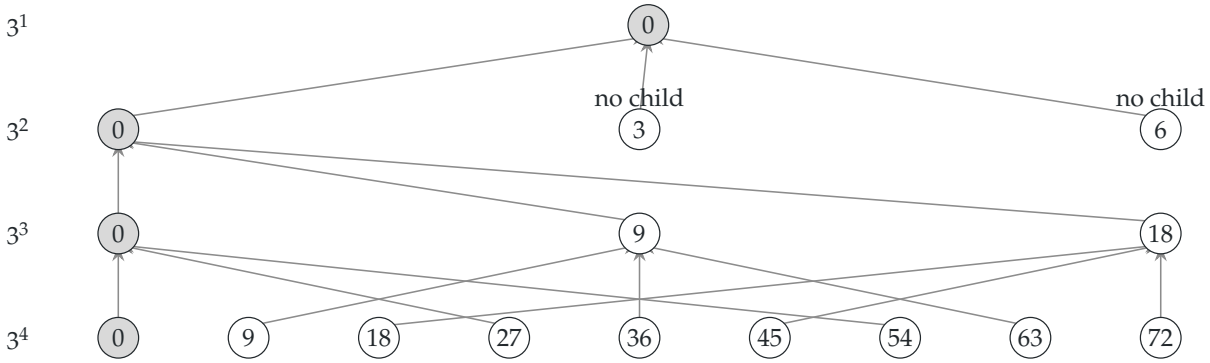
> **Read this as:** Weights change the finite-depth delay. After activation, this chart still has three independent digit streams.

Plate X.7 / SINGULAR POWER SEEDS

Many approximate roots, one limit

Main text: Theorem 6.4 and Corollary 6.6.

Now impose $x^2 = 0$ modulo 3^m. The condition is exactly $2v_3(x) \geq m$, so the roots are the multiples of $3^{\lceil m/2 \rceil}$ in that quotient. Their number is $3^{m-\lceil m/2 \rceil}$.



More finite roots, but only the zero stream persists through every precision.

Count vertices and count branches separately

There are three roots modulo nine: $0, 3, 6$. But the only roots modulo 27 are $0, 9, 18$, and all three reduce to zero modulo nine. The other two vertices at the previous level have no children. Counting them as permanent branches would be incorrect.

$$\#X_m = 1, 3, 3, 9, 9, 27, \ldots, \qquad Z(T) = \frac{T + 3T^2}{1 - 3T^2}.$$

The count-growth exponent is $\lim_m \log_3(\#X_m)/m = 1/2$. Nevertheless, a compatible branch reconstructs $x \in \mathbb{Z}_3$ with $x^2 = 0$. Since $\mathbb{Z}_3$ has no nonzero nilpotent, the only such branch is the zero branch.

More generally, $x^e = 0$ has count $q^{m-\lceil m/e \rceil}$ and count-growth exponent $1 - 1/e$. These exponents describe finite thickenings. They do not assert that the set of complete roots has a fractional geometric dimension or more than one point.

> **Read this as:** The seed zeta function counts finite-level solutions. It is not a generating function for distinct infinite branches.

Plate X.8 / A SCALAR COMPANION MODEL

One equation, several local outcomes

Main text: Theorem 8.2, Corollary 8.3, and Example 8.4.

Consider the fixed scalar equation $x^2 + 1 = 0$ in $\mathbb{Z}/p^m\mathbb{Z}$. The following are exact counts for five primes and three depths. The coefficient rings and the scalar nature of the unknown are fixed throughout this page.

	$p = 2$	$p = 3$	$p = 5$	$p = 7$	$p = 13$
$m = 1$	1	0	2	0	2
$m = 2$	0	0	2	0	2
$m = 3$	0	0	2	0	2

Counts of $x \in \mathbb{Z}/p^m\mathbb{Z}$ with $x^2 + 1 = 0$.

Where failure occurs matters

At $p = 3$ and $p = 7$, no residual root exists. At $p = 2$, the residual root exists but dies immediately: every odd square is one modulo four. At $p = 5$ and $p = 13$, the two residual roots are simple, and the usual one-digit equation gives one child above each root at every depth.

At five, one branch begins $2, 7, 57, 182$. The other is its negative modulo each power of five. Arbitrarily many digits in either branch cannot supply the absent scalar root over $\mathbb{Z}_3$. Nor can one root modulo two satisfy the hypothesis that every finite level is nonempty.

The chapter's system-completion theorem uses a supplied common algebraic spine, finite nonempty levels at every coefficient place, coherent reductions, and closed local conditions. The scalar example isolates why these are substantive requirements. It is not a disproof of a companion theorem whose coefficients may be enlarged or whose objects are higher-dimensional representations.

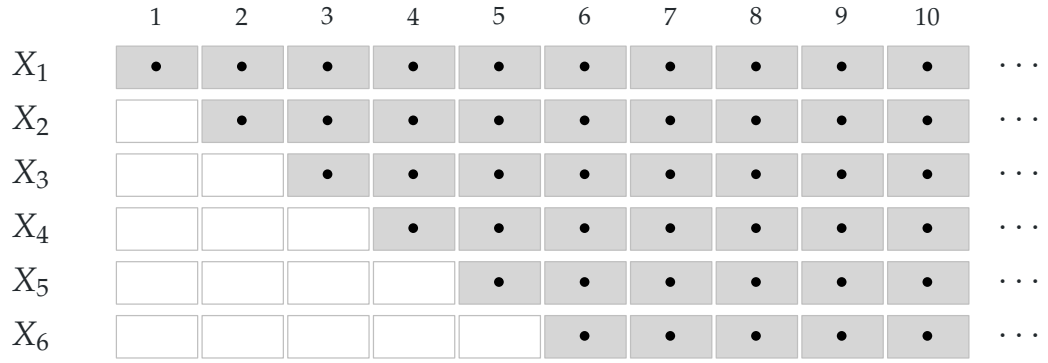
> **Read this as:** A residual root, roots at all depths, and roots at every required coefficient place are three distinct tests.

Plate X.9 / AN INFINITE-LEVEL COUNTERMODEL

Why finiteness enters compactness

Main text: Theorem 3.3 and its finite-branching hypotheses.

Let $X_m = \{n \in \mathbb{Z} : n \geq m\}$, with the inclusion $X_{m+1} \hookrightarrow X_m$ as transition map. Every level is nonempty. Every vertex has at most one child. Yet there is no compatible infinite branch.



$X_m = \{n \in \mathbb{N} : n \geq m\}$; every reduction is an inclusion.

The missing hypothesis is not surjectivity

Compatibility would force all coordinates of a branch to be one fixed integer n. Membership at every level would then require $n \geq m$ for every m, which is impossible. Each vertical column in the drawing eventually ends. A finite window always contains columns extending beyond it, but no single column survives all windows.

There is no contradiction with finite-branch compactness: these levels are infinite. Attaching a common artificial root would give that root infinitely many children. The finitely branching rooted-tree argument therefore does not apply.

In the chapter, finite nonempty levels and actual total reduction maps provide compactness even when some vertices die and some transition maps are not onto. One can retain only vertices with descendants at arbitrarily large depths. Finiteness makes that surviving part nonempty and permits compatible choices.

$$X_m \neq \varnothing \text{ for all } m \quad \not\Longrightarrow \quad \varprojlim_m X_m \neq \varnothing \quad \text{without the compactness hypotheses.}$$

> **Read this as:** Arbitrarily long finite paths are not automatically one infinite path. The finite-level hypothesis prevents escape to new roots.

Plate X.10 / A TWO-PACKET COCHAIN COMPLEX

See cycles, boundaries, and classes

Main text: Theorem 4.2 and the finite-complex example.

Over $\mathbb{F}_5$, use bases (e_0, e_1), (u_0, u_1, u_2, u_3), and (f_0, f_1) in degrees zero, one, and two. The only nonzero differential values are $d_0 e_0 = u_0$ and $d_1 u_1 = f_0$. Consequently $d_1 d_0 = 0$.

$d_0 : C^0 \to C^1$

	e_0	e_1
u_0	1	0
u_1	0	0
u_2	0	0
u_3	0	0

$d_1 : C^1 \to C^2$

	u_0	u_1	u_2	u_3
f_0	0	1	0	0
f_1	0	0	0	0

$$d_1 d_0 = 0$$

$$H^0 = \langle [e_1] \rangle$$

$$H^1 = \langle [u_2], [u_3] \rangle$$

$$H^2 = \langle [f_1] \rangle$$

Packet A is exact. Packet B has zero differentials and carries all cohomology.

Passing to the quotient

In degree one, the cycles are $\operatorname{span}(u_0, u_2, u_3)$, but u_0 is a boundary. Thus the cohomology is represented by u_2, u_3, not by every vector killed by d_1. In the other degrees the surviving classes are e_1 and f_1.

$$\dim(H^0, H^1, H^2) = (1, 2, 1).$$

Packet A contains e_0, u_0, u_1, and f_0. It is exact and contributes $(0, 0, 0)$. Packet B contains e_1, u_2, u_3, and f_1, with zero differentials, and contributes $(1, 2, 1)$. Adding the packet cohomologies reconstructs the total.

The projectors must be specified in every degree and commute with the differential. Their existence is not a consequence of the printed dimensions. Identifying this finite complex with the actual continuous cohomology of an arithmetic deformation problem would also require a separate comparison.

> **Read this as:** Resolve the differential first, then pass from cycles to classes. Counting packet dimensions alone does not compute cohomology.

Plate X.11 / THE CHAIN-MAP CONDITION

An arbitrary cut need not descend

Main text: Theorem 4.2 and the scope of the cohomology splitting.

Retain the previous complex. The valid A-packet projectors are $P^0 = \operatorname{diag}(1,0)$, $P^1 = \operatorname{diag}(1,1,0,0)$, and $P^2 = \operatorname{diag}(1,0)$. They satisfy $d_r P^r = P^{r+1} d_r$.

valid P^1_A invalid Q^1 $d_0 P^0_A - Q^1 d_0$

1	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

0	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

1	0
0	0
0	0
0	0

e_0 is retained at the source but its image u_0 is deleted at the target.

All displayed degreewise maps are idempotent. The differential test distinguishes them.

A boundary can cross an incompatible cut

Replace the degree-one projector by $Q^1 = \operatorname{diag}(0,1,0,0)$ while keeping P^0. Then

$$d_0 P^0 e_0 = u_0, \qquad Q^1 d_0 e_0 = 0, \qquad (d_0 P^0 - Q^1 d_0) e_0 = u_0 \neq 0.$$

The selected degree-zero component is sent outside the selected degree-one component. There is no subcomplex with these cuts, so the cohomology packet theorem cannot be applied to them. Both matrices remain perfectly valid idempotents. The failure is compatibility with the differential.

This example separates two kinds of obstruction. The nonzero matrix above obstructs this proposed decomposition. A nonzero class in H^2 may obstruct a lift only after the deformation problem and its obstruction map have been identified. One should not equate every failed projector commutator with an arithmetic obstruction class.

The exact checks verify both commuting diagrams for the valid family and reject the altered family. They also enumerate the finite kernels and images, so the dimension comparison is not inferred from the drawing.

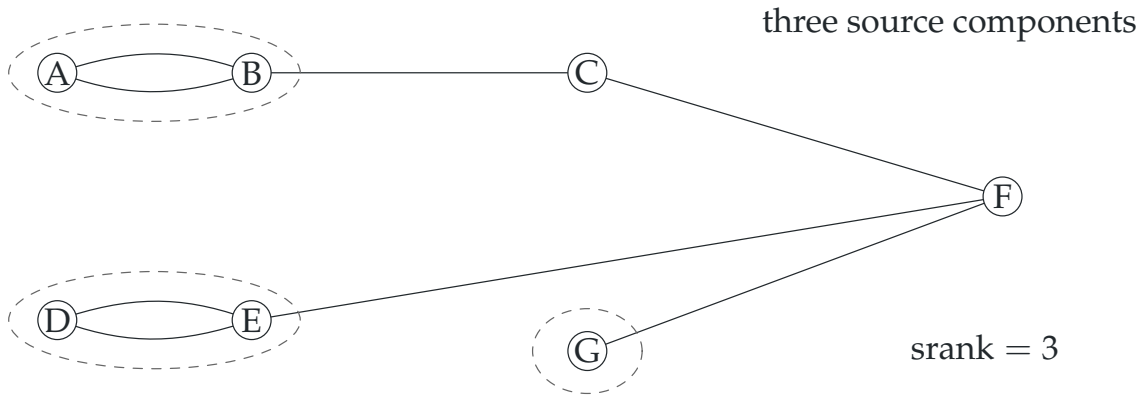
> **Read this as:** A projector on each vector space is not yet a projector on the complex. The differential must respect the chosen packets.

Plate X.12 / DIRECTED TRANSPORT

How many starting seeds are necessary?

Main text: Theorem 7.3 and the seven-root example.

The arrows below are the entire supplied transport graph. Vertices joined by paths in both directions form strongly connected components. The components $\{A, B\}$, $\{D, E\}$, and $\{G\}$ have no incoming edge from outside themselves.



A minimum set meets each dashed source component.

A lower bound and a construction

Any seed set reaching every vertex must meet all three source components. Otherwise there is no path into the omitted source. Conversely, one representative from each source reaches C and F as well. The directed seed rank is therefore three.

There are precisely four minimum seed sets:

$$\{A, D, G\}, \quad \{A, E, G\}, \quad \{B, D, G\}, \quad \{B, E, G\}.$$

The example checks this by exhaustive enumeration of all 2^7 subsets, independently of the strongly connected component calculation. Adding an admissible edge cannot increase the required seed count. In particular, adding $F \to G$ makes G reachable from the other sources, reducing the count to two.

The graph theorem is exact for the graph supplied. It does not establish that a proposed twist, induction, or arithmetic transfer preserves a determinant, local condition, or occupancy theorem. Those facts justify the arrows before graph optimization begins.

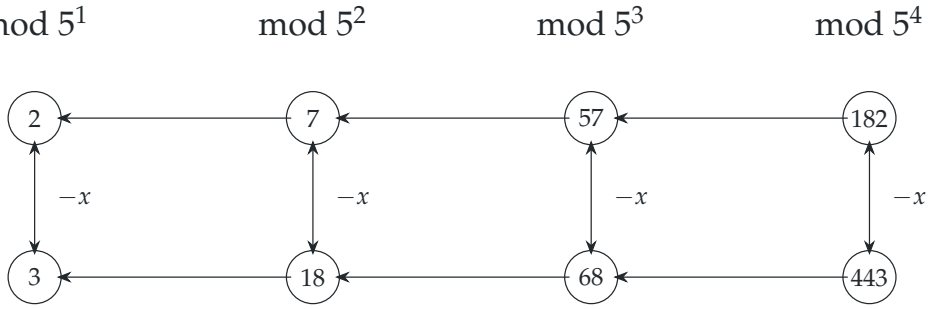
> **Read this as:** Count source components, not all components and not all vertices. But do not add an arithmetic arrow merely because it would improve the count.

Plate X.13 / ONE REVERSIBLE EXAMPLE

A transport must work at every depth

Main text: Definition 7.1 and the reduction-compatibility requirement.

For the two roots of $x^2 + 1 = 0$ modulo powers of five, negation is an actual reversible transport. It preserves the equation, is its own inverse, and commutes with reduction.



Negation commutes with reduction. Translation by 1 fails already at $2 \mapsto 3 \mapsto 4 \bmod 5$.

One successful input is not a natural transformation

At every depth m,

$$N_m(x) = -x \pmod{5^m}, \qquad r_{m+1,m} N_{m+1} = N_m r_{m+1,m}.$$

The two compatible branches beginning at 2 and 3 are exchanged. Because $2 \not\equiv 3 \pmod 5$, this transport changes the residual root. It is horizontal rather than a positive-depth correction.

Translation by one is a useful false control. Modulo five it takes the root 2 to the root 3, but takes the root 3 to 4, which is not a root. A single successful arrow between selected points does not define an operation on the whole deformation problem.

Even negation is admissible only for the equation and conditions stated here. An extra condition that fixes the residual root at 2 would not be preserved. Likewise, a representation-theoretic dual or twist needs its own compatibility checks for the chosen local data and carrier.

> **Read this as:** A transport is a family of maps respecting equations and reductions. A convenient pair of endpoints is not enough.

Both branches are computed by exact one-digit lifting. Negation is checked on all roots at the displayed depths. Its universal reduction identity follows directly from ring homomorphism laws.

Plate X.14 / THE ARITHMETIC SPINE

Common coefficients, different residues

Main text: Definition 8.1 and Theorem 8.2.

A common spine records algebraic data before reducing at a coefficient prime. For a small model, take the one polynomial

$$P(T) = T^2 - 5T + 6 = (T - 2)(T - 3) \in \mathbb{Z}[T].$$

Its reductions at five and seven look different, but they come from the same coefficients.

$$P(T) = T^2 - 5T + 6 = (T - 2)(T - 3)$$

coefficient images under the declared maps

	T^2	T	1
$\mathbb{Z}$	1	-5	6
$\mathbb{F}_5$	1	0	1
$\mathbb{F}_7$	1	2	6

Different residue rows come from the same algebraic polynomial.

Agreement means descent from supplied data

Modulo five the polynomial is $T^2 + 1$. Modulo seven it is $T^2 + 2T + 6$. The roots 2 and 3 are the reductions of the same integer roots in both places. The coefficient lists are not required to be equal as raw residues in different rings.

The chapter's common spine is larger: it includes characteristic polynomials at the permitted arithmetic places and the specified determinant, weight, and local labels. Requiring every finite member to match those reductions is a compatibility condition on existing objects. It is not a construction of the objects.

Finite agreement also leaves ambiguity. Adding 35 to an integer coefficient changes neither its residue modulo five nor its residue modulo seven. Thus two finite coefficient observations do not determine a unique global polynomial without a height bound or further exact information. Conversely, one arbitrarily chosen family of local objects need not descend to an algebraic spine.

> **Read this as:** The common spine is input shared across coefficient places. It is not created by comparing unrelated residue labels after the fact.

Plate X.15 / CHARACTERS BEFORE ARITHMETIC PRIMES

A finite trace matrix can separate types

Main text: Theorem 9.1; retained scalar/Mellin continuation: arithmetic-system separation.

Use the three irreducible characters of S_3: trivial, sign, and standard. On the identity, a transposition, and a three-cycle, their values form the matrix below. Rows are samples. Columns are constituent types.

C: actual S_3 character table

	trivial	sign	standard
1	1	1	2
(12)	1	−1	0
(123)	1	1	−1

$$\boldsymbol{m} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{b} = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}$$

$$\det C = 6$$

$$C^{-1}\boldsymbol{b} = (2, 0, 1)^T$$

Class representatives are finite group elements, not a proof that suitable Frobenius primes occur.

What full rank implies

Since $\det C = 6$, these samples distinguish every rational multiplicity vector. A target with two trivial constituents, no sign constituent, and one standard constituent has

$$\boldsymbol{m} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{b} = C\boldsymbol{m} = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}.$$

Solving $C\boldsymbol{m} = \boldsymbol{b}$ recovers the unique rational vector. The standard character is not introduced as an arbitrary list: the checks construct matrices for a transposition and a three-cycle, verify their group relations, and obtain their traces.

This finite example explains the linear algebra after separating samples are available. For the arithmetic theorem, the samples must be genuine unramified Frobenius classes with the required compatibility and allowed-prime conditions. Selecting abstract group elements does not construct such primes.

The scalar/Mellin construction supplies a separator assembly conditional on arithmetic coverage. The checked source Claim91ConstructedClosure discharges that interface for its rational-base contract. Neither result follows from the finite character table displayed here.

> **Read this as:** Invertibility answers reconstruction from these samples. Arithmetic occurrence answers whether the required samples can be taken at actual primes.

Plate X.16 / INTEGRAL, NONNEGATIVE, WITHIN CAPACITY

A solution is not yet a summand

Main text: Theorem 9.2 and its multiplicity-recovery example.

Keep the same separating matrix C and suppose the available carrier has capacities $\boldsymbol{c} = (2,1,1)$. A recovered vector describes a sum of selected constituents of that carrier only if each entry is an integer and $0 \le m_i \le c_i$.

trace data $\boldsymbol{b}$	$C^{-1}\boldsymbol{b}$	capacity $(2,1,1)$
$(4,2,1)$	$(2,0,1)$	allowed
$(1,0,0)$	$(\frac{1}{6},\frac{1}{6},\frac{1}{3})$	nonintegral
$(2,-2,-1)$	$(-1,1,1)$	negative
$(5,3,2)$	$(3,0,1)$	over capacity

Three independent rejection tests

The trace vector $(1,0,0)$ gives $(1/6,1/6,1/3)$, so it fails integrality. The vector $(2,-2,-1)$ gives $(-1,1,1)$, which is integral but not nonnegative. The vector $(5,3,2)$ gives $(3,0,1)$, which exceeds the available capacities. In each case the linear equation itself has a unique rational solution.

The accepted example $\boldsymbol{b} = (4,2,1)$ recovers $(2,0,1)$. When compatible constituent families are already supplied, the same admissible multiplicities define the corresponding direct sum at every coefficient place. This is the finite algebraic core of summand transport.

It does not supply a compatible family for an arbitrary object, construct all local attachments, or ensure that a carrier contains the desired representation. The carrier theorem requires explicit constituent-copy presentations and supplied local comparisons.

> **Read this as:** Invert the trace matrix, then check the representation-theoretic constraints. A unique vector in a field need not be a realizable multiplicity vector.

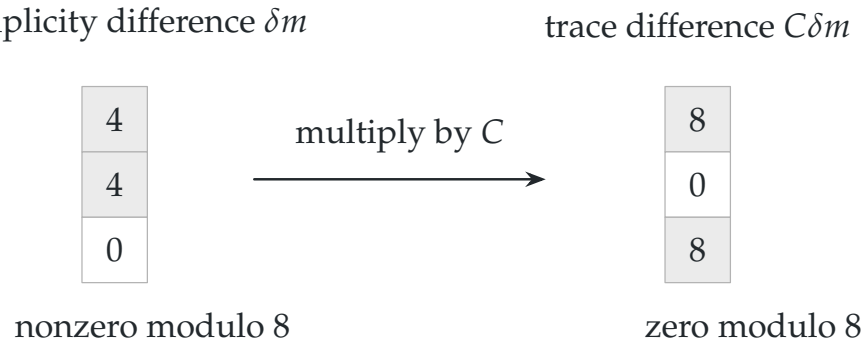
All accepted and rejected vectors are solved over exact rationals. The code rejects the wrong cases by their stated reason, not by a floating-point tolerance.

Plate X.17 / EXACT PRECISION LOSS

Separation can consume a digit

Main text: The finite trace-recovery discussion in §9.

The character matrix has determinant six, so its inverse contains a factor of two in the denominator. That arithmetic fact is visible as a loss of one binary digit in trace reconstruction.



Modulo 16, the traces separate them. $v_2(\det C) = v_2(6) = 1$.

A concrete ambiguity modulo eight

The multiplicity difference $\delta\boldsymbol{m} = (4, 4, 0)$ is nonzero modulo eight, but

$$C\,\delta\boldsymbol{m} = (8, 0, 8) \equiv 0 \pmod 8.$$

Thus traces known modulo eight do not distinguish those two residue vectors. The same trace difference is nonzero modulo sixteen, so one more trace digit resolves this ambiguity.

More generally, the kernel of C modulo 2^n has exactly two elements:

$$(0, 0, 0), \qquad (2^{n-1}, 2^{n-1}, 0).$$

To see this, the second equation gives $m_1 = m_2$. The third gives $m_3 = 2m_1$. The first then gives $6m_1 = 0$. Three is a unit modulo 2^n, leaving precisely the displayed possibilities.

Conversely, $C\delta\boldsymbol{m} \equiv 0 \pmod{2^{n+1}}$ implies $6\delta\boldsymbol{m} \equiv 0 \pmod{2^{n+1}}$ by the adjugate identity, hence $\delta\boldsymbol{m} \equiv 0 \pmod{2^n}$. Capacity bounds can remove ambiguity earlier in a particular carrier, but do not change this universal linear precision calculation.

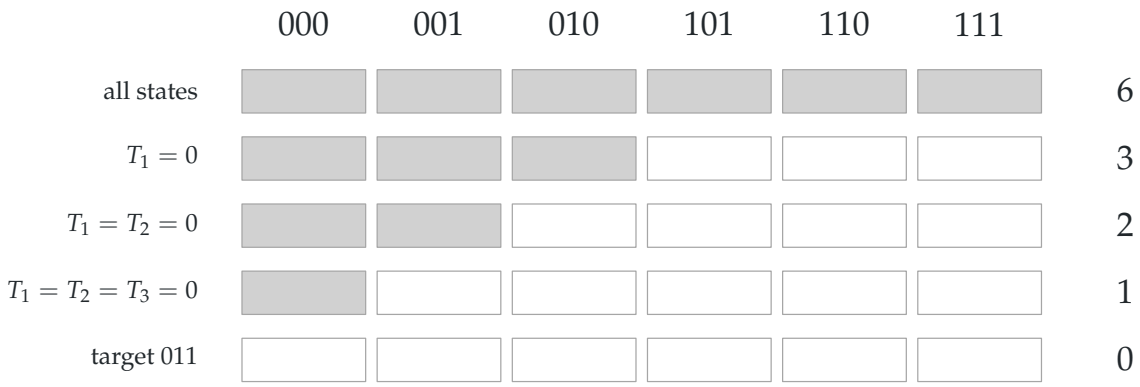
> **Read this as:** Full rank over characteristic zero is not the same as invertibility over every residue ring. Denominator valuations determine the needed extra precision.

Plate X.18 / EXACT JOINT PROJECTORS

Pairwise occupancy can still fail jointly

Main text: Theorem 10.1 and the finite occupancy models.

Let a six-dimensional rational carrier have the joint eigenvalue labels shown below for three commuting diagonal operators T_1, T_2, T_3. Each coordinate is zero or one. Selecting a zero uses $I - T_i$. Selecting a one uses T_i.



Each pair of requirements from 011 is occupied. Their triple intersection is empty.

The full conjunction is a different test

For target $(0,0,0)$, successive conditions reduce the ranks from 6 to 3, then 2, then 1. The joint projector is nonzero, so that label is occupied in this carrier.

For target $(0,1,1)$, every pair of conditions is occupied: $(0,1,0)$ witnesses the first two, $(0,0,1)$ the first and third, and $(1,1,1)$ the second and third. But the triple itself is absent. Multiplying all three selectors gives the zero matrix.

$$E_{\boldsymbol{a}} = \prod_{i=1}^{3} \begin{cases} I - T_i, & a_i = 0, \\ T_i, & a_i = 1. \end{cases} \qquad \operatorname{Tr}(E_{\boldsymbol{a}}) = \operatorname{rank}(E_{\boldsymbol{a}}).$$

The checks compute all eight joint projectors, their pairwise orthogonality, and their sum. The sum is the identity even though two labels are absent: those two projectors are simply zero.

> **Read this as:** A formula for a projector does not force its image to be nonzero. Occupancy must be tested on the complete set of required conditions.

Plate X.19 / TRACE AND SEMISIMPLICITY

Why the hypotheses are needed

Main text: Theorem 10.1 and the checked false controls following it.

The finite occupancy theorem uses a split semisimple commuting family in characteristic zero. Two matrices show why neither the coefficient condition nor semisimplicity should disappear from the statement.

I_5 over $\mathbb{F}_5$

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

rank 5, trace 0

$N^2 = 0$

0	1
0	0

only eigenvalue 0

$\dim \ker N = 1$

sole-eigenvalue product $= I_2$

Keep characteristic zero for trace detection, and split semisimplicity for the Lagrange formula.

Trace zero need not mean an empty image

On $\mathbb{F}_5^5$, the identity is an idempotent of rank five, but its field-valued trace is $5 = 0$ in $\mathbb{F}_5$. Rank and localization still detect its nonzero image. A characteristic-zero integer trace cannot be silently transferred to this setting.

An eigenvalue label need not select ordinary eigenvectors

For the nilpotent matrix $N = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$, the only eigenvalue is zero. A Lagrange selector that multiplies over all other eigenvalues is an empty product, hence I. Its image has dimension two, while $\ker N$ has dimension one. The selector does not project onto the ordinary eigenspace.

The whole two-dimensional space is the generalized zero-eigenvalue packet. Chapter XII's Artin-packet treatment is designed to retain such nilpotent structure. That is a different target from Chapter X's split semisimple joint-eigenspace formula.

> **Read this as:** A correct theorem in one coefficient category is not a theorem in all categories. Changing from ordinary eigenspaces to generalized packets also changes the object being detected.

Plate X.20 / CLOSEDNESS OF THE LOCAL CONDITIONS

A condition must survive the limit

Main text: Theorems 8.2 and 11.1; the inverse-limit closure hypotheses.

In $\mathbb{Z}_3$, every term of the sequence $x_k = 3^k$ is nonzero. Yet $x_k \to 0$ in the three-adic topology. The property "nonzero" is not closed, so a limit of objects satisfying it can lose that property.

	3^0	3^1	3^2	3^3	3^4	3^5	3^6
$a_k = 3^0$	1	0	0	0	0	0	0
$a_k = 3^1$	0	1	0	0	0	0	0
$a_k = 3^2$	0	0	1	0	0	0	0
$a_k = 3^3$	0	0	0	1	0	0	0
$a_k = 3^4$	0	0	0	0	1	0	0
$a_k = 3^5$	0	0	0	0	0	1	0

Every a_k is nonzero. Every fixed finite digit window eventually reads only zeros.

Polynomial conditions behave differently

For a continuous polynomial f, if a convergent sequence satisfies $f(x_k) = 0$ for every k, then continuity gives $f(x) = 0$ at its limit. More generally, a compatible residue branch satisfying $f(x_m) \equiv 0 \pmod{3^m}$ reconstructs an exact root once the complete ring and its reduction maps have been identified.

The sequence 3^k here is *not* presented as a compatible branch of nonzero residues modulo 3^k: those particular residues are zero. It is a convergent sequence of nonzero elements of the complete ring. The distinction prevents an incorrect finite-level interpretation of the picture.

In an arithmetic application, local conditions may be given by quotient equations, integral structures, or other data. The completion theorem requires a proof that the chosen condition is preserved by the limit under construction. The word "admissible" does not itself provide that proof.

> **Read this as:** Compatibility reconstructs an object. Closedness is what carries the declared equations or conditions to that object.

The finite diagram displays the retreat of the first nonzero digit. The convergence assertion follows from $|3^k|_3 = 3^{-k}$, not from extrapolating a finite numerical plot.

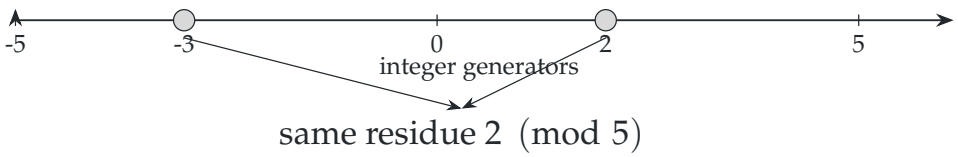
Plate X.21 / ELEMENTS VERSUS IDEALS

The sign of a norm is arithmetic data

Main text: §9, arithmetic residue-counting continuation; signed-norm comparison.

Even in the number field $\mathbb{Q}$, congruent generators need not have congruent absolute norms. The integers 2 and -3 are congruent modulo five. Their signed field norms are congruent, but their absolute values are not.

$$N_{\mathbb{Q}/\mathbb{Q}}(-3) = -3 \qquad N_{\mathbb{Q}/\mathbb{Q}}(2) = 2$$



same residue 2 $\pmod 5$

$$N((-3)) = 3 \qquad N((2)) = 2$$

The ideal norm is positive. Passing to absolute values must retain the real sign.

A counterexample over the integers

For $K = \mathbb{Q}$,

$$N_{K/\mathbb{Q}}(2) = 2, \qquad N_{K/\mathbb{Q}}(-3) = -3 \equiv 2 \pmod 5,$$

whereas

$$N((2)) = 2, \qquad N((-3)) = 3 \not\equiv 2 \pmod 5.$$

Here $N((a))$ is the positive index of the ideal (a), equal to $|a|$. The determinant norm of an element is signed. Taking absolute values is not a ring operation and does not commute with reduction in this example.

The retained scalar/Mellin continuation counts the two real norm signs separately before reassembling its generator-to-ideal sums. Its exact comparison also includes the roots-of-unity multiplicity and the norm of the chosen inverse-class representative. Omitting those factors would change the arithmetic sum, not merely its normalization.

The finite-field norm on the following pages is a determinant in a finite algebra. It has no ordering or absolute-value sign. Its role is to control residue characters before the separate real-lattice counting argument.

> **Read this as:** Do not infer an ideal-norm residue from a generator congruence without accounting for the sign in the element-to-ideal comparison.

Plate X.22 / THE QUOTIENT ALGEBRA $\mathbb{F}_{25}$

A norm is a determinant you can compute

Main text: §9; retained scalar/Mellin continuation: actual quotient determinant norms.

Use $A = \mathbb{F}_5[u]/(u^2 - 2)$. Since two is not a square modulo five, this quotient is a field with 25 elements. Every element is uniquely $a + bu$, with $a, b \in \mathbb{F}_5$.

$$N(a + bu) = a^2 - 2b^2 \pmod 5$$

	$b=0$	$b=1$	$b=2$	$b=3$	$b=4$
$a=0$	0	3	2	2	3
$a=1$	1	4	3	3	4
$a=2$	4	2	1	1	2
$a=3$	4	2	1	1	2
$a=4$	1	4	3	3	4

$$u^2 = 2 \text{ in } \mathbb{F}_{25}$$

$$M_{a+bu} = \begin{pmatrix} a & 2b \\ b & a \end{pmatrix}$$

$$\#N^{-1}(0) = 1$$

$$\#N^{-1}(r) = 6 \text{ for } r \neq 0$$

All 25 elements are shown. Every nonzero element is a unit.

Coordinates, multiplication, and norm agree

In the basis $(1, u)$, multiplication by $a + bu$ has matrix

$$M_{a,b} = \begin{pmatrix} a & 2b \\ b & a \end{pmatrix}, \qquad N(a + bu) = \det M_{a,b} = a^2 - 2b^2 \pmod 5.$$

The table shows every norm value. Only the zero element has norm zero. Each nonzero norm value has exactly six preimages. This makes the later character cancellation visible without estimating any sums.

The checks enumerate all 25 matrices, verify the determinant formula, and verify norm multiplicativity on every pair of field elements. They also change to a different invertible basis and check that conjugating the multiplication matrix leaves its determinant unchanged.

This model is the reduction of the quadratic integer algebra $\mathbb{Z}[\sqrt{2}]$ modulo five. Its finite arithmetic illustrates the determinant construction used in the continuation. It is not a computation of all ideal classes or a general ideal-counting theorem.

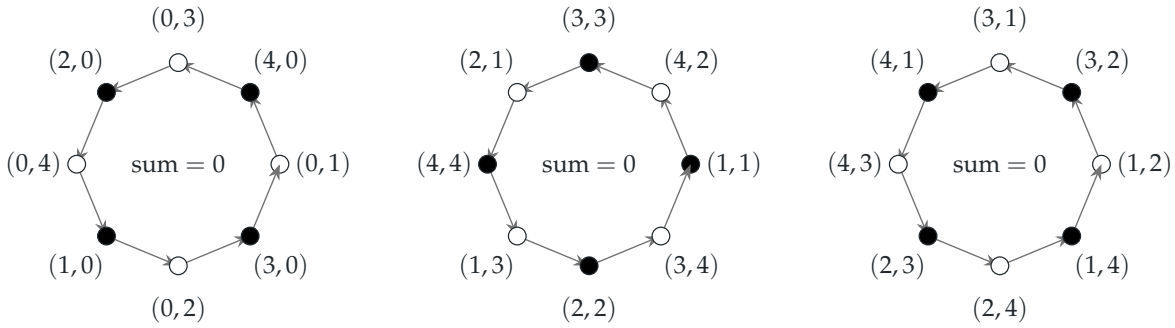
> **Read this as:** A quotient norm is built from the actual multiplication operator. Its basis independence is a determinant identity, not a choice of convenient coordinates.

Plate X.23 / A NONSCALAR UNIT WITNESS

Permutation forces a zero mean

Main text: §9; retained scalar/Mellin continuation: norm-residue lattice mean cancellation.

Let χ be the quadratic character modulo five, extended by $\chi(0) = 0$. In the field on the preceding page, choose $v = 2u$. Its norm is $N(v) = 2$, so $\chi(N(v)) = -1$.



Multiply by $v = 2u$. Its norm is 2, so $\chi(N(vz)) = -\chi(N(z))$.

Filled node: $+1$. Open node: -1. The omitted zero element has weight 0.

The whole sum is paired with its negative

Multiplication by v permutes the 25 elements. Norm multiplicativity gives

$$\sum_{z \in A} \chi(N(z)) = \sum_{z \in A} \chi(N(vz)) = -\sum_{z \in A} \chi(N(z)).$$

The sum is an integer, so it is zero. The origin contributes zero. The other 24 elements fall into three cycles of length eight under multiplication by v. The character signs alternate around each cycle.

The same reasoning works with a complex root of unity $\chi(N(v)) \neq 1$: subtracting the two equal sums gives $(1 - \chi(N(v)))S = 0$. The essential inputs are an actual permutation, a multiplicative norm, and a nontrivial character value. They are not consequences of drawing an alternating cycle.

In the arithmetic continuation the quotient is the actual additive module J/mJ associated with an ideal representative. This finite-field example isolates the cancellation mechanism while keeping that broader module and lattice comparison separate.

> **Read this as:** The witness is one unit that changes the character value. Once it exists, the permutation cancels the entire finite mean.

Plate X.24 / THE EXACT SCALAR TEST

A nontrivial character may still be exceptional

Main text: §9. Scalar/Mellin construction: scalar-power and nonscalar-witness boundaries.

A scalar $a \in \mathbb{F}_5^\times$ acts on a degree-d algebra with determinant a^d. Therefore the scalar cancellation route tests $\chi(a^d)$, not merely $\chi(a)$.

	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$
trivial	no	no	no	no	no	no
quartic	yes	yes	yes	no	yes	yes
quadratic	yes	no	yes	no	yes	no
conjugate quartic	yes	yes	yes	no	yes	yes

For $\chi_k(2) = i^k$, a witness exists exactly when $kd \not\equiv 0 \pmod 4$.

A failed sufficient test is not an impossibility theorem

Let $\chi_4(2) = i$ generate the character group of $\mathbb{F}_5^\times$, and write $\chi_k = \chi_4^k$. Since 2 generates the unit group,

$$\exists a : \chi_k(a^d) \neq 1 \quad \Longleftrightarrow \quad kd \not\equiv 0 \pmod 4.$$

For the quadratic character $k = 2$ and degree $d = 2$, every scalar test is trivial. Yet page Plate X.23 supplies the nonscalar witness $2u$, proving cancellation in that specific algebra. Scalar exceptional does not mean exceptional for every possible quotient unit.

Conversely, the quartic character $k = 1$ works in degree two even though $\gcd(2,4) \neq 1$. Thus the simpler degree–totient coprimality criterion is sufficient, not necessary. With degree three, every nontrivial character passes because exponentiation by three permutes the units.

The retained scalar/Mellin continuation states its public hypothesis in the weaker exact scalar-power form. It does not claim to construct a nonscalar witness for every scalar-exceptional arithmetic character.

> **Read this as:** Test the character after the degree-power map. Nontriviality before that map can disappear after it.

Plate X.25 / A CRT MODEL FOR REPRESENTATIVE CHOICE

Avoid finitely many primes constructively

Main text: §9; retained scalar/Mellin continuation: finite-prime avoidance and automatic representatives.

A representative needed by an arithmetic argument often has to avoid a prescribed finite collection of primes. A small integer version asks for

$$x \equiv 1 \pmod 2, \qquad x \equiv 2 \pmod 3, \qquad x \equiv 4 \pmod 5.$$

There is exactly one solution modulo 30: $x = 29$.

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29

Choose one residue in each quotient

$$a \equiv 1 \pmod 2$$

$$a \equiv 2 \pmod 3$$

$$a \equiv 4 \pmod 5$$

$$a = 29, \quad \gcd(29, 30) = 1$$

Shaded values avoid 2, 3, and 5 simultaneously.

Avoidance and ideal class are different constraints

The selected residues are nonzero modulo each forbidden prime, so $\gcd(29, 30) = 1$. The eight units modulo 30 are displayed separately. They all avoid the primes, but only one also satisfies the three specific residue prescriptions.

The continuation proves an ideal-theoretic analogue with more structure: in every prescribed inverse ideal class it constructs a nonzero integral representative whose norm is coprime to the nonzero modulus. That removes a representative-choice assumption from the public norm-character bound. It does not remove the remaining character, lattice, or Frobenius-occurrence obligations.

The integer CRT calculation is an illustration, not a proof of the ideal-class representative theorem. In particular, enumerating integers cannot impose a nontrivial ideal class. The full construction uses prime-ideal avoidance and its source comparison to the class group.

> **Read this as:** A finite avoidance argument supplies an admissible representative. It does not by itself make the arithmetic object being counted, or the desired Frobenius class, occur.

All 30 residue classes are tested exactly. The unique CRT result and the unit count are independent finite outputs of the check script.

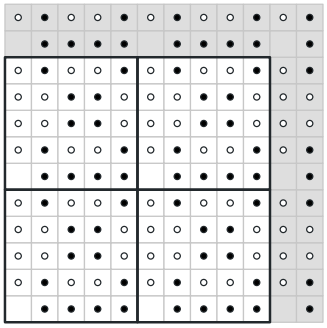
Plate X.26 / A FULL LATTICE-BOX CALCULATION

Cancellation leaves a boundary problem

Main text: §9; retained scalar/Mellin continuation: additive residue counts and norm-character bounds.

Extend the function $f(a,b) = \chi(a^2 - 2b^2 \bmod 5)$ periodically to $\mathbb{Z}^2$. Every complete 5×5 residue tile has sum zero by the finite-field calculation. Only incomplete tiles can contribute to a box sum.

$$[0,12)^2 \cap \mathbb{Z}^2$$



four full 5×5 cells

100 bulk points: sum 0

44 boundary points

actual total $S_{12} = 1$

$|S_L| \le L^2 - (L - (L \bmod 5))^2$ for this square-lattice model.

Separate bulk cancellation from boundary size

For $0 \le a, b < 12$, four complete tiles fill a 10×10 square and contribute zero. The remaining 44 points lie in the two boundary strips. Their exact sum is one. Since $|f| \le 1$, a bound of 44 follows without needing that final cancellation.

For a general positive integer L, write $L = 5q + r$, $0 \le r < 5$. The complete tiled square has side $L - r$, so

$$|S_L| \le L^2 - (L-r)^2 = 2Lr - r^2 \le 8L.$$

The area is $X = L^2$, giving a bound proportional to the square root of the area in this square model. The proof is the exact tile decomposition. Checking many L values is only a regression check of the implementation.

Number-field ideal sums involve different regions, lattice indices, signs, and generator multiplicities. Their $X^{1-1/d}$ exponent cannot be justified simply by relabeling this square picture as an ideal-norm ball.

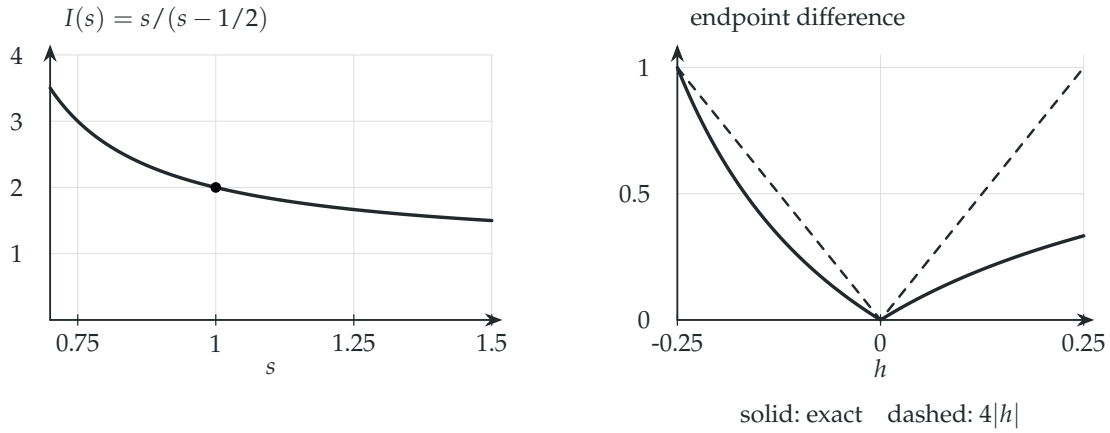
> **Read this as:** A zero finite mean removes the bulk. The remaining estimate depends on the actual geometry of the region and its boundary.

Plate X.27 / A MELLIN ENDPOINT MODEL

Analytic consequences of a power-saving bound

Main text: §9; retained scalar/Mellin continuation: summatory measurability and first-order Mellin control.

Use the explicit envelope $A(x) = x^{1/2}$ for $x \geq 1$. It is not an ideal-counting function. It is a model showing why a power saving can make a Mellin expression regular at $s = 1$.



Compute the endpoint rather than just naming it

For real $s > 1/2$,

$$I(s) = s \int_1^\infty A(x) x^{-s-1}\,\mathrm{d}x = \frac{s}{s-1/2}.$$

At $s = 1 + h$ this gives

$$I(1+h) - I(1) = -\frac{h}{h+1/2}, \qquad |I(1+h) - I(1)| \leq 4|h| \quad (|h| \leq 1/4).$$

The derivative at one is -2. The logarithmic moment is finite as well: $\int_1^\infty x^{-3/2} \log x\,\mathrm{d}x = 4$. These are explicit calculations. The drawn curve does not establish differentiability or convergence.

The continuation performs the corresponding analytic steps for its actual summatory functions, with their explicit measurability and growth hypotheses. The arithmetic Euler-product identity, nonzero pole residues, and the required prescribed-prime divergence remain separate. Regularity of one Mellin transform does not construct all of those inputs.

> **Read this as:** An analytic estimate controls a function already tied to arithmetic data. It is not a substitute for proving that tie.

Plate X.28 / THE REMAINING FROBENIUS STEP

Finite splitting data are not a density theorem

Main text: Theorem 9.1. Latest continuation: GeneralArithmeticCoverage and ArithmeticSystemSeparation.

For the quadratic polynomial $X^2 + 1$, reduction modulo an odd prime either gives two roots or no roots. This finite scan displays both behaviors. It is a useful model for the distinction between splitting and nonsplitting classes.

odd primes p

3	5	7	11	13	17	19	23
0	2	0	0	2	2	0	0

roots of $x^2 + 1$ in $\mathbb{F}_p$

two factorization types
already appear in this finite list

repeat one class

1	1	2
1	1	2
1	1	2

three rows, rank one

Finite character separation needs enough different evaluations, not just more evaluations.

Existence must survive arbitrary exclusions

The continuation's missing general coverage proposition asks for an actual unramified rational prime in a prescribed conjugacy class, outside any prescribed finite forbidden set, together with the corresponding absolute Frobenius witness. A list of small primes does not establish that quantifier.

The finite character argument on page Plate X.15 also needs samples that jointly separate the constituents. Repeating the identity-class row $(1, 1, 2)$ of the S_3 table gives rank one no matter how often it is repeated. Many samples are not necessarily informative samples.

The later checked rational-base separator composes its constructed arithmetic coverage theorem with finite linear algebra. The earlier scalar/Mellin route retained a coverage premise. The displayed scan certifies neither route: it is a finite example, not an occurrence theorem.

> **Read this as:** A separating group element, a witnessed Frobenius at one prime, and a new prime after every finite exclusion are distinct levels of evidence.

The scan exhaustively checks scalar roots for the shown primes. No density, infinitude, or general prescribed-conjugacy statement is inferred from it.

Plate X.29 / SCALAR ABSENCE VERSUS REPRESENTATION EXISTENCE

Changing coefficients changes the problem

Main text: Example 8.4 and the declared coefficient categories in §§8–9.

Over $\mathbb{F}_3$, no scalar solves $x^2 + 1 = 0$. But the matrix

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \in M_2(\mathbb{F}_3) \quad \text{satisfies} \quad A^2 + I = 0.$$

And over $\mathbb{F}_9 = \mathbb{F}_3[u]/(u^2 + 1)$, the scalars u and $-u$ solve the same polynomial.

Scalar over $\mathbb{F}_3$	Matrix over $\mathbb{F}_3$	Scalar over $\mathbb{F}_9$
$x = 0, 1, 2$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	$u^2 = -1$
$x^2 + 1 = 1, 2, 2$		roots $u, -u$
no solution	$A^2 + I = 0$	coefficient extension

Fix the coefficient field, dimension, and point category before declaring an occupancy failure.

Three outputs with different types

The failed scalar problem over $\mathbb{F}_3$, the successful matrix problem over $\mathbb{F}_3$, and the successful scalar problem over $\mathbb{F}_9$ are not contradictory answers. They have different allowed objects. The matrix is multiplication by u in the $\mathbb{F}_3$-basis $(1, u)$ of the extension field.

This also has a characteristic-zero analogue. The matrix with the same entries gives a rational two-dimensional representation of a cyclic group of order four. Its scalar eigenvalues require a field containing a square root of -1. No one-dimensional rational eigencharacter is created by writing the two-dimensional matrix.

Accordingly, the scalar companion failure on page Plate X.8 is a counterexample only to the fixed scalar problem. It is not a counterexample to a representation problem that permits coefficient extension or a different dimension. Those permissions must be declared before comparing fibers across primes.

> **Read this as:** An empty fiber is meaningful only after its coefficient ring and object type are fixed. Do not repair an empty fiber by silently changing either.

Plate X.30 / FINAL READER'S CHECKLIST

Read the evidence at its actual scope

Main text: Appendix D. Formal guide: statement register, dependency sources, and replay boundary.

The three contracts have different hypotheses. Their coverage counts do not make them interchangeable.

Record	What it establishes
Base 17/20	Scalar/Mellin contract. Claims 9.1, 9.2 and 11.1 remain partial in that contract.
Checked 20/20	The amended conditional occupied-system assembly. Completed source members, closed target profiles, compatible carrier presentations, and local comparisons remain explicit inputs.
Stronger 19/20	A stronger residual-system contract with an unproved terminal endpoint. It does not recheck or replace the conditional assembly.

Two conclusions, not an unstated identification

The terminal theorem constructs a transported target family and independently recovers a summand of the supplied carrier. To identify them requires an explicit comparison. The carrier result is pointwise at an arbitrary coefficient place. Recovering carriers at every place needs the relevant presentations there. Zero multiplicity is allowed. Nonzero occupancy needs a separate nonzero premise.

What the examples verify

The appendix supplies reproducible finite examples and exact Python checks. A plotted pixel does not decide a congruence, rank, determinant, root or character sum. The formal companion states the scope of the Lean verification. Sources and supporting records accompany the volume.

> **Read this as:** A completed conditional theorem is a theorem with its hypotheses, not a claim that every arithmetic application supplies them.

CHAPTER XI

Tame and Wild Packet Geometry

For local Langlands

The preceding chapter separates the arithmetic objects supplied as inputs from those obtained conditionally. Local packet geometry sharpens the realization questions.

Tame and wild packet geometry organizes the local arithmetic constructions. The research companions retain both restricted progress and the unresolved obligations of Conjecture 66.

IN THIS CHAPTER

Mathematical text 1260

Visual studies 1367

Formal companion 2384

Further arithmetic studies 1482

Sources and references 2693

Scope

Conjecture 66 remains open. Its unresolved requirements concern general arithmetic realization, H5 for the same candidate, corrected full-character occurrence and actual native data under the stronger contract.

The final chapter gives a conditional route through patching support and Hecke compatibility. The open realization obligations remain separate.

Abstract. Let K be a nonarchimedean local field. We resolve several finite layers that occur inside the local Langlands correspondence for general linear groups. Frobenius semisimple Weil Deligne parameters first split into irreducible Weil atoms and finite monodromy chains. In the tame case, character projectors show that every irreducible atom is one transitive Frobenius orbit, and Mackey theory gives an exact greatest common divisor and least common multiple rule for tensor packets. Regular finite field characters produce primitive cuspidal projectors and finite Gauss phases. For wild data, projective intertwiners recover an alternating commutator form. Its radical free quotient determines a finite Heisenberg representation, and maximal isotropic averaging gives an exact selector. We prove a restricted one break bridge: an equivariant symplectic isometry between a declared Galois envelope and a type side quotient induces the Heisenberg comparison uniquely up to scalar, while one cyclic phase condition removes the remaining ambiguity. Odd residue characteristic is governed by finite Weil phases. Residue characteristic two requires a quadratic refinement and its Arf invariant. A matrix holonomy theorem then gives the exact gluing criterion on a finite transport graph. The general arithmetic packet realization is stated as Conjecture 66. Detailed ordinary proofs give restricted arithmetic realizations in all odd prime-power ranks and in all dyadic ranks for the stated increasing-break families. The complete comparison of pair local factors remains a conditional input for the same arithmetic candidate.

A route through Chapter XI

This chapter presents the finite packet theory and retains the detailed restricted arithmetic constructions. Two separate research companions explain the later progress toward Conjecture 66. They do not replace this chapter's proved statements or turn its conditional comparison into an existence theorem.

The mathematical text and its companions

Reader	Route and purpose
Mathematical text	Finite tame and wild theory, restricted arithmetic constructions, and the exact realization/comparison boundary.
Core visual guide	Geometric route through the mathematical text. The original arithmetic examples retain their stated scope.
Formal companion	Formal statements, literal excerpts and evidence lineage. The guide distinguishes the selected source census, compiler replay and arithmetic hypotheses.
Progress paper	Toward Arithmetic Packet Realization. Connected proofs, failed routes, conductor conventions and conditions on native markings.
Research visual atlas	Arithmetic Packet Realization. Thirty-two worked plates accompany the progress paper.

The endpoint is a quantified construction problem

Conjecture 66 asks for an actual arithmetic witness for each admitted source with its supplied H1–H4 datum. The comparison proposition then uses H5 for that same realized candidate. The formal companion distinguishes this conjecture from the differently numbered statements in the source registers.

The research contract in Section 1 and Plate XI.1 make the quantifiers explicit. Supplying finite operators does not construct an arithmetic representation. Likewise, proving a factor identity for a different candidate does not identify the packet's realizer.

How to use the links

The reading links connect the mathematical text, visual studies, research companions and formal companion. Section and result numbers are local to the named treatment. Plate references include the chapter number, so Plate XI.32 identifies the corresponding visual plate.

Scope and research routes

Question	Route beyond the core statement
Parameter, full character and recovery	Progress Sections 7–10 and Plates XI.6–XI.10. The full character, central character and noncompact completion remain fixed.
Monomial, Clifford and occurrence routes	Progress Part II. The original character route is excluded under its hypotheses. A different corrected character passes a scalar test, but occurrence in the same candidate is still open.
Conductor and capacity	Progress Sections 22–25 and Plates XI.11–XI.19. Swan, ordinary conductor and epsilon exponent for an additive character are named separately. Equal numerical invariants need not give equal quotient capacity.
Native markings and allowed repairs	Progress Sections 26–29 and Plates XI.20–XI.29. A freely transported marking is different from a prescribed native marking. Finite corrections require their stated inputs and permission to change the relevant map.
Remaining arithmetic certificate	Progress Section 30 and Plates XI.30–XI.32. General realization, H5, corrected-character occurrence and the stronger native certificate remain separate tasks.

The construction and its limits

The breakwise phase product uses its moving symplectic spaces. The type-side pairing fixes a prime-field character, and the tame comparison retains the boundaries for the rank-one conductor and the Gauss sum of the trivial character. Interaction carriers are distinguished from minimal character orbits.

The detailed construction appendices give ordinary proofs under their stated type-theory hypotheses. The selected Henniart–Lemaire induction inputs [20] retain their characteristic, rank and normalization conditions. They do not establish H5 or corrected-character occurrence. The formal companion identifies the actual compiler and axiom coverage.

1 Introduction

The local Langlands correspondence for $\mathrm{GL}_n(K)$ is an established theorem. It matches irreducible smooth representations of $\mathrm{GL}_n(K)$ with Frobenius semisimple Weil Deligne representations of dimension n. Its proof and normalization draw on global methods, harmonic analysis, types, local factors, and the geometry of Shimura varieties [19, 17, 8]. This chapter asks a narrower structural question. Which parts of the local correspondence admit finite packet coordinates whose projectors, transfers, phases, and gluing laws can be proved directly?

The answer has a sharp division. The tame layers are explicit. A tame irreducible Weil representation is induced from a character of an unramified extension. Its inertia restriction is one Frobenius orbit. Regular characters of elliptic finite field tori give cuspidal blocks through the Deligne and Lusztig construction [13]. Their tensor products are controlled by a greatest common divisor and least common multiple decomposition. The corresponding cycle orientation and finite Gauss phases can be computed exactly. These features are closely related to the tame lifting and essentially tame correspondence of Bushnell and Henniart [4, 5, 6, 7, 9].

Wild data are different. The restriction to wild inertia need not split into cyclic characters. On the type side, simple character theory produces finite symplectic quotients and Heisenberg representations [11]. Canonical beta extensions fix a phase ambiguity in the extension to the normalizer [2, 14]. On the Galois side, higher ramification gives a canonical filtration and strong numerical constraints [10]. The missing construction is not another finite matrix calculation. It is a choice free passage from an arbitrary wild inertia atom to a finite symplectic envelope that agrees with the type side quotient.

We therefore separate three levels of assertion. First, this chapter proves finite algebraic theorems about tame orbits, Heisenberg groups, twisted group algebras, finite Weil phases, and transport graphs. Second, it proves a restricted one break bridge once a normal wild datum and an equivariant symplectic identification are declared. Third, it records the exact additional input needed to pass from that restricted bridge to an independent construction of all wild atoms.

The finite Heisenberg theory used here is classical. We use the Stone von Neumann theorem, finite Weil representations, and standard projective representation theory [15, 25, 26, 22, 16]. The contribution is the way these results are arranged into one comparison diagram and the matrix holonomy theorem that controls finite gluing. In residue characteristic two, the alternating form does not determine the phase. A quadratic refinement and its Arf invariant are required. The underlying quadratic form theory is standard [24].

Local factors provide the final test. Henniart showed that pair epsilon factors characterize the local correspondence in the required range [18]. The tame packet calculations below recover exact local polynomials and finite Gauss phases for the declared tame atoms. They do not supply the complete pair factor comparison for arbitrary wild atoms. This boundary is retained throughout this chapter.

Arithmetic realization and identification of an actual candidate are separate problems. The realization endpoint is conjecture 66. The restricted constructions are developed in section 19 and the detailed construction appendices. They have ordinary mathematical proofs with retained internal reviews. Their formal coverage is stated separately in the formal companion.

The accompanying sources include detailed proofs, construction notes, proof reviews and research records. These documents distinguish established results, conditional results, and the remaining obligations associated with Conjecture 66.

This chapter is organized as follows. Weil Deligne chains appear in section 2. Tame Frobenius packets, finite field cuspidal projectors, tensor refinement, and Gauss phases are developed in sections 3 to 6. Finite Heisenberg groups and projective commutator envelopes appear in sections 7 and 8. The restricted one break Galois envelope and its comparison with the type side are treated in

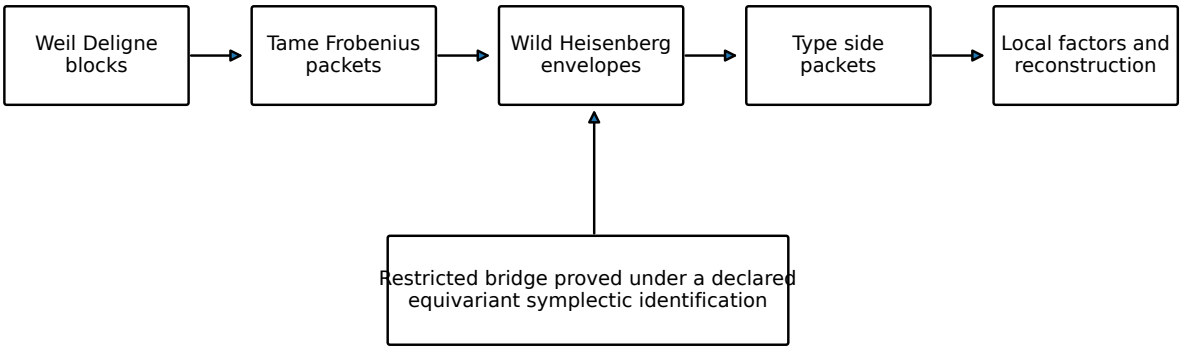


Figure 1. The finite theorem layers and the remaining wild boundary.

sections 9 to 11. Odd and dyadic phase data are separated in sections 12 and 13. Matrix holonomy and breakwise reconstruction appear in sections 14 and 15. The boundary for local factors appears in section 16. Examples and scope follow in sections 17 and 18. Section 19 and the appendices develop the restricted arithmetic constructions. The separate progress paper reconciles subsequent routes, corrections and remaining obligations without replacing these source records.

2 Weil Deligne chains

Let W_K be the Weil group of K, let I_K be inertia, let P_K be wild inertia, and choose a geometric Frobenius lift Φ. A Weil Deligne representation is a pair (r, N) with

$$r(w)Nr(w)^{-1} = |w|_K N.$$

Standard Fact 2.1 (Indecomposable monodromy blocks)**.** Every Frobenius semisimple Weil Deligne representation is a direct sum of blocks

$$\mathrm{Sp}_m(\tau),$$

where τ is an irreducible Weil representation and the nilpotent operator is one Jordan chain of length m.

This standard decomposition reduces the construction of a local correspondence to two tasks. One must understand irreducible Weil atoms, then one must attach the correct generalized Steinberg chain. The second task is formal once the first is known.

Definition 2.2 (Monodromy packet chain)**.** Let $V_0, \dots, V_{m-1}$ be copies of the space of τ. Put

$$V = \bigoplus_{j=0}^{m-1} V_j.$$

A monodromy packet chain is a nilpotent operator N with

$$N(V_j) \subseteq V_{j-1}$$

for $j > 0$, with $N(V_0) = 0$, together with a Frobenius operator whose conjugation scales N by q^{-1}.

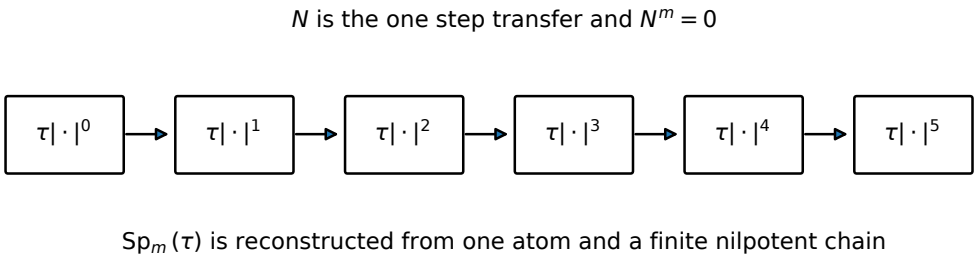


Figure 2. A Weil Deligne chain is reconstructed from one atom and finitely many nearest neighbor transfers.

Proposition 2.3 (Exact block reconstruction)**.** *Let P_j be the projector onto V_j. Then*

$$N = \sum_{j=1}^{m-1} P_{j-1} N P_j, \qquad N^m = 0.$$

Every power N^r is the sum of the packet paths of length r along the chain.

Proof. The projectors resolve the identity. Since N maps only from V_j to V_{j-1}, every block except the displayed nearest neighbor blocks vanishes. A product of r nonzero blocks moves down exactly r levels. No path of length m exists, so $N^m = 0$. □

Proposition 2.4 (Scalar covariance model)**.** *Let*

$$F = \mathrm{diag}(1, q, q^2, \dots, q^{m-1})$$

and let $N_{j-1,j} = 1$. Then

$$FNF^{-1} = q^{-1}N.$$

Proof. The only nonzero entry in column j and row $j-1$ is multiplied by $q^{j-1}q^{-j} = q^{-1}$ under conjugation. □

Theorem 2.5 (Formal atomic reconstruction)**.** *Assume that every irreducible Weil representation τ has been assigned a supercuspidal representation σ_τ, compatibly with isomorphism and unramified twist. Then the rules*

$$\mathrm{Sp}_m(\tau) \longmapsto \mathrm{St}_m(\sigma_\tau)$$

and normalized parabolic induction determine a candidate correspondence on all Frobenius semisimple Weil Deligne parameters.

Proof. Decompose the parameter into indecomposable monodromy blocks. Replace each Weil atom by its assigned supercuspidal atom and each monodromy chain by the corresponding generalized Steinberg block. Order the resulting essentially square integrable factors in Langlands order and take the unique irreducible quotient. The multiset of indecomposable Weil Deligne blocks is intrinsic, so the result does not depend on the chosen display. □

Remark 2.6. Theorem 2.5 is a reconstruction theorem. It does not produce the atomic assignment $\tau \mapsto \sigma_\tau$, and it does not prove the required local factor identities.

3 Tame Frobenius packets

Assume that P_K acts trivially. After passing to a finite quotient, tame inertia is cyclic. Let t generate a quotient of order M, and put $\zeta_M = \mathrm{e}^{2\pi i/M}$. For $j \in \mathbb{Z}/M\mathbb{Z}$, define

$$P_j = \frac{1}{M} \sum_{a=0}^{M-1} \zeta_M^{-ja} r(t)^a.$$

Proposition 3.1 (Tame character projectors)**.** *The operators P_j are pairwise orthogonal idempotents whose sum is the identity. Frobenius permutes them by*

$$\Phi P_j \Phi^{-1} = P_{qj}.$$

Proof. The first assertion is the Fourier idempotent calculation for a cyclic group. For the geometric Frobenius convention of section 2, the relation is $\Phi^{-1} t \Phi = t^q$. Equivalently, $\Phi t \Phi^{-1} = t^{q^{-1}}$, with inverse taken modulo M since $\gcd(q, M) = 1$. Conjugating the Fourier sum and changing its summation index therefore gives $\Phi P_j \Phi^{-1} = P_{qj}$. The inverse Frobenius convention reverses the label direction while preserving the same orbits. □

Theorem 3.2 (Tame orbit induction)**.** *Let τ be an irreducible complex representation of W_K with trivial wild inertia. Then the nonzero tame character packets form one transitive Frobenius orbit of length f. If K_f/K is the unramified extension of degree f, there is a character ξ of W_{K_f} such that*

$$\boxed{\tau \cong \mathrm{Ind}_{W_{K_f}}^{W_K} \xi.}$$

Proof. The nonzero tame packets are permuted by Frobenius. A union of Frobenius orbits is stable under W_K, so irreducibility forces one orbit. Choose one packet V_θ. Its stabilizer is $W_{K_f} = \langle I_K, \Phi^f \rangle$. A nonzero proper stable subspace of V_θ would generate a nonzero proper W_K stable subspace after summing its Frobenius translates. Thus V_θ is irreducible for the stabilizer. Inertia acts on it by the scalar character θ, and the remaining action is generated by Φ^f. The stabilizer action on this packet is therefore one dimensional. Induction reconstructs the complete orbit. □

Corollary 3.3 (Dimension and regularity)**.** *The dimension of τ equals the Frobenius orbit length of its tame inertia character. The inducing character is regular, meaning that no smaller positive power of Frobenius fixes it.*

Proof. The induced representation has one line for every Frobenius translate of the chosen character. A shorter orbit would produce a proper stabilizer and contradict the minimality of f. □

Proposition 3.4 (Dual and self dual tame packets)**.** *For a tame atom*

$$\tau_\xi = \mathrm{Ind}_{W_{K_f}}^{W_K} \xi,$$

we have

$$\tau_\xi^\vee \cong \mathrm{Ind}_{W_{K_f}}^{W_K} \xi^{-1}.$$

The atom is self dual exactly when

$$\xi^{-1} = \xi^{\Phi^j}$$

for some $j \in \mathbb{Z}/f\mathbb{Z}$.

Proof. Duality commutes with finite index induction, giving the first statement. Two regular induced tame atoms are isomorphic exactly when their inducing characters lie in the same Frobenius orbit. Applying this to ξ and ξ^{-1} gives the criterion. □

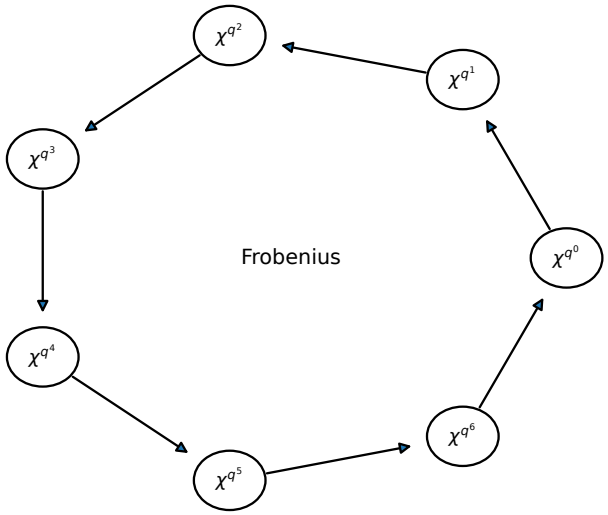


Figure 3. A tame irreducible atom is one transitive Frobenius orbit.

Proposition 3.5 (Unramified twist)**.** *Let ν be an unramified character of W_K. Then*

$$\tau_\xi \otimes \nu \cong \operatorname{Ind}_{W_{K_f}}^{W_K} \left(\xi\, \nu|_{W_{K_f}} \right).$$

The tame inertia orbit is unchanged. If the original Frobenius cycle has closing scalar a, the twisted cycle has closing scalar

$$a\nu(\Phi)^f.$$

Proof. The tensor identity follows from the projection formula for induction. Since ν is trivial on inertia, it does not change any tame character packet. It multiplies every Frobenius step by $\nu(\Phi)$, so a closed cycle of length f gains the factor $\nu(\Phi)^f$. □

3.1 Regular orbit counts

Characters of $\mathbb{F}_{q^n}^\times$ are indexed by exponents modulo $q^n - 1$, and Frobenius acts by multiplication by q.

Proposition 3.6 (Möbius orbit count)**.** *The number of regular Frobenius orbits of length n is*

$$\boxed{c_n(q) = \frac{1}{n} \sum_{d \mid n} \mu(d) \left(q^{n/d} - 1 \right).}$$

Proof. For every divisor $r \mid n$, the characters fixed by Frob^r are exactly those descending to $\mathbb{F}_{q^r}^\times$, and there are $q^r - 1$ of them. Möbius inversion gives the number of characters of exact orbit length n. Division by n gives the number of orbits. □

Example 3.7 (Degree four over $\mathbb{F}_2$)**.** The regular exponent orbits modulo fifteen are

$$(1, 2, 4, 8), \qquad (3, 6, 12, 9), \qquad (7, 14, 13, 11).$$

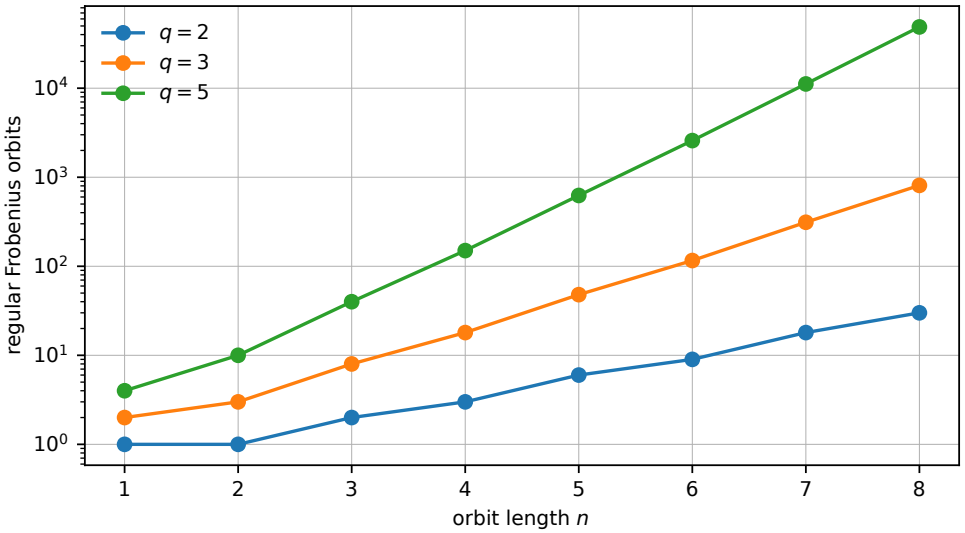


Figure 4. Regular Frobenius orbit counts for three residue field sizes.

The formula gives

$$\frac{1}{4}\big((2^4-1)-(2^2-1)\big)=3.$$

4 Finite field cuspidal projectors

Let

$$G_n=\mathrm{GL}_n(\mathbb{F}_q),\qquad T_n\cong\mathbb{F}_{q^n}^{\times}$$

with T_n an elliptic maximal torus.

Standard Fact 4.1 (Regular elliptic torus characters)**.** A regular character $\theta : T_n \to \mathbb{C}^{\times}$ determines an irreducible cuspidal representation κ_θ of G_n. Frobenius conjugate characters determine isomorphic representations, and every irreducible cuspidal representation of G_n occurs in this way.

This is the finite field input from Deligne and Lusztig theory [13]. We use standard finite group character theory for the central idempotents [28, 21]. Once κ_θ is known, its block is selected by a central idempotent.

Proposition 4.2 (Primitive cuspidal idempotent)**.** *Let χ_θ be the irreducible character of κ_θ, and let $d_\theta=\chi_\theta(1)$. Then*

$$e_\theta=\frac{d_\theta}{|G_n|}\sum_{g\in G_n}\chi_\theta(g^{-1})g$$

is the primitive central idempotent selecting the κ_θ block in $\mathbb{C}[G_n]$.

Proof. On an irreducible representation of character ψ, the central element acts by a scalar. Taking traces shows that the scalar is

$$\frac{d_\theta}{|G_n|\,\psi(1)}\sum_{g\in G_n}\chi_\theta(g^{-1})\psi(g).$$

Character orthogonality makes this scalar one for $\psi=\chi_\theta$ and zero otherwise. The element is therefore central, idempotent, and primitive. □

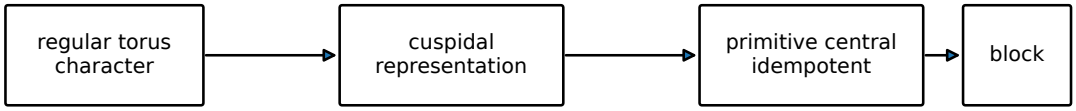


Figure 5. A regular torus character determines a cuspidal block and its primitive central projector.

Proposition 4.3 (Parabolic annihilation)**.** *Let $P = LU$ be a proper parabolic subgroup of G_n, and put*

$$e_U = \frac{1}{|U|} \sum_{u \in U} u.$$

If κ_θ is cuspidal, then

$$\boxed{e_U e_\theta = e_\theta e_U = 0.}$$

Proof. The operator e_U projects onto the subspace of U fixed vectors. A cuspidal representation has no nonzero fixed vector under the unipotent radical of a proper parabolic. Thus e_U acts by zero on the κ_θ block. Since e_θ is central, both products vanish. □

4.1 Depth zero reconstruction and cycle orientation

The depth zero construction inflates a finite cuspidal representation to $\mathrm{GL}_n(\mathcal{O}_K)$, extends it to the standard compact modulo center subgroup, and then compactly induces it to $\mathrm{GL}_n(K)$. This is classical type theory [11, 8].

The induced tame Weil atom has an elementary cycle invariant. In an orbit basis $e_0, \dots, e_{n-1}$, write

$$\tau(\Phi) e_j = e_{j+1}$$

for $j < n - 1$, and

$$\tau(\Phi) e_{n-1} = a e_0.$$

Proposition 4.4 (Frobenius cycle polynomial)**.** *The Frobenius transfer matrix has characteristic polynomial*

$$X^n - a$$

and determinant

$$\boxed{\det \tau(\Phi) = (-1)^{n-1} a.}$$

Proof. The matrix is the companion matrix of $X^n - a$. Its only nonzero determinant term is the closed n cycle. That cycle has sign $(-1)^{n-1}$, and the closing edge contributes a. □

Corollary 4.5 (Unramified orientation character)**.** *The sign $(-1)^{n-1}$ is the orientation of the closed Frobenius packet cycle. It is unchanged by rescaling the individual orbit basis vectors.*

Proof. A basis rescaling conjugates the transfer matrix by a diagonal matrix, so the determinant and characteristic polynomial remain unchanged. □

5 Tensor refinement of tame atoms

Let

$$\tau_\xi = \operatorname{Ind}_{W_{K_n}}^{W_K} \xi, \qquad \tau_\eta = \operatorname{Ind}_{W_{K_m}}^{W_K} \eta,$$

where K_n/K and K_m/K are unramified. Put

$$d = \gcd(n,m), \qquad L = \operatorname{lcm}(n,m).$$

Theorem 5.1 (Greatest common divisor and least common multiple packets)**.** *There is a decomposition*

$$\boxed{\tau_\xi \otimes \tau_\eta \cong \bigoplus_{i=0}^{d-1} \operatorname{Ind}_{W_{K_L}}^{W_K} \left(\xi_{L,n}\eta_{L,m}^{\Phi^i}\right),}$$

where $\xi_{L,n}$ and $\eta_{L,m}$ are the norm lifts to K_L.

Proof. Apply Mackey decomposition [23]. The double cosets

$$W_{K_n} \backslash W_K / W_{K_m}$$

are the orbits of simultaneous translation on $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$. There are d orbits, each of length L. The stabilizer of a point in each orbit is W_{K_L}. The second character is shifted by the chosen Frobenius representative. Summing the induced terms over the double cosets gives the decomposition. □

Corollary 5.2 (Packet count and dimension)**.** *The tensor product has d induced interaction packets, each with carrier dimension and Frobenius coset-cycle length L. The interaction character may have a shorter inertia orbit, in which case a packet may split further into irreducible atoms. The total dimension is preserved because*

$$dL = nm.$$

Proof. The double coset calculation in theorem 5.1 gives exactly d summands, and every stabilizer is W_{K_L}, so every induced packet has dimension L. The identity $dL = nm$ follows from the definitions of greatest common divisor and least common multiple. □

5.1 Exponent coordinates

Represent the residue characters by exponents

$$a \pmod{q^n - 1}, \qquad b \pmod{q^m - 1}.$$

Proposition 5.3 (Refined exponent formula)**.** *The ith interaction packet on $\mathbb{F}_{q^L}^\times$ has exponent*

$$\boxed{c_i = a\frac{q^L - 1}{q^n - 1} + bq^i\frac{q^L - 1}{q^m - 1} \pmod{q^L - 1}.}$$

It is trivial on tame inertia exactly when $c_i = 0$.

Proof. Norm lifting a character exponent from $\mathbb{F}_{q^n}^\times$ to $\mathbb{F}_{q^L}^\times$ multiplies the exponent by $(q^L - 1)/(q^n - 1)$. The Frobenius shift multiplies the second exponent by q^i. Multiplication of characters adds exponents. Triviality is equivalent to the resulting exponent being zero modulo $q^L - 1$. □

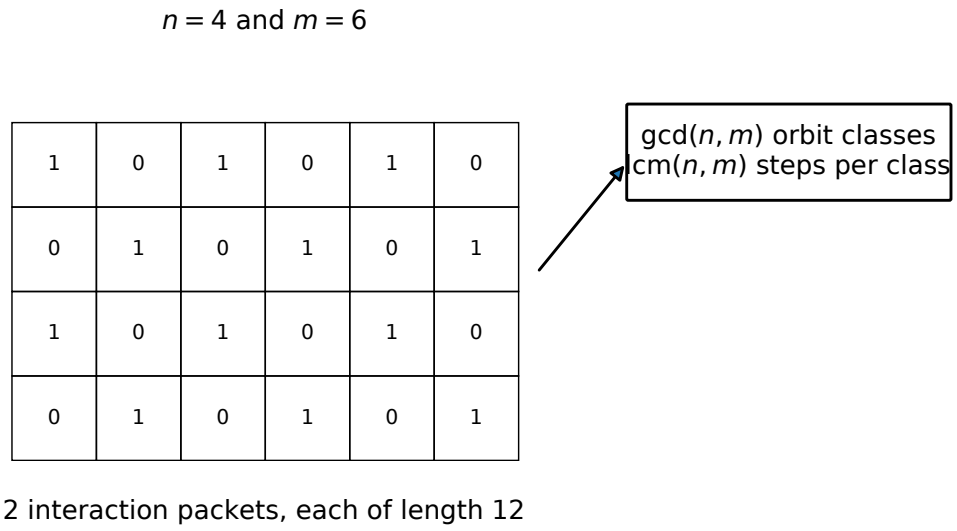


Figure 6. Tensor refinement gives as many packets as the greatest common divisor, each with carrier dimension equal to the least common multiple.

5.2 Tame local polynomials

Let V be one interaction packet. If its inertia character is nontrivial, then $V^{I_K} = 0$. If the inertia character is trivial, Frobenius acts on the invariant packet by a companion cycle.

Proposition 5.4 (Local polynomial of one tame packet)**.** *Suppose an interaction packet has orbit length r and closing scalar a. Its local polynomial is*

$$\det\left(I - u\Phi \mid V^{I_K}\right) = \begin{cases} 1 - au^r, & \text{if the inertia character is trivial,} \\ 1, & \text{otherwise.} \end{cases}$$

Proof. For nontrivial inertia, the invariant space is zero. In the trivial case, Frobenius on the invariant packet is the companion matrix of $X^r - a$. Substituting $X = u^{-1}$ and multiplying by u^r gives $1 - au^r$. □

Corollary 5.5 (Tame tensor local factor)**.** *The local factor of the tame tensor product is the product of*

$$(1 - a_i u^{r_i})^{-1}$$

over precisely those refined packets whose inertia exponent vanishes.

Proof. The invariant space is the direct sum of the invariant refined packets, and determinant is multiplicative on direct sums. □

Proposition 5.6 (Trivial tensor packet criterion)**.** *For every refined interaction packet, let r_i be its cycle length and a_i its closing scalar. The dimension of the W_K fixed space in $\tau_\xi \otimes \tau_\eta$ equals the number of packets satisfying both*

$$c_i = 0 \qquad \text{and} \qquad a_i = 1.$$

Proof. A nonzero fixed vector must lie in an inertia trivial packet, so $c_i = 0$. On that packet, Frobenius is a companion cycle with characteristic polynomial $X^{r_i} - a_i$. Its fixed space is the eigenspace for eigenvalue one. This eigenspace is one dimensional exactly when $a_i = 1$, and it is zero otherwise. Summing over the refined packets proves the formula. □

Corollary 5.7 (Tame Hom test)**.** *For irreducible tame atoms,*

$$\dim \mathrm{Hom}_{W_K}(\tau_\xi^\vee, \tau_\eta)$$

is zero or one, and it is one exactly when the refined tensor packet test in proposition 5.6 finds one trivial packet.

Proof. The Hom space is the fixed space of $\tau_\xi \otimes \tau_\eta$. Schur's lemma gives dimension zero or one, and proposition 5.6 gives the packet criterion. □

5.3 Tame conductor

Proposition 5.8 (Conductor of a nontrivial tame atom)**.** *Let τ be an irreducible tame Weil representation of dimension $n > 1$. Then*

$$\mathrm{Swan}(\tau) = 0, \qquad \dim \tau^{I_K} = 0, \qquad \mathrm{Art}(\tau) = n.$$

Proof. Wild inertia acts trivially, so the Swan conductor is zero. An inertia fixed vector would generate a nonzero Frobenius stable sum of trivial character packets. Irreducibility would then make inertia act trivially on all of τ. An irreducible representation of the quotient generated by Frobenius would be one dimensional, contrary to $n > 1$. The Artin conductor is therefore

$$\dim \tau - \dim \tau^{I_K} + \mathrm{Swan}(\tau) = n.$$

□

6 Finite Gauss phases

Let $k = \mathbb{F}_q$, let $\psi : k \to \mathbb{C}^\times$ be a nontrivial additive character, and let $\chi : k^\times \to \mathbb{C}^\times$ be a multiplicative character. Define

$$G(\chi, \psi) = \sum_{x \in k^\times} \chi(x)\psi(x).$$

The standard theory of Gauss and Jacobi sums will be used in the form recorded in [12, 1].

Proposition 6.1 (Gauss magnitude)**.** *If χ is nontrivial, then*

$$\boxed{|G(\chi, \psi)| = q^{1/2}.}$$

Proof. Compute

$$|G(\chi, \psi)|^2 = \sum_{x,y \in k^\times} \chi(xy^{-1})\psi(x - y).$$

Write $x = ty$. The inner sum over y equals $q - 1$ when $t = 1$ and equals -1 otherwise. Since $\sum_{t \in k^\times} \chi(t) = 0$, the result is q. □

Definition 6.2 (Normalized Gauss phase)**.** For nontrivial χ, define

$$\gamma(\chi, \psi) = q^{-1/2} G(\chi, \psi).$$

Then $|\gamma(\chi, \psi)| = 1$.

Proposition 6.3 (Duality of Gauss phases)**.** *For nontrivial χ,*

$$\boxed{\gamma(\chi, \psi)\gamma(\chi^{-1}, \psi) = \chi(-1).}$$

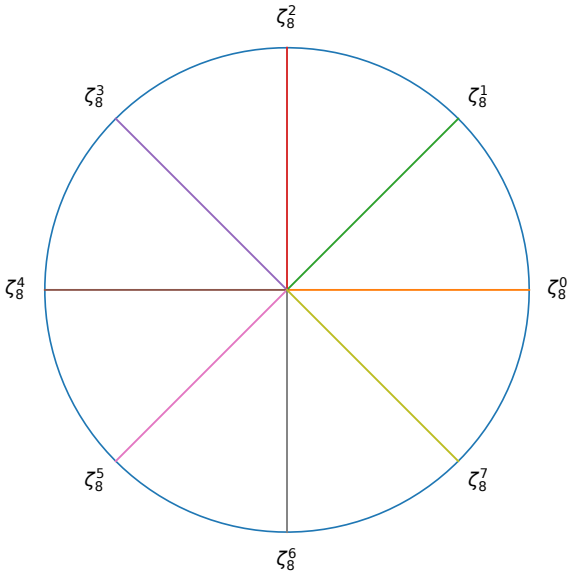


Figure 7. Normalized finite Gauss sums lie on the unit circle.

Proof. The standard Gauss identity gives

$$G(\chi,\psi)G(\chi^{-1},\psi) = \chi(-1)q.$$

Divide by q. □

Standard Fact 6.4 (Hasse and Davenport lifting)**.** Let k_r/k be the extension of degree r. Lift χ by the norm and ψ by the trace. Then the lifted Gauss sum satisfies the Hasse and Davenport product relation, with a sign determined by the chosen additive character normalization.

Remark 6.5. Finite Gauss phases reproduce the finite field part of tame and depth zero normalization. They do not by themselves give the full epsilon factor of a general wild pair.

7 Finite Heisenberg packets

Let p be odd. Let X be an r dimensional vector space over $\mathbb{F}_p$, and put

$$V = X \oplus X^*.$$

Define the alternating form

$$b\big((x,\xi),(y,\eta)\big) = \eta(x) - \xi(y).$$

Let $Z = \mathbb{F}_p$, and define the Heisenberg group

$$\mathfrak{H}(V) = V \times Z$$

with multiplication

$$(v,z)(w,t) = \left(v+w, z+t+\frac{1}{2}b(v,w)\right).$$

Fix the additive character

$$\psi(z) = \mathrm{e}^{2\pi i z/p}.$$

Definition 7.1 (Schrödinger operators)**.** On $\mathcal{H} = \mathbb{C}^X$, define

$$\big(\rho(x,\xi,z)f\big)(u) = \psi\left(z + \xi(u) + \frac{1}{2}\xi(x)\right) f(u+x).$$

Theorem 7.2 (Finite Schrödinger representation)**.** *The map $\rho : \mathfrak{H}(V) \to \mathrm{GL}(\mathcal{H})$ is a representation. It is irreducible, its center acts by ψ, and*

$$\dim \mathcal{H} = p^r = \sqrt{|V|}.$$

Proof. A direct calculation gives

$$\rho(v,z)\rho(w,t) = \rho\left(v+w, z+t+\frac{1}{2}b(v,w)\right).$$

Translations by X and multiplications by characters from X^* generate the full matrix algebra on $\mathbb{C}^X$. Indeed, the character multipliers give all diagonal projectors by finite Fourier inversion, and translations move each coordinate line to every other coordinate line. The commutant is therefore scalar, so the representation is irreducible. The dimension is $|X| = p^r$. □

Proposition 7.3 (Character support)**.** *The character of the Schrödinger representation satisfies*

$$\boxed{\operatorname{Tr}\rho(v,z) = \begin{cases} p^r\psi(z), & v = 0, \\ 0, & v \neq 0. \end{cases}}$$

Proof. If the translation component x is nonzero, the operator permutes the standard basis without a fixed point, so its trace is zero. If $x = 0$ but $\xi \neq 0$, the trace is

$$\psi(z)\sum_{u\in X}\psi(\xi(u)) = 0.$$

When $x = \xi = 0$, the operator is the scalar $\psi(z)$. □

Standard Fact 7.4 (Finite Stone von Neumann theorem)**.** For a finite Heisenberg group with nondegenerate commutator form and a fixed nontrivial central character, there is one irreducible representation up to isomorphism. Its dimension is the square root of the order of the symplectic quotient.

The standard fact follows from the twisted group algebra argument in section 8. It is also the finite form used in the classical Weil representation literature [15, 26].

7.1 Maximal isotropic selectors

Let $L \subseteq V$ be maximal isotropic. Choose a subgroup lift $\widetilde{L} \subseteq \mathfrak{H}(V)$ on which the central coordinate is zero.

Proposition 7.5 (Isotropic averaging projector)**.** *The operator*

$$P_L = \frac{1}{|L|}\sum_{\ell\in\widetilde{L}}\rho(\ell)$$

is an orthogonal projector. In the Schrödinger model with $L = X$, it has rank one and selects the constant functions.

Proof. The lifted isotropic subgroup is abelian, so group averaging is an idempotent. Every operator in the average is unitary, and the subgroup is closed under inversion, so the average is self adjoint. For $L = X$, the average is the translation average

$$(P_L f)(u) = \frac{1}{|X|}\sum_{x\in X} f(u+x),$$

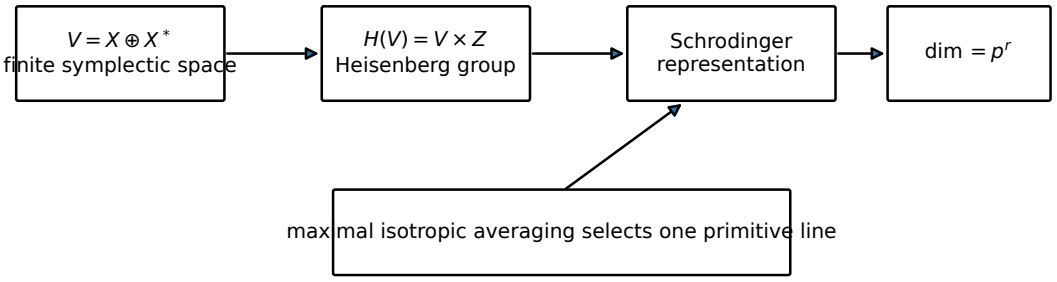


Figure 8. The finite Heisenberg packet and its maximal isotropic selector.

which is the constant function with value equal to the mean of f. Its range is one dimensional. □

Corollary 7.6 (Primitive line selector)**.** *Every maximal isotropic subgroup selects a one dimensional fixed line in the irreducible Heisenberg representation.*

Proof. The projector rank equals its trace. The identity element contributes p^r, while every nonidentity element of the lifted isotropic subgroup is noncentral and has trace zero by proposition 7.3. Since $|L| = p^r$, we get

$$\operatorname{Tr} P_L = \frac{p^r}{p^r} = 1.$$

Thus the projector has rank one. □

8 Twisted group algebras and commutator envelopes

Let A be a finite abelian group, and let

$$c : A \times A \to \mathbb{C}^\times$$

be a normalized two cocycle. The twisted group algebra $\mathbb{C}^c[A]$ has basis $(u_a)_{a \in A}$ and multiplication

$$u_a u_b = c(a,b) u_{a+b}.$$

Define the commutator bicharacter

$$b_c(a,b) = c(a,b)c(b,a)^{-1}.$$

Proposition 8.1 (Scalar choice invariance)**.** *Replacing u_a by $\lambda_a u_a$, with $\lambda_a \in \mathbb{C}^\times$, changes c by a coboundary and leaves b_c unchanged.*

Proof. The new cocycle is

$$c'(a,b) = \lambda_a \lambda_b \lambda_{a+b}^{-1} c(a,b).$$

The scalar coboundary factor is symmetric in a and b, so it cancels from $c'(a,b)c'(b,a)^{-1}$. □

Proposition 8.2 (Alternating commutator form)**.** *The map b_c is an alternating bicharacter. Its radical*

$$R = \{a \in A : b_c(a, x) = 1 \text{ for every } x \in A\}$$

is a subgroup, and b_c descends to a nondegenerate alternating bicharacter on A/R.

Proof. Associativity of the twisted product gives multiplicativity of scalar commutators in each variable. We have $b_c(a, a) = 1$ and $b_c(b, a) = b_c(a, b)^{-1}$. The radical is the kernel of the homomorphism from A to its character group defined by $a \mapsto b_c(a, \cdot)$. The induced pairing on the quotient is nondegenerate by construction. □

Theorem 8.3 (Nondegenerate twisted group algebra)**.** *Assume that b_c is nondegenerate. Then $|A|$ is a square, and*

$$\boxed{\mathbb{C}^c[A] \cong M_{\sqrt{|A|}}(\mathbb{C}).}$$

Consequently, $\mathbb{C}^c[A]$ has one irreducible module up to isomorphism, of dimension $\sqrt{|A|}$.

Proof. Let

$$z = \sum_{a \in A} z_a u_a$$

be central. Commuting with u_x gives

$$z_a b_c(x, a) = z_a$$

for every x. Nondegeneracy forces $z_a = 0$ for every nonzero a. Thus the center is $\mathbb{C}$. To prove semisimplicity directly, choose a linear projection P onto a submodule of a finite dimensional $\mathbb{C}^c[A]$-module and form

$$\overline{P} = \frac{1}{|A|} \sum_{a \in A} u_a P u_a^{-1}.$$

Every summand maps into the submodule and is the identity there. Conjugation by any u_x permutes the summands: the cocycle scalars cancel between each operator and its inverse. Hence $\overline{P}$ is an equivariant projection, and its kernel is an invariant complement. The algebra is therefore semisimple. Its center $\mathbb{C}$ permits only one simple factor. A finite dimensional central simple algebra over $\mathbb{C}$ is a full matrix algebra. Its dimension is $|A|$, so the matrix size is $\sqrt{|A|}$. □

Corollary 8.4 (Radical free envelope)**.** *For an arbitrary scalar projective action of A, the quotient*

$$\boxed{A^{\mathrm{symp}} = A/\operatorname{rad} b_c}$$

is the maximal quotient detected by noncommuting projective intertwiners. If A is an elementary abelian p group, then A^{symp} is a symplectic $\mathbb{F}_p$ space.

Proof. Every radical element commutes with all projective generators, so it contributes only central data. Quotienting by the radical leaves the nondegenerate commutator form. In the elementary abelian case, multiplicative values in the group of pth roots can be written as an alternating bilinear form over $\mathbb{F}_p$. □

Remark 8.5. The envelope depends only on the projective representation. It does not by itself prove that a given wild inertia representation carries a canonical finite quotient with the required projective action.

9 A restricted one break Galois envelope

We now isolate a finite Clifford setting in which the quotient projective action is defined on a multiplicity space. This avoids assigning a quotient action directly to arbitrary representatives of G/N.

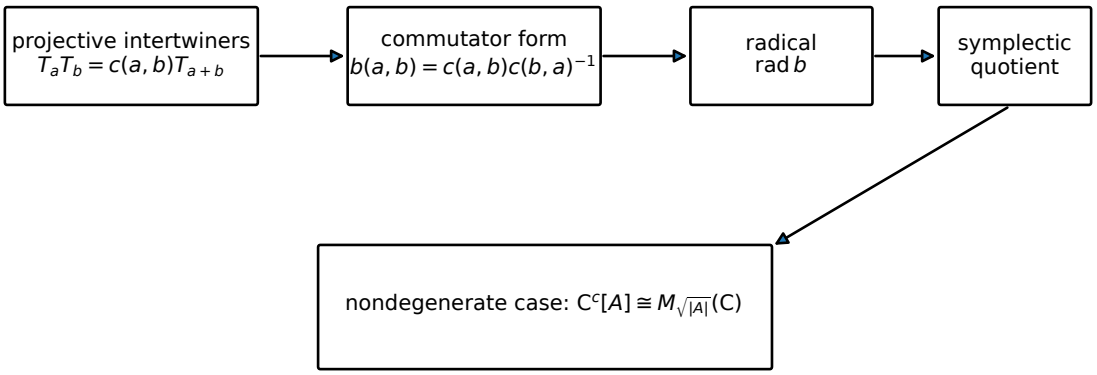


Figure 9. Projective intertwiners determine a radical free symplectic envelope.

Definition 9.1 (One break Clifford datum)**.** A one break Clifford datum consists of finite groups and representations

$$N \triangleleft G, \qquad A = G/N, \qquad \Pi : G \to \mathrm{GL}(\mathcal{X}), \qquad \rho : N \to \mathrm{GL}(W),$$

with the following properties.

(1) The quotient A is a finite elementary abelian p group.
(2) The representations Π and ρ are irreducible.
(3) The isomorphism class of ρ is fixed by G.
(4) The restriction of Π to N is ρ isotypic.
(5) The quotient records one declared ramification break.

The multiplicity space is

$$M_\rho = \mathrm{Hom}_N(W, \mathcal{X}).$$

The isotypic condition gives an N equivariant evaluation isomorphism

$$W \otimes M_\rho \longrightarrow \mathcal{X}.$$

Choose a section $s : A \to G$. For every $a \in A$, choose an intertwiner I_a from ρ to its conjugate by $s(a)$. Under the evaluation isomorphism, there is a unique operator T_a on M_ρ such that

$$\Pi(s(a)) = I_a \otimes T_a.$$

The operators T_a form a projective representation of A.

Proposition 9.2 (Clifford multiplicity action)**.** *The projective equivalence class of the action $(T_a)_{a\in A}$ is independent of the section, the intertwiners, and the chosen isotypic evaluation isomorphism. The projective action is irreducible.*

Proof. Changing an intertwiner I_a by a scalar changes T_a by the inverse scalar. Changing the isotypic evaluation isomorphism conjugates all T_a by one element of $\mathrm{GL}(M_\rho)$. Changing the section multiplies

$s(a)$ by an element of N. That factor is absorbed by the ρ action on W, leaving a scalar change on the multiplicity operator by Schur's lemma. Thus all choices give projectively equivalent actions.

If a nonzero proper subspace of M_ρ were stable under all projective operators T_a, then its tensor product with W would give a nonzero proper G stable subspace of $\mathcal{X}$. This contradicts irreducibility of Π. □

Definition 9.3 (Clifford cocycle class)**.** Choose scalars $c(a,b)$ such that

$$T_a T_b = c(a,b) T_{a+b}.$$

The cohomology class

$$[c_\rho] \in H^2(A, \mathbb{C}^\times)$$

is the Clifford cocycle class of the datum.

Proposition 9.4 (Intrinsic commutator pairing)**.** *The pairing*

$$b_\rho(a,b) = c(a,b)c(b,a)^{-1}$$

is independent of every choice in definition 9.1. It is an alternating bicharacter

$$b_\rho : A \times A \to \mu_p.$$

Proof. The projective equivalence class is intrinsic by proposition 9.2. Conjugation does not change scalar commutators, and scalar changes alter the cocycle by a coboundary. The commutator pairing is therefore unchanged by proposition 8.1. Its alternating and bilinear properties follow from proposition 8.2. Since every element of A has order p, every value has order dividing p. □

Definition 9.5 (One break symplectic envelope)**.** The one break symplectic envelope is

$$\boxed{V_\rho = A / \operatorname{rad} b_\rho.}$$

Theorem 9.6 (Nondegenerate one break packet)**.** *If b_ρ is nondegenerate, then*

$$\mathbb{C}^{c_\rho}[A] \cong \operatorname{End}(M_\rho), \qquad \dim M_\rho = \sqrt{|A|}.$$

Proof. The projective action gives a nonzero representation of the twisted group algebra on M_ρ. The algebra is a full matrix algebra by theorem 8.3. Its unique irreducible module has dimension $\sqrt{|A|}$. The projective multiplicity action is irreducible by proposition 9.2, so it is that module. □

Proposition 9.7 (Isomorphism invariance)**.** *The cocycle class and the equivariant isometry class of V_ρ depend only on the isomorphism class of the one break Clifford datum.*

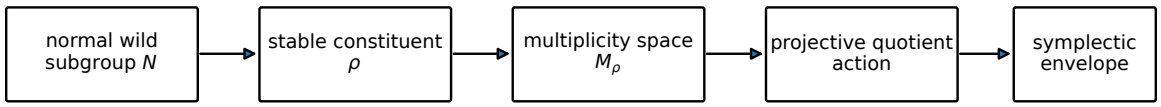
Proof. An isomorphism of data transports the restriction, the multiplicity space, and its projective quotient action. It therefore transports the cocycle class, the commutator pairing, its radical, and the quotient. □

Proposition 9.8 (Tame stabilizer action)**.** *Suppose a finite group C acts by automorphisms of the one break Clifford datum. Then C acts on V_ρ by symplectic automorphisms.*

Proof. The automorphism action transports the intrinsic projective multiplicity representation. It therefore preserves the scalar commutator pairing and its radical. The induced action on the quotient is symplectic. □

Corollary 9.9 (Heisenberg representation attached to one break)**.** *Fix a nontrivial central character. The envelope V_ρ determines one irreducible finite Heisenberg representation up to isomorphism.*

Proof. Apply the finite Stone von Neumann theorem to the nondegenerate symplectic quotient. □



The construction is intrinsic for the declared one break Clifford datum, up to equivariant isometry.

Figure 10. The restricted one break construction is made on the Clifford multiplicity space.

Remark 9.10. The group cohomology class records the extension obstruction for the multiplicity action. General background on projective representations and group cohomology can be found in [22, 3]. The theorem remains restricted because the finite quotient and the one break isotypic datum are assumed rather than constructed from arbitrary wild inertia.

10 The type side symplectic quotient

Let θ be a maximal simple character for $\mathrm{GL}_n(K)$, with compact open subgroups

$$H^1_\theta \subseteq J^1_\theta \subseteq J_\theta.$$

The quotient

$$V_\theta = J^1_\theta / H^1_\theta$$

is regarded as a finite vector space over $\mathbb{F}_p$, where p is the residue characteristic. Fix a nontrivial additive character $\psi_0 : \mathbb{F}_p \to \mu_p \subset \mathbb{C}^\times$, where μ_p is the group of p-th roots of unity. The simple-character commutator takes values in μ_p. Since ψ_0 is injective, the equation

$$\psi_0\big(h_\theta(\bar{x}, \bar{y})\big) = \theta([x, y])$$

uniquely defines the $\mathbb{F}_p$-valued commutator pairing h_θ. We use the same identification of μ_p with the additive group of $\mathbb{F}_p$ on the Galois side when comparing the two symplectic spaces.

Standard Fact 10.1 (Type side Heisenberg quotient)**.** The pairing h_θ is nondegenerate and alternating. There is one irreducible representation η_θ of J^1_θ containing θ, and its dimension is

$$\sqrt{\left|J^1_\theta / H^1_\theta\right|}.$$

This is part of the simple character and Heisenberg representation theory used in the construction of types [11]. The extension of η_θ from J^1_θ to J_θ is not unique before a normalization is chosen.

Standard Fact 10.2 (Canonical beta extension)**.** The canonical beta extension is obtained from a primary extension together with finite symplectic and tame orientation corrections. The construction is functorial in the declared simple character data and removes the remaining extension ambiguity.

The normalization and its relation to finite Weil representations are developed in [2, 14]. This chapter uses this theorem as type side input. It does not rederive simple character theory.

Proposition 10.3 (Type side projective commutator)**.** *The projective action of J_θ / J_θ^1 on the Heisenberg representation preserves the commutator form on V_θ. Any two choices of projective lifts give the same alternating form.*

Proof. Conjugation by J_θ preserves θ, so it preserves the commutator pairing defined from $\theta([x, y])$. Two choices of projective lifts differ by scalars. The scalar commutator is therefore unchanged by proposition 8.1. □

11 The restricted Galois to type bridge

The full bridge would construct an intrinsic Galois envelope for every wild atom and identify it canonically with V_θ. We prove the comparison after the required symplectic identification has been supplied.

Definition 11.1 (Bridge datum)**.** A restricted bridge datum consists of

(1) a one break Galois envelope (V_ρ, b_ρ)
(2) a type side quotient (V_θ, h_θ)
(3) an equivariant symplectic isometry
$$\iota : V_\rho \longrightarrow V_\theta$$
for a finite cyclic stabilizer $C = \langle c \rangle$
(4) the same nontrivial central character on the two Heisenberg groups
(5) when $p = 2$, compatible quadratic refinements and an isomorphism of the resulting Heisenberg central extensions lifting ι.

Theorem 11.2 (Restricted Heisenberg bridge)**.** *A restricted bridge datum determines an isomorphism between the two Heisenberg representations, unique up to a nonzero scalar. More precisely, there is an invertible map*
$$U : \mathcal{H}_\rho \longrightarrow \mathcal{H}_\theta$$
such that
$$U\omega_\rho(h) = \omega_\theta(\iota_H(h))U$$
for every Heisenberg element h, and every other such map is a scalar multiple of U.

Proof. When p is odd, the symplectic isometry and the common central character induce the standard isomorphism ι_H of the two Heisenberg groups. When $p = 2$, the required lift of ι is part of the bridge datum because the alternating form alone does not determine the central extension. Pulling the type side representation back through ι_H gives an irreducible Heisenberg representation with the same central character as the Galois side representation. The finite Stone von Neumann theorem gives an isomorphism. Schur's lemma gives uniqueness up to scalar. □

The stabilizer actions need not agree after choosing U. Their discrepancy is scalar because both actions implement the same automorphisms of the irreducible Heisenberg representation.

Proposition 11.3 (Cyclic discrepancy character)**.** *Assume that the two projective stabilizer actions have been linearized to honest representations of the cyclic group C. Then there is a character*
$$\delta : C \to \mathbb{C}^\times$$

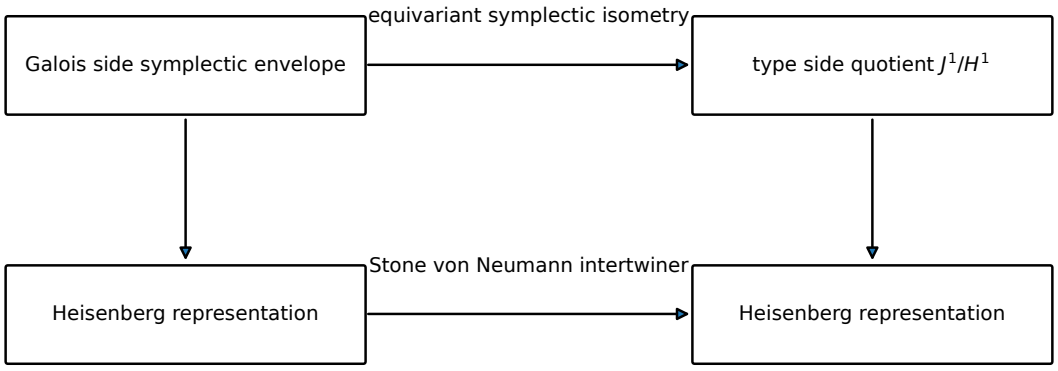


Figure 11. An equivariant symplectic isometry reduces the finite comparison to one scalar cyclic phase.

such that

$$U\Omega_\rho(g) = \delta(g)\Omega_\theta(g)U$$

for every $g \in C$.

Proof. For each g, the operator

$$\Omega_\theta(g)^{-1}U\Omega_\rho(g)U^{-1}$$

commutes with the type side Heisenberg representation. It is therefore scalar. Since both stabilizer actions are honest representations, multiplication of group elements shows that the scalars form a character. □

Lemma 11.4 (Linearization over a cyclic stabilizer)**.** *Every finite dimensional projective representation of a finite cyclic group over* $\mathbb{C}$ *is projectively equivalent to an honest representation.*

Proof. Let $C = \langle c \rangle$ have order m, and let T be a lift of the projective generator. Since $c^m = 1$, the operator T^m is scalar. Choose $\lambda \in \mathbb{C}^\times$ with $\lambda^m T^m = I$. Replacing T by λT gives an honest cyclic representation. □

Theorem 11.5 (One phase closes the cyclic bridge)**.** *In a restricted bridge datum, choose linearizations of the cyclic stabilizer actions. If the two actions have the same scalar phase on the generator c, then the Heisenberg intertwiner can be chosen to intertwine the full semidirect product action.*

Proof. By proposition 11.3, the discrepancy is a character δ. Equality of the generator phase gives $\delta(c) = 1$. Since c generates C, the character is trivial. The chosen map U then intertwines every stabilizer element as well as the Heisenberg group. □

Corollary 11.6 (Bridge uniqueness)**.** *When the semidirect product representation is irreducible, the completed bridge is unique up to scalar.*

Proof. Any two completed bridges differ by an endomorphism commuting with the semidirect product representation. Schur's lemma applies. □

Remark 11.7. The theorem does not construct ι. It proves that, once an equivariant symplectic isometry exists, the remaining finite comparison is one dimensional for a cyclic stabilizer.

12 Odd residue characteristic phases

Let p be odd, let (V, b) be a finite symplectic $\mathbb{F}_p$ space of dimension $2r$, and let

$$g \in \mathrm{Sp}(V)$$

have no fixed vector. Fix one of the standard finite Weil normalizations used in the literature [15, 25, 16].

Standard Fact 12.1 (Fixed point free Weil trace)**.** Under the declared normalization,

$$\boxed{\mathrm{Tr}\,\omega(g) = \left(\frac{-1}{p}\right)^r \left(\frac{\det(g-I)}{p}\right).}$$

Here $\left(\frac{\cdot}{p}\right)$ denotes the quadratic character of $\mathbb{F}_p^\times$.

Definition 12.2 (Odd finite Weil phase)**.** For fixed point free g, define

$$\Phi_p(g, V) = \left(\frac{-1}{p}\right)^r \left(\frac{\det(g-I)}{p}\right).$$

Proposition 12.3 (Symplectic invariance)**.** *If*

$$\iota : (V, b) \longrightarrow (V', b')$$

is a symplectic isometry and $g' = \iota g \iota^{-1}$, then

$$\Phi_p(g, V) = \Phi_p(g', V').$$

Proof. The dimensions agree, and conjugate matrices have the same determinant of $g - I$. □

Proposition 12.4 (Orthogonal multiplicativity)**.** *Suppose*

$$V = V_1 \perp \cdots \perp V_s$$

and g preserves every summand. If no $g|_{V_j}$ has a fixed vector, then

$$\boxed{\Phi_p(g, V) = \prod_{j=1}^{s} \Phi_p(g|_{V_j}, V_j).}$$

Proof. The half dimensions add, and

$$\det(g - I \mid V) = \prod_j \det(g - I \mid V_j).$$

The quadratic character is multiplicative. □

Corollary 12.5 (Characteristic polynomial certificate)**.** *Let*

$$P_g(X) = \det(XI - g \mid V).$$

Then

$$\boxed{\Phi_p(g, V) = \left(\frac{-1}{p}\right)^r \left(\frac{P_g(1)}{p}\right).}$$

Proof. Since $\dim V$ is even,

$$\det(g - I) = \det(I - g) = P_g(1).$$

□

Proposition 12.6 (Fixed and moving decomposition)**.** *Assume that $g \in \mathrm{Sp}(V)$ has order prime to p. Then*

$$V = \ker(g - I) \perp \mathrm{im}(g - I).$$

The restriction of the symplectic form to $\mathrm{im}(g - I)$ is nondegenerate, and $g - I$ is invertible on that moving summand.

Proof. The order hypothesis makes g semisimple, so

$$V = \ker(g - I) \oplus \mathrm{im}(g - I).$$

If $x \in \ker(g - I)$ and $y = (g - I)z$, then

$$b(x, y) = b(x, gz) - b(x, z) = b(g^{-1}x, z) - b(x, z) = 0.$$

Thus the two summands are orthogonal. Nondegeneracy on each summand follows from nondegeneracy on V. The last statement is immediate from the direct sum decomposition. □

Definition 12.7 (Moving phase)**.** Under the hypotheses of proposition 12.6, define

$$\Phi_p^{\mathrm{mov}}(g, V) = \Phi_p\left(g|_{\mathrm{im}(g-I)}, \mathrm{im}(g - I)\right).$$

Proposition 12.8 (Hyperbolic phase formula)**.** *Let U be a finite dimensional $\mathbb{F}_p$ space, let $a \in \mathrm{GL}(U)$ have no eigenvalue one, and let g act on*

$$H(U) = U \oplus U^*$$

by

$$g = (a, (a^{-1})^*).$$

Then

$$\boxed{\Phi_p(g, H(U)) = \left(\frac{\det a}{p}\right).}$$

Proof. Put $d = \dim U$. We have

$$\det\big((a^{-1})^* - I\big) = (-1)^d (\det a)^{-1} \det(a - I).$$

Hence

$$\det(g - I) = (-1)^d (\det a)^{-1} \det(a - I)^2.$$

Multiplying its quadratic character by the normalization factor $(\frac{-1}{p})^d$ cancels the sign and the square. Since a quadratic character equals its inverse, the result is $(\frac{\det a}{p})$. □

Corollary 12.9 (Phase only bridge test)**.** *Two odd characteristic moving packets have the same normalized phase whenever their moving half dimensions agree and the determinants of $g - I$ lie in the same square class.*

Proof. Apply the definition of Φ_p. □

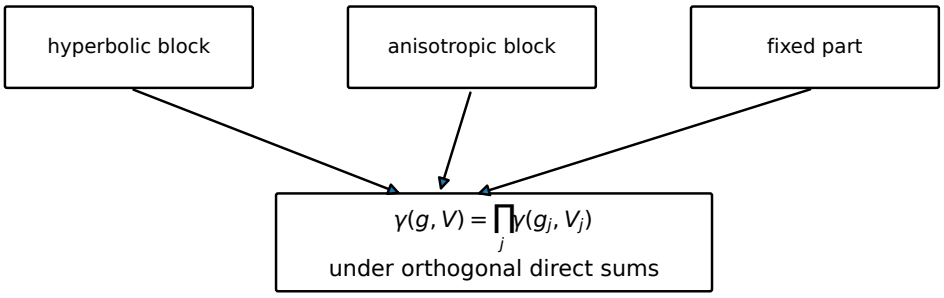
Remark 12.10. When g has fixed vectors, one first separates the fixed part and applies the formula to the moving quotient. Additional normalization data may enter the full trace on the fixed part.

13 Dyadic quadratic refinements

Let (V, b) be a symplectic vector space over $\mathbb{F}_2$. The alternating form alone is not enough to determine the finite phase.

Definition 13.1 (Quadratic refinement)**.** A quadratic refinement of b is a map

$$q : V \to \mathbb{F}_2$$



In odd characteristic the normalized finite Weil phase is controlled blockwise.

Figure 12. Odd finite Weil phases multiply across orthogonal atomic blocks.

satisfying

$$q(x+y)-q(x)-q(y)=b(x,y).$$

Proposition 13.2 (Affine space of refinements)**.** *If one quadratic refinement exists, then the set of all quadratic refinements is an affine space over V^*.*

Proof. The difference of two refinements has zero polar form, so it is additive and hence linear over $\mathbb{F}_2$. Conversely, adding a linear form does not change the polar form. □

Definition 13.3 (Arf invariant)**.** Choose a symplectic basis $(e_i,f_i)_{i=1}^r$. Define

$$\operatorname{Arf}(q)=\sum_{i=1}^{r} q(e_i)q(f_i)\in\mathbb{F}_2.$$

Standard Fact 13.4 (Arf invariance and classification)**.** The value in definition 13.3 is independent of the symplectic basis. Two nondegenerate quadratic refinements of one finite symplectic space are isometric exactly when their Arf invariants agree.

This is the classical classification of nonsingular quadratic forms over $\mathbb{F}_2$ [24].

Theorem 13.5 (Dyadic Gauss sign)**.** *Let $\dim V=2r$, and let q be nondegenerate. Then*

$$\boxed{\sum_{x\in V}(-1)^{q(x)}=(-1)^{\operatorname{Arf}(q)}2^r.}$$

Proof. Orthogonally decompose (V,q) into two dimensional nonsingular planes. On a plane with Arf invariant zero, the four values of $(-1)^q$ sum to 2. On a plane with Arf invariant one, they sum to -2. The total sum is the product of the plane sums. The parity of the number of anisotropic planes is the Arf invariant. □

Corollary 13.6 (Orthogonal dyadic multiplicativity)**.** *For an orthogonal sum*

$$(V,q)=(V_1,q_1)\perp(V_2,q_2),$$

we have

$$\operatorname{Arf}(q)=\operatorname{Arf}(q_1)+\operatorname{Arf}(q_2)$$

and the normalized Gauss sign is the product of the two normalized signs.

Proof. Take symplectic bases of the two summands and concatenate them. The Gauss sum factors as a product over the Cartesian decomposition. □

Proposition 13.7 (The form alone does not determine the phase)**.** *On every nonzero symplectic $\mathbb{F}_2$ space, there are quadratic refinements of both Arf invariants. Their polar forms agree, but their normalized Gauss signs are opposite.*

Proof. On a symplectic plane, the standard hyperbolic refinement has Arf invariant zero. Adding a suitable nonzero linear form produces a refinement with Arf invariant one. Orthogonally adjoining a fixed refinement on the complement gives the two choices in every positive even dimension. The signs differ by theorem 13.5. □

Proposition 13.8 (Translation of quadratic refinements)**.** *For $v \in V$, put*

$$q_v(x) = q(x) + b(v, x).$$

Then

$$\boxed{\operatorname{Arf}(q_v) = \operatorname{Arf}(q) + q(v).}$$

Proof. Using the quadratic identity,

$$q_v(x) = q(x + v) + q(v).$$

Therefore

$$\sum_x (-1)^{q_v(x)} = (-1)^{q(v)} \sum_x (-1)^{q(x)}.$$

Compare the signs in theorem 13.5. □

Theorem 13.9 (Number of dyadic refinements)**.** *Let $\dim V = 2r$. The number of quadratic refinements with Arf invariant zero is*

$$\boxed{2^{r-1}(2^r + 1),}$$

and the number with Arf invariant one is

$$\boxed{2^{r-1}(2^r - 1).}$$

Proof. Every refinement is q_v for a unique $v \in V$, because b identifies V with V^*. By proposition 13.8, the Arf invariant changes according to $q(v)$. Let Z and O be the numbers of zeros and ones of q. Then

$$Z + O = 2^{2r}$$

and theorem 13.5 gives

$$Z - O = (-1)^{\operatorname{Arf}(q)} 2^r.$$

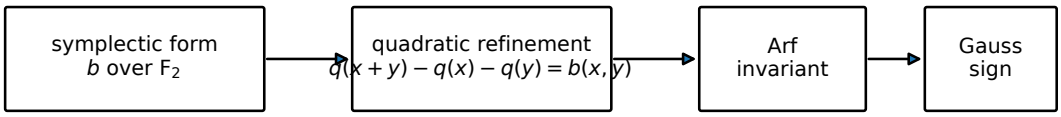
Solving gives the two displayed numbers, with their roles exchanged when the starting refinement has Arf invariant one. Since translation runs through every refinement, the final counts are independent of the starting choice. □

14 Matrix holonomy and finite gluing

Let Γ be a connected finite graph. At every vertex v, place finite dimensional complex vector spaces V_v and W_v of the same dimension. For every oriented edge $e : u \to v$, let

$$A_e : V_u \longrightarrow V_v, \qquad B_e : W_u \longrightarrow W_v$$

be invertible. The reversed edge carries the inverse map.



$$\sum_{x}(-1)^{q(x)} = (-1)^{\mathrm{Arf}(q)}2^{r}$$

The alternating form alone does not determine the dyadic phase.

Figure 13. At residue characteristic two, the quadratic refinement and its Arf invariant carry phase information not contained in the alternating form.

Definition 14.1 (Path transport)**.** For an oriented path

$$\gamma = (v_0 \to v_1 \to \cdots \to v_m),$$

define

$$A_\gamma = A_{v_m v_{m-1}} \cdots A_{v_1 v_0}, \qquad B_\gamma = B_{v_m v_{m-1}} \cdots B_{v_1 v_0}.$$

For a loop based at r, these are the based holonomies.

Theorem 14.2 (Matrix holonomy criterion)**.** *Fix a root r and an isomorphism*

$$U_r : V_r \longrightarrow W_r.$$

The following conditions are equivalent.

(1) There are isomorphisms $U_v : V_v \to W_v$, with the declared U_r, such that

$$U_v A_e = B_e U_u$$

for every edge $e : u \to v$.

(2) For one fundamental cycle basis based at r,

$$U_r A_\gamma = B_\gamma U_r.$$

(3) The same identity holds for every based loop γ.

Proof. If the vertex maps exist, multiply the edge equations around a loop. Every intermediate vertex map cancels, giving condition three. Condition three implies condition two.

Conversely, choose a spanning tree. Starting from U_r, define U_v by transport along the unique tree path from r to v. The edge equations hold on every tree edge by construction. Every non tree edge closes one fundamental cycle. The edge equation on that edge is equivalent to the root holonomy identity for its fundamental cycle. Condition two therefore gives all remaining edge equations. □

Corollary 14.3 (Unbased gauge classification)**.** *There exists some family of vertex isomorphisms if and only if the two holonomy representations*

$$\pi_1(\Gamma, r) \longrightarrow \mathrm{GL}(V_r), \qquad \pi_1(\Gamma, r) \longrightarrow \mathrm{GL}(W_r)$$

are conjugate.

Proof. Apply theorem 14.2 and allow U_r to vary. □

Corollary 14.4 (Scalar and projective gluing)**.** *If every based loop acts by a scalar, finite gluing is equivalent to equality of the scalar holonomies on a fundamental cycle basis. The number of independent tests is*

$$\boxed{|E| - |V| + 1.}$$

Proof. A connected graph has a spanning tree with $|V| - 1$ edges. Every remaining edge gives one fundamental cycle. In the scalar case, conjugacy has no effect. □

Proposition 14.5 (Projective cocycle form)**.** *Suppose the edge maps are defined only up to scalar, in the setting where all based holonomies of both systems are scalar. Choose representatives. The ratio of the two scalar holonomies on each loop is unchanged by vertex rescaling. It defines a character of the first homology group when the scalar transport is abelian.*

Proof. A vertex rescaling appears once with positive exponent and once with negative exponent around a closed path, so it cancels. Multiplication of loops multiplies the scalar ratios. In the abelian case, the ratio factors through the abelianization of the fundamental group, which is the first homology group. □

Proposition 14.6 (Exact edge residual from cycle defect)**.** *Fix a spanning tree and construct the vertex maps by tree transport from U_r. Let $e : u \to v$ be a non tree edge. Let p_u and p_v be the tree paths from the root to u and v. Define the fundamental cycle defect*

$$D_e = U_r A_{p_v}^{-1} A_e A_{p_u} - B_{p_v}^{-1} B_e B_{p_u} U_r.$$

Then the edge residual is exactly

$$\boxed{U_v A_e - B_e U_u = B_{p_v} D_e A_{p_u}^{-1}.}$$

Proof. Substitute

$$U_u = B_{p_u} U_r A_{p_u}^{-1}, \qquad U_v = B_{p_v} U_r A_{p_v}^{-1}$$

into the left side and factor B_{p_v} on the left and $A_{p_u}^{-1}$ on the right. □

Corollary 14.7 (Quantitative finite gluing)**.** *With the notation of proposition 14.6,*

$$\|U_v A_e - B_e U_u\| \le \left\|B_{p_v}\right\| \|D_e\| \left\|A_{p_u}^{-1}\right\|.$$

Thus small fundamental cycle defects give small non tree edge residuals whenever the tree transports are uniformly conditioned.

Proof. Take operator norms in the exact identity. □

Remark 14.8. The theorem is an exact finite gluing result. It assumes that the packet spaces and edge transports have already been constructed on both sides.

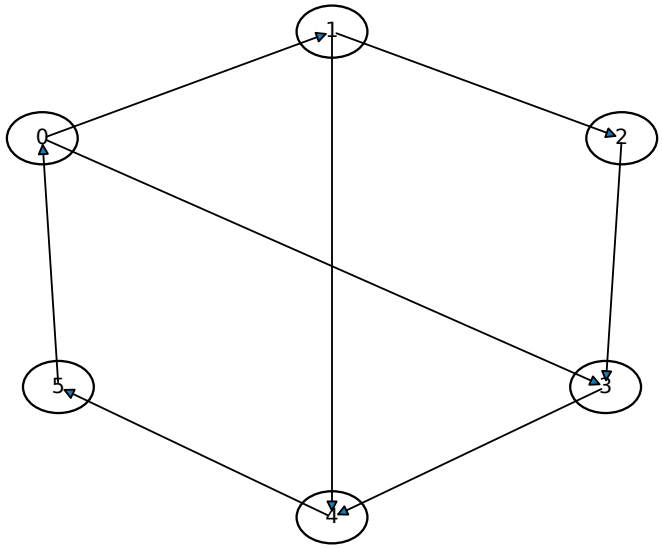


Figure 14. A spanning tree fixes the vertex gauges. The non tree edges test the remaining cycle holonomies.

15 Breakwise assembly

Wild inertia carries an upper ramification filtration. In favorable settings, a wild atom separates into finitely many break pieces. We state the finite algebra needed once such pieces are intrinsic.

Definition 15.1 (Orthogonal break decomposition)**.** An orthogonal break decomposition is a symplectic direct sum

$$V = \operatorname*{\perp}_{s \in S} V_s$$

that is preserved by the declared tame stabilizer. Each V_s is the finite symplectic packet assigned to one break.

Proposition 15.2 (Heisenberg tensor factorization)**.** *Fix compatible central characters. The irreducible Heisenberg representation of an orthogonal sum is the tensor product of the irreducible Heisenberg representations of its summands:*

$$\boxed{\omega_V \cong \bigotimes_{s \in S} \omega_{V_s}.}$$

Proof. The Heisenberg group of an orthogonal sum is the central product of the Heisenberg groups of the summands. The tensor product representation has the declared central character and has dimension

$$\prod_s \sqrt{|V_s|} = \sqrt{|V|}.$$

It is irreducible by the finite Stone von Neumann theorem, so it is the required representation. □

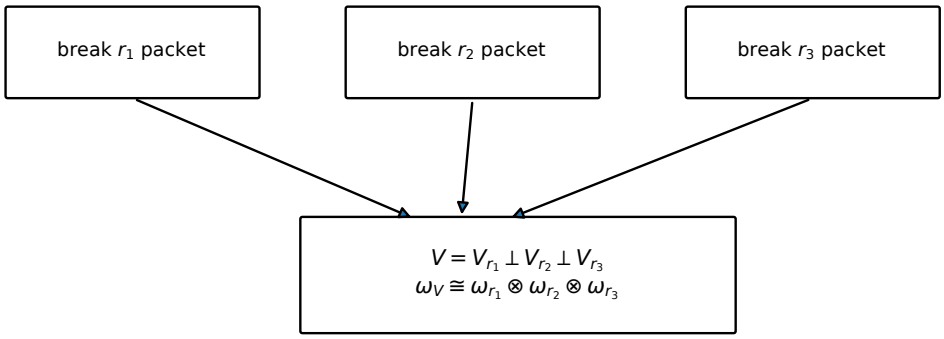


Figure 15. Orthogonal break packets produce tensor factors of the finite Heisenberg representation.

Corollary 15.3 (Breakwise phase product)**.** *In odd residue characteristic, let $g \in \mathrm{Sp}(V)$ preserve every break packet and suppose its restriction to $M = \mathrm{im}(g - I)$ is fixed point free. Put $g_s = g|_{V_s}$ and $M_s = \mathrm{im}(g_s - I)$. The moving restrictions are symplectic, and*

$$\boxed{\Phi_p(g|_M, M) = \prod_{s \in S} \Phi_p(g_s|_{M_s}, M_s).}$$

In residue characteristic two, the analogous normalized Gauss sign is the product of the signs determined by the breakwise quadratic refinements.

Proof. The invariant orthogonal decomposition gives $M = \perp_{s \in S} M_s$. Symplectic invariance gives $M^\perp = \ker(g - I)$. Fixed point freeness on M therefore makes the restricted form nondegenerate on M and on each M_s. Each restriction of $g_s - I$ to M_s is invertible, so proposition 12.4 applies to the moving spaces. A zero moving space contributes one. In characteristic two, apply corollary 13.6. □

Theorem 15.4 (Breakwise restricted bridge)**.** *Suppose every break has a restricted bridge datum, the symplectic isometries are compatible with the stabilizer action, and the scalar phases agree on a generating set of the stabilizer. Then the tensor product of the breakwise Heisenberg intertwiners gives a bridge for the full finite envelope. Any remaining gluing obstruction is exactly the matrix holonomy obstruction of theorem 14.2.*

Proof. Tensor the breakwise Heisenberg intertwiners from theorem 11.2. Phase agreement makes the tensor map equivariant on the chosen stabilizer generators. The breakwise direct sum and tensor factorization make the local comparisons compatible. If the packets are arranged over a transport graph, the only remaining condition is consistency of the local maps around cycles, which is exactly the holonomy criterion. □

16 Local factor tests and the remaining bridge

Local class field theory identifies one dimensional Weil characters with characters of $K^\times$ [27, 29]. The tame induction theorem extends this explicit picture to higher dimensional tame atoms. The local polynomial and conductor calculations in propositions 5.4 and 5.8 therefore give explicit finite

invariants for the declared tame packets. Finite Gauss sums give the residue field phase. The full local factor, however, also includes ramification depth, Haar measure normalization, the additive character conductor, and wild interaction data.

Proposition 16.1 (Finite tame comparison package)**.** *For a tame induced atom, the following data are determined by its packet coordinates:*

(1) the dimension and inertia orbit
(2) the Frobenius cycle polynomial
(3) the Artin conductor
(4) the local polynomial
(5) the tensor refinement into interaction packets
(6) the finite Gauss sum of each declared residue character, with unit-modulus phase normalization when that character is nontrivial.

Proof. The dimension and orbit follow from theorem 3.2. The cycle polynomial is proposition 4.4. For rank greater than one, the conductor is proposition 5.8. In rank one, the Swan conductor is zero, and the Artin conductor is zero for trivial inertia and one otherwise, by the formula $\mathrm{Art} = \dim V - \dim V^{I_K} + \mathrm{Sw}$. The local polynomial is proposition 5.4. Tensor refinement is theorem 5.1. The Gauss sum is its defining finite sum. For a nontrivial multiplicative character its unit phase is definition 6.2. For the trivial character and a nontrivial additive character ψ, the sum is $G(1, \psi) = -1$, because the sum of ψ over the whole field is zero. □

16.1 Arithmetic realization and the conditional comparison

Fix this chapter's reciprocity, additive-character, local-factor, and phase normalizations. Let $\mathcal{A}$ be the declared class of irreducible wild Weil atoms under consideration. For $\rho \in \mathcal{A}$, the supplied assembled finite datum d_ρ consists of the following inputs, historically denoted H1–H4:

(1) an intrinsic finite break decomposition on the Galois side;
(2) the declared equivariant symplectic bridges to the supplied breakwise type-side quotients;
(3) compatible odd or dyadic central-extension and phase data on every break, including every prescribed finite marking;
(4) a compatible transport system with vanishing matrix holonomy obstruction.

These are finite comparison inputs. Supplying separate breakwise target types does not itself supply one common arithmetic candidate. If a common candidate is already supplied, that is a stronger restricted hypothesis.

An *arithmetic realization witness* $\mathcal{R}(d_\rho, \pi)$ consists of actual arithmetic inducing data for an admissible supercuspidal π, together with explicit comparison and recovery maps carrying the prescribed assembled finite packet. The maps preserve its declared stabilizer action, all character values explicitly included in the supplied datum, central-extension and phase data, and compatible transports. Recovery returns that same π, with its fixed central character and noncompact completion. An enlarged actual inducing carrier is permitted. The whole packet is not required to be the original maximal-simple-type quotient. Any stronger native-marking or numerical-layer requirement is part of the declared datum and must be verified separately.

Conjecture 66 (Arithmetic packet realization)**.** For every $\rho \in \mathcal{A}$ with its supplied H1–H4 assembled finite datum d_ρ, there exists an actual admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(K)$ and an arithmetic realization witness $\mathcal{R}(d_\rho, \pi)$.

This is the realization problem isolated from historical Theorem 16.2 and original proof obligation 66 (row 95 of the historical all-records register). It asks for one witness for each admitted source, rather than for success of every proposed host or every noncompact completion. No proof of the unrestricted statement is claimed.

Proposition 16.3 (Comparison of one realized candidate)**.** *Let $\rho \in \mathcal{A}$, and let π be one actual admissible supercuspidal representation of $\mathrm{GL}_{\dim \rho}(K)$ with $\mathcal{R}(d_\rho, \pi)$. Suppose H5 holds for this same π: its required pair local factors equal those prescribed for ρ on the stipulated characterizing family of twists, with the fixed normalizations. Assume the stated local-factor characterization applies to that family and class. Then π is isomorphic to the representation assigned to ρ by the established local Langlands correspondence.*

Proof. H5 identifies the tested local factors for this actual candidate. The applicable local-factor characterization identifies its isomorphism class with the established correspondence [18]. The realization witness ensures that it is this same candidate which carries the prescribed assembled packet. No existence conclusion is inferred from the finite gluing conditions alone. □

Remark 16.4. H5 remains a conditional premise about the same fixed candidate. It is not a formal consequence of the finite symplectic bridge, conductor profiles, or a choice of noncompact root. This proposition adds no unconditional pair-factor obligation to the construction conjecture.

16.2 Exact open boundary

The following issues delimit the unrestricted program. The first four construct and normalize the finite inputs for broader source classes. Conjecture 66 asks for arithmetic realization once H1–H4 are supplied. The fifth is the independent route to replacing the retained conditional H5 input.

(1) Construct a choice free finite symplectic envelope from an arbitrary irreducible wild inertia atom, with its ramification filtration and tame stabilizer action.
(2) Prove a canonical equivariant symplectic isometry from that envelope to the type side quotient J^1_θ / H^1_θ.
(3) Recover the canonical beta extension phase in every break, including the required dyadic quadratic refinement.
(4) Prove matrix holonomy compatibility for the resulting packet transport system.
(5) Compare the full family of pair L, epsilon, and gamma factors without invoking the established correspondence.

The first item is a construction problem. The second is a comparison problem. The third is a normalization problem. The fourth is a descent problem. The fifth concerns proving the local factor premise needed for identification. Conjecture 66 does not assert this fifth step. None is replaced by a larger finite computation.

17 Examples and finite certificates

The examples in this section illustrate the exact finite formulas. They are not used as proofs of the general theorems.

Example 17.1 (A degree four tame orbit)**.** Over $\mathbb{F}_2$, the exponent orbit

$$1 \longmapsto 2 \longmapsto 4 \longmapsto 8 \longmapsto 1$$

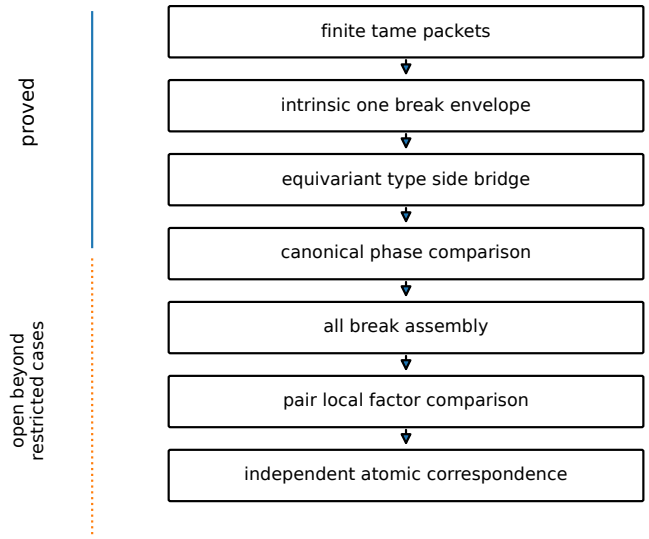


Figure 16. The proved finite layers and the remaining independent local correspondence steps.

has length four. It therefore gives a tame atom induced from the unramified extension of degree four. The Frobenius transfer has one four cycle, and the number of regular degree four orbits is three by proposition 3.6.

Example 17.2 (Tensor packets of lengths four and six)**.** For $n = 4$ and $m = 6$,

$$\gcd(4,6) = 2, \qquad \operatorname{lcm}(4,6) = 12.$$

The tensor product splits into two interaction packets of length twelve. Their total dimension is $2 \cdot 12 = 24$, equal to the product $4 \cdot 6$.

Example 17.3 (Rank one Heisenberg group)**.** Take $V = \mathbb{F}_3^2$. The Heisenberg group has order $3^3 = 27$, and its irreducible representation with a fixed nontrivial central character has dimension three. The trace vanishes on every noncentral element. Averaging the translation subgroup gives a rank one projector.

Example 17.4 (Nondegenerate twisted algebra)**.** Let $A = \mathbb{F}_p^2$ with its standard alternating form. A cocycle with that commutator form gives

$$\mathbb{C}^c[A] \cong M_p(\mathbb{C}).$$

The unique irreducible projective representation has dimension p.

Example 17.5 (Dyadic phase ambiguity)**.** On a symplectic plane over $\mathbb{F}_2$, choose refinements q_0 and q_1 with Arf invariants zero and one. Their polar forms are the same, while

$$\sum_x (-1)^{q_0(x)} = 2, \qquad \sum_x (-1)^{q_1(x)} = -2.$$

Thus the alternating packet alone cannot select the dyadic sign.

Example 17.6 (One cycle gluing obstruction)**.** Let a connected graph have one non tree edge. Tree transport determines every vertex map from one root map. The final edge equation holds exactly when the two holonomies around the unique fundamental cycle intertwine. Changing the scalar on that edge changes the cycle holonomy and produces a genuine obstruction.

18 Scope of the finite reconstruction

This chapter proves the following statements under their displayed hypotheses.

(1) Tame irreducible Weil atoms are transitive Frobenius packets induced from unramified extensions.
(2) Regular tame packets have the stated Möbius count.
(3) Finite cuspidal blocks have exact primitive central projectors.
(4) Tame tensor products satisfy the greatest common divisor and least common multiple packet formula.
(5) Tame local polynomials, conductors, and finite Gauss phases are explicit.
(6) Nondegenerate scalar projective commutators give full matrix twisted group algebras and unique finite Heisenberg representations.
(7) A declared one break Clifford datum has a choice independent radical free symplectic envelope.
(8) An equivariant symplectic isometry gives a Heisenberg bridge unique up to scalar.
(9) One generator phase removes the scalar discrepancy for a cyclic stabilizer.
(10) Odd finite Weil phases and dyadic Arf signs are multiplicative across orthogonal break packets.
(11) Matrix holonomy on a fundamental cycle basis is the exact finite gluing obstruction.

This chapter does not independently prove the local Langlands correspondence, simple character theory, the canonical beta extension, or universal wild companion existence. It does not construct the Galois to type symplectic isometry for an arbitrary wild atom. It does not compare all pair local factors in the wild case.

19 Restricted arithmetic realization theorems

The constructions provide actual arithmetic candidates and exact finite packet recovery in substantial restricted families. They do not establish conjecture 66 for every admitted source. Their detailed ordinary proofs are typeset in the appendices, and the original reviewed Markdown files accompany the volume. In these appendices F denotes the base local field, while $K = F[\beta]$ denotes a parameter field. These are local notation conventions. All references to "claim 66" in the retained notes mean the general endpoint now isolated as Conjecture 66.

19.1 The odd-prime trace-zero construction

Theorem 19.1 (Restricted odd-prime arithmetic realization)**.** *Let p be an odd prime, $F = \mathbb{F}_p((t))$, fix $d \geq 1$, and let*

$$0 < h_1 < \cdots < h_d, \qquad p \nmid h_i, \qquad q_i = (p+1)h_i, \qquad N = p^d.$$

There is an explicit maximal simple stratum in $M_N(F)$ with totally ramified full-degree parameter field, a literal defining sequence of degrees $p, p^2, \ldots, p^d$, and levels $n_j = p^{d-j}q_{d-j+1}$. Every native parameter and consecutive correction has trace zero over its recursive base field. The stratum admits a full simple character trivial on U_F^1. Its fixed canonical beta extension can be completed to an actual irreducible admissible supercuspidal π with any prescribed smooth central quasicharacter trivial on U_F^1, when its completion choices are free or already compatible.

Assume actually supplied labeled full-character extensions with whole two-dimensional symplectic quotients whose balanced complex-scalar central product has d planes, one vertex, and trivial stabilizer. Then the same π admits a full scalar-pushout realization of that packet and exact recovery of its unchanged inducing representation.

The prescribed central character must lie in the covered tame class. No literal embedding of the old matrix groups, canonical native marking, or prescribed numerical host-depth placement is inferred.

Proof. The explicit Artin–Schreier recursion and its normalized filtered lift construct the maximal parameter and literal defining sequence. See appendix B. Its trace transport identity makes every native trace zero, rather than relying on amplification by p. Native-rank full-character construction, full beta-determinant transfer, trace-zero correction, and maximal-field character extension give the scalar triviality and central completion. See appendix C. The actual leading symplectic layer has dimension $(p-1)p^{d-1} \geq 2d$ and survives every later approximation (appendix D). A labeled symplectic embedding gives a center-fixing full scalar-pushout isomorphism. Finite Stone–von Neumann decomposition separates the source factor from its orthogonal complement. Maximal-isotropic averaging on the complementary Heisenberg factor selects a one-dimensional representation line and hence a source-sized sector. The retained induction/idempotent recovery returns the same Λ and π, as proved in appendix A. □

For $p > 3$, a native minimal rank-p type quotient has dimension $p-1$, so it is not automatically a plane. The theorem requires genuinely supplied whole-plane data. Independence of source tensor factors is also an input. An arbitrary single-factor wild determinant need not be tame.

Under additional actual-inertia hypotheses, genuine scalar phases on the entire source carrier are realized by actual central matrices of a finite cover and descend to the stipulated operators. A direct matrix homomorphism from the stabilizer itself need not exist. The exact statement and descent proof are in appendix E.

19.2 The weighted dyadic construction and enlarged carriers

Theorem 19.2 (Restricted dyadic marked assembly realization)**.** *Let $F = \mathbb{F}_2((t))$, fix an additive character of conductor $t\mathcal{O}_F$, fix $d \geq 1$, and let $h_1 < \cdots < h_d$ be positive odd integers. Put $q_i = 3h_i$, $N = 2^d$, and $n_j = 2^{d-j}q_{d-j+1}$. The explicit recursive matrix generates a separable totally ramified degree-N maximal parameter field, normalizes the complete lattice chain, and has the literal defining sequence of degrees $2, 4, \ldots, 2^d$. It admits a full simple character trivial on U_F^1 and an actual supercuspidal completion with any prescribed central quasicharacter trivial on U_F^1, when the noncompact completion is free or already compatible.*

For actually supplied full marked whole-plane scalar-pushout data with one vertex, trivial stabilizer, and prescribed source central character trivial on U_F^1, an arithmetic assembly carrier exists at every length and recovers the same candidate: inside its original type for $d \geq 3$, and in explicit enlarged inducing carriers for $d = 1, 2$. Every supplied full character value and transported whole finite subgroup, lift, square, and phase marking is retained at the represented marking level. Any central-character kernel already factored out remains recorded separately. Scalar pushout does not recover its invisible elements. This does not identify the transported marking with an independently prescribed canonical native marking.

Proof. The Artin–Schreier filtered lift has native trace one. Consequently each native correction δ satisfies

$$v_F(\operatorname{tr}\delta) + s \geq 1 \quad (s = 1),$$

which makes its additive scalar correction trivial on U_F^1. The actual field and defining-sequence proof is appendix H. Full beta-determinant transfer and character extension at each new maximal native field prove the full-character claim, including the odd-critical endpoint $2(\lfloor r/2 \rfloor + 1) - r = 1$. See appendix I. The exact original quotient has dimension $2^d - 2$, which accommodates the required $2d$ source dimensions precisely for $d \geq 3$ (appendix J).

At rank two, with $q = 2m+1$, enlarge $J = U_K^1 U_A^{m+1}$ to $P = U_K^1 U_A^m$. The actual character ratio $\lambda = \theta^g/\theta$ for $g \notin J$ is nontrivial and sign-valued. With $H_0 = \ker\lambda$, the quotient is $P/H_0 \simeq \mathbb{F}_2^2$, and

$\mathrm{Ind}_J^P \theta$ is an irreducible two-dimensional Heisenberg carrier (appendix L). At rank four, the actual outer quadratic centralizer gives $P = JU_B^m$, $[P : J] = 2$, and $P/H_0 \simeq \mathbb{F}_2^4$, with carrier $\mathrm{Ind}_J^P \eta$ of dimension four (appendix M). The finite scalar center can be larger than μ_2. No literal extraspecial quotient is claimed.

At both ranks $K^\times$ normalizes P. Thus $\mathcal{N} = K^\times J$ is normal of index two in $\mathcal{M} = K^\times P$, and

$$\Sigma = \mathrm{Ind}_{\mathcal{N}}^{\mathcal{M}} \Lambda, \qquad \pi \simeq \mathrm{cInd}_{\mathcal{M}}^{\mathrm{GL}_N(F)} \Sigma.$$

Induction in stages preserves the full original Λ, its central character, and its noncompact root. See appendices N and O. The full marked scalar-pushout comparison applies to these actual enlarged pairs as well as to the original higher-rank types (appendix K). The full synthesis and scope are in appendix G. □

The original low-rank maximal-type capacity obstruction remains true. The new pair is a common arithmetic assembly carrier. It is not renamed as the original maximal-simple-character pair. For an independently fixed native quadratic marking, a preassigned embedding must satisfy $q_W(f(v)) = q_{\mathrm{src}}(v)$. At equal capacity, a free embedding requires equal Arf invariant. Spare capacity for $d \geq 4$ admits either source Arf type when the embedding is free. Retaining only the central $\{\pm 1\}$ would erase the distinction between whole marked finite extensions such as D_8 and Q_8.

19.3 Scope, numerical invariants, and evidence

For the same candidates the constructions give

	p odd	$p = 2$
$\dim_{\mathbb{F}_p}(J^1/H^1)$	$p^d - 1$	$2^d - 2$
$\mathrm{Swan}(\pi)$	$p^{d-1}(p+1)h_d$	$3 \cdot 2^{d-1} h_d$

and torsion number one. Here Swan denotes the Swan conductor, and a below denotes the ordinary pair conductor. An epsilon exponent for a specified additive character is a different quantity. The separate progress paper, Sections 22–24, reconciles these conventions and the actual-order hypotheses. Their self-pair conductor formulas are

$$a(\pi^\vee \times \pi) = p^{2d} - 1 + \sum_{i=1}^{d} (p^{2i} - p^{2i-2}) h_i \quad (p \text{ odd}),$$

$$a(\pi^\vee \times \pi) = N^2 - 1 + \sum_{i=1}^{d} 3 \cdot 4^{i-1} h_i \qquad (p = 2).$$

These identities do not prove H5, identify an endo-class, or construct a canonical source-to-host marking. At rank eight, the retained skipped-degree host has the same displayed numerical profiles as the full tower and a smaller Heisenberg quotient. The explicit construction, rather than numerical matching alone, supplies the arithmetic object.

The proofs use the stated Bushnell–Kutzko simple-type theorems [11] and the retained conductor inputs. Exact source locations and hypothesis checks are recorded in the component proofs and accompanying audits. The finite rank-nine and rank-eight runs check particular matrices, traces, normalized lifts, and filtration grades. They are bounded checks, not arbitrary-length proofs. No new Lean build of these arithmetic constructions is claimed.

Nontrivial prescribed symplectic stabilizer actions, independently fixed canonical native markings, arbitrary source/transport diagrams, and any required numerical layer placement remain outside the restricted results. The current totally ramified host cannot implement nontrivial prime-to-p symplectic actions by the stronger character-preserving inner-normalizer mechanism. This obstruction is stated

precisely in appendix F. That stronger mechanism is not silently imposed on the original conjecture. H5 remains conditional for the same fixed candidate.

The accompanying archive retains the mathematical reviews, computational evidence, and source records underlying the displayed scope.

Detailed ordinary proofs and construction records

The appendices give the selected construction proofs with their local notation, hypotheses and scope. The synthesis appendices combine the results. Other notes develop narrower lemmas, alternative routes and failed implications. The classical application of arithmetic induction in the leading-capacity argument is not an input to the direct constructor. Negative capacity results concern the original maximal-type carrier. Low-rank enlargement of the assembly carrier does not contradict them. These are ordinary mathematical proofs and do not prove Conjecture 66. Source notes are listed at the back.

A Odd-prime synthesis and complete scope

Source record: `Paper_XI_Base_Realization165.zip`.

This is a synthesis of the independently reviewed proofs in this continuation. The result is an ordinary mathematical proof, using the stated classical simple-type theorems. It is not a Lean certificate. Universal claim 66 remains open. Original H5 remains the original conditional premise about the same fixed candidate. No unconditional pair-factor comparison is asserted here.

A.1 1. The arithmetic construction now proved

Let p be odd, $F = \mathbb{F}_p((t))$, and

$$0 < h_1 < \cdots < h_d, \qquad p \nmid h_i, \qquad q_i = (p+1)h_i, \qquad N = p^d.$$

For every such tuple there is an explicit matrix $\beta \in M_N(F)$ whose field $K = F[\beta]$ is totally ramified of degree N, normalizes the complete lattice chain, and gives a maximal simple stratum. It has a literal defining sequence of degrees $p, p^2, \ldots, p^d$, with leading level n_1 and successive critical levels $n_2, \ldots, n_d$ given by

$$n_j = p^{d-j} q_{d-j+1} \quad (1 \le j \le d).$$

Every native parameter in the construction, every amplified approximation, and every consecutive correction has matrix trace zero over its base field.

For this actual stratum there exists a full simple character θ with $\theta|_{U_F^1} = 1$. Retaining its canonical beta extension, the residue character and noncompact extension can be selected so that the resulting irreducible admissible supercuspidal π has any prescribed smooth central quasicharacter ω trivial on U_F^1. These completion choices must be free or already compatible. Incompatible immutably supplied choices are not replaced.

The same candidate has

$$\begin{aligned} \operatorname{rank}(\pi) &= N, & t(\pi) &= 1, \\ \operatorname{Sw}(\pi) &= p^{d-1}(p+1)h_d, & \dim_{\mathbb{F}_p}(J^1/H^1) &= N-1, \\ \dim \eta &= p^{(N-1)/2}, & & \\ a(\pi^\vee \times \pi) &= p^{2d} - 1 + \sum_{i=1}^{d} (p^{2i} - p^{2i-2}) h_i. & & \end{aligned}$$

These numerical identities do not identify a Weil representation, an endo-class, or the pair epsilon factors required by H5.

A.1.1 Construction and proof

At a recursive base (L, t_L), introduce

$$E = L(\alpha), \qquad \alpha^p - \alpha = t_L^{-q_d}.$$

This is a separable totally ramified extension of degree p, with $v_E(\alpha) = -q_d$ and $\operatorname{Tr}_{E/L} \alpha = 0$. Choose $1 \le b < p$ with $q_d b \equiv -1 \pmod p$, set $a = (q_d b + 1)/p$, and put $z = t_L^a \alpha^b$. Then z is an explicit uniformizer. Write its Eisenstein polynomial as $f(X) = X^p + \cdots + a_0$.

Let P_0 extract the constant coefficient in the basis $1, z, \ldots, z^{p-1}$. The normalized corestriction $s = \kappa \operatorname{pr}$, where $\kappa = -f'(z)/z^{p-1}$, satisfies

$$u_0 = s(P_0) = a_0/z^p \in \mathcal{O}_E^\times, \qquad \operatorname{Lift}_E(C) = u_0^{-1} C P_0, \qquad s(\operatorname{Lift}_E(C)) = C.$$

The lift preserves the exact filtration valuation. Its crucial additional identity is

$$\mathrm{tr}_L(\mathrm{Lift}_E(C)) = \mathrm{Tr}_{E/L}\big(\kappa^{-1}\,\mathrm{tr}_E(C)\big).$$

Define

$$B(L, t_L; h_1, \dots, h_d) = \alpha I_{p^{d-1}} + \mathrm{Lift}_E\big(B(E, z; h_1, \dots, h_{d-1})\big),$$

with the degree-p multiplication matrix as the base case. Induction proves trace zero at every native rank. This is stronger than the automatic zero trace obtained by multiplying any matrix multiplicity by p in characteristic p.

For the inductive field step $d \geq 2$, put $n = p^{d-1}q_d$ and $R = p^{d-2}q_{d-1}$. The derived matrix is the recursively constructed maximal-field parameter, with $k_0 = -q_1$. The inequalities $n > pR > 2R$ imply $-2R > -n - q_1$. Bushnell–Kutzko Remark 2.2.7 therefore applies to the actual displayed matrix. It supplies its maximal field, normalization, and critical exponent. Valuation preservation and literal truncation of the recursion supply the whole defining sequence. The full details, quotient count, and conductor dependencies are in `tower/TRACE_ZERO_AS_TOWER166.md` and `tower/RECURSIVE_TOWER166.md`.

The full-character argument uses a specific order of operations. First construct a full character at native rank p^j, trivial on U_F^1. Transfer its full beta-determinant unit character to multiplicity p, using BK 3.6.13(ii) and 3.6.14. On a scalar $u \in U_F^1$, this gives $\theta_j(u^p) = 1$. Next add the actual trace-zero correction, using BK 3.3.18. It remains trivial on the scalar intersection of its domain. Finally extend at the next **maximal native parameter field**: the forced deep character and the trivial character of U_F^1 agree on their intersection, so they extend through a finite abelian unit quotient. BK 3.2.5 constructs the full simple character.

This procedure never substitutes an unjustified extension through a nonmaximal centralizer. It proves full-character compatibility, including shallow units, rather than merely cancellation of a leading trace. The residue factor fixes ω on $\mathbb{F}_p^\times$, and an N-th-root choice for the noncompact generator fixes $\omega(t)$. Every choice is recorded before H5. See `central/TRACE_ZERO_CHARACTER_INDUCTION166.md` for the complete induction.

A.2 2. The finite packet that this same candidate realizes

Assume the one-vertex scalar-pushout contract: there are d labeled, actually supplied full character extensions with whole symplectic quotients V_i of dimension two over $\mathbb{F}_p$. Their balanced $\mathbb{C}^\times$-central product has quotient $V_{\mathrm{src}} = \bigoplus_i V_i$ and Heisenberg carrier dimension $p^d = N$. Initially take the declared stabilizer $C = \{1\}$.

For $p = 3$, this applies directly to the retained cubic family with whole rank-three break-type quotients. For $p > 3$, it applies only when the supplied plane data genuinely satisfy the stated original bridge contract. A native minimal rank-p type quotient has dimension $p - 1$, so it must not automatically be called a plane.

The actual candidate from Section 1 has a nondegenerate leading subspace

$$S \subset J^1/H^1, \qquad \dim_{\mathbb{F}_p} S = (p-1)p^{d-1} \geq 2d.$$

Here S is the image of the concrete congruence subgroup at half the leading even level. BK's high-depth equality preserves this subgroup and its commutator pairing through every later approximation. See `capacity/LEADING_LAYER_CAPACITY166.md`, Section 1. This part of that note applies to the direct constructor and does not require its separate classical application of arithmetic induction.

Choose a labeled symplectic embedding $V_{\mathrm{src}} \to S$. For each original full character form

$$\mathcal{P}_i = (\mathbb{C}^\times \times J_i^1)/\{(\theta_i(h)^{-1}, h) : h \in H_i^1\}.$$

The balanced product and the corresponding host restriction have the same alternating commutator. The center-fixing extension isomorphism, obtained with matched Darboux lifts and complex roots, therefore preserves every supplied value $\theta_i(h)$, including the shallow values.

Write $W = V_{\mathrm{src}} \perp V_{\mathrm{src}}^{\perp}$. Finite Stone–von Neumann gives $\eta_W \simeq \eta_{\mathrm{src}} \otimes \eta_{\perp}$. Choose an isotropic eigenspace giving a line in the complementary factor. Its actual projector in the Heisenberg image algebra e has rank N. The retained induction/idempotent theorem recovers the **same** inducing representation Λ from $e\Lambda$, and then the same π by compact induction. No candidate is changed at recovery.

This attachment preserves the full source characters as scalar-pushout data. It does not identify the old matrix groups with literal subgroups of the new type, retain the old lower summand conjugacy classes, or place each source label at its old numerical host depth. Those stronger requirements have not been proved.

When the retained independent extraspecial tensor source has $d \geq 2$, its full determinant is trivial on wild inertia: each factor determinant has order dividing p, and tensor determinants raise it to p^{d-1}. Thus its central character belongs to the class in Section 1. For $d = 1$, wild triviality of the source determinant must be checked separately. Independence of the tensor factors is an input, not a consequence of distinct h_i.

A.3 3. A proved extension to genuine scalar inertia phases

There is also a positive result beyond $C = \{1\}$. Suppose the finite prime-to-p stabilizer acts trivially on the source scalar extension, and its prescribed honest operators on the **entire** rank-N Weil carrier are $\chi(c)I_N$, each lying in the actual inertia image. Assume the candidate's central character is the determinant of that same source under the fixed rank-one reciprocity convention.

Then a finite cover $\widetilde{C} \twoheadrightarrow C$ acts by actual central matrices in the unchanged candidate. Its represented operators descend to exactly χ on C. Hence the attachment and recovery above are equivariant for that action. A direct matrix homomorphism from C itself is not asserted and can fail.

Indeed, for residue cardinality q, let $B = \omega(\mathrm{Teich}(k^{\times}))$. The actual inertia hypothesis gives $\chi(C)^N \subset B$. Since N is a power of p and $\chi(C)$ has order prime to p, this implies $\chi(C) \subset B$. Choose k_0 inverse to N modulo $q-1$, and set

$$\delta(u) = \omega(\mathrm{Teich}(u))^{k_0}, \qquad \widetilde{C} = \{(c,u) : \chi(c) = \delta(u)\}.$$

The matrix $\mathrm{Teich}(u^{k_0})I_N$ acts by $\chi(c)$. The kernel acts trivially on the entire sector induction and ambient compact induction, because these matrices are central. This preserves the fixed Λ, its canonical beta-extension factor, and all completion choices. It does not assert that the canonical beta extension alone has scalar character χ.

The full theorem and the nonsplitting example are in `stabilizer/SCALAR_NATIVE_ATTACHMENT166.md`, with an independent review in `audit/SCALAR_NATIVE_REVIEW166.md`. An arbitrary abstract phase or a phase on a proper multiplicity space does not satisfy its hypotheses merely because it is scalar.

A.4 4. What was repaired, and what remains

The old fixed-field central argument genuinely failed. At $p = 3$, $(h_1, h_2) = (1, 5)$, its naturally transferred full character takes the value $\zeta_3 \neq 1$ on $1 + t^2$, although the desired wild center is trivial. A shallow character choice cannot repair a forced value. The two-stage changing-field repair is retained in `central/CENTRAL_REPAIR166.md`. The trace-zero tower above resolves this central issue uniformly in every length.

The construction does not use an inverse established correspondence or the classical arithmetic-induction/descent candidate to select π. It does use the named BK construction and type theorems, the retained type/formal-degree conductor formulas, and exact sector recovery. The capacity note separately retains an alternative consequence for a candidate supplied by the substantial classical arithmetic constructor. That is a different route and is not an input to Section 1.

The unresolved universal obligations are precise:

1. **Nontrivial prescribed symplectic stabilizer actions.** For this totally ramified full-degree host, the stronger inner-normalizer interpretation forces prime-to-p actions on the quotient to be trivial. It therefore cannot implement a genuinely nontrivial supplied symplectic action. The raw original contract does not automatically require that stronger interpretation. It still needs an actual valid arithmetic comparison, not relabeling of external operators.
2. **Arbitrary admissible source packets and transport diagrams.** The proved attachment has the whole-plane, one-vertex, finite scalar-pushout hypotheses above. It does not establish all source classes, arbitrary graphs, or prescribed numerical layer placement.
3. **The dyadic construction with all markings.** A separate field-level recursion and some capacity statements exist at $p = 2$. The odd-prime trace-zero/full-character theorem here does not settle that case.

Original H5 stays exactly where it was: if supplied for this fixed candidate and the stipulated tests, the retained conditional endpoint applies. No claim that the displayed profiles imply H5, or that one of the noncompact root choices must satisfy H5, is made. The unresolved construction issues above are not renamed epsilon-factor obligations.

A.5 5. Evidence and reproducibility

The arbitrary-length theorem is proved symbolically and independently reviewed. The exact $p = 3, d = 2, (h_1, h_2) = (1, 2)$ computation in `tower/as_rank9check/` verifies the displayed matrix, trace identities, corestriction, and levels. It is a finite check, not the universal proof. The separate rank-27 run in `tower/run_rank27/` checks the earlier non-trace-zero recursion and must not be described as an AS central completion example. No Lean build was run.

`STATUS166.json` records the closed and open obligations. The component audits identify the exact reviewed files and hashes. The dependency file pins prior checkpoints and the authorized primary-source copy without redistributing that publication.

B The trace-zero Artin–Schreier tower

B.1 Exact statement

Let p be odd, F=F_p((t)), and 0<h_1<…<h_d with p not dividing any h_i. Put q_i=(p+1)h_i. The constructor below gives an actual matrix B_d over F of size p^d with exactly the field, simplicity, approximation-level, odd-prime quotient, and conductor conclusions of `RECURSIVE_TOWER166.md`. In addition, its matrix trace is exactly zero. Every native shorter constructor and every amplified arithmetic approximation also has matrix trace zero. Therefore every difference between consecutive arithmetic approximations has matrix trace zero.

The trace statement is an identity over the current coefficient field, not merely a congruence, a field trace inferred from characteristic zero, or a division by the matrix size. All coefficient fields have characteristic p, so division by p^j would be invalid.

B.2 1. An explicit separable minimal extension with a traceless parameter

At an arbitrary recursive step let L have residue field F_p and uniformizer t_L. Given q>0 prime to p, set

```
E=L(alpha),  alpha^p-alpha=t_L^(-q).
```

The right side is not an Artin–Schreier coboundary in L: the negative valuation of x^p-x, if x has negative valuation, is divisible by p. Thus the polynomial is irreducible of degree p. The extension is separable, and the valuation equation forces ramification index p and

```
v_E(alpha)=-q.
```

It is therefore totally ramified and alpha is minimal over L. For odd p its field trace is

```
Tr_(E/L)(alpha)=sum_(c in F_p)(alpha+c)=0.
```

Choose the unique b in {1,…,p-1} with qb congruent to -1 modulo p, put a=(qb+1)/p, and define

```
z=t_L^a alpha^b.
```

Then v_E(z)=pa-qb=1, so z is a uniformizer and E=L(z). Its monic minimal polynomial $f(X) = X^p + a_{p-1}X^{p-1} + \cdots + a_0$ is Eisenstein. It is an explicit polynomial: compute the product over the p conjugates

```
f(X)=product_(c in F_p) (X-t_L^a(alpha+c)^b)
```

and reduce coefficients using $\alpha^p - \alpha = t_L^{-q}$. The invariant coefficients belong to L. This is a finite algebraic instruction, with no choice of an unknown extension or matrix.

B.3 2. Normalized corestriction and an explicit filtered right inverse

Use the basis 1,z,…,z^(p-1) of E over L. If T_ij is its coefficient matrix unit and pr is the trace corestriction, then

```
pr(T_ij)=z^i z_j^*,
z_j^*=sum_(k=j+1)^p a_k z^(k-j-1) / f'(z),  a_p=1.
```

Set

```
kappa=-f'(z)/z^(p-1),  s=kappa pr.
```

The displayed dual-basis formula gives

```
s(T_ij)=-z^(i-j) + terms of strictly larger E-valuation.
```

Indeed v_E(a_k)>=p for 1<=k<p, so the contribution indexed by a_k has valuation at least k greater than the first contribution. Applying s entrywise on E^m gives a B-bimodule corestriction from End_L(E^m) to B=End_E(E^m). With the complete interleaved period-pm L-chain and period-m E-chain, it preserves grades and satisfies $s(P^r) = Q^r$.

Let P0 extract the constant coefficient on every E-coordinate. The constant dual coefficient satisfies

```
z_0^*=-a_0/(z f'(z)),
u_0=s(P0)=kappa z_0^*=a_0/z^p in O_E^times.
```

For any C in M_m(E), define the actual L-linear matrix

```
Lift_E(C)=u_0^(-1) C P0.
```

Then

```
s(Lift_E(C))=C.
```

This lift preserves the exact filtration valuation. To see this, first multiply C by the unit u_0^(-1), which does not change its valuation in the E-chain. Expand each entry in the basis 1,z,…,z^(p-1). The only nonzero columns of Lift are its constant-coefficient columns. The L-grade of a coefficient c_i z^i E_uv is

```
pm*v_L(c_i)+mi+u-v,
```

equal to its E-grade, and distinct i have valuations distinct modulo p. Thus the minimum cannot cancel. This also proves the stated surjectivity $s(P^r) = Q^r$ without a compactness assumption or an assumption that a lift exists.

B.4 3. Exact trace identity

For y in E, extraction of its constant coefficient is

```
coeff_0(y)=Tr_(E/L)(z_0^* y).
```

The ordinary L-matrix trace of the rank-one operator yP0 is coeff_0(y). Consequently, for every C in M_m(E),

```
tr_L(Lift_E(C))
   =Tr_(E/L)(z_0^* u_0^(-1) tr_E(C))
   =Tr_(E/L)(kappa^(-1) tr_E(C)).            (T)
```

Here tr_E(C) is the ordinary m-by-m matrix trace over E. In particular, if tr_E(C)=0, then the lift has ordinary L-matrix trace zero exactly.

B.5 4. The recursive actual matrix

Define B(L,t_L;h_1,…,h_d) by induction. For d=1 use the multiplication matrix of the alpha from Section 1 with q=q_1. For d>=2 use Section 1 with q=q_d, put m=p^(d-1), recursively construct

```
C=B(E,z;h_1,...,h_(d-1)),
```

and set

```
B(L,t_L;h_1,...,h_d)=alpha I_m+Lift_E(C).      (ASR)
```

Everything in (ASR) is explicit in finite field operations and the displayed degree-p polynomial extensions. It differs from the old conjugacy-preserving two-summand construction. Preserving old lower summand conjugacy classes is not claimed.

Inductively tr_E(C)=0. Equation (T) and the Artin–Schreier trace give

```
tr_L(B)=m Tr_(E/L)(alpha)+tr_L(Lift_E(C))=0.
```

For the induction base the trace is Tr_(E/L)(alpha)=0. Thus this argument proves tracelessness in the current base field for every native rank, not merely tracelessness after an amplification whose size is divisible by p.

B.6 5. Actual fields and profiles

Set gamma=alpha I_m and b=Lift_E(C). The minimal outer stratum has

```
n=p^(d-1)q_d,  k_0(gamma)=-n.
```

The inner actual maximal field supplied by induction has

```
R=p^(d-2)q_(d-1),  v(C)=-R,  k_0(C)=-q_1.
```

We have s(b)=C, v(b)=-R, and n>pR>2R because p is odd and the h_i increase. Hence

```
-2R > -n-q_1 = k_0(gamma)+k_0(C).
```

BK Remark 2.2.7 applies to this pure derived maximal field. It gives the field generated by the actual displayed gamma+b, with degree and ramification p^d, residue degree one, normalization of the complete chain, and k_0=-q_1. Its valuation remains -n. Thus [A,n,0,B] is simple.

This repeats the precise argument of `RECURSIVE_TOWER166.md` with a new outer minimal extension and normalized right inverse. No step of that argument required the old polynomial e^p+t e+t.

B.7 6. Native approximations, amplification, and traceless corrections

For the full rank p^d matrix put, for 1<=j<=d,

```
B^[j]=B(F,t;h_(d-j+1),...,h_d) tensor I_(p^(d-j)).
```

These are literal truncations of (ASR). At the outer level the same largest q_d selects the same Artin–Schreier extension and uniformizer in all truncations. At the next level the same q_(d-1) does the same, and so on. Thus no hidden change of bases or incompatible parameter fields enters the truncations. Lift_E commutes with amplification by an identity because P0 acts separately on each coordinate.

Every native shorter B has trace zero by Section 4. Its amplification therefore has trace zero. In particular

```
tr_F(B^[j])=0,
tr_F(B^[j+1]-B^[j])=0.                          (Z)
```

The same is true at native rank p^(j+1): the correction between B(F,t;h_(d-j),…,h_d) and the identity-p amplification of B(F,t;h_(d-j+1),…,h_d) has trace zero. This native assertion is the one relevant when constructing characters before each transfer.

The field degrees and critical levels of $B^{[j]}$ are exactly

```
degree p^j,  n_j=p^(d-j)q_(d-j+1).
```

Valuation preservation by every Lift gives

```
v_A(B^[j+1]-B^[j])=-n_(j+1).
```

Thus the approximation and defining-sequence proof of the first note applies unchanged. The odd-prime J/H dimensions, grade-index formula, Swan conductor and self-pair conductor depend on these same levels and centralizer degrees, so all their formulas carry over unchanged.

B.8 7. Precisely what the trace supplies

For any displayed correction delta and any scalar u in L^times for which its additive correction character is defined,

```
psi_delta(u I)=psi_L((u-1)tr_L(delta))=1.
```

This proves triviality of the actual additive correction on the scalar subgroup of its domain. It does not, by itself, construct a simple character with prescribed values on larger scalar groups. Compatible extension, transfer, and intersection arguments for the full simple character are separate obligations.

The note also does not identify the constructed candidates with any Weil source, choose them by inverse LLC, realize arbitrary stabilizer operators, or prove same-candidate H5. Its new output is an explicit arithmetic defining sequence whose trace vanishes at every native and amplified stage, suitable as input to that separate character argument.

B.9 Primary dependency

Bushnell–Kutzko, The admissible dual of GL(N) via compact open subgroups, Remark 2.2.7 (printed p. 55), with the defining-sequence and ring recursion already pinned in `RECURSIVE_TOWER166.md`. Retained source SHA256: `d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8`. The trace-dual and Artin–Schreier calculations above are given in full.

C Native-rank full characters and tame central completion

Source record: Paper_XI_Base_Realization165. Arithmetic input: `attachment166/tower/TRACE_ZERO_AS_TOWER166.md`, together with its `RECURSIVE_TOWER166.md` defining-sequence proof. Input hashes are recorded in `TRACE_ZERO_CHARACTER_PINS166.json`.

Evidence: ordinary proof using the explicitly constructed arithmetic parameters and the named Bushnell–Kutzko character and transfer theorems. No arithmetic-induction/descent constructor or inverse correspondence is used. No Lean verification is claimed.

C.1 1. Statement

Let p be odd, F=F_p((t)), and

```
0<h_1<...<h_d,       p not dividing any h_i,
q_i=(p+1)h_i,        N=p^d.
```

Let beta_d be the actual trace-zero Artin–Schreier matrix constructor in the named tower note. There exists an actual full simple character theta for its maximal simple stratum such that

```
theta restricted to U_F^1 is the trivial character.            (1)
```

Every step constructing theta is specified below: a finite abelian character extension, the standard simple-character transfer, or the explicit additive correction from the displayed defining sequence. The field, order, and parameter are the actual tower ones, not an unspecified arithmetic realization.

The actual canonical beta extension can then be completed to a maximal extended type and supercuspidal pi with any prescribed smooth central quasicharacter omega_F trivial on U_F^1. Its rank, Swan conductor, torsion number and self-pair conductor remain those proved for the same tower. The residue and noncompact extension choices must be free or already compatible with omega_F.

For p=3, this gives full scalar-pushout attachment and same-candidate sector recovery for any number d of the supplied cubic source planes, with declared stabilizer C={1} and the one-vertex contract. All original full source character values are retained in the attachment. A general nontrivial stabilizer, transport diagram, or H5 conclusion is not part of this statement.

C.2 2. The exact arithmetic inputs

For 1<=j<=d, put

```
b_j=B(F,t;h_(d-j+1),...,h_d) in M_(p^j)(F),
```

where B is the trace-zero constructor. The tower proof gives:

1. K_j=F[b_j] is a totally ramified maximal field of degree p^j, normalizing the native complete period-p^j chain A_j.
2. For $1 \leq j < d$, in rank p^(j+1), gamma_j=b_j tensor I_p is a simple approximation to b_(j+1), at the critical level r_j=q_(d-j)>0 of b_(j+1). They are equivalent as strata at that level.
3. For $1 \leq j < d$, delta_j=b_(j+1)-gamma_j has grade -r_j and F-matrix trace zero.
4. At native rank p, b_1 is multiplication by alpha in the explicit Artin–Schreier field
   `E=F(alpha),       alpha^p-alpha=t^(-q_d).`
   Its actual field trace is zero, since p is odd.

For completeness, the tower's exact trace identity is

```
tr_L Lift_E(C)=Tr_(E/L)(kappa^(-1) tr_E(C)),
```

where s=kappa*pr is its normalized trace corestriction and Lift_E(C)=s(P0)^(-1) C P0. The recursive inner matrix has trace zero, so this proves zero trace at every native rank. It is not merely the automatic zero obtained by amplifying any matrix p times in characteristic p. The literal compatibility of the truncations is proved in Section 6 of the tower note.

C.3 3. A finite character extension fact

Let K/F be a finite totally ramified local extension. Suppose chi is a smooth character of U_K^a, and chi is trivial on U_K^a intersect U_F^1. Then chi extends to a smooth character of U_K^1 trivial on U_F^1.

Indeed, on U_K^a U_F^1 define chi_0(ku)=chi(k). Agreement on the intersection makes this a character. Choose M>=a such that chi is trivial on U_K^M. The character chi_0 descends to a subgroup of the finite abelian group U_K^1/U_K^M. It extends to that whole group by successively adjoining generators and selecting complex roots. Pulling it back proves the assertion. This also describes an actual finite procedure for making the extension choices.

This fact will be used only when the parameter field is maximal in the current native matrix rank. Direct use in a nonmaximal centralizer would require checking its determinant image and would not follow from the scalar intersection alone.

C.4 4. Base character at native rank p

Let E=K_1, and let n_1=q_d be the native level of alpha. Since n_1 is even, set a=n_1/2+1. On U_E^a the forced character of field units is

```
phi(v)=psi_F(Tr_(E/F)(alpha(v-1))).
```

It is trivial on U_E^a intersect U_F^1 because Tr_(E/F)(alpha)=0. Section 3 gives a full character chi_E of U_E^1 extending phi and trivial on U_F^1. No normalization of the additive character is needed for this implication: the trace is exactly zero.

For the native minimal stratum, $H^1 = U_E^1$ U_(A_1)^(a). Define theta_1 by chi_E on U_E^1 and by psi_alpha on U_(A_1)^a. They agree on the intersection, so BK 3.2.1 supplies the actual full simple character. It satisfies theta_1 | U_F^1=1.

C.5 5. Transfer the full field character before adding the next layer

Suppose theta_j has been constructed on the native rank-p^j stratum of b_j, with theta_j | U_F^1=1. Its restriction to U_(K_j)^1 is the full beta-determinant character chi_j, since its native centralizer is K_j itself.

Use the standard simple-character transfer to the multiplicity-p realization gamma_j=b_j tensor I_p in rank p^(j+1). Denote the transferred full character by Theta_j. The exact primary result is BK 3.6.13(ii), with existence in Theorem 3.6.14: transfer preserves the full beta-determinant character on U_(K_j)^1. Consequently its restriction to the centralizer order is chi_j composed with det_(K_j).

For every literal scalar u in U_F^1 this gives

```
Theta_j(u I)=chi_j(u^p)=theta_j(u^p)=1.            (2)
```

The determinant exponent p is essential. We do not replace (2) by an unsupported pullback along a scalar matrix embedding. This transfer also deals with nonminimal b_j: beta-determinant preservation is the general theorem, not just the minimal-character formula.

C.6 6. Add the actual correction and extend at maximal rank

Let r=r_j=q_(d-j), m=floor(r/2)=r/2, beta=b_(j+1), gamma=gamma_j, and delta=beta-gamma. The actual defining sequence gives

```
D=H^(m+1)(beta)=H^(m+1)(gamma).
```

By BK 3.3.18, the character

```
vartheta=Theta_j restricted to D times psi_delta
```

is an actual simple character at level m for beta. For every scalar u in D intersect U_F^1, equation (2) and the exact trace identity give

```
vartheta(u)=Theta_j(u)
               psi_F((u-1)tr_F(delta))=1.        (3)
```

The new parameter field K=K_(j+1) is maximal and totally ramified, and the native chain has K-period one. Thus

```
U_K^1 intersect D=U_K^(m+1),
U_F^1 intersect D=U_F^ceil((m+1)/p^(j+1)).
```

In particular (3) supplies the exact compatibility condition for extending vartheta | U_K^(m+1) while imposing triviality on all U_F^1. Apply Section 3 to obtain such a character chi_K of U_K^1.

BK 3.2.5, including its explicit extension formula, now constructs

```
theta_(j+1)(kd)=chi_K(k) vartheta(d),
k in U_K^1, d in D.
```

The proposition supplies normalization of vartheta by the field units, well-definedness, and the full simple-character condition. This theta_(j+1) has the claimed trivial wild center. Induction proves (1) at the actual final rank p^d.

The proof deliberately performs every arbitrary character extension at maximal native rank. Extending directly along amplified intermediate centralizers M_(p^(d-j))(K_j) would introduce determinant-image compatibility conditions. Merely knowing that their ambient matrix trace is zero would not discharge those conditions. Transfer first, followed by maximal-rank extension, removes that ambiguity rigorously.

C.7 7. Complete this same type with the prescribed tame center

Keep the actual canonical beta extension kappa of the resulting theta. The final K/F is totally ramified of degree N, with residue rank one. The compact type group satisfies

```
J/J^1 = k_K^times = F_p^times.
```

On U_F^1 the beta extension acts by theta, hence trivially. For a prescribed omega_F trivial on U_F^1, choose the residue character

```
sigma=omega_F|F_p^times divided by omega_kappa|F_p^times.
```

Its inflation gives lambda=kappa tensor sigma with the right center on O_F^times. Complete lambda to an actual extension Lambda_0 on the inducing group K^times J using the retained maximal-type completion theorem. If its center is omega_0, choose s in C^times with

```
s^N=omega_F(t)/omega_0(t).
```

Multiply Lambda_0 by the character of K^times J trivial on J and taking a K-uniformizer to s. The quotient (K^times J)/J is Z, and t has image N. This proves exact agreement of the resulting central character with omega_F on all F^times. Canonical kappa is unchanged.

The resulting pi=c-Ind Lambda is irreducible, admissible and supercuspidal. All these choices are made before H5. If a residue character or noncompact extension has been immutably supplied, it must already satisfy this compatibility. An incompatible datum is not silently replaced.

By the exact same parameter and defining-sequence calculations,

```
rank(pi)=N=p^d,
torsion(pi)=1,
Swan(pi)=p^(d-1)(p+1)h_d,
a(pi^vee x pi)=p^(2d)-1
                + sum_i (p^(2i)-p^(2i-2))h_i.
```

No determinant twist changes the depth. These are the proved arithmetic tower profile formulas, not a proof of H5.

C.8 8. Full packet attachment in arbitrary cubic length

For p=3, retain d original supplied full characters theta_i and their actual character groups H_i^1 normal in J_i^1, with whole symplectic quotients V_i that are the d labeled planes. Assume the declared stabilizer is C={1} and the diagram is one vertex. The source balanced scalar extension has quotient V_src of dimension 2d and Heisenberg carrier dimension 3^d, the original rank.

The actual final host has total quotient dimension N-1. Its leading nondegenerate symplectic layer has dimension

```
(p-1)p^(d-1)=2*3^(d-1) >= 2d.
```

This follows either from the explicit defining-sequence quotient calculation or the independently proved leading-layer lemma. Choose a labeled symplectic embedding of the supplied V_src in that layer. Write W=V_src perpendicular-sum V_perp. Choose a maximal isotropic subspace L in V_perp. Its associated eigenspace in the actual Heisenberg carrier has dimension 3^d, and L-perp/L is V_src.

Form the actual reduced scalar extension using the full host character and its extension to the inverse image of L. Form each original full pushout

```
P_i=(C^times x J_i^1)/
          {(theta_i(h)^(-1),h): h in H_i^1}.
```

The center-fixing symplectic-extension isomorphism, constructed by matched Darboux lifts and complex roots, maps their balanced product to that actual reduced scalar extension. In particular each original h in H_i^1 maps to the scalar theta_i(h). Every prescribed shallow and deep source character value is retained. Source character attachment does not claim that the original field units are the same literal scalar matrices in the new host.

Finite Stone–von Neumann supplies the full carrier intertwiner. The retained sector induction and recovery idempotent then recover this same Lambda and this same pi from the source-sized sector. The canonical beta extension, residue factor, central character, and noncompact scalar are those actually selected in Section 7. No source-independent alternate candidate is substituted in recovery.

For the retained independent tensor family with d>=2, each factor's wild determinant has order dividing p, because the abelianization of its extraspecial order-p^3 image is elementary abelian. The tensor determinant raises each factor determinant to p^(d-1), so the full source determinant is trivial on wild inertia. Thus the source's prescribed central quasicharacter lies in the class covered by Section 7. At d=1, matching this central conclusion requires the source determinant itself to be trivial on wild inertia. It is not inferred merely by taking a tensor power of length one.

For p>3, the same arithmetic host and character theorem hold, and the same attachment works for any actually supplied valid packet whose whole quotient and markings meet the stated symplectic

contract. A proposed source plane is not automatically a whole rank-p type quotient: that latter quotient has dimension p-1. No admission of all such plane sources into original H2 is inferred.

C.9 9. Boundary and primary-source check

This provides the previously missing full-character and actual wild-center construction for the explicit odd-prime arithmetic tower in every prime-power rank. The proof uses no classical automorphic-induction/descent candidate. It retains the supplied finite characters through the declared scalar-pushout attachments.

It does not preserve old summand matrix conjugacy classes, produce a prescribed nontrivial stabilizer action, preserve arbitrary numerical depth labels, handle arbitrary transport graphs, settle the dyadic case, or prove original same-candidate H5. It therefore does not close unrestricted claim 66.

The retained primary BK text, SHA256 `d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8`, was directly checked at:

- Definition 3.2.1 and Proposition 3.2.5, printed pp. 98–100: minimal character condition and explicit full-character extension.
- Corollary 3.3.18, printed pp. 112–113: the actual additive correction takes the approximation character to the next simple character.
- Condition 3.6.13(ii) and Theorem 3.6.14, printed pp. 136–137: transfer preserves the full beta-determinant character, including at m0=0.

The actual field and defining-sequence assertions are the separately proved and independently reviewed trace-zero tower inputs, with BK 2.2.7 and the retained simple-stratum theorems. No newly downloaded source or new novelty claim is involved.

D The surviving leading symplectic layer

Status: ordinary proof from the retained Bushnell–Kutzko theorems and, for the source-defined candidate, the substantial published Bushnell–Henniart arithmetic induction/descent constructor. No Lean build or bounded computation is used. The accompanying audit files identify the scope of the internal review.

D.1 The useful simplification

Full critical-profile rigidity is unnecessary for the trivial-stabilizer finite attachment contract. An even leading level produces a symplectic subspace that survives every later simple approximation. For the independent rank-p Heisenberg tensor sources, this single subspace already has enough dimension to hold every source plane, however many there are.

The argument below concerns the actual type in the source-defined arithmetic candidate. It does not identify that type with a separately guessed binder parameter.

D.2 1. Leading-layer theorem

Let F have residue field k of cardinality q and residue characteristic p. Let pi be an actual irreducible supercuspidal representation of GL_n(F) with torsion number one, where n=p^r. Suppose

```
a(pi)=n+N,     N>0 even,     v_p(N)=r-1.
```

Then an actual maximal simple type in pi, with Heisenberg quotient $W = J^1/H^1$, contains a nondegenerate symplectic F_p-subspace S of dimension

```
[k:F_p] (p-1)p^(r-1).
```

For k=F_p this is exactly (p−1)p^(r−1). The subspace is the image of the concrete congruence group U^(N/2)(A) in J^1/H^1. It is present regardless of the later critical exponents, their parities, or the number of intermediate approximation fields.

D.2.1 1.1. Actual parameter field and leading level

By BK maximal-type existence, pi contains a maximal simple type. Its positive level follows from a(pi)>n. Write its stratum as [A,N_A,0,beta], with parameter field K=F[beta]. BK 6.2.5 gives

```
t(pi)=n/e(K/F).
```

Thus e(K/F)=n. Since [K:F] divides n, K/F is totally ramified of degree n and has residue degree one. The principal-order assertion after BK 6.2.5 gives e(A)=n. Accordingly A is the complete principal hereditary order. The established conductor formula

```
a(pi)=n+n N_A/e(A)
```

now implies N_A=N. This is the same source-checked formula used in Block165, originating in BH1999 Section 6.1 Lemma 2 and retained in `PRINCIPAL_ORDER_CONDUCTOR161.md`.

Put P=rad(A), $U^a(A) = 1 + P^a$ for a>=1, and m=N/2.

D.2.2 1.2. The terminal minimal approximation has degree p

Apply the defining-sequence construction BK 2.4.1–2 to beta. Write gamma for its terminal minimal approximation. Every transition has a critical level c<N, and gamma has the same A-level N as beta. All approximation residue degrees divide f(K/F)=1, and all approximation ramification degrees divide n. Therefore E=F[gamma] is totally ramified of degree d dividing n.

The scalar case is impossible: a scalar of F has A-valuation divisible by n, whereas v_p(N)=r−1. Thus gamma is noncentral and minimal in the ordinary field sense. Its field valuation is

```
−v_E(gamma)=Nd/n.
```

Minimality, by BK 1.4.14–15, requires gcd(Nd/n,d)=1. Since d is a p-power and v_p(N)=r−1, integrality first gives p|d, while coprimality forces d=p. Hence E/F is totally ramified of degree p.

Set B=End_E(F^n), B_order=A∩B, and Q=rad(B_order). The same complete lattice chain has period n/p over E. Each P-grade has k-dimension n and each Q-grade has k-dimension n/p. In particular,

```
dim_k P^m/(Q^m+P^(m+1))=n−n/p.
```

No assertion that E is literally a subfield of the final field K is needed. It is the actual minimal approximation field in the defining sequence.

D.2.3 1.3. The high subgroup survives every approximation

The minimal definitions BK 3.1.7–8 give

```
J_order(gamma)=B_order+P^m,
H_order(gamma)=B_order+P^(m+1).
```

At a nonminimal transition beta_i→beta_(i+1), with critical level c_i<N, the recursive J-order definition uses the threshold ceil(c_i/2). Since ceil(c_i/2)<=m, the inclusion P^m⊂J_order(beta_(i+1)) implies P^m⊂J_order(beta_i). Starting at gamma, induction proves

```
U^m(A)⊂J^1(beta).
```

For H, BK 3.1.9(ii) states that $H^a(\beta_i) = H^a(\beta_{i+1})$ as soon as

```
a>=floor(c_i/2)+1.
```

Every c_i is at most 2m−1, so this threshold is at most m. Consequently

```
H^m(beta)=H^m(gamma).
```

Here H^m means $H^1 \cap U^m(A)$. Therefore the image S of U^m(A) in W is canonically

```
S = U^m(A)/H^m(beta)
  = U^m(A)/H^m(gamma)
  ≅ P^m/(Q^m+P^(m+1)).
```

The last group identification uses m>=1, so multiplication is additive modulo P^(m+1). Thus |S|=q^(n−n/p).

D.2.4 1.4. Its actual commutator form is nondegenerate

Choose the actual simple character theta contained in the type of pi. By the standard deep-restriction property of simple characters, theta restricts to psi_beta on U^(m+1)(A). This follows directly from BK 3.2.1–4 by induction along the defining sequence.

For u,v in U^m(A), their group commutator is in $U^{2m}(A) = U^N(A)$, which lies in U^(m+1)(A). The terminal approximation satisfies

```
beta−gamma∈P^(1−N),
```

because every critical level in the sequence is less than N. Hence

```
(beta−gamma)([u,v]−1)∈P.
```

With the standard level-one additive character psi_A=psi_F∘tr_A/F, psi_A is trivial on P. We therefore have the exact equality

```
theta([u,v]) = psi_beta([u,v]) = psi_gamma([u,v]).
```

The right side is the Heisenberg commutator pairing for the minimal simple stratum of gamma. Its quotient $J^1(\gamma)/H^1(\gamma)$ is precisely $U^m(A)/H^m(\gamma)$, because $J^1(\gamma) = U^1(B_{\mathrm{order}})U^m(A)$, with $U^1(B_{\mathrm{order}}) \subset H^1(\gamma)$. BK's nondegeneracy theorem for simple-character commutators therefore says that the form on S is nondegenerate.

For general k, this is a nondegenerate alternating F_p-form of dimension $[k : \mathbb{F}_p](n - n/p)$. For $k = \mathbb{F}_p$, it has dimension $n - n/p$. This proves the theorem.

The argument uses no self-pair conductor and no determination of the later critical levels. Those invariants remain useful for finer arithmetic identification, but they are not needed for this finite-sector capacity claim.

D.3 2. Arbitrary-length odd-prime source family

Fix an odd prime p and F=F_p((t)). Let the complete marked Weil source rho be an independent tensor product of r rank-p wild Heisenberg factors at distinct positive levels

```
0<h_1<...<h_r,     p does not divide h_i.
```

Here independence means the stipulated balanced-central-product description of the wild image is valid, and rho|P_F is irreducible. We do not infer independence from the inequalities alone. Suppose the retained Tsushima source calculation supplies the usual highest slope

```
slope(rho)=(p+1)h_r/p.
```

Then rank(rho)=n=p^r and

```
Swan(rho)=N=(p+1)p^(r−1)h_r.
```

Let Pi_BH(rho) be the published source-defined arithmetic induction/descent candidate, with its full prescribed central character, as audited in Block165. The construction is applied to the complete actual rho, including its scalar normalization. Its codomain is the torsion-one supercuspidal class, and its standard epsilon-function property gives a(Pi_BH(rho))=n+N. These are uses of the substantial classical constructor, not consequences of finite Heisenberg algebra alone and not a selection of an inverse LLC image.

For odd p, N is even and v_p(N)=r−1. The leading-layer theorem therefore puts a symplectic space of dimension

```
D=(p−1)p^(r−1)
```

inside the actual Heisenberg quotient of a type in this same Pi_BH(rho).

The source packet is the orthogonal sum of r planes, so its symplectic dimension is 2r. The elementary inequality

```
(p−1)p^(r−1)>=2r
```

holds for every odd p and r>=1: it suffices that 3^(r−1)>=r, which follows by induction. Thus the actual candidate's leading symplectic layer alone can receive the entire source packet.

D.3.1 2.1. Full characters and recovery for declared C={1}

Work under the same finite scalar-pushout contract as `SAME_CANDIDATE_ATTACHMENT165.md`: declared stabilizer C={1}, one vertex, and full source character extensions retained, with no requirement of literal old matrix-group inclusion or equality of old numerical depths with host depths.

Let P_src be the balanced C×-central product of the r supplied full source character extensions. Its quotient V_src has dimension 2r over F_p and its central-character-one Heisenberg representation has dimension p^r=n. Choose a symplectic embedding V_src→S⊂W.

The C×-pushout of the actual type character, restricted to this subspace, has the same alternating commutator as P_src. The retained extension theorem supplies a center-fixing isomorphism from P_src to that restriction. Consequently every source h∈H_i^1 is carried to the scalar theta_i(h), preserving the full character value. No shallow source values are discarded.

Let V be the image and write W=V⊥V_perp. Stone–von Neumann gives

```
eta_W ≅ eta_src⊗eta_perp.
```

Choose a line in eta_perp and the corresponding rank-n projector e in the actual finite Heisenberg image algebra. Then e eta_W≅eta_src. Applying the retained Block159 induction/idempotent theorem to the unchanged actual extended type Lambda in Pi_BH(rho) recovers Lambda and then this exact Pi_BH(rho). All source normalization and the actual noncompact scalar remain fixed.

This proves same-candidate finite attachment for arbitrary r in this C={1} odd-prime family. Original H5, when supplied for this same candidate, remains exactly the original conditional comparison premise. We do not prove it here.

D.3.2 2.2. First new rank: GL_27

For p=3,r=3, the source has rank 27 and V_src has dimension 6. Put h=h_3 with 3∤h. The actual candidate has

```
N=36h,       m=18h,
dim_F3 S=27−9=18.
```

Thus its high layer contains the entire three-plane source packet. At least a 3^9-dimensional Heisenberg contribution is present, while the required source sector has dimension 27=3^3. The six surplus symplectic pairs can be projected onto a line in their representation, together with any additional lower layers. This proves the required capacity and sector realization for the new rank 27 family without solving its two possible lower critical levels.

In particular, for (h_1,h_2,h_3)=(1,2,4), N=144 and the same high subspace has dimension 18. These numbers are a specialization of the universal proof, not a numerical experiment.

D.4 3. Why trying to force every critical level was unnecessary

At rank 27, a hypothetical complete degree pattern 27→9→3 has leading level N=36h_3 and lower critical levels a<b<N. The standard full-profile conductor count would give

```
Swan(End rho)=18N+6b+2a.
```

Comparing with the stipulated source formula 8(h_1+9h_2+81h_3) only gives

```
a+3b=4h_1+36h_2.
```

That single linear relation does not determine both a and b, and by itself does not force each parity. Nor does it establish that all three degrees occur rather than skipping degree 9. This observation is only a warning about the insufficient equations. It is not presented as an arithmetic counterexample or as existence of every numerical profile satisfying the equation.

The leading-layer theorem bypasses these unresolved profile choices. It supplies a fixed nondegenerate high subspace in every actual candidate with the two required invariants.

D.5 4. Dyadic capacity boundary

The leading-layer theorem itself is valid at p=2. If an actual torsion-one candidate of rank 2^r has

```
N=3·2^(r−1)h,     h odd,     r>=2,
```

then its high subspace has dimension 2^(r−1) over F_2. This alone is at least 2r for r>=4, with equality at r=4. It does not supply enough capacity by itself for r=2 or 3. For r=1, N is odd and this theorem does not apply.

This is a capacity statement only. Applying it to a dyadic source requires the actual source family, its arithmetic candidate and whatever quadratic/phase markings the original contract requires. No universal dyadic attachment theorem is claimed from dimension alone.

D.6 5. Boundaries that remain

1. The full character attachment here is through center-fixing scalar-pushout maps. Literal embeddings of prescribed old arithmetic matrix groups are not proved.
2. No nontrivial prescribed stabilizer action is attached. Capacity does not imply equivariant placement.
3. All planes may be placed in the leading host layer. Preservation of supplied numerical ramification depths, or a specified map into individual host depth layers, would be an additional requirement and is not claimed.
4. General packet transport graphs are not handled here. The statement is for C={1} and one vertex.
5. The arithmetic induction/descent theorem is an explicit major external input. This is a new capacity/attachment consequence for its actual output, not a new elementary reconstruction of that classical constructor.
6. H5 remains conditional and refers to that unchanged actual candidate. No additional pair-factor premise is introduced.
7. No Lean verification is claimed.

D.7 6. Primary locators and retained dependencies

The authorized primary BK text was read at these exact results: 1.4.14–15 (minimality), 2.4.1–2 (defining sequence and degree divisibility), 3.1.7–9 (recursive orders and high-depth equality), 3.2.1–4 (deep restriction of simple characters), 3.4.1 (nondegenerate Heisenberg commutator), and 6.2.5 with the following paragraph (torsion number and principal period).

The standard conductor and published arithmetic-constructor properties are reused from the source-pinned Block165 audit and Block161 conductor proof. The finite scalar-pushout and sector-recovery arguments are exactly those retained and restated in `SAME_CANDIDATE_ATTACHMENT165.md`. No entire copyrighted source is copied into this continuation.

E Actual scalar inertia phases and finite-cover descent

Evidence: ordinary proof using rank-one reciprocity, the given central character, and elementary finite group arguments. The result constructs actual central matrix operators through a finite cover and proves their descent. It does not construct a new arithmetic candidate or alter a canonical beta extension.

E.1 1. Exact statement

Let F be a nonarchimedean local field with residue field k of cardinality q and residue characteristic p. Put Q=q-1 and n=p^r, allowing r=0. Let rho:W_F -> GL_n(C) be a smooth Weil representation with finite inertia image. Use the fixed reciprocity convention identifying its determinant with a character omega:F× -> C×.

Let C be a finite group of order prime to p, and suppose its prescribed operators on the **full source carrier** are

```
Omega_src(c)=chi(c)I_n,
```

where chi is a character and each Omega_src(c) belongs to the actual matrix image rho(I_F). A finite cyclic scalar tame image is the intended example. This hypothesis refers to actual source operators, not freely chosen phases on an unrelated multiplicity-space factor.

Let Pi be any already constructed irreducible smooth GL_n(F) representation with central character omega. Suppose it has its supplied actual inducing representation Lambda on N, with F× contained in N, and an attached finite sector E=eLambda realizing the prescribed source Heisenberg packet. Assume the prescribed C action on the source scalar extension is trivial, and that the honest scalar phase on the packet is the same chi above.

Theorem. There is a finite group C_tilde surjecting onto C and a genuine homomorphism

```
g:C_tilde -> F× I_n contained in N
```

whose action on Lambda and on Pi descends to the prescribed honest C action chi(c)I. Every finite sector is stable under this action. Any center-preserving Heisenberg attachment and its intertwiner are C-equivariant. The retained recovery maps from sector induction recover the same Lambda and Pi equivariantly. All actual inducing data, including the fixed canonical beta extension, remain unchanged.

The arithmetic action is supplied by actual central matrices, with an explicit finite-cover descent. The theorem does not assert that C itself embeds as a matrix subgroup. Such an embedding can fail even for a genuine scalar tame source, as Section 4 shows.

E.2 2. Construction and proof

Let D=k×, and let Teich:D -> O_F× be the canonical multiplicative lift. For p=2 this is still the prime-to-p root-of-unity subgroup. When q=2, D is trivial and the argument below forces chi trivial as well.

Local reciprocity identifies the determinant image of inertia with

```
det rho(I_F)=omega(O_F×).
```

The unit group decomposes as

```
O_F× = Teich(D) times U_F¹.
```

The smooth character image omega(U_F¹) is a finite p-group, while omega(Teich(D)) has order prime to p. Therefore the prime-to-p elements of omega(O_F×) are exactly omega(Teich(D)).

For every c in C, the hypothesis on the actual source implies

```
chi(c)^n=det(Omega_src(c)) in det rho(I_F).
```

This value has order prime to p, and hence lies in omega(Teich(D)). Since n is a p-power and chi(C) has order prime to p, the map z -> z^n permutes chi(C). Consequently

```
chi(C) is contained in omega(Teich(D)).                (1)
```

In particular every source phase has order dividing Q. This conclusion does not require selecting a splitting of the tame quotient or choosing Frobenius representatives.

Choose an integer k_0 with n k_0 congruent to 1 modulo Q. When Q=1 take k_0=1. Define

```
delta(u)=omega(Teich(u))^(k_0),       u in D.
```

Because exponentiation by k_0 is an automorphism on every group of roots of unity of order dividing Q, delta(D)=omega(Teich(D)). Define the finite fiber product

```
C_tilde={(c,u) in C times D : chi(c)=delta(u)}.          (2)
```

It is a subgroup, and projection pr:C_tilde -> C is surjective by (1). Define the actual central matrix homomorphism

```
g(c,u)=Teich(u^(k_0)) I_n.                              (3)
```

Its action on Lambda and Pi is the scalar from the central character

```
Lambda(g(c,u))=omega(Teich(u^(k_0)))I
              =delta(u)I=chi(c)I.                      (4)
```

If (1,u) belongs to ker pr, (4) is the identity. Hence the represented action factors uniquely through C. It is exactly the prescribed source action. Although selecting a particular matrix representative for c may require a choice of u, the represented operator is independent of that choice. Changing k_0 by a multiple of Q leaves (3) unchanged.

All matrices (3) are central. Their conjugation on actual groups, characters, and scalar pushouts is therefore the identity, and their Lambda operators are scalar. Thus they preserve every e and every sector E. The source extension action is also the identity by hypothesis, so any attachment of those scalar extensions is equivariant. Both honest carrier actions are chi(c)I, so every Heisenberg intertwiner intertwines C without a new phase choice.

For the recovery step, put P={a in N:Lambda(a)eLambda(a)^(-1)=e}. Then F× is contained in P. The retained maps

```
Phi:Ind_P^N E -> Lambda,
j:Lambda -> Ind_P^N E
```

are N-equivariant. They therefore commute with all actual matrices (3), and ker pr also acts trivially on Ind_P^N E: its matrices are central elements of P acting trivially on E, hence trivially on every induced translate. Thus the equality descends through pr to C. The recovering idempotent, and then compact induction to GL_n(F), preserve this equivariance. They recover the same fixed Lambda and Pi. QED.

E.3 3. What this closes and what it does not

For a source packet whose marked tensor carrier is the entire rank-n source, this proves the stabilizer step whenever its genuine tame operators are scalar and an actual host with the correct central character and full-character attachment is already present. There is no remaining scalar-phase equation to solve in that situation: the determinant and central character supply it.

If only the source symplectic C action is known to be trivial, the preceding `STABILIZER_CONTRACT166.md` proves triviality of its full scalar-extension action for prime-to-p C. Schur then gives a scalar packet character. To apply the present theorem one must still identify that character with the action of actual inertia on the full source. That identification holds when the supplied honest packet carrier is the whole source carrier. It is not automatic for a proper multiplicity-space factor, whose omitted factor can contribute to the determinant.

For several factors the total packet character is the product of the supplied factor characters. The theorem matches that product on the whole attached packet. It does not represent independent factor actions by distinct central matrices unless those additional marked actions are separately specified and verified.

The result uses the fixed full inducing representation Lambda. If Lambda | J=kappa_can tensor sigma, then kappa_can and sigma are both unchanged. Equation (4) concerns Lambda, and does not assert that kappa_can alone has the same scalar character. A contract prescribing an action specifically on the canonical beta extension requires that distinction to remain explicit.

No nontrivial source symplectic action is realized here. No new base candidate, new H5 comparison, dyadic quadratic refinement, or general rank theorem is claimed. Existing quadratic refinements are preserved because the actual central conjugation action is the identity.

E.4 4. Why a direct map C into residue units is too strong

Take q=5, n=5^r, and let omega on k×=F5× be the quadratic character. Let C=C2 and let chi be its nontrivial character. Such scalar tame behavior is obtained by twisting a tame-trivial rank-5^r source by the quadratic tame character, whenever such a source is supplied. The retained odd-prime two-factor family supplies rank 25 examples.

A group homomorphism a:C2 -> F5× must send its generator to 1 or -1. Both have quadratic character 1. Consequently

```
omega(a(c)) = chi(c)^n
```

is impossible at the nontrivial c, since n is odd. The proposed direct section is not forced by genuine scalar tame source data. Reciprocity supplies a quotient of residue units. That quotient need not split.

The construction above still works. Here k_0=1 modulo 4, C_tilde is the cyclic group of order 4, and its central matrices act by the quadratic character. The kernel of C_tilde -> C2 acts trivially on the fixed candidate, so the actual operator action descends correctly. This example explains why finite-cover descent is necessary rather than a cosmetic reformulation.

E.5 Dependencies and review status

External input: rank-one local class field theory with the fixed normalization for $\omega = \det \rho$, the supplied candidate's central-character identity, and sector recovery. The finite-cover construction and descent are proved above without local Langlands candidate selection. No Lean certificate is claimed.

F The precise stabilizer-action contract and normalizer obstruction

Evidence: direct audit of the original statement, a new elementary reduction, and an explicit actual-source/actual-type H1–H2 witness. No Lean closure, class-wide arithmetic realization, or H5 witness is claimed.

Primary audit verdict: incomplete, with the smallest missing implication being the identification of the declared finite action with the action intended on a newly constructed common arithmetic candidate. The original text does not define that identification by a common matrix normalizer.

F.1 1. Exact contract, and what is already an input

The controlling text is `closure_attempt162/inputs/original_reading.txt`:

- Proposition 9.8, starting line 4834: a finite group acting by automorphisms of the one-break Clifford datum acts symplectically on its intrinsic envelope.
- Definition 11.1, lines 4911–4923: a bridge supplies an equivariant isometry to an actual type quotient for a finite cyclic stabilizer C, with common central character and additional dyadic lift data.
- Proposition 11.3 and Theorem 11.5: once the two honest cyclic actions are chosen, their discrepancy is a character, killed by equality of phases at the generator.
- Theorem 15.4 and Theorem 16.2: breakwise comparisons and transports are supplied and tensored. They do not construct a common arithmetic host from separate actual type targets.

Thus H2 and H3 already settle the finite action comparison **on each supplied break target**. They do not leave a second free stabilizer action to invent on that same target. Their transport into a new common arithmetic type is another assertion. Neither a map C into its matrix normalizer nor arbitrary transported operators are identified with that assertion by the original text.

The cyclic group in Definition 11.1 is not explicitly required there to have order prime to p. The reduction below assumes that order condition. It applies to the declared tame prime-to-p case, and cannot silently be applied to an arbitrary finite-order Frobenius operator.

F.2 2. Full scalar-extension rigidity for a native tame action

Let an actual maximal simple type in GL_n(F) have a totally ramified parameter field K/F of degree n=p^r and residue matrix rank one. Let

$$W=J^1/H^1,\quad P=(C\times J^1)/\{(\theta(h)^{-1},h):h\in H^1\},$$

and let eta be its irreducible Heisenberg representation. Let C be finite of order prime to p. Suppose the C action on P fixes the scalar center and is genuinely induced by character-preserving matrix normalizers of this type. It is enough that it is an honest action whose action on W comes from those normalizers. Then:

1. C acts trivially on W.
2. C acts trivially on the full scalar extension P.
3. Every honest C action Omega on eta implementing this action is Omega(c)=chi(c)I for one character chi of C.

Proof. The retained normalizer theorem in Block162 gives

$$N=K\times J^1,\qquad N/(F\times J^1)=\mathbb{Z}/n\mathbb{Z}.$$

The action on W factors through this cyclic p-group, so a homomorphic image of the prime-to-p group C is trivial.

For any central extension P of the elementary abelian p-group W by C×, the kernel of

```
Aut_(C×)(P) -> Aut(W)
```

is naturally Hom(W,C×). Indeed, an automorphism phi fixing the quotient and the center has phi(x)=a(x_bar)x. The scalar a is independent of the lift x and multiplication shows that a is a character. Conversely every such character defines an automorphism. Since W has exponent p, this kernel is a finite elementary abelian p-group. The action of C lies in this kernel and therefore is trivial. Finally an implementing operator commutes with the irreducible P action, so is scalar by Schur. Honesty makes its scalar a character. QED.

This strengthens the earlier W-level obstruction: passing to the full scalar pushout cannot hide a nontrivial native prime-to-p action above an identity quotient action.

F.3 3. Exact attachment criterion in this native setting

Suppose a marked source scalar extension P_src has a symplectic embedding into W, and an actual finite sector of eta realizes that extension. Fix its center-preserving attachment and a Heisenberg intertwiner T. If the host has the native tame action of Section 2, the attachment intertwines the full honest C actions if and only if

```
the source action on P_src is trivial, and
chi_src = chi_host.
```

Necessity follows from equivariance and Section 2. For sufficiency, trivial extension actions make both honest carrier actions scalar by Schur, and equal characters commute through every Heisenberg intertwiner. Any sector is stable under the scalar host action.

If the source C action on its symplectic quotient is trivial, its action on the full scalar extension is already trivial by the Hom-kernel argument. For several break factors, the source character is the product of their characters. Thus the criterion is a single equality of scalar characters, rather than a further system of symplectic placement equations.

This applies to the original C={1} fixture immediately. For nontrivial C, H3 equates source and **supplied break-target** phases. It does not equate their product with a separately built common host's phase without a dictionary. We do not change a fixed canonical beta extension to enforce this equality.

F.4 4. H2 and matching carrier dimensions do not force trivial C

Here is an actual H1–H2 witness under the literal original allowance of separate type targets. It is not a claimed H5-admitted input, and does not assert equal ambient matrix ranks for source and break target.

Take F=Fp((t)), p odd. The retained primary source by Tsushima, Definition 3.1, Lemmas 2.6, 2.8, 3.9, 3.12, 3.14 and Corollary 3.15, with R(X)=X^p and m=1, constructs a genuine rank-p Weil representation irreducible on wild inertia. Its wild image is extraspecial of order p^3. The source envelope is a symplectic plane, and its actual tame image C=mu_(p+1) acts by multiplication on the roots of $X^{(p^2)}+X$. This action is nontrivial and has order p+1. The source's finite Heisenberg carrier has dimension p.

For the actual type target, take the unramified quadratic extension E/F and the maximal order A=End_OF(OE). Let beta=t^(-2)a, where a is a unit generating the residue extension. This is a minimal simple stratum of even level two. The standard minimal simple-character formulas give

```
J¹=U_A¹,  H¹=U_E¹ U_A²,
W_theta=End_Fp(Fp²)/Fp²,     dim_Fp W_theta=2,
dim eta_theta=p.
```

These are actual groups of a simple character, not an abstract replacement. The type residue units act by conjugation. Write sigma(x)=x^p and decompose

```
End_Fp(Fp²)=Fp² direct-sum Fp² sigma.
```

On the quotient represented by b sigma, conjugation by u in Fp²× is

```
b -> u^(1-p)b.
```

Its image is the norm-one torus, of order p+1. Consequently the actual projective residue action of Fp²×/Fp× on the two-dimensional quotient is nontrivial. It has the same symplectic C-module as the source: identify the cyclic groups using the displayed ratio map. The two invariant alternating forms differ by a nonzero Fp scalar, and multiplication by lambda in Fp²× rescales an alternating form by Norm(lambda). Surjectivity of the norm gives an equivariant symplectic isometry.

This supplies the actual source, actual type quotient, matching degree-p Heisenberg carriers, and nontrivial equivariant H2 action. For p=3, C has order 4 and the action is an irreducible rotation of the F3 plane.

The target here is in GL_2(F), while the source has rank p. This is permitted by the literal breakwise definitions, which contain no relation between the individual type targets' ambient ranks. Block161 already recorded that exact omission. If a stricter intended contract requires every break target in a specified GL_(p^r)(F), this example does not settle that stricter question. We have not supplied H5 for a common candidate or certified fixed canonical beta-extension phases for this example. It refutes only the proposed inference that actual H2 plus equality of finite carrier dimensions forces C trivial.

F.5 5. Result for the next construction attempt

There are three distinct cases:

Given contract	Exact remaining issue
Original C={1} fixture	No stabilizer issue remains.
Native prime-to-p C action in a degree-p^r totally ramified host	Source extension action must be trivial. Then compare one scalar character.
Nontrivial source symplectic C action under the original finite contract	A dictionary for the arithmetic action on a common host is needed. Requiring an inner normalizer makes this impossible for the preceding host. Transporting external operators alone was already done in Block 161.

Thus "solve arbitrary nontrivial stabilizer attachment by adjusting the current totally ramified binder" is not a viable native-inner target. That is a precise failure of that stronger construction route, not a refutation of 16.2.

F.6 Evidence and self-review

Dependencies: original Sections 9–11,15–16; Block162's source-backed normalizer theorem; Block165's base realization contract and same-candidate attachment; the retained published Tsushima text; the retained minimal-stratum group formulas. The new Hom-kernel lemma and attachment criterion are ordinary elementary proofs. The witness uses the displayed arithmetic constructor inputs, not a

numerical experiment. No independent review of this new note is claimed in this file, and no new Lean compilation was attempted.

Self-review explicitly checked: the prime-to-p qualification, honest action versus projective implementers, center-fixed scalar extension automorphisms, separate target rank versus original rank, and the absence of an H5 witness.

G Dyadic synthesis and assembly scope for all lengths

The odd-prime construction used here is pinned in `TARGET167.md`. Evidence: ordinary mathematical proof and independent component reviews. No Lean verification or universal claim 66 closure is asserted.

G.1 1. Combined arithmetic theorem

Let $F = \mathbb{F}_2((t))$, fix an additive character ψ_F of conductor $t\mathcal{O}_F$, and take increasing odd positive integers $h_1 < \cdots < h_d$. Put

$$q_i = 3h_i, \qquad N = 2^d, \qquad n_j = 2^{d-j} q_{d-j+1}.$$

There is an explicit $\beta \in M_N(F)$ generating a separable totally ramified degree-N field that normalizes the complete lattice chain. It gives a maximal simple stratum with leading level n_1, final critical exponent $-q_1$, and successive approximation-field degrees $2, 4, \ldots, 2^d$, with critical levels $n_2, \ldots, n_d$.

The actual stratum admits a full simple character θ trivial on U_F^1. Its canonical beta extension can be completed to an actual irreducible admissible supercuspidal π of $\mathrm{GL}_N(F)$ with any prescribed smooth central quasicharacter ω trivial on U_F^1, provided the noncompact completion is free or already compatible. Over this residue field, such ω is unramified.

For this same constructed candidate,

$$t(\pi) = 1, \qquad \mathrm{Sw}(\pi) = 3\,2^{d-1} h_d,$$
$$\dim_{\mathbb{F}_2} W = N - 2, \qquad \dim \eta = 2^{N/2-1}, \qquad W = J^1/H^1,$$
$$a(\pi^\vee \times \pi) = N^2 - 1 + \sum_{i=1}^{d} 3\,4^{i-1} h_i.$$

Matching these profiles does not identify the source, its endo-class, or the pair factors in original H5.

G.2 2. The explicit construction and its depth invariant

At a recursive coefficient field L of characteristic two, with residue field $\mathbb{F}_2$ and uniformizer t_L, put

$$E = L(\alpha), \qquad \alpha^2 - \alpha = t_L^{-q}, \qquad a = (q+1)/2, \qquad z = t_L^a \alpha.$$

The extension is separable and totally ramified of degree two, and z is a uniformizer and

$$z^2 + t_L^a z + t_L = 0, \qquad \mathrm{Tr}_{E/L}(\alpha) = 1.$$

For constant-coefficient projection P_0 in the basis $1, z$, the normalized trace corestriction and exact filtered right inverse are

$$\kappa = \alpha^{-1}, \qquad s = \kappa\,\mathrm{pr}, \qquad s(P_0) = t_L/z^2, \qquad \mathrm{Lift}_E(C) = (z^2/t_L) C P_0.$$

They satisfy $s(\mathrm{Lift}_E(C)) = C$, preserve the exact filtration valuation, and give the identity

$$\mathrm{tr}_L\,\mathrm{Lift}_E(C) = \mathrm{Tr}_{E/L}\big(\alpha\,\mathrm{tr}_E\,C\big).$$

The actual recursive matrix is

$$B(L, t_L; h_1, \ldots, h_d) = \alpha I_{2^{d-1}} + \mathrm{Lift}_E\big(B(E, z; h_1, \ldots, h_{d-1})\big),$$

with multiplication by α as the rank-two base case.

Every native constructor has trace one. At every next native rank, the preceding parameter amplified by two has trace zero, so the native correction also has trace one. At full ambient rank, the earlier amplified approximations instead have trace zero and the final one has trace one. These are different

statements. No amplified trace is used as a substitute for the native trace needed in character construction.

At an inductive field step $d \geq 2$, set $n = 2^{d-1}q_d$, $R = 2^{d-2}q_{d-1}$. Then $n > 2R$ and $-2R > -n - q_1$. BK Remark 2.2.7 gives the maximal field generated by the actual displayed matrix. Literal truncation and the filtered lift give the defining sequence. Nonzero native trace also proves separability: the irreducible characteristic polynomial has a nonzero coefficient of X^{N-1}, so its derivative is nonzero. See `tower/DYADIC_AS_TOWER167.md` for all field and filtration details.

The replacement for trace zero is the depth inequality

$$\boxed{v_F(\operatorname{tr}\delta) + s \geq 1.}$$

It makes the additive correction trivial on scalar U_F^s, because $(u-1)\operatorname{tr}\delta \in t\mathcal{O}_F$. Our native corrections have trace one, so the inequality holds at $s = 1$. This is the exact source-to-target transfer of the weighted-visibility principle. The Euler coefficient theorem is motivation, not an input to the arithmetic proof. In characteristic two, the weights are uniformizer depths, rather than integer powers of two in the coefficient field.

G.3 3. Full characters, including the odd critical endpoint

At the minimal rank-two base, extend the forced character on $U_E^{(q+1)/2}$ to U_E^1, imposing triviality on U_F^1. The prescriptions agree on their intersection because the parameter trace is one and $\psi_F(u-1) = 1$. This is an actual finite abelian character extension and gives the full minimal simple character.

For a later native step, first transfer the full beta-determinant unit character to multiplicity two. BK 3.6.13(ii) and 3.6.14 give, on scalar units,

$$\Theta_j(uI) = \theta_j(u^2) = 1.$$

For odd critical level r, use $m = \lfloor r/2 \rfloor$ and $D = H^{m+1}(\beta) = H^{m+1}(\gamma)$. The endpoint is valid because $2(m+1) - r = 1$: the cross term for the additive character lies in the radical on which the matrix additive character is trivial. BK 3.3.18 supplies $\vartheta = \Theta_j|_D\, \psi_{\beta-\gamma}$. Integral trace makes it trivial on $D \cap U_F^1$.

The next parameter field is maximal. Extend the forced character on its deep units together with the trivial character of U_F^1, then apply BK 3.2.5 to obtain the next full character. All new extensions occur at maximal native rank. No unverified determinant-image extension in a nonmaximal centralizer is used.

For the final totally ramified field, $J/J^1 = \mathbb{F}_2^\times = 1$, so $J = J^1$ and the canonical beta extension is η. Extend to $K^\times J$, and choose the noncompact scalar to match $\omega(t)$. The needed N-th root exists in $\mathbb{C}^\times$. The compact type stays fixed. All choices precede H5. See `central/DYADIC_FULL_CHARACTER167.md` for the complete proof.

G.4 4. Exact packet capacity and marked attachment

The dyadic floors cannot be replaced by the odd-prime dimension formula. At stage j, the centralizer grade has dimension 2^{d-j}. Its contribution to J^1/H^1 is

$$\left(2^{d-j+1} - 2^{d-j}\right)\mathbf{1}_{n_j \text{ even}}.$$

Every n_j for $j < d$ is even, whereas $n_d = q_1$ is odd and contributes zero. The sum is exactly $N-2$. BK nondegeneracy makes this a symplectic space. A whole source packet of d planes fits if and only if $N - 2 \geq 2d$, equivalently $d \geq 3$.

Ambient rank	Host quotient dimension	Host Heisenberg carrier	Required source carrier	Outcome
2	0	1	2	Insufficient
4	2	2	4	Insufficient
8	6	8	8	Exact capacity
16	14	128	16	Spare capacity

Now assume $d \geq 3$, the one-vertex contract, declared $C = \{1\}$, and actually supplied full character extensions with labeled whole-plane quotients. Retain their complete dyadic finite central-extension markings, including their specified central embeddings, lifts, square values, refinements and phases. Any central-character kernel already factored out must be recorded. A scalar pushout cannot recover invisible kernel elements.

Under the scalar-pushout realization contract, with no independently fixed conflicting host marking, choose a labeled symplectic embedding into W. Matching commutators gives a center-fixing isomorphism between the source $\mathbb{C}^\times$ extension and the corresponding host restriction. **Retain the image of the whole marked finite subgroup and every lift.** For this isomorphism F_0,

$$F_0(a)^2 = F_0(a^2).$$

Thus its square bits, refinement and Gauss sign survive, along with all supplied full character values. They are not inferred from the unmarked pushout. Stone–von Neumann gives the source factor inside η. A complementary line gives an actual rank-N image-algebra projector. The retained induction/recovery idempotent recovers the same Λ and the same π.

This gives an actual represented marked finite subgroup in the scalar extension of the actual type. It permits complex scalars times type operators, as the declared contract does. It does not assert literal inclusion of old matrix groups, numerical depth placement, or identification with a separately fixed canonical native finite subgroup.

If a genuine host-native quadratic refinement and its finite μ_2-central extension are independently supplied, quadratic embedding has a sharper condition. With otherwise free embeddings, at rank eight the equal-dimensional source and host must have equal Arf invariants. For $d \geq 4$, the host has at least one spare symplectic plane, so either source Arf type embeds. If the original embedding is preassigned, it must instead satisfy the pointwise equation $q_W(f(v)) = q_{\mathrm{src}}(v)$. These conclusions require the actual host marking as input. Dimension does not manufacture it.

The full finite proofs are in `markings/DYADIC_MARKINGS167.md`. In particular, the D8 and Q8 unmarked complex pushouts coincide while their whole marked finite μ_2-central extensions differ. Merely recording the central $\{\pm 1\}$ subgroup would lose that distinction.

G.5 5. Failed repairs and boundaries

The rank-two and rank-four carrier failures cannot be repaired by changing the maximal simple parameter while keeping the specified Swan exponent and torsion number. At rank two, odd Swan forces a minimal odd-level stratum and $J^1 = H^1$. At rank four, Swan $6h_2$ with h_2 odd forces the degree pattern $4 \to 2$. The maximal quadratic derived field forces an odd lower critical level. Every such maximal-type carrier has dimension two, below the required four. Scalar twists preserve this capacity.

This is an obstruction to the whole-source sector inside that maximal-type carrier. It is not a counterexample to the original theorem or a proof that a different arithmetic carrier in the same representation cannot work. Section 7 constructs such carriers and proves their exact recovery.

At rank eight, numerical profiles alone also fail to determine capacity. `capacity/DYADIC_CAPACITY167.md`, Section 6, constructs an actual skipped-degree $8 \to 2$ host with the same Swan, torsion and self-pair conductor but quotient dimension four rather than six. Our successful result uses the explicit full tower, rather than inferring its profile from invariants.

Thus the completed dyadic scope is:

- Actual fields, full characters and central completion in every rank 2^d.
- Full marked scalar-pushout packet realization for the stated family when $d \geq 3$, with $C = 1$, one vertex, and transported markings retained.
- A conditional quadratic embedding into an independently supplied native marked host, with the exact Arf or pointwise condition above.

Section 7 constructs different actual arithmetic carriers at the two low ranks and recovers the same candidate from them. The maximal-type capacity failure above remains true. Original H5 stays the original same-candidate conditional premise. These conductor identities and noncompact root choices do not prove it.

G.6 6. Evidence

The arbitrary-length field and full-character arguments are ordinary proofs. The exact rank-eight run for $h = (1, 3, 5)$ checks actual matrix identities, normalized lifts, leading and adjacent-correction grades $-60, -18, -3$, native traces one and ambient traces $0, 0, 1$. It outputs a matrix with 25 nonzero Laurent polynomial entries. The corresponding critical exponents are proved symbolically. The run does not prove simplicity, full-character existence, or universal marked realization. Those conclusions use the separate proofs and their stated scope. No Lean build was run.

The base synthesis before the low-rank continuation is preserved in `history/PROOF167_BEFORE_ENLARGEMENT.md`, together with its review. Its previously open enlarged-carrier route is executed next.

G.7 7. The low-rank repair: enlarge the actual inducing carrier

The maximal-type carrier is not the only arithmetic carrier available in the constructed representation. The following constructions keep the original β, θ, Λ and π fixed.

G.7.1 Rank two

Write the odd minimal level as $q = 2m + 1$, with $m \geq 1$, and let $J = H^1 = J^1 = U_K^1 U_A^{m+1}$. Define

$$P = U_K^1 U_A^m.$$

This is an actual compact open group, with $J \triangleleft P$, $[P : J] = 2$ and $P \cap (K^\times J) = J$. For $g \in P \setminus J$, the actual intertwining theorem makes $\lambda = \theta^g/\theta$ nontrivial. The commutator $[P, J] \subset U_A^{m+1}$ lies in the forced domain of the additive character, so λ has values $\{\pm 1\}$. Put $H_0 = \ker \lambda$. Then

$$P/H_0 \simeq \mathbb{F}_2^2, \qquad E = \mathrm{Ind}_J^P \theta \text{ is irreducible of dimension 2.}$$

The actual represented generators are a diagonal sign and an off-diagonal matrix that anticommute. Thus their commutator form is nondegenerate, and $\theta|_{H_0}$ is the full character of an actual new arithmetic Heisenberg pair. No abstract operators create that group action.

The identity compact-induction double coset realizes E inside the same π, as functions supported on $(K^\times J)P$. The complete group and recovery proof is `lowrank/RANK2_ENLARGED_CARRIER167.md`.

G.7.2 Rank four

Use the actual rank-four outer approximation field E_0/F of degree two and its centralizer $B = \mathrm{End}_{E_0}(F^4)$. The odd lower critical level is $r = 3h_1 = 2m+1$. The half leading level is $a = 3h_2 > r$. Define

$$P = J\,U_B^m,$$

where $J = J^1$ is the actual compact type group. The filtered centralizer estimate for the actual final field, without assuming $E_0 \subset K$, proves that P is a compact group with J normal of index two. For a concrete new coset representative u, $\lambda = \theta^u/\theta$ on the actual $H = H^1$ is nontrivial with values $\{\pm 1\}$. Put $H_0 = \ker\lambda \subset H$. The calculation proves

$$P/H_0 \simeq \mathbb{F}_2^4, \qquad E = \mathrm{Ind}_J^P \eta \text{ is irreducible of dimension 4.}$$

The new represented commutators are scalar, its quotient has exponent two, and nondegeneracy follows from its exact scalar preimage and irreducibility. The finite scalar center may be larger than μ_2. A literal extraspecial quotient is not asserted. The full character values are retained by the actual finite Heisenberg extension. Again E occurs inside the same π through the identity double coset. See `lowrank/RANK4_ENLARGED_CARRIER167.md`.

G.7.3 Exact recovery, with the noncompact normalization retained

At both low ranks, $K^\times$ normalizes the new P. In rank two this follows directly from normalization of A and its radical. In rank four, a K-uniformizer is approximated by the derived chain shift R times a unit of B, with relative error in P_A^{n-r}. Its conjugation error on U_B^m lies in $U_A^{n-r+m} \subset J$, which proves normalization. The exact argument is `lowrank/RANK4_INDUCTION_BY_STAGES167.md`.

Consequently $\mathcal{M} = K^\times P$ is an actual open group compact modulo the center, $\mathcal{N} = K^\times J$ is normal of index two in $\mathcal{M}$, and

$$\Sigma = \mathrm{Ind}_{\mathcal{N}}^{\mathcal{M}} \Lambda, \qquad \Sigma|_P \simeq E, \qquad \boxed{\pi \simeq \mathfrak{c}\mathrm{Ind}_{\mathcal{M}}^G \Sigma.}$$

Restriction to P is irreducible, so Σ is irreducible. Induction in stages proves the last equality for the original Λ, including its fixed central character and noncompact root. This is an actual enlarged inducing representation. It changes neither the candidate nor its original canonical beta-extension data.

There is also a recovery from compact data explicitly recording that root. With $A_0 = \Lambda(\varpi_K)$ and $X = \mathfrak{c}\mathrm{Ind}_J^G \eta$, define $(Tf)(g) = A_0^{-1} f(\varpi_K g)$. Then $X/(T-1)X \simeq \pi$. After imposing the fixed central character, $\mathcal{N}/(F^\times J)$ is cyclic of order N, and the finite average $N^{-1}\sum_{i=0}^{N-1} T^i$ selects this same extension. These operators act on complex representation spaces, so their scalar division by N is valid. This is a different recovery from the Block159 projector onto a sector of η. The exact kernel and finite projector are proved in `lowrank/ENLARGED_CARRIER_CONTRACT167.md`.

G.7.4 What the low-rank repair admits

Apply the marked scalar-pushout theorem of Section 4 to the **actual new** pair (H_0, P) and its carrier E. It has exactly the capacity of the whole source at each low rank. All full source character values and the transported whole finite subgroup, lifts, squares and phase markings can be attached and retained. The displayed Σ and induction in stages recover the same candidate.

The old H2 break targets remain their supplied actual simple-character quotients. The enlarged pair is their common arithmetic assembly carrier. It is not renamed as the original maximal-simple H^1, J^1 pair. If a stronger reading requires the whole packet itself to be the quotient of a maximal simple character in the same candidate, the earlier dimension obstruction still applies. Likewise, agreement

with an independently fixed native quadratic marking must be checked separately. The contract note proves this distinction explicitly.

G.8 8. Final scope and next construction question

The weighted dyadic mechanism now gives actual arithmetic fields, full characters and tame central completion in every rank 2^d of the stated increasing odd-break family. Under the one-vertex, $C = 1$, full marked scalar-pushout **assembly** contract, the packet carrier can be realized at every length: inside the original type for $d \geq 3$, and in the explicit enlarged inducing carrier at $d = 1, 2$. The source's prescribed central character must belong to the covered tame class. In particular no arbitrary single-factor wild determinant is assumed to be trivial.

Remaining universal obligations are construction/comparison of independently prescribed canonical native markings, nontrivial symplectic stabilizer actions, general source/transport diagrams and any required numerical layer placement. Original H5 remains conditional for this same fixed candidate. No equality of profiles or choice among noncompact roots is promoted to H5.

The next concrete target is equivariance of the arithmetic attachment for a nontrivial prescribed symplectic action. Another matrix search inside the old low-rank maximal-type carrier would repeat its already proved obstruction. `STATUS167.json` and `ROUTES167.md` record the achieved scopes and open routes.

H The explicit dyadic arithmetic tower

No character, quotient-dimension, source attachment, pair-factor, or universal Chapter XI closure is asserted here.

H.1 1. Statement

Let F=F_2((t)), let 0<h_1<…<h_d be odd positive integers, and set

```
q_i=3h_i,  N=2^d.
```

The explicit recursion below constructs B_d in M_N(F) such that

```
F[B_d] is a totally ramified maximal field normalizing
   the complete period-N lattice chain,
v_A(B_d)=-2^(d-1)q_d,
k_0(B_d,A)=-q_1,
tr_F(B_d)=1.
```

There are explicit native constructors of ranks 2^j for the largest j exponents. Their identity-amplifications give the actual defining sequence with field degrees 2,4,…,2^d and critical levels

```
n_j=2^(d-j)q_(d-j+1),  1<=j<=d.
```

Every parameter in this ambient defining sequence has the same leading level n_1. The quantities n_2,…,n_d are its successive critical approximation levels, not new leading levels of the ambient matrices.

Every native constructor has trace 1. Every correction from an identity-2 amplification of the preceding native constructor to the next native constructor also has trace 1. In full ambient rank 2^d, the earlier amplified approximations instead have trace 0, because their multiplicities are even. This distinction is essential.

H.2 2. The minimal degree-two building block

Let L be a complete discretely valued field of characteristic 2 with residue field F_2, and choose a uniformizer t_L. For odd q>0 put a=(q+1)/2 and define

```
E=L(alpha),  alpha^2-alpha=t_L^(-q),
z=t_L^a alpha.
```

The Artin–Schreier polynomial is irreducible: an element x^2-x of negative L-valuation has even negative valuation. Its derivative is 1, and the valuation equation forces ramification index 2. Thus E/L is separable and totally ramified, v_E(alpha)=-q, and alpha is minimal.

The element z has E-valuation 2a-q=1 and satisfies

```
f(z)=z^2+t_L^a z+t_L=0.
```

Thus 1,z is a uniformizer power basis, and

```
alpha=z/t_L^a,  Tr_(E/L)(alpha)=1.
```

H.3 3. An explicit normalized corestriction and lift

On E use the coefficient basis 1,z. Its trace-dual elements are

```
z_0^*=(z+t_L^a)/t_L^a,
z_1^*=1/t_L^a.
```

Let pr(T_ij)=z^i z_j^* be the trace corestriction. Put

```
kappa=t_L^a/z=alpha^(-1),
s=kappa pr,
u_0=s(P0)=t_L/z^2.
```

Here P0 is constant-coefficient projection, applied separately to each coordinate when acting on E^m. The identity for u_0 follows from f(z)=0. It is a unit: v_E(t_L)=v_E(z^2)=2. More explicitly,

```
s(T_i0)=z^i(1+t_L^a/z),
s(T_i1)=z^(i-1).
```

Since v_E(t_L^a/z)=2a-1=q>0, these preserve the expected leading grades. On End_L(E^m), take the complete interleaved chain in the basis w_(mi+b)=z^i v_b. Its centralizer is M_m(E) with its complete E-chain. The formulas give s(P^r) contained in Q^r for all integers r.

For C in M_m(E), define the explicit L-linear operator

```
Lift_E(C)=u_0^(-1) C P0=(z^2/t_L) C P0.
```

B-bimodularity gives s(Lift_E(C))=C. The lift preserves exact valuation: u_0^(-1) is an E-unit, and expansion in 1,z gives coefficient valuations distinct modulo 2. Each coefficient entry has the same grade in the interleaved L-chain and the E-chain. In particular $s(P^r) = Q^r$.

H.4 4. Exact trace transport

For every C in M_m(E), ordinary matrix traces satisfy

```
tr_L(Lift_E(C))
   =Tr_(E/L)(z_0^* u_0^(-1) tr_E(C))
   =Tr_(E/L)(kappa^(-1) tr_E(C))
   =Tr_(E/L)(alpha tr_E(C)).                (T2)
```

The first equality follows by taking the L-trace of each rank-one operator yP0, which is the constant coefficient of y. This coefficient is Tr_(E/L)(z_0^* y).

Thus the trace of a lift is 1 whenever the inner ordinary E-matrix trace is 1. It is not asserted that arbitrary trace values are unchanged.

H.5 5. Recursive actual matrices and fields

Define B(L,t_L;h_1,…,h_d) as follows. For d=1 use the multiplication matrix of alpha from Section 2 with q=q_1. For d>=2 apply Section 2 with q=q_d, set m=2^(d-1), recursively construct

```
C=B(E,z;h_1,...,h_(d-1)),
```

and put

```
B(L,t_L;h_1,...,h_d)=alpha I_m+(z^2/t_L) C P0.       (D2)
```

This is a finite algebraic recipe. It uses specified separable quadratic extensions and no unknown support, genericity choice, or correspondence inverse.

The degree and simplicity proof is by induction. The base alpha is minimal and generates E. At a later step let gamma=alpha I_m and b=Lift_E(C). The outer minimal stratum has

```
n=2^(d-1)q_d,  k_0(gamma,A)=-n.
```

The inner induction hypothesis gives an actual maximal field E[C] normalizing the complete E-chain with

```
R=2^(d-2)q_(d-1),  v(C)=-R,  k_0(C)=-q_1.
```

Because h_d>h_(d-1), we have n>2R. Therefore

```
-2R > -n-q_1 = k_0(gamma,A)+k_0(C).
```

The derived stratum with parameter s(b)=C is pure, and its field is maximal. Bushnell–Kutzko Remark 2.2.7 consequently gives the field generated by the actual gamma+b, not merely an equivalent replacement. It has ramification and degree 2^d, residue degree one, normalizes the complete chain, and has k_0=max(-n,-q_1)=-q_1. Since v(b)>v(gamma), its level is n and [A,n,0,gamma+b] is simple.

The derived model E[C] is not identified with L[gamma+b]. In particular no literal embedding of E into the latter field is claimed.

H.6 6. Trace 1 at every native stage

The base matrix has trace Tr_(E/L)(alpha)=1. Inductively tr_E(C)=1, so (T2) gives tr_L(Lift_E(C))=1. For d>=2 the L-trace of alpha I_m is

```
m Tr_(E/L)(alpha)=0,
```

because m is even. Hence tr_L(B)=1 at every native rank.

For any j>=1, let A_j denote the native rank-2^j constructor using a fixed list of the largest j exponents. The next native rank-2^(j+1) constructor A_(j+1) uses the same largest exponents and adds the next smaller one. With their compatible interleaved bases, put

```
Delta_(j+1)=A_(j+1)-A_j tensor I_2.
```

Then

```
tr(Delta_(j+1))=1-2*1=1                    in characteristic 2.
```

This native trace identity is stronger and more relevant than a claim obtained merely by amplifying to a rank divisible by two.

H.7 7. Full arithmetic defining sequence

For the final rank 2^d define

```
B^[j]=B(F,t;h_(d-j+1),...,h_d) tensor I_(2^(d-j)).
```

All truncations use the same top q_d to select the first extension and uniformizer, the same next q_(d-1) to select the next extension, and so on. The coefficient projection and lift commute with amplification. Thus these are literal truncations of (D2).

Their parameter fields have degree 2^j, and their critical levels in the ambient complete chain are n_j=2^(d-j)q_(d-j+1). Valuation-preserving lifts give

```
v_A(B^[j+1]-B^[j])=-n_(j+1).
```

Later differences have strictly greater valuation, since n_j>2n_(j+1). This proves that the displayed parameters form an actual defining sequence of simple approximations.

Their ambient traces are

```
tr_F(B^[j])=0 for j<d,  tr_F(B^[d])=1.
```

Consequently ambient corrections before the last step have trace 0, and the last correction has trace 1. Every one of these trace values is integral. On a scalar principal unit u, the additive correction character has value psi_F((u-1)tr(delta)). This is 1 for an additive character trivial on the maximal ideal. No assertion about full simple character extension follows here without its separate intersection and transfer argument.

H.8 8. Concrete rank-eight instance and limits

For h=(1,3,5), the construction gives actual approximation-field degrees 2,4,8 and critical levels 60,18,3. The supplementary exact program constructs its Laurent polynomial matrices, verifies the normalized lift identities, and checks the native/ambient trace distinctions and matrix grades. It does not prove field existence or simplicity by bounded computation: those are consequences of Section 5.

The odd-prime Heisenberg quotient and conductor counting from Block166 cannot simply be copied: q_1 is odd here. This note makes no dyadic quotient, full-character, stabilizer, phase, source-attachment, or H5 claim. It also changes lower parameter conjugacy classes, as in the corresponding odd-prime Artin–Schreier construction.

H.9 Dependencies and provenance

Primary theorem: Bushnell–Kutzko, The admissible dual of GL(N) via compact open subgroups, Remark 2.2.7 (printed p. 55), together with the amplification normalization and defining-sequence conventions already checked in Block166.

Retained primary source text SHA256: `d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8`.

Block166 inputs are pinned separately in `INPUT_PINS167.json` and are not modified by this continuation. The Artin–Schreier, trace-dual, and trace calculations needed specifically for p=2 are proved above.

I Dyadic full characters and the odd critical endpoint

Source record: Paper_XI_Dyadic167. Dependency: Paper_XI_Attachment166.

Evidence: ordinary mathematical proof using the actual Artin–Schreier matrix tower and the retained Bushnell–Kutzko character and transfer theorems. No Lean verification is claimed. This note proves no dyadic quotient-capacity, quadratic-phase, packet-attachment, or H5 result.

I.1 1. Exact theorem

Let F=F_2((t)), let psi_F be an additive character of conductor t O_F (trivial there and nontrivial on O_F), and let

```
0<h_1<...<h_d,       each h_i odd,
q_i=3h_i,           N=2^d.
```

Use the dyadic Artin–Schreier recursion specified in Section 2. It gives the actual maximal simple parameter beta_d with its native arithmetic defining sequence. There is an actual full simple character theta of its maximal simple stratum such that

```
theta restricted to U_F^1 is trivial.                    (1)
```

The construction retains the actual canonical beta extension and can be completed with any prescribed central quasicharacter omega_F trivial on U_F^1. Because $k_F^\times = \mathbb{F}_2^\times$ is trivial, such an omega_F is unramified. Thus this covers every tamely ramified central quasicharacter for this particular residue field.

The corresponding compact induction is an actual irreducible admissible supercuspidal of rank N, by the retained maximal-simple-type theorem. No claim is made here about its Heisenberg quotient size or about realization of a supplied dyadic finite packet.

The trace-zero argument from Block 166 does not apply verbatim: at p=2 the primitive Artin–Schreier parameter has trace one. The correct invariant is **integral trace**, not necessarily zero trace. Since psi_F kills t O_F, integral correction trace is sufficient on scalar principal units. The recursion below preserves the stronger exact value one.

I.2 2. Actual dyadic recursion and exact native traces

At a recursive step let L be a characteristic-two local field with residue field F_2, uniformizer t_L, and a positive odd q. Define

```
E=L(alpha),       alpha^2-alpha=t_L^(-q).
```

The right side is not an Artin–Schreier coboundary: a negative valuation of x^2-x is even, whereas -q is odd. The degree-two polynomial is therefore irreducible and separable. Its valuation equation gives total ramification and v_E(alpha)=-q. Put

```
a=(q+1)/2,       z=t_L^a alpha.
```

Then z is an actual uniformizer and its Eisenstein polynomial is

```
f(Z)=Z^2+t_L^a Z+t_L.                                    (2)
```

Its two conjugate alpha values are alpha and alpha+1, so

```
Tr_(E/L)(alpha)=1.                                       (3)
```

Use the coefficient basis 1,z, the trace corestriction pr, and the same normalized corestriction convention as in Block 166:

```
kappa=-f'(z)/z = t_L^a/z = alpha^(-1),
s=kappa pr.
```

Minus and plus agree in characteristic two. If P0 extracts the constant coefficient on every E-coordinate, then

```
u_0=s(P0)=t_L/z^2 is a unit,
Lift_E(C)=u_0^(-1) C P0,
s(Lift_E(C))=C.
```

The trace-dual calculation in Block 166 gives, as an identity of L-matrix traces for every matrix C over E,

```
tr_L Lift_E(C)
   =Tr_(E/L)(kappa^(-1) tr_E(C))
   =Tr_(E/L)(alpha tr_E(C)).                                  (4)
```

The lift preserves the complete-chain filtration valuation, by expansion in 1,z. Thus it is the actual filtered right inverse used in the field constructor, not an arbitrary inverse on vector spaces.

Define B(L,t_L;h_1,…,h_s) recursively. At s=1 use the L-matrix of multiplication by alpha with q=q_1. At s>=2 use (2) with q=q_s, m=2^(s-1), and

```
C=B(E,z;h_1,...,h_(s-1)),
B(L,t_L;h_1,...,h_s)=alpha I_m+Lift_E(C).                     (5)
```

The base trace is one by (3). If tr_E(C)=1, then (4) gives lift trace one. Since m is even, the trace of alpha I_m as an L-matrix is m Tr_(E/L)(alpha)=0. Consequently

```
tr_L B(L,t_L;h_1,...,h_s)=1                                   (6)
```

for every native recursion, by induction. There is no division by the rank and no confusion between a native trace and its amplified trace.

The actual field proof from Block 166 applies with this degree-two extension and normalized lift. At a nontrivial step its hypotheses are n=2^(s-1)q_s, R=2^(s-2)q_(s-1), s(Lift(C))=C, and

```
n>2R,
-2R>-n-q_1.
```

The first strict inequality follows from h_s>h_(s-1). These are the conditions used in BK Remark 2.2.7. The displayed beta itself generates the totally ramified degree-2^s field normalizing the complete chain, and its final critical exponent is -q_1. This statement does not identify the final field with its auxiliary derived field.

For the final length d write the native shorter matrices as

```
b_j=B(F,t;h_(d-j+1),...,h_d),       1<=j<=d.
```

They are the compatible native truncations of (5), because the outer extension, uniformizer, and lift depend on the same retained largest h at each stage. For $1 \leq j < d$, in rank 2^(j+1), put

```
gamma_j=b_j tensor I_2,
delta_j=b_(j+1)-gamma_j.
```

The defining-sequence calculation of Block 166 gives gamma_j as a simple approximation at the critical level r_j=q_(d-j) of b_(j+1), and delta_j has grade -r_j. Equation (6) gives the exact scalar identity

```
tr_F(gamma_j)=0,
tr_F(delta_j)=1.                                              (7)
```

Thus for every scalar u in U_F^1 on which its additive correction is defined,

```
psi_(delta_j)(u I)=psi_F((u-1)tr_F(delta_j))
                  =psi_F(u-1)=1.                              (8)
```

It is (8), not a false trace-zero assertion, that drives the full character construction.

I.3 3. The odd-critical-level rounding

The critical levels r=q_i are odd. Write r=2s+1, and set

```
m=floor(r/2)=s,        m+1=(r+1)/2.
```

The relevant BK statements use the floor, with no assumption that r or the residue characteristic is odd or even:

- BK 3.1.9(ii) identifies H^k(beta) and H^k(gamma) for k>=floor(r/2)+1.
- BK 3.3.18 gives the correction bijection at floor(r/2)<=m<=r-1.
- BK 3.2.5 extends the resulting character at floor(r/2) to a full character, its fibers being the indicated determinant characters.

At our exact endpoint m=s, the correction character is legitimate. For x,y in the radical power P^(s+1),

```
delta*x*y belongs to P^(2(s+1)-r)=P^1.
```

The level-one additive matrix character is trivial on P^1, so the multiplicative cross term contributes one. Therefore psi_delta is a character on U^(s+1), and hence on the recursive subgroup used below. This checks the borderline odd-level rounding directly. Replacing m by r/2 as if it were an integer would be incorrect.

The same check applies at the native minimal base level q, which is odd: its forced additive character begins at floor(q/2)+1=(q+1)/2, and 2*(q+1)/2-q=1.

I.4 4. Finite character extension at maximal rank

We reuse the elementary extension procedure with its compatibility condition explicit. If K/F is a finite totally ramified field, and chi is a smooth character of U_K^a trivial on U_K^a intersect U_F^1, define a character on U_K^a U_F^1 by

```
chi_0(ku)=chi(k).
```

It is well defined by the intersection condition. Choose M>=a with chi|U_K^M=1. Then chi_0 is a character of a subgroup of the finite abelian group U_K^1/U_K^M. Extend it by adjoining generators and choosing the required complex roots, and pull it back. The resulting chi_K extends chi and is trivial on U_F^1.

All freely chosen character extensions below take place at maximal native rank, where the centralizer is K itself. We do not apply this scalar-intersection argument directly inside a nonmaximal matrix centralizer, which would require a separate determinant-image check.

I.5 5. Base full character with trace one

At native rank two, let E=F[alpha], with q=q_d and a=floor(q/2)+1. The forced character on U_E^a is

```
phi(v)=psi_F(Tr_(E/F)(alpha(v-1))).
```

For scalar u in U_E^a intersect U_F^1, equation (3) gives

```
phi(u)=psi_F((u-1)Tr_(E/F)(alpha))=psi_F(u-1)=1.
```

Section 4 constructs chi_E on U_E^1 extending phi and trivial on U_F^1. Define theta_1 on $H^1 = U_E^1$ U_(A_1)^a by chi_E on the first factor and psi_alpha on the second. The restrictions agree on the intersection. BK 3.2.1 gives the actual full simple character, with theta_1 | U_F^1=1.

This uses the true trace value one and only the stated conductor condition on psi_F. It does not require an artificial zero-trace primitive parameter in a quadratic extension.

I.6 6. Native-rank induction via full beta-determinant transfer

Assume theta_j is a full simple character for native b_j in rank 2^j, with theta_j | U_F^1=1. Let K_j=F[b_j]. Since K_j is maximal, the character chi_j=theta_j | U_(K_j)^1 is its full beta-determinant character.

Transfer theta_j to the multiplicity-two realization gamma_j=b_j tensor I_2 in rank 2^(j+1), using BK 3.6.14. Denote the full transferred character by Theta_j. BK 3.6.13(ii) preserves the entire beta-determinant character on U_(K_j)^1 at level m0=0. Hence for every literal scalar u in U_F^1,

```
Theta_j(u I)=chi_j(det_(K_j)(u I_2))
           =chi_j(u^2)=theta_j(u^2)=1.                    (9)
```

This statement includes nonminimal b_j. The determinant exponent two is essential and has not been suppressed.

Now set beta=b_(j+1), gamma=gamma_j, delta=delta_j, r=r_j, m=floor(r/2), and

```
D=H^(m+1)(beta)=H^(m+1)(gamma).
```

The equality and endpoint m were verified in Section 3. BK 3.3.18 constructs the actual recursive simple character

```
vartheta=(Theta_j restricted to D)*psi_delta.
```

For u in D intersect U_F^1, equations (8) and (9) prove

```
vartheta(u)=1.                                            (10)
```

The new field K=F[beta] is maximal and totally ramified in native rank 2^(j+1). Its chain has K-period one, so

```
D intersect U_K^1=U_K^(m+1),
D intersect U_F^1=U_F^ceil((m+1)/2^(j+1)).
```

Thus (10) is exactly the compatibility required by Section 4. Extend vartheta | U_K^(m+1) to chi_K on U_K^1 trivial on U_F^1. BK 3.2.5 constructs the full simple character by

```
theta_(j+1)(kd)=chi_K(k)*vartheta(d),
k in U_K^1, d in D.
```

That proposition supplies normalization and well-definedness. Its proof is an argument extending characters and imposes no exclusion of residue characteristic two. The new theta_(j+1) is trivial on U_F^1. Induction proves (1) in every native rank 2^d.

More generally the character argument needs only integral base field trace and integral traces for the native corrections. The dyadic tower establishes their exact value one, while the odd-prime tower of Block 166 establishes zero.

I.7 7. Canonical beta extension and prescribed tame center

Let K/F be the final totally ramified degree-N parameter field and let eta be the Heisenberg representation for the constructed theta. The compact type group is

```
J=O_K^times J^1.
```

Here $\mathcal{O}_K^\times = U_K^1$ because the residue field is F_2. Since U_K^1 is contained in J^1, we have

```
J=J^1,       J/J^1=F_2^times={1}.                    (11)
```

Thus the actual canonical beta extension on J equals eta. There is no residue-character choice and no change of that extension. It acts trivially on the central U_F^1 by (1). Equation (11) does not identify any externally prescribed dyadic quadratic phases.

Let omega_F be any smooth quasicharacter trivial on U_F^1. Since $\mathcal{O}_F^\times = U_F^1$, it is determined by its value omega_F(t). The actual maximal type extends to K^times J. Concretely, (K^times J)/J is infinite cyclic, generated by a K-uniformizer. An intertwiner for its conjugation action on eta gives an extension Lambda_0. Denote its central character by omega_0.

Choose a complex scalar s with

```
s^N=omega_F(t)/omega_0(t).
```

Multiply Lambda_0 by the character of K^times J trivial on J and taking the K-uniformizer to s. Because t has valuation N in K, this changes its scalar by s^N and gives the requested central quasicharacter on all F^times. The value at every element of J, including the canonical beta extension, is unchanged. This does not assume that a chosen K-uniformizer raised to N equals t.

The retained maximal-simple-type theorem now gives the actual irreducible admissible supercuspidal pi=c-Ind_(K^times J)^GL_N(F) Lambda. The noncompact scalar is fixed before any H5 comparison. An incompatible immutably prescribed extension is not overwritten. As in Block 166, its value must be free or already compatible.

I.8 8. What this closes and what it does not

This closes the full-character and actual tame-central-character construction for the explicit dyadic arithmetic tower. The failure of trace zero at p=2 is repaired by the correct integral trace invariant. Both the odd-level floor and the multiplicity-two determinant transfer are accounted for explicitly.

This note does not compute J^1/H^1 or prove that it contains a particular dyadic source packet. It does not establish a quadratic refinement, canonical phase matching, nontrivial stabilizer action, general transport graph, source admission, or H5. In particular, an actual supercuspidal host with the required central character is not asserted to be a completed realization of the original dyadic packet without those separate proofs.

I.9 9. Exact source checks and input pins

The following final Block 166 files were read and reused without editing:

- `attachment166/central/TRACE_ZERO_CHARACTER_INDUCTION166.md` SHA256 `99cb6e80c419e0bf7325e2b2b3fbaba1391903448a3b6a0c4cd9f3944ed406a1`.
- `attachment166/tower/TRACE_ZERO_AS_TOWER166.md` SHA256 `49a18fb733f0209b3e38118f5132839a060961edd0bdc0211e1f7b441a4e5653`.
- `attachment166/tower/RECURSIVE_TOWER166.md` SHA256 `6f39a077df18eb03f9fab2b6e426529dbe14cf4f7e1cfce95ddb0fcd187d7b33`.
- `attachment166/audit/TOWER_CENTRAL_REVIEW166.md` SHA256 `4bc71dd050ddb321a253105c8a4ee7a86185af255532d5605caa7a9de383e26b`.

Primary retained source: Bushnell–Kutzko, *The admissible dual of GL(N) via compact open subgroups*, text SHA256 `d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8`. The current dyadic check reread Definitions 3.1.7–8 and Proposition 3.1.9(ii) (printed pp. 90–91), Definition 3.2.1 and Proposition 3.2.5 (printed pp. 98–100), and Corollary 3.3.18 (printed pp. 112–113). Their level

thresholds are floor(r/2)+1 or floor(r/2), as used above, and their statements do not exclude p=2. The transfer's preservation of the full beta-determinant character is BK 3.6.13(ii), with existence Theorem 3.6.14 (printed pp. 136–137), already directly checked in Block 166 and reread for this application.

The character-extension and correction proofs use neither division by two nor an odd-order averaging argument. The exact dyadic trace calculation and the borderline correction-character check are given in Sections 2–3. No new publication lookup, external arithmetic-induction/descent constructor, or inverse LLC is used.

J Exact dyadic capacity and numerical-profile obstruction

Status: ordinary proof from the supplied actual defining sequence and the retained primary Bushnell–Kutzko results. No Lean or numerical computation is used. This note calculates the quotient of an actual simple type. It does not prove universal H5 or identify an explicitly constructed host with the source-defined arithmetic candidate.

J.1 1. Exact hypotheses and result

Let F=F_2((t)). Fix d>=1 and increasing positive odd integers

```
h_1<...<h_d,        q_i=3h_i,        n=2^d.
```

Suppose the actual arithmetic constructor supplies a full-degree totally ramified parameter beta on the complete period-n chain, with actual approximating fields of degrees

```
2,4,...,2^d
```

and ambient critical levels

```
nu_j=2^(d-j) q_(d-j+1),        1<=j<=d.
```

These are the supplied dyadic specialization of the recursive tower's field/profile conclusions. The calculation applies equally to the original polynomial tower of Block166 and to an Artin–Schreier variant once that variant's field/profile conclusions are established. No trace-zero assertion is needed here.

The exact actual Heisenberg quotient is

```
dim_F2(J^1/H^1)=2^d-2,
dim eta=2^((2^d-2)/2).
```

Consequently its symplectic dimension is at least the 2d dimensions of the whole d-plane source packet exactly when d>=3. In rank 8, d=3, the quotient has dimension 6 and eta has dimension 8: the source carrier fits exactly, with no complementary Heisenberg factor needed.

d	Ambient rank	Actual quotient dimension	eta dimension	Source carrier dimension	Capacity
1	2	0	1	2	Fails
2	4	2	2	4	Fails
3	8	6	8	8	Exact
4	16	14	128	16	Sufficient
general	2^d	2^d−2	$2^{2^{d-1}-1}$	2^d	Sufficient iff d>=3

This improves the sufficient threshold obtained from the leading layer alone d>=4 to the exact threshold obtained from the whole profile d>=3. It also exposes a genuine low-rank limitation of this proposed whole-carrier realization.

J.2 2. Floors and ceilings in the actual BK orders

Write P=rad(A), and let B_j be the A-centralizer of the degree-2^j approximation. Its complete-chain grade has F_2-dimension

```
b_j=2^(d-j),        b_0=n.
```

Put

```
a_j=ceil(nu_j/2),       t_j=floor(nu_j/2)+1.
```

The exact BK definitions, 3.1.7–8, use a_j as the J cutoff and t_j as the H cutoff. Thus a_j=t_j for an odd critical level, while t_j=a_j+1 for an even critical level.

The increasing h_i give

```
nu_j>2 nu_(j+1),       1<=j<d.
```

This strict separation is useful: the centralizer approximation estimate from BK 3.1.5 puts each newly introduced centralizer into the preceding one modulo a term above the preceding half-level. In the notation of these stages, the relevant error level is nu_j−nu_(j+1)>nu_j/2. Therefore the successive grade dimensions are b_j. There is no unproved assumption that the approximation fields are literally nested inside the final field.

Applying the ring recursion gives the J-grade staircase:

- below a_d the grade dimension is b_d=1;
- on a_(j+1)<=k<a_j it is b_j;
- from a_1 upward it is b_0=n.

The H staircase uses the corresponding t_j. Hence the exact contribution of transition j to J^1/H^1 is

```
(b_(j−1)−b_j) · 1_(nu_j even).
```

One can also obtain this by subtracting the exact logarithmic indices of H and J in 1+P. It is an equality of the group index $|J^1/H^1|$, not a guess from the visible leading matrix support. BK 3.4.1 makes the resulting commutator pairing nondegenerate.

For j<d the level nu_j has a factor of 2 and is even. The final level nu_d=q_1 is odd. Therefore

```
dim_F2 J^1/H^1
  =sum_(j=1)^(d−1) (2^(d−j+1)−2^(d−j))
  =2^d−2.
```

The omitted final increment would have dimension 1. It cannot be treated as a symplectic plane or silently restored: at that odd level the two defining lattice cutoffs are exactly equal. Its disappearance is also consistent with the even dimension required of a nondegenerate alternating space.

The source packet requires dimension 2d. At d=1,2, 2^d−2<2d. At d=3 equality holds. If the inequality holds for d>=3, then

```
2^(d+1)−2−2(d+1) = (2^d−2−2d)+(2^d−2)>0,
```

so it continues to hold for every larger d. This proves the exact threshold.

J.3 3. Conductors still match despite the lost final contribution

This section reuses the same retained formal-degree and self-pair conductor identities already audited in Blocks 161 and 165. It is not needed to prove the dimension formula, but shows why numerical conductor checks do not expose the low-rank capacity failure.

Put Delta_j=b_(j−1)−b_j. The exact grade count gives

```
I=log_2[(1+P):J^1]
  =sum_j Delta_j ceil(nu_j/2)−(n−1).
```

For this profile only the final level is odd and Delta_d=1. Therefore

```
2I=sum_j Delta_j nu_j+1−2(n−1).
```

The Heisenberg dimension exponent is D/2, where D=n−2. The retained formal-degree identity for a full-degree, totally ramified, torsion-one maximal type gives

```
a(Pi^vee×Pi)=n^2−1+(n−1)+2I+D.
```

Substitution yields exact cancellation of the odd-cutoff correction:

```
a(Pi^vee×Pi)=n^2−1+sum_j Delta_j nu_j
          =2^(2d)−1+sum_(i=1)^d 3·4^(i−1) h_i.
```

The ordinary conductor is n+nu_1, so Swan(Pi)=2^(d−1)·3h_d, and the torsion number is one. These are the usual three numerical invariants of the stipulated independent dyadic Heisenberg tensor source.

Thus all three numerical invariants can match while eta is too small for the whole source carrier at ranks 2 and 4. This is a capacity failure, not a conductor mismatch. It does not by itself refute any original H1–H5 statement.

J.4 4. Rank 2 failure is forced by Swan and torsion alone

Let Pi be any actual supercuspidal representation of GL2(F) with torsion number one and odd Swan exponent N=3h_1. BK 6.2.5 forces its maximal parameter field K/F to be totally ramified of degree 2 and its hereditary order to have complete period 2. The ordinary conductor formula then fixes its native level as N.

Every parameter beta of a maximal simple type in Pi has v_K(beta)=−N, which is odd. Since the residue extension is trivial, BK 1.4.14–15 say beta is minimal over F. The two order definitions are therefore

```
J_order=O_K+P^ceil(N/2),
H_order=O_K+P^(floor(N/2)+1).
```

The exponents coincide because N is odd. Hence $J^1 = H^1$, eta is one-dimensional, and no two-dimensional symplectic source plane can embed into this quotient.

This proves failure for every maximal simple type of every candidate with these two invariants. It is stronger than failure of one explicit matrix formula. It remains a statement about attachment inside the actual maximal-simple-type Heisenberg carrier, not a proof that no other arithmetic realization relation could work.

J.5 5. Rank 4 failure is also forced by Swan and torsion

Let Pi be any actual supercuspidal representation of GL4(F) with torsion number one and Swan exponent N=6h_2, where h_2 is odd. Its maximal parameter field is totally ramified of degree 4, its order has period 4, and its native level is N.

Here v_2(N)=1. The defining-sequence/minimality argument of Block166 forces the terminal minimal approximation field E/F to have degree 2. There is no possible intermediate degree between 2 and 4. The scalar terminal case is excluded by 4 not dividing N, and a minimal full-degree 4 parameter is excluded because gcd(N,4)=2. Thus the actual degree pattern is forced to be 4→2.

Write c=−k_0(beta,A), so 1<=c<N. Select the critical approximation furnished by the construction in BK 2.4.1 and the remark preceding 2.4.7. Because the original field is maximal, this approximation can be selected with a simple maximal derived stratum in B=End_E(F^4)≅M2(E). Equivalently one can replace its derived stratum by an equivalent simple representative and use BK 2.2.2–3. The derived field has degree 2 and ramification index 2 over E. A simple stratum [B_order,c,c−1,alpha] is minimal. Since the complete B-chain has period 2 and the derived field is totally ramified, BK 1.4.14 forces

```
gcd(c,2)=1.
```

Therefore c is odd. It cannot be made even while preserving this degree pattern and a genuine maximal derived field.

Let j=(c+1)/2. The actual orders are

```
J_order=O_K+Q^j+P^(N/2),
H_order=O_K+Q^j+P^(N/2+1).
```

The final low transition contributes nothing, while the high transition has dimension 4−2=2. Thus every such Pi has

```
dim_F2 J^1/H^1=2,       dim eta=2.
```

The desired two-plane source carrier has dimension 4, so it cannot be a sector of this eta. Requiring the self-pair conductor additionally fixes c=3h_1 through

```
a(Pi^vee×Pi)=15+2N+c,
```

but is unnecessary for the capacity failure.

J.6 6. Rank 8: the three invariants do not force the good profile

The full 2→4→8 tower has quotient dimension 6 and enough capacity. Nevertheless Swan, torsion number and self-pair conductor alone do not force that degree pattern. Here is an actual skipped-degree arithmetic construction with the same three numerical invariants and a smaller quotient.

Fix increasing odd h_1<h_2<h_3 and put

```
q=3h_3,       N=4q=12h_3,       c=h_1+4h_2.
```

Then c is odd and 0<c<N. Indeed c<5h_3, so even 2c<N.

Construct the separable totally ramified quadratic extension

```
E=F(alpha),       alpha^2+alpha=t^(-q).
```

Put a=(q+1)/2 and z=t^a alpha. Then z is a uniformizer and

```
z^2+t^a z+t=0.
```

The element alpha has E-valuation −q and is minimal over F. On E^4, gamma=alpha I_4 is minimal of ambient level N in the complete period-eight order A.

Now construct the separable Eisenstein quartic extension over E

```
L=E(e),       e^4+z e+z=0,
```

and let C be multiplication by e^(−c) on L as a four-dimensional E-space. Since c is odd, e^(−c) is minimal, generates L/E, and defines a simple maximal derived stratum [B_order,c,c−1,C].

Let P0 extract the coefficient of 1 on each E-coordinate in the basis 1,z. The normalized outer corestriction from the Artin–Schreier right-inverse calculation has

```
s(P0)=t/z^2.
```

Consequently the explicit F-linear lift

```
b=(z^2/t) C P0
```

satisfies s(b)=C and has valuation −c. Define the actual rank-eight matrix

```
beta=alpha I_4+(z^2/t) C P0.
```

BK 2.2.3 applies because gamma is simple at level c<N and the derived stratum is simple maximal. It proves that F[beta]/F is totally ramified of degree 8, normalizes the complete chain, has leading level N, and has critical exponent −c. Its defining degree pattern is 8↠2, skipping degree 4.

Since N is even and c is odd, the actual Heisenberg quotient has dimension

```
(8−4)·1+(4−1)·0=4,
```

so eta has dimension 4. Yet its invariants are

```
Swan=N=12h_3,
torsion=1,
a(Pi^vee×Pi)=63+4N+3c
            =63+48h_3+12h_2+3h_1,
```

exactly the same as the full three-stage profile. Any maximal simple character and its standard type completion give an actual supercuspidal Pi with these values. The required full three-plane source carrier would have dimension 8, larger than this eta.

This is an arithmetic counterexample to the proposed implication from those three numerical invariants to adequate rank-eight Heisenberg capacity. It is not an H5 counterexample, does not identify the skipped host with the source-defined classical candidate, and does not claim that it preserves the supplied full source characters.

J.7 7. Which repairs have actually been tested

J.7.1 Replacing the displayed type or parameter at fixed Swan and torsion

At ranks 2 and 4 this fails by Sections 4–5 for every maximal simple type satisfying those invariants. A new matrix search or a different critical coefficient cannot manufacture the missing quotient dimensions.

At rank 8 the full tower supplies the good profile, but the three invariants alone do not force it inside a separately selected candidate. Section 6 proves the failure of that proposed inference with an explicit actual host. A finer endo-class or ramification comparison would be needed for that same-candidate assertion.

J.7.2 Skipping approximation degrees or changing critical parity

At rank 2 there is no degree to skip. At rank 4 the level excludes both a scalar minimal approximation and a full-degree minimal approximation. The remaining quadratic derived step has odd critical exponent by minimality. Thus neither skipping nor declaring that exponent even is a valid repair at fixed invariants.

At rank 8 a degree skip is genuinely possible, but it decreases capacity in the explicit example above.

J.7.3 Scalar twists, shallow or deep

A scalar twist of the existing actual inducing representation does not increase its dimension. The exact Block164 pushout transport identifies the same quotient J/H and leaves its commutator pairing unchanged. Thus it cannot turn an insufficient eta into a whole source carrier.

For an actual parameter, replacing beta by beta+aI with a∈F leaves its field, centralizer and commutator map unchanged. A shallow scalar addition does not change the leading level. A dominating scalar addition can change that level, but it contributes no new noncentral Heisenberg directions: in the defining sequence the added scalar stage has equal H and J orders. This agrees with

the exact representation-level twist transport. Enlarging a displayed numerical level is therefore not a capacity repair.

This paragraph does not assert that every arbitrary scalar character twist is represented by one globally specified additive parameter. The already proved group/representation transport suffices for its negative capacity conclusion.

J.7.4 Adding residue multiplicity or increasing ambient rank

The torsion-one condition forces a full-degree totally ramified parameter field and residue multiplicity one. A host with an unramified or residual multiplicity factor does not preserve that invariant. Increasing the ambient rank changes the required source rank. Neither maneuver closes the current fixed-rank construction.

J.7.5 Using a larger arithmetic carrier inside the same representation

A larger compact subgroup representation, a genuinely different arithmetic quotient, or an induction-stage carrier might have enough dimension. Such an enlargement is not ruled out here. It would need an actual finite Heisenberg action carrying the prescribed full source data and a proved recovery in the unchanged candidate. Artificially adding vector-space copies or transporting abstract operators to an induced space does not provide that arithmetic attachment or prove its kernel compatibility. No such repair is constructed in this note.

J.8 8. Quadratic refinements and marked centers

For d>=3, the dimension result gives a symplectic embedding of the source quotient into the actual host quotient. With C trivial, a C×-central pushout can then be matched by its alternating commutator, and Stone–von Neumann supplies the corresponding source sector. This is the same finite scalar-pushout attachment contract used previously.

At p=2, that statement must not be promoted to an identification of arbitrary preselected native quadratic refinements. The C× pushout permits rescaling lifts by i and can identify the pushouts of the dihedral and quaternion extraspecial planes, even though their native square functions differ. A chosen source embedding can preserve that source's full finite central extension and its character values. It does not thereby identify a separately marked target mu_2 extension or its Arf invariant.

In particular, symplectic-isotropic reduction respects a prescribed quadratic form only when the reduced subspace is totally singular for that form. The retained `DYADIC_AND_WEAK_ATTACHMENT159.md` gives the exact quadratic/Arf obstruction. Additional source or target marked centers, chosen lifts, quadratic refinements and nontrivial stabilizer actions therefore require their own comparison. They are neither erased from the task nor supplied by a dimension inequality.

J.9 9. Scope and source provenance

The strongest proved conclusion is exact capacity for the specified actual full defining profile, together with invariant low-rank failures and an explicit rank-eight counterexample to numerical profile forcing. No universal impossibility for original Theorem 16.2 is asserted. In particular this note supplies no complete original H1–H5 counterexample and does not decide whether that endpoint requires this precise whole-maximal-type-carrier realization.

Primary source directly reread: Bushnell–Kutzko, *The admissible dual of GL(N) via compact open subgroups*, 1.4.14–15; 2.2.2–3; 2.4.1–2 and the maximal-field remark immediately preceding 2.4.7; 3.1.5, 3.1.7–9; 3.2.1–4; 3.4.1; 6.2.5 and its following paragraph.

Authorized retained text:

```
closure_recovery/extracted/Paper_XI_Continuation158_General_Sectors_and_Arit
hmetic_Constraints/scratch158/information158/input_cache/BK_source.txt
SHA256 d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8
```

Internal inputs reused at their stated scope:

- `attachment166/tower/RECURSIVE_TOWER166.md`: actual defining degrees, levels, filtered right inverse and separated grade count. Its field constructor permits p=2.
- `attachment166/tower/TRACE_ZERO_AS_TOWER166.md`, Sections 1–2: normalized corestriction and explicit right inverse. Their algebraic formulas apply at p=2. The odd-prime trace-zero assertion is not used.
- `attachment166/capacity/LEADING_LAYER_CAPACITY166.md`: the leading subspace argument and the minimal terminal degree calculation.
- `base_realization165/polarization/PROFILE_RIGIDITY165.md` and `closure_recovery/full161/scratch161/placement161/ADJOINT_CONDUCTOR_TEST161.md`: retained formal-degree/self-pair conductor identities and normalization.
- `input_pair164/TWIST_TRANSPORT.md`: exact character-twist/pushout transport.
- `closure_recovery/extracted/Paper_XI_Continuation159_Type_Completion_and_Sector_Recovery/scratch159/coverage159/DYADIC_AND_WEAK_ATTACHMENT159.md`: distinction between symplectic and marked quadratic reduction.

No full copyrighted source is redistributed here. No new external correspondence theorem or H5 premise is introduced.

K Whole finite extensions and dyadic marking criteria

Evidence: direct reading of the original dyadic contract and ordinary finite group proofs. This note does not prove a dyadic arithmetic field construction, its simple-character existence, nontrivial stabilizer attachment, H5, or a Lean closure. It identifies exactly what a supplied actual host and sector must preserve, and proves the marked finite attachment step for C={1}.

K.1 1. The original datum includes more than the alternating form

The original text `closure_attempt162/inputs/original_reading.txt` states in Definition 11.1, starting line 4911, that at p=2 a restricted bridge additionally supplies compatible quadratic refinements and an isomorphism of the resulting Heisenberg central extensions lifting the given symplectic isometry. Sections 13.1–13.9 define the refinements and their Arf/Gauss signs.

The retained visual appendix is particularly explicit, at lines 6892–6947: with center {I,-I} fixed, squares of lifts carry the quadratic bit. Multiplying a lift by i changes the particular finite central-extension realization. The appendix says that allowing complex scalar rescaling **without recording this distinction** erases data that the bridge is required to retain.

The finite record must therefore retain at least:

- the labeled symplectic quotient and its pairing;
- the supplied quadratic refinement q;
- the specified finite central extension, its central embedding/character, and any marked lifts or specified extension isomorphism;
- the resulting normalized Gauss sign, and all declared phase operators;
- the complete original character values retained by the scalar pushout.

The last item alone does not retain the preceding finite-subgroup/lift marking unless that marking is explicitly carried along. The raw text does not require literal embeddings of the old arithmetic matrix groups into a new type. Nor does it define a unique quadratic refinement extracted automatically from an unmarked C× scalar extension of every arithmetic type.

K.2 2. General criterion for finite central extensions

Let V be an elementary abelian 2-group with ordered basis v_1,…,v_m. Let E_s and E_t be central extensions of V by the same abelian group Z. Fix the identity on Z and a quotient isomorphism f. Choose lifts a_i of v_i in E_s and b_i of f(v_i) in E_t. Write

```
a_i²=s_i,  b_i²=t_i,  [a_i,a_j]=c_ij, [b_i,b_j]=d_ij.
```

Every element has a unique normal form

```
z a_1^(epsilon_1) ... a_m^(epsilon_m), epsilon_i in {0,1},
```

and analogously on the target. The multiplication is completely determined by the displayed squares and commutators.

Criterion. A center-fixing isomorphism lifting f exists precisely when

```
c_ij=d_ij for every i,j,
s_i/t_i belongs to Z² for every i.                    (1)
```

Indeed, every such isomorphism must take a_i to lambda_i b_i for some lambda_i in Z, forcing lambda_i²=s_i/t_i. Conversely these equations make the assignment a_i -> lambda_i b_i preserve all relations, and the normal forms prove it is bijective.

If the **particular lifts** a_i and b_i must correspond exactly, the condition is s_i=t_i, with lambda_i=1. If a full chosen section is marked, rather than only generating lifts, its multiplication coefficients must also correspond:

```
a(v)a(w)=c_s(v,w)a(v+w),
b(fv)b(fw)=c_t(fv,fw)b(f(v+w)),
```

and exact section preservation requires c_s(v,w)=c_t(fv,fw) for every pair. Transporting the complete section under an existing group isomorphism retains these identities automatically. Preserving just its squares need not retain an additional independently specified section.

K.2.1 2.1. Fixed two-element center

For $Z=\{+1,-1\}$, $Z^2=\{1\}$. The lift square is independent of the lift within its fiber and defines

```
a(v)²=(-1)^(q_s(v)).
```

The identity $(ab)^2=a^2b^2[b,a]$ gives the quadratic refinement law. Criterion (1) becomes the exact requirement

```
q_t(f(v))=q_s(v) for every v.                    (2)
```

For an unspecified symplectic isomorphism between equal-dimensional nondegenerate spaces, such an f exists exactly when their Arf invariants agree. For a preassigned f, equality of Arf invariants alone is insufficient: (2) must hold for that f.

K.2.2 2.2. Four-element and complex centers

For Z=mu_4, multiplication of a lift by i changes its square by -1. The unmarked extension criterion preserves square classes modulo Z^2=mu_2. In particular, squares +1 and -1 do not distinguish unmarked mu_4 extensions. A separately marked finite mu_2-central extension subgroup (such as the full D8 or Q8), or prescribed lifts, still distinguishes them.

For Z=C× every square ratio has a square root. Thus matching commutators is sufficient for an unmarked center-fixing extension isomorphism. This fact does not delete a marked finite subgroup from the objects being compared.

K.3 3. Exact D8/Q8 check

In dimension two over C put

```
X=[[0,1],[1,0]],       Z=[[1,0],[0,-1]].
```

Then $X^2=Z^2=I$, XZ=-ZX, and [X,Z]=-I. The group generated by X and Z has order 8 and is D8. Its refinement on $V=F2^2$ is

```
q_D(x,y)=xy,       Arf(q_D)=0,
```

with square table (+1,+1,+1,-1) at (0,e,f,e+f), and Gauss sum +2.

The generators iX,iZ also anticommute, but each squares to -I and their product also squares to -I. They generate Q8, with

```
q_Q(x,y)=xy+x+y,   Arf(q_Q)=1,
```

square table (+1,-1,-1,-1), and Gauss sum -2.

Both C× pushouts identify with the same ambient matrix group

```
P={z X^a Z^b : z in C×, a,b in {0,1}}.
```

Nevertheless the marked subgroups

```
H_D=<X,Z>,          H_Q=<iX,iZ>
```

are different. No isomorphism fixing the two-element center can map H_D onto H_Q: it would preserve q and hence the Arf invariant, which differs. There is also no center-fixing automorphism of P carrying the marked H_D onto H_Q. Their unmarked pushouts are isomorphic, but the marked objects (P,H_D) and (P,H_Q) are not.

Conversely a Q8 source embeds perfectly into the same P by sending its generators to iX and iZ. This preserves every source square and its negative Gauss sign. It does not identify those generators with the independently marked D8 lifts X,Z. This is the distinction needed for the arithmetic sector.

K.4 4. Marked scalar-pushout attachment with C={1}

Assume an actual type, its full character theta, and its Heisenberg quotient W have already been constructed. Let P_W be its actual C× scalar pushout, acting on its actual Heisenberg representation eta_W. Let V_src be a finite nondegenerate symplectic F2-space of dimension 2s, with its complete supplied dyadic marked extension data. For several breaks use their labeled orthogonal sum, balanced central product, and supplied product refinement.

Assume the following precise finite contract:

1. The source full-character pushout P_src has quotient V_src and the same represented scalar center as P_W.
2. The supplied finite central extensions and their marked lifts are retained as subgroups/elements of P_src, or as already specified maps into P_src. Any central character kernel already factored out must be recorded. A scalar pushout cannot recover invisible elements of that kernel.
3. The host has enough symplectic capacity, dim W>=2s, and the new attachment into this host has no independently fixed conflicting quadratic marking.
4. The declared stabilizer is C={1}, and the graph has one vertex.

Then a complete marked attachment exists in the **same actual candidate**.

Proof. Choose a symplectic embedding f:V_src -> W and let V be its image. The restriction P_V of P_W and P_src have the same commutator. Criterion (1) with center C× produces an isomorphism

```
F:P_src -> P_V
```

fixing every scalar and lifting f. Carry every supplied finite subgroup H_src to the actual subgroup F(H_src) of P_V, and every marked lift a to F(a). This operation does not erase square data:

```
F(a)²=F(a²),        [F(a),F(b)]=F([a,b]).              (3)
```

Since F fixes the scalar center, a²=(-1)^q(v) implies F(a)²=(-1)^q(v). The transported refinement is therefore exactly q, not a new choice with the same polar form. All marked section products, specified extension maps, and finite group relations are preserved by the same F. Consequently the Arf invariant and normalized Gauss sign are unchanged.

Every original h in a full character domain still maps to theta_i(h), because F fixes the entire scalar center. Thus the dyadic marking is retained together with the full original character values.

Write W=V perpendicular-sum U. Finite Stone–von Neumann gives

```
eta_W isomorphic to eta_src tensor eta_U.
```

Choose a line in eta_U and its projector e. It lies in the actual Heisenberg image algebra, and the marked source subgroup acts on e eta_W by precisely its supplied irreducible Heisenberg representation. Its lift squares remain the operators in (3). The existing sector recovery theorem then

recovers the unchanged inducing representation Lambda, and the same arithmetic candidate. There is no C-equivariance equation to solve because C={1}. QED.

This realizes an actual finite marked subgroup in the scalar extension of the actual type, with actual represented operators that are complex scalars times type operators. It does not assert that every marked source element is the image of an unscaled element of the old or new arithmetic matrix group. That stronger requirement of a literal embedding is absent from the present contract.

It also does not assert that the newly marked subgroup equals a separately specified canonical native subgroup of the host. The marked subgroup is part of the result and must be retained in the artifact. Omitting it and recording only the unmarked P_W would reproduce the loss displayed by D8/Q8.

K.5 5. If the host refinement is independently fixed

Suppose instead that the actual host arrives with a particular nonsingular quadratic refinement q_W and its actual finite mu_2 central extension H_W. An attachment into that marked extension requires a quadratic isometric embedding, not merely a symplectic embedding. There is a sharp capacity test.

Theorem. Let dim V_src=2s and dim W=2R. With embeddings otherwise free, a quadratic isometric embedding

```
(V_src,q_src) -> (W,q_W)
```

exists if and only if R>=s and, in the equal-capacity case R=s,

```
Arf(q_src)=Arf(q_W).                                    (4)
```

If R>s, either source Arf type embeds.

Proof. Necessity of dimension is immediate. An equal-dimensional embedding is an isometry, forcing (4). If R>s, choose a nondegenerate quadratic space U of dimension 2(R-s) with

```
Arf(q_U)=Arf(q_W)+Arf(q_src).
```

Both Arf types exist in every positive even dimension, by original Proposition 13.7. The orthogonal sum V_src perpendicular-sum U then has the same dimension and Arf invariant as W. Original Standard Fact 13.4 gives an isometry to W. Restriction to V_src gives the desired embedding. QED.

After this embedding is chosen, the preimage of its image in H_W has exactly q_src. The criterion for a fixed two-element center in Section 2 gives the extension isomorphism. Generating lifts with the specified squares can be matched. Additional fully prescribed section elements still require their exact product relations, as described in Section 2.

Thus **one spare symplectic plane absorbs the Arf mismatch**, without changing the host's own refinement. This conclusion remains conditional on the host's actual marked refinement/finite extension being supplied or constructed. It does not manufacture a canonical arithmetic quadratic datum from its dimension.

For a preassigned embedding f, the missing equation is instead

```
q_W(f(v))=q_src(v) for all v.                            (5)
```

Its discrepancy q_W composed with f minus q_src is a linear functional. Strict capacity allows a different quadratic embedding when it is free. It does not authorize replacing a fixed original map, fixed depth placement, or fixed transport marking.

K.6 6. Exact outcome for the dyadic construction

Under the retained one-vertex, C={1}, full scalar-pushout realization contract, an actual host with sufficient symplectic capacity does support every supplied finite dyadic marking, provided the

transported finite subgroup, lifts, refinement, and phase are explicitly kept. The proof is Section 4, rather than an assertion that commutators determine the dyadic sign.

If a particular host-native quadratic extension is fixed in advance, use Section 5. Strict capacity suffices for its quadratic embedding. Equal capacity requires the Arf equation, and a preassigned embedding requires (5). If only the type quotient is known and a specifically canonical host quadratic datum is requested, constructing and comparing that datum remains an arithmetic obligation. The scalar-pushout proof must not silently claim to have done it.

No part of this note handles nontrivial prescribed C actions. In that case quadratic embedding, full extension action, lift markings, and chosen honest phase operators must all be equivariant. Dimension and Arf equality alone do not settle those equations.

K.7 Evidence and author self-review

Read directly: original Definition 11.1, Sections 13 and 15, and visual appendix pages V–22/V–23 in the retained raw file; final Block166 `PROOF166.md`, Section 2, and `LEADING_LAYER_CAPACITY166.md`, Sections 2.1 and 4. The extension criterion and D8/Q8 calculation are complete finite proofs. The quadratic embedding theorem uses the Arf classification already explicitly supplied by the original paper. The sector conclusion reuses the retained exact recovery theorem and does not assert new arithmetic existence.

Self-review checked that complex lift rescaling is never treated as an automorphism changing squares within a fixed marked extension, that the marked subgroup remains in the output, that preassigned maps differ from freely chosen embeddings, and that fixed native refinements require their own actual input. No independent review of this note or new Lean execution is claimed here.

L The actual enlarged rank-two inducing carrier

Status: ordinary proof. The construction below supplies an actual arithmetic finite Heisenberg pair and exact recovery of an already fixed representation. It does not change the previously audited zero-dimensional BK quotient at rank two. No Lean verification or H5 comparison is claimed.

L.1 1. Exact setting and result

Let F=F_2((t)) and G=GL2(F). Let [A,q,0,beta] be a minimal simple stratum with K=F[beta] totally ramified of degree 2 and A its complete period-two hereditary order. Assume

```
q=2m+1>=3,        m>=1.
```

Write R=rad(A), $U_A^a = 1 + R^a$ for a>=1, and $U_K^q = 1 + \mathfrak{p}_K^q$. Let theta be a full simple character. Since q is odd and the residue field is F2, the actual maximal simple type group and character groups coincide:

```
J=J^1=H^1=U_K^1 U_A^(m+1).
```

Fix the full actual extension Lambda of theta to

```
N=K×J,
```

and the already fixed irreducible supercuspidal candidate

```
Pi=c-Ind_N^G Lambda.
```

Lambda is one-dimensional here. Its value on a uniformizer of K, including the noncompact scalar normalization, is part of the fixed data.

Define the larger compact open arithmetic subgroup

```
P=U_K^1 U_A^m.
```

Then the following statements hold.

1. J is normal of index 2 in P, and P∩N=J.
2. E=Ind_J^P theta is an actual irreducible two-dimensional P-representation.
3. There is an explicitly defined normal subgroup H_new⊂J of index 2 such that P/H_new≅F2² and theta_new=theta|H_new has nondegenerate commutator pairing on that quotient. E is its Heisenberg representation.
4. E embeds into the restriction of the same Pi, as the subspace of inducing functions supported on NP.
5. With X=c-Ind_J^G theta, a uniformizer varpi_K and s=Lambda(varpi_K), there is an exact quotient
   ```
   Pi ≅ X/(T−s)X,
   (Tf)(g)=f(varpi_K g).
   ```
   Induction in stages identifies X with c-Ind_P^G E. This gives an exact same-candidate recovery using the retained noncompact scalar, not merely a nonzero cyclic-generation assertion.

The new Heisenberg pair is an actual pair of compact subgroups of GL2(F), but is not asserted to be the BK H^1/J^1 pair of a maximal simple type. Whether the original H2 realization relation allows this enlargement is a separate obligation.

L.2 2. The compact enlargement is a group, with normal subgroup of index two

K× normalizes A and every power of its radical, so U_K^1 normalizes U_A^m. Therefore P is a subgroup and is compact open. It is contained in U_A^1 because m>=1 and $U_K^1 \subset U_A^1$.

The standard radical calculation gives

```
[U_A^a,U_A^b]⊂U_A^(a+b),        a,b>=1.
```

Indeed, for x∈R^a and y∈R^b, the difference (1+x)(1+y)−(1+y)(1+x) is in R^(a+b), and multiplication by unit inverses preserves that ideal.

Consequently

```
[U_A^m,U_K^1]⊂U_A^(m+1),
[U_A^m,U_A^(m+1)]⊂U_A^(2m+1)⊂U_A^(m+1).
```

These facts, and normalization by U_K^1, show that J is normal in P. They also show

```
[P,J]⊂U_A^(m+1).                                    (1)
```

To calculate the index, use

```
U_A^m∩J=U_K^m U_A^(m+1).
```

For if ku∈U_A^m with k∈U_K^1 and u∈U_A^(m+1), then $k \in K^\times \cap U_A^m = U_K^m$. The reverse inclusion is immediate. The complete period-two A-grade R^m / R^{m+1} has F2-dimension 2, and its field grade $\mathfrak{p}_K^m / \mathfrak{p}_K^{m+1}$ has dimension 1. Thus

```
[P:J]=[U_A^m:U_K^m U_A^(m+1)]=2.                    (2)
```

Finally, if kj∈P with k∈K× and j∈J, then $k \in Pj^{-1} \subset U_A^1$. Hence k∈U_K^1 and kj∈J. Therefore

```
P∩N=J.                                              (3)
```

These arguments use the actual group intersections. They do not replace P or J by a vector space surrogate.

L.3 3. Irreducibility and the conjugate-character ratio

For the minimal simple character, the primary BK intertwining theorem gives

```
I_G(theta)=J K×J=N.
```

Here K× normalizes J and theta. In particular, for g∈P\J, (3) says g does not intertwine theta. Because J is normal in P, this means that the two characters

```
theta,
theta^g(j)=theta(gjg^−1)
```

are distinct. The index-two induction E=Ind_J^P theta is therefore irreducible. This also follows directly from the two distinct J-weight lines, which are interchanged by g.

Define the actual character of J

```
lambda(j)=theta^g(j)/theta(j).
```

It is nontrivial. By (1),

```
lambda(j)=theta(gjg^−1j^−1),
```

and the argument of theta lies in U_A^(m+1). The full simple character has exact restriction

```
theta(1+x)=psi_F(tr_F(beta x)),       x∈R^(m+1).
```

Since F has characteristic 2, its additive characters take values in {1,−1}. Consequently

```
lambda(J)={1,−1}.                                    (4)
```

No assertion about shallow values of theta is used here. Theta may have a larger finite image on J. Only its ratio with its actual conjugate is proved to have order 2.

As g²∈J, conjugation by g² fixes the character theta. Thus $\lambda^g = \lambda^{-1} = \lambda$. It follows that

```
H_new=ker(lambda)
```

is normal in P. It has index 2 in J and index 4 in P.

L.4 4. The actual finite Heisenberg pair

Use the function model of E with f(jp)=theta(j)f(p) and right translation by P. Coordinates are f(1),f(g). Then for j∈J,

```
E(j)=diag(theta(j),theta^g(j))
    =theta(j) diag(1,lambda(j)),
```

while

```
E(g)=[[0,1],[theta(g²),0]].                          (5)
```

Therefore E(g)²=theta(g²)I. Choose j_0∈J with lambda(j_0)=−1, and put

```
Z=theta(j_0)^−1 E(j_0)=diag(1,−1),
X=E(g).
```

The exact matrix identity is

```
XZX^−1=−Z.                                           (6)
```

The projective image of J has order 2, generated by Z, while X is off-diagonal and contributes an independent projective involution. Thus the projective image of P is exactly C2×C2, with nontrivial commutator (6). Its projective kernel is precisely H_new. In particular

```
P/H_new≅F2².
```

On H_new, E is the scalar theta_new=theta|H_new. This character is P-invariant: conjugation by J fixes theta and conjugation by g changes theta by lambda, which is trivial on H_new. The resulting alternating form

```
(pH_new,p'H_new)↦theta_new([p,p'])
```

is nondegenerate, since its two displayed generators have commutator −1. Therefore (H_new,P,theta_new) is an actual arithmetic finite Heisenberg pair, and its actual irreducible carrier E has dimension 2.

All representations here have finite image on the compact group P: theta has open kernel, the induction has finite dimension, and an open normal core of that kernel acts trivially. Thus no issue with an infinite projective image is being suppressed.

This proof does not require rescaling the actual matrix g inside GL2(F). Its represented square theta(g²) can be an arbitrary retained scalar. Rescaling X is only allowed later in a C× pushout if that is the attachment contract.

L.5 5. Explicit embedding into the same fixed candidate

Realize Pi as functions F:G→C with

```
F(nx)=Lambda(n)F(x),
```

compact support modulo N, and right translation by G. For f∈E define i(f) by

```
i(f)(np)=Lambda(n)f(p),       n∈N, p∈P,
```

and zero off NP. This is well-defined because N∩P=J and Lambda|J=theta. The set NP is a finite union of N-cosets, by [P:J]=2, so the support is compact modulo N. The map i is P-equivariant and injective.

Its image is exactly the two-dimensional P-stable subspace of Pi consisting of inducing functions supported on NP. Thus the enlarged carrier is already inside the given Pi. No new candidate is chosen, and all noncompact values of Lambda remain fixed.

Since Pi is irreducible and i(E) is nonzero, its G-span is Pi. More precisely, there is the canonical G-equivariant quotient

```
Q_E:c-Ind_P^G E→Pi,
Q_E(F)=sum_(Px∈P\G) Pi(x^−1)i(F(x)).                (7)
```

The sum is finite and independent of representatives. If x is replaced by px, then F(px)=E(p)F(x) and P-equivariance of i cancels the additional Pi(p^−1). Formula (7) is nonzero and hence onto. The next section identifies its kernel exactly.

L.6 6. Exact recovery by the noncompact Hecke eigenvalue

Let X=c-Ind_J^G theta, in the same left-covariant/right-translation convention. Induction in stages gives an explicit isomorphism

```
c-Ind_P^G E → X,
F ↦ [x↦F(x)(1)].                                   (8)
```

The inverse sends f∈X to the E-valued function F(x)(p)=f(px). Both covariance statements follow directly from the definitions, and the finite index [P:J] ensures the support conditions agree.

Fix a uniformizer varpi=varpi_K. Because K× normalizes theta, the operator

```
(Tf)(x)=f(varpi x)                                 (9)
```

is a well-defined invertible G-equivariant operator on X. Put s=Lambda(varpi), which is nonzero. Since k_K=F2 and J contains U_K^1,

```
N/J≅K×/U_K^1≅Z,
```

with generator varpi. Define

```
Q:X→Pi,
(Qf)(x)=sum_(ell∈Z) s^(−ell) f(varpi^ell x).       (10)
```

For each x the sum is finite, since f has compact support modulo the open subgroup J. Equation (10) satisfies

```
(Qf)(varpi x)=s(Qf)(x),
(Qf)(jx)=theta(j)(Qf)(x),      j∈J,
```

and therefore the required N-covariance. Its support is compact modulo N. It commutes with G-right translation and satisfies

```
QT=sQ.                                             (11)
```

It is onto: a function in Pi has finite support modulo N, and on each occupied N-coset its value can be lifted to a function supported on one J-coset. Extend those finitely many choices using theta.

To compute the kernel, decompose X as the algebraic direct sum of its subspaces supported on the left N-cosets in G. Each such summand is the space of finitely supported sequences on N/J≅Z, with the fixed theta covariance. The operator T is a bilateral shift. On that summand, (10) is evaluation of

its Laurent polynomial at the nonzero scalar s, up to the harmless convention of replacing the Laurent variable by its inverse. The kernel is therefore the principal ideal generated by T−s. The elementary identity

```
ker(C[z,z^−1]→C, z↦s)=(z−s)
```

proves, summand by summand, that

```
ker Q=(T−s)X.
```

All vectors occupy finitely many such summands, so no completion or closure is needed. Thus

```
Pi ≅ X/(T−s)X ≅ (c-Ind_P^G E)/(T−s)(c-Ind_P^G E),    (12)
```

where T is transported through (8).

Under (8), formula (10) agrees with (7): on the standard copy of E supported on P, it gives exactly i, and G-equivariance then gives equality everywhere. Thus (12) identifies the kernel of the canonical recovery map, not merely the kernel of an unrelated surjection.

The same description can be phrased as evaluation of the type Hecke algebra at the character determined by Lambda. Its intertwining support is N, each supported intertwining space is one-dimensional, and the retained extension Lambda normalizes the Laurent generator. The explicit formulas above avoid any ambiguity about convolution orientation.

The scalar s is essential. E alone, being compact data, does not determine it. Changing s can change the unramified normalization of the candidate. The construction retains the original Lambda and s, so it recovers the exact original Pi rather than an unspecified member of its inertial class.

L.7 7. Full scalar-pushout attachment and remaining marking obligations

Form the C× pushout of the actual pair (H_new,P,theta_new). It is a Heisenberg extension of the actual symplectic plane P/H_new. Any supplied source plane with the same chosen alternating form has a center-preserving pushout isomorphism into it. Every source full-character value is then preserved in the center, as in the retained full-character attachment theorem. Stone–von Neumann identifies its source carrier with the actual two-dimensional E.

Together with the embedding i and exact quotient (12), this provides an actual arithmetic carrier and recovery inside the same fixed candidate. It closes the carrier-capacity failure of the old rank-two eta by changing the compact arithmetic pair, not by increasing the old eta or discarding its dimension calculation.

At p=2, a C× pushout does not automatically identify a separately marked native mu_2 extension or quadratic refinement. The source finite central extension can be retained through its chosen embedding, while a required comparison with the target's preselected square function remains an additional question. No arbitrary native Arf invariant is asserted here.

Nontrivial stabilizer operators, transport graphs, and original H2's requirement that a target be a particular BK type-side quotient are likewise separate. In particular, the theorem proves that the new pair is an actual arithmetic finite Heisenberg pair, but does not rename it as the unchanged maximal-simple-type pair. No new hypothesis is silently added to original Theorem 16.2.

L.8 8. Scope, smallest boundary, and sources

The inequality q>=3 matters. If q=1, then m=0 and the displayed enlargement is not the group used above. Over residue field F2 the complete Iwahori unit group equals U_A^1, and a naive level-zero replacement does not give the asserted index-two enlargement. No q=1 claim is made. The source levels q=3h with positive odd h satisfy the required q>=3.

The proof uses the actual one-dimensional Lambda because the maximal parameter field has full degree and residue field F2. It does not apply the same scalar argument automatically to a higher-rank inducing representation.

Primary BK passages: 3.1.7–8 (odd minimal H/J equality); 3.2.1–2 (deep restriction and K× normalization); 3.3.2 and 5.1.1/5.1.8 (intertwining); and 6.2.1–5 (the fixed maximal-type extension and compact induction). The subgroup, matrix, Mackey-support, and Laurent-evaluation calculations are proved explicitly above. No external correspondence theorem is used in this enlargement.

Retained primary text:

```
closure_recovery/extracted/Paper_XI_Continuation158_General_Sectors_and_Arit
hmetic_Constraints/scratch158/information158/input_cache/BK_source.txt
SHA256 d69db0827256707a34cd2562af68b9de11fc889ab3f43535ec47b620d58ca9c8
```

The previously reviewed negative statement concerned eta of the original BK J^1/H^1. That quotient remains zero. The new positive statement concerns a different, explicitly constructed arithmetic pair inside the same ambient representation, with exact recovery that preserves the noncompact normalization.

M The actual enlarged rank-four inducing carrier

M.1 Result and boundary

For the actual rank-four Artin–Schreier candidate of Block167, one can enlarge the compact group J1 to an actual compact subgroup P with

```
J1 normal in P,  [P:J1]=2,
E=Ind_(J1)^P(eta) irreducible of dimension 4.
```

The representation E occurs inside the restriction of the same actual supercuspidal candidate pi. There is an explicit normal subgroup H0 of P with [P:H0]=16 such that P/H0 is elementary abelian and the commutator form supplied by the actual character on H0 is nondegenerate. Thus the carrier has an actual four-dimensional binary symplectic quotient. No abstract transfer of matrix operators is used to create this action.

The finite image of P has scalar center theta(H0), which can be larger than {+1,-1}. It is literally an extraspecial group if this scalar center has order two. That additional assertion is not assumed or proved here. The construction supplies an actual finite Heisenberg central extension with nondegenerate projective quotient F_2^4. Matching an independently prescribed quadratic phase is also a separate assertion.

M.2 1. Actual arithmetic groups and notation

Let F=F_2((t)), and take the audited rank-four constructor with odd 0<h_1<h_2. Write

```
q_i=3h_i,  n=2q_2,  a=n/2=q_2,
r=q_1=2m+1,  m=(r-1)/2>=1.
```

We have a>r, so n>2r. Let A be the complete period-four hereditary order, with radical P_A. Let gamma be the actual degree-two outer minimal approximation, E=F[gamma], B=End_E(F^4)=M_2(E), B0=A cap B, and Q=rad(B0). The actual final parameter is beta=gamma+delta, and

```
s(delta)=c,
```

where c is the minimal degree-two Artin–Schreier parameter over E, of B0-level r. The final field K=F[beta] is totally ramified of degree four. No embedding E subset K is asserted.

Use $U_A^j = 1 + P_A^j$ and $U_B^j = 1 + Q^j$. BK 3.1.7–9 give the actual groups

```
H=H1(beta)=U_K^1 U_B^(m+1) U_A^(a+1),
J=J1(beta)=U_K^1 U_B^(m+1) U_A^a.
```

In particular J/H has dimension two over F_2. Let theta be the actual full simple character constructed for beta, and let eta be its unique Heisenberg representation on J, of dimension two. The argument below works with this supplied actual theta. It does not replace its values.

Choose the outer minimal simple character theta_gamma from its defining sequence. On H cap U_A^(m+1), the recursion says

```
theta=theta_gamma psi_delta.
```

Moreover H is contained in H1(gamma)=U_B^1 U_A^(a+1). The character theta_gamma is invariant under U_B^1: on U_B^1 it factors through the E-determinant, and on U_A^(a+1) it is the gamma-additive character.

M.3 2. The actual final field has the needed graded line

For every integer j>=1 and k in K cap P_A^j, the minimal-approximation commutator estimate gives

```
k=b+e,  b in Q^j,  e in P_A^(j+n-r).              (1)
```

Indeed [gamma,k]=-[delta,k] belongs to P_A^(j-r), and the minimal corestriction exact sequence for gamma solves its commutator with an error of grade j+n-r. The remaining difference centralizes gamma and therefore belongs to B. This is the usual minimal case of BK 1.4.10.

Applying s to [beta,k]=0 and using (1) gives

```
[c,b] in Q^(j+n-2r).
```

Hence the leading B-grade of b commutes with the leading grade of c. Let R be the complete period-two E-chain shift, with R^2=z I, where z is the chosen uniformizer of E. In the constructor's basis

```
c congruent to R^(-r) modulo Q^(1-r).
```

Because r is odd, the centralizing subspace on every B-grade j is the one-dimensional line spanned by R^j. On the other hand K is a totally ramified maximal field normalizing the complete A-chain, so (K cap P_A^j)/(K cap P_A^(j+1)) has dimension one over F_2. Estimate (1) is injective on the leading grade, so its image is exactly that centralizing line.

This proves the required graded assertion for the actual K, without replacing it by the derived model E[c] or assuming nested fields.

M.4 3. Normality and index of the enlarged group

Set

```
P=J U_B^m.
```

We first prove this is a group. The group U_B^m normalizes U_B^(m+1) and every U_A^j. For k in U_K^1, (1) approximates k by an element of U_B^1 with error in U_A^(n-r+1). Therefore

```
[U_B^m,U_K^1]
   subset U_B^(m+1) U_A^(m+n-r+1)
   subset H,
```

where the last inclusion follows from a>r=2m+1. Thus U_B^m normalizes both H and J, and P is an actual compact subgroup with J normal in P.

The quotient U_B^m / U_B^{m+1} has dimension two. By Section 2 the image of J cap U_B^m on this grade is exactly its centralizing line, of dimension one. For the converse inclusion needed here, the B-approximation of a K-element of grade m differs from it by grade m+n-r>a. That error lies in U_A^a subset J. Thus the whole centralizing line really occurs in the intersection, not only in an abstract associated graded space. Consequently

```
[P:J]=[U_B^m:J cap U_B^m]=2.                    (2)
```

For a concrete representative choose D=diag(1,0) in B and put

```
x=R^m D,  u=1+x.
```

Its leading grade is outside the centralizing line, so u represents the nontrivial coset of P/J. Since $u^2 = 1 + x^2$ in characteristic two, we have u^2 in U_B^(2m) subset U_B^(m+1) subset H.

M.5 4. The actual character has an orbit of length two

Define the character of H

```
lambda(h)=theta(u h u^(-1))/theta(h).
```

For h in H the preceding normality argument gives [u,h] in H cap U_A^(m+1). Since theta_gamma is invariant under u,

```
lambda(h)=psi_delta([u,h]) in {+1,-1}.          (3)
```

Thus lambda has order dividing two, and its values are actual values of the supplied simple character. It is J-invariant, since both theta and its u-conjugate are J-invariant.

It is nontrivial. Take y=R^(m+1)D and h=1+y in U_B^(m+1). The leading commutator in (3), after corestriction, has value

```
psi_E(tr_(M_2(E)/E)(c[x,y]))=-1.
```

Here the commutator remainder has B-grade at least 3m+1, which becomes positive after multiplication by c of grade -2m-1. On the leading grades, writing X=R^m diag(A,B) and Y=R^(m+1)diag(C,D), one obtains

```
tr(R^(-2m-1)[X,Y])=(A+B)(C+D) in F_2.
```

Both sums equal one for the chosen D=diag(1,0). The normalized psi_E is nontrivial on the residue field and trivial on its maximal ideal. For this AS normalization one can check the sign directly from psi_E(1)=psi_F(Tr_(E/F)(alpha))=psi_F(1)=-1.

Hence theta^u differs from theta, eta^u differs from eta, and Clifford theory applied to (2) gives

```
E_carrier=Ind_J^P(eta) irreducible, dim(E_carrier)=4.       (4)
```

M.6 5. An actual nondegenerate symplectic quotient of dimension four

Put H0=ker(lambda) inside H, and theta0=theta restricted to H0. In the two-coset induction model, h in H acts by

```
diag(theta(h) I_2, theta^u(h) I_2).
```

Thus it acts scalarly precisely when h belongs to H0. We check that H0 is the entire scalar preimage and that its quotient is symplectic.

First, lambda is invariant under u. Indeed u^2 belongs to J and theta is J-invariant, so $\lambda^u = \lambda^{-1} = \lambda$ by (3). Consequently H0 is normal in P, and theta0 is P-invariant.

Second, lambda is trivial on U_A^(a+1). There the correction formula is valid, gamma commutes with u, and the delta contribution to a commutator has grade at least a+1+m-r=a-m>0. Therefore it lies in the kernel of the additive character.

Write J=H U_A^a. Since lambda is a character and J-invariant, it kills [H,H] and [H,U_A^a]. It also kills $[U_A^a, U_A^a]$, which lies in U_A^(2a). Hence lambda([J,J])=1. It kills every square in J as well: squares of H have lambda-value one, squares of U_A^a lie in U_A^(2a), and the mixed commutator contributions have just been handled.

Finally lambda([u,H])=1 by u-invariance, and [u,U_A^a] belongs to U_A^(a+m) subset U_A^(a+1), so lambda([u,J])=1. Also lambda(u^2)=1 because u commutes with u^2. It follows that

```
[P,P] subset H0,  g^2 in H0 for every g in P,
[P:H0]=[P:J][J:H][H:H0]=2*4*2=16.
```

Therefore P/H0 is an actual elementary abelian group of dimension four. If an element of J acts scalarly on the carrier, its first eta-block is scalar, so it belongs to H by the original nondegenerate

Heisenberg quotient J/H. It then belongs to H0. An element outside J exchanges the two coset summands and cannot act scalarly. Thus H0 is precisely the scalar preimage.

The pairing

```
(gH0,kH0) |-> theta0([g,k])
```

is alternating and well defined. Since the projective quotient has exponent two, its values lie in {+1,-1}. It is nondegenerate: an element in its radical commutes with the entire irreducible carrier image and therefore acts scalarly by Schur's lemma, forcing it into H0.

This is the actual four-dimensional binary Heisenberg quotient sought for capacity. Its operators are supplied by elements of P.

M.7 6. The carrier occurs in the same supercuspidal candidate

Let pi=c-Ind_N^(GL_4(F))(Lambda) be the existing actual candidate, with N=K^times J. The residue field is F_2, so the compact extended type restricts to eta on J. Since P is contained in U_A^1,

```
N cap P=J.
```

Indeed an element k j in P with k in K^times has k in U_A^1, hence k in U_K^1 subset J. The identity double-coset term in the restriction of compact induction to P is therefore

```
Ind_J^P(eta)=E_carrier.
```

It is realized concretely by the functions supported on N P. Thus the four-dimensional carrier occurs in the same pi, with its already fixed full type character and noncompact extension scalar. No new arithmetic candidate, beta extension, or inverse correspondence is used.

M.8 7. Exact limits: finite image versus literal extraspecial group

The actual finite image is P/ker(theta0), because H0 is the scalar preimage. Its center is the scalar cyclic 2-group theta0(H0), and its commutator subgroup has order two. Its quotient by its center is F_2^4.

If theta0(H0) has order two, this finite image is extraspecial of order 32. If its scalar center is larger, quotienting arbitrary central elements need not preserve its commutator: in a cyclic 2-group the order-two subgroup can lie inside the squares. Thus a literal extraspecial quotient is not justified solely by the calculations above.

The stronger statement proved is the one with full actual character: an actual compact subgroup, actual induced carrier, actual scalar subgroup and nondegenerate actual commutator pairing. Passing to the usual full scalar pushout retains all theta0-values. It does not prove that an independently supplied quadratic refinement, stabilizer action, or source phase agrees with this carrier. Those requirements must be checked separately. No H5 or universal claim 66 closure is asserted.

M.9 Provenance

Uses the audited dyadic parameter/character construction, BK 1.4.10, BK 3.1.7–9, and the defining-sequence character formula. The source normalization and all field/chain data are the actual ones in `dyadic167/tower/DYADIC_AS_TOWER167.md`. This note changes neither that proof nor its recorded exact computation. Clifford induction and the identity compact-induction double-coset term are used in their ordinary finite-index forms, with hypotheses proved above.

N Rank-four normalization and induction in stages

This is a corollary of the enlarged-carrier construction in Appendix M.

Use that note's actual groups and notation:

```
n=2a,  r=2m+1,  a>r,
H subset J=J1(beta) normal in P=J U_B^m,
[P:J]=2,  N=K^times J,
pi=c-Ind_N^G(Lambda),  Lambda|J=eta.
```

The field K normalizes P. Consequently the following are actual groups and actual smooth representations:

```
M=K^times P,
Sigma=Ind_N^M(Lambda),
dim Sigma=4,
Sigma|P=Ind_J^P(eta),
pi is isomorphic to c-Ind_M^G(Sigma).
```

The representation Sigma is irreducible, and M is compact modulo the center of G. Thus the enlarged carrier has an exact arithmetic inducing datum for the same candidate, not merely an occurrence in its restriction.

N.1 1. Normalization by the actual parameter field

The units O_K^times equal U_K^1 because the residue field is F_2. They belong to J and hence normalize P. It remains to check a uniformizer k of K. This element has A-grade one and normalizes A.

The approximation estimate proved for the actual field in Section 2 of the carrier note gives

```
k=b+e,  b in Q^1,  e in P_A^(n-r+1).
```

The leading grade of b is the nonzero centralizing line of the primitive derived parameter. Over F_2 it is exactly the leading grade of R, the period-two E-chain shift. Therefore

```
b=R v,  v in 1+Q^1.
```

In particular b is invertible and normalizes both the complete E-chain and the complete interleaved F-chain. We can write

```
k=b(1+epsilon),  epsilon in P_A^(n-r).
```

For u in U_B^m, the usual filtration commutator bound gives

```
(1+epsilon)u(1+epsilon)^(-1)u^(-1)
   in U_A^(n-r+m).
```

This group is contained in U_A^a subset J, because

```
n-r+m=2a-m-1>a.
```

Conjugation by b preserves both relevant filtrations. Since b normalizes U_B^m, it follows that

```
k U_B^m k^(-1) subset U_B^m J=P.
```

Also kJk^(-1)=J, as K^times normalizes its actual simple-character groups. Thus kPk^(-1) is contained in P and contains J. Its index over J is two, equal to [P:J], so it equals P. This proves normalization by k, its inverse, and hence all K^times.

No equality or inclusion between the fields E and K was used. The element b belongs to the actual outer centralizer B and only approximates k to the stated precision.

N.2 2. The two inducing groups have index two

Since K^times normalizes P, the product M=K^times P is a group. Both P/J and the conjugation action of K^times on it are defined. P/J has order two, whose automorphism group is trivial. Therefore

```
[K^times,P] subset J.
```

Together with J normal in P and K^times normalizing J, this proves N=K^times J normal in M. The intersection K^times cap P is U_K^1, which is contained in J, since P subset U_A^1. Consequently

```
[M:N]=[P:J]=2.
```

The quotient $K^\times/F^\times$ is compact because K/F is a finite totally ramified extension, and P is compact. Hence M is compact modulo F^times. It is open in G because it contains the open subgroup J.

N.3 3. The enlarged inducing representation is irreducible

Let u in P represent its nontrivial coset modulo J. The carrier note proved that theta^u differs from theta, and therefore eta^u differs from eta. It follows that Lambda^u differs from Lambda, since their restrictions to J already differ. Applying finite-index Clifford theory to N normal in M gives

```
Sigma=Ind_N^M(Lambda) irreducible,
dim Sigma=2 dim Lambda=4.
```

There is one N–P double coset in M, and N cap P=J. Thus restriction to P gives the actual four-dimensional Heisenberg carrier:

```
Sigma|P is isomorphic to Ind_J^P(eta).
```

Its central character on F^times is unchanged from Lambda, because F^times is central and lies in N.

N.4 4. Exact recovery by induction in stages

Induction in stages for these open subgroups gives

```
c-Ind_M^G(Sigma)
   is isomorphic to c-Ind_M^G(Ind_N^M Lambda)
   is isomorphic to c-Ind_N^G(Lambda)
   =pi.
```

The second induction is finite-index, so ordinary and compact induction from N to M agree. This is an exact equality of the resulting isomorphism classes and applies to the same fixed Lambda, including its noncompact scalar choice. It introduces no new candidate and no H5 premise.

N.5 Scope

This establishes the arithmetic inducing datum and exact recovery for the enlarged carrier. It leaves unchanged the distinction between the original two-dimensional BK quotient J/H and the new four-dimensional actual quotient P/H0. It also leaves unchanged the source-phase and literal extraspecial-center limitations recorded in the carrier note.

Dependency pin: `RANK4_ENLARGED_CARRIER167.md` SHA256 `1517e26138776fe0f32d74a174d309ec5dd6a0f572a979473d8af3d20b7a4155`.

O Exact enlarged-carrier recovery and comparison with the original contract

Evidence: a bounded contract audit and elementary conditional recovery lemmas. This note does not prove that the proposed compact enlargement is a group, that its induced representation is Heisenberg, or that the source markings attach to it. Those are the arithmetic inputs to be verified by the separate rank-two and rank-four constructions.

Verdict: an enlarged compact carrier is not, merely by being Heisenberg or an equivalent type, a new instance of the original H2 quotient. It can serve as a common assembly carrier for the already supplied H2 targets if its actual occurrence, marked attachment, and same-candidate recovery are proved and retained. The recovery mechanism differs from the finite sector idempotent used when the packet lies inside the original inducing carrier.

O.1 1. What original H2 literally requires

Original Section 10 fixes an actual maximal simple character theta, its actual groups H_theta¹ subset J_theta¹ subset J_theta, and

```
V_theta=J_theta¹/H_theta¹.
```

Its Heisenberg representation is the particular eta_theta containing theta. Definition 11.1 supplies an equivariant symplectic isometry from a source envelope to that type-side quotient, together with the central character, and at p=2 the compatible quadratic refinements and central-extension lift. Standard Fact 10.2 separately supplies the canonical beta extension of this simple-character datum.

Consequently the following implications are invalid without more proof:

```
Xi is a finite Heisenberg representation
   therefore Xi=eta_theta' for a maximal simple character theta';

(P,Xi) is equivalent to a known arithmetic type
   therefore it has that type's J¹/H¹ or canonical beta extension.
```

An equivalence of generated representation categories does not identify the particular finite group quotient in Definition 11.1. If a fixed conductor and torsion profile forces every maximal simple-character carrier in the candidate to have dimension one or two, constructing a larger Xi in that candidate does not refute the dimension theorem. Xi is another arithmetic carrier.

There is nevertheless an important distinction between **replacing an H2 target** and **assembling already supplied H2 targets**. The accepted sector construction itself uses a finite sector that need not be a whole maximal simple-character carrier. Its old actual break targets remain unchanged and are attached to that sector. An enlarged P-carrier can play the analogous assembly role, provided the complete dictionary and recovery below are proved. Calling it merely an arbitrary vector space would understate that result. Calling it a new canonical maximal simple type would overstate it.

O.2 2. Precise conditional setup

Let G=GL_n(F), Z=F×I_n, and let the already constructed candidate be

```
pi=c-Ind_N^G Lambda,
```

with its fixed actual extended type and fixed central character omega. Let J be a compact open normal subgroup of N such that

```
eta=Lambda|J is irreducible,       N/J is infinite cyclic.
```

In the proposed F2-residue, full-degree totally ramified setting, J is the actual compact type group J^1=J, not a new group called J merely to fit this notation. Let P be an actual compact open subgroup of G satisfying

```
J subset P,          N intersect P=J.
```

Define the genuine finite-dimensional compact induction

```
Xi=Ind_J^P eta.                                    (1)
```

Neither irreducibility nor Heisenberg structure of Xi is assumed to follow from its dimension. They must be proved separately when needed.

If the known eta intertwining satisfies I_G(eta)=N, then Xi is irreducible: the finite Mackey endomorphism formula gives

```
End_P(Ind_J^P eta)=End_J(eta)=C,
```

because no nontrivial double coset in P lies in I_G(eta). Finite-dimensional smooth representations of compact P factor through a finite group and are semisimple, so scalar endomorphisms imply irreducibility. This criterion requires its actual group and intertwining hypotheses. Dimension alone is not a substitute.

O.3 3. Xi occurs in the same pi, and can be an equivalent compact type

The subspace of the actual compact-induction model of pi consisting of functions supported on NP is stable under P. Mackey restriction identifies that subspace with

```
Ind_(N intersect P)^P(Lambda|_(N intersect P))=Xi.     (2)
```

Thus Xi is an actual compact P-subrepresentation of the same pi, with no change of arithmetic candidate or noncompact scalar.

Induction in stages also gives the exact G-isomorphism

```
c-Ind_P^G Xi isomorphic to c-Ind_J^G eta.            (3)
```

If (J,eta) is already a type for a stated Bernstein block and Xi is irreducible, then (P,Xi) detects the same irreducible representations:

```
Hom_P(Xi,V) isomorphic to Hom_J(eta,V).
```

This is the precise statement about equivalent types that follows from (1). It does not assert that Xi is the eta of another maximal simple character, or attach a canonical beta extension to Xi. The known canonical extension in Lambda remains the one originally supplied.

O.4 4. Exact recovery before fixing the center: a quotient

Choose u in N mapping to a generator of N/J, and retain the actual matrix Lambda(u). Use the model

```
X=c-Ind_J^G eta,
f(jg)=eta(j)f(g),
```

with G acting by right translation. Define the invertible G-equivariant shift operator

```
(Tf)(g)=Lambda(u)^(-1)f(ug).                       (4)
```

Normality of J and the actual intertwining relation for Lambda(u) ensure that Tf has the same J-covariance. Its normalization explicitly retains the fixed noncompact extension. Changing Lambda(u) changes the relation below.

Define

```
(Qf)(g)=sum_(k in Z) Lambda(u)^(-k)f(u^k g).          (5)
```

The sum is finite on each argument and has the required compact support modulo N. It is N-covariant and therefore lies in pi. It is surjective and satisfies Q(Tf)=Q(f). Decomposing support into N/J-orbits reduces its kernel to the elementary augmentation kernel on finitely supported sequences on Z. That kernel consists exactly of consecutive differences. Hence

```
X/(T-1)X isomorphic to pi.                           (6)
```

Through (3), this is a recovery of pi from the enlarged compact carrier plus the original noncompact relation. It does not select the correct relation from the finite Heisenberg packet alone.

In general this quotient cannot be replaced by a G-equivariant split inclusion before central normalization. A nonzero vector of X has finite support modulo compact J. The valuation of its determinant support is a finite subset of Z. A scalar uniformizer z=tI_n shifts these valuations by n. Consequently X has no nonzero eigenvector for z. Every vector of pi is such an eigenvector, with eigenvalue omega(t). Therefore no nonzero G-map pi -> X exists. In particular (6) does not split.

This is why the Block159 sector summand projector cannot simply be quoted: there the finite sector was inside Lambda and its stabilizer contained the noncompact center. Here Xi is larger than Lambda and lies in pi|P, with P compact. These are different induction problems.

O.5 5. After retaining the center: a finite Hecke idempotent

There is also an exact summand version once the already fixed central character is imposed. Put J_Z=ZJ and P_Z=ZP. Assume, as in the proposed totally ramified full-degree setting, that

```
N/J_Z is finite cyclic of order ell
```

(here ell=n). Let Lambda_0=Lambda|J_Z. Because Xi occurs in the same pi, its action on Z intersect P agrees with omega. It therefore extends to P_Z by letting Z act through omega. Call this Xi_omega. Induction in stages gives

```
Y=c-Ind_(P_Z)^G Xi_omega
  isomorphic to c-Ind_(J_Z)^G Lambda_0.              (7)
```

Using the same normalized shift (4), now on Y, one has T^ell=1. Indeed u^ell belongs to J_Z and the J_Z covariance cancels Lambda(u)^ell. The finite operator

```
e_Lambda=(1/ell) sum_(j=0)^(ell-1) T^j               (8)
```

is a G-equivariant idempotent whose image is exactly the fixed pi. To see this directly, N-covariant functions embed in J_Z-covariant functions because N/J_Z is finite. On that embedded pi the finite version of (5) equals ell times the identity. Its average is precisely (8), and its image consists exactly of the N-covariant functions.

Equivalently, Ind_(J_Z)^N Lambda_0 is Lambda tensored with the regular representation of N/J_Z. Formula (8) selects the Lambda summand. It uses the already fixed Lambda(u), including its root choice, so it preserves the same candidate. Other root choices select the other extensions in this finite induction. They are not interchangeable input data.

This is a finite Hecke/induction projector after normalization of the central character. It is not the original finite projector in the image algebra onto a sector of eta. A final artifact must retain (7), (8), and the normalization Lambda(u), rather than describing recovery merely as “the carrier is an equivalent type.”

O.6 6. The exact remaining admissibility checks

For either proposed low-rank enlargement, the new proof must supply all of the following concrete inputs:

1. **Actual compact group:** prove that the proposed P is a subgroup, is compact open, contains the actual J, and satisfies P intersect N=J. Compute [P:J] and the resulting Xi dimension.
2. **Actual carrier:** verify eta=Lambda | J, its actual intertwining if the irreducibility criterion is used, and the Mackey realization (2).
3. **Actual finite Heisenberg structure:** exhibit the scalar preimage in P and the resulting elementary abelian projective quotient. Compute its represented commutator and prove nondegeneracy. An irreducible carrier of dimension two or four need not automatically have this structure.
4. **Original attachments:** attach the unchanged supplied actual H2 break targets, retaining their full source character values and dyadic finite extension/lift/square markings. The precise distinction between transported and native markings remains the one proved in `DYADIC_MARKINGS167.md`.
5. **Recovery data:** retain the actual Lambda, central character and noncompact generator normalization, and prove the isomorphisms and quotient/idempotent above for those exact groups.

When these are supplied, the result is stronger than occurrence of an unrelated finite vector space: it is a marked arithmetic carrier in the same candidate, equivalent through explicit induction and recovering that candidate exactly. For C={1} and one vertex, it can repair the failed low-rank **assembly-carrier** realization under the retained scalar-pushout contract. The original H2 targets remain their original simple-character quotients. They are not renamed Xi.

If the intended target instead requires the source packet itself to equal J_theta'[1]/H_theta'[1] for a maximal simple character in this same candidate, the enlargement does not solve it. The retained low-rank dimension obstruction still applies. A separate theorem identifying Xi with such an eta would be needed, and under the fixed obstructed profile it cannot be inferred from type equivalence.

No H5 condition is added or discharged. If H5 is subsequently supplied, it must concern this fixed recovered candidate. Nontrivial stabilizer actions, canonical native quadratic comparison, and arbitrary transport graphs remain outside this bounded argument.

O.7 Evidence and self-review

Read directly: original Section 10 and Definition 11.1 in the raw manuscript; the Block159 `NONINVARIANT_SECTOR_RECONSTRUCTION159.md`; current `PROOF167.md` and `TARGET167.md`; the retained dyadic marking theorem. The Mackey, induction, shift-quotient, and central-normalized projector arguments above are proved conditionally for the displayed groups, rather than assumed for the proposed congruence formula. No actual low-rank P computation or new Lean execution is claimed in this contract note.

Reading scope

This guide follows the chapter's mathematical route from finite Frobenius packets to finite wild comparison, then explains the actual arithmetic constructions developed in the detailed odd-prime and dyadic construction appendices.

The general arithmetic realization endpoint is stated as **Conjecture 66**. The comparison with the established local Langlands correspondence remains a theorem conditional on an actual realization and H5 for that same candidate.

All mathematical artwork is deterministic Python/Matplotlib vector output. The complete editable source and its figure generator accompany this guide.

How to use this companion

Read the first two parts as an atlas of finite mathematical objects. The third part explains where finite objects acquire an actual arithmetic carrier. The final part separates the remaining realization conjecture from the conditional comparison theorem.

Each spread develops one construction or one distinction. Displays preserve the normalizations used in the accompanying mathematical text. The source line identifies an original manuscript section or a retained proof note. These are retained source anchors, rather than a claim that every argument has been newly formalized. The separate research atlas provides the later reconciliation of conductors, occurrence and native markings.

The symbol K denotes the original local field in the discussion of the finite theory. In the explicit arithmetic constructions, $F = \mathbb{F}_p((t))$ is the base field and $K = F[\beta]$ is its totally ramified parameter field. A symplectic quotient dimension and a Heisenberg representation dimension are different quantities.

Theorem numbering. The historical proof obligation called "66" corresponds to the endpoint printed as Theorem 16.2 in the original manuscript. Conjecture 66 now isolates its general arithmetic realization requirement. It is not a claim that row 66 of the historical theorem register is this endpoint.

Supplementary materials. The source collection includes detailed proofs and construction notes. Their scope distinguishes established results, conditional results and the remaining obligations of Conjecture 66.

Evidence. The odd-prime and dyadic constructions have ordinary mathematical proofs using the stated classical type-theory results. Their bounded exact computations support specified identities. They are not fresh Lean certificates for the arbitrary-length arithmetic theorems. The separate Lean guide states the actual formalization scope.

Two visual routes, one claim map

The core visual guide explains the mathematical objects and restricted arithmetic constructions. The research atlas develops the comparison tests, failed implications and remaining certificate. Read the two as complementary routes through the argument.

Start with the question	Continue to the linked reader
What is being realized?	Monograph realization contract and Plate XI.1. Finite comparison and arithmetic existence are different assertions.
How does recovery preserve the candidate?	Monograph restricted constructions and Plate XI.10. A changed assembly carrier does not mean a changed inducing representation.
Why do numerical tests not settle attachment?	Progress Sections 22–25 and Plates XI.11–XI.19. The full and skipped rank-eight profiles have capacities six and four despite the same three numerical invariants.
What does a dyadic marking retain?	Progress Sections 26–29 and Plates XI.20–XI.29. Squares, section products, native choices and phase data must not be collapsed into one commutator form.
What is formally checked?	Lean guide: the existence input and the conditional signature. A supplied field in a structure is not constructed by typesetting its declaration.

Keep the five reader PDFs together. These links use the actual companion pages, while the original lessons keep their source-note and section anchors. No diagram in either guide certifies arithmetic inputs that its source argument leaves open.

Reading corrections without changing the contract

One candidate throughout. A source-defined induction candidate and a directly constructed type candidate are not identified merely by notation. Occurrence, full-character normalization and the noncompact extension must concern the same representation. The Part II occurrence analysis retains the excluded original route and the corrected route's remaining obligation.

A transported mark is not a prescribed native mark. The positive scalar-pushout attachment carries the supplied finite marking through a center-fixing homomorphism. It does not identify that output with an independently fixed host subgroup. Conversely, the stronger native problem must not be inserted as a new premise into the different transported-marking result.

The equal-capacity example is conditional. At rank eight the full profile has exactly six quotient dimensions. The finite Arf test applies when the actual native refinement is independently prescribed. The abstract six-dimensional forms in the research atlas are examples, not a computation of that candidate's native square bits.

A successful finite repair has inputs. The transvection starts with an allowed symplectic map and an isotropic discrepancy vector. The odd-action correction starts with genuine supplied finite actions. Neither constructs the arithmetic normalizer or chooses every honest phase. See Plates XI.25–XI.29.

Evidence and scope

The finite illustrations and ordinary construction arguments retain their stated hypotheses and evidence boundaries. The formal companion records the separate compiler replay and its scope. A PDF build does not certify arithmetic constructors.

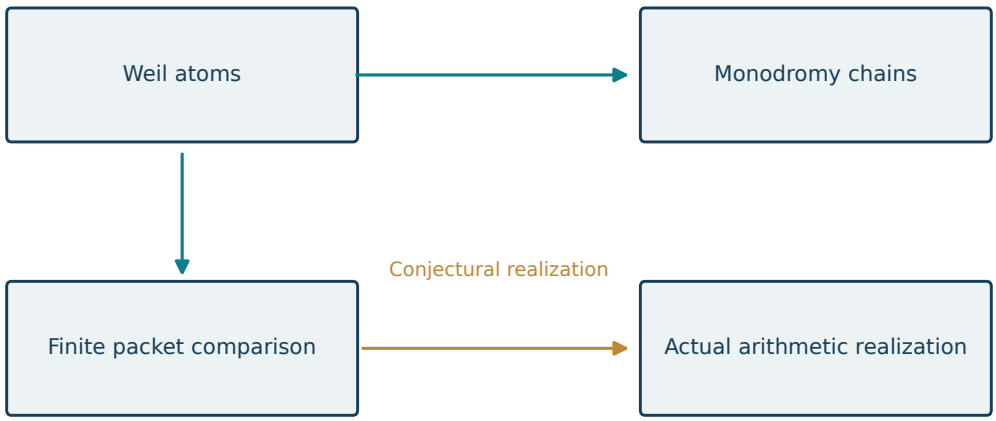
The source ledger distinguishes recorded verification from the open arithmetic endpoints. The retained source audit documents the checked Henniart–Lemaire interface and its normalization conventions. The complete editable archive for Block 2 remains unavailable. Its pages in the research reader are preserved. No external peer review is claimed.

For the final same-candidate comparison, read the monograph comparison proposition alongside Plate XI.32. The conjecture itself does not assert the independent H5 step.

PART I / FINITE TAME COORDINATES

1 Three mathematical objects

Source: Original manuscript, Sections 2, 11, 16; contract audit 162.



The finite comparison does not create the arithmetic witness.

An atom, its finite packet data, and its arithmetic realization have different roles.

A Weil atom is an irreducible Weil representation. A finite packet records selected quotient spaces, projectors, central extensions, phases, and transports. An arithmetic realization is an actual admissible representation of the required general linear group together with a comparison and recovery of that packet.

Monodromy organizes an atom into a finite Weil–Deligne chain. It does not supply an atom that is missing. Likewise a finite Heisenberg comparison identifies already supplied finite representations, rather than automatically constructing a representation of $\mathrm{GL}_n(K)$.

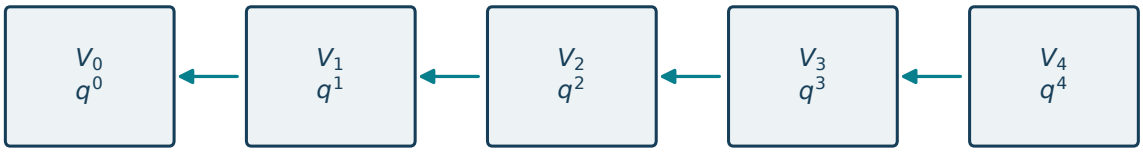
Reading the diagram. The final existence arrow is the content of Conjecture 66. Finite comparison, arithmetic construction, and local-factor characterization are separate steps.

PART I / FINITE TAME COORDINATES

2 Monodromy is a finite transfer chain

Source: Original Section 2.

$$F = \mathrm{diag}(1, q, q^2, q^3, q^4), \qquad FNF^{-1} = q^{-1}N$$



$$N(V_j) \subseteq V_{j-1}; \qquad N^5 = 0$$

The arrows point from V_j to V_{j-1}, the direction of N.

Put $V = \bigoplus_{j=0}^{m-1} V_j$ and let P_j be the coordinate projector. If $N(V_j) \subseteq V_{j-1}$ and $N(V_0) = 0$, then

$$N = \sum_{j=1}^{m-1} P_{j-1} N P_j, \qquad N^m = 0.$$

Every term in N^r follows a path of length r down the chain. There is no path of length m. With $F = \mathrm{diag}(1, q, \dots, q^{m-1})$, each nonzero transfer block gains the conjugation factor $q^{j-1}q^{-j} = q^{-1}$.

Reading the diagram. Nilpotence and covariance follow from the displayed block structure. Assigning the initial supercuspidal atom is a separate arithmetic input.

PART I / FINITE TAME COORDINATES

3 Fourier projectors isolate tame inertia

Source: Original Section 3; geometric Frobenius convention.

Let t generate a finite tame quotient of order M, with $(q, M) = 1$. For $j \in \mathbb{Z}/M\mathbb{Z}$ define

$$P_j = \frac{1}{M} \sum_{a=0}^{M-1} \zeta_M^{-ja} r(t)^a, \qquad \zeta_M = e^{2\pi i/M}.$$

The finite Fourier identities give $P_j P_k = \delta_{jk} P_j$ and $\sum_j P_j = I$. Thus $P_j V$ is the ζ_M^j character space for $r(t)$. The guide uses geometric Frobenius with

$$\Phi^{-1} r(t) \Phi = r(t)^q.$$

Equivalently $\Phi r(t) \Phi^{-1} = r(t)^{q^{-1}}$, where the inverse is taken modulo M. Substituting in the Fourier sum gives

$$\Phi P_j \Phi^{-1} = P_{qj}.$$

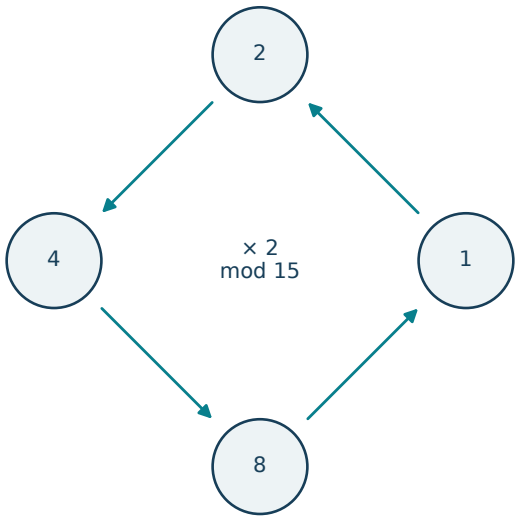
Scope. If the forward relation $\Phi r(t) \Phi^{-1} = r(t)^q$ is used instead, the projector label is $q^{-1}j$. The two directions have the same orbit sets but different conjugation identities. The geometric Frobenius convention is used consistently here.

Reading the diagram. A packet is an actual eigenspace selected by an idempotent. Frobenius carries it to the next character packet.

PART I / FINITE TAME COORDINATES

4 One orbit gives one tame atom

Source: Original Theorem 3.2.



For $q = 2$ modulo 15, the displayed geometric projector orbit is $(1, 2, 4, 8)$.

For an irreducible tame Weil representation, a union of Frobenius orbits of nonzero inertia packets would give a stable subspace. Irreducibility therefore forces one orbit, of some length f.

The stabilizer of a packet is $W_{K_f} = \langle I_K, \Phi^f \rangle$, where K_f/K is the unramified extension of degree f. Inertia acts on the packet by one scalar character. Its remaining stabilizer action is generated by Φ^f. An irreducible complex action of this cyclic quotient is one dimensional. Hence

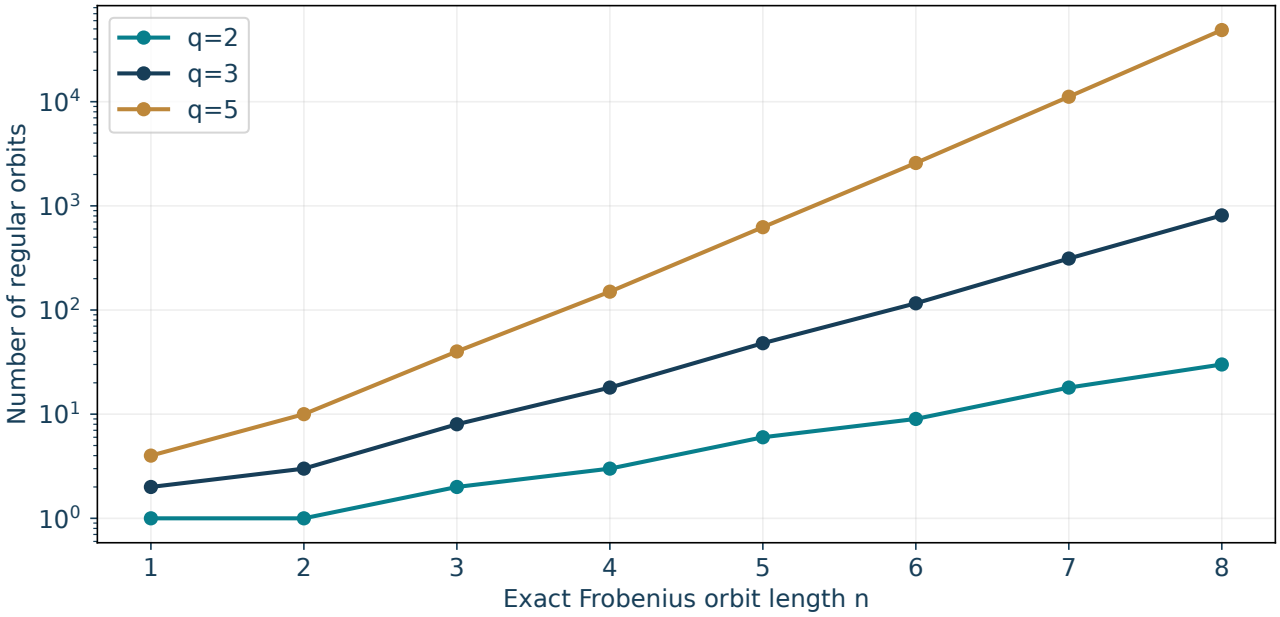
$$\tau \simeq \operatorname{Ind}_{W_{K_f}}^{W_K} \xi, \qquad \dim \tau = f.$$

Reading the diagram. Orbit transitivity supplies both the dimension and the induced model. It is a theorem about the tame case.

PART I / FINITE TAME COORDINATES

5 Regular orbits have exact counts

Source: Original Proposition 3.6.



Exact integer counts. The vertical scale is logarithmic.

Characters of $\mathbb{F}_{q^n}^{\times}$ are indexed modulo $q^n - 1$. Characters fixed by a Frobenius power of degree $r \mid n$ number $q^r - 1$. Möbius inversion removes those belonging to smaller orbits:

$$c_n(q) = \frac{1}{n} \sum_{d \mid n} \mu(d)(q^{n/d} - 1).$$

For $q = 2, n = 4$, the regular exponent orbits modulo 15 are

$$(1, 2, 4, 8), \quad (3, 6, 12, 9), \quad (7, 14, 13, 11).$$

Their number is $\frac{1}{4}(15 - 3) = 3$. The plotted values are computed by the exact divisor sum and retained in `figure_data.json`.

Reading the diagram. Regularity is an exact orbit property. The plot illustrates an integer formula rather than supplying its proof.

PART I / FINITE TAME COORDINATES

6 A primitive projector selects a cuspidal block

Source: Original Section 4.

Let $G_n = \mathrm{GL}_n(\mathbb{F}_q)$ and $T_n \simeq \mathbb{F}_{q^n}^\times$ be an elliptic torus. A regular torus character determines an irreducible cuspidal representation κ_θ by the classical finite-field construction. If its character is χ_θ and degree is d_θ, then

$$e_\theta = \frac{d_\theta}{|G_n|} \sum_{g \in G_n} \chi_\theta(g^{-1}) g$$

is its primitive central idempotent in $\mathbb{C}[G_n]$.

On an irreducible representation with character ψ, this central element acts by a scalar. Taking traces and using character orthogonality makes the scalar 1 on κ_θ and 0 on every other irreducible block.

For a proper parabolic $P = LU$, averaging over its unipotent radical gives

$$e_U = |U|^{-1} \sum_{u \in U} u, \qquad e_U e_\theta = e_\theta e_U = 0.$$

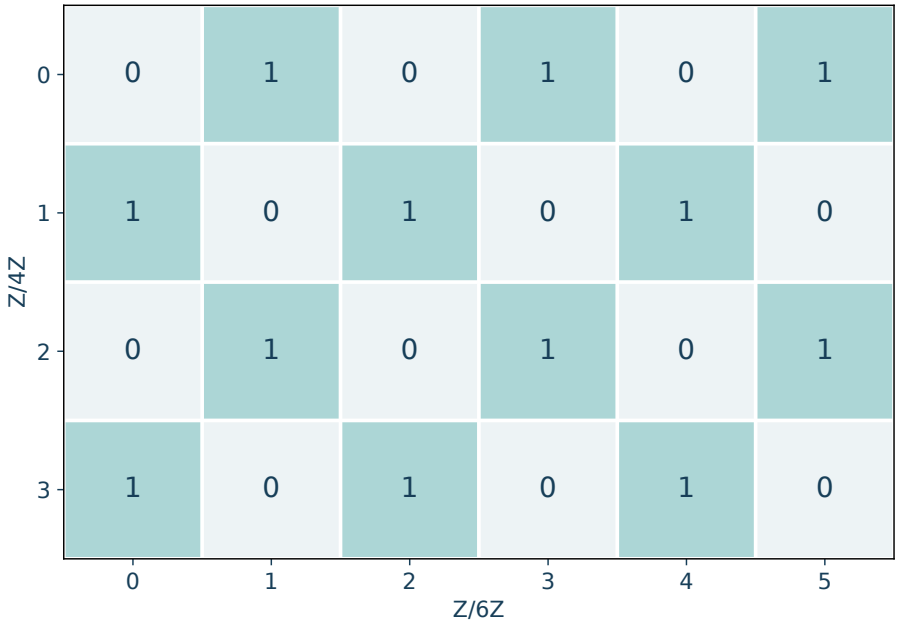
The average selects U-fixed vectors. The cuspidal block has none.

Scope. The existence and classification of κ_θ are classical inputs. The idempotent calculation explains its exact finite selector.

PART I / FINITE TAME COORDINATES

7 Tensor refinement is a diagonal orbit problem

Source: Original Theorem 5.1.



The 4×6 grid separates into two diagonal translation orbits, labeled 0 and 1.

The simultaneous translation action on $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ has

$$d = \gcd(n, m) \quad \text{orbits, each of length} \quad L = \operatorname{lcm}(n, m).$$

Mackey decomposition then gives

$$\operatorname{Ind}_{W_{K_n}}^{W_K} \xi \otimes \operatorname{Ind}_{W_{K_m}}^{W_K} \eta \simeq \bigoplus_{i=0}^{d-1} \operatorname{Ind}_{W_{K_L}}^{W_K} \left(\xi_{L,n} \eta_{L,m}^{\Phi^i} \right).$$

Here the subscripts denote norm lifts and the Frobenius superscript follows the fixed source convention. The dimension identity is exact: $dL = nm$.

Scope. These are interaction packets before combining isomorphic summands. A refined induced summand need not remain irreducible.

PART I / FINITE TAME COORDINATES

8 Exponent coordinates test inertia triviality

Source: Original Proposition 5.3.

Write the residue character exponents as $a \bmod (q^n - 1)$ and $b \bmod (q^m - 1)$. The ith refined packet has exponent

$$c_i = a\frac{q^L - 1}{q^n - 1} + bq^i\frac{q^L - 1}{q^m - 1} \pmod{q^L - 1}.$$

Norm lifting multiplies an exponent by the quotient of group orders. Shifting the second character by Frobenius multiplies its exponent by q^i. Multiplication of characters adds the two terms.

For $q = 2, n = 4, m = 6$, the common modulus is $2^{12} - 1 = 4095$. Taking $a = b = 1$, the two packets have

$$c_0 = 273 + 65 = 338, \qquad c_1 = 273 + 130 = 403.$$

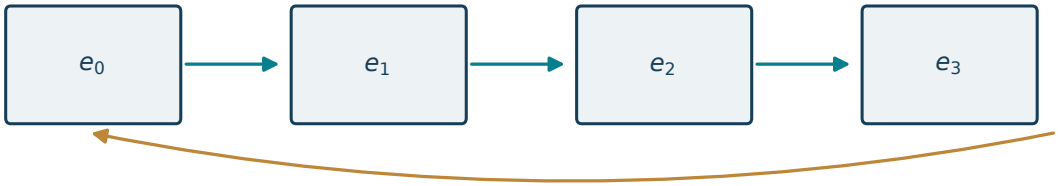
Both are nonzero modulo 4095, so neither packet contributes inertia invariants.

Reading the diagram. The test is $c_i = 0$ in the common modulus. A numerical-looking residue formula is an exact character calculation.

PART I / FINITE TAME COORDINATES

9 The closing scalar carries noncompact information

Source: Original Propositions 3.5 and 4.4.



Closing edge a: characteristic polynomial $X^4 - a$

The closing scalar is independent of the basis after multiplication around the cycle.

In an orbit basis let $\Phi e_j = e_{j+1}$ for $j < n-1$ and $\Phi e_{n-1} = a e_0$. Then

$$\det(XI - \Phi) = X^n - a, \qquad \det \Phi = (-1)^{n-1} a.$$

Rescaling the individual basis vectors changes edge coefficients, but their product remains a. An unramified twist ν does not change the inertia orbit and sends the closing scalar to $a\nu(\Phi)^n$.

Reading the diagram. Compact inertia information alone does not determine the unramified completion. The cyclic scalar is additional data.

PART I / FINITE TAME COORDINATES

10 Local polynomials and fixed vectors

Source: Original Propositions 5.4–5.8.

For a packet of orbit length r and closing scalar a,

$$\det(I - u\Phi \mid V^{I_K}) = \begin{cases} 1 - au^r, & \text{if inertia acts trivially,} \\ 1, & \text{otherwise.} \end{cases}$$

Thus tame tensor local factors multiply $(1 - a_i u^{r_i})^{-1}$ over precisely the refined packets whose inertia exponent vanishes.

An inertia-trivial companion cycle has a Weil-fixed vector exactly when $a_i = 1$, and then its fixed space is one dimensional. Therefore

$$\dim(\tau_{\xi^\vee} \otimes \tau_\eta)^{W_K} = \#\{i : c_i = 0,\ a_i = 1\}.$$

For an irreducible nontrivial tame atom of dimension $n > 1$, wild inertia acts trivially and inertia has no fixed vectors. Hence

$$\mathrm{Sw}(\tau) = 0, \qquad \mathrm{Art}(\tau) = n.$$

Scope. These exact tame formulas do not establish the full wild pair-factor identities required by H5.

PART I / FINITE TAME COORDINATES

11 Finite Gauss phases are unit scalars

Source: Original Section 6.

For a nontrivial additive character ψ of $\mathbb{F}_q$ and a nontrivial multiplicative character χ of $\mathbb{F}_q^\times$, define

$$G(\chi,\psi)=\sum_{x\in\mathbb{F}_q^\times}\chi(x)\psi(x),\qquad \gamma(\chi,\psi)=q^{-1/2}G(\chi,\psi).$$

The exact identities are

$$|G(\chi,\psi)|^2=q,\qquad \gamma(\chi,\psi)\gamma(\chi^{-1},\psi)=\chi(-1).$$

The magnitude calculation substitutes $x=ty$ in the double sum. The inner sum is $q-1$ for $t=1$ and -1 otherwise. Orthogonality of multiplicative characters supplies the result.

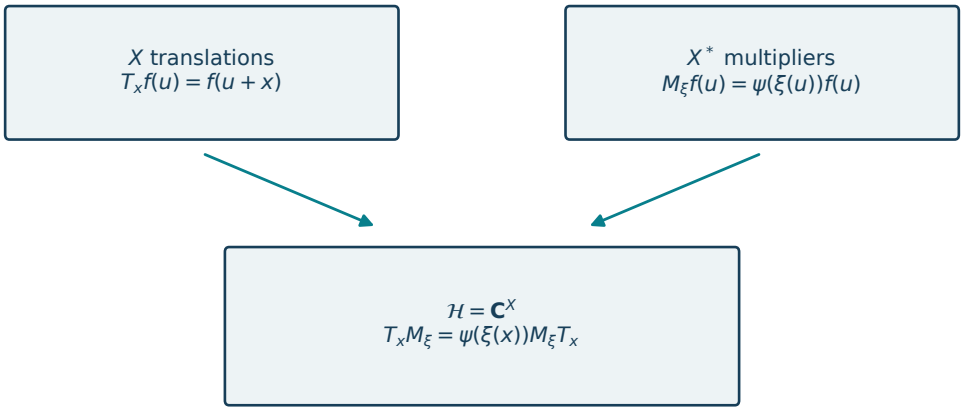
The normalized scalar therefore lies on the unit circle. Its value depends on the declared additive character and lifting normalization. Finite Gauss phases capture a part of normalization arising from the residue field.

Scope. A full wild local factor also contains ramification depth, conductor of the additive character, measure normalization, and pair interaction data. One finite phase is not that full factor.

PART II / FINITE WILD COMPARISON

12 Translations and multipliers create Heisenberg packets

Source: Original Section 7.



Noncommuting operators generate the full matrix algebra.

The commutator is central and nontrivial when the pairing is nonzero.

For odd p let $V = X \oplus X^*$ and

$$b((x,\xi),(y,\eta)) = \eta(x) - \xi(y).$$

The group $H(V) = V \times \mathbb{F}_p$ has product

$$(v,z)(w,t) = (v+w, z+t+\tfrac{1}{2}b(v,w)).$$

On $\mathcal{H} = \mathbb{C}^X$ the Schrödinger operator is

$$(\rho(x,\xi,z)f)(u) = \psi(z+\xi(u)+\tfrac{1}{2}\xi(x))f(u+x).$$

Translations and character multipliers generate every matrix unit by finite Fourier inversion. Thus the representation is irreducible and has dimension $|X| = \sqrt{|V|}$.

PART II / FINITE WILD COMPARISON

13 An isotropic average selects one line

Source: Original Propositions 7.3–7.6.

The Schrödinger character is supported on the center:

$$\operatorname{Tr}\rho(v,z)=\begin{cases}|X|\psi(z), & v=0,\\ 0, & v\neq 0.\end{cases}$$

A nonzero translation has no fixed coordinate. A nontrivial multiplier has vanishing character sum.

For a maximal isotropic subspace L and a subgroup lift $\widetilde{L}$ with trivial central coordinate, the average

$$P_L=\frac{1}{|L|}\sum_{\ell\in\widetilde{L}}\rho(\ell)$$

is a self-adjoint idempotent. Its trace is $|X|/|L|=1$, because only the identity contributes. Its range is therefore one dimensional.

For $L=X$ this range consists of constant functions:

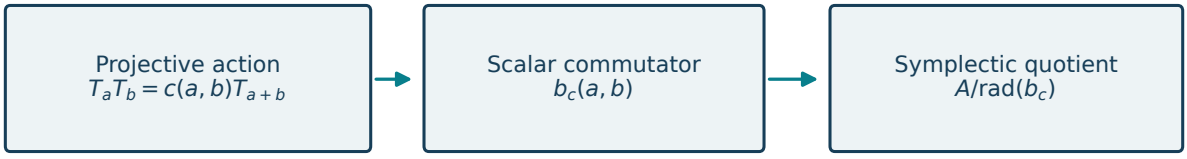
$$(P_Lf)(u)=|X|^{-1}\sum_{x\in X}f(u+x).$$

Reading the diagram. The projector onto the complementary line later used in arithmetic attachment is an exact operator in the Heisenberg image algebra. Its existence is finite representation theory.

PART II / FINITE WILD COMPARISON

14 Projective choices leave a commutator invariant

Source: Original Section 8.



Radical directions commute with every projective generator.

A cocycle contains choices. Its alternating commutator forgets scalar rescaling.

For $T_aT_b = c(a,b)T_{a+b}$ define $b_c(a,b) = c(a,b)c(b,a)^{-1}$. Replacing T_a by $\lambda_a T_a$ multiplies c by the symmetric coboundary $\lambda_a\lambda_b\lambda_{a+b}^{-1}$, so b_c is unchanged.

The radical consists of a commuting projectively with every b. Its quotient carries a nondegenerate alternating pairing. In the nondegenerate case the twisted group algebra has scalar center and is a full matrix algebra:

$$\mathbb{C}_c[A] \simeq M_{\sqrt{|A|}}(\mathbb{C}).$$

Scope. This constructs the finite envelope for a supplied projective action. It does not produce a canonical action for every arbitrary wild inertia atom.

PART II / FINITE WILD COMPARISON

15 Clifford multiplicity is the correct action space

Source: Original Section 9.

Supply finite groups $N \triangleleft G$, elementary abelian quotient $A = G/N$, and irreducibles Π and ρ, with ρ fixed up to isomorphism by G and $\Pi|_N$ isotypic. The multiplicity space is

$$M_\rho = \mathrm{Hom}_N(W, X), \qquad W \otimes M_\rho \simeq X.$$

For a section $s : A \to G$ and intertwiners I_a of ρ, write

$$\Pi(s(a)) = I_a \otimes T_a.$$

The operators T_a give a projective action on M_ρ. Section and intertwiner changes alter it only by scalar changes and conjugation. Its commutator, radical, and symplectic quotient are intrinsic to this supplied datum.

If the commutator is nondegenerate, $\dim M_\rho = \sqrt{|A|}$. A tame stabilizer acting by automorphisms of the datum preserves the induced symplectic form.

Scope. The normal group, isotypic constituent, and one-break interpretation are assumptions of the restricted construction. They have not been constructed for every wild atom.

PART II / FINITE WILD COMPARISON

16 The type-side quotient is an actual group quotient

Source: Original Section 10.

For an actual maximal simple character θ of $\mathrm{GL}_n(K)$, classical simple-type theory supplies

$$H_\theta^1 \subseteq J_\theta^1 \subseteq J_\theta, \qquad V_\theta = J_\theta^1 / H_\theta^1.$$

The commutator pairing is encoded by

$$\psi_0(h_\theta(\bar{x}, \bar{y})) = \theta([x, y]).$$

It is nondegenerate and alternating. The corresponding irreducible Heisenberg representation η_θ of J_θ^1 contains θ and satisfies

$$\dim \eta_\theta = \sqrt{|J_\theta^1 / H_\theta^1|}.$$

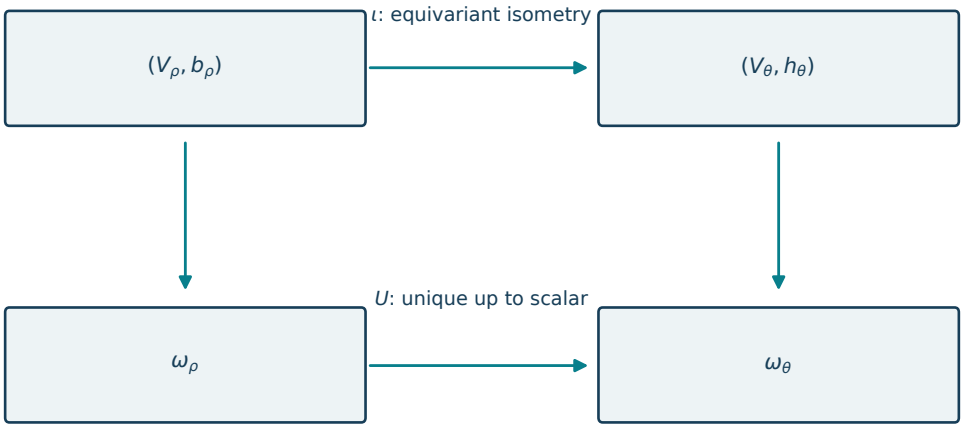
The canonical beta extension supplies a normalized extension of this representation using the declared type-theory data and finite corrections.

Scope. An arbitrary abstract symplectic space is not automatically a quotient of an actual simple character. An equivalent inducing carrier need not have this original quotient or a newly invented canonical beta extension.

PART II / FINITE WILD COMPARISON

17 A supplied bridge reduces comparison to a scalar

Source: Original Section 11.



A cyclic generator phase removes the discrepancy character.

The symplectic isometry is an input. Stone–von Neumann supplies the lower intertwiner.

Supply an equivariant symplectic isometry $\iota : V_\rho \to V_\theta$, a common nontrivial central character, and, in characteristic two, the compatible refinements and lift of central extensions.

The two pulled-back irreducible Heisenberg representations then have the same central character. Stone–von Neumann gives an invertible U with

$$U\omega_\rho(h) = \omega_\theta(\iota_H(h))U.$$

Schur's lemma makes U unique up to a nonzero scalar. After honest cyclic stabilizer linearizations, their discrepancy is a character $\delta : C \to \mathbb{C}^\times$. Equality at one generator kills this character.

Scope. Finite linearization does not identify the stabilizer action with an arithmetic action on a newly constructed common candidate.

PART II / FINITE WILD COMPARISON

18 Odd phases have a determinant certificate

Source: Original Section 12.

For an odd prime p, a symplectic space V of dimension $2r$, and fixed-point-free $g \in \mathrm{Sp}(V)$, the declared finite Weil normalization gives

$$\Phi_p(g, V) = \left(\frac{-1}{p}\right)^r \left(\frac{\det(g - I)}{p}\right).$$

With $P_g(X) = \det(XI - g)$, even dimension gives $\det(g - I) = P_g(1)$. The phase is therefore read from a characteristic polynomial and a square class.

For a hyperbolic space $U \oplus U^*$ with $g = (a, (a^{-1})^*)$ and $a - I$ invertible,

$$\Phi_p(g, U \oplus U^*) = \left(\frac{\det a}{p}\right).$$

If g has order prime to p, then

$$V = \ker(g - I) \perp \mathrm{im}(g - I).$$

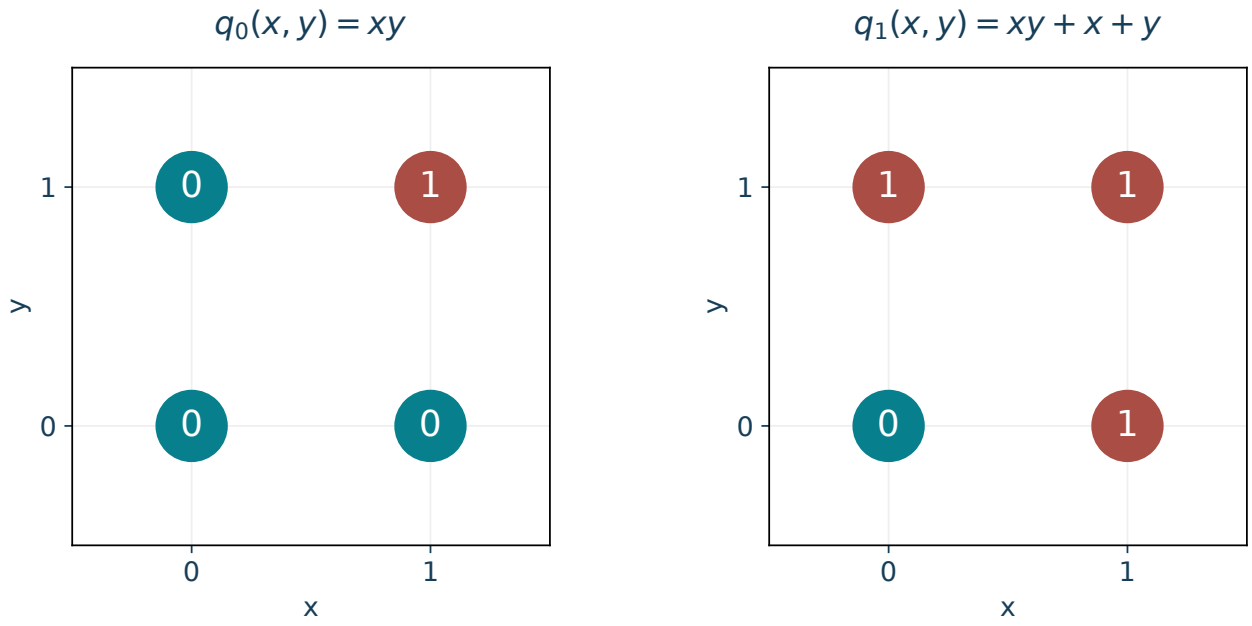
The moving summand is nondegenerate and fixed-point-free. Orthogonal block phases multiply.

Scope. The fixed-point-free formula applies to the moving part. The full trace with a fixed part requires its declared normalization data.

PART II / FINITE WILD COMPARISON

19 Two dyadic refinements can have opposite signs

Source: Original Section 13.



The entries are exact values in $\mathbb{F}_2$. Both polar forms are $xy' + x'y$.

A quadratic refinement satisfies $q(x+y) + q(x) + q(y) = b(x,y)$. On a plane,

$$q_0(x,y) = xy, \qquad q_1(x,y) = xy + x + y.$$

For q_0 there are three zeros and one one, so its Gauss sum is 2. For q_1 there is one zero and three ones, so its sum is -2. The alternating forms agree but the Arf invariants differ.

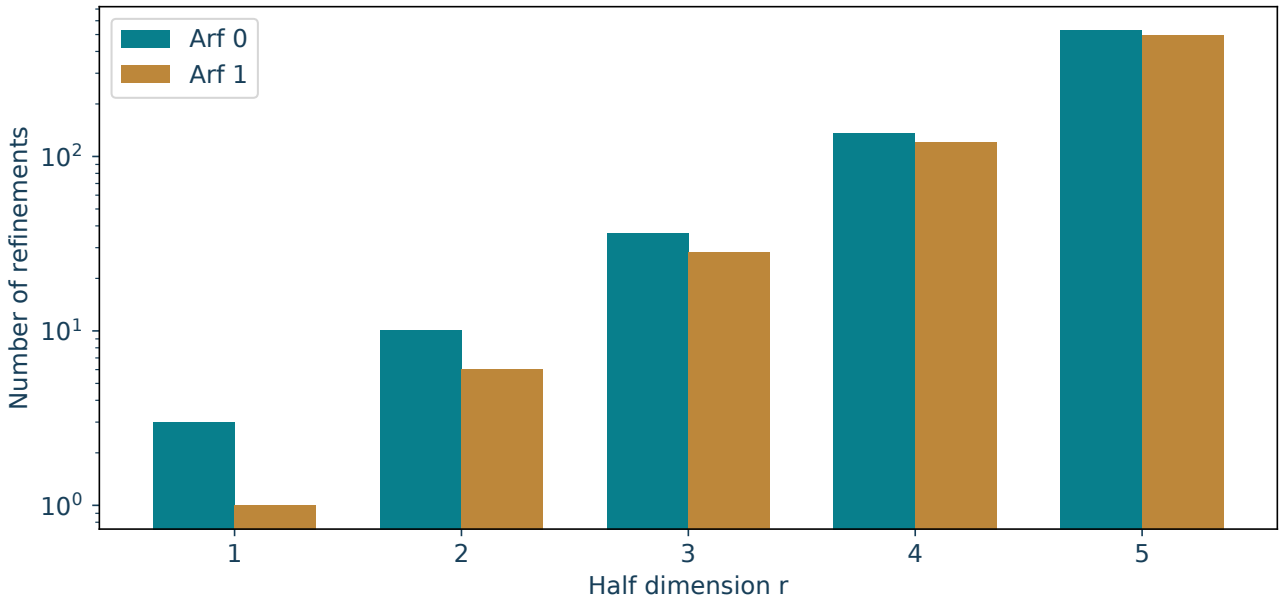
$$\sum_{x \in V} (-1)^{q(x)} = (-1)^{\mathrm{Arf}(q)} 2^r, \qquad \dim V = 2r.$$

Reading the diagram. Keeping the alternating form alone loses dyadic phase information. The refinement and the marked central extension must travel with the packet.

PART II / FINITE WILD COMPARISON

20 Refinements form an affine family

Source: Original Propositions 13.2 and 13.8–13.9.



Exact counts of refinements of a fixed nondegenerate alternating space.

The difference between two refinements is linear. Nondegeneracy identifies every linear form with $b(v,\cdot)$, so all refinements are

$$q_v(x) = q(x) + b(v,x), \qquad \mathrm{Arf}(q_v) = \mathrm{Arf}(q) + q(v).$$

The numbers of the two Arf types are

$$2^{r-1}(2^r+1), \qquad 2^{r-1}(2^r-1).$$

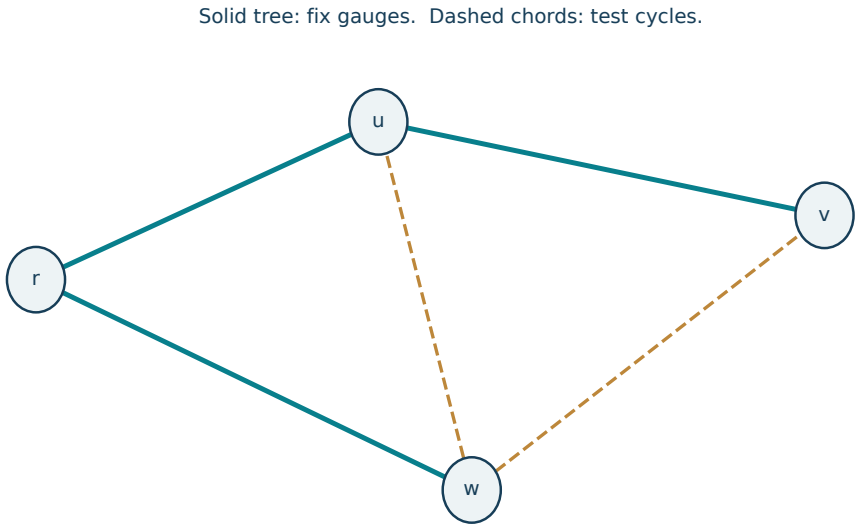
For $r = 1$ these are 3 and 1. For $r = 3$ they are 36 and 28. Orthogonal sums add Arf invariants and multiply normalized Gauss signs.

Reading the diagram. Equal dimension and equal alternating form do not force equal refinement. An embedding fixed in advance must preserve q pointwise.

PART II / FINITE WILD COMPARISON

21 A spanning tree fixes the vertex gauges

Source: Original Theorem 14.2.



Four vertices and five edges leave two fundamental cycle tests.

Let $A_e : V_u \to V_v$ and $B_e : W_u \to W_v$ be invertible transports, with reverse-edge inverses. Fix a root comparison U_r.

Choose a spanning tree. Along its unique root path p_v define

$$U_v = B_{p_v} U_r A_{p_v}^{-1}.$$

This forces every tree-edge equation $U_v A_e = B_e U_u$. Each non-tree edge closes one fundamental cycle. Its edge equation holds exactly when the root comparison intertwines that cycle's holonomy.

Reading the diagram. All edge equations are equivalent to the fundamental cycle identities $U_r A_\gamma = B_\gamma U_r$. Holonomy need not be individually trivial. The two holonomy representations must agree through the root map.

PART II / FINITE WILD COMPARISON

22 Cycle defects measure edge residuals exactly

Source: Original Proposition 14.6.

For a non-tree edge $e : u \to v$, let

$$D_e = U_r A_{p_v}^{-1} A_e A_{p_u} - B_{p_v}^{-1} B_e B_{p_u} U_r.$$

Substituting the tree-defined gauges gives the exact identity

$$U_v A_e - B_e U_u = B_{p_v} D_e A_{p_u}^{-1}.$$

Consequently

$$\|U_v A_e - B_e U_u\| \le \|B_{p_v}\| \, \|D_e\| \, \|A_{p_u}^{-1}\|.$$

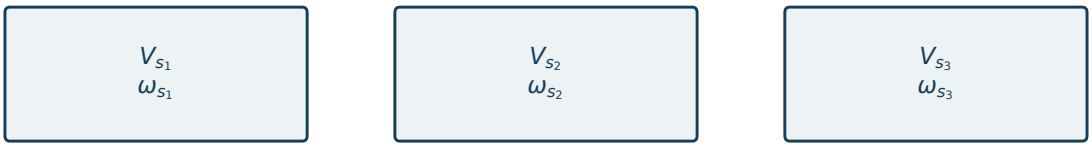
Small cycle defects give small edge residuals when the tree transports are conditioned. Zero defects give exact finite gluing. In the scalar case there are $|E| - |V| + 1$ independent tests on a connected graph.

Scope. The theorem assumes both sets of fibers and transports already exist. Gauge propagation is not an arithmetic existence theorem.

PART II / FINITE WILD COMPARISON

23 Orthogonal breaks become tensor factors

Source: Original Section 15.



$$V = V_{s_1} \perp V_{s_2} \perp V_{s_3}$$

$$\omega_V \simeq \omega_{s_1} \otimes \omega_{s_2} \otimes \omega_{s_3}$$

Symplectic direct sum and Heisenberg tensor product are compatible.

For $V = \perp_s V_s$ and compatible central characters,

$$\omega_V \simeq \bigotimes_s \omega_{V_s}, \qquad \prod_s \sqrt{|V_s|} = \sqrt{|V|}.$$

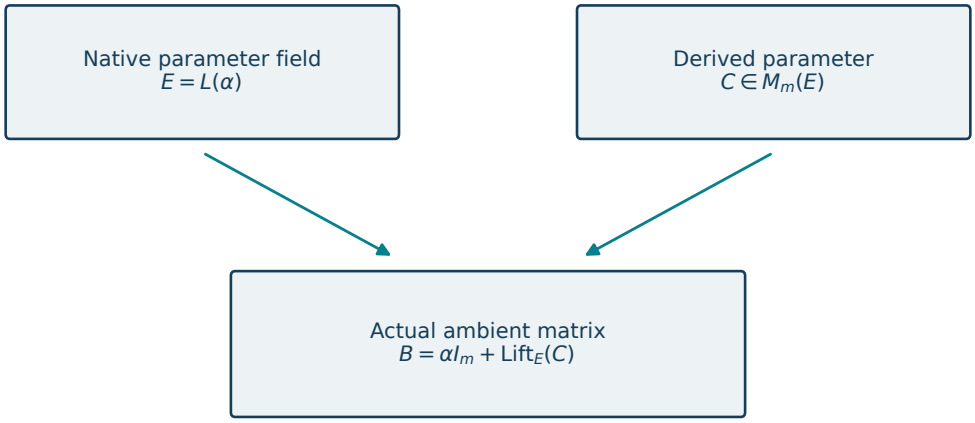
The groups combine by a central product. Tensoring the individual intertwiners gives the full finite bridge, once the stabilizer actions and phases agree. Compatibility on the transport graph is then the matrix holonomy condition.

Scope. Separate actual type quotients at each break do not, merely by tensoring, become one actual maximal simple type in the source atom's rank. Constructing a common arithmetic carrier and recovering a candidate are the extra realization requirements.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

24 A recursive matrix gives a literal parameter

Source: Odd-prime and dyadic constructions: tower notes.



The derived field and the final field are not identified.

The recursive lift is an actual filtered right inverse of normalized corestriction.

Start over a coefficient field L with uniformizer t_L and form

$$E = L(\alpha), \qquad \alpha^p - \alpha = t_L^{-q}.$$

At a recursive stage, C is an actual matrix over E. The outer matrix is

$$B = \alpha I_m + \mathrm{Lift}_E(C).$$

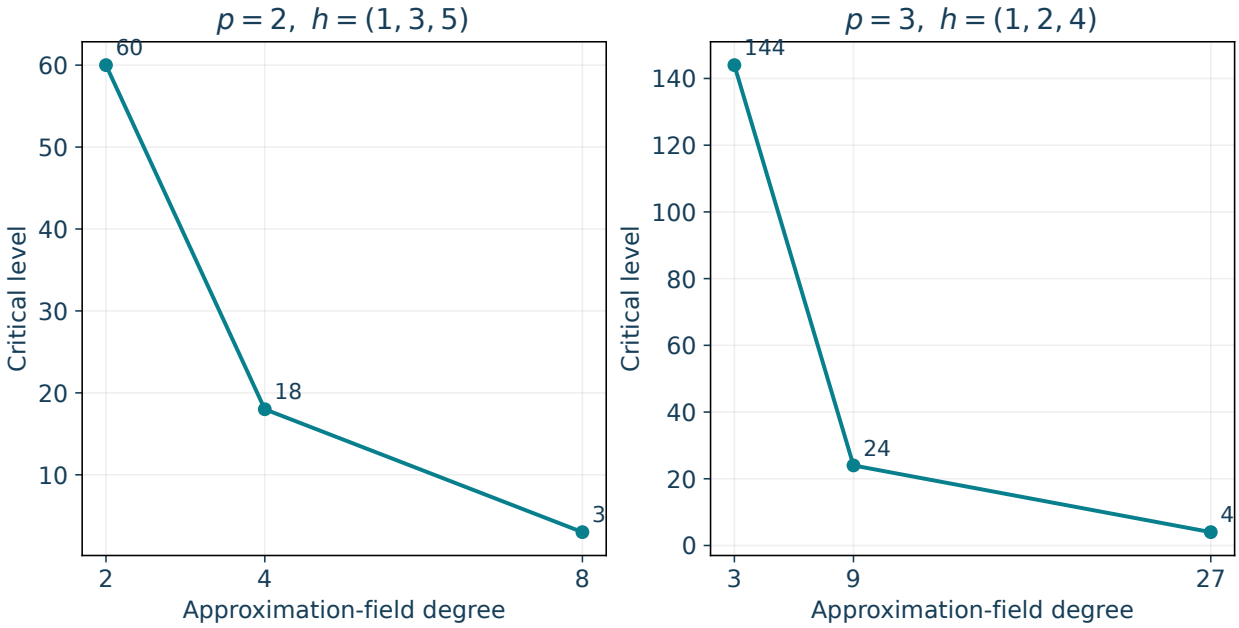
The lift has prescribed corestriction and preserves its exact filtration valuation. The field and simple-stratum proof uses the actual displayed matrix, together with the cited type-theory derived-stratum criterion.

Scope. The derived field $E[C]$ is not identified with the final field $L[B]$. The recursion proves a literal defining sequence without silently declaring intermediate fields to be nested inside the final field.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

25 The approximation degrees and critical levels

Source: Odd-prime and dyadic constructions: combined arithmetic statements.



These examples were recomputed exactly. The assertions for all lengths are proved separately.

For increasing permitted $h_1 < \cdots < h_d$, put $q_i = (p+1)h_i$ and $N = p^d$. The approximation-field degrees are $p, p^2, \ldots, p^d$, with

$$n_j = p^{d-j} q_{d-j+1}.$$

For odd p, require $p \nmid h_i$. For $p = 2$, require all h_i odd and positive. The leading ambient level is n_1, and $n_2, \ldots, n_d$ are the successive critical approximation levels.

Scope. Every amplified ambient approximation still has leading level n_1. The declining values plotted here are critical levels, not new leading valuations for those ambient matrices.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

26 Odd towers retain native trace zero

Source: Odd-prime construction: recursive tower and full-character proofs.

For odd p, $\mathrm{Tr}_{E/L}(\alpha) = 0$. Choose b with $qb \equiv -1 \pmod{p}$, let $a = (qb+1)/p$, and set $z = t_L^a \alpha^b$. Then z is a uniformizer.

For its Eisenstein polynomial f and constant-coordinate projection P_0, the normalized corestriction uses $\kappa = -f'(z)/z^{p-1}$. Writing $u_0 = s(P_0)$,

$$\mathrm{Lift}_E(C) = u_0^{-1} C P_0, \qquad \mathrm{tr}_L \,\mathrm{Lift}_E(C) = \mathrm{Tr}_{E/L}(\kappa^{-1}\,\mathrm{tr}_E\, C).$$

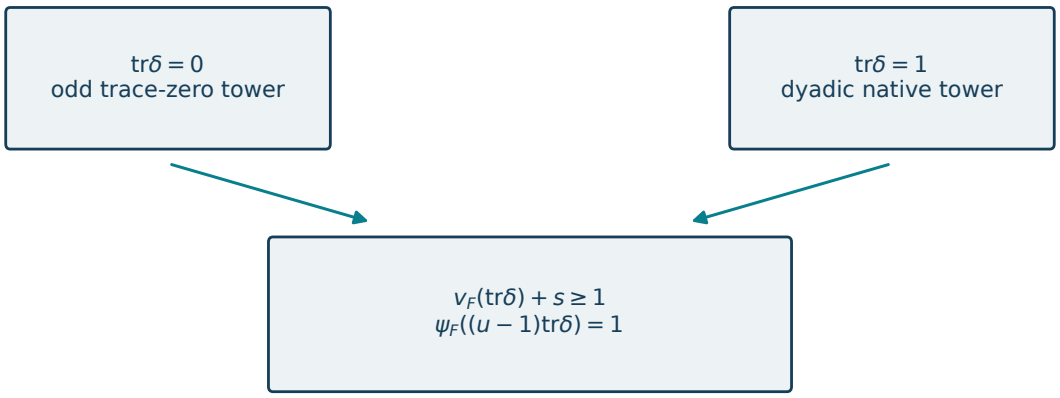
If $\mathrm{tr}_E\, C = 0$, the lift has trace zero. Adding αI_m preserves trace zero. Induction therefore gives trace zero at every native stage and every native correction.

Reading the diagram. This is stronger than the automatic ambient trace cancellation from a multiplicity divisible by p. Native trace controls the scalar restriction used in full-character construction.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

27 Dyadic trace one is deep enough

Source: Dyadic construction: depth invariant.



The scalar correction disappears when its argument lies in $t\mathcal{O}_F$.

Fix ψ_F of conductor $t\mathcal{O}_F$. On a scalar unit $u \in U_F^s$, the additive correction sees $(u-1)\,\mathrm{tr}\,\delta$. Hence

$$v_F(\mathrm{tr}\,\delta) + s \geq 1 \quad \Longrightarrow \quad \psi_F((u-1)\,\mathrm{tr}\,\delta) = 1.$$

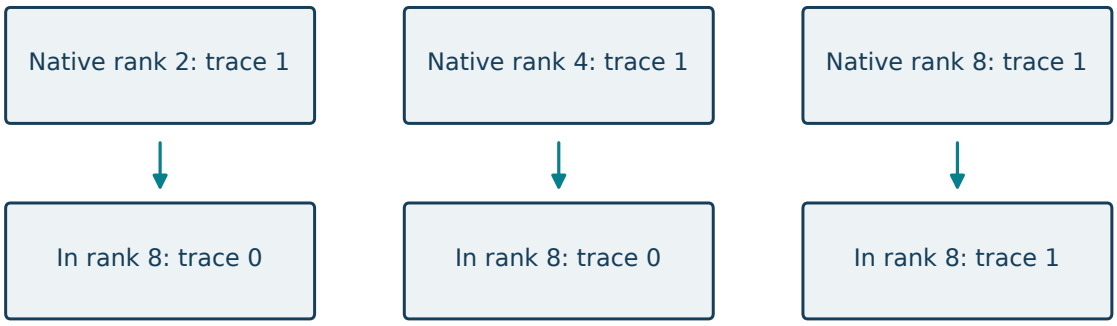
The dyadic native corrections have trace 1. Its valuation is 0, so $s = 1$ already satisfies the inequality. Thus trace zero is sufficient but unnecessary.

Scope. Euler's weighted-visibility viewpoint motivated this step. No Euler coefficient theorem is an arithmetic proof input here. In characteristic two, the weights are uniformizer depths, not integer powers of two in the coefficient field.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

28 Native trace and ambient trace are different

Source: Dyadic construction: exact trace transport.



Native trace and amplified ambient trace have different roles.

A trace-one native matrix amplified by an even multiplicity has ambient trace zero.

For odd q, let $a = (q+1)/2$ and $z = t_L^a \alpha$. Then

$$z^2 + t_L^a z + t_L = 0, \quad \mathrm{Tr}_{E/L}\,\alpha = 1, \quad \mathrm{Lift}_E(C) = (z^2/t_L)CP_0.$$

The trace identity is

$$\mathrm{tr}_L\,\mathrm{Lift}_E(C) = \mathrm{Tr}_{E/L}(\alpha\,\mathrm{tr}_E\,C).$$

It sends inner trace 1 to outer trace 1. At native rank 2^j, every constructor has trace 1. Its next native correction has trace $1 - 2 \cdot 1 = 1$. In rank 8, the earlier amplified approximations instead have traces $0, 0$, and the final matrix has trace 1.

Reading the diagram. Only the appropriate native trace establishes the scalar-character compatibility at each recursive stage.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

29 Full characters require shallow compatibility

Source: Odd-prime and dyadic constructions: central character induction.

The full-character induction has an essential order. First construct the full native character, then transfer it to a higher multiplicity, add the actual correction on its forced domain, and extend at the next maximal native parameter field.

On a scalar $u \in U_F^1$, multiplicity p transfer gives

$$\Theta_j(uI) = \theta_j(u^p) = 1.$$

The correction is trivial on the scalar intersection by trace zero in the odd case, or integral trace and the depth inequality in the dyadic case. The forced deep character therefore agrees with the imposed trivial scalar character on their intersection. A finite abelian unit-character extension supplies the full next character.

For an odd dyadic critical level r, put $m = \lfloor r/2 \rfloor$. The endpoint satisfies $2(m+1) - r = 1$, making the additive cross term lie in the character's radical.

Scope. Checking a leading trace alone does not check shallow forced values. The earlier fixed-field central argument failed on an actual scalar unit. The recursive native-field proof repairs that problem.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

30 The same arithmetic candidate has exact profiles

Source: Combined statements 166–167.

Let $N = p^d$ and retain the constructed full character, canonical beta extension, and compatible central completion. Both families give an actual irreducible admissible supercuspidal candidate of $\mathrm{GL}_N(F)$ with torsion number 1.

For odd p,

$$\mathrm{Sw}(\pi) = p^{d-1}(p+1)h_d,$$
$$\dim_{\mathbb{F}_p}(J^1/H^1) = N-1, \quad \dim \eta = p^{(N-1)/2},$$
$$a(\pi^\vee \times \pi) = p^{2d} - 1 + \sum_{i=1}^{d}(p^{2i} - p^{2i-2})h_i.$$

For $p = 2$,

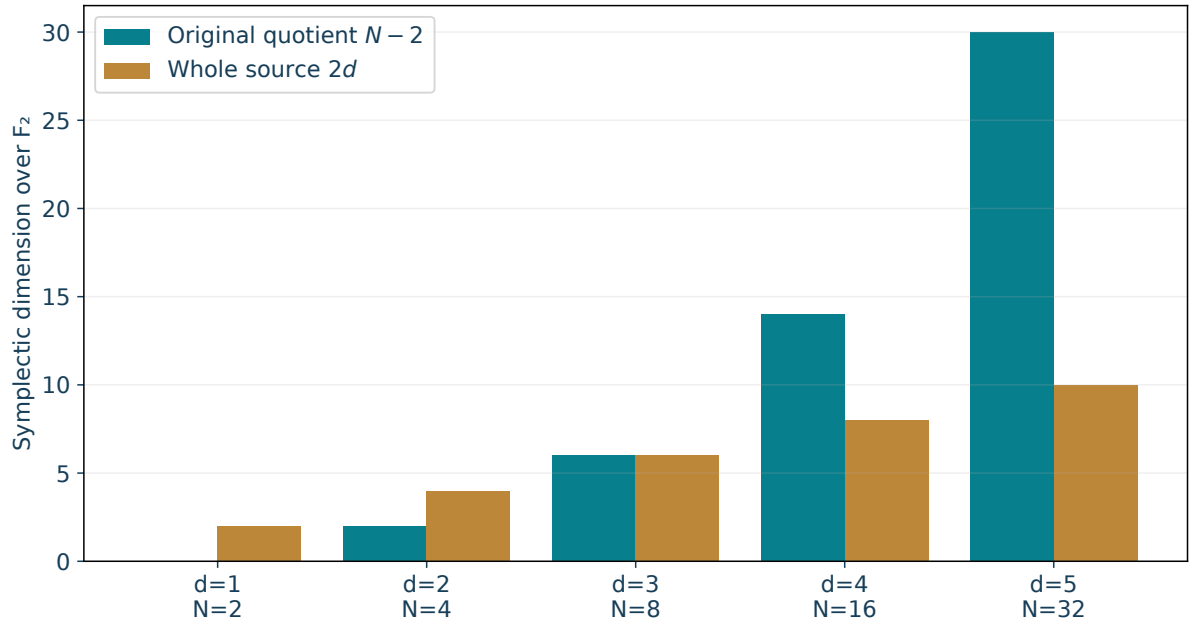
$$\mathrm{Sw}(\pi) = 3\,2^{d-1}h_d,$$
$$\dim_{\mathbb{F}_2}(J^1/H^1) = N-2, \quad \dim \eta = 2^{N/2-1},$$
$$a(\pi^\vee \times \pi) = N^2 - 1 + \sum_{i=1}^{d} 3\,4^{i-1}h_i.$$

Scope. These identities belong to the same constructed candidate. Matching numerical profiles does not identify a source Weil atom, an endo-class, or H5 pair factors.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

31 Dyadic capacity has two distinct dimensions

Source: Dyadic construction: capacity proof.



Compare symplectic dimensions first. The original low-rank quotient remains too small.

The host symplectic quotient has dimension $N-2$ over $\mathbb{F}_2$, and the supplied whole source of d planes has dimension $2d$. Its original-type embedding is possible by dimension exactly when $d \geq 3$.

N	host quotient	source quotient	host carrier	source carrier
2	0	2	1	2
4	2	4	2	4
8	6	6	8	8
16	14	8	128	16

Scope. The low-rank obstruction concerns the original maximal-simple-character carrier. Enlarged actual arithmetic carriers repair ranks 2 and 4 without changing this quotient theorem.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

32 A sector can retain the whole source packet

Source: Odd-prime and dyadic constructions: scalar-pushout attachment.

Suppose the supplied full break characters have labeled plane quotients and satisfy the one-vertex scalar-pushout assembly contract, initially with $C = 1$. Their source representation has dimension $p^d = N$.

For odd p, the actual leading subspace has dimension $(p-1)p^{d-1} \geq 2d$. For dyadic $d \geq 3$, use the host quotient dimension $N - 2 \geq 2d$. Choose the declared symplectic embedding into the actual host.

An orthogonal splitting $W = V_{\mathrm{src}} \perp V_{\mathrm{src}}^{\perp}$ gives

$$\eta_W \simeq \eta_{\mathrm{src}} \otimes \eta_{\perp}.$$

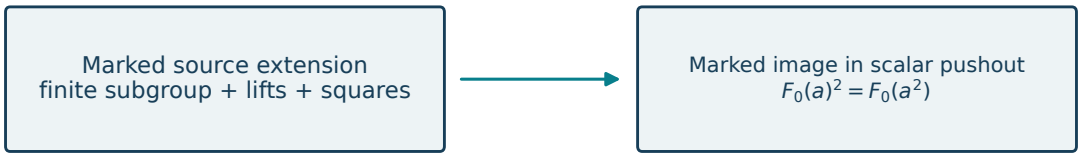
A maximal-isotropic average on the complementary Heisenberg factor selects a one-dimensional representation line. Tensoring its projector with the identity gives a rank-N projector in the image algebra. The retained induction/recovery theorem recovers the same Λ and then the same π.

Scope. For $p > 3$, genuinely supplied plane data are required: a native minimal rank-p quotient has dimension $p - 1$ and is not automatically a plane. The construction retains source labels but does not prove their old numerical depth placement.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

33 Retain the whole marked finite extension

Source: Dyadic construction: markings proof.



The whole finite subgroup travels through the map.

A separately fixed native marking needs its own comparison.

Squares are preserved by the actual map of marked extensions.

For a full character θ_i, its scalar pushout is

$$\mathcal{P}_i = (\mathbb{C}^\times \times J_i^1)/\{(\theta_i(h)^{-1}, h) : h \in H_i^1\}.$$

A center-fixing comparison retains every supplied character value. In the dyadic case retain the image of the entire marked finite subgroup, its central embedding, lifts, squares, refinements, and phases. An actual extension map F_0 obeys $F_0(a)^2 = F_0(a^2)$.

Scope. The unmarked complex pushouts of D_8 and Q_8 coincide, while their marked finite μ_2 extensions differ. Retaining only the central $\{\pm 1\}$ loses that distinction. Invisible kernel elements already removed by the central character cannot be recovered by pushout.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

34 A native quadratic marking adds an embedding condition

Source: Dyadic construction: markings theorem.

Transported source markings and an independently prescribed native host marking are different inputs. If the native host refinement q_W is independently supplied, a prescribed embedding f must satisfy

$$q_W(f(v)) = q_{\mathrm{src}}(v) \quad \text{for every } v.$$

With a free embedding and equal dimensions, quadratic classification requires equal Arf invariants. This is the rank-eight case: both quotient spaces have dimension 6.

For $d \geq 4$, the dyadic host has at least one spare symplectic plane. The complement's Arf type can absorb the difference, so either source Arf type admits an embedding into the supplied native quadratic host.

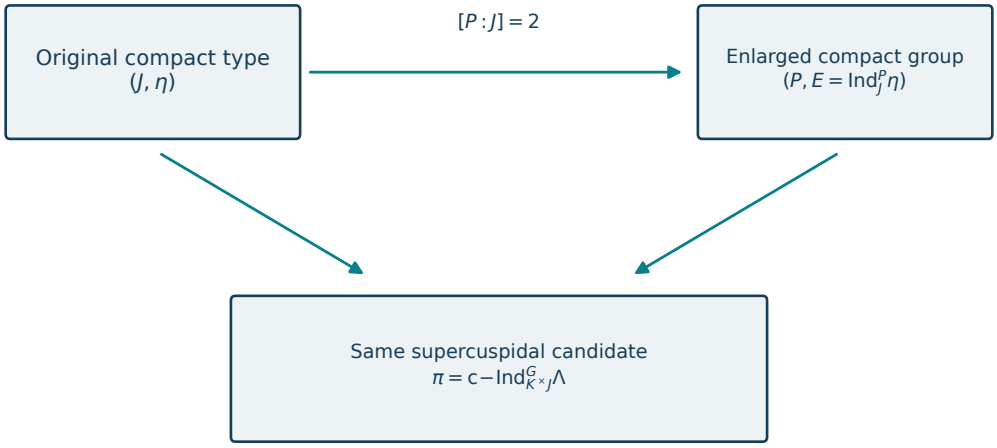
Reading the diagram. Dimension controls symplectic room. Arf controls a quadratic isometry class. A pointwise equation controls an already chosen embedding.

Scope. Capacity does not manufacture a canonical native finite marking or identify it with the transported marking. That comparison remains outside the general realization proof.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

35 Low-rank repair enlarges an actual compact group

Source: Dyadic construction: low-rank constructions.



The new carrier occurs in the original candidate. It is not an abstract extra tensor factor.

At rank two, write $q = 2m + 1$, $m \geq 1$, and use

$$J = U_K^1 U_A^{m+1}, \qquad P = U_K^1 U_A^m.$$

Then $J \triangleleft P$, $[P : J] = 2$, and $P \cap (K^\times J) = J$. The conjugate character ratio $\lambda = \theta^g / \theta$ is nontrivial and sign-valued. With $H_0 = \ker \lambda$,

$$P / H_0 \simeq \mathbb{F}_2^2, \qquad E = \mathrm{Ind}_J^P \theta, \quad \dim E = 2.$$

The represented diagonal sign and off-diagonal generators anticommute, proving the nondegenerate Heisenberg structure.

Scope. The original quotient J^1 / H^1 is still zero-dimensional. The new pair is a separate arithmetic assembly carrier.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

36 Rank four gains a four-dimensional carrier

Source: Dyadic construction: rank-four enlarged carrier proof.

Let E_0/F be the actual outer quadratic approximation field and $B = \mathrm{End}_{E_0}(F^4)$. With lower odd critical level $r = 3h_1 = 2m + 1$ and half leading level $a = 3h_2 > r$, define

$$P = J\,U_B^m.$$

The estimate for the filtered centralizer proves this is a compact group with J normal of index two. It does not assume $E_0 \subset K$.

For a new coset representative u, the ratio $\lambda = \theta^u/\theta$ on the actual H^1 is nontrivial and sign-valued. Setting $H_0 = \ker \lambda$ gives

$$P/H_0 \simeq \mathbb{F}_2^4, \qquad E = \mathrm{Ind}_J^P \eta, \quad \dim E = 4.$$

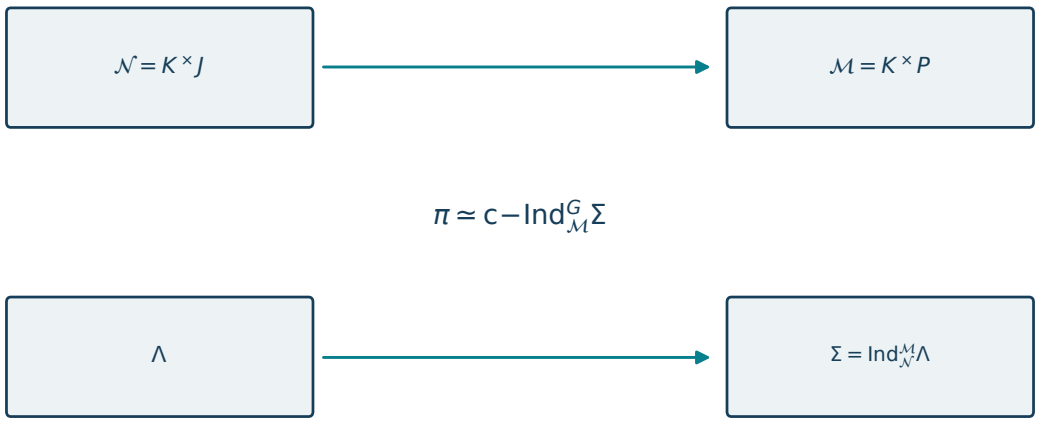
Its represented commutators are scalar, and the quotient has exponent two. Irreducibility and the exact scalar preimage prove nondegeneracy. The finite scalar center can exceed μ_2.

Scope. No literal extraspecial quotient is claimed. Full character values and marked attachment are retained through the actual finite extension.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

37 Induction in stages recovers the same candidate

Source: Dyadic construction: low-rank recovery proof.



The noncompact extension Λ is fixed before forming Σ.

At both low ranks, $K^\times$ normalizes P. Put

$$\mathcal{N} = K^\times J, \quad \mathcal{M} = K^\times P, \qquad \Sigma = \mathrm{Ind}_{\mathcal{N}}^{\mathcal{M}} \Lambda.$$

The subgroup $\mathcal{N}$ is normal of index two, $\Sigma|_P \simeq E$, and Σ is irreducible. Induction in stages proves

$$\pi \simeq \mathrm{c\text{-}Ind}_{\mathcal{M}}^{G} \Sigma.$$

Reading the diagram. The parameter, full character, original canonical beta-extension data, central character, and noncompact scalar remain those of the original candidate. The inducing carrier changes.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

38 Compact recovery must remember the noncompact root

Source: Dyadic construction: enlarged-carrier contract.

Let $X = \mathrm{Ind}_J^G \eta$ and retain $A_0 = \Lambda(\varpi_K)$. Define the normalized shift

$$(Tf)(g) = A_0^{-1} f(\varpi_K g).$$

The original candidate is the quotient

$$X/(T-1)X \simeq \pi.$$

This does not split before central normalization: a vector of X has finite support modulo compact J, whereas a scalar uniformizer shifts determinant valuations. It has no nonzero eigenvector with the required central eigenvalue.

After imposing the already fixed central character, the relevant quotient is cyclic of order N. Then $T^N = 1$ and the finite average

$$e_\Lambda = \frac{1}{N} \sum_{i=0}^{N-1} T^i$$

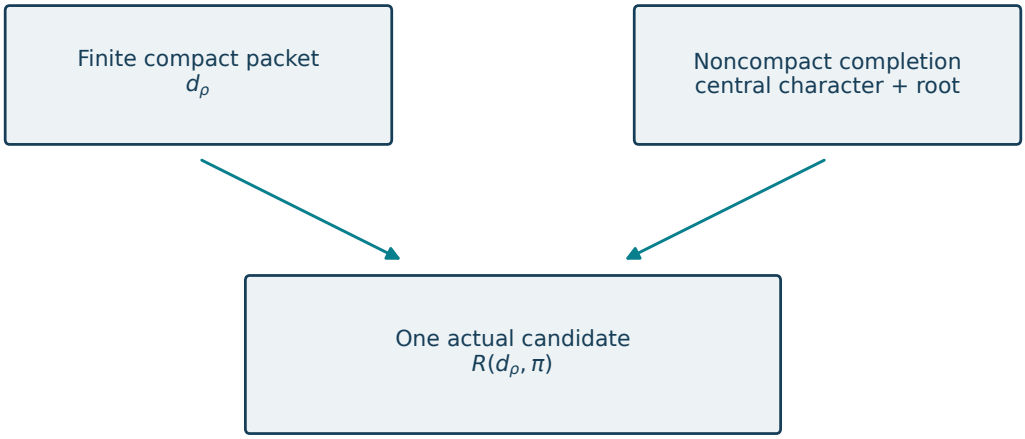
selects the same extension. The operator acts on a complex representation space, where division by N is valid.

Scope. This is a different recovery mechanism from a finite sector inside the original η. The root is retained, not reconstructed from the compact packet alone.

PART III / ACTUAL ARITHMETIC CONSTRUCTION

39 Central completion is a construction choice

Source: Odd-prime and dyadic constructions: full-character completion.



This diagram starts with an already constructed compatible arithmetic host.

The full character is trivial on U_F^1. Residue data and the noncompact scalar can then match a prescribed smooth central quasicharacter ω trivial on U_F^1, when completion choices are free or already compatible. For $F = \mathbb{F}_2((t))$, the residue unit group is trivial, so these covered central characters are unramified. An Nth root in $\mathbb{C}^\times$ fixes the action required by $\omega(t)$. All choices are recorded before invoking H5.

Scope. An incompatible immutably supplied completion is not silently replaced. Nor is an arbitrary one-factor wild determinant assumed trivial. The source central character must belong to the covered class.

Reading the diagram. A root choice completes an existing candidate. It does not prove the general realization conjecture or the same candidate's pair-factor equalities.

PART IV / REALIZATION AND COMPARISON

40 Why nontrivial stabilizer actions need another step

Source: Odd-prime construction: stabilizer contract.

For the current totally ramified degree-p^r maximal-type host, character-preserving normalizers have

$$N/(F^\times J^1) \simeq \mathbb{Z}/p^r\mathbb{Z}.$$

A native action of a finite prime-to-p group C on the quotient therefore factors through a p-group and is trivial.

The kernel of the action of center-fixing scalar-extension automorphisms on W is $\mathrm{Hom}(W, \mathbb{C}^\times)$, an elementary abelian p-group. Hence the full native scalar-extension action is trivial as well. Honest implementers on an irreducible Heisenberg carrier are scalar by Schur's lemma.

Scope. This is an obstruction to the native inner-normalizer route for this host. It does not refute the original finite contract or Conjecture 66. Original breakwise H2 may supply a genuinely nontrivial symplectic action on separate actual type targets. Transporting abstract operators does not by itself identify a common arithmetic action.

PART IV / REALIZATION AND COMPARISON

41 Genuine scalar phases can descend from a finite cover

Source: Odd-prime construction: scalar native attachment.

There is a proved extension beyond $C = 1$ when the prime-to-p stabilizer acts trivially on the source extension and its honest operators on the entire rank-N source carrier are $\chi(c)I_N$ in the actual inertia image.

Under the same-source determinant central-character condition, the required scalar phase can be implemented by actual central matrices after a finite cover $\widetilde{C} \twoheadrightarrow C$. The kernel acts trivially on the whole recovered candidate, so the represented action descends to C.

Reading the diagram. A cover may be needed even when the represented operator is a scalar. A direct matrix homomorphism from C need not exist.

Scope. The theorem requires a genuine inertia phase on the whole carrier, the fixed reciprocity convention, and compatible completion. It does not apply merely because an arbitrary abstract phase on a multiplicity subspace is scalar. It does not change a fixed canonical beta extension to force a desired action.

PART IV / REALIZATION AND COMPARISON

42 Numerical matching does not construct realization

Source: Profile and quotient-capacity comparisons.

For compact $J \subset \mathrm{GL}_n(F)$, the homomorphism $v_F \circ \det$ has finite image in $\mathbb{Z}$, hence image zero. An unramified character

$$\nu_a(g) = a^{v_F(\det g)}$$

is therefore trivial on J. The compact packet restrictions of π and $\pi \otimes \nu_a$ agree, while the central character changes by a^n on a scalar uniformizer. Choosing $a^n \neq 1$ gives a different atom.

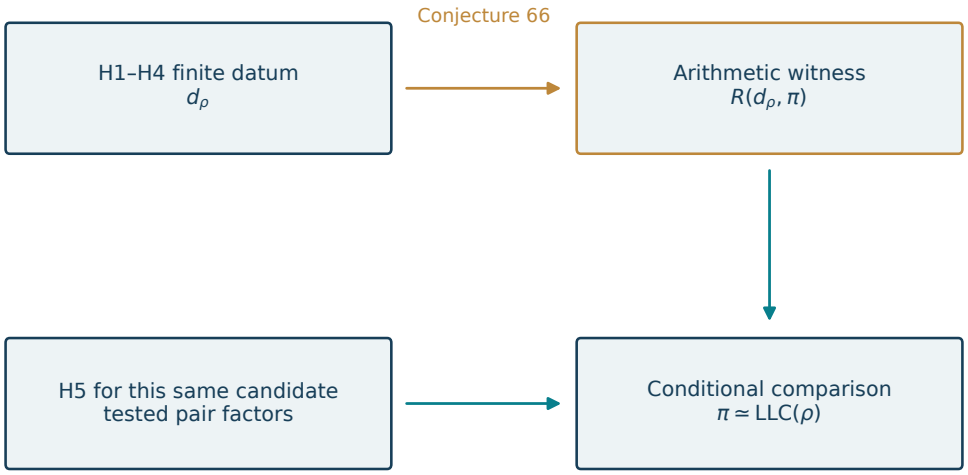
The dyadic construction also gives a rank-eight skipped-degree $8 \to 2$ host with the same torsion, Swan, and self-pair conductor as the full tower, but quotient dimension 4 rather than 6.

Reading the diagram. The finite packet, numerical profiles, and noncompact normalization contain different information. No one of these alone is a substitute for an arithmetic witness preserving the full declared datum.

PART IV / REALIZATION AND COMPARISON

43 Conjecture 66 isolates the missing existence claim

Source: Historical Theorem 16.2; current realization contract.



The upper existence arrow is conjectural. The lower comparison follows conditionally.

Conjecture 66 (arithmetic packet realization). Fix the declared class $\mathcal{A}$ of wild atoms and the paper's normalizations. For every $\rho \in \mathcal{A}$ with supplied H1–H4 assembled finite datum d_ρ, there is one actual admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(F)$ and an arithmetic realization witness $R(d_\rho, \pi)$ recovering the packet and preserving its declared stabilizer action, central-extension and phase data, and compatible transports.

Here R includes actual arithmetic inducing data and explicit comparison and recovery. It is not merely the existence of abstract operators with the right commutators.

Scope. The conjecture is restricted to the declared admitted class. It does not assert that arbitrary coefficient targets, arbitrary partial hosts, or independently conflicting completion choices can all be extended.

PART IV / REALIZATION AND COMPARISON

44 H5 compares this same actual candidate

Source: Contract audit 162; conditional comparison proposition.

Suppose π is an actual admissible supercuspidal representation of the correct rank and $R(d_\rho, \pi)$ is supplied. If H5 gives equality of its required pair local factors with those of $\mathrm{LLC}(\rho)$ on the stipulated characterizing twist family, with the fixed normalizations, then

$$\pi \simeq \mathrm{LLC}(\rho).$$

The proof applies the supplied local-factor characterization to these two actual representations. It does not need existence for other diagrams or H5 for every alternative completion.

There is a logical difference between

$$\forall \pi\, [R(d, \pi) \Rightarrow \mathrm{H5}(\pi)] \quad \text{and} \quad \exists \pi\, [R(d, \pi) \wedge \mathrm{H5}(\pi)].$$

The first statement can be vacuous. The second includes an actual witness. Choosing the established reference atom only supplies an object with a matching profile. It leaves its recovery of d unproved.

Reading the diagram. H5 remains a conditional premise about the same independently constructed candidate. It is not renamed as an unconditional theorem comparing factors.

PART IV / REALIZATION AND COMPARISON

45 What the restricted realizations establish

Source: Odd-prime and dyadic constructions: final composition reviews.

- Actual totally ramified parameter fields, literal defining sequences, full simple characters, and covered central completions exist throughout the stated odd-prime and dyadic families.
- Under the one-vertex, $C = 1$, supplied full-character scalar-pushout contract, the whole marked packet has an actual arithmetic assembly carrier and same-candidate recovery.
- In the dyadic low ranks, the actual enlarged carriers restore capacity without changing the original maximal-simple quotient.
- Genuine scalar inertia phases have the separately stated finite-cover descent result.

The remaining general obligations include nontrivial prescribed symplectic stabilizer actions on a common arithmetic candidate, general sources and transport diagrams, comparison with independently fixed canonical native markings, and numerical layer placement when the declared contract requires it.

Scope. The constructions are substantial restricted realization theorems. Their scopes do not close universal Conjecture 66, and their profile identities do not prove H5.

PART IV / EVIDENCE AND REPRODUCIBILITY

46 Exact computation checks a bounded identity

Source: Dyadic construction: computation contract; final verification records.

The rank-eight run for $h = (1, 3, 5)$ produces an actual matrix with 25 nonzero Laurent polynomial entries. Its checked leading and adjacent-correction grades are

$$-60, \quad -18, \quad -3.$$

It checks normalized lifts, actual matrix identities, native traces 1, and amplified ambient traces $0, 0, 1$. The odd $p = 3$ example $h = (1, 2, 4)$ has levels $144, 24, 4$ and native trace zero.

The finite outputs have explicit contracts, scripts, inputs and execution records. Critical exponents, simplicity, full-character existence and arbitrary-length assertions use separate ordinary mathematical arguments.

Scope. A bounded matrix run cannot establish the universal realization claim. The arbitrary-length arithmetic constructions have ordinary proofs. Their formal coverage is stated separately in the formal companion.

SOURCE MAP / SUPPLEMENTARY MATERIALS

47 Where the complete arguments are retained

Source: Source collection.

Topic	Retained source anchor
Finite tame and wild route	Original manuscript, Sections 2–16. The mathematical text uses the geometric Frobenius convention and states the realization boundary explicitly.
Original endpoint contract	`ORIGINAL_CONTRACT_162.md`: common-type versus separate-type readings, same-candidate H5, and exact witness requirement.
Odd arithmetic constructor	`PROOF166.md`; `TRACE_ZERO_AS_TOWER166.md`; `RECURSIVE_TOWER166.md`.
Full-character induction	`TRACE_ZERO_CHARACTER_INDUCTION166.md`; `DYADIC_FULL_CHARACTER167.md`.
Dyadic construction	`PROOF167.md`; `DYADIC_AS_TOWER167.md`; `DYADIC_CAPACITY167.md`.
Marked finite extensions	`DYADIC_MARKINGS167.md`, including conditions on native refinements and transport of whole subgroups.
Enlarged low ranks	`ENLARGED_CARRIER_CONTRACT167.md`; the rank-two, rank-four, and induction-in-stages notes.
Stabilizer boundary	`STABILIZER_CONTRACT166.md`; `SCALAR_NATIVE_ATTACHMENT166.md`.
Reviews and computation	The accompanying audit directories, exact computation contracts, source pins, and execution manifests.

The source files accompany the volume. Their titles and review records do not enlarge the hypotheses or conclusions stated in the mathematical text.

CHAPTER XII

Patching Support and Hecke Packets

Conditional automorphic completion

The arithmetic sequence culminates in a conditional argument using patching support, Hecke compatibility and the stated representation-theoretic inputs.

Patching support and compatible Hecke packets complete a conditional arithmetic argument. Each application must supply the geometric, automorphic and representation-theoretic inputs that the theorem uses.

IN THIS CHAPTER

Mathematical text 1418

Visual studies 1455

Formal companion 2617

Sources and references 2702

Scope

Actual residual and deformation data, applicable patching and full support, a compatible same-field Hecke point, classicality and an attached representation are required. Comparisons of compatible families assume those families already exist. Chapter XI's Conjecture 66 is not assumed proved.

The formal companions at the back of the book describe their own precise coverage. The glossary provides a route back to the principal definitions.

A reading map

This volume develops the algebraic steps in conditional automorphic completion. Read its support and packet theorems as deductions for the rings, modules, complexes, and arithmetic inputs that are actually supplied. They do not create missing residual occupancy or missing classicality.

The companion visual guide gives 27 worked lessons. The formal companion exposes the production sources, the 49-claim crosswalk, and the hypotheses of the formal interfaces. The links below are reading routes, not additional theorem certificates.

Topic	Main statement	Visual	Lean section
Local conditions	3.1	Plate XII.3	4
Support and quotients	6.9	Plate XII.10	8
Derived support	6.16	Plate XII.11	9
Hecke packets	8.2	Plate XII.12	10
Stable cores	8.11	Plate XII.15	11
Companion completion	10.5	Plate XII.23	11
Conditional realization	12.6	Plate XII.17	12

What the verification record means

The inherited formal record covers all 49 revised internal claims. The ledger of original wording remains separate: 41 complete, seven partial, and one unchecked. Amended statements are distinguished from their stronger predecessors.

The formal companion identifies the recorded Lean compilation and the separate source, computation and correspondence checks. Representability, applicable patching, classicality, attached representations, Frobenius density, and the comparison inputs retain their stated external status.

The final theorem requires a compatible Hecke point over the stated field. The propagation step compares two already existing families associated with the same automorphic object. Chapter XI's general realization conjecture is not an input to this chapter.

ABSTRACT. Let $\bar{\rho}$ be a residual Galois representation and let R be a global deformation ring with declared local conditions. We separate the global automorphy problem into residual occupancy, deformation support, Hecke occupancy, and arithmetic realization. Local conditions are encoded by quotient rings rather than by idempotents, since irreducible components of a local deformation ring need not be clopen. For an ordinary or derived patching datum, full support descends through the augmentation quotient. Near faithfulness is shown to be equivalent to full support for finite modules over Noetherian rings. An isomorphism $R \cong \mathbb{T}$ transports support, Fitting ideals, and point fibers, but a quotient map can erase an entire component. Finite Hecke algebras are resolved into canonical Artin packets, so simultaneous generalized eigenspaces are available without assuming semisimplicity. Nested packets in one finite carrier stabilize. For changing finite carriers, nonzero stable cores give compatible inverse limit vectors. These results give a pointwise criterion: a characteristic zero deformation point in automorphic support produces a Hecke eigensystem. With attached Galois representations, local global compatibility, and classicality, the point is automorphic. A fixed compatible system then propagates the conclusion to every coefficient place. For finite coefficient quotients, compatible conjugacy classes admit compatible continuous representatives. Declared polynomial quotient conditions survive their adic reconstruction. These completion statements require explicit finite occupancy, continuity, and integral Frobenius data. Universal companion existence and unrestricted reciprocity are not proved.

The realization boundary

Theorem 12.6 brings the five realization requirements into one statement. The diagrams throughout the chapter distinguish supplied arithmetic data from algebraic deductions. The formal companion maps each claim to its source and describes the scope of the recorded verification.

Formal evidence is mapped to all forty-nine revised internal claims. The original formulations have a separate record: forty-one complete, seven partial and one unchecked. The stronger original formulations and the revised statements are different contracts. The completed tensor and point-factorization results are algebraic deductions for supplied deformation data. Representability, residual occupancy, applicable patching and classicality theorems, attached Galois representations, Frobenius density, semisimplification, and the general Brauer–Nesbitt theorem remain external arithmetic or algebraic inputs. Neither Chapter XI's general Conjecture 66 nor universal companion existence is assumed.

1 Introduction

Automorphy lifting joins several kinds of mathematics that are easy to conflate. A residual representation must first occur in an automorphic carrier. The chosen characteristic zero representation must lie on a deformation problem controlled by local conditions. A patching argument must show that the automorphic module reaches the relevant part of the deformation space. A comparison such as $R \cong \mathbb{T}$ must identify the arithmetic and Hecke coordinates. Finally, the resulting Hecke point must be classical and must carry the expected Galois representation with the required local compatibility.

None of these steps is merely a change of notation. A deeper lift above one residual representation cannot create a missing residual companion. Freeness over an auxiliary group algebra does not imply that every component of a deformation ring is occupied. An isomorphism of rings transports support that already exists, but it does not create a nonzero automorphic vector. A nonzero class in completed or torsion cohomology need not be classical without a theorem that says so.

The Taylor Wiles method and its later forms solve these problems in many important settings [27, 24, 9, 14, 12]. Higher dimensional and derived versions use a broader range of local deformation rings, residual image conditions, torsion classes, and cohomological carriers [8, 3, 6, 25, 17, 22, 7, 19].

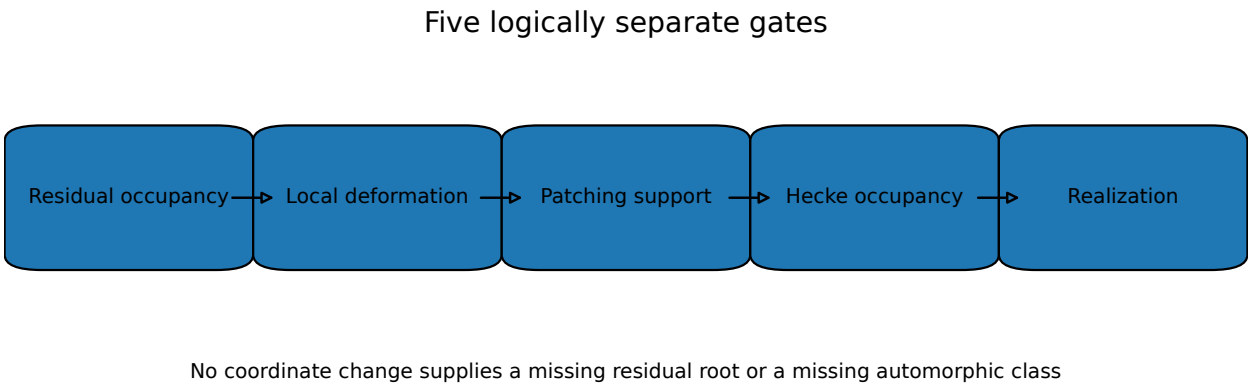


Figure 1. The global argument has several logically separate gates.

The present chapter does not replace those theorems. It isolates the algebraic support statements that explain exactly what a patching package supplies once its arithmetic hypotheses have been verified.

The first correction concerns local packets. A local deformation ring is usually complete and local. Its irreducible components are rarely open and closed. They cannot generally be represented by central idempotents. The correct object is a quotient ring defining a local condition, together with the support of a module or complex on the resulting spectrum. Central idempotents reappear later, where they belong, inside finite Artin Hecke algebras and finite dimensional characteristic zero carriers.

The second correction concerns patching. The useful global conclusion is full support, often expressed as near faithfulness. For a finite module over a Noetherian ring, near faithfulness is equivalent to full support. Full support descends through the quotient that removes the auxiliary variables. This statement is elementary, but it is the exact bridge from a patched module to componentwise occupancy at finite level. The same conclusion holds for perfect complexes through derived support.

The third correction concerns Hecke packets. Semisimplicity is not required to isolate a maximal Hecke packet. Every finite commutative algebra over a field is a product of Artin local algebras. Its canonical central idempotents project onto simultaneous generalized eigenspaces. Trace is only a secondary criterion in characteristic zero. Support and localization are the primary notions.

The last part assembles these statements into a conditional theorem. A characteristic zero point of R is automorphic when it lies in the support of the automorphic carrier, the comparison with the Hecke algebra is available, and the corresponding Hecke fiber satisfies the declared realization theorem. If the point belongs to a fixed strictly compatible system, equality of Frobenius polynomials propagates the automorphic identification to every coefficient place by Chebotarev density and Brauer Nesbitt [5, 18, 12].

Local quotient conditions are developed in section 3. Adequacy and residual occupancy appear in section 4. Auxiliary group algebras are treated in section 5. Ordinary and derived patching support appear in section 6. The effect of $R \cong \mathbb{T}$ is isolated in section 7. Finite Hecke packets and infinite occupancy are developed in section 8. Pointwise automorphy and compatible systems appear in sections 9 and 10. The coefficient prime two is discussed in section 11. The final conditional theorem is stated in section 12.

2 Global deformation data and exact boundaries

Throughout, S is a finite set of places. Frobenius elements are geometric Frobenius elements. All Hecke polynomials, attached representations, and common algebraic spines use that same convention and the normalization specified by the declared realization theorem.

Let F be a number field. Fix a prime p, a finite extension $E/\mathbb{Q}_p$, its valuation ring $\mathcal{O}$, uniformizer ϖ, and residue field k. Let

$$\bar{\rho} : G_{F,S} \longrightarrow \mathrm{GL}_n(k)$$

be a continuous residual representation, where S is a finite set of places containing every place at which the deformation problem is allowed to ramify.

Definition 2.1 (Global deformation datum)**.** A global deformation datum consists of the following objects.

(1) A residual representation $\bar{\rho}$.
(2) For every $v \in S$, a framed local deformation ring $R_v^{\square}$.
(3) A quotient $R_v^{\tau_v}$ representing the chosen local condition whenever such a quotient is known to exist.
(4) A global deformation ring $R^{\square}$ with a continuous local map
$$R^{\mathrm{loc}} := \widehat{\otimes}_{v\in S,\mathcal{O}} R_v^{\square} \longrightarrow R^{\square}.$$
(5) The induced global ring with local conditions
$$R^{\tau} := R^{\square} \,\widehat{\otimes}_{R^{\mathrm{loc}}}\, R^{\mathrm{loc},\tau}, \qquad R^{\mathrm{loc},\tau} := \widehat{\otimes}_{v\in S,\mathcal{O}} R_v^{\tau_v}.$$

Representability of deformation functors and the construction of local condition quotients require hypotheses. The standard foundations are due to Mazur and Schlessinger, with many later refinements for local conditions and pseudorepresentations [16, 21, 4].

Standard Fact 2.2 (Represented deformation problem)**.** Under the usual finiteness and Schur type hypotheses, framed lifts of $\bar{\rho}$ are represented by complete Noetherian local $\mathcal{O}$ algebras. A declared local deformation condition is represented by a quotient when the corresponding subfunctor is representable. Tangent and obstruction spaces are described by the appropriate first and second cohomology groups.

Standard Fact 2.3 (Automorphy lifting package)**.** Published automorphy lifting theorems begin with residual automorphic occupancy and impose hypotheses on the residual image, local deformation conditions, regularity, and the global deformation problem. They then use ordinary or derived patching to compare a deformation ring with a Hecke algebra and to prove automorphy for characteristic zero points in the supported locus. The precise assumptions depend on the theorem being used [8, 3, 6, 25, 17, 1].

The chapter treats standard facts 2.2 and 2.3 as external inputs. Its results begin after the relevant rings, modules, complexes, and comparison maps have been supplied.

Definition 2.4 (Residual automorphic occupancy)**.** The residual datum $(\bar{\rho}, \tau)$ is automorphically occupied when a nonzero localized automorphic or cohomological Hecke carrier realizes the residual representation and the declared local conditions.

This is a depth zero hypothesis. A positive depth deformation argument above another residual root cannot produce it.

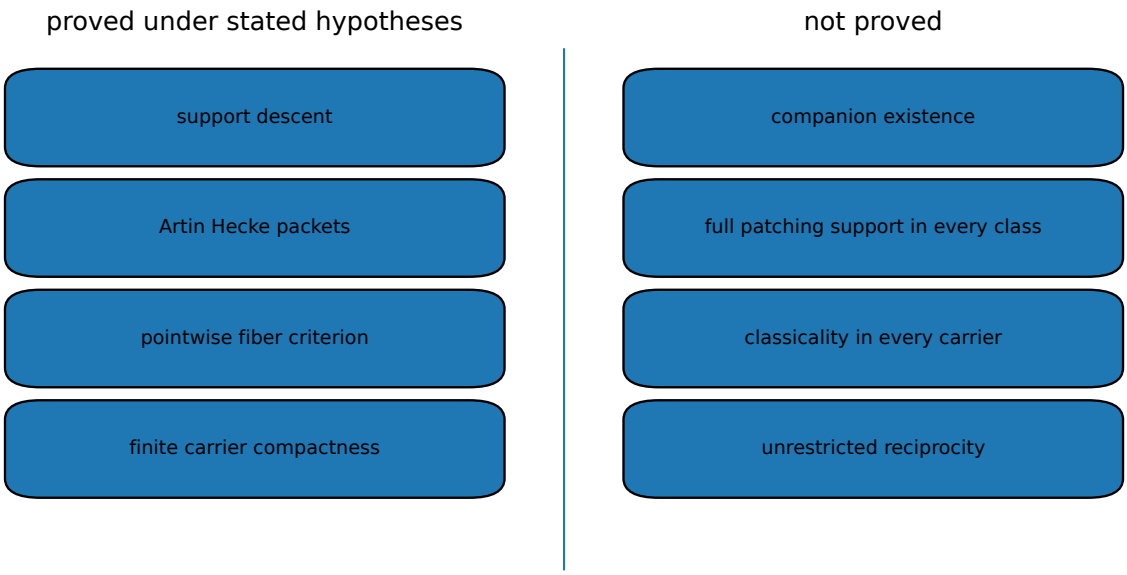


Figure 2. The chapter proves formal support and occupancy theorems. It does not prove the universal existence statements shown on the right.

3 Local conditions are quotient packets

The completed tensor product combines the local deformation conditions. The first theorem records the exact quotient law.

Theorem 3.1 (Completed tensor quotient)**.** *Let $R_1, \dots, R_s$ be complete Noetherian local $\mathcal{O}$ algebras with common residue field k. Let $I_i \subseteq R_i$ be closed ideals. Put*

$$R^{\mathrm{loc}} = R_1 \mathbin{\widehat{\otimes}}_{\mathcal{O}} \cdots \mathbin{\widehat{\otimes}}_{\mathcal{O}} R_s$$

and let J be the closed ideal generated by the images of the I_i. Then there is a canonical isomorphism

$$\boxed{(R_1/I_1) \mathbin{\widehat{\otimes}}_{\mathcal{O}} \cdots \mathbin{\widehat{\otimes}}_{\mathcal{O}} (R_s/I_s) \cong R^{\mathrm{loc}}/J.}$$

Proof. Write $S = \bigotimes_{i=1}^{s} R_i$ for the ordinary tensor product. Its ideal of definition is

$$\mathfrak{a} = \mathfrak{m}_{\mathcal{O}} S + \sum_i \mathfrak{m}_{R_i} S, \qquad L = \sum_i I_i S.$$

The scalar term retains the base adic topology when the family is empty. The ordinary quotient theorem gives a canonical algebra isomorphism

$$S/L \cong \bigotimes_{i=1}^{s} (R_i/I_i),$$

carrying the image of $\mathfrak{a}$ to the defining ideal on the right. The quotient topology on S/L is precisely this image ideal topology.

Here completion means the actual Hausdorff completion for the additive uniformity of the indicated adic topology. For every adic ring S and ideal L, there is a canonical bicontinuous ring isomorphism

$$\widehat{S}\big/\overline{L\widehat{S}} \cong \widehat{S/L}.$$

Indeed, the completed quotient map kills the closed extended ideal. In the other direction, the quotient map from S/L to $\widehat{S}/\overline{L\widehat{S}}$ extends to its completion. The latter quotient is complete: the adic uniformity has a countable base, and the quotient of a complete countably based additive group is complete. Its closed defining subgroup makes it Hausdorff. The two composites agree with the identity on dense quotient representatives, so continuity proves both inverse laws.

Applying this comparison to the displayed ordinary tensor isomorphism proves the claim. The closed extended ideal is exactly J, since extension commutes with the finite sum of the factor ideals. The maps preserve coefficients and each canonical factor map. The common residue field identifies $S/\mathfrak{a}$ with k. Finite generation of $\mathfrak{a}$, supplied by the Noetherian factor hypotheses, identifies the completed topology with its maximal ideal topology. Proper quotients retain locality. If a condition is the unit ideal, the ring comparison remains valid with the zero ring, without treating it as a local ring with residue field k. □

Corollary 3.2 (Simultaneous local conditions)**.** *A continuous point of R^{loc} with Hausdorff target factors continuously through R^{loc}/J if and only if its restriction to every R_i factors continuously through R_i/I_i. Each factor is unique. If the original point is local, its quotient factor is local.*

Proof. Let $f : R^{\mathrm{loc}} \to A$ be the point. Since A is Hausdorff and f is continuous, $\ker f$ is closed. Thus it contains the closed ideal J if and only if it contains the algebraic ideal generated by the images of the I_i. This is equivalent to every local restriction killing I_i. The universal property of the actual ideal quotient gives the factors, and the quotient topology gives their continuity. Surjectivity of the quotient map gives uniqueness. If f reflects units, lift an element of the quotient to the original ring to see that its factor also reflects units. □

Proposition 3.3 (Global base change for local conditions)**.** *In the global deformation datum of definition 2.1, let $R^{\square}$ be a complete Noetherian local $\mathcal{O}$ algebra receiving a continuous local $\mathcal{O}$ algebra map from R^{loc}. Then*

$$R^{\tau} = R^{\square} \mathbin{\widehat{\otimes}}_{R^{\mathrm{loc}}} R^{\mathrm{loc},\tau}$$

represents the base change of the global deformation problem to the simultaneous local conditions. A continuous local $\mathcal{O}$ algebra point of $R^{\square}$ with complete Hausdorff coefficient target factors uniquely and continuously through R^{τ} exactly when all of its local restrictions satisfy the declared quotient conditions.

Proof. Put $A = R^{\mathrm{loc}}$, $B = R^{\square}$, and let $J \subseteq A$ be the closed local condition ideal. By theorem 3.1, the independently constructed ring $R^{\mathrm{loc},\tau}$ is canonically A/J. The ordinary tensor identity

$$B \otimes_A (A/J) \cong B/JB$$

carries the defining ideal of the tensor to the image of $\mathfrak{m}_B$. Here the structural map is local by the stated global deformation datum, so it carries $\mathfrak{m}_A$ into $\mathfrak{m}_B$. Completing, and applying the completion and quotient comparison from the preceding proof, gives a bicontinuous isomorphism

$$R^{\tau} \cong B/\overline{JB}.$$

The construction commutes with the canonical map from B, which is therefore surjective. It is also unchanged by transporting the tensor factor from A/J to the actual completed tensor of the local quotients. The transported ideal is exactly its natural defining ideal.

For a continuous algebra point $x : B \to C$ with complete Hausdorff target, the kernel of x is closed. Consequently x has a unique continuous factor through R^{τ} if and only if it kills JB, equivalently if its restriction to A kills J. By corollary 3.2, this is exactly the requirement that all local restrictions satisfy their quotient conditions. The factor is an algebra map because its composite with the canonical global map is x. If x is local, surjectivity of that map shows that its factor reflects units and is local. Uniqueness also proves naturality under coefficient preserving maps of targets. This identifies the represented base change. It does not supply representability of a new arithmetic deformation problem. □

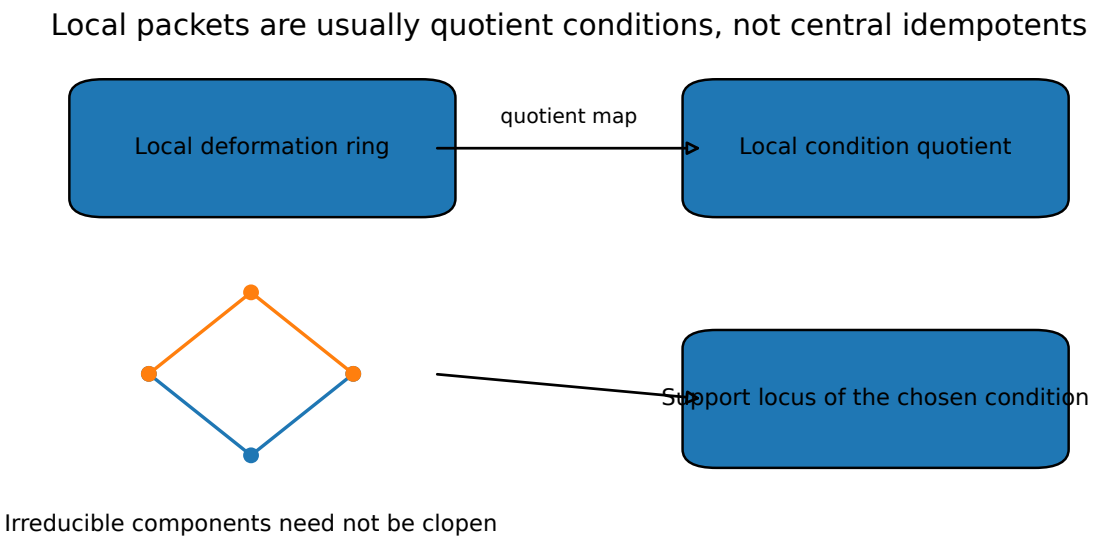


Figure 3. Local conditions are imposed by quotient rings. Components are support loci unless a genuine product decomposition exists.

The local coefficient points in proposition 3.3 are test rings in the represented deformation category. Later field valued points $x : R \to E'$ are algebraic or generic specializations, and continuity of ρ_x is imposed separately. Such a field specialization need not be a local ring map: under $\mathcal{O} \to E$, a uniformizer becomes a unit.

Proposition 3.4 (Idempotents in a local ring)**.** *If R is a commutative local ring, then its only idempotents are zero and one.*

Proof. Let $e^2 = e$. Since $e + (1 - e) = 1$, at least one of e and $1 - e$ is a unit. If e is a unit, then $e = 1$. If $1 - e$ is a unit, then $e = 0$. □

Remark 3.5. An irreducible component of $\operatorname{Spec} R$ for a local deformation ring need not be open and closed. It therefore need not be selected by an idempotent. Components and local conditions should be described by quotient ideals and module support. Idempotent packet formulas are reserved for finite Artin algebras or other settings with an actual product decomposition.

4 Adequacy and the residual input

Adequacy is one of the residual image hypotheses used in several automorphy lifting theorems. The definition and its role are external to the support arguments below [13, 25, 17].

Standard Fact 4.1 (Adequate image)**.** Let $G \subseteq \mathrm{GL}(V)$ act absolutely irreducibly over a finite field k of characteristic p. Under the standard definition, adequacy imposes vanishing conditions on $H^1(G, k)$ and $H^1(G, \mathrm{ad}^0 V)$, a scalar condition, and a generalized eigenspace detection condition for every irreducible submodule of $\mathrm{ad}^0 V$.

The following small example makes the projector detection condition explicit. It is a certificate for one finite representation, not a new adequacy criterion.

Let $G = S_3$, let $k = \mathbb{F}_7$, and let V be the standard two dimensional representation. In the basis

$$v_1 = (1, -1, 0), \qquad v_2 = (0, 1, -1),$$

a transposition and a three cycle act by

$$s = \begin{pmatrix} 6 & 1 \\ 0 & 1 \end{pmatrix}, \qquad t = \begin{pmatrix} 0 & 6 \\ 1 & 6 \end{pmatrix}.$$

Theorem 4.2 (Finite adequacy certificate)**.** *The standard representation of S_3 over $\mathbb{F}_7$ is adequate. More precisely,*

$$H^0(S_3, \mathrm{ad}^0 V) = 0, \qquad H^1(S_3, \mathbb{F}_7) = 0, \qquad H^1(S_3, \mathrm{ad}^0 V) = 0,$$

and

$$\mathrm{ad}^0 V \cong \mathrm{sgn} \oplus V.$$

The sign summand is detected by an eigenprojector of a three cycle, and the standard summand is detected by an eigenprojector of a transposition.

Proof. The coordinate map

$$(a, b) \longmapsto (a, -a + b, -b)$$

identifies the displayed matrices with the permutation action on the sum zero subspace of $\mathbb{F}_7^3$. This action is faithful. The four matrices $1, s, t, st$ span M_2 over every field of characteristic seven: the determinant of their coordinate matrix is $4 \neq 0$. A subspace stable under the group is therefore stable under every endomorphism, so the representation is absolutely irreducible. The characteristic polynomials of the identity, transpositions, and three cycles are respectively

$$(X - 1)^2, \qquad (X - 1)(X + 1), \qquad (X - 2)(X - 4).$$

Thus all eigenvalues already lie in $\mathbb{F}_7$.

A trace zero matrix fixed by conjugation by both s and t is zero, as follows by comparing its four entries. This proves $H^0(S_3, \mathrm{ad}^0 V) = 0$. For any $\mathbb{F}_7[S_3]$ module M and one cocycle f, put

$$u = -6^{-1} \sum_{h \in S_3} f(h).$$

Summing $f(gh) = gf(h) + f(g)$ over h gives $gu - u = f(g)$. Hence every one cocycle is a coboundary and $H^1(S_3, M) = 0$, giving both asserted first cohomology vanishings.

Write

$$W(a, b, c) = \begin{pmatrix} a & b \\ c & -a \end{pmatrix}, \qquad Z = W(2, 3, 4), \quad A = W(1, 1, 2), \quad B = W(1, 5, 6).$$

The three coordinate columns have determinant $3 \neq 0$. Under conjugation,

$$sZ = -Z, \quad tZ = Z, \qquad sA = -A, \quad sB = A + B, \quad tA = B, \quad tB = -A - B.$$

Consequently $(z, (a, b)) \mapsto zZ + aA + bB$ is an equivariant isomorphism from $\mathrm{sgn} \oplus V$ to $\mathrm{ad}^0 V$.

The sign witness is $W_{\mathrm{sgn}} = Z$, while $W_{\mathrm{std}} = W(6, 4, 3) = 3A + 3B$ belongs to the standard summand. The corresponding projectors and trace pairings are

$$P_{t,2} = \begin{pmatrix} 2 & 4 \\ 3 & 6 \end{pmatrix}, \qquad \mathrm{Tr}(P_{t,2} W_{\mathrm{sgn}}) = 3 \neq 0,$$

$$P_{s,1} = \begin{pmatrix} 0 & 4 \\ 0 & 1 \end{pmatrix}, \qquad \mathrm{Tr}(P_{s,1} W_{\mathrm{std}}) = 6 \neq 0.$$

For an operator D with $(D - \alpha I)(D - \beta I) = 0$ and $\alpha \neq \beta$, the canonical projector is

$$P_{D,\alpha} = (\alpha - \beta)^{-1} (D - \beta I).$$

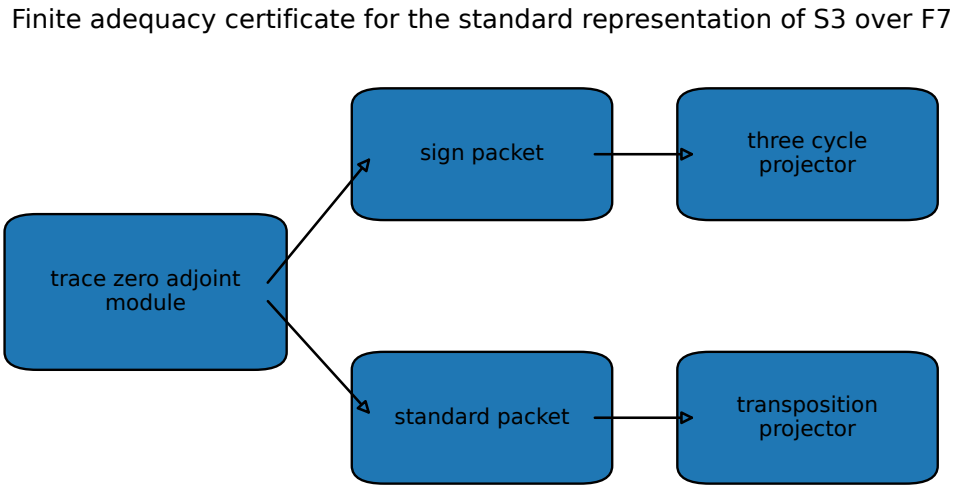


Figure 4. The two irreducible adjoint packets are detected by different generalized eigenspace projectors.

Its image equals $\ker((D-\alpha I)^n)$ for every $n \geq 1$, and it kills $\ker((D-\beta I)^n)$. This identifies the displayed matrices with the generalized eigenspace projectors required in the adequacy definition.

For detection on every irreducible submodule, one can check the stronger assertion that canonical projectors detect every nonzero matrix. In addition to the two projectors above, use

$$P_{s,-1} = \begin{pmatrix} 1 & 3 \\ 0 & 0 \end{pmatrix}, \qquad P_{tst^{-1},1} = \begin{pmatrix} 1 & 0 \\ 4 & 0 \end{pmatrix}.$$

The four trace pairings with these matrices have trivial common kernel in $M_2(\mathbb{F}_7)$, by row reduction. Every irreducible adjoint submodule contains a nonzero matrix, so some canonical projector has nonzero trace pairing on that submodule. This verifies the last adequacy clause and completes the certificate. □

Remark 4.3. The theorem verifies adequacy for one finite image. A global automorphy lifting argument must verify the published residual image hypothesis for its actual representation. The calculation does not replace that step, and it does not provide a uniform coefficient prime two theorem.

5 Auxiliary group algebras and augmentation

Let Δ be a finite abelian p group and let

$$\Lambda = \mathcal{O}[\Delta].$$

The augmentation map is

$$\epsilon : \Lambda \longrightarrow \mathcal{O}, \qquad \sum_{g \in \Delta} a_g g \longmapsto \sum_{g \in \Delta} a_g.$$

Its kernel is the augmentation ideal $\mathfrak{a}$.

Proposition 5.1 (Functorial group algebra transfer)**.** *A group homomorphism $q : \Delta' \to \Delta$ induces an $\mathcal{O}$ algebra homomorphism*

$$q_* : \mathcal{O}[\Delta'] \longrightarrow \mathcal{O}[\Delta].$$

If q is surjective, then q_* is surjective. Augmentation commutes with the transfer:

$$\epsilon_\Delta q_* = \epsilon_{\Delta'}.$$

Proof. Extend q linearly from the group basis. Multiplicativity follows from $q(gh) = q(g)q(h)$. Surjectivity follows because every basis element of $\mathcal{O}[\Delta]$ has a preimage. Both augmentation maps send every group element to one, which gives the final identity. □

Theorem 5.2 (Residual augmentation algebra)**.** *Suppose*

$$\Delta \cong \prod_{i=1}^{r} C_{p^{n_i}}.$$

Choose generators γ_i, and let $X_i = \gamma_i - 1$. Then

$$\boxed{k[\Delta] \cong k[X_1, \dots, X_r]/(X_1^{p^{n_1}}, \dots, X_r^{p^{n_r}}).}$$

The augmentation ideal is generated by the X_i, and $k[\Delta]$ is local.

Proof. In characteristic p,

$$(1 + X_i)^{p^{n_i}} - 1 = X_i^{p^{n_i}}.$$

Thus the assignment $X_i \mapsto \gamma_i - 1$ gives a surjective homomorphism from the displayed truncated polynomial ring to $k[\Delta]$. Both sides have dimension $\prod_i p^{n_i} = |\Delta|$ over k, so the map is an isomorphism. The quotient by the ideal generated by the X_i is k. Since each X_i is nilpotent, that ideal is the unique maximal ideal. □

Corollary 5.3 (No residual character splitting)**.** *The residual group algebra $k[\Delta]$ has no nontrivial idempotents. In particular, character projectors do not split a finite p group algebra in residue characteristic p.*

Proof. The first assertion follows from locality in theorem 5.2. For the general finite p group assertion, let $e \in k[G]$ be idempotent and write $\epsilon : k[G] \to k$ for augmentation. Suppose $\epsilon(e) = 0$ and $e \neq 0$. The $\mathbb{F}_p$ span W of the left translates ge is a nonzero finite dimensional $\mathbb{F}_p$ vector space, stable under G. Its cardinality is divisible by p. The orbit counting formula implies that its fixed point set also has cardinality divisible by p. Since zero is fixed, there is a nonzero fixed vector $v \in W$.

A vector in the regular representation fixed by all left translations has constant coefficients. Consequently $ve = \epsilon(e)v = 0$. On the other hand, $e^2 = e$ implies $we = w$ for every $w \in W$, hence $ve = v$, a contradiction. Thus every idempotent of augmentation zero is zero. Since $\epsilon(e)^2 = \epsilon(e)$ in the field k, either $\epsilon(e) = 0$ or $\epsilon(e) = 1$. In the latter case, apply the preceding argument to $1 - e$. □

For $M \otimes_{\Lambda,\epsilon} \mathcal{O}$, we use a right Λ module M. The left module convention is $\mathcal{O} \otimes_{\Lambda,\epsilon} M$. Throughout, $\mathfrak{a}M$ denotes the augmentation relation submodule: it is generated by ma for right modules and by am for left modules, with $a \in \mathfrak{a}$. This convention also applies to nonabelian groups in corollary 11.2. For abelian groups, both tensor orders agree over the commutative group algebra.

Theorem 5.4 (Coinvariant recovery)**.** *Let M be a free Λ module of rank d. Then*

$$\boxed{M/\mathfrak{a}M \cong M \otimes_{\Lambda,\epsilon} \mathcal{O} \cong \mathcal{O}^d.}$$

The statement is valid for every coefficient prime, including two.

Proof. Choose a Λ basis of M. Tensoring the resulting isomorphism $M \cong \Lambda^d$ along augmentation gives $\mathcal{O}^d$. Tensoring by $\Lambda/\mathfrak{a}$ is the same as quotienting by $\mathfrak{a}M$. □

Standard Fact 5.5 (Iwasawa limit)**.** For $\Delta_m = (\mathbb{Z}/p^m\mathbb{Z})^r$, the inverse limit of $\mathcal{O}[\Delta_m]$ is the Iwasawa algebra

$$\mathcal{O}[[\mathbb{Z}_p^r]] \cong \mathcal{O}[[X_1, \dots, X_r]],$$

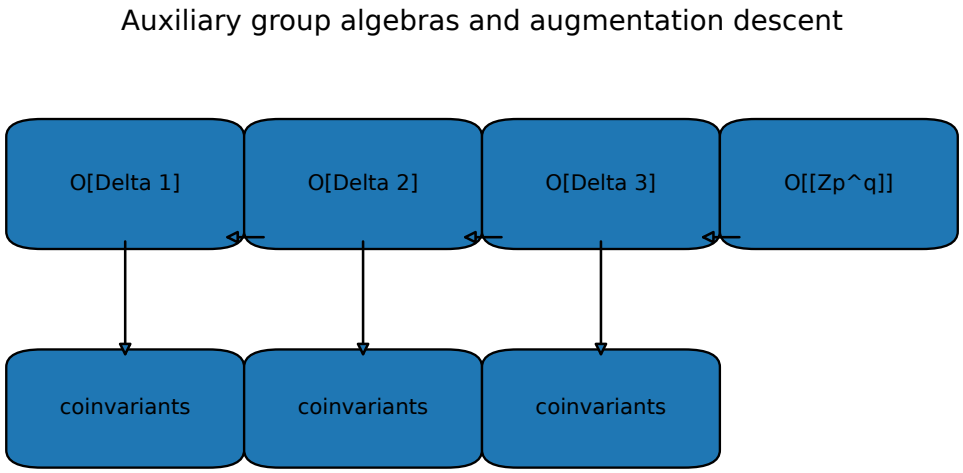


Figure 5. Auxiliary group algebras refine through quotient maps, while augmentation recovers the coarse level.

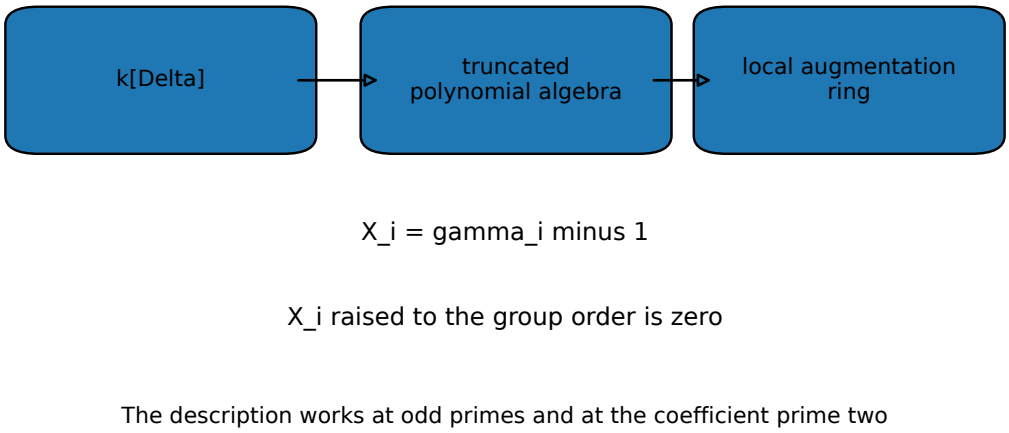


Figure 6. The residual auxiliary algebra is a local truncated polynomial algebra at every coefficient prime.

where X_i corresponds to a chosen topological generator minus one [20].

The augmentation formalism is uniform in p. The difficult coefficient two questions lie elsewhere. Residual image hypotheses, local deformation geometry, and classicality may change at two even though coinvariant recovery remains exact.

6 Ordinary and derived patching support

The algebraic conclusion supplied by a patching argument is most naturally stated through support. This section isolates the commutative algebra needed to descend that conclusion from a patched object to finite level.

Definition 6.1 (Support and near faithfulness)**.** Let R be a commutative ring and let M be an R module. Its support is

$$\operatorname{Supp}_R(M) = \{\mathfrak{p} \in \operatorname{Spec} R : M_{\mathfrak{p}} \neq 0\}.$$

If R is Noetherian and M is finite, then M is *nearly faithful* when $\operatorname{Ann}_R(M)$ is a nilpotent ideal.

Theorem 6.2 (Finite module support)**.** *Let R be a commutative ring and let M be a finite R module. Then*

$$\boxed{\operatorname{Supp}_R(M) = V\big(\operatorname{Ann}_R(M)\big).}$$

Proof. Let $\mathfrak{p} \in \operatorname{Spec} R$. If $\operatorname{Ann}_R(M) \not\subseteq \mathfrak{p}$, choose $a \in \operatorname{Ann}_R(M) \setminus \mathfrak{p}$. The element a becomes a unit in $R_{\mathfrak{p}}$, so $M_{\mathfrak{p}} = 0$.

Conversely, suppose $M_{\mathfrak{p}} = 0$. Choose generators $m_1, \dots, m_d$ of M. For each i, there is an element $s_i \notin \mathfrak{p}$ such that $s_i m_i = 0$. The product $s = s_1 \cdots s_d$ lies outside $\mathfrak{p}$ and annihilates every generator. Thus $s \in \operatorname{Ann}_R(M) \setminus \mathfrak{p}$. This proves the equality. □

Theorem 6.3 (Fitting support criterion)**.** *Let R be Noetherian and let M be a finite R module. Then*

$$\boxed{\operatorname{Supp}_R(M) = V\big(\operatorname{Fitt}_0^R(M)\big).}$$

Proof. If $M = 0$, the statement follows from the usual convention $\operatorname{Fitt}_0(0) = R$. Assume $M \neq 0$, and choose a finite presentation

$$R^m \xrightarrow{A} R^n \longrightarrow M \longrightarrow 0$$

with $n \geq 1$. The zeroth Fitting ideal is generated by the n by n minors of A. After localization at a prime $\mathfrak{p}$, the module $M_{\mathfrak{p}}$ vanishes exactly when $A_{\mathfrak{p}}$ is surjective. A map onto the free module $R_{\mathfrak{p}}^n$ is surjective exactly when one maximal minor is a unit. Thus $M_{\mathfrak{p}} = 0$ exactly when $\operatorname{Fitt}_0^R(M) \not\subseteq \mathfrak{p}$. □

Theorem 6.4 (Higher Fitting strata)**.** *Let R be Noetherian and let M be finite. For $r \geq 1$, put*

$$\mu_{\mathfrak{p}}(M) = \dim_{\kappa(\mathfrak{p})} \left(M \otimes_R \kappa(\mathfrak{p})\right).$$

Then

$$\boxed{\{\mathfrak{p} \in \operatorname{Spec} R : \mu_{\mathfrak{p}}(M) \geq r\} = V\big(\operatorname{Fitt}_{r-1}^R(M)\big).}$$

Proof. Use a finite presentation with target rank n. If $r > n$, every fiber has dimension at most n, so the left-hand locus is empty. The convention $\operatorname{Fitt}_{r-1}(M) = R$ for $r - 1 \geq n$ makes the right-hand locus empty as well. It remains to consider $1 \leq r \leq n$. At $\mathfrak{p}$, the fiber dimension is

$$n - \operatorname{rank} A(\mathfrak{p}).$$

It is at least r exactly when the rank is at most $n - r$, which is equivalent to the vanishing of every minor of size $n - r + 1$. Those minors generate $\operatorname{Fitt}_{r-1}^R(M)$. □

Proposition 6.5 (Fitting ideals and base change)**.** *Let $R \to S$ be a ring map and let M be a finitely presented R module. Then for every $j \geq 0$,*

$$\boxed{\operatorname{Fitt}_j^S(M \otimes_R S) = \operatorname{Fitt}_j^R(M)S.}$$

Proof. Base change carries a finite presentation matrix for M to a presentation matrix for $M \otimes_R S$. Every defining minor is carried to the corresponding minor over S. □

Remark 6.6. Support records nonvanishing. The higher Fitting strata record lower bounds on the number of generators in each fiber. They should not be confused with automorphic multiplicity formulas, which require additional representation theoretic input.

Theorem 6.7 (Near faithfulness is full support)**.** *Let R be Noetherian and let M be a finite R module. The following conditions are equivalent.*

(1) M is nearly faithful.
(2) $\operatorname{Supp}_R(M) = \operatorname{Spec} R$.
(3) $\operatorname{Ann}_R(M) \subseteq \operatorname{Nil}(R)$.

If R is reduced, these conditions are equivalent to faithfulness.

Proof. By theorem 6.2, full support is equivalent to

$$V\big(\operatorname{Ann}_R(M)\big) = \operatorname{Spec} R.$$

This holds exactly when $\operatorname{Ann}_R(M)$ is contained in every prime ideal, hence in the nilradical. Because R is Noetherian, the ideal $\operatorname{Ann}_R(M)$ is finitely generated. An ideal generated by finitely many nilpotent elements is nilpotent. This proves the equivalence of the first three conditions. If R is reduced, its nilradical is zero. □

Corollary 6.8 (Support on every irreducible component)**.** *Under the hypotheses of theorem 6.7, a nearly faithful module is nonzero at the generic point of every irreducible component of* $\operatorname{Spec} R$.

Proof. Every generic point is a prime of R, and full support contains every prime. □

Theorem 6.9 (Support after a quotient)**.** *Let R be Noetherian, let $I \subseteq R$ be an ideal, and let M be a finite R module. Under the canonical identification*

$$\operatorname{Spec}(R/I) \cong V(I) \subseteq \operatorname{Spec} R,$$

one has

$$\boxed{\operatorname{Supp}_{R/I}(M/IM) = \operatorname{Supp}_R(M) \cap V(I).}$$

Proof. Let $\mathfrak{p} \supseteq I$. Localizing gives

$$(M/IM)_{\mathfrak{p}} \cong M_{\mathfrak{p}}/I_{\mathfrak{p}}M_{\mathfrak{p}}.$$

If $M_{\mathfrak{p}} = 0$, the quotient is zero. If $M_{\mathfrak{p}} \neq 0$, then $I_{\mathfrak{p}}$ lies in the maximal ideal of the local ring $R_{\mathfrak{p}}$. Nakayama's lemma shows that $M_{\mathfrak{p}}/I_{\mathfrak{p}}M_{\mathfrak{p}} \neq 0$. The stated equality follows. □

Definition 6.10 (Ordinary patching datum)**.** An ordinary patching datum consists of a Noetherian ring R_∞, an ideal $\mathfrak{a} \subseteq R_\infty$, a finite R_∞ module M_∞, and identifications

$$R \cong R_\infty/\mathfrak{a}, \qquad M \cong M_\infty/\mathfrak{a}M_\infty.$$

In automorphy lifting applications, R_∞ is a patched deformation ring and M_∞ is a patched automorphic module. The definition itself contains no automorphy assertion.

Theorem 6.11 (Ordinary patching support descent)**.** *Let $(R_\infty, \mathfrak{a}, M_\infty, R, M)$ be an ordinary patching datum. If*

$$\operatorname{Supp}_{R_\infty}(M_\infty) = \operatorname{Spec} R_\infty,$$

then

$$\boxed{\operatorname{Supp}_R(M) = \operatorname{Spec} R.}$$

Equivalently, if M_∞ is nearly faithful, then M is nearly faithful. If R is reduced, then M is faithful.

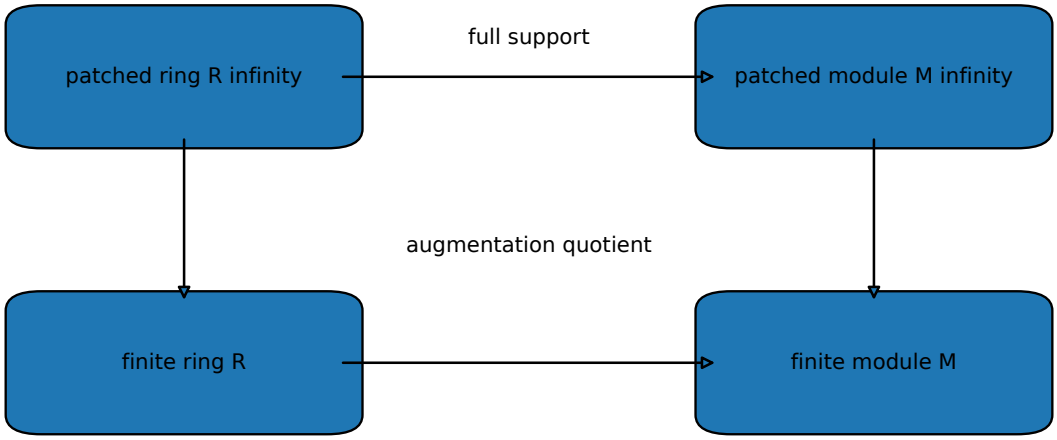


Figure 7. Full support at the patched level descends through the augmentation quotient.

Proof. Apply theorem 6.9 with $I = \mathfrak{a}$. The support of the quotient is

$$\operatorname{Spec} R_\infty \cap V(\mathfrak{a}) = V(\mathfrak{a}),$$

which identifies with $\operatorname{Spec} R$. The remaining statements follow from theorem 6.7. □

Corollary 6.12 (Componentwise finite level occupancy)**.** *In the setting of theorem 6.11, every irreducible component of* $\operatorname{Spec} R$ *is contained in the support of* M*. In particular, no component of the finite level deformation ring is erased by the quotient from the patched level.*

Proof. The module M has full support by theorem 6.11. □

Example 6.13 (Freeness over auxiliary variables is not enough)**.** Let k be a field, let $S = k[[t]]$, and put

$$R = S[[x]]/(x(x - t)).$$

The ring R is reduced and has two irreducible components. Let

$$M = R/(x) \cong S.$$

Then M is free of rank one over S, but

$$\operatorname{Supp}_R(M) = V(x)$$

is only one component of $\operatorname{Spec} R$. Thus freeness over the auxiliary power series ring does not imply full support over the deformation ring.

Proof. The two nonassociate prime factors x and $x - t$ occur with multiplicity one, so the principal quotient is reduced. The displayed isomorphism makes M free over S. Its annihilator as an R module is (x), hence

$$\operatorname{Supp}_R(M) = V(x)$$

by theorem 6.2. The generic point $(x - t)$ of the other component does not contain x, so it is not in the support. □

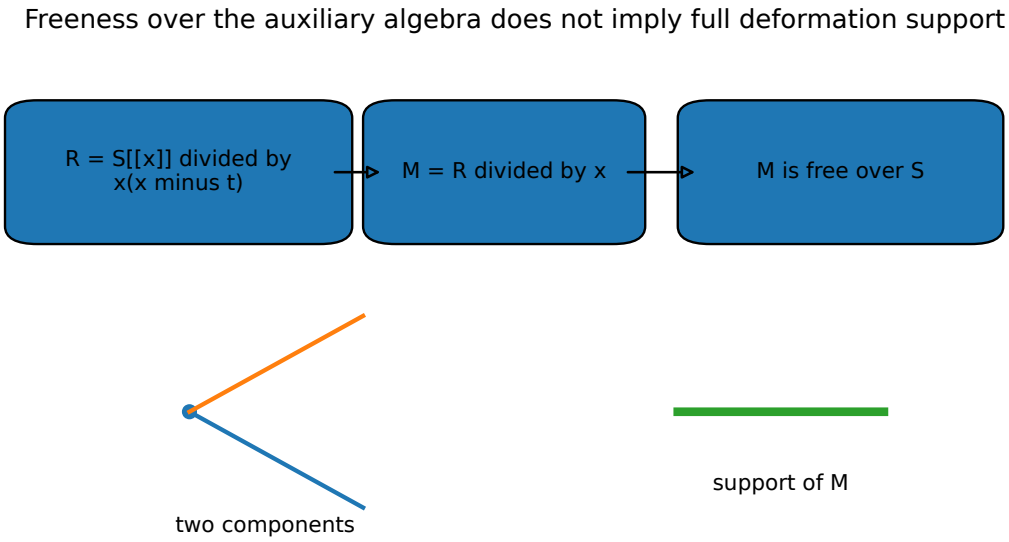


Figure 8. A module can be free over the auxiliary ring while occupying only one deformation component.

Derived patching replaces one module by a perfect complex. The correct support is detected after derived specialization.

Definition 6.14 (Derived support)**.** Let R be a commutative ring and let C be a perfect complex of R modules. Its derived support is

$$\operatorname{Supp}_R(C) = \left\{ \mathfrak{p} \in \operatorname{Spec} R : C \overset{\mathbf{L}}{\otimes}_R \kappa(\mathfrak{p}) \not\simeq 0 \right\}.$$

Standard Fact 6.15 (Cohomological description of derived support)**.** If R is Noetherian and C is perfect, then

$$\operatorname{Supp}_R(C) = \bigcup_i \operatorname{Supp}_R H^i(C).$$

In particular, derived support is closed and is empty exactly when $C \simeq 0$ [26, 11].

Theorem 6.16 (Derived quotient support)**.** *Let R be a commutative ring, let $I \subseteq R$ be an ideal, and let C be a perfect complex over R. Under $\operatorname{Spec}(R/I) \cong V(I)$, one has*

$$\boxed{\operatorname{Supp}_{R/I}\left(C \overset{\mathbf{L}}{\otimes}_R R/I \right) = \operatorname{Supp}_R(C) \cap V(I).}$$

Proof. Choose a bounded complex P of finite projective R modules representing C. Scalar extension to a commutative R algebra S is computed degree by degree as $S \otimes_R P$. The canonical symmetry of tensor products gives the displayed order $C \otimes^{\mathbf{L}}_R S$. Scalar extension preserves the bounds, finite generation, and projectivity of the terms. A derived isomorphism between two such projective models lifts to a homotopy equivalence, which scalar extension preserves. The resulting derived object and its residue field fibers are therefore independent of the chosen model.

Let $q \in \operatorname{Spec}(R/I)$, and let $\mathfrak{p}$ be its inverse image in R. The composite point $R \to R/I \to \kappa(q)$ has kernel $\mathfrak{p}$, so it factors through a field embedding $\kappa(\mathfrak{p}) \to \kappa(q)$. The ordinary tensor associator in every degree gives an isomorphism of complexes

$$\kappa(q) \otimes_{R/I} \left((R/I) \otimes_R P \right) \cong \kappa(q) \otimes_R P \cong \kappa(q) \otimes_{\kappa(\mathfrak{p})} \left(\kappa(\mathfrak{p}) \otimes_R P \right).$$

Extension of fields is faithfully flat. It preserves and reflects exactness in each degree, hence also acyclicity. The left-hand complex is thus nonzero in the derived category exactly when the residue field fiber of C at $\mathfrak{p}$ is nonzero. This proves the support identity on corresponding primes. The image of $\operatorname{Spec}(R/I) \to \operatorname{Spec} R$ is exactly $V(I)$, giving the stated ambient spectrum equality, including the case $I = R$. □

Definition 6.17 (Derived patching datum)**.** A derived patching datum consists of a Noetherian ring R_∞, an ideal $\mathfrak{a}$, a perfect complex C_∞ over R_∞, and identifications

$$R \cong R_\infty/\mathfrak{a}, \qquad C \simeq C_\infty \overset{\mathbf{L}}{\underset{R_\infty}{\otimes}} R.$$

Theorem 6.18 (Derived patching support descent)**.** *Let $(R_\infty, \mathfrak{a}, C_\infty, R, C)$ be a derived patching datum. If*

$$\operatorname{Supp}_{R_\infty}(C_\infty) = \operatorname{Spec} R_\infty,$$

then

$$\boxed{\operatorname{Supp}_R(C) = \operatorname{Spec} R.}$$

For every point $x : R \to E'$ to a field, the derived fiber

$$C \overset{\mathbf{L}}{\underset{R,x}{\otimes}} E'$$

is nonzero.

Proof. Apply theorem 6.16 to the patched complex and use the actual ring identification $R \cong R_\infty/\mathfrak{a}$. Full patched support restricts to $V(\mathfrak{a})$, which corresponds to all of $\operatorname{Spec} R$. The supplied derived isomorphism with C preserves this support by the projective model argument in that theorem.

For a field valued point $x : R \to E'$, put $\mathfrak{p} = \ker x$. The point factors as $R \to \kappa(\mathfrak{p}) \to E'$, where the second map is a field embedding. Tensor association on a projective model identifies the fiber at x with the extension to E' of the fiber at $\mathfrak{p}$. The latter is nonzero by full support. Faithful flatness of the field extension reflects exactness in every degree, so the fiber at x is nonzero as well. This applies to every field valued point, with no closed point restriction. □

Corollary 6.19 (A nonzero cohomological degree survives)**.** *Under the hypotheses of theorem 6.18, for every field valued point $x : R \to E'$, there is an integer i such that*

$$H^i\left(C \overset{\mathbf{L}}{\underset{R,x}{\otimes}} E'\right) \neq 0.$$

Proof. The derived fiber is nonzero by theorem 6.18. If every cohomology object vanished, a cochain model of that fiber would be acyclic. Its image in the derived category would then be zero, a contradiction. Hence some integer degree has nonzero cohomology. □

Remark 6.20. The support theorem does not decide the cohomological degree in which the point occurs. It also does not prove that the surviving class is classical. Those are separate arithmetic inputs.

7 Transport through an isomorphism $R \cong \mathbb{T}$

An isomorphism between a deformation ring and a Hecke algebra is an exact coordinate change. It transports support and finite presentation data, but it does not create a supported module.

Theorem 7.1 (Support and Fitting transport)**.** *Let $\varphi : R \to \mathbb{T}$ be an isomorphism of Noetherian rings. Let M be a finite $\mathbb{T}$ module, viewed as an R module through φ. Then the homeomorphism*

$$\varphi^* : \operatorname{Spec} \mathbb{T} \longrightarrow \operatorname{Spec} R$$

satisfies

$$\boxed{\varphi^*\big(\operatorname{Supp}_{\mathbb{T}}(M)\big) = \operatorname{Supp}_R(M).}$$

For every $j \geq 0$,

$$\boxed{\varphi\left(\operatorname{Fitt}_j^R(M)\right) = \operatorname{Fitt}_j^{\mathbb{T}}(M).}$$

Proof. Localization commutes with transport along an isomorphism, so a localization of M vanishes on one side exactly when the corresponding localization vanishes on the other. This proves the support identity. A finite presentation matrix over R is carried entry by entry to a presentation matrix over $\mathbb{T}$. The minors generating each Fitting ideal are therefore carried to the corresponding minors. □

Corollary 7.2 (Full support is invariant under $R \cong \mathbb{T}$)**.** *In the setting of theorem 7.1, M has full R support if and only if it has full $\mathbb{T}$ support.*

Proof. Apply the support homeomorphism in theorem 7.1. □

Proposition 7.3 (A quotient can erase a component)**.** *Let*

$$R = k[[x, y]]/(xy)$$

and let

$$\mathbb{T} = R/(x) \cong k[[y]].$$

Viewed as an R module, $\mathbb{T}$ has support $V(x)$, which misses the generic point (y) of the other irreducible component. Thus a surjection $R \twoheadrightarrow \mathbb{T}$ does not by itself imply full component occupancy.

Proof. The annihilator of the R module $\mathbb{T}$ is (x). Hence its support is $V(x)$ by theorem 6.2. The prime (y) does not contain x, so it is not in this support. □

Theorem 7.4 (Field fiber criterion)**.** *Let R be a commutative ring, let M be a finite R module, and let $x : R \to E'$ be a homomorphism to a field. Put $\mathfrak{p} = \ker x$. Then*

$$\boxed{M \otimes_{R,x} E' \neq 0 \quad \Longleftrightarrow \quad \mathfrak{p} \in \operatorname{Supp}_R(M).}$$

Proof. The map x factors through the local ring $R_{\mathfrak{p}}$ and then through an embedding of the residue field $\kappa(\mathfrak{p})$ into E'. Since M is finite, Nakayama's lemma gives

$$M_{\mathfrak{p}} \neq 0 \quad \Longleftrightarrow \quad M \otimes_R \kappa(\mathfrak{p}) \neq 0.$$

Extension of scalars from $\kappa(\mathfrak{p})$ to E' preserves nonzero vector spaces. □

Corollary 7.5 (Supported deformation points give nonzero Hecke fibers)**.** *Let $R \cong \mathbb{T}$, and let M be a finite Hecke module. If a point $x : R \to E'$ lies in $\operatorname{Supp}_R(M)$, then the corresponding Hecke fiber*

$$M \otimes_{\mathbb{T},x} E'$$

is nonzero.

Proof. Transport the support through theorem 7.1, then apply theorem 7.4. □

8 Finite Hecke packets and infinite occupancy

Finite Hecke carriers admit canonical packets even when the Hecke action is not semisimple. The packets come from the Artin local factors of the finite commutative Hecke algebra.

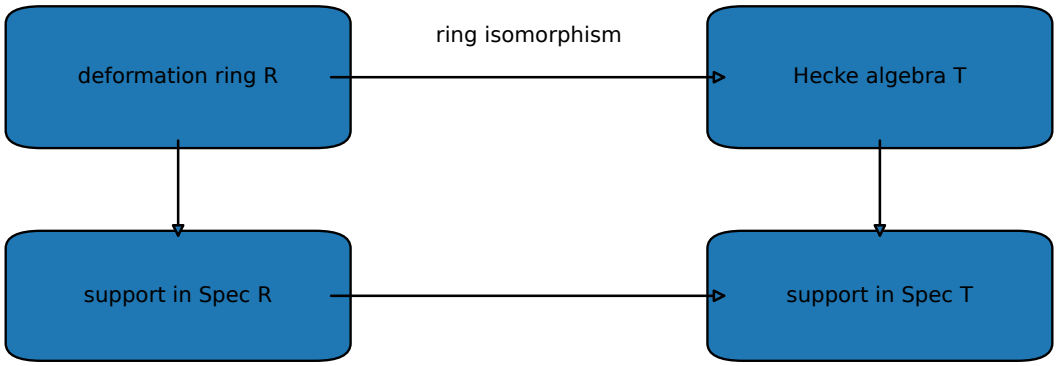


Figure 9. An isomorphism transports occupied support exactly. A quotient can remove a component.

Standard Fact 8.1 (Artin decomposition)**.** Let E' be a field and let $\mathbb{T}$ be a finite dimensional commutative E' algebra. Then

$$\boxed{\mathbb{T} \cong \prod_{\mathfrak{m} \in \operatorname{Max} \mathbb{T}} \mathbb{T}_{\mathfrak{m}}.}$$

Each factor is an Artin local algebra. Equivalently, there is a unique complete family of central idempotents $(e_{\mathfrak{m}})$ such that

$$e_{\mathfrak{m}} e_{\mathfrak{n}} = 0 \quad (\mathfrak{m} \neq \mathfrak{n}), \qquad \sum_{\mathfrak{m}} e_{\mathfrak{m}} = 1,$$

and $e_{\mathfrak{m}} \mathbb{T} \cong \mathbb{T}_{\mathfrak{m}}$ [2, 15].

Theorem 8.2 (Canonical Hecke packet decomposition)**.** *Let $\mathbb{T}$ be a finite dimensional commutative algebra over a field E', and let V be a finite $\mathbb{T}$ module. Then*

$$\boxed{V = \bigoplus_{\mathfrak{m} \in \operatorname{Max} \mathbb{T}} e_{\mathfrak{m}} V.}$$

For every maximal ideal $\mathfrak{m}$, the canonical map gives

$$\boxed{e_{\mathfrak{m}} V \cong V_{\mathfrak{m}}.}$$

The packet $e_{\mathfrak{m}} V$ is nonzero exactly when $\mathfrak{m} \in \operatorname{Supp}_{\mathbb{T}}(V)$.

Proof. The idempotent relations in standard fact 8.1 give the direct sum decomposition. Localization at $\mathfrak{m}$ kills every factor except the $\mathfrak{m}$ factor, where $e_{\mathfrak{m}}$ becomes one. Thus the localization map identifies $e_{\mathfrak{m}} V$ with $V_{\mathfrak{m}}$. The final statement is the definition of support. □

Corollary 8.3 (Simultaneous generalized eigenspaces)**.** *Assume that E' is algebraically closed and that $\mathbb{T}$ is generated by commuting operators $T_1, \dots, T_d$ on V. For a maximal ideal*

$$\mathfrak{m} = (T_1 - \lambda_1, \dots, T_d - \lambda_d),$$

there is an integer $N \geq 1$ such that

$$\boxed{e_{\mathfrak{m}} V = \left\{ v \in V : (T_i - \lambda_i)^N v = 0 \text{ for every } i \right\}.}$$

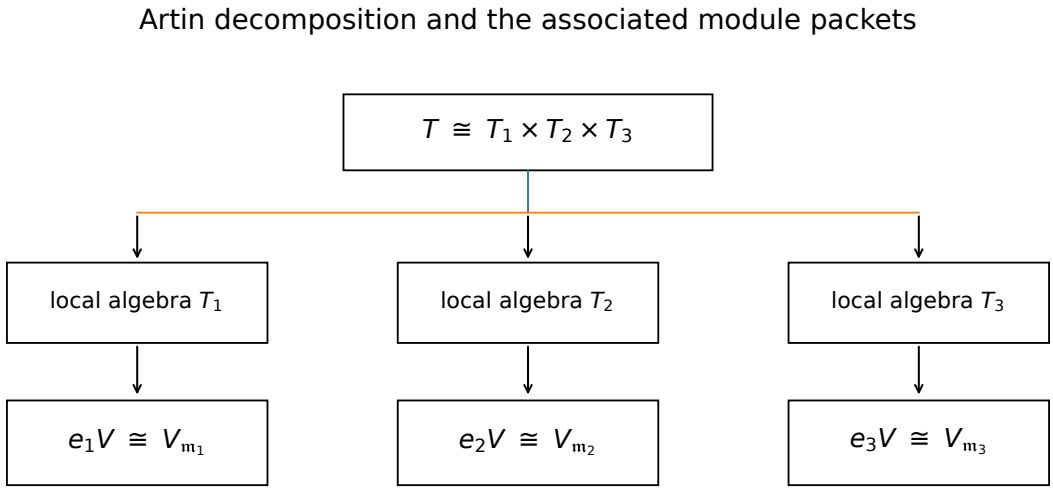


Figure 10. A finite commutative Hecke algebra splits into canonical Artin local packets.

Thus the canonical Artin packet is the simultaneous generalized eigenspace.

Proof. The maximal ideal of the Artin local algebra $\mathbb{T}_{\mathfrak{m}}$ is nilpotent. Hence a common power of every $T_i - \lambda_i$ kills the local packet. On every other local factor, at least one $T_i - \lambda_i$ is a unit, so no nonzero vector can satisfy all the displayed nilpotence conditions.

Conversely, suppose every $(T_i - \lambda_i)^N$ kills v. Every monomial of total degree greater than $d(N-1)$ contains one factor to exponent at least N. Therefore a power of $\mathfrak{m}$ kills v, which places v in the $\mathfrak{m}$ primary packet. □

Corollary 8.4 (Rank and trace detection in characteristic zero)**.** *Assume that E' has characteristic zero. For every maximal Hecke packet,*

$$\boxed{\operatorname{Tr}_{E'}(e_{\mathfrak{m}}|V) = \operatorname{rank}_{E'}(e_{\mathfrak{m}}|V) = \dim_{E'} e_{\mathfrak{m}}V.}$$

In particular, the packet is occupied exactly when its projector has positive rank, equivalently nonzero trace.

Proof. An idempotent linear map has minimal polynomial dividing $X(X-1)$, which has distinct roots. It is diagonalizable with eigenvalues zero and one. Its trace and rank both equal the multiplicity of the eigenvalue one. A positive integer is nonzero in characteristic zero. □

Remark 8.5. The trace criterion can fail in positive characteristic when the packet dimension is divisible by the characteristic. The rank and localization criteria remain valid.

Example 8.6 (A nonsemisimple Hecke packet)**.** Let

$$\mathbb{T} = E'[u]/(u^2) \times E'[v]/(v^3)$$

and let $V = \mathbb{T}$ be the regular module. The two canonical central idempotents split

$$V = V_u \oplus V_v$$

with dimensions two and three. Multiplication by u is nilpotent on V_u, and multiplication by v is nilpotent on V_v. Neither local factor is semisimple, but the two maximal Hecke packets are isolated exactly by the Artin idempotents.

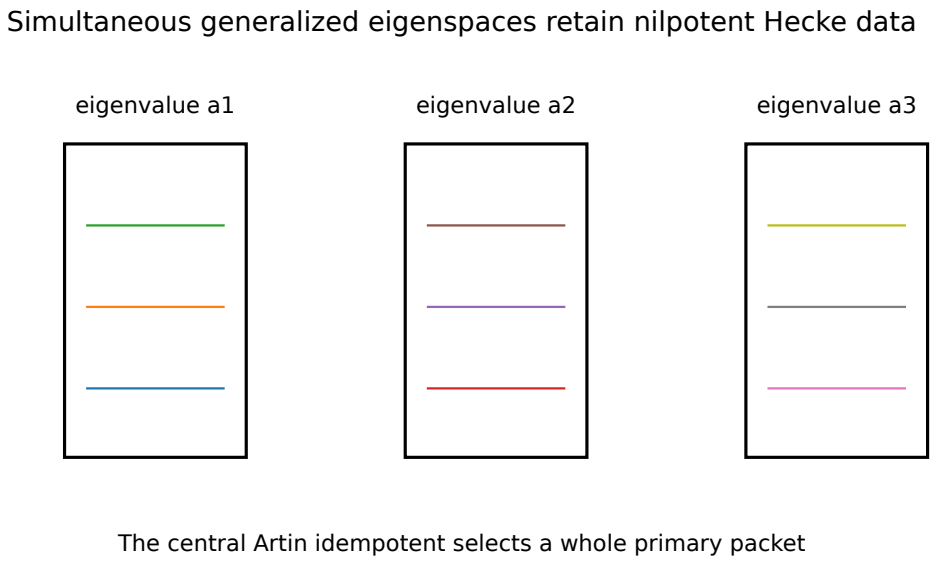


Figure 11. An Artin packet is a simultaneous generalized eigenspace, not necessarily an ordinary eigenspace.

Proof. The product decomposition gives the two central idempotents $(1,0)$ and $(0,1)$. Their ranges are the two local factors. The classes of u and v are nilpotent of orders two and three, respectively. □

The next two theorems distinguish a fixed finite carrier from a sequence of changing carriers.

Theorem 8.7 (Nested packets in one finite carrier)**.** *Let V be a finite dimensional vector space. Let $e_1, e_2, \dots$ be idempotents such that*

$$e_{m+1}e_m = e_m e_{m+1} = e_{m+1}$$

for every m. If $e_m V \neq 0$ for every m, then the descending chain stabilizes and

$$\boxed{\bigcap_{m\geq 1} e_m V \neq 0.}$$

Proof. The displayed relations give

$$e_{m+1}V \subseteq e_m V.$$

The nonnegative dimensions form a nonincreasing sequence of positive integers, so they eventually become constant. Once two consecutive dimensions agree, the corresponding inclusion is equality. Thus the ranges stabilize at a nonzero subspace, which is their intersection. □

Corollary 8.8 (Finite witness of incompatibility)**.** *Under the hypotheses of theorem 8.7, if*

$$\bigcap_{m\geq 1} e_m V = 0,$$

then $e_m V = 0$ for some finite m.

Proof. This is the contrapositive of theorem 8.7. □

Definition 8.9 (Stable core)**.** Let

$$V_1 \longleftarrow V_2 \longleftarrow V_3 \longleftarrow \cdots$$

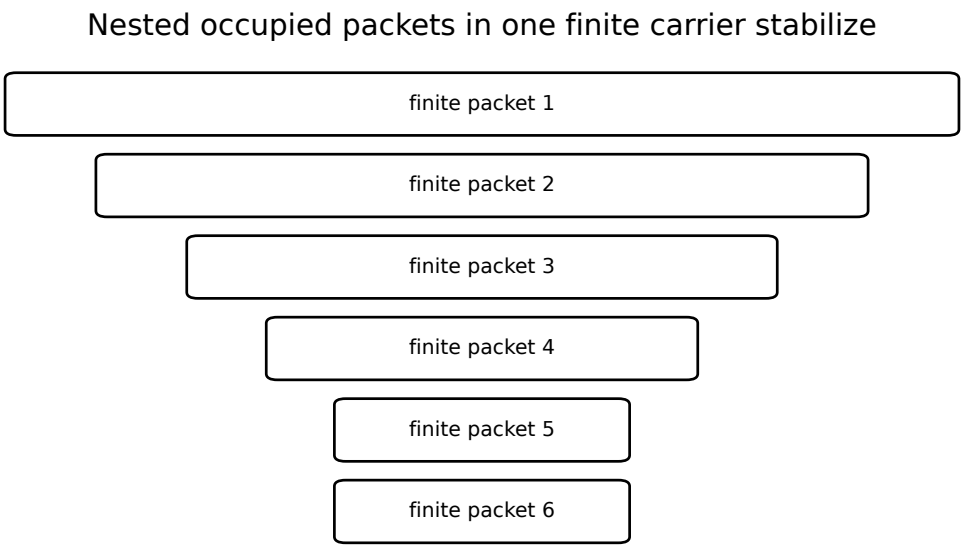


Figure 12. Nested nonzero packets in one finite carrier stabilize after finitely many steps.

be an inverse system of finite dimensional vector spaces. For $n \geq m$, write $f_{n,m} : V_n \to V_m$. The stable core at level m is

$$Y_m = \bigcap_{n \geq m} \operatorname{im}(f_{n,m}).$$

Lemma 8.10 (Surjectivity on stable cores)**.** *For every m, the transition map $V_{m+1} \to V_m$ induces a surjection*

$$\boxed{Y_{m+1} \twoheadrightarrow Y_m.}$$

Proof. Fix $y \in Y_m$. For $n \geq m+1$, let

$$A_n = \{z \in \operatorname{im}(f_{n,m+1}) : f_{m+1,m}(z) = y\}.$$

Each A_n is a nonempty affine subspace of V_{m+1}, and the sequence (A_n) is descending. A descending sequence of affine subspaces in a finite dimensional vector space stabilizes whenever every term is nonempty. Choose an element in its stable value. It lies in Y_{m+1} and maps to y. □

Theorem 8.11 (Nonzero stable core criterion)**.** *Let (V_m) be an inverse system of finite dimensional vector spaces. If*

$$Y_m \neq 0$$

for every m, then

$$\boxed{\varprojlim_m V_m \neq 0.}$$

More precisely, the inverse limit contains a compatible sequence whose first coordinate is any prescribed nonzero element of Y_1.

Proof. Choose $0 \neq y_1 \in Y_1$. By lemma 8.10, choose recursively $y_{m+1} \in Y_{m+1}$ mapping to y_m. Every y_m is nonzero because it maps to the nonzero vector y_1. The sequence (y_m) is a nonzero element of the inverse limit. □

Corollary 8.12 (Surjective systems)**.** *If every transition map $V_{m+1} \to V_m$ is surjective and every V_m is nonzero, then the inverse limit is nonzero.*

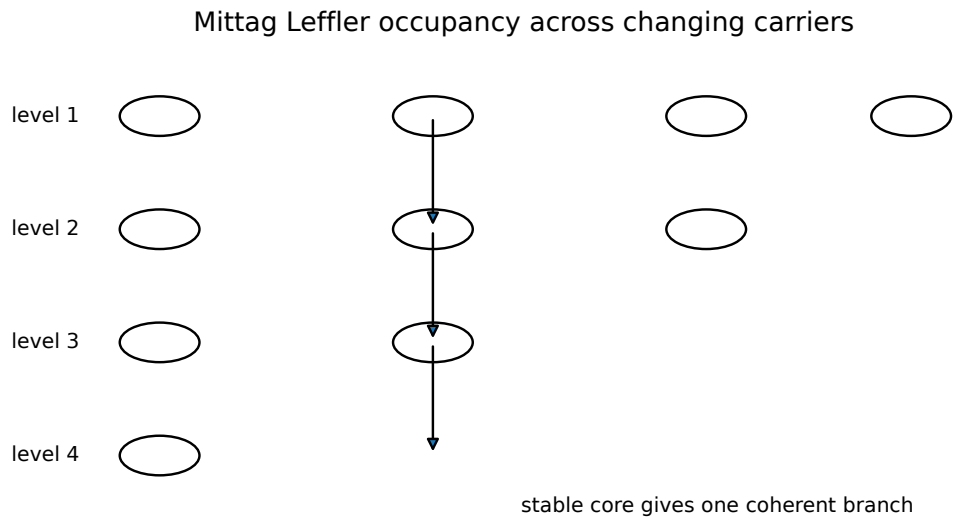


Figure 13. Changing finite carriers require a nonzero stable core, not only nonzero individual levels.

Proof. Surjectivity gives $Y_m = V_m$. Apply theorem 8.11. □

Example 8.13 (Tail escape in an infinite carrier)**.** Let $V = \ell^2(\mathbb{N})$, and let P_m be the orthogonal projection onto the closed span of

$$e_m, e_{m+1}, e_{m+2}, \dots.$$

Every range $P_m V$ is nonzero and the ranges are nested, but

$$\boxed{\bigcap_{m \geq 1} P_m V = 0.}$$

Thus the finite carrier hypothesis in theorem 8.7 cannot be removed.

Proof. A vector in every tail range has its first $m-1$ coordinates equal to zero for every m. All of its coordinates vanish. □

9 Pointwise automorphy from support

Support produces a nonzero Hecke fiber. A realization theorem is still required to turn that fiber into a classical automorphic representation and to identify its Galois representation.

Definition 9.1 (Automorphic Hecke datum)**.** An automorphic Hecke datum consists of the following objects.

(1) A commutative $\mathcal{O}$ algebra $\mathbb{T}$.
(2) A finite $\mathbb{T}$ module H, or a perfect $\mathbb{T}$ complex C, obtained from an automorphic or cohomological carrier.
(3) For every unramified place v outside a finite set, a polynomial
$$P_v(X) \in \mathbb{T}[X].$$
(4) A declared characteristic zero locus on which a nonzero field fiber is known to be classical and is known to carry an attached Galois representation whose Frobenius polynomial at v is the specialization of $P_v(X)$.

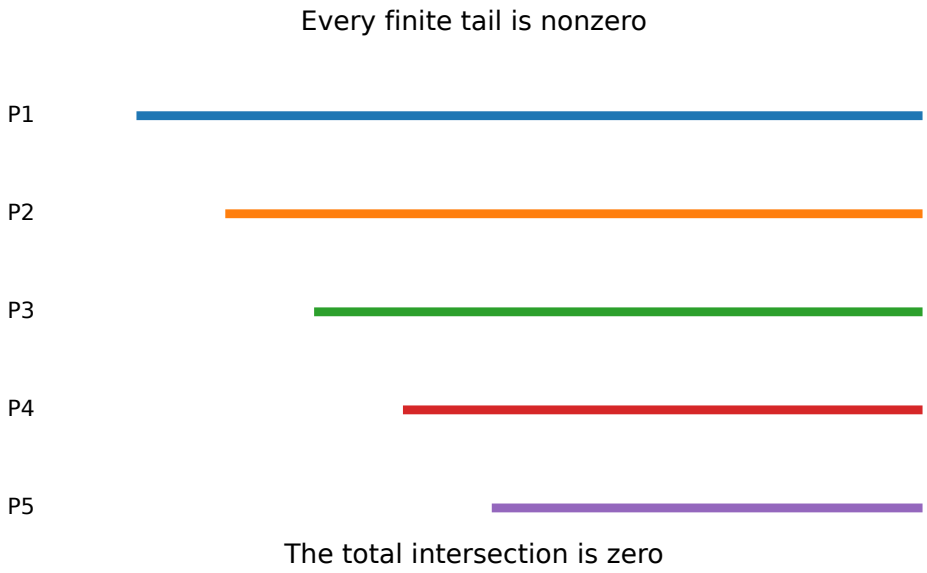


Figure 14. Nested nonzero ranges can escape to infinity in an infinite dimensional carrier.

Standard Fact 9.2 (Chebotarev and Brauer Nesbitt identification)**.** Let E' be a Hausdorff topological field of characteristic zero, with continuous ring operations. Let

$$\rho_1, \rho_2 : G_{F,S} \longrightarrow \mathrm{GL}_n(E')$$

be continuous semisimple representations. If their characteristic polynomials agree at every Frobenius element outside a density zero exceptional set, then

$$\boxed{\rho_1 \cong \rho_2.}$$

This follows from Chebotarev density and the Brauer Nesbitt theorem [5, 18, 12].

Theorem 9.3 (Support gives a Hecke eigensystem)**.** *Let $R \cong \mathbb{T}$, let H be a finite $\mathbb{T}$ module, and let $x : R \to E'$ be a field valued point. If*

$$\ker x \in \mathrm{Supp}_R(H),$$

then

$$\boxed{H_x := H \otimes_{\mathbb{T},x} E' \neq 0.}$$

Every element of $\mathbb{T}$ acts on H_x through the scalar prescribed by x.

Proof. Nonvanishing follows from corollary 7.5. The tensor product relation

$$th \otimes 1 = h \otimes x(t)$$

shows that $t \in \mathbb{T}$ acts through $x(t)$. □

Theorem 9.4 (Pointwise Galois realization)**.** *Let E' be a Hausdorff topological field of characteristic zero, with continuous ring operations. Require both $\rho_x = x \circ \rho_R$ and the attached representation below to be continuous. Assume the following data.*

(1) A represented deformation ring R with universal representation

$$\rho_R : G_{F,S} \longrightarrow \mathrm{GL}_n(R).$$

(2) An isomorphism $R \cong \mathbb{T}$.

(3) An automorphic Hecke datum $(\mathbb{T}, H, (P_v))$.

(4) *A characteristic zero point* $x : R \to E'$ *lying in* $\operatorname{Supp}_R(H)$.

(5) *A classical automorphic representation* π_x *occurring in the fiber* H_x*, together with a semisimple attached representation*

$$\rho_{\pi_x} : G_{F,S} \longrightarrow \mathrm{GL}_n(E')$$

whose unramified characteristic polynomial is $x(P_v(X))$.

(6) *Equality*

$$\det\left(X - \rho_x(\mathrm{Frob}_v)\right) = x(P_v(X))$$

for $\rho_x = x \circ \rho_R$ *at every unramified place in a Chebotarev dense set.*

Then

$$\boxed{\rho_x^{\mathrm{ss}} \cong \rho_{\pi_x}.}$$

If ρ_x *is irreducible, then* $\rho_x \cong \rho_{\pi_x}$.

Proof. The coefficients of the characteristic polynomials of ρ_x and ρ_{π_x} are continuous and constant on conjugacy classes. Their equality on the declared dense union of Frobenius conjugacy classes therefore extends to every element of $G_{F,S}$, by Hausdorffness. A semisimplification has the same characteristic polynomials as the original representation. The algebraic Brauer Nesbitt input in standard fact 9.2 thus gives $\rho_x^{\mathrm{ss}} \cong \rho_{\pi_x}$. If ρ_x is irreducible, it is already semisimple. □

Corollary 9.5 (Full support gives pointwise automorphy on the realized locus)**.** *In the setting of theorem 9.4, suppose*

$$\operatorname{Supp}_R(H) = \operatorname{Spec} R.$$

For every characteristic zero point $x : R \to E'$ *with* $\ker x \in U \subseteq \operatorname{Spec} R$*, assume the coefficient topology and continuity hypotheses of theorem 9.4 and the declared classicality and Galois realization input. Then every such point is automorphic in the precise sense of theorem 9.4.*

Proof. Every point of U lies in the support, so theorem 9.3 gives a nonzero Hecke fiber. Apply the declared realization input and then theorem 9.4. □

Theorem 9.6 (Derived pointwise realization)**.** *Let* $e : R \cong \mathbb{T}$*, and let* C *be a perfect automorphic Hecke complex with full derived support. Choose a bounded finite projective model for* C*, supported in degrees* $[a, b]$*. For every characteristic zero point* $x : R \to E'$*, write* $y = x \circ e^{-1}$*. Then some integer* $i \in [a, b]$ *has*

$$H^i\left(C \overset{\mathbf{L}}{\underset{\mathbb{T},y}{\otimes}} E'\right) \neq 0.$$

Every $t \in \mathbb{T}$ *acts on this cohomology through* $y(t)$.

For the realization conclusion, give E' *a Hausdorff topological field structure. Supply a universal representation* ρ_R*, a continuous specialization* $\rho_x = x \circ \rho_R$*, and a semisimplification of* ρ_x*. Suppose its characteristic polynomials on a Chebotarev dense set are* $y(P_v(X))$*. Let the declared classicality range contain* $[a, b]$*. Assume that every nonzero Hecke eigensystem in that range is realized by a classical automorphic representation occurring in that degree, with a continuous semisimple attached representation whose characteristic polynomials on the same dense set are* $y(P_v(X))$*. With the standard Chebotarev density and Brauer Nesbitt inputs of standard fact 9.2, the specialization* ρ_x *is automorphic after semisimplification. If* ρ_x *is irreducible, it is itself isomorphic to the attached representation.*

Proof. Full derived support gives a nonzero fiber at y. Scalar extension of the chosen projective model preserves its degree bounds. Its cohomology therefore vanishes below a and above b. Since vanishing in every degree is equivalent to being zero in the derived category, some degree $i \in [a, b]$ has nonzero cohomology. Choose a nonzero vector in that cohomology module.

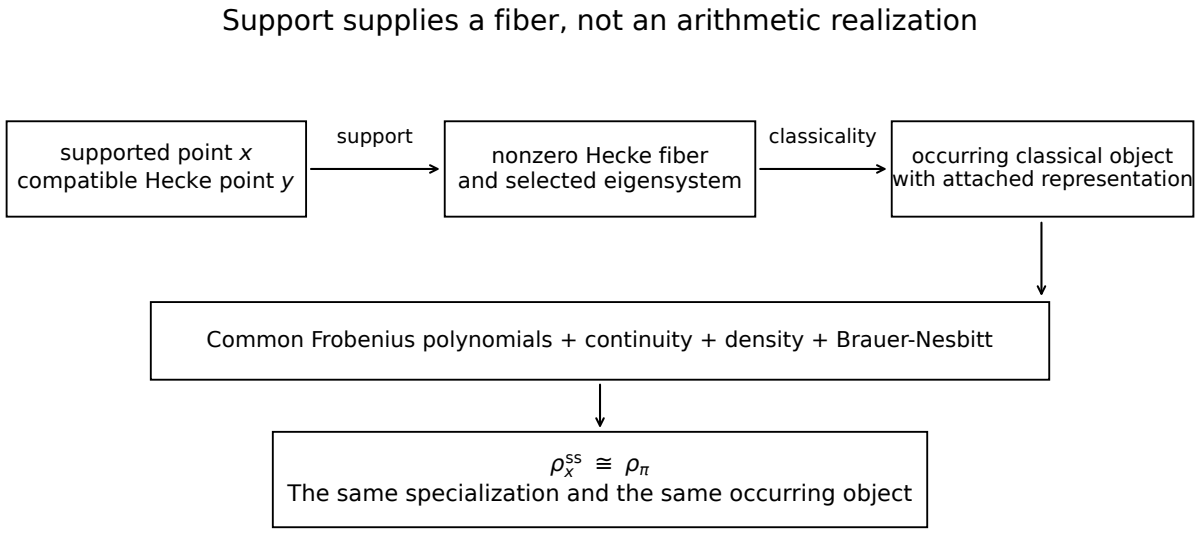


Figure 15. Support gives a nonzero Hecke fiber. Classicality and Galois realization complete the pointwise argument.

Multiplication by $t \in \mathbb{T}$ is a cochain endomorphism of the original model. In the scalar extension the tensor relation identifies this endomorphism with $y(t)$ times the identity. Passing to the derived category and then to cohomology preserves this equality. Thus the chosen vector is a common Hecke eigenvector with character y.

The interval inclusion puts this eigensystem in the declared classicality range. Apply the realization input to obtain an occurring automorphic representation and its continuous semisimple attached representation. Its Frobenius characteristic polynomials agree with those of ρ_x on the prescribed dense set. Continuity extends the equality to every group element. The supplied semisimplification preserves these polynomials, so the general Brauer Nesbitt input gives an invertible linear intertwiner. If ρ_x is irreducible, it is already semisimple and the same comparison applies directly. □

Remark 9.7. Derived support does not identify the cohomological degree, Hodge type, or classical weight of the surviving eigensystem. Those properties must come from the external local and global realization theorem.

10 Compatible systems and companion completion

Pointwise automorphy at one coefficient place and existence of companions at other coefficient places are different statements. They meet only after a common algebraic Frobenius spine has been fixed.

Definition 10.1 (Algebraic Frobenius spine)**.** An algebraic Frobenius spine consists of a number field L, a finite set of places S of F, and polynomials

$$P_v(X) \in L[X]$$

for every finite place $v \notin S$, together with any additional local labels required by the declared compatibility problem.

Definition 10.2 (Finite companion forest)**.** Fix a coefficient place λ of L. A finite companion forest for the spine is an inverse system of finite sets

$$X_{\lambda,1} \longleftarrow X_{\lambda,2} \longleftarrow X_{\lambda,3} \longleftarrow \cdots$$

whose points are admissible realizations modulo λ^m, with edges given by reduction. A complete realization is admissible for this forest when all its finite reductions belong to the displayed levels; the empty level obstruction below concerns this declared admissibility problem.

Theorem 10.3 (Companion branch criterion)**.** *Suppose every level $X_{\lambda,m}$ is finite and nonempty, and every point reduces along the displayed forest to the fixed residual root. Then*

$$\boxed{\varprojlim_m X_{\lambda,m} \neq \varnothing.}$$

Proof. The disjoint union of the levels is an infinite rooted tree. It is locally finite because every level is finite. It contains a vertex at every depth. König's infinity lemma gives an infinite branch [10]. That branch is a point of the inverse limit. □

Corollary 10.4 (Empty finite companion obstruction)**.** *If $X_{\lambda,m} = \varnothing$ for some m, then no complete λ adic realization admissible for this forest exists.*

Proof. A complete realization would reduce to a point at every finite level. □

Finite forests alone give compatible classes. To obtain actual continuous representations, we use the following explicit coefficient and admissibility data. They also delimit the meaning of completion in the next theorem.

Fix a topological group Γ, a finite matrix index set D, and a set Λ of coefficient indices. For each λ, let $\mathcal{O}_\lambda$ be a commutative ring, complete and separated for a finitely generated ideal I_λ, with finite residue ring $\mathcal{O}_\lambda/I_\lambda$. Give $\mathcal{O}_\lambda$ its ideal adic topology and every quotient its discrete topology. Let

$$i_\lambda : \mathcal{O}_\lambda \longrightarrow K_\lambda, \qquad e_\lambda : L \longrightarrow K_\lambda$$

be supplied ring homomorphisms, where L and K_λ are fields, K_λ has a declared topology, and i_λ is continuous.

For every positive integer n, let $X_{\lambda,n}$ be a finite nonempty set of conjugacy classes of actual continuous representations

$$\Gamma \longrightarrow \mathrm{GL}_D(\mathcal{O}_\lambda/I_\lambda^n).$$

Conjugacy means change of basis by an invertible matrix over that quotient. Assume reduction preserves these sets and that $X_{\lambda,1}$ consists of the declared residual class. Each admissible class must have a continuous representative. The auxiliary level zero is the unique class over the trivial quotient and imposes no extra occupancy hypothesis.

Choose elements $\gamma_v \in \Gamma$ on a sample set Ω, with declared good subsets $\Omega_\lambda \subseteq \Omega$. In the arithmetic application these are the chosen Frobenius elements at the permitted places. For each $v \in \Omega_\lambda$, supply an integral polynomial $P_{\lambda,v} \in \mathcal{O}_\lambda[X]$ such that

$$i_\lambda(P_{\lambda,v}) = e_\lambda(P_v) \quad \text{in } K_\lambda[X].$$

Here and below a ring homomorphism applied to a polynomial acts on its coefficients. Assume every representative of every class in $X_{\lambda,n}$ has characteristic polynomial $P_{\lambda,v}$ mod I_λ^n at γ_v. Also supply a subgroup $H_\lambda \leq \Gamma$ on which all these finite representations are trivial. This is an inertia condition when H_λ is the specified inertia subgroup.

Finally, put

$$B_\lambda = \mathcal{O}_\lambda[x_{g,a,b} : g \in \Gamma,\ a, b \in D].$$

For each declared local condition index u, supply an ideal $J_{\lambda,u} \subseteq B_\lambda$. Evaluation at a representation sends $x_{g,a,b}$ to its (a,b) matrix entry at g. Assume that evaluation at every admissible finite representative kills $J_{\lambda,u}$, with polynomial coefficients reduced modulo I_λ^n. For this class based theorem, the equations must hold in every admissible frame. A condition imposed only in one chosen frame needs a separate compatible frame argument. No arbitrary arithmetic condition is identified with such an ideal without a separate argument.

Theorem 10.5 (Conditional simultaneous companion completion)**.** *Under the coefficient and finite admissibility hypotheses above, one can choose continuous representations*

$$\rho_\lambda : \Gamma \longrightarrow \mathrm{GL}_D(\mathcal{O}_\lambda) \qquad (\lambda \in \Lambda)$$

whose reduction class belongs to $X_{\lambda,n}$ for every positive n. They retain the declared residual class. After applying i_λ, they are continuous, trivial on H_λ, and satisfy

$$\det(X - i_\lambda(\rho_\lambda(\gamma_v))) = e_\lambda(P_v), \qquad v \in \Omega_\lambda.$$

Their integral evaluation points factor through every actual quotient $B_\lambda / J_{\lambda,u}$.

Proof. The finitely generated ideal and finite residue ring give finite coefficient quotients at every precision. An individual conjugacy class has finitely many frames because the group of invertible basis matrices over each quotient is finite. Thus the finitely many admissible classes give a finite nonempty set of actual continuous representations at each positive level. Continuity is invariant under conjugation. Reduction gives total maps between these sets. Compactness, as in theorem 10.3, selects strictly compatible representatives. This step neither assumes transition surjectivity nor lifts a prescribed conjugating matrix.

Completeness and separatedness identify $\mathrm{GL}_D(\mathcal{O}_\lambda)$ with the compatible inverse limit of the quotient matrix groups. Lift entries and inverse entries to construct this identification. The two reduction and reconstruction identities hold entrywise. Finite quotient continuity implies continuity for the ideal adic topology, including the matrix inverse, so the reconstructed representation is continuous.

Each coefficient of its characteristic polynomial agrees with the prescribed integral coefficient modulo every positive power of I_λ. Separatedness gives equality of the full polynomials. The same argument applied to the matrix entries of $\rho_\lambda(g) - 1$ gives triviality on H_λ. Continuous coefficient change and the matching equations for $P_{\lambda,v}$ give the asserted common spine over the local fields.

For $p \in J_{\lambda,u}$, evaluation commutes with reduction. Its value is zero modulo every positive power of I_λ, hence is zero integrally by separatedness. Therefore the evaluation homomorphism kills the whole ideal and factors uniquely through $B_\lambda / J_{\lambda,u}$. Choose a realization for each λ. □

The quotient conclusion is a proved effectivity statement for the declared polynomial relations. For local conditions presented by continuous maps from represented local rings, corollary 3.2 also accounts for the closed ideal generated by their defining ideals. Neither statement constructs a deformation ring for a condition that has not been shown representable. Hodge conditions, Weil–Deligne conditions, integral lifts of arbitrary Frobenius polynomials, and the finite occupancy hypotheses still require their own arithmetic inputs. The theorem retains a residual conjugacy class, not a prescribed residual matrix frame, and asserts no uniqueness up to integral conjugacy.

Standard Fact 10.6 (Strict compatibility and one member determination)**.** A strictly compatible semisimple system is determined at every coefficient place by its common unramified characteristic polynomials. If an automorphic representation gives a compatible system with the same polynomials, then Chebotarev density and Brauer Nesbitt identify the two systems member by member after semisimplification [23, 3].

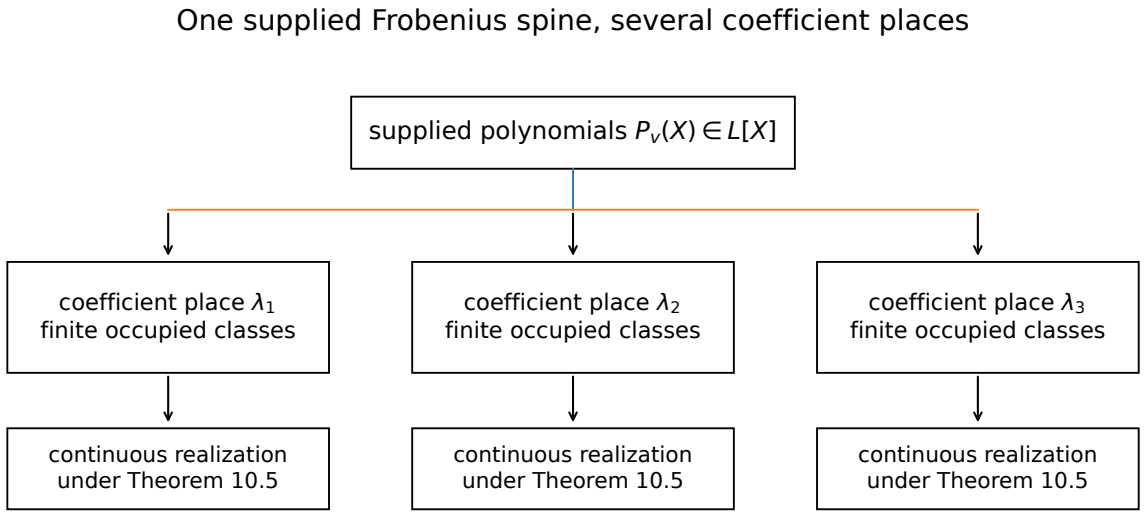


Figure 16. Under the coefficient and effectivity hypotheses of theorem 10.5, local companion branches realize the Frobenius spine fixed in advance.

Theorem 10.7 (One member propagation in a fixed system)**.** *Let*

$$\{\rho_\lambda\}_{\lambda\in\Lambda}$$

be a semisimple strictly compatible system with Frobenius spine (P_v). Suppose that for one coefficient place λ_0, the representation ρ_{λ_0} is identified with the Galois representation attached to an automorphic representation π. Assume that the compatible system attached to π has the same polynomials $P_v(X)$. Then for every $\lambda \in \Lambda$,

$$\boxed{\rho_\lambda \cong \rho_{\pi,\lambda}^{\mathrm{ss}}.}$$

Proof. For every λ, the two semisimple representations have the same unramified characteristic polynomials at all places outside one finite set. Apply standard fact 9.2. □

Remark 10.8. The theorem propagates automorphy inside a compatible system that already exists. It does not construct a missing coefficient place member and does not prove universal companion existence.

Proposition 10.9 (Finite Frobenius separator)**.** *Let $t_1, \dots, t_r$ be linearly independent functions on a set Ω with values in a field E'. Then there are points $\omega_1, \dots, \omega_r \in \Omega$ such that the matrix*

$$\big(t_i(\omega_j)\big)_{j,i}$$

is invertible. In particular, when the trace functions of finitely many compatible constituents are linearly independent on the unramified Frobenius set, finitely many Frobenius traces recover their multiplicities in any known semisimple carrier.

Proof. Let

$$W = \operatorname{span}\{(t_1(\omega), \dots, t_r(\omega)) : \omega \in \Omega\} \subseteq E'^r.$$

If $W \neq E'^r$, a nonzero linear functional on E'^r would vanish on every evaluation vector, giving a nontrivial linear relation among the functions. Thus $W = E'^r$. Choose r evaluation vectors forming a basis. □

11 The coefficient prime two

The auxiliary augmentation algebra works at every coefficient prime. What changes at two is the availability of integral character projectors and, in many applications, the residual image and local deformation hypotheses.

Proposition 11.1 (Failure of integral sign projectors)**.** *Let $C_2 = \langle g \rangle$. Over $\mathbb{Q}_2$, the two sign projectors are*

$$\boxed{e_+ = \frac{1+g}{2}, \qquad e_- = \frac{1-g}{2}.}$$

Neither belongs to $\mathbb{Z}_2[C_2]$. After reduction modulo two,

$$\boxed{\mathbb{F}_2[C_2] \cong \mathbb{F}_2[X]/(X^2), \qquad X = g - 1.}$$

This ring is local and has no nontrivial idempotents.

Proof. The displayed rational elements are the unique primitive idempotents because they project to the two characters of C_2. Their coefficients contain $1/2$, so they are not integral. In characteristic two,

$$(g-1)^2 = g^2 - 2g + 1 = 0.$$

The map $X \mapsto g - 1$ gives the displayed isomorphism. The quotient by (X) is $\mathbb{F}_2$, and (X) is nilpotent. Thus the ring is local. □

Corollary 11.2 (Dyadic coinvariant recovery)**.** *Let Δ be a finite two group, let $\Lambda = \mathcal{O}[\Delta]$ with $\mathcal{O}$ a finite extension of $\mathbb{Z}_2$, and let M be finite free over Λ. Then*

$$\boxed{M \otimes_{\Lambda,\epsilon} \mathcal{O} \cong M/\mathfrak{a}M}$$

is free over $\mathcal{O}$ of the same Λ rank. No semisimple character decomposition is required.

Proof. Use the right module convention fixed in section 5. Quotienting by all balancing relations identifies $M \otimes_{\Lambda,\epsilon} \mathcal{O}$ with $M/\mathfrak{a}M$, without any commutativity assumption on Δ. Applying ϵ to the coordinates of a finite Λ basis gives an $\mathcal{O}$ basis with the same index set. □

Remark 11.3. The corollary is an algebraic statement about augmentation and coinvariants. It does not supply a coefficient two adequacy theorem, a local deformation ring with the desired geometry, or a classicality theorem. Those inputs must be checked in the automorphy lifting theorem being used.

12 Conditional automorphic completion

The following theorem assembles the preceding statements as a chain of gates. Each gate has a distinct mathematical source.

Definition 12.1 (Ordinary completion datum at a point)**.** An ordinary completion datum at a characteristic zero point $x : R \to E'$ consists of the following objects.

(1) A represented global deformation ring R with declared local quotient conditions and universal representation ρ_R.

(2) An ordinary patching datum
$$(R_\infty, \mathfrak{a}, M_\infty, R, H)$$
whose finite level module H is an automorphic Hecke carrier.

(3) Full patched support
$$\operatorname{Supp}_{R_\infty}(M_\infty) = \operatorname{Spec} R_\infty.$$

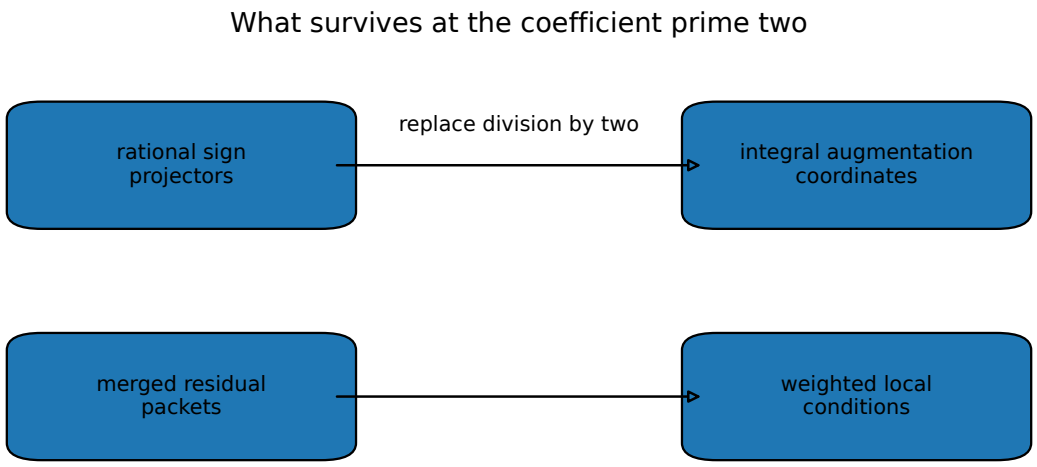


Figure 17. Augmentation survives at two, while semisimple character projectors and arithmetic hypotheses require separate treatment.

(4) An isomorphism
$$R \cong \mathbb{T}$$
with the Hecke algebra acting on H. The R action on H supplied by the patched control isomorphism is the restriction of this $\mathbb{T}$ action along the displayed isomorphism.

(5) An automorphic Hecke datum $(\mathbb{T}, H, (P_v))$ and a pointwise classicality and Galois realization theorem satisfying all the inputs of theorem 9.4 for the nonzero fiber at x: an occurring classical automorphic representation with a continuous semisimple attached representation; equality of the characteristic polynomials of both that representation and $\rho_x = x \circ \rho_R$ with $x(P_v(X))$ on the same declared Chebotarev dense union of Frobenius conjugacy classes; and the standard semisimplification, density, and general Brauer Nesbitt inputs. The coefficient topology and continuity requirements are retained in theorem 12.2.

Theorem 12.2 (Ordinary conditional automorphic completion)**.** *Let $x : R \to E'$ admit an ordinary completion datum. Assume E' is a Hausdorff topological field of characteristic zero with continuous ring operations, and that both $\rho_x = x \circ \rho_R$ and its attached comparison representation are continuous. Then*
$$\boxed{H \otimes_{\mathbb{T},x} E' \neq 0.}$$
The specialization
$$\rho_x = x \circ \rho_R$$
is identified after semisimplification with the Galois representation attached to a classical automorphic representation in that fiber. If ρ_x is irreducible, it is itself automorphic.

Proof. Full patched support descends to full support of H over R by theorem 6.11. The isomorphism $R \cong \mathbb{T}$ transports that support by theorem 7.1. The field fiber is nonzero by theorem 9.3. The declared classicality and realization theorem supplies an automorphic representation in the fiber and an attached Galois representation. The comparison with ρ_x follows from theorem 9.4. □

Definition 12.3 (Derived completion datum at a point)**.** A derived completion datum is defined in the same way, with a perfect patched complex C_∞ of full derived support and a finite level automorphic complex

$$C \simeq C_\infty \overset{\mathbf{L}}{\underset{R_\infty}{\otimes}} R$$

in place of the ordinary modules. Under the Hecke ring isomorphism $e : R \cong \mathbb{T}$, require an actual derived carrier identification between the scalar transport of this finite level complex and the automorphic Hecke complex. This identification retains the R action induced by patched control as the restriction of the $\mathbb{T}$ action along e.

Theorem 12.4 (Derived conditional automorphic completion)**.** *Let $x : R \to E'$ admit a derived completion datum, with Hecke identification $e : R \cong \mathbb{T}$, and put $y = x \circ e^{-1}$. Then*

$$\boxed{C \overset{\mathbf{L}}{\underset{\mathbb{T},y}{\otimes}} E' \not\simeq 0.}$$

Some integer cohomological degree contains a nonzero Hecke eigensystem with character y. Its degree lies between the bounds of any chosen finite projective model for C.

For the conditional conclusion, retain the Hausdorff coefficient topology, continuous specialization, supplied semisimplification, and common Frobenius polynomial hypotheses in theorem 9.6. If the declared classicality and Galois realization theorem applies to this eigensystem and supplies an occurring classical automorphic representation with a continuous semisimple attached representation and those Frobenius polynomials, then ρ_x is automorphic after semisimplification. If ρ_x is irreducible, it is itself isomorphic to the attached representation. This pointwise applicability condition does not require a uniform classicality assertion in other degrees.

Proof. Full support descends from the patched perfect complex by theorem 6.18. Scalar transport through e, followed by the actual derived isomorphism with the automorphic Hecke complex, gives full support for C. Its fiber at y is therefore nonzero. The bounded model and induced Hecke action argument in theorem 9.6 supplies a nonzero cohomology eigenvector in an integer degree within the model's bounds.

If realization applies to this eigensystem, it supplies the occurring automorphic object and a continuous semisimple attached representation with the declared Frobenius polynomials. The representation comparison in theorem 9.6 then identifies the supplied semisimplification of ρ_x with that attached representation. Only applicability to this eigensystem is used. A classicality theorem for every degree in a prescribed interval is unnecessary. □

Theorem 12.5 (Compatible system completion from one automorphic member)**.** *Let*

$$\{\rho_\lambda\}_{\lambda \in \Lambda}$$

be a semisimple strictly compatible system with common Frobenius spine. Suppose the member at λ_0 admits an ordinary or derived completion datum and is identified with the Galois representation attached to an automorphic representation π. Assume that the compatible system attached to π has the same Frobenius spine. Then

$$\boxed{\rho_\lambda \cong \rho^{\mathrm{ss}}_{\pi,\lambda} \quad \text{for every } \lambda \in \Lambda.}$$

Proof. The completion theorem at λ_0 gives the automorphic representation π. Apply theorem 10.7 to the two compatible systems. □

Theorem 12.6 (Five gate formulation)**.** *Fix a characteristic zero deformation point $x : R \to E'$. Give E' a Hausdorff topological field structure. The conditional argument uses the following five gates.*

(1) ***Residual gate.*** *Residual occupancy in the declared automorphic carrier and representability of the chosen deformation problem are supplied arithmetic inputs. In particular, the deformation ring R and its universal representation ρ_R are available.*

(2) ***Patching gate.*** *An applicable patching theorem supplies an ordinary finite module or a perfect complex with full patched support, together with the coefficient quotient and the module or derived control isomorphism. Write M for the resulting finite level carrier over R.*

(3) ***Comparison gate.*** *Let H be a finite ordinary Hecke module or a perfect Hecke complex over $\mathbb{T}$. Supply a ring map $f : R \to \mathbb{T}$ satisfying*

$$(f^*)^{-1}\big(\mathrm{Supp}_R(M)\big) \subseteq \mathrm{Supp}_{\mathbb{T}}(H).$$

Use ordinary or derived support according to the carrier. Actual scalar extension satisfies this inclusion, as does transport through a ring isomorphism with the corresponding carrier identification.

(4) ***Fiber gate.*** *Supply a Hecke point $y : \mathbb{T} \to E'$ with $y \circ f = x$. Then the fiber of H at y is nonzero. It contains a nonzero Hecke eigenvector, in some integer cohomological degree in the derived case, with character y.*

(5) ***Realization gate.*** *The declared classicality theorem applies to the resulting eigensystem and supplies an occurring classical automorphic representation π, with a continuous semisimple attached representation. Require the specialization $\rho_x = x \circ \rho_R$ to be continuous and supply its semisimplification. On the same Chebotarev dense set, assume that the characteristic polynomials of both ρ_x and the attached representation are $y(P_v(X))$. Retain the standard density and general Brauer Nesbitt inputs of standard fact 9.2.*

Then ρ_x^{ss} is isomorphic to the representation attached to π. If ρ_x is irreducible, the same conclusion holds for ρ_x itself.

For propagation, suppose both a semisimple compatible family $\{\rho_\lambda\}$ containing ρ_x and a family $\{\rho_{\pi,\lambda}\}$ attached to this same π already exist. Require the distinguished members to be ρ_x and its attached representation, respectively. Assume the coefficient, continuity, density, semisimplification and common algebraic Frobenius spine hypotheses of theorem 10.7, with that spine agreeing with the specialized Hecke polynomials at the distinguished place. Then

$$\rho_\lambda \cong \rho_{\pi,\lambda}^{\mathrm{ss}} \qquad \text{for every } \lambda.$$

Proof. The residual and representability inputs specify the setting in which the supplied patching theorem applies. The formal deduction begins with its actual rings, carriers and comparison maps. In the ordinary branch, finite generation and full support at finite level follow from the coefficient quotient and module control isomorphism. In the derived branch, the derived control isomorphism gives a perfect finite level model, and theorem 6.18 gives full support.

The comparison inclusion transports this full support to H. More specifically, $y \circ f = x$ identifies the contraction of $\ker y$ with $\ker x$, so support at the source point gives support at the Hecke point. In the ordinary case, theorem 7.4 gives a nonzero actual tensor fiber. In the derived case, the support criterion gives a nonzero actual derived fiber and hence a nonzero cohomology vector. Its integer degree lies within the bounds of the chosen finite projective model. In both cases, multiplication on the original carrier induces the scalar $y(t)$ on the specialized vector for every $t \in \mathbb{T}$.

Choose this nonzero eigensystem and apply the declared realization theorem to it. No realization statement in other degrees is needed. The two continuous representations have the same characteristic polynomials on the declared dense Frobenius conjugacy classes. Continuity extends their equality to the whole group. The supplied semisimplification preserves those polynomials, and the general Brauer Nesbitt input gives the required invertible linear intertwiner. The irreducible conclusion follows as in theorem 9.4.

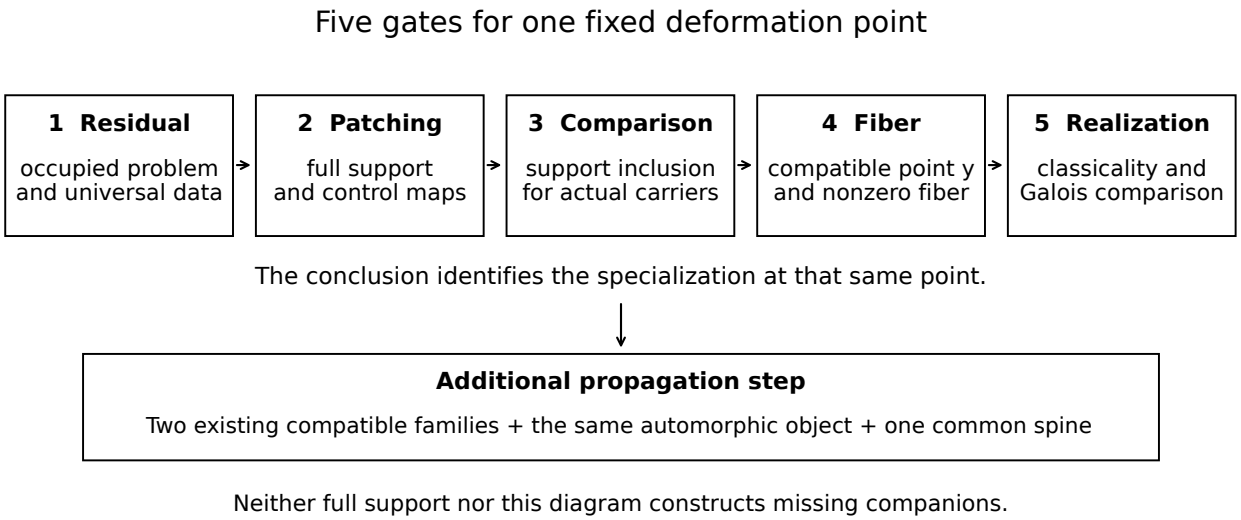


Figure 18. The final theorem is a composition of separate residual, support, comparison, fiber, and realization gates.

Table 1. Proved formal statements and arithmetic inputs that remain external.

Proved in this chapter	Not proved in this chapter
Local conditions combine through completed quotient rings	Representability of every desired local deformation condition
Full ordinary and derived support descends through the patching quotient	Near faithfulness or full support for every global deformation problem
An isomorphism $R \cong \mathbb{T}$ transports support and Fitting data	An $R \cong \mathbb{T}$ theorem outside published hypotheses
Finite Hecke algebras split into Artin packets	Positive automorphic occupancy of every residual or characteristic zero packet
Nonzero stable cores produce compatible inverse limit points	Universal residual companion existence
One automorphic member determines a fixed compatible system	Existence of a compatible system for every geometric representation
Augmentation and coinvariant recovery work at coefficient prime two	A universal coefficient two replacement for residual image, local geometry, and classicality hypotheses

Finally, retain the same π in its attached family. The distinguished member identities connect both families to the pointwise construction, while the common algebraic spine gives the required Frobenius polynomials at every coefficient place. Apply theorem 10.7 to these two existing families. The construction does not produce a missing companion or an attached compatible family. □

13 Exact boundary of the chapter

The formal results above sharpen the logical role of patching support, Hecke packets, and compatible systems. They do not replace the arithmetic theorems that supply the gates.

Remark 13.1. The strongest global conclusion is conditional automorphic realization inside a declared class for which the five gates have been verified. The chapter does not prove unrestricted global Langlands reciprocity.

Remark 13.2. A numerical certificate can check the finite algebra appearing in an example. It cannot verify a published automorphy lifting theorem, a classicality theorem, or a universal existence statement.

Formal verification scope

The formal records report 748 named theorem
declarations written for Paper XII, including supporting lemmas and interface results. A further 81 theorem declarations are reused unchanged from Paper X and counted separately. The cumulative verification rebuilds all 96 registered project modules with Lean 4.34.0-rc2 and warnings treated as errors. The transitive axiom audit covers 1,229 named declarations and permits only `propext`, `Classical.choice`, and `Quot.sound`. Three deliberately invalid controls test rejection of a false proof, a placeholder, and an incorrect empty family normalization. The pinned Mathlib dependency cache is not itself rebuilt. Exact compiler results, source hashes, and the verification scope are recorded with the archive.

The formal sources map all 49 amended internal numbered
claims. The mathematical text retains those proof bodies and exposes
the hypotheses on the local map and coefficient target in proposition 3.3, makes the $r > n$ convention explicit in the proof of theorem 6.4, states all the pointwise comparison inputs in definition 12.1, and declares a finite ramification set and coherent geometric Frobenius normalization. These editorial clarifications are recorded separately from the inherited eight amendments. They are not counted as newly compiled Lean certificates. No new statement amendment was needed for the completion and tensor results. The eight inherited amendments remain explicitly separate from the original wording, and the cited arithmetic inputs remain outside the internal claim census. Formal proof coverage is recorded claim by claim in the roadmap. These checks certify the formal statements under their explicit hypotheses, not every prose claim, novelty assertion, or external arithmetic result.

The completion block proves theorem 3.1 and proposition 3.3 using actual algebraic tensor products, actual Hausdorff completions, and the actual closed generated ideals. Both inverse laws, continuity in both directions, and the canonical factor maps are checked. An independent comparison with the algebraic inverse limit completion identifies the topology and local ring structure. The global theorem retains continuous local algebra points and their naturality, rather than only an unstructured point existence statement. The empty family retains the scalar adic topology. The zero condition and unit ideal boundaries are checked separately.

The preceding Fitting block proves theorems 6.3, 6.4 and 7.1 and proposition 6.5. Its intrinsic Fitting ideals are independent of the chosen finite presentation, including the number of generators and relations. Arbitrary base change, all higher fiber dimension strata, and the complete support and Fitting transport statement are checked. The Noetherian and finite module hypotheses of the manuscript provide the finite presentation needed by the general construction.

The new checks for proposition 5.1, theorem 5.4, and corollary 11.2 include actual group algebra transfer and augmentation. Coinvariant recovery uses the quotient by all balancing relations, with its universal property proved. In the commutative case, this construction is identified with the ordinary tensor product. The augmentation relations generate exactly the augmentation ideal acting on the module. Applying augmentation to the coordinates of a supplied free basis gives a coefficient basis

with the same index set. The right module argument applies to arbitrary groups, including nonabelian two groups, and includes rank zero.

The new checks for theorem 5.2, corollaries 5.3 and 8.4, and proposition 11.1 construct the full multivariable truncated polynomial algebra isomorphism and identify its augmentation generators. The residual group algebra is local for abelian p groups in characteristic p. For each supplied Hecke idempotent, the trace is the image dimension cast into the coefficient field. For that idempotent, occupancy is equivalent to positive rank and, in characteristic zero, nonzero trace. The dyadic calculation gives the two rational sign projectors, proves that neither lies in the integral group algebra, and identifies the residual algebra with $\mathbb{F}_2[X]/(X^2)$.

Corollary 5.3 is now checked for every finite p group, including nonabelian groups, over every field of characteristic p. The proof uses a finite prime field orbit span and a nonzero fixed vector. The original unrestricted statement is restored, and the earlier explicit abelian restriction is no longer needed.

The checks for proposition 7.3, theorem 8.2, and corollary 8.3 now include the actual formal power series quotient and the canonical Hecke packet maps. In the node $k[[x,y]]/(xy)$, quotienting by (x) gives $k[[y]]$, with support $V(x)$. Lean verifies both irreducible components and the exclusion of the generic point (y). The components can still meet, so this is not a claim that their supports are disjoint.

For the Hecke results, the canonical ring decomposition in standard fact 8.1 remains a declared external input. Its coordinates are the actual localization maps. The module direct sum, the packet identification with the localized module, and the formula $v \mapsto v/1$ are constructed in Lean. Nilpotence in the selected Artin local factor, together with invertibility of a generator on each other factor, gives a single positive exponent for the simultaneous generalized eigenspace. The result includes the explicit generators $T_i - \lambda_i$, agrees with the packet used in the trace theorem, and does not assume semisimplicity.

The revised five gate statement in theorem 12.6 is now checked for both ordinary and derived carriers and for a general support comparison. The comparison is an actual inclusion of subsets of prime spectra, and scalar extension is proved to satisfy it. The coefficient quotient and ordinary or derived control maps yield finite level support. That conclusion is not supplied as a separate patching assumption. The specialized tensor module or derived cohomology supplies an actual nonzero eigenvector. The declared realization theorem is applied to this particular eigensystem, and the resulting representation comparison is an actual invertible linear intertwiner. The system conclusion uses the family attached to the very same automorphic object, with distinguished member identities and a common algebraic polynomial spine.

The original comparison wording did not explicitly supply a compatible Hecke point over the fixed coefficient field. Support preservation alone is insufficient: extension from $\mathbb{Q}$ to $\mathbb{Q}(i)$ preserves full support, but there is no ring map $\mathbb{Q}(i) \to \mathbb{Q}$ extending the identity. The statement imposes $y \circ f = x$ and exposes the arithmetic comparison and existing-family inputs. Residual occupancy, representability and the applicable patching theorem remain supplied arithmetic provenance. The original full statement remains partial. The completed tensor quotients and intrinsic Fitting ideals are now checked as described above, including the full Fitting transport clause.

The derived realization checks cover the revised theorems 9.6 and 12.4. They prove vanishing outside the bounds of an actual finite projective model and construct a nonzero cohomology vector inside those bounds. The Hecke action is induced from multiplication on the original cochain model, carried through scalar extension and derived localization, and proved to act by the specialized character on actual cohomology. The conditional arithmetic step gives an actual invertible linear intertwiner from the declared realization input and common Frobenius polynomials.

The original formulation of theorem 9.6 did not ensure that the surviving degree lies in the classicality range. A complex concentrated in degree one has full support over a field, while a classicality condition restricted to degree zero says nothing about its nonzero fiber. The statement needs the

declared range to contain the model interval. The conditional conclusion of theorem 12.4 retains applicability at the selected eigensystem itself. Both statements now expose the topology, continuity, semisimplification, and common polynomial inputs needed for the representation comparison. Their original unconditional cohomological conclusions are checked, but their original complete statements remain partial. The general support comparison and compatible-point formulation is now checked in the revised theorem 12.6, as described above.

The derived support checks prove theorems 6.16 and 6.18 and corollary 6.19 as originally stated. Perfect complexes are actual derived category objects equipped with bounded finite projective models and derived isomorphisms to those models. Scalar extension is constructed on cochain complexes. Its associativity, preservation of homotopy equivalences, and independence under actual derived isomorphisms are checked. The quotient support formula uses the actual map of spectra and its image $V(I)$. Every field valued point factors through the residue field at its kernel, and faithful flatness of field extensions preserves and reflects acyclicity. The surviving integer degree is expressed using the actual derived cohomology functor, with a corresponding witness in the specialized cochain model. These proofs do not require the external cohomological support description in standard fact 6.15. The external classicality theorems remain supplied arithmetic inputs. The conditional derived realization argument is checked as described above.

The adequacy checks prove theorem 4.2 as originally stated. They construct the faithful standard representation from the actual permutation module, prove irreducibility after extension to the algebraic closure, and split every group element's characteristic polynomial over $\mathbb{F}_7$. The trace zero adjoint is the kernel of matrix trace with its conjugation action, and its sign plus standard decomposition is an actual equivariant linear isomorphism. The averaging proof is connected to Mathlib's actual first group cohomology, while the invariant calculation gives actual degree zero vanishing. Canonical projectors are identified with generalized eigenspace projectors for every positive exponent. Their trace pairings detect every nonzero endomorphism and hence every irreducible adjoint submodule. The two printed constituent witnesses and their traces are checked exactly. The adequacy definition remains a cited convention. No external adequacy theorem is assumed in this finite certificate.

The realization checks cover theorems 9.4, 10.7, 12.2 and 12.5 and corollary 9.5. Characteristic polynomial coefficients are continuous by the principal minor formula. Equality on the declared Frobenius conjugacy classes therefore extends to the whole Galois group. The finite dimensional algebra input is explicit: a supplied semisimple representative preserves every characteristic polynomial, and a general Brauer Nesbitt theorem identifies semisimple representations with equal polynomials. Lean proves independence of this representative and the irreducible conclusion. Construction of semisimplification, the general Brauer Nesbitt theorem, and arithmetic Chebotarev density remain external inputs.

The ordinary completion argument derives finite generation, full finite level support, and an actual nonzero Hecke tensor fiber from the patched support and control isomorphisms. The declared classicality input then supplies the occurring automorphic object and its attached representation. Both existing compatible systems, with their common polynomial spine, are retained as hypotheses in the propagation results. These results do not construct missing companions or an attached compatible system.

The pointwise realization and ordinary completion statements now explicitly require Hausdorff coefficient topology and continuity of the compared representations, including the universal specialization. The corresponding locus statement is restricted to such points. These three revised claims are counted as complete. Their original wording remains partial. This clarification also makes the coefficient topology in standard fact 9.2 explicit without claiming a new proof of that external arithmetic fact.

Two statement qualifications are essential. In corollary 6.12, every irreducible component is contained in the full support. The previous sentence reversed that containment. In theorem 10.5, the coefficient, continuity, integrality, class finiteness, and quotient condition hypotheses are now explicit. Finite forests by themselves establish branches, not all arithmetic effectivity assertions. Corollary 3.2 now gives the closed kernel argument and the continuity and uniqueness of its factors. The original wording and its claim records accompany the sources. In that wording census, 41 claims are fully checked, seven are partial, and one is not checked as written. The corrected corollary 6.12 and restricted theorem 10.5 are counted as proved only for their revised statements. The original unrestricted completion wording is not retrospectively marked proved. Source hashes, the exact amendments, compiler logs, and a mathematical self-review accompany the code. Compiler and axiom checks are distinguished from semantic proof review and external arithmetic results.

Plate XII.1 / A STANDALONE READER'S COMPANION

Seeing the objects

Patching Support, Hecke Packets, and Conditional Automorphic Completion Main statement 12.6

Komorebi P.

The chapter repeatedly asks three different questions: does an object exist, where is a supplied object nonzero, and when does that object have the desired arithmetic meaning? This guide makes those distinctions tangible through examples that can be followed by hand.

Existence	Is there a represented deformation problem, a nonzero carrier, or a compatible family to start with?
Support	At which prime ideals does the supplied module or complex survive?
Realization	Does the surviving eigensystem come from a classical automorphic object with the required attached representation?

Three models, rather than one overloaded analogy

Model	What it lets you see	Lessons
A formal thickening	Conditions on power series, infinitesimal points, obstructed lifts, and compatible precision levels.	Plates XII.3–XII.5
A presented module	Matrix rank, support, Fitting ideals, changes of coordinates, and nonreduced structure.	Plates XII.6–XII.8
A finite Artin algebra	Generalized eigenspaces, nilpotent directions, canonical packets, and empty packet modules.	Plates XII.12–XII.14

The models are connected by the chapter's theorems, but they are not the same ring. A polynomial coordinate plane is not a complete local spectrum, and a local branch is not an Artin product factor. Explicitly changing models prevents a picture from silently changing the hypotheses.

A route through the guide

Use the object atlas on Plate XII.2 when a symbol becomes unfamiliar. The local model starts on Plate XII.3, and the matrix model starts on Plate XII.6. Plates XII.9–XII.11 explain what quotients, patching, and derived fibers preserve. Plates XII.15–XII.16 separate finite occupancy from genuine infinite compatibility. The exact arithmetic boundary is on Plate XII.17.

How to read a visual claim

Each lesson gives an explicit object, a calculation, and a source reference. Coordinate sketches display only the stated real points. Coefficient arrays and finite specialization diagrams describe the stated algebraic data exactly. Line thickness never stands for nilpotent multiplicity.

Read this as: These examples explain the mathematical constructions. They add no assumptions or Lean certificates. The accompanying scripts check the explicit finite calculations. They do not establish universal arithmetic existence.

Plate XII.2 / ORIENTATION

An atlas of the objects

Main text: Sections 2, 3, 6, 7, 8 and 12 Main statement 3.3

A *ring* describes possible points. A *module over the ring* describes what is carried at those points. A nonempty space of possibilities does not imply a nonzero carrier at every point.

Symbol	Object and role	Keep separate
$\mathcal{O},\ k$	Complete coefficient ring and its residue field. The formal teaching model uses $\mathcal{O} = \mathbb{Z}_5, k = \mathbb{F}_5$.	Coefficients versus variables.
$R_i,\ I_i$	A supplied local ring and the ideal imposing a declared local condition.	A quotient condition is not an idempotent.
R^{loc}	Completed tensor product of the local rings over $\mathcal{O}$.	Completion uses the declared adic topology.
J	Closed ideal generated by the images of all I_i in R^{loc}.	Closure, not just a finite list of tests.
$R^{\mathrm{loc},\tau}$	Completed tensor of the local quotients, identified with R^{loc}/J.	A constructed comparison, not assumed notation.
$R^{\square}$	Supplied global ring receiving the local structural maps.	Representability is arithmetic input.
R^{τ}	Completed global base change imposing the simultaneous local conditions.	Does not create a missing residual point.
$R_\infty,\ M_\infty$	Patched ring and its module. A perfect complex is used in the derived version.	Freeness over an auxiliary ring is weaker than full support.
$R,\ M$	Finite-level ring and carrier after the specified quotient and control isomorphism.	The module may be zero on a component.
$T,\ H$	Hecke algebra and its ordinary or derived carrier.	Ring comparison does not imply classicality.
$e_{\mathfrak{m}}H$	Packet selected by a canonical Artin idempotent.	A generalized packet need not be an eigenspace.
Y_m	Stable image at level m of an inverse system.	Nonzero levels alone do not give a nonzero limit.

Why arrows reverse when points appear

A ring map $f : R \to T$ sends a prime of T to its inverse image in R:

$$f^* : \operatorname{Spec} T \longrightarrow \operatorname{Spec} R, \qquad \mathfrak{q} \longmapsto f^{-1}(\mathfrak{q}).$$

Similarly, a point $y : T \to E'$ gives the point $y \circ f : R \to E'$. The equality $y \circ f = x$ is an actual requirement in the final theorem, not a consequence of drawing a connecting arrow.

Two kinds of point

Local quotient factorization is stated for continuous local coefficient-algebra maps to deformation test rings. The later field-valued point $x : R \to E'$ uses its kernel and scalar fiber. A characteristic-zero field point is generally not a local ring map: the coefficient uniformizer becomes a unit. These two uses of “point” must retain their own categories.

Plate XII.3 / MODEL A / A FORMAL THICKENING

Local conditions remove directions

Main text: Theorem 3.1 and Corollary 3.2 Main statement 3.1

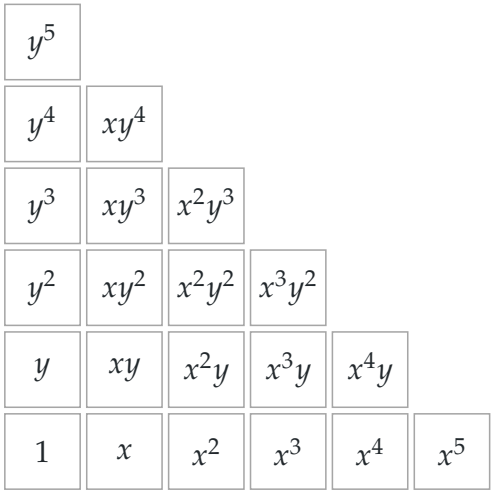
Fix $\mathcal{O} = \mathbb{Z}_5$. Take two independent local variables and impose two different kinds of condition:

$$R_1 = \mathcal{O}[[x]], \quad I_1 = (x^2), \qquad R_2 = \mathcal{O}[[y]], \quad I_2 = (y).$$

Their completed local tensor is $A = \mathcal{O}[[x, y]]$. The closed generated ideal is $J = (x^2, y)$, and both routes through Theorem 3.1 give

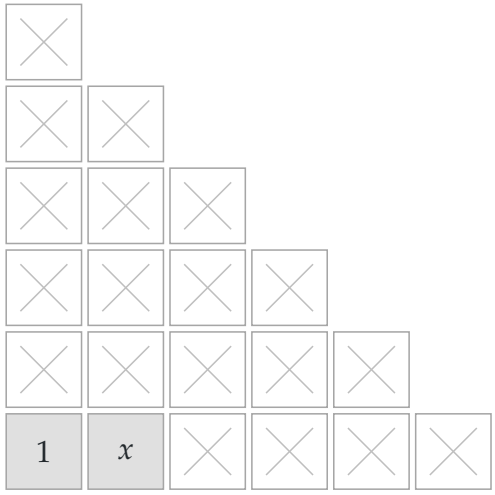
$$(R_1/I_1) \mathbin{\widehat{\otimes}}_{\mathcal{O}} (R_2/I_2) \cong A/J \cong \mathcal{O}[\varepsilon]/(\varepsilon^2).$$

Before imposing the ideals



$i + j < 6$ shown; coefficients lie in $\mathcal{O}$.

After imposing (x^2, y)



$i + j < 6$ shown; coefficients lie in $\mathcal{O}$.

An exact finite window of monomials over $\mathcal{O}$. Crosses mean that the monomial is zero in the quotient. They do not represent a small numerical coefficient. All terms outside the window satisfy the same divisibility rule.

The calculation behind the image

For $f = \sum_{i,j \geq 0} a_{ij} x^i y^j$, reduction gives

$$f \longmapsto a_{00} + a_{10}\varepsilon.$$

Every term involving y vanishes. Every term involving x^2 vanishes. The class of x does *not* vanish: it becomes a nonzero element whose square is zero. The quotient is a free $\mathcal{O}$-module with basis $1, \varepsilon$.

Why a point picture alone would miss this

After passing to the special fiber, $k[\varepsilon]/(\varepsilon^2)$ has the same underlying one-point spectrum as k. The extra direction is visible in the algebra and in maps from test rings, not as a second point. It is also not a second component or a second Artin packet.

Read this as: The condition $y = 0$ deletes a variable. The condition $x^2 = 0$ retains a first-order nilpotent direction. Both are quotient conditions, but their effects are different.

Plate XII.4 / MODEL A / FACTORIZATION AND OBSTRUCTION

A point is a test of the equations

Main text: Proposition 3.3; this explicit example also exhibits a failed infinitesimal lift Main statement 3.3

Let $R^{\square} = \mathcal{O}[[u, v]]$ and send $x \mapsto u$, $y \mapsto v - u$. The local conditions become

$$u^2 = 0, \qquad v - u = 0, \quad \text{so} \quad R^{\tau} = \mathcal{O}[[u, v]]/(u^2, v - u) \cong \mathcal{O}[\varepsilon]/(\varepsilon^2).$$

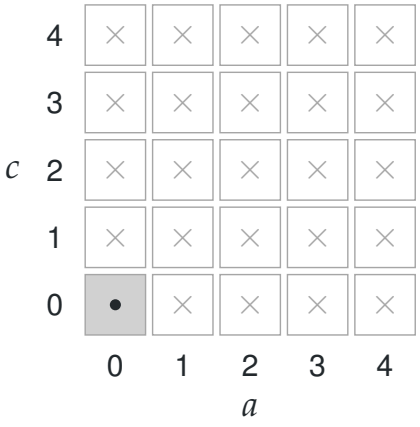
Probe this ring using $D_3 = \mathbb{F}_5[\delta]/(\delta^3)$, with the fixed coefficient map $\mathbb{Z}_5 \to \mathbb{F}_5$. A continuous local point is specified by

$$u \longmapsto a\delta + b\delta^2, \qquad v \longmapsto c\delta + d\delta^2, \qquad a, b, c, d \in \mathbb{F}_5.$$

It factors through R^{τ} precisely when

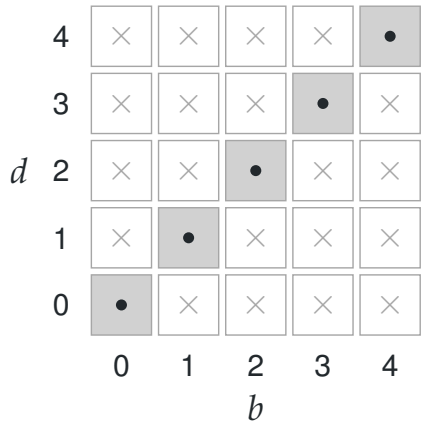
$$a^2 = 0, \quad c = a, \quad d = b \quad \Longleftrightarrow \quad a = c = 0,\ d = b.$$

First-order coefficients



Only the slope pair $(a, c) = (0, 0)$ survives. Each square initially has 25 choices of (b, d).

Remaining second-order coefficients



Once $a = c = 0$, the five diagonal choices $d = b$ give all accepted points.

An obstruction you can calculate

There are $5^4 = 625$ candidate D_3-points and exactly five accepted points. Over $D_2 = \mathbb{F}_5[\delta]/(\delta^2)$, however, the five maps $u = v = a\delta$ all satisfy the equations. Reduction of *every* accepted D_3-point gives the zero D_2-point. The four nonzero tangent directions therefore do not lift to order three.

> **Read this as:** A point satisfies equations in a target ring. A tangent direction can exist while its next-order lift fails. The factorization theorem identifies the condition. It does not promise that every point lifts.

Plate XII.5 / MODEL A / COMPATIBLE COEFFICIENTS

Completion retains every precision

Main text: Theorem 3.1; reconstruction and finite occupancy in Section 10 Main statement 10.5

For $B = \mathbb{Z}_5[\varepsilon]/(\varepsilon^2)$, consider the finite quotient rings

$$B_n = (\mathbb{Z}/5^n\mathbb{Z})[\varepsilon]/(\varepsilon^2), \qquad |B_n| = 5^{2n}, \qquad B \cong \varprojlim_n B_n.$$

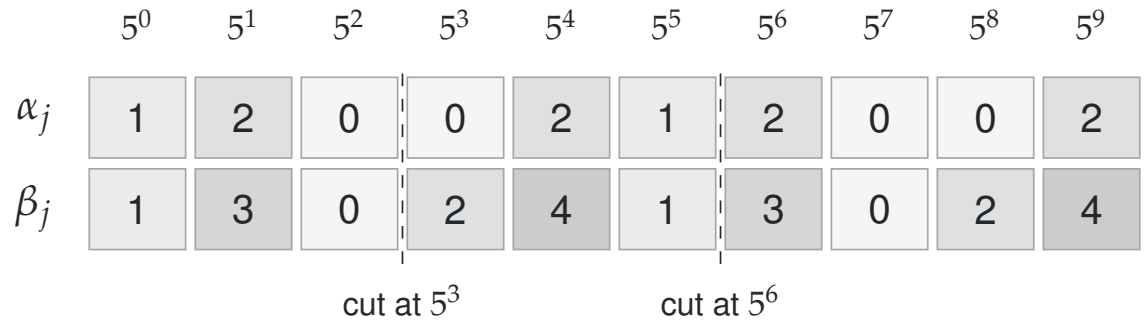
A point of the limit is a *single compatible choice* at every level, not unrelated choices from the nonempty finite rings.

Two explicit 5-adic numbers

Choose the canonical digits

$$\alpha_j = (j^2 + 1) \bmod 5, \qquad \beta_j = (2j + 1) \bmod 5,$$

with values in $\{0, 1, 2, 3, 4\}$. Set $a = \sum_{j\geq 0} \alpha_j 5^j$ and $b = \sum_{j\geq 0} \beta_j 5^j$. The element $a + \varepsilon b$ is determined by the following coefficient array.



Cutting before column 5^n produces reduction modulo 5^n. The digit formulas define every column, including those not drawn.

n	1	2	3	4	5
5^n	5	25	125	625	3125
$a \bmod 5^n$	1	11	11	11	1261
$b \bmod 5^n$	1	16	16	266	2766

Every longer prefix reduces to the previous one. The digit formulas specify all later coefficients.

Where the topology enters

The 5-adic and $(5, \varepsilon)$-adic topologies on B agree: for every n,

$$(5, \varepsilon)^{n+1} \subseteq 5^n B \subseteq (5, \varepsilon)^n.$$

For the general quotient theorem, the condition ideal must be closed. A continuous map to a Hausdorff target has closed kernel: vanishing on the original generators therefore kills their closure.

Plate XII.6 / MODEL B / FITTING STRATA

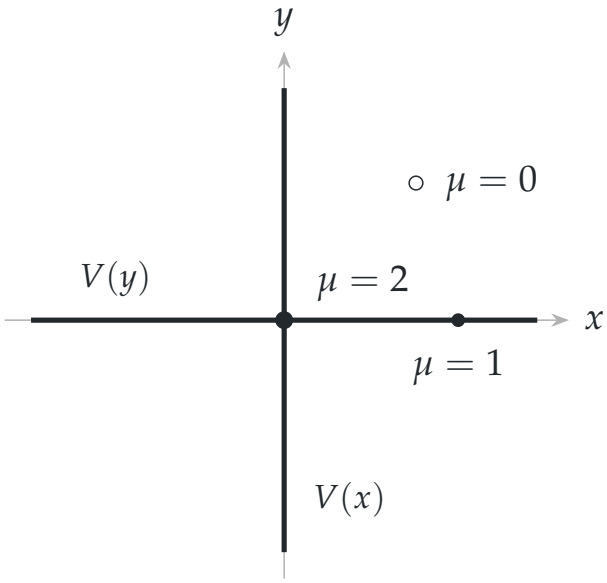
A matrix becomes a support locus

Main text: Theorems 6.2, 6.3 and 6.4 Main statement 6.3

Now change models. Let $R = k[x, y]$ for a field k, and present a module by

$$R^2 \xrightarrow{A} R^2 \longrightarrow M \longrightarrow 0, \qquad A = \begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix}.$$

The relations are $xe_1 = 0$ and $ye_2 = 0$, so $M \cong R/(x) \oplus R/(y)$.



The minor calculation

$$\begin{aligned} \mathrm{Fitt}_0(M) &= (xy), \\ \mathrm{Fitt}_1(M) &= (x, y), \\ \mathrm{Fitt}_j(M) &= R \quad (j \geq 2). \end{aligned}$$

At a prime $\mathfrak{p}$, reduce the matrix to its residue field:

$$\mu_{\mathfrak{p}}(M) = 2 - \operatorname{rank} A(\mathfrak{p}).$$

Two independent relations disappear at the origin. Only one disappears along either punctured axis.

For the drawing, take $k = \mathbb{R}$. It shows real closed points, not every point of $\operatorname{Spec} R$. The displayed ideals and rank criterion describe the full algebraic loci over any field.

Position	Matrix rank	Fiber dimension	What survives
$x \neq 0, y \neq 0$	2	0	neither generator
$x = 0, y \neq 0$	1	1	the e_1 direction
$x \neq 0, y = 0$	1	1	the e_2 direction
$x = y = 0$	0	2	both directions

Thus $\operatorname{Supp}(M) = V(xy) = V(x) \cup V(y)$, whereas the locus where the fiber needs at least two generators is $V(x, y)$. The latter is a *higher* Fitting stratum. For $r > 2$, the locus $\mu_{\mathfrak{p}}(M) \geq r$ is empty, corresponding to the unit Fitting ideal.

> **Read this as:** Support asks whether any direction survives. Higher Fitting strata ask how many generators the residue-field fiber needs. This is not an automorphic multiplicity formula.

Plate XII.7 / MODEL B / NILPOTENT STRUCTURE

The same support can hide more algebra

Main text: Theorems 6.3–6.4; distinguish an ideal from its vanishing set Main statement 6.4

Compare the previous module with a slightly different presentation:

$$M = \operatorname{coker}\begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix}, \qquad M' = \operatorname{coker}\begin{pmatrix} x^2 & 0 \\ 0 & y \end{pmatrix}.$$

Their support sets coincide. Their residue-field fiber dimensions also coincide at every prime. But their Fitting ideals do not.

	M	M'
Zeroth Fitting ideal	(xy)	(x^2y)
First Fitting ideal	(x, y)	(x^2, y)
Support as a set	$V(x) \cup V(y)$	$V(x) \cup V(y)$
Fiber dimension at (x, y)	2	2

Magnify transversely, rather than draw a thicker line

Take the slice $y = 1$. The second summand vanishes because y becomes a unit. What remains is

$$M|_{y=1} \cong k[x]/(x), \qquad M'|_{y=1} \cong k[x]/(x^2).$$

The first has basis 1. The second has basis $1, x$. Multiplication by x distinguishes them exactly:

reduced slice

$$1 \xrightarrow{\times x} 0$$

$$[x] = [0]$$

nilpotent slice

$$1 \xrightarrow{\times x} x \xrightarrow{\times x} 0$$

$$[x] = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

A test ring detects the thickening of the higher stratum

Let $D = k[\delta]/(\delta^2)$ and send $x \mapsto \delta$, $y \mapsto 0$. The ideal (x^2, y) maps to zero, but (x, y) does not. Therefore this D-point lies in the scheme defined by the first Fitting ideal of M', but not in the corresponding scheme for M.

No extra ordinary point has appeared. What changed is the behavior on nilpotent test rings. This is the same distinction encountered in the formal model on Plate XII.3.

> **Read this as:** Taking $V(I)$ forgets nilpotent thickness. Fitting *ideals* can retain information that both the support set and the function giving fiber dimensions fail to distinguish.

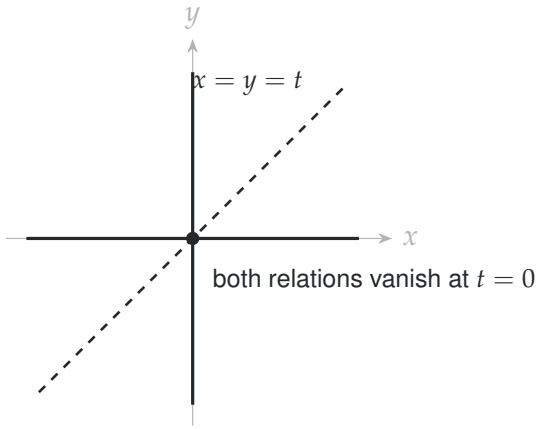
Plate XII.8 / MODEL B / SPECIALIZATION AND TRANSPORT

Base change is an algebraic pullback

Main text: Proposition 6.5 and Theorem 7.1 Main statement 6.5

Return to $M = \operatorname{coker} \operatorname{diag}(x, y)$ over $R = k[x, y]$. Pull back along the diagonal $R \to k[t]$, $x \mapsto t$, $y \mapsto t$. The base-changed presentation is literal:

$$M \otimes_R k[t] \cong \operatorname{coker} \begin{pmatrix} t & 0 \\ 0 & t \end{pmatrix} \cong \big(k[t]/(t)\big)^2.$$



Both relations vanish together

$$(xy) \longmapsto (t^2), \qquad (x, y) \longmapsto (t).$$

The support is the origin, but the origin has fiber dimension two. Replacing (t^2) by its radical (t) would lose the zeroth Fitting ideal.

This map is not flat: it is the quotient by $x - y$. The Fitting base-change theorem still applies. It does not require flatness.

A view of the real points on the diagonal slice. The double algebraic contribution is certified by the minor t^2, not by the visual thickness of the intersection point.

An isomorphism changes coordinates, not occupancy

Over a field of characteristic other than two, the isomorphism

$$k[x, y] \xrightarrow{\sim} k[a, b], \qquad x \mapsto a + b, \quad y \mapsto a - b$$

sends the Fitting ideals to $(a^2 - b^2)$ and (a, b). The two axes become the two lines $a = b$ and $a = -b$. No previously missing module vector is created by this relabeling.

An extra presentation generator changes nothing

Replacing $\operatorname{diag}(x, y)$ by $\operatorname{diag}(x, y, 1)$ adds a generator killed by the relation 1. The new maximal minor is xy, and the new next-size minors generate $(xy, x, y) = (x, y)$. The module and its intrinsic Fitting ideals are unchanged. This is a small visible instance of presentation independence of Fitting ideals.

> **Read this as:** Specialization substitutes into the actual relation matrix. An isomorphism transports all the data. Neither operation may be replaced by an informal edit of the support picture.

Plate XII.9 / LOCAL GEOMETRY / AN EXACT FINITE SPECTRUM

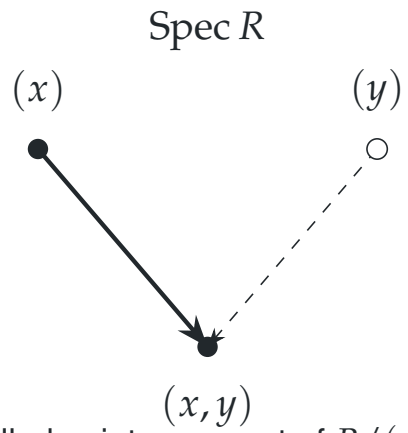
A branch is not an Artin packet

Main text: Proposition 3.4 and Proposition 7.3 Main statement 7.3

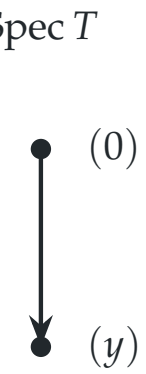
Consider the complete local node

$$R = k[[x,y]]/(xy), \qquad T = R/(x) \cong k[[y]].$$

Every prime of R contains x or y. Since $k[[x]]$ and $k[[y]]$ are discrete valuation rings, the following diagram lists *all three* prime ideals of R and their specialization relations.



Filled points: support of $R/(x)$.
Arrows: specialization of prime ideals.



The quotient retains one generic point and the common closed point.

What the quotient removes

As an R-module, T has annihilator (x), hence

$$\mathrm{Supp}_R(T) = V(x) = \{(x), (x,y)\}.$$

The other generic point (y) is missing because it does not contain x. The quotient has not created a disconnected projector summand inside the original local ring. It has changed the ring by imposing an equation.

Why no idempotent selects just this branch

In a commutative local ring, $e^2 = e$ implies $e = 0$ or $e = 1$. Indeed, $e + (1-e) = 1$ forces at least one of $e, 1-e$ to be a unit. Canceling in $e(1-e) = 0$ gives the conclusion. Thus this node has no nontrivial idempotent even though it has two irreducible components.

By contrast, a product of two Artin local rings has two disjoint maximal points, with canonical idempotents $(1,0)$ and $(0,1)$. Those idempotents act on *any supplied module over the product*. Their two ranges are the packets studied on Plate XII.12.

> **Read this as:** Intersecting local components are not product factors. Use quotient ideals and support for the node. Use canonical idempotents only where an actual product decomposition exists.

Plate XII.10 / PATCHING / A COUNTEREXAMPLE WITH A VISIBLE SPECIAL FIBER

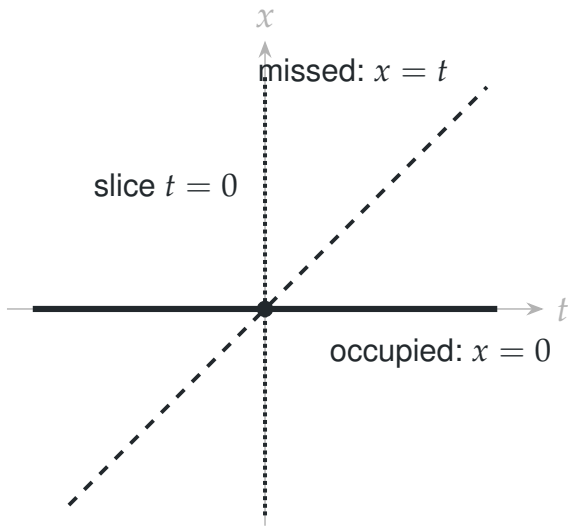
Free over what? Supported where?

Main text: Theorems 6.7, 6.9 and 6.11; Example 6.13 Main statement 6.11

Let

$$S = k[[t]], \qquad R_\infty = S[[x]]/(x(x-t)), \qquad M_\infty = R_\infty/(x) \cong S.$$

The module is free of rank one over S. Yet over R_∞ its support is only $V(x)$. The generic point $(x-t)$ of the other branch is not occupied.



Remove the auxiliary variable

Set $t = 0$. Then

$$R = R_\infty/(t) \cong k[[x]]/(x^2),$$

$$M = M_\infty/tM_\infty \cong k.$$

The two branches meet in a nonreduced special fiber. The relation becomes $x^2 = 0$, not $x = 0$.

The drawing is a real algebraic shadow of $x(x-t) = 0$. It is not the complete local spectrum. The special fiber's nilpotent structure is recorded by its quotient algebra, not by the point drawn at the intersection.

Full support need not mean faithful action

The ring R has one prime, (x). The nonzero module $M = k$ therefore has full support. But x annihilates M, so the action is not faithful. It is *nearly faithful*: the annihilator (x) is nilpotent. This distinction disappears over a reduced ring.

The actual descent theorem is one-way

For a finite module over a Noetherian ring,

$$\operatorname{Supp}_{R_\infty/(t)}(M_\infty/tM_\infty) = \operatorname{Supp}_{R_\infty}(M_\infty) \cap V(t).$$

If the initial support is full, the resulting support is full on the quotient. Our example shows that full support at the quotient can occur even when initial support was not full. It also shows why freeness over S alone is not the required patching hypothesis.

> **Read this as:** A statement about freeness must name the base ring. A statement about occupancy must name the spectrum. Switching between them without a theorem loses the missing component.

Plate XII.11 / DERIVED SUPPORT / AN ACTUAL TWO-TERM COMPLEX

A vanishing differential changes the fiber

Main text: Definition 6.14 and Theorem 6.16 Main statement 6.16

Take $R = k[t]$ and the perfect complex

$$C = \left[R \xrightarrow{t} R\right] \qquad \text{in cohomological degrees } 0, 1.$$

Specialization at $t = a$ gives $C_a = [k \xrightarrow{a} k]$. When $a \neq 0$, the differential is invertible and the complex is acyclic. When $a = 0$, the differential is zero and both cohomology groups survive.

point a	-2	-1	0	1	2
$\dim H^0(C_a)$	0	0	1	0	0
$\dim H^1(C_a)$	0	0	1	0	0

the zero differential creates two cohomology groups

A few fibers over $k = \mathbb{Q}$ are displayed. The statement for every $a \neq 0$ follows from invertibility of the scalar a. It is not inferred from the sample points.

Compute before and after specialization

Before specialization,

$$H^0(C) = 0, \qquad H^1(C) = k[t]/(t).$$

After *derived* specialization at zero,

$$C \otimes^{\mathbf{L}}_R k = [k \xrightarrow{0} k], \quad H^0 \cong k, \quad H^1 \cong k.$$

If one retained only $H^1(C)$ and took its ordinary tensor product with k, one would see a single k and miss the extra cohomology in degree zero. The projective complex keeps the information needed for specialization.

The support is still a precise locus

At any prime not containing t, the residue-field differential is invertible. At the prime (t) it is zero. Therefore

$$\operatorname{Supp}_R(C) = V(t).$$

Theorem 6.16 identifies support after derived quotient with the intersection of original support and the quotient locus.

> **Read this as:** Derived nonvanishing means that some cohomology group survives after derived specialization. It neither specifies a classical automorphic class nor eliminates the need for a realization theorem.

Plate XII.12 / MODEL C / A NONSEMISIMPLE HECKE ALGEBRA

A packet contains an entire Jordan chain

Main text: Theorem 8.2, Corollary 8.3 and Example 8.6 Main statement 8.2

Work over $\mathbb{Q}$ and let

$$T = \mathbb{Q}[z]/\big((z-1)^2(z+1)^3\big), \qquad V = T.$$

The Chinese remainder theorem gives

$$T \cong \underbrace{\mathbb{Q}[\varepsilon]/(\varepsilon^2)}_{T_+,\ z=1+\varepsilon} \times \underbrace{\mathbb{Q}[\eta]/(\eta^3)}_{T_-,\ z=-1+\eta}.$$

Multiplication by z is an operator H. In the ordered local bases $1, \varepsilon$ and $1, \eta, \eta^2$, its matrix is

$$H = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{array}\right).$$

packet at $z = 1$

$$1 \xrightarrow{H-1} \varepsilon \xrightarrow{H-1} 0$$

two independent basis vectors

packet at $z = -1$

$$1 \xrightarrow{H+1} \eta \xrightarrow{H+1} \eta^2 \xrightarrow{H+1} 0$$

three independent basis vectors

These arrows are the actual nilpotent actions $H-1$ and $H+1$. Each nonzero displayed basis vector belongs to the indicated packet. Only the last nonzero vector in each chain is an ordinary eigenvector.

The projectors can be written explicitly

Modulo $(z-1)^2(z+1)^3$, put

$$e_+(z) = \frac{(5-3z)(z+1)^3}{16}, \qquad e_-(z) = 1 - e_+(z).$$

Then $e_+ \equiv 1 \pmod{(z-1)^2}$ and $e_+ \equiv 0 \pmod{(z+1)^3}$. Consequently

$$e_+(H) = \operatorname{diag}(1,1,0,0,0), \qquad e_-(H) = \operatorname{diag}(0,0,1,1,1).$$

The packet dimensions are two and three. The ordinary eigenspaces each have dimension one. The generalized eigenspaces have dimensions two and three.

> **Read this as:** Canonical idempotents isolate the full primary pieces, including their nilpotent structure. Semisimplicity is not required, and replacing a packet by an ordinary eigenspace can discard most of it.

Plate XII.13 / MODEL C / OCCUPANCY IS MODULE DATA

A possible packet can still be empty

Main text: Theorem 7.4, Theorem 8.2 and Corollary 8.4 Main statement 7.4

Keep the algebra $T = T_+ \times T_-$ from Plate XII.12, but compare two different T-modules:

$$V = T, \qquad W = T_+,$$

where T acts on W through projection to its first factor. The algebra still has two maximal ideals, but only one of them is in $\mathrm{Supp}_T(W)$. Here T is a supplied algebra. Its action on W is *not* assumed faithful.



Squares refer to the local bases of the regular module. Crossed reference slots are absent from W. They are not additional basis vectors of W.

	regular module V	first-factor module W
$\dim e_+(-)$	2	2
$\dim e_-(-)$	3	0
Support	both maximal points	only the $+$ point
Trace of e_-	3	0

A generalized packet is not its residue-field fiber

At the point $z = 1$, the full local packet e_+V has dimension two, but its scalar fiber is

$$V \otimes_{T,\, z\mapsto 1} \mathbb{Q} \cong T_+/(\varepsilon) \cong \mathbb{Q}.$$

The packet has the basis $1, \varepsilon$. Its scalar fiber kills ε. The ordinary eigenspace is instead the kernel of multiplication by ε, spanned by ε. In this example the last two spaces have the same dimension, but they are obtained by different operations.

For W, the fiber at $z = -1$ is zero. At $z = 1$ it is nonzero. The field-fiber criterion detects exactly the support of the supplied module.

> **Read this as:** A ring factor describes an available packet location. A nonzero localized module certifies occupancy there. A field fiber then specializes that module. It is not the whole generalized packet.

Plate XII.14 / A COEFFICIENT-PRIME BOUNDARY

At two, division is the obstruction

Main text: Section 5, Remark 8.5 and Section 11 Main statement 11.1

Let $C_2 = \langle g : g^2 = 1 \rangle$. Over $\mathbb{Q}_2$, the two familiar sign projectors are

$$e_+ = \frac{1+g}{2}, \qquad e_- = \frac{1-g}{2}, \qquad \mathbb{Q}_2[C_2] \cong \mathbb{Q}_2 \times \mathbb{Q}_2.$$

They do not belong to $\mathbb{Z}_2[C_2]$. Therefore one cannot first use them integrally and then reduce modulo two.

Compute the actual residual algebra

Over $\mathbb{F}_2$, put $\delta = g - 1 = g + 1$. Then $\delta^2 = g^2 - 2g + 1 = 0$ and

$$\mathbb{F}_2[C_2] \cong \mathbb{F}_2[\delta]/(\delta^2).$$

The two sign eigenvalues have merged. The ring is local and retains a nilpotent direction instead of two residual projector pieces.

Element	Square	Idempotent?
0	0	yes
1	1	yes
δ	0	no
$1+\delta$	1	no

In the basis $1, \delta$, multiplication by $g = 1 + \delta$ is

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad G^2 = I \quad \text{over } \mathbb{F}_2.$$

It is not the identity: $G - I$ is nonzero and square-zero.

A trace can vanish while a packet is occupied

The identity projector on $\mathbb{F}_2^2$ has rank two and trace $2 = 0$. Hence nonzero rank and nonzero trace are not equivalent in this characteristic. The localization and rank criteria still detect the nonzero space. The characteristic-zero qualification in Corollary 8.4 is essential.

Recovery without averaging

The augmentation map sends g to 1 and has kernel $(g-1)$. Thus

$$\mathbb{Z}_2[C_2]/(g-1) \cong \mathbb{Z}_2, \qquad \big(\mathbb{Z}_2[C_2]^r\big)/(g-1)\mathbb{Z}_2[C_2]^r \cong \mathbb{Z}_2^r.$$

These quotient identities require no division by two. They explain why augmentation and coinvariants remain usable even when sign-projector formulas do not.

> **Read this as:** Use the algebra available at the coefficient prime. Augmentation recovery is valid here. It still does not prove full deformation support or supply a universal automorphy theorem at two.

Plate XII.15 / INVERSE SYSTEMS / THE STABLE CORE

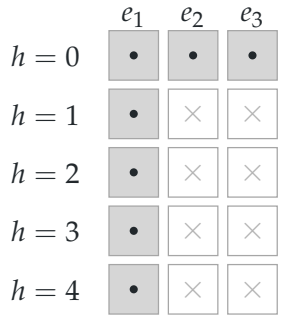
Nonzero at every level is not enough

Main text: Definition 8.9, Lemma 8.10 and Theorem 8.11 Main statement 8.11

Let every level be $V_m = k^3$. For a transition matrix F, compatibility means $v_m = Fv_{m+1}$. The directions visible at level m after looking h levels ahead form $\operatorname{im}(F^h)$.

A surviving direction

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad P^2 = P.$$



At every positive horizon, precisely the e_1 direction remains visible.

Every direction eventually dies

$$N = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad N^3 = 0.$$

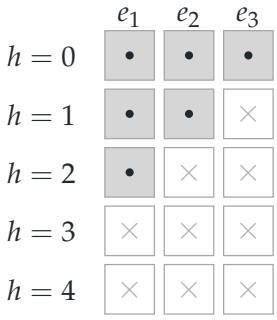


Image dimensions are $3, 2, 1, 0, 0, \ldots$, although every level still has dimension three.

Now compute the limits, not just their level dimensions

For the P-system, the limit consists of the constant compatible sequences $(ae_1, ae_1, \ldots)$, so it is isomorphic to k. For the N-system, every compatible sequence satisfies

$$v_m = N^3 v_{m+3} = 0 \quad \text{for every } m,$$

and the limit is zero. This is an all-level argument from the identity $N^3 = 0$, not a finite experimental guess.

What the stable core measures

In a general inverse system of finite-dimensional spaces,

$$Y_m = \bigcap_{n \geq m} \operatorname{im}(f_{n,m})$$

contains exactly the vectors visible from arbitrarily remote levels. The chapter proves that the transition maps restrict to surjections $Y_{m+1} \twoheadrightarrow Y_m$. A prescribed nonzero vector in Y_1 can then be lifted recursively to a coherent nonzero sequence.

> **Read this as:** The stable core tests persistence through the transition maps. Counting nonzero spaces at separate levels does not test persistence.

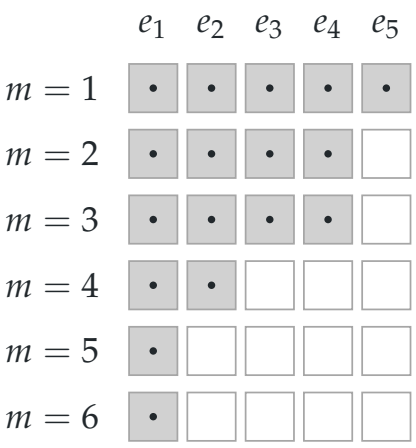
Plate XII.16 / TWO DIFFERENT COMPACTNESS MECHANISMS

Finite dimension prevents tail escape

Main text: Theorem 8.7, Corollary 8.8 and Example 8.13 Main statement 8.7

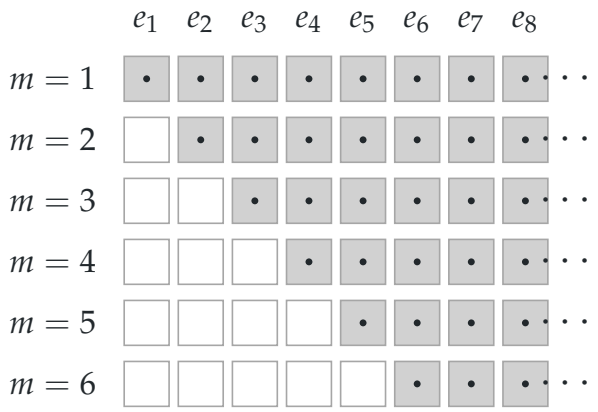
Nested nonzero subspaces inside *one fixed finite-dimensional space* behave differently from nested tails inside an infinite-dimensional space.

Fixed carrier k^5



The projectors retain initial basis directions with dimensions $5, 4, 4, 2, 1, 1, \ldots$. In this explicitly defined example the projector is constant from level five onward.

Infinite carrier $\ell^2(\mathbb{N})$



Row m retains all basis vectors e_j with $j \geq m$. Every row continues infinitely to the right. The display is a finite window, not a truncation of the actual carrier.

Why the finite chain eventually stops changing

Suppose idempotents satisfy

$$e_{m+1}e_m = e_m e_{m+1} = e_{m+1}.$$

Their ranges are nested. In dimension d, nonzero ranges allow at most $d - 1$ strict dimension drops. There may be long gaps between drops, but only finitely many drops. The chain therefore becomes constant and has nonzero intersection.

> A single repeated dimension does *not* certify final stabilization. The finite example deliberately has a plateau $4, 4$ before another drop. Finite dimension bounds the number of strict drops, not the time of the last one.

Why every infinite tail can be nonzero while their intersection vanishes

Let P_m project onto the closed span of $e_m, e_{m+1}, \ldots$. Each range is nonzero. But a vector belonging to every range has its j-th coordinate zero once $m > j$. Every coordinate is therefore zero:

$$\bigcap_{m \geq 1} P_m \ell^2(\mathbb{N}) = \{0\}.$$

> **Read this as:** Stabilization in a fixed finite carrier, stable images in an inverse system, and infinite tails in a Hilbert space are different situations. Their hypotheses cannot be exchanged.

Plate XII.17 / THE CONDITIONAL THEOREM / INPUTS AND CONSEQUENCES

Where the arithmetic still enters

Main text: revised Theorem 12.6 and Section 13 Main statement 12.6

The final theorem is a composition of proved implications with explicitly supplied data. Use a single coherent Frobenius normalization in every polynomial comparison. These are not competing labels: an implication may be internally proved while its arithmetic premises remain external.

Gate	What must be supplied	What the formal deduction gives
Residual	Residual automorphic occupancy and a represented deformation problem.	The rings, universal representation, and starting carrier are available as inputs.
Patching	An applicable patching package: full patched support, quotient, and control isomorphism.	Full support descends to the finite-level module or perfect complex.
Comparison	A ring map $f : R \to T$ with the stated support comparison, or the suitable isomorphism and carrier identification.	Support is transported at compatible points.
Fiber	A Hecke point $y : T \to E'$ over the same field, with $y \circ f = x$.	A nonzero scalar fiber and an actual eigensystem. In the derived case, one survives in a cohomological degree.
Realization	Classicality for that eigensystem, an attached representation, continuity, equality of the universal specialization and attached Frobenius polynomials on a dense union of conjugacy classes, and semisimplification/Brauer–Nesbitt inputs.	The specialization is identified after semisimplification with the attached representation.

Support alone does not supply a point over the same field

The inclusion $\mathbb{Q} \to \mathbb{Q}(i)$ preserves full support of the nonzero scalar-extended carrier. But no $\mathbb{Q}$-algebra map $\mathbb{Q}(i) \to \mathbb{Q}$ exists: the image of i would be a rational number a with $a^2 = -1$. Thus one cannot infer the required Hecke point y merely from a support statement.

Propagation needs two existing families and the same automorphic object

After producing an automorphic object π, retain *that same* π. The propagation statement assumes both the original compatible family and the family attached to π already exist, with the common Frobenius polynomial data and the stated comparison hypotheses. It then identifies corresponding members. It does not construct a missing family.

> **Read this as:** The algebra locates and transports supplied nonzero data. Arithmetic inputs establish occurrence, classicality, and attachment. A visualization must not turn one of those inputs into an automatic consequence.

Plate XII.18 / READING ROUTE / SOURCE CROSSWALK / CHECKS

Return to the proofs with a concrete object

Reading guide to Chapter XII and its formal contract Main statement 3.1

Use the example on the left to keep the relevant object in view while reading the exact statement on the right. The guide is explanatory. The main text supplies the theorem hypotheses and the proof contract.

Plate	Question or calculation	Chapter XII reference
XII.3	Quotient monomials before or after completed combination	Theorem 3.1; Corollary 3.2
XII.4	A test point factors exactly when its local equations vanish	Proposition 3.3
XII.5	Compatible precision data and the closed-kernel argument	Section 3; Section 10
XII.6–XII.7	Minors, fiber dimensions, support, and nilpotent structure	Theorems 6.3–6.4
XII.8	Base change and transport of intrinsic Fitting ideals	Proposition 6.5; Theorem 7.1
XII.9	A quotient loses a generic point without creating a projector	Propositions 3.4 and 7.3
XII.10	Auxiliary freeness versus full deformation support	Theorems 6.7, 6.9, 6.11; Example 6.13
XII.11	A two-term projective complex and its derived fibers	Definition 6.14; Theorem 6.16
XII.12–XII.13	Whole primary packets, localization, and scalar fibers	Theorems 7.4, 8.2; Corollaries 8.3–8.4
XII.14	Sign-projector failure and augmentation recovery	Section 5; Remark 8.5; Section 11
XII.15–XII.16	Stable images, finite stabilization, and tail escape	Section 8, items 8.7–8.13
XII.17	Compatible Hecke points and conditional realization	Theorem 12.6; Section 13

Three short proofreading tests

Can a module have full support but fail to be faithful? Yes. On Plate XII.10, the module k over $k[[x]]/(x^2)$ has a nonzero nilpotent annihilator and full support.

Do equal support sets force equal Fitting ideals? No. On Plate XII.7, (xy) and (x^2y) have the same vanishing set but different infinitesimal behavior.

Do nonzero finite levels force a nonzero inverse limit? No. On Plate XII.15, $N^3 = 0$ forces every compatible sequence to vanish, although every level is k^3.

Provenance and reproducibility

The source is Chapter XII, *Patching Support, Hecke Packets, and Conditional Automorphic Completion*. The theorem references above identify statements in that chapter. Its formal companion gives the source and evidence map. The existing 49-claim revised formal contract and its inherited original-wording boundaries are unchanged.

The accompanying Python script performs 46 exact checks: exhaustive finite test-ring enumeration, symbolic minors and CRT identities, exact matrix calculations, and the identities behind the stable-core examples. All drawings are reproducible vector graphics. Further worked details appear on Plates XII.19–XII.25. The inherited revised/original claim boundary is on Plate XII.26. The illustrative examples are not newly certified in Lean by this guide. Its finite checks are recorded separately. Infinite claims are supported by the arguments given here, not by extrapolating a finite image.

Plate XII.19 / TOPOLOGICAL ALGEBRA / THE COMPARISON MAP

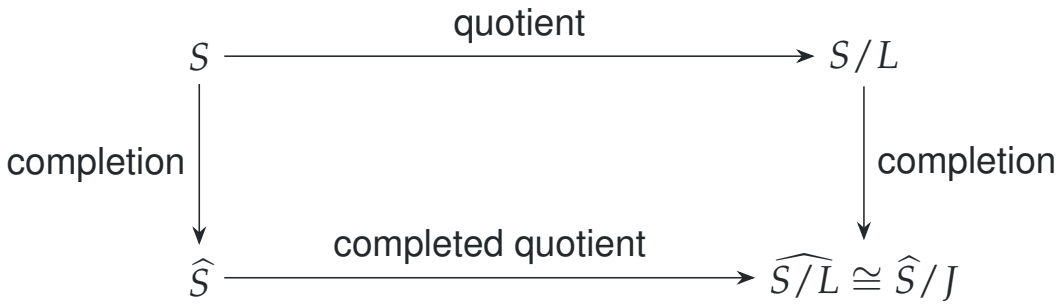
Complete the actual topology

Main text: Theorem 3.1, Corollary 3.2 and Proposition 3.3 Main statement 3.1

The hat in $\widehat{\otimes}$ names an actual completion. For the finite family of local rings, start with the ordinary tensor and its ideal of definition:

$$S = \bigotimes_{i=1}^{s} R_i, \qquad \mathfrak{a} = \mathfrak{m}_{\mathcal{O}} S + \sum_i \mathfrak{m}_{R_i} S, \qquad L = \sum_i I_i S.$$

The comparison uses the $\mathfrak{a}$-adic additive uniformity on S, the image ideal topology on S/L, and the closed extended ideal $J = \overline{L\widehat{S}}$.



The square names the actual quotient and completion maps. The lower identification is bicontinuous and preserves the coefficient map and every canonical factor map. The square does not assert that completion leaves S unchanged.

The explicit model revisited

For $R_1 = \mathbb{Z}_5[[x]]$ and $R_2 = \mathbb{Z}_5[[y]]$, the completion of their ordinary tensor in this topology is $\mathbb{Z}_5[[x, y]]$. With L generated by x^2, y, the lower quotient is $\mathbb{Z}_5[\varepsilon]/(\varepsilon^2)$ with its $(5, \varepsilon)$-adic topology. The containment on Plate XII.5 identifies this with its 5-adic topology.

Boundary cases are part of the contract

Case	Actual object and consequence
Empty family	$S = \mathcal{O}$ and $\mathfrak{a} = \mathfrak{m}_{\mathcal{O}}$. The scalar adic topology remains. An empty sum of factor ideals does not erase it.
All $I_i = 0$	$J = 0$; the comparison returns the completed tensor itself.
Some $I_i = R_i$	$J = \widehat{S}$ and both sides are the zero ring. This comparison does not give a local ring with residue field k.

For global base change, the structural map $A \to B$ belongs to the continuous local coefficient-algebra setting. Then $B \widehat{\otimes}_A (A/J) \cong B/\overline{JB}$. For a continuous algebra point to a complete Hausdorff target, closedness of its kernel gives the factorization test. Locality of a point, when supplied, passes to its unique quotient factor.

> **Read this as:** Closure and topology are structural data. A list of equations in a finite model illustrates the quotient but does not replace the actual completed comparison.

Plate XII.20 / RESIDUAL IMAGE / ONE EXPLICIT CERTIFICATE

A finite adequacy calculation

Main text: Theorem 4.2; label `thm:s3_adequacy`. Adequacy convention: external Standard Fact 4.1. Main statement 4.2

Over $k = \mathbb{F}_7$, the standard representation of S_3 has matrices

$$s = \begin{pmatrix} 6 & 1 \\ 0 & 1 \end{pmatrix}, \qquad t = \begin{pmatrix} 0 & 6 \\ 1 & 6 \end{pmatrix}.$$

All entries and calculations on this page are modulo seven. Direct multiplication gives $s^2 = t^3 = I$ and $sts = t^{-1}$. Their eigenvalues are $\{-1, 1\}$ and $\{2, 4\}$. Distinct eigenvalues allow an exact projector formula

$$P_{D,\alpha} = (\alpha - \beta)^{-1}(D - \beta I).$$

sign constituent

$$W = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad P_{t,2} = \begin{pmatrix} 2 & 4 \\ 3 & 6 \end{pmatrix} \qquad \operatorname{Tr}(PW) = 3$$

---

standard constituent

$$W = \begin{pmatrix} 6 & 4 \\ 3 & 1 \end{pmatrix} \qquad P_{s,1} = \begin{pmatrix} 0 & 4 \\ 0 & 1 \end{pmatrix} \qquad \operatorname{Tr}(PW) = 6$$

Each row displays the supplied matrix, an actual canonical eigenprojector and its nonzero trace pairing. It detects that stated adjoint constituent. It does not create a global automorphic carrier.

Write $W(a,b,c) = \left(\begin{smallmatrix} a & b \\ c & -a \end{smallmatrix}\right)$. The manuscript exhibits $Z = W(2,3,4)$, $A = W(1,1,2)$ and $B = W(1,5,6)$. Their coordinate determinant is 3. Thus they are a basis of the trace-zero adjoint. Conjugation splits this space as $\mathrm{sgn} \oplus V$. In the drawing the standard witness is $W(6,4,3) = 3A + 3B$.

Why the cohomology argument is uniform in the module

Since $|S_3| = 6$ is a unit in $\mathbb{F}_7$, a cocycle $f : S_3 \to M$ is a coboundary: set $u = -6^{-1} \sum_h f(h)$ and sum $f(gh) = gf(h) + f(g)$ to obtain $gu - u = f(g)$. This proves $H^1(S_3, M) = 0$ for every $\mathbb{F}_7[S_3]$-module M. The printed invariant calculation separately proves $H^0(S_3, \mathrm{ad}^0 V) = 0$.

What makes this a finite certificate

The manuscript also checks absolute irreducibility and projector detection on every irreducible adjoint submodule. In this example $7 \nmid 2$, and every group element's eigenvalues lie in $\mathbb{F}_7$, as required by the stated convention. The two displayed pairings alone are a worked part of that complete certificate.

> **Read this as:** The calculation is exact for $S_3/\mathbb{F}_7$. Applying an automorphy lifting theorem still requires its actual residual image hypotheses and residual occupancy. No adequacy theorem at coefficient prime two follows from this picture.

Plate XII.21 / AUXILIARY ALGEBRA / AN EXACT MONOMIAL BASIS

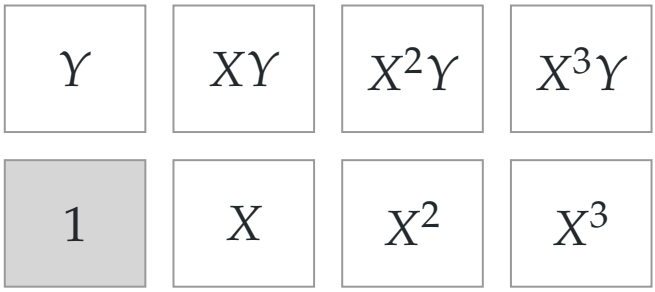
Augmentation in two variables

Main text: Proposition 5.1, Theorem 5.2 and Theorem 5.4 Main statement 5.2

Take $\Delta = C_4 \times C_2$ in residue characteristic two. If g and h generate its two factors, put $X = g - 1$, $Y = h - 1$. Then

$$\mathbb{F}_2[\Delta] \cong \mathbb{F}_2[X, Y]/(X^4, Y^2).$$

The identity $(1 + X)^4 - 1 = X^4$ and its Y analogue provide the relations. The eight surviving monomials provide the exact dimension $|\Delta| = 8$ and establish the isomorphism after surjectivity.



augmentation: $1 \mapsto 1$; $X, Y \mapsto 0$

All eight basis monomials are displayed. The filled scalar slot survives augmentation. All other displayed monomials lie in the augmentation ideal. This is a basis array, not a spectrum with eight points.

The residual ring is local

The augmentation ideal is (X, Y), its quotient is $\mathbb{F}_2$, and $(X, Y)^5 = 0$: a degree-five monomial has X-exponent at least four or Y-exponent at least two. Every element outside (X, Y) has the form $1 + n$ for nilpotent n, with inverse the finite geometric series. Thus (X, Y) is the unique maximal ideal, and the only idempotents are $0, 1$.

A quotient of groups gives a quotient of algebras

Map $C_4 \times C_2 \to C_2$ by $g \mapsto \bar{g}$, $h \mapsto 1$. The induced algebra map sends X to $\bar{X}$ with $\bar{X}^2 = 0$ and sends Y to zero. It is surjective. Augmentation commutes with this map because every group basis element is sent to another group element of augmentation one.

Recover a free carrier, with its base ring named

For $\Lambda = \mathbb{Z}_2[\Delta]$ and the right module $M = \Lambda^r$,

$$M \otimes_{\Lambda, \epsilon} \mathbb{Z}_2 \cong M/M \ker \epsilon \cong \mathbb{Z}_2^r.$$

The quotient applies augmentation to each basis coordinate. This remains valid for nonabelian finite two groups with the stated right-module convention. It establishes the rank over the coefficient ring after coinvariants, without establishing full support over a separate deformation ring.

> **Read this as:** The augmentation quotient keeps the scalar coordinate. It needs no averaging denominator and supplies no missing hypothesis on patching support.

Plate XII.22 / FINITE OCCUPANCY / MAPS NEED NOT BE SURJECTIVE

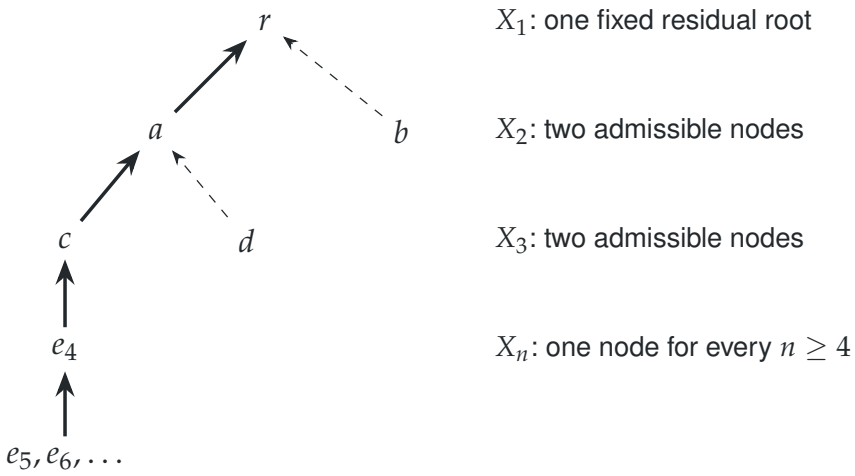
A finite forest selects a branch

Main text: Theorem 10.3 and Corollary 10.4 Main statement 10.3

A finite nonempty set at every precision, with total reduction maps, can select a compatible branch even when some nodes have no deeper lift. Here is an explicit system:

$$X_1 = \{r\}, \quad X_2 = \{a, b\}, \quad X_3 = \{c, d\}, \quad X_n = \{e_n\} \quad (n \geq 4).$$

Both a, b reduce to r; both c, d reduce to a; e_4 reduces to c; and e_{n+1} reduces to e_n for every $n \geq 4$.



Every displayed edge is an actual reduction map. The formula specifies the infinite continuation. The chosen branch is $r, a, c, e_4, e_5, \ldots$. The nodes b, d are admissible at their own precision and have no successor.

What the compactness argument proves

For general finite levels, the reduction forest is a locally finite rooted tree with vertices at every depth. An infinite branch exists. Surjectivity is sufficient but is not necessary: the map $X_3 \to X_2$ above misses b, and $X_4 \to X_3$ misses d.

Finite occupancy does not guarantee a branch through a prescribed node at a higher level. It also does not prove arithmetic occupancy of the finite sets. Those sets must already have been supplied as admissible realizations of the declared spine and residual root.

Why the counterexample with vector spaces is consistent with this

The nilpotent system on Plate XII.15 has a compatible sequence: the zero sequence. Its finite levels are nonempty sets because they contain zero. The question there is whether the inverse limit contains a *nonzero* vector, which needs stable-core data. Here the question is nonemptiness of an inverse limit of admissible finite sets. Different conclusions require different hypotheses.

> **Read this as:** Finite set compactness chooses a branch among existing admissible nodes. It neither extends every node nor produces an arithmetic node absent from the finite forest.

Plate XII.23 / COMPANION COMPLETION / ACTUAL REPRESENTATIVES

Classes are not matrix frames

Main text: revised Theorem 10.5 and its preceding admissibility hypotheses Main statement 10.5

Let $\Gamma = C_2$ with the discrete topology, and represent its generator by $D = \mathrm{diag}(1, -1)$ over $\mathbb{Z}_5$. At precision n, its conjugacy class is an admissible class in $\mathrm{GL}_2(\mathbb{Z}/5^n\mathbb{Z})$. These classes are compatible. Arbitrary representatives need not be.

modulo 25 **modulo** 5

$$\begin{pmatrix} 1 & 23 \\ 0 & 24 \end{pmatrix} \xrightarrow{\text{reduction}} \begin{pmatrix} 1 & 3 \\ 0 & 4 \end{pmatrix} \neq \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

same conjugacy class; incompatible chosen matrices

$$\begin{pmatrix} 1 & 0 \\ 0 & 24 \end{pmatrix} \xrightarrow{\text{reduction}} \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

strictly compatible representatives

The upper representatives are conjugate to the same diagonal involution, but their reductions are unequal matrices. The lower choices use D consistently at both precisions. All entries displayed are canonical residues.

Calculate the mismatch

For $Q = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$,

$$QDQ^{-1} = \begin{pmatrix} 1 & -2 \\ 0 & -1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 23 \\ 0 & 24 \end{pmatrix} \pmod{25}.$$

It reduces to $\left(\begin{smallmatrix} 1 & 3 \\ 0 & 4 \end{smallmatrix}\right)$ modulo five, not the chosen diagonal matrix $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 4 \end{smallmatrix}\right)$. Nevertheless, their residual conjugacy classes agree.

The proof chooses representatives through a finite tree

A finite coefficient quotient has a finite group of invertible basis matrices. Each of the finitely many admissible classes therefore has only finitely many representatives. Take *all* actual continuous representatives of the admissible classes at each level. Reduction is a total map between these finite nonempty sets because admissibility is preserved under reduction. A branch then gives strictly compatible matrices. Completeness reconstructs entries and inverse entries over $\mathcal{O}_\lambda$. Finite-level continuity gives continuity in the ideal-adic topology.

This argument does not assume that a chosen conjugating matrix lifts, nor that every representative has an admissible successor. The theorem retains the declared residual *class*. It specifies neither a prescribed residual frame nor uniqueness up to integral conjugacy.

> **Read this as:** Compatibility of classes is the input. Strictly compatible continuous representatives are a constructed output, obtained with the hypotheses on finite frames and coefficients in the revised statement.

Plate XII.24 / ADIC EFFECTIVITY / DECLARED POLYNOMIAL CONDITIONS

Equations survive every precision

Main text: revised Theorem 10.5 and Corollary 3.2 Main statement 10.5

The preceding involution gives an exact polynomial condition. For a matrix $A = (a_{ij})$, require all entries of $A^2 - I$ to vanish. In

$$B = \mathbb{Z}_5[a_{11}, a_{12}, a_{21}, a_{22}],$$

the ideal J is generated by

$$a_{11}^2 + a_{12}a_{21} - 1, \qquad a_{12}(a_{11} + a_{22}),$$
$$a_{21}(a_{11} + a_{22}), \qquad a_{21}a_{12} + a_{22}^2 - 1.$$

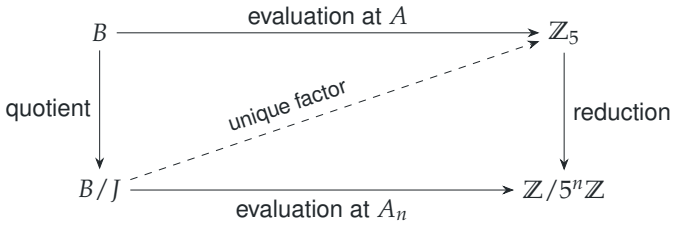
For D and every conjugate QDQ^{-1} these equations vanish. The condition is valid in every admissible frame. By contrast, the equation $a_{12} = 0$ holds for D but fails for the conjugate displayed on Plate XII.23.

The reconstruction argument is coefficientwise

Let A_n be compatible matrices modulo 5^n, each satisfying these four equations. Completeness reconstructs $A \in M_2(\mathbb{Z}_5)$. For each polynomial $f \in J$, evaluation commutes with reduction, so

$$f(A) \in \bigcap_{n \geq 1} 5^n \mathbb{Z}_5 = \{0\}.$$

Separatedness supplies the last equality. Thus the actual evaluation map $B \to \mathbb{Z}_5$ kills J and factors uniquely through B/J. If inverse matrices were reconstructed as well, this is an actual representation in $\mathrm{GL}_2(\mathbb{Z}_5)$.



Every finite evaluation kills the declared ideal. Reduction commutes with evaluation, and separatedness gives integral vanishing. The displayed diagram represents this exact polynomial condition.

The scope of the effectivity conclusion

The companion theorem needs integral Frobenius polynomials, finite admissible continuous classes, reduction closure, finite occupancy, and the declared triviality subgroup. A condition in one frame needs an argument using compatible frames. An arbitrary Hodge or Weil–Deligne condition is not automatically a polynomial ideal in matrix entries.

Represented local quotient conditions use continuous ring maps and the closed-kernel argument on Plate XII.19. These are different supplied objects from the polynomial evaluation model.

> **Read this as:** The precise coefficient and admissibility hypotheses preserve declared equations. Representability of a desired arithmetic condition remains a separate input.

Plate XII.25 / A FINITE SEPARATOR / A KNOWN SEMISIMPLE CARRIER

Recover multiplicities from traces

Main text: Proposition 10.9 Main statement 10.9

Suppose a known semisimple carrier is a sum of two known constituents with multiplicities m_1, m_2. Consider the independent trace functions

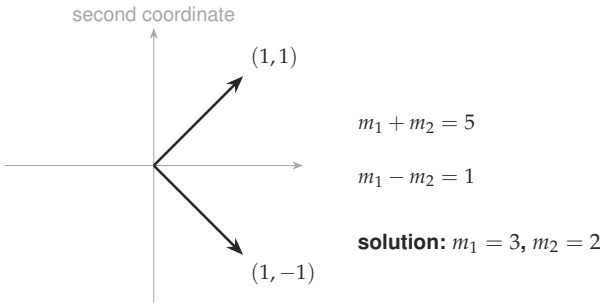
$$t_1(\omega) = 1, \qquad t_2(\omega) = \omega, \qquad \omega \in \mathbb{Q}.$$

At the two sample points $\omega = 1, -1$, their evaluation matrix is

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \det A = -2.$$

It is invertible over $\mathbb{Q}$. If the measured trace vector is $(5, 1)$, solve

$$A \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}.$$



The vectors $(1, 1)$ and $(1, -1)$ are the two independent rows of the exact evaluation matrix. The coefficients $3, 2$ are recovered by the displayed linear equations. This illustration uses functions on $\mathbb{Q}$, not a newly constructed arithmetic Frobenius set.

Why finitely many evaluations suffice in general

Independent functions $t_1, \ldots, t_r$ on Ω have evaluation vectors spanning E'^r. Otherwise an annihilating nonzero linear functional would give a forbidden relation among the functions. Choosing r independent evaluations gives an invertible matrix that recovers coefficients in their known sum.

Two checks before using this in arithmetic

The trace functions must be independent on the permitted Frobenius set. The carrier must already be semisimple with the stated constituents. Arbitrary samples supply neither premise. Nonnegative integral multiplicities come from representation data, not every rational solution of a linear system.

The coefficient field matters again

The displayed integer matrix reduces modulo two to a matrix with determinant zero and two coincident rows. Those residual samples cannot recover the two coefficients. This matches the trace caution on Plate XII.14: characteristic-zero trace detection cannot be silently transported to residue characteristic two.

> **Read this as:** A finite separator reconstructs coefficients in supplied independent trace data. It is not a construction of the constituents, their compatible systems, or automorphic realizations.

Plate XII.26 / PROVENANCE / REVISED SCOPE AND ORIGINAL WORDING

Read the exact claim contract

Source: the revised formal contract and original-wording comparison in the formal companion Main statement 12.6

This guide explains the revised manuscript and retains its inherited verification boundaries. There are two censuses, with different statements as their subjects:

Inherited contract	Complete	Partial	Unchecked
Revised 49 internal claims	49	0	0
Original wording of those claims	41	7	1

The eight inherited amendments

Claim	Scope that the revised statement makes explicit
6.12	Each irreducible component is contained in full support. The original containment was reversed.
9.4	Hausdorff coefficient topology and continuity for the compared representations.
9.5	The realized locus consists of points satisfying those topology and continuity hypotheses.
9.6	The surviving cohomological degree belongs to the declared classicality range. Comparison inputs are explicit.
10.5	Complete separated coefficients, finite continuous admissible classes, integrality, and actual quotient equations in every admissible frame.
12.2	Continuous characteristic-zero pointwise comparison under the stated realization input.
12.4	Applicability of realization at the selected derived eigensystem, with the comparison inputs exposed.
12.6	A compatible Hecke point $y \circ f = x$, actual ordinary or derived carriers, and both existing families for the same automorphic object.

In the inherited record, original 6.12 is unchecked. Original 10.5 and the other six amended claims are partial. Revised completeness does not certify the stronger original wording. Claims 3.1 and 3.3 required no additional amendment in the recorded completion.

Scope clarifications

The companions expose finite S, a coherent Frobenius normalization, the local coefficient-test category in 3.3, the $r > n$ Fitting boundary, and both compared Frobenius polynomial identities in ordinary completion. These clarifications are distinct from the eight inherited amendments. No new Lean replay certifies them.

Plate XII.27 / VALIDATION / FOUR DISTINCT KINDS OF EVIDENCE

How to assess the evidence

Inherited source records, current illustrative calculations and final editorial review Main statement 12.6

What this companion adds and what its checks mean

The guide preserves the earlier worked examples. TikZ generates every drawing from exact objects. There are no raster figures. Python checks finite enumerations, minors, projectors and matrices. The text argues uniform conclusions. These checks supply no new Lean certificate or census update. Supporting Markdown (.md) files accompany the source.

> **Read this as:** The source's internal formal implications are distinguished from the arithmetic inputs: representability, residual occupancy, applicable patching, classicality, attached representations and compatible systems, Frobenius density, semisimplification and Brauer–Nesbitt. Universal companions and unrestricted reciprocity remain outside the proved scope.

Inherited labels and current mathematical judgments

The original-status labels record historical verification coverage. The current mathematical audit judges the actual statements. These are different questions:

Original	Inherited label	Current judgment of the statement
6.12	unchecked	The reversed containment is false: in the node of Plate XII.9, $M = R$ has full support, which is not contained in either single component.
10.5	partial	The unrestricted effectivity conclusion lacks the coefficient, continuity, integrality and frame hypotheses made explicit in the statement.

Use each check for its actual subject

Evidence	What its subject is
Inherited Lean record	The mapped revised statements under their supplied hypotheses. This guide neither replays that build nor changes the original-wording record.
Current exact Python checks	Forty-six illustrative assertions: 31 inherited checks and 15 added checks on the printed finite examples and identities.
Mathematical arguments	Uniform quotient, specialization and all-level conclusions, with their hypotheses retained in the text. A finite diagram is not an infinite proof.
PDF and rebuild checks	Topic alignment, bounds, vector-only assets, resolved references, and an isolated byte/text/page comparison. These are document checks.

The source and claim maps identify all 49 internal statements and their exact mathematical labels. Their records separate formal coverage, finite checks and ordinary arguments. Finite calculation does not verify residual automorphic existence, classicality, attached compatible systems or a universal arithmetic theorem.

Further studies

The following studies develop the restricted arithmetic constructions, obstructions and remaining realization questions associated with Chapter XI.

READING ROUTES

Arithmetic realization and its obstructions 1483

Visual studies of arithmetic realization 1551

FURTHER STUDY

Arithmetic realization and its obstructions

These studies distinguish finite packet constructions, actual arithmetic candidates, full-character occurrence and comparison for the same candidate.

READING ROUTES

Research companion 1484

Scope

Conjecture 66 remains open. Its unresolved requirements concern general arithmetic realization, H5 for the same candidate, corrected full-character occurrence and actual native data under the stronger contract.

Reading the arithmetic studies

The three parts of this companion develop restricted constructions, compare alternative arithmetic routes, and isolate the remaining conductor and marking obligations.

Part I. Restricted arithmetic constructions. Sections 1–13 develop odd-prime and weighted dyadic recursion, full-character induction, native quotient obstructions, and low-rank enlarged-carrier recovery.

Part II. Monomial routes, Clifford descent and exact obstructions. Sections 14–21 separate admissible induction towers, genuine multiplicities, character occurrence and self-twist failure, compact binders, and exact corner obstructions.

Part III. Conductors, native markings and the equal-capacity boundary. Sections 22–30 examine conductor normalization and candidate filters, separate pair invariants from quotient capacity, and give the fixed-refinement criterion with explicit conditional finite corrections.

Source provenance. Part I remains a preserved reader because its complete editable source archive was not recovered. Parts II and III, the assembly programs, exact suites, source maps, and evidence ledger have editable sources. The retained compact source package contains selected supporting research records, not the entire source archive.

Mathematical boundary. The general realization conjecture, same-candidate H5, and corrected-character occurrence retain their separate open statuses. Host dimension does not supply a canonical native quadratic marking. The finite correction theorems require the stated actions and permitted changes. The recorded conductor and marking audit includes no fresh Lean build or external peer review.

Remaining arithmetic task. The chosen candidate’s actual native marking and phases must still be supplied. The visual studies illustrate this obligation without replacing it.

Reading scope

This study collects the mathematical progress toward the arithmetic realization endpoint of Chapter XI. Its central object is not a numerical completion score. It is the proposed passage from a supplied finite packet, with its character values, phases and transport maps, to an actual supercuspidal representation carrying that same packet.

The general passage remains Conjecture 66. That fact does not erase the constructions obtained along the way. The available record contains finite comparison theorems, arithmetic dictionaries, explicit parameter constructions, full character arguments, exact obstructions to particular hosts, and a low-rank dyadic repair using larger arithmetic inducing groups. The purpose of these studies is to develop these results in their own right, without confusing a restricted construction with a universal realization theorem.

The direct constructions make the parameter recursion, full characters, low-rank recovery, corestriction normalization, native traces, depth endpoints, and noncompact inducing data explicit. The principal Bushnell–Kutzko inputs were checked against the retained book and remain classical inputs. No new Lean compilation or external peer review is claimed. Earlier records described as independent reviews refer to internal reviews.

The evidence record distinguishes manuscript coverage, ordinary proofs, classical inputs, exact computation, inherited formal evidence, and unresolved research obligations. These are separate kinds of evidence. None is treated as a substitute for another.

This research companion develops the arithmetic questions left open in Chapter XI. The retained source archive contains detailed proofs, construction notes, audits, and research records. Its machine-readable source map identifies the files behind each section and distinguishes executed finite checks from inherited evidence.

The monomial, Clifford-descent, and binder branches are developed in Sections 14–21. Conductor and marking questions are examined in Sections 22–30. The synthesis does not claim to incorporate every argument in every retained research record. The formal evidence and source map preserve the remaining provenance boundaries.

1 The realization problem and its quantifiers

1.1 Three different mathematical tasks

Chapter XI separates finite comparison, arithmetic existence and local-factor characterization. These tasks interact, but they have different inputs. A finite Heisenberg representation can be constructed from a nondegenerate alternating pairing. A compatible family of intertwiners can be constructed after its source and target representations are supplied. Neither statement, by itself, constructs the required representation of a general linear group over a local field.

Fix the reciprocity, additive-character, local-factor and phase conventions of Chapter XI. Let $\mathcal{A}$ denote the declared class of irreducible wild Weil atoms. For $\rho \in \mathcal{A}$, write d_ρ for its *supplied* assembled finite datum. The historical names H1–H4 refer to the following data: an intrinsic finite break decomposition, equivariant symplectic bridges to the supplied breakwise type quotients, compatible central-extension and phase data including the declared markings, and a coherent transport system whose matrix holonomy obstruction vanishes. These are assumptions on a specified source, not a theorem that every wild Weil representation already comes equipped with a canonical instance [1, 3].

Definition 1.1 (Arithmetic realization witness)**.** A witness $R(d_\rho, \pi)$ consists of actual arithmetic inducing data for an admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(F)$, together with comparison and recovery maps carrying the prescribed assembled finite packet. Those maps must preserve the declared stabilizer action, supplied character values, central-extension and phase data, and compatible transports. Recovery returns the same π, with the fixed central character and noncompact completion.

An enlarged actual inducing carrier is permitted by this contract. The entire assembled packet need not be the original maximal-simple-type quotient. Conversely, an independently fixed native marking, an original matrix subgroup embedding, or a prescribed numerical depth is not automatically preserved merely because the scalar extension is preserved. Such requirements must be stated and proved separately.

1.2 The conjecture and the surviving comparison theorem

The endpoint is the following statement, with its historical identifier retained:

> **Conjecture 66.** For every $\rho \in \mathcal{A}$ with its supplied H1–H4 datum d_ρ, there exist an actual admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(F)$ and a witness $R(d_\rho, \pi)$.

This is the realization requirement isolated from historical Theorem 16.2, original proof obligation 66, and row 95 of the all-records register. It is not row 66 of that register. It asks for one witness per admitted source. It does not ask that every proposed host work, or that mutually inconsistent noncompact completions be reconciled.

Proposition 1.2 (Comparison of the same realized candidate)**.** *Suppose $R(d_\rho, \pi)$ is supplied for one actual admissible supercuspidal π of the required rank. Suppose H5 holds for this π: its required pair local factors agree with the factors prescribed for ρ on the stipulated characterizing family of twists, in the fixed normalizations. Assume the applicable local-factor characterization for that family and class. Then $\pi \simeq \mathrm{LLC}(\rho)$.*

Proof. The characterization applies to the two actual representations π and the reference representation $\mathrm{LLC}(\rho)$. H5 provides equality of the characterizing factors, so their isomorphism classes agree. The

witness R identifies this same candidate as the carrier of the prescribed packet. The argument neither constructs π nor derives H5 from the finite datum. □

Evidence and scope. This is the pointwise conditional implication in the current paper, not a new local converse theorem. The local-factor characterization is a stated external input.

The distinction can be written as

$$\forall \pi\, [R(d,\pi) \Rightarrow H5(\pi)] \qquad \text{versus} \qquad \exists \pi\, [R(d,\pi) \wedge H5(\pi)].$$

The first statement is true when there are no realizers. The second supplies a realizer. Likewise, a witness of $R(d,\pi_1)$ and a verification of H5 for an unrelated π_2 cannot be combined. This elementary point is important because the general formal interface can quantify over several realizers, while the pointwise comparison uses one fixed candidate.

1.3 What a host obstruction does and does not refute

Let $\mathcal{H}(d)$ be the class of admissible hosts for a datum d. A failed construction establishes that one chosen H does not realize d, or perhaps that no H in a particular subclass does. To refute Conjecture 66 for that datum would require excluding every allowed arithmetic witness. The capacity and inner-normalizer obstructions below are therefore substantive negative results without being counterexamples to the conjecture. They identify constraints a successful construction must avoid.

2 What the formal continuation actually contributes

2.1 The original obligation count is not a measure of distance

The inherited formal record organizes the original paper around 66 obligations. Its retained 65/66 checkpoint means that the final general realization obligation has not been closed under the original scope. It does not mean that one short lemma is left, or that the remaining task is one sixty-sixth of the mathematical work.

2.2 From abstract dictionaries to arithmetic attachments

The continuation sequence matters because it repeatedly strengthens the interpretation of a formal object. A module can prove a theorem about a lattice index without identifying that lattice with the canonical lattice of a supplied arithmetic type. A structure can contain an existence field without constructing an inhabitant of that field. An arbitrary character with the correct values on a tested subgroup need not be the original full character.

Continuation 110 develops Morita-lattice indices, exact one-plus comparisons, and same-carrier dictionaries. Its remaining task is not to rename an abstract parameter as canonical. It is to attach the arithmetic groups, the original character, the canonical extension, and the residue factor to the exact formal data. Continuation 111 moves the character and index attachment further and separates a valid one-break

conditional comparison from the general common-host problem. Separate actual break types do not automatically assemble into one arithmetic type of the total rank [4, 5].

Continuations 112 and 113 make the boundary more concrete. The former supplies complex scalar-cochain transport and honest cyclic-action machinery. Its degree obstruction concerns a whole maximal-type carrier, rather than refuting every permissible assembly carrier. The latter records a cubic non-occurrence test: the first proposed simple character does not occur in the original automorphic-induction candidate. A corrected wild-unit proposal passes the stated self-twist test, but that positive test alone does not prove occurrence [6, 7].

2.3 The retained contract for formal evidence

A Lean theorem certifies its displayed conclusion from its displayed binders and imported declarations. It does not silently instantiate a local field or produce an arithmetic representation that was supplied as a field of a structure. In the general comparison interface, `Realization.exists_atom` is an existence input. The source can still prove nontrivial descent, comparison and uniqueness results using that input. The existence obligation remains distinct.

3 Central products remove one proposed finite obstruction

The finite extension diagnosis [8] isolates a useful positive fact. Once the full factor extensions and their orthogonality are supplied, no further mixed scalar extension class is automatically left over. The following proofs reconstruct the elementary core of that diagnosis.

Proposition 3.1 (Orthogonal factors form a balanced central product)**.** *Let $1 \to Z \to P \to A \to 1$ be a central extension, with $A = \bigoplus_{i=1}^{d} A_i$ with $d \geq 1$. Let P_i be the inverse image of A_i. Assume elements of P_i and P_j commute for $i \neq j$. Then multiplication induces an isomorphism*

$$\Big(\prod_i P_i\Big)\Big/\{(z_i) \in Z^d : \prod_i z_i = 1\} \;\simeq\; P.$$

Proof. Commutation makes multiplication a homomorphism. Given $x \in P$, decompose its image in A as $\sum_i a_i$ and choose a lift $x_i \in P_i$ of each a_i. The difference between x and $\prod_i x_i$ is central and can be absorbed into one factor. Thus multiplication is onto. If $\prod_i x_i = 1$, its image in the direct sum gives $a_i = 0$ for every i. Hence each x_i lies in Z, and the only remaining condition is $\prod_i x_i = 1$. This is exactly the displayed kernel. □

This proposition uses the actual factor extensions, not just a list of dimensions or break labels. Distinct break numbers do not, by themselves, prove the required independence or commutation.

Proposition 3.2 (Same commutator over a divisible scalar center)**.** *Let A be a finite abelian group. Two central extensions of A by $\mathbb{C}^\times$ with the same alternating commutator are isomorphic by a map inducing the identity on A and on $\mathbb{C}^\times$. The isomorphism need not preserve additional independently fixed lifts or finite-center markings.*

Proof. Choose sections and write the two normalized cocycles as a_1, a_2. Their ratio $b = a_2/a_1$ has trivial commutator, so its extension is abelian. Write $A = \bigoplus_j \mathbb{Z}/n_j\mathbb{Z}$ and choose a lift u_j of each cyclic generator

in that abelian extension. Its n_j th power is a scalar. Since $\mathbb{C}^\times$ has n_j th roots, rescale u_j so that $u_j^{n_j} = 1$. The adjusted commuting lifts define a splitting. Thus b is a coboundary and the two original extensions are equivalent. The rescaling choices need not fix any prescribed lift. □

Automorphisms inducing the identity on quotient and center form $\mathrm{Hom}(A, \mathbb{C}^\times)$: they multiply a lift of a by a character value. This residual freedom is exactly why an unmarked scalar-extension equivalence is not a canonical identification of all source data.

3.1 Quadratic information survives only when its marking travels

In characteristic two, let $V = \mathbb{F}_2^2$ and

$$q_+(x,y) = xy, \qquad q_-(x,y) = x + y + xy.$$

They have the same polarization $b((x,y),(x',y')) = xy' + x'y$. Their Gauss sums are different:

$$\sum_{v \in V} (-1)^{q_+(v)} = 2, \qquad \sum_{v \in V} (-1)^{q_-(v)} = -2.$$

The numbers of zeros are three and one. Thus the alternating form alone cannot recover the quadratic refinement.

For a literal representation, put

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Both pairs (X, Z) and (iX, iZ) have commutator $-I$. In the first pair the generating lifts square to I. In the second they square to $-I$. They generate the usual dihedral and quaternion groups of order eight. After the center is extended to $\mathbb{C}^\times$, the scalar groups generated together with all scalar matrices coincide. Their distinguished finite subgroups and lift-square functions do not coincide.

This example explains the dyadic attachment rule. An extension map F_0 preserves a chosen square relation when the whole marked finite subgroup is carried along, because $F_0(a)^2 = F_0(a^2)$. It does not prove that this transported marking equals an independently fixed native marking of the host [19]. A kernel already killed by the scalar pushout cannot be recovered afterwards by an extension isomorphism.

4 Abstract stabilizer lifting is not an arithmetic normalizer

Two elementary arguments remove finite scalar obstructions under a coprimality hypothesis. They must not be misread as constructing matrices in the required arithmetic normalizer.

Proposition 4.1 (Coprime lifting of an extension action)**.** *Let A be a finite abelian p-group, and let P be a central extension by $\mathbb{C}^\times$. Let a finite group C of order m prime to p act on A, preserving the extension class. Then the action lifts to an action on P fixing the center.*

Proof. Choose a lift q_c for each $c \in C$. The kernel of the group of extension automorphisms over the quotient and center is $M = \mathrm{Hom}(A, \mathbb{C}^\times)$, a finite abelian p-group. Conjugation defines the C-module structure on M. Write the defect of the chosen lifts additively as

$$q_c q_d = f(c,d) q_{cd}, \qquad f(c,d) \in M.$$

Associativity gives

$$f(c,d) + f(cd,e) = cf(d,e) + f(c,de).$$

Let $S(c) = \sum_{e \in C} f(c,e)$. Summing the identity over e gives

$$mf(c,d) = S(c) + cS(d) - S(cd).$$

Multiplication by m is invertible on M. Put $b(c) = m^{-1}S(c)$ and replace q_c by $(-b(c))q_c$. Their new multiplication defect is zero. The adjusted lifts form the required action. □

The hypothesis that each quotient automorphism preserves the extension class supplies the initial individual lifts. Proposition 3.2 supplies that hypothesis when the quotient action preserves the commutator in the finite abelian setting. Neither proposition realizes those lifts by conjugation in $\mathrm{GL}_n(F)$.

Proposition 4.2 (Coprime linearization of projective intertwiners)**.** *Let C be finite of order m. Suppose U_c are invertible operators on a complex vector space of dimension D, with $U_c U_d = a(c,d)U_{cd}$ for a scalar cocycle a. If $\gcd(D,m) = 1$, the operators can be rescaled to a genuine representation of C.*

Proof. Taking determinants shows that a^D is the coboundary of $c \mapsto \det U_c$. Set $A(c) = \prod_{x \in C} a(c,x)$. The cocycle identity, multiplied over x, gives $a(c,d)^m = A(c)A(d)/A(cd)$. Choose integers r,s with $rD + sm = 1$. Then

$$a = \delta\big((\det U)^r A^s\big).$$

Rescaling by the inverse of this cochain eliminates the cocycle. □

For a Heisenberg carrier of dimension p^k and a prime-to-p group C, the coprimality condition holds. Two honest linearizations can still differ by a character. Thus this argument does not select a canonical beta-extension phase. Nor does it cover a stabilizer with p-torsion without further work [8, 23].

5 The compact binder theorem and the missing solvability input

The binder construction asks whether finite compatibility conditions can be joined into one global coefficient choice. The compactness theorem [9] identifies the precise missing input: joint finite solvability within one fixed compact model.

Theorem 5.1 (A fixed compact coefficient binder)**.** *Let F be a nonarchimedean local field, $\mathcal{O}$ its compact ring of integers, and ϖ a uniformizer. Let $X \subseteq \mathcal{O}^m$ be closed and let $f_j : X \to \mathcal{O}$ be continuous for $j \geq 1$. Define*

$$C_n = \{x \in X : f_j(x) \in \varpi^n \mathcal{O} \text{ for every } j \leq n\}.$$

There is an $x \in X$ with $f_j(x) = 0$ for all j if and only if every C_n is nonempty.

Proof. An exact zero belongs to every C_n. Conversely, each C_n is a closed subset of the compact space X, and $C_{n+1} \subseteq C_n$. Nonempty nested closed subsets of a compact space have nonempty intersection. Choose x in that intersection. For any fixed j, one has $f_j(x) \in \varpi^n \mathcal{O}$ for all $n \geq j$. The intersection of these ideals is zero, so $f_j(x) = 0$. □

The theorem does not require surjectivity of a chosen sequence of reduction maps. It does require a single compact coefficient space and simultaneous finite solvability within that space. Allowing the rank, pole order or number of coefficients to grow with precision changes the space and invalidates this argument unless a separate compact model is supplied.

5.1 When a finite empty level is an actual certificate

For polynomial equations with integral coefficients, or rational equations whose denominators remain units, a fixed finite residue template gives a finite test at each precision. If the template exactly describes X modulo ϖ^n, then emptiness of its complete solution set certifies failure of that fixed template. If there is no exact solution in the fixed compact model, Theorem 5.1 guarantees some empty level.

This is a semidecision procedure for nonexistence within that model. It provides neither a uniform bound on the first empty level nor a proof that a partial search was exhaustive. A failed random search, a resource cutoff and an actual empty finite level have different logical meanings.

Example 5.2 (Individual equations are not simultaneous solvability)**.** Over $\mathbb{Z}_2$, the equations $x = 0$ and $x = 1$ are each soluble to every precision. They have no common solution even modulo two. The nonempty sets needed in Theorem 5.1 contain joint solutions to the first n equations, not a separate solution for each equation.

Example 5.3 (A nonclosed admissibility condition can disappear)**.** The elements $x_n = 2^n$ are nonzero in $\mathbb{Z}_2$ and satisfy $x_n \equiv 0 \pmod{2^n}$. Their limit is zero. A requirement such as $x \neq 0$, or an unspecified nonzero determinant, is not automatically a closed condition. A fixed unit leading coefficient or a prescribed nonzero residue class is a different condition and can survive the limit.

5.2 The actual matrix binder obligation

The retained matrix template uses a square-zero variable x and expressions of the form

$$Q(x) = -s(xb_0x), \qquad \beta(x) = \gamma' + [x, b_0] - xb_0x, \qquad x^2 = 0.$$

The purpose is to force desired corestriction and leading-lane behavior inside actual matrix algebras. Compactness can assemble solutions only after the source-to-equation encoding is faithful, the admissibility conditions define a closed fixed template, and every finite collection of equations is jointly soluble. The general H1–H4 assumptions have not been proved to imply those three facts. The binder theorem therefore sharpens the realization problem without closing it.

6 Profiles, twists and the identity of the candidate

6.1 Compact restrictions forget noncompact information

Proposition 6.1 (Unramified twists are invisible on compact subgroups)**.** *Let J be a compact subgroup of $\mathrm{GL}_n(F)$. Then $v_F \circ \det$ vanishes on J. For any $a \in \mathbb{C}^\times$, the unramified character $\nu_a(g) = a^{v_F(\det g)}$ is trivial on J.*

Proof. The continuous image of the compact group J in the discrete group $\mathbb{Z}$ is a finite subgroup. The only finite subgroup of $\mathbb{Z}$ is zero. The character formula now gives the assertion. □

Consequently $\pi|_J$ and $(\pi \otimes \nu_a)|_J$ agree, while the central character on a scalar uniformizer changes by a^n. Choosing $a^n \neq 1$ makes this a visible difference. Compact packet recovery therefore has to retain the noncompact extension, not just its compact restriction.

6.2 Information tests do not replace full pair factors

The inherited rank-nine information test [10] is more specific. Its rank-nine examples, conditional on the pinned source construction used there, have the same recorded determinant, Frobenius matrix, slope, Swan conductor, projective Weil map and self-pair data. They differ in an actual scalar quotient map and are separated by a stated twist test. In the recorded test the Artin conductors are 33 and 45. The source does not claim equality of H5, and these examples are not counterexamples to local-factor characterization [10].

Twist transport has a positive use as well. Once a compatible base realization is genuinely supplied, its character and inducing data can be transported under the stated twist. That transports an existing witness. It does not supply the missing base witness. The distinction is the same one encountered in the formal existence field.

6.3 Equal numerical profiles do not determine quotient capacity

The later dyadic record gives an exact arithmetic comparison of two host patterns. At rank eight with $(h_1, h_2, h_3) = (1, 3, 5)$, the full defining tower has levels $(60, 18, 3)$, Swan conductor 60, torsion number 1 and self-pair conductor 342. Its symplectic type quotient has dimension 6. A skipped-degree pattern $8 \to 2$ can have the same three numerical invariants with quotient dimension 4 [18].

The formula makes the comparison transparent. Put $L = 12h_3$ and $c = h_1 + 4h_2$. For the stated skipped-degree construction,

$$a_{\mathrm{skip}} = 63 + 4L + 3c = 63 + 48h_3 + 12h_2 + 3h_1 = a_{\mathrm{full}}.$$

Yet the two capacity formulas remain different. The recorded exact checks verify the displayed arithmetic identities. The assertion that both patterns arise from actual simple strata remains an ordinary arithmetic claim established in the retained source.

7 The direct odd-prime construction

Evidence and scope. The odd-prime construction [11–14] uses ordinary parameter and character arguments relative to the explicit Bushnell–Kutzko inputs in Section 13. The cited statements and their principal hypotheses were checked in the recorded source audit. This supplies neither Lean proofs of simple-character theory nor a proof of H5.

7.1 Notation and the construction contract

Let p be odd and put $F = \mathbb{F}_p((t))$. Choose

$$0 < h_1 < \cdots < h_d, \qquad p \nmid h_i, \qquad q_i = (p+1)h_i, \qquad N = p^d.$$

The construction produces a literal matrix $\beta \in M_N(F)$, rather than a formal symbol for an unknown arithmetic parameter. Its generated field is maximal and totally ramified, and it normalizes a complete lattice chain. A subsequent, separate argument constructs a full simple character and an extended inducing representation. Only after those steps is there an actual supercuspidal candidate.

At an intermediate local field L, write $\mathcal{O}_L$ for its ring of integers and t_L for its chosen uniformizer. For a hereditary order $\mathfrak{A}$ with radical $\mathfrak{P}$, put $U^r_{\mathfrak{A}} = 1 + \mathfrak{P}^r$ for $r \geq 1$. The valuation $v_{\mathfrak{A}}(x)$ is the largest integer r with $x \in \mathfrak{P}^r$. On a complete period-N chain, scalar valuations satisfy $v_{\mathfrak{A}}(a) = N v_L(a)$. We use ordinary matrix trace over the displayed coefficient field. No trace is divided by a matrix size.

7.2 A separable degree-p step with a native trace identity

Lemma 7.1 (Artin–Schreier parameter and uniformizer)**.** *Let L have characteristic p and residue field $\mathbb{F}_p$, and let $q > 0$ with $p \nmid q$. The equation*

$$\alpha^p - \alpha = t_L^{-q}$$

defines a separable totally ramified extension E/L of degree p. With normalized valuation on E, one has $v_E(\alpha) = -q$. Choose $1 \le b < p$ and $a \in \mathbb{Z}$ with $qb \equiv -1 \pmod{p}$ and $pa = qb + 1$. Then $z = t_L^a \alpha^b$ is a uniformizer and $E = L(z)$. For odd p, $\mathrm{Tr}_{E/L}(\alpha) = 0$.

Proof. If $x \in L$ has negative valuation, then $v_L(x^p - x) = pv_L(x)$ is divisible by p. If its valuation is nonnegative, so is the valuation of $x^p - x$. Thus the right side is not an Artin–Schreier coboundary. All roots differ by constants in $\mathbb{F}_p$. The splitting field obtained by adjoining one root has Galois group a subgroup of the additive group of $\mathbb{F}_p$, so its degree is either one or p. The absence of a root in L forces degree p.

The derivative is -1, which proves separability. If $e = e(E/L)$, the valuation equation gives $pv_E(\alpha) = -eq$. Since $p \nmid q$, one has $p \mid e$. The extension has degree p, hence $e = p$ and $v_E(\alpha) = -q$. The residue degree is one. Finally

$$v_E(z) = pa - qb = 1, \qquad \mathrm{Tr}_{E/L}(\alpha) = \sum_{c \in \mathbb{F}_p} (\alpha + c) = 0$$

for odd p. An element of valuation one cannot belong to L, whose value group in E is $p\mathbb{Z}$, so $L(z) = E$. □

The minimal polynomial $f(X) = \sum_{k=0}^{p} a_k X^k$ of z, with $a_p = 1$, is Eisenstein. It is explicitly computable from

$$f(X) = \prod_{c \in \mathbb{F}_p} \bigl(X - t_L^a (\alpha + c)^b\bigr),$$

reducing by the Artin–Schreier relation. This instruction uses only finite field operations. It does not call an existence oracle for the desired final field.

7.3 The corestriction normalization and its filtered right inverse

In the basis $1, z, \dots, z^{p-1}$ of E/L, let T_{ij} be the coefficient matrix unit. The trace-dual basis is

$$z_j^* = \frac{\sum_{k=j+1}^{p} a_k z^{k-j-1}}{f'(z)}, \qquad 0 \le j < p.$$

The trace corestriction pr satisfies $\mathrm{pr}(T_{ij}) = z^i z_j^*$. Set

$$\kappa = -\frac{f'(z)}{z^{p-1}}, \qquad s = \kappa\,\mathrm{pr}.$$

The leading term of $s(T_{ij})$ is $-z^{i-j}$. Every other term has strictly higher valuation. Indeed $v_E(a_k) \ge p$ for $0 \le k < p$, and the term indexed by $k \ge 1$ has valuation at least k above the leading term. This is the normalization used below, not an arbitrary scalar multiple whose effect on the character is left unspecified.

Let P_0 extract the constant coefficient in each E-coordinate of E^m. The equation $f(z) = 0$ gives

$$z_0^* = -\frac{a_0}{z f'(z)}, \qquad u_0 = s(P_0) = \frac{a_0}{z^p} \in \mathcal{O}_E^\times.$$

For $C \in M_m(E)$ define the L-linear operator

$$\mathrm{Lift}_E(C) = \nu_0^{-1} C P_0. \tag{1}$$

Here the order of composition matters. In a wildly ramified extension, replacing P_0 by an E-linear commuting perturbation would destroy the intended derived contribution.

Proposition 7.2 (Exact corestriction and trace identities)**.** *For the complete interleaved chains on E^m, the operator in* (1) *satisfies*

$$\begin{aligned} s(\mathrm{Lift}_E(C)) &= C, \\ v_{\mathfrak{A}}(\mathrm{Lift}_E(C)) &= v_{\mathfrak{B}}(C) \quad (C \neq 0), \\ \mathrm{tr}_L(\mathrm{Lift}_E(C)) &= \mathrm{Tr}_{E/L}\big(\kappa^{-1} \mathrm{tr}_E(C)\big). \end{aligned}$$

Here $\mathfrak{B}$ is the complete E-order and $\mathfrak{A}$ the corresponding interleaved L-order. In particular $s(\mathfrak{P}^r) = \mathfrak{Q}^r$ at every integer grade.

Proof. The corestriction is an $M_m(E)$-bimodule map. Thus its value on $\nu_0^{-1} C P_0$ is $\nu_0^{-1} C \nu_0 = C$.

For the filtration assertion, take the basis $w_{mi+u} = z^i v_u$, with $0 \le i < p$ and $0 \le u < m$. Multiplication by the unit ν_0^{-1} preserves the E-chain valuation. A term $c_i z^i E_{uv}$ in an entry of $\nu_0^{-1} C$ contributes to a constant-coefficient column of the lift at grade

$$pm\, v_L(c_i) + mi + u - v.$$

This is precisely its E-chain grade. For a fixed entry, the valuations of different powers z^i are distinct modulo p, so a minimum cannot disappear by cancellation. The minimum over all entries proves equality of grades. The leading-term formula for $s(T_{ij})$ gives filtration preservation for s itself, while the explicit lift gives surjectivity on every grade.

Finally the L-trace of the rank-one operator yP_0 is its constant coefficient, namely $\mathrm{Tr}_{E/L}(z_0^* y)$. Apply this entrywise to the diagonal of the lift and use $z_0^* \nu_0^{-1} = \kappa^{-1}$. □

The proposition is not a bimodule splitting of a wildly ramified integral corestriction sequence. The lift is a chosen L-linear coefficient construction. In particular it need not commute with the embedded field. This distinction agrees with the warning following BK 1.4.10 and with BK 2.2.6 [25].

7.4 Recursion, native trace, and literal arithmetic fields

For $d = 1$, let $B(L, t_L; h_1)$ be multiplication by the Artin–Schreier element for q_1. At $d \ge 2$, use the largest q_d for the outer extension E/L, put $m = p^{d-1}$, and set

$$B(L, t_L; h_1, \dots, h_d) = \alpha I_m + \mathrm{Lift}_E\big(B(E, z; h_1, \dots, h_{d-1})\big). \tag{2}$$

Proposition 7.3 (Native trace zero)**.** *Every native matrix in* (2) *has ordinary trace zero over its own coefficient field. Every identity amplification, and every difference of two consecutive compatible native approximations, also has trace zero.*

Proof. The base trace is zero by Lemma 7.1. If C is the inner matrix, induction gives $\mathrm{tr}_E(C) = 0$. Proposition 7.2 then gives

$$\mathrm{tr}_L(B) = m\,\mathrm{Tr}_{E/L}(\alpha) + \mathrm{Tr}_{E/L}(\kappa^{-1}\,\mathrm{tr}_E(C)) = 0.$$

Amplification multiplies a trace by its multiplicity. Both traces in each consecutive difference already vanish at their native ranks, so the difference has trace zero as well. □

This proof is stronger than observing that a multiplicity divisible by p kills a trace. The latter observation says nothing about the native matrix before amplification. Native trace is the datum used when the next full character is constructed.

Theorem 7.4 (Literal parameter field, relative to BK refinement)**.** *Assume the standard simple-stratum definitions and BK Remark 2.2.7. The displayed rank-N matrix $\beta = B(F, t; h_1, \dots, h_d)$ generates a totally ramified maximal field K/F, normalizes the complete chain, and gives a simple stratum $[\mathfrak{A}, n, 0, \beta]$ with*

$$n = p^{d-1}q_d, \qquad k_0(\beta, \mathfrak{A}) = -q_1.$$

The literal approximation fields have degrees $p, p^2, \dots, p^d$, with defining levels

$$n_j = p^{d-j}q_{d-j+1} \qquad (1 \le j \le d).$$

Proof. The degree-p case is minimal because $p \nmid q_1$ and its residue extension is trivial. At the next step write $\gamma = \alpha I_m$, $b = \mathrm{Lift}_E(C)$ and

$$n = p^{d-1}q_d, \qquad R = p^{d-2}q_{d-1}.$$

The outer minimal parameter has $k_0(\gamma, \mathfrak{A}) = -n$. By induction, $E[C]$ is a maximal field in $M_m(E)$, its derived stratum at level R is pure, and $k_0(C, \mathfrak{B}) = -q_1$. The lift supplies $s(b) = C$ and $v_{\mathfrak{A}}(b) = -R$. Since $q_d > q_{d-1}$,

$$n > pR > 2R, \qquad -2R > -n - q_1.$$

These are the actual inequalities required by BK Remark 2.2.7. Its initial stratum is $[\mathfrak{A}, n, R, \gamma]$, its perturbation is $b \in \mathfrak{P}^{-R}$, and its derived field is the supplied maximal $E[C]$. The remark gives the field generated by the actual sum $\gamma + b$, not merely by a conjugate or by a separate model. Its ramification index is p^d, and its critical exponent is $\max(-n, -q_1) = -q_1$. Its valuation remains $-n$ because $n > R$.

For the approximation statement, truncate the same recursion after j native steps and amplify by $I_{p^{d-j}}$. All truncations use the same largest parameters and the same chosen uniformizers at shared outer steps. The lift commutes with identity amplification. Its exact grade preservation gives the level n_j for the corresponding consecutive correction. The shorter native field has degree p^j, completing the defining sequence. □

The derived field $E[C]$ is an input to the refinement theorem. It is not thereby identified with the final field $F[\gamma + b]$. In particular this proof makes no assertion that the approximation fields are nested as literal subfields of K. Likewise, trace zero alone does not prove separability of an arbitrary final field in characteristic p. The source result used here is the maximal, totally ramified field statement. The outer Artin–Schreier fields are separately known to be separable.

7.5 A compatible full character is more than a deep trace calculation

The extension step is elementary once it is placed at the correct maximal parameter field.

Lemma 7.5 (Extension with a fixed trivial subgroup)**.** *Let A be a compact abelian group, $B \subseteq A$ an open subgroup, and $C \subseteq A$ a closed subgroup. A smooth character $\chi : B \to \mathbb{C}^\times$ has a smooth extension to A that is trivial on C if and only if χ is trivial on $B \cap C$.*

Proof. Necessity follows by restriction. For sufficiency, define $\chi'(bc) = \chi(b)$ on BC. Intersection compatibility makes this well defined. Its kernel contains the open subgroup $\ker\chi$ of A, so it factors through a finite subgroup of $A/\ker\chi$. A character of a subgroup of a finite abelian group extends to the whole group: adjoin a generator whose multiple belongs to the subgroup and choose the required root of its prescribed character value in $\mathbb{C}^\times$, then repeat. The resulting character is smooth because it factors through that finite quotient. □

We apply the lemma with $A = U_K^1$, $B = U_K^{m+1}$ and $C = U_F^1$. It does not authorize an arbitrary extension through a nonmaximal matrix centralizer. The determinant condition and the normalization in BK 3.2.5 still have to hold.

For a minimal native parameter α at level q, put $m = \lfloor q/2 \rfloor$. The deep character is ψ_α on $U_{\mathfrak{A}}^{m+1}$. Its restriction to scalar units is trivial because $\mathrm{tr}_F(\alpha) = 0$. Its field-unit restriction therefore extends to U_E^1 trivially on U_F^1 by Lemma 7.5. The minimal simple-character definition, BK 3.2.1, joins these agreeing characters on

$$H^1(\alpha) = U_E^1 U_{\mathfrak{A}}^{m+1}.$$

At a nonminimal native step, the following order of operations is essential.

Transfer the whole preceding character. Let θ_j be the full native character already constructed at rank p^j, trivial on U_F^1. Amplify its parameter by multiplicity p and transfer θ_j to that actual parameter γ. BK 3.6.13(ii) and 3.6.14 preserve the whole beta-determinant unit character χ_j, not just its restriction to deep units. On a scalar $u \in U_F^1$, the transferred value is

$$\Theta_j(uI) = \chi_j\big(\det_{M_p(K_j)}(uI_p)\big) = \chi_j(u^p) = 1. \tag{3}$$

The determinant exponent p cannot be silently dropped. In the transfer theorem the centralizer orders may have different periods, but the full-character transfer uses $m_1 = m_2 = 0$. Thus both normalized indices $\lfloor m_i/e_i \rfloor$ are zero, exactly as required.

Apply the actual correction at the permitted depth. At the next native matrix $\beta = \gamma + \delta$, let $r = q_{d-j}$ and $m = \lfloor r/2 \rfloor$. The two strata are equivalent at level r, and their groups at depth $m+1$ agree. Since

$$\lfloor r/2 \rfloor \le m \le r-1, \qquad r < n,$$

BK 3.3.18 gives the character

$$\vartheta = \Theta_j|_{H^{m+1}(\beta)}\,\psi_\delta \quad \text{in} \quad \mathcal{C}(\mathfrak{A}, m, \beta).$$

The sign is fixed by the source bijection $\theta \mapsto \theta\psi_\delta^{-1}$. Proposition 7.3 makes ψ_δ trivial on scalar units in its domain, so ϑ is trivial on its intersection with U_F^1.

Extend at the new maximal native field. Now $K = F[\beta]$ is maximal at this native rank. Its centralizer is K itself, so the relevant determinant is the identity. The restriction of ϑ to U_K^{m+1} has the intersection compatibility required by Lemma 7.5. Choose its extension χ_K to U_K^1, still trivial on U_F^1. BK 3.2.5, used at full index zero, gives

$$\theta(kh) = \chi_K(k)\vartheta(h), \qquad k \in U_K^1, \quad h \in H^{m+1}(\beta). \tag{4}$$

Agreement on the intersection makes the formula well defined. The normalization and determinant hypotheses are supplied by the simple-character theorem. This proves full shallow-unit triviality inductively, not merely the cancellation of one leading coefficient.

7.6 Central completion without changing the canonical extension

Retain the chosen canonical beta extension κ_θ. The maximal residue factor is a character of $k_K^\times = \mathbb{F}_p^\times$. Where that factor is free to be chosen, it adjusts the compact central character to a prescribed smooth ω trivial on U_F^1. An independently fixed incompatible residue character is not replaced.

The group $\mathcal{N} = K^\times J$ has quotient $\mathcal{N}/J \simeq \mathbb{Z}$. Choose an extension Λ_0 of the fixed compact type. Twisting it by the character that takes a chosen generator to $s \in \mathbb{C}^\times$ changes its value on a scalar uniformizer by s^N. Choose s with

$$s^N = \omega(t)/\omega_{\Lambda_0}(t).$$

This does not assume that a uniformizer of K has literal Nth power t. Their quotient is a unit already handled by the fixed compact data. Once chosen, s becomes part of the candidate and is retained in every recovery argument.

BK 6.2.2 applies to the resulting maximal simple type and its actual extension. It yields

$$\pi = \mathfrak{c}\text{-Ind}_{\mathcal{N}}^{\mathrm{GL}_N(F)} \Lambda$$

as an irreducible supercuspidal representation. The current proof uses this classical theorem. It does not obtain supercuspidality from a finite Heisenberg dimension count.

7.7 Capacity and the precise packet admitted by this construction

The odd-prime profile formulas [11] are

$$t(\pi) = 1, \qquad \mathrm{Sw}(\pi) = p^{d-1}(p+1)h_d,$$
$$\dim_{\mathbb{F}_p}(J^1/H^1) = N-1, \qquad \dim \eta = p^{(N-1)/2},$$
$$a(\pi^\vee \times \pi) = N^2 - 1 + \sum_{i=1}^{d}(p^{2i} - p^{2i-2})h_i.$$

The conductor identities retain their separately pinned arithmetic inputs. They are not consequences of the trace lemmas alone.

There is a useful capacity argument that does not need every lower conductor. Put $n = p^{d-1}q_d$ and $M = n/2$. The leading minimal approximation has degree p. The concrete subgroup $U_{\mathfrak{A}}^M$ survives in $J^1(\beta)$ through all lower corrections. The recursive BK definitions keep it there because every lower critical level is less than $2M$. They also identify

$$H^M(\beta) = H^M(\gamma), \qquad S \simeq \mathfrak{P}^M/(\mathfrak{Q}^M + \mathfrak{P}^{M+1}).$$

The complete-chain grades have dimensions N and N/p, respectively. Therefore $\dim_{\mathbb{F}_p} S = (p-1)p^{d-1}$. On this subgroup, $\beta - \gamma \in \mathfrak{P}^{1-n}$ and commutators lie in $1 + \mathfrak{P}^n$, so the two character commutators agree. BK 3.4.1 makes the resulting form nondegenerate.

Now assume a one-vertex packet with trivial stabilizer and d actually supplied, labeled, full character extensions, each with a whole plane quotient. Their balanced central product has quotient dimension $2d$ and Heisenberg carrier dimension $p^d = N$. Since

$$(p-1)p^{d-1} \geq 2d,$$

a labeled symplectic embedding of the source into S exists. Proposition 3.2 attaches the full scalar extension, including its supplied character values. A line in the complementary Heisenberg factor gives the rank-N sector. The retained induction and idempotent argument recovers the same extended type and the same π [14].

For $p = 3$, the retained native cubic factors provide plane quotients. For $p > 3$, a native minimal rank-p quotient normally has dimension $p - 1$, not two. The plane admission hypothesis must be verified. Neither this capacity proof nor scalar-extension equivalence supplies a nontrivial arithmetic stabilizer action, independently fixed native markings, or literal placement at previously specified numerical depths.

For $p = 3$, $d = 2$ and $(h_1, h_2) = (1, 2)$, the rank is nine, the defining levels are $(24, 4)$, the Swan conductor is 24, the quotient dimension is eight, $\dim \eta = 81$, and the self-pair conductor is 232. The preserved sparse Laurent program has been replayed for this matrix. Its exact identities check the implementation, while the field and character conclusions above use the written proof and its classical inputs.

8 The weighted dyadic construction

Evidence and scope. The dyadic construction uses a different trace mechanism, not a formal substitution $p = 2$ in the odd-prime trace-zero proof. This section reconstructs the native trace, depth, and full-character arguments from [15–17]. The general parameter and simple-character steps remain relative to the classical inputs listed in Section 13.

8.1 Trace one can be invisible at the required depth

Lemma 8.1 (Depth-weighted trace visibility)**.** *Let ψ_F be an additive character trivial on $\mathfrak{p}_F$. Suppose $u \in U_F^s$, with $s \geq 1$, and a matrix δ satisfies*

$$v_F(\operatorname{tr}_F \delta) + s \geq 1.$$

Then $\psi_F((u-1)\operatorname{tr}_F \delta) = 1$.

Proof. The product has valuation at least $s + v_F(\operatorname{tr}_F \delta)$, so it belongs to $\mathfrak{p}_F$, where the character is trivial. □

For $F = \mathbb{F}_2((t))$, choose the additive character reading the constant coefficient. It is trivial on $t\mathcal{O}_F$ and takes the value -1 on 1. A trace of 1 is therefore invisible when multiplied by $u - 1$ for $u \in U_F^1$. A trace of t^{-1} is not: $u = 1 + t$ gives $(u-1)t^{-1} = 1$.

The Euler work motivated replacing literal zero by a weighted visibility condition. It is not an arithmetic input to this proof. The weights here are valuation depths. Multiplication by the integer two in characteristic two would erase the trace for an entirely different reason and would not prove the native assertion.

8.2 The literal quadratic step and exact trace recursion

Take odd integers $0 < h_1 < \cdots < h_d$, put $q_i = 3h_i$ and $N = 2^d$. At a recursive base L, use

$$\alpha^2 - \alpha = t_L^{-q}, \qquad a = (q+1)/2, \qquad z = t_L^a \alpha.$$

The valuation argument of Lemma 7.1 applies with $p = 2$, but the trace is now one. Direct substitution gives

$$f(z) = z^2 + t_L^a z + t_L = 0, \qquad \mathrm{Tr}_{E/L}(\alpha) = 1. \tag{5}$$

The polynomial is Eisenstein and its derivative t_L^a is nonzero. Its corestriction normalization is

$$\kappa = \frac{t_L^a}{z} = \alpha^{-1}, \qquad s(P_0) = \frac{t_L}{z^2}, \qquad \mathrm{Lift}_E(C) = \frac{z^2}{t_L} C P_0.$$

Consequently

$$s(\mathrm{Lift}_E(C)) = C, \qquad \mathrm{tr}_L(\mathrm{Lift}_E(C)) = \mathrm{Tr}_{E/L}(\alpha \,\mathrm{tr}_E(C)). \tag{6}$$

These are the same coefficient and dual-trace computations as Proposition 7.2, with the signs evaluated in characteristic two.

Proposition 8.2 (Three distinct trace statements)**.** *Every native dyadic matrix in the recursion has trace one over its own coefficient field. A preceding parameter amplified by multiplicity two has trace zero. The consecutive correction at the next native rank has trace one.*

Proof. At rank two the trace is $\mathrm{Tr}_{E/L}(\alpha) = 1$. At greater native rank write $B = \alpha I_m + \mathrm{Lift}_E(C)$, with $m = 2^{d-1}$ even. Inductively $\mathrm{tr}_E(C) = 1$. Thus

$$\mathrm{tr}_L(B) = m \,\mathrm{Tr}_{E/L}(\alpha) + \mathrm{Tr}_{E/L}(\alpha) = 0 + 1 = 1.$$

An identity amplification of multiplicity two has trace $2 \cdot 1 = 0$. Subtracting it from the next native matrix gives correction trace one. □

At full rank eight for $(h_1, h_2, h_3) = (1, 3, 5)$, the distinction is visible without any asymptotic notation:

Native rank	Native trace	Multiplicity in rank eight	Ambient trace	Defining level
2	1	4	0	60
4	1	2	0	18
8	1	1	1	3

The ordinary trace used for a new character is the native one. The table does not identify the three trace operations or infer a character from the zeros in its fourth column.

8.3 Refinement, fields, and a separability consequence

For a step of length $d \geq 2$, the outer level and the derived level are

$$n = 2^{d-1} q_d, \qquad R = 2^{d-2} q_{d-1}.$$

Since $q_d > q_{d-1}$, one has $n > 2R$. Hence $-2R > -n - q_1$, the exact inequality in BK Remark 2.2.7. The pure derived maximal field, filtered lift, and critical exponent are supplied as in the odd case. The

theorem gives the actual maximal totally ramified field generated by the displayed sum. It also gives the literal defining levels

$$n_j = 2^{d-j} q_{d-j+1}, \qquad k_0(\beta, \mathfrak{A}) = -q_1.$$

No equality between the final field and its derived centralizer model is used.

Proposition 8.3 (Trace one forces separability after maximality)**.** *Suppose the rank-N dyadic matrix β generates a field K/F of degree $N = 2^d$. Its native trace-one identity implies that K/F is separable.*

Proof. Since the generated field has degree equal to the matrix size, the minimal polynomial of β equals its characteristic polynomial. The coefficient of X^{N-1} is $-\operatorname{tr}_F(\beta) = 1$. Its derivative contains the nonzero term $(N-1)X^{N-2} = X^{N-2}$ in characteristic two. An irreducible polynomial with nonzero derivative is separable. □

This is an elementary consequence of two already explicit hypotheses. Trace one alone does not prove that the matrix generates a field of full degree. The maximality step still uses the refinement theorem. In particular this observation is not an independent replacement for the simple-stratum construction.

8.4 The floor in the correction depth is essential

Every native critical level $r = q_i$ is odd. Put $m = (r-1)/2$, so $r = 2m+1$. For $x, y \in \mathfrak{P}^{m+1}$ and $\delta \in \mathfrak{P}^{-r}$,

$$\delta x y \in \mathfrak{P}^{2(m+1)-r} = \mathfrak{P}. \tag{7}$$

The level-one matrix additive character is trivial there. Thus the error between evaluating the additive expression at $(1+x)(1+y)$ and multiplying its values at $1+x$ and $1+y$ vanishes. This is the depth calculation behind the correction character on the relevant congruence subgroup.

Replacing $m+1$ by m would give $2m - r = -1$, not a positive depth. Writing "approximately half the level" is therefore inadequate here. The endpoint $m = \lfloor r/2 \rfloor$ lies in the closed interval permitted by BK 3.3.18, and the character is evaluated on H^{m+1}.

8.5 The full shallow-character induction

For a minimal rank-two parameter, the deep character ψ_α is trivial on its scalar intersection with U_F^1, since $\operatorname{tr}_F(\alpha) = 1$ and Lemma 8.1 applies. Extend its field-unit character to U_E^1 trivially on U_F^1 using Lemma 7.5. The minimal BK definition then gives the full character on H^1.

Assume a full native character θ_j, trivial on U_F^1, has already been constructed. Transfer it to the multiplicity-two parameter γ. Preservation of the beta-determinant character gives $\Theta_j(uI) = \chi_j(u^2) = 1$ on scalar units. This uses the entire native unit character. It does not transfer a guessed deep approximation and then assume its shallow values.

At the next actual native parameter $\beta = \gamma + \delta$, Proposition 8.2 gives $\operatorname{tr}_F \delta = 1$. Put $r = 2m+1$ as above. The strata and depth range meet BK 3.3.18, so

$$\vartheta = \Theta_j|_{H^{m+1}(\beta)}\, \psi_\delta$$

is a simple character at that depth. Lemma 8.1 gives $\vartheta = 1$ on $U_F^1 \cap H^{m+1}(\beta)$. The actual final field at this native step is maximal. Its forced unit character on U_K^{m+1} and the trivial character on U_F^1 therefore agree on the intersection. Lemma 7.5 extends them to U_K^1, and BK 3.2.5 yields the full formula (4).

The residue field is $\mathbb{F}_2$, so $k_K^\times$ is trivial and $J = J^1$. The compact type is the Heisenberg representation η, with no nontrivial residue character to choose. A prescribed central character trivial on U_F^1 is unramified here. Its remaining value is fixed by the noncompact extension exactly as in the odd case. No division by two takes place in the local coefficient field.

8.6 What the dyadic parameter theorem does and does not realize

The construction supplies a parameter and full character for the stated arbitrary lengths, with

$$t(\pi) = 1, \qquad \mathrm{Sw}(\pi) = 3 \cdot 2^{d-1} h_d,$$
$$\dim_{\mathbb{F}_2}(J^1/H^1) = N - 2, \qquad \dim \eta = 2^{N/2-1},$$
$$a(\pi^\vee \times \pi) = N^2 - 1 + \sum_{i=1}^{d} 3 \cdot 4^{i-1} h_i.$$

The precise conductor inputs remain identified separately. A full source packet of d planes has quotient dimension $2d$ and carrier dimension 2^d. The original maximal-type quotient has enough capacity only for $d \geq 3$. Thus even a valid parameter and full-character construction does not settle attachment at ranks two and four.

Section 9 retains that genuine original-carrier obstruction. Section 10 changes the actual inducing group and proves recovery of the unchanged candidate. Neither operation weakens the dimension theorem for the original quotient. The rank-eight sparse Laurent construction has been replayed, including native and amplified traces and the grades $(60, 18, 3)$. Those finite identities are checks of the displayed matrix algorithm, not substitutes for the full arithmetic proof.

9 Capacity, markings and the low-rank obstruction

9.1 The two dimensions that must not be conflated

For a finite Heisenberg quotient W of dimension $2r$ over $\mathbb{F}_p$, the irreducible carrier with its fixed nontrivial central character has dimension p^r. Quotient dimension and representation dimension measure different objects. A rank-N Weil source assembled from d planes has quotient dimension $2d$ and carrier dimension $N = p^d$. The type carrier can be substantially larger than N.

In the dyadic full tower, the graded contribution at stage j is

$$(2^{d-j+1} - 2^{d-j})\, \mathbf{1}_{n_j \text{ even}}.$$

The last level is odd and the earlier levels are even. Thus

$$\dim W = \sum_{j=1}^{d-1} 2^{d-j} = 2^d - 2.$$

The source quotient fits in this maximal-type quotient precisely when $2^d - 2 \geq 2d$, or $d \geq 3$.

d	N	Host dim W	Host dim η	Source carrier
1	2	0	1	2
2	4	2	2	4
3	8	6	8	8
4	16	14	128	16

These exact values explain why the original rank-two and rank-four maximal-type carriers cannot contain the whole source packet. The later enlargement does not change these numbers. It changes the arithmetic assembly carrier.

9.2 Transported markings versus a fixed native refinement

For $d \geq 3$, the retained scalar-pushout attachment embeds the source symplectic quotient and carries the entire declared marked finite extension. It preserves the squares of those transported lifts and the corresponding refinement data. If an independent native quadratic form q_W is also prescribed, that is an additional compatibility problem [19].

At $d = 3$, source and host have equal dimension six. A native quadratic embedding must be an isometry, so the Arf invariants must agree. For $d \geq 4$, there is at least one complementary hyperbolic-size plane available at the level of quadratic-space classification. When the embedding can be chosen, a complementary Arf contribution permits either source type. For an already fixed embedding f, however, the actual requirement is pointwise:

$$q_W(f(v)) = q_{\mathrm{src}}(v) \quad \text{for every } v.$$

A dimension count or an equality of total Arf invariants does not establish this identity for that fixed map. These assertions about native embeddings remain distinct from preserving the source marking under a constructed extension map.

9.3 Why profile adjustments did not resolve the capacity issue

The capacity record considers changes of parameters and scalar twists while retaining the prescribed numerical profile. Its low-rank negative conclusions concern the original maximal-simple-character carriers in that profile. They do not show that no larger compact subgroup of the same $\mathrm{GL}_N(F)$ can carry the packet. This distinction makes the next construction possible without contradicting the earlier obstruction.

10 Enlarged arithmetic carriers recover the same representation

Evidence and scope. This section expands the rank-two and rank-four ordinary constructions [20–22]. The groups are actual subgroups of the same general linear group. Their character and filtration inputs are retained explicitly. A general finite-index recovery proposition then keeps the full inducing representation, including its noncompact extension. No new candidate is selected through the reference correspondence.

10.1 Rank two: enlarge the group rather than the original quotient

Let $F = \mathbb{F}_2((t))$, $G = \mathrm{GL}_2(F)$ and $[\mathfrak{A}, q, 0, \beta]$ be the actual minimal dyadic stratum, with $q = 2m + 1 \geq 3$ and $K = F[\beta]$. Its complete period-two order has

$$J = J^1 = H^1 = U_K^1 U_{\mathfrak{A}}^{m+1}.$$

The full simple character θ is one-dimensional on J. Fix its actual extension Λ on $\mathcal{N} = K^\times J$ and the candidate $\pi = \mathfrak{c}\text{-}\mathrm{Ind}_{\mathcal{N}}^{G} \Lambda$. The central character and the value of the chosen noncompact generator are fixed data throughout the argument.

Set

$$P = U_K^1 U_{\mathfrak{A}}^m. \tag{8}$$

The field normalizes the complete order, so this product is a compact open group. The filtration commutator estimates give $[P, J] \subseteq U_{\mathfrak{A}}^{m+1}$, and hence $J \triangleleft P$. On the grade $\mathfrak{P}^m/\mathfrak{P}^{m+1}$, the ambient order contributes dimension two over $\mathbb{F}_2$ and the field line contributes dimension one. Thus

$$[P : J] = 2, \qquad P \cap \mathcal{N} = J.$$

For the intersection, an element $kj \in P$ with $k \in K^\times$ has $k \in U_{\mathfrak{A}}^1$, hence $k \in U_K^1 \subseteq J$.

The maximal minimal-character intertwining formula gives $I_G(\theta) = \mathcal{N}$. Choose $g \in P \setminus J$. It does not intertwine θ, so $\theta^g \neq \theta$. Consequently

$$\lambda(j) = \theta(gjg^{-1})/\theta(j)$$

is nontrivial. Its values lie in $\{1, -1\}$ because its commutators lie in the deep subgroup where θ is the additive character of a characteristic-two matrix trace. The entire shallow character θ need not have order two.

Since $g^2 \in J$ and θ is a character, $\lambda^g = \lambda^{-1} = \lambda$. Thus $H_0 = \ker \lambda$ is normal in P, with $[P : H_0] = 4$. Finite-index induction gives

$$\Xi = \mathrm{Ind}_J^P \theta,$$

an irreducible two-dimensional representation. Its restriction to J is the sum of the two distinct characters θ and θ^g, and g exchanges their lines.

10.2 The rank-two quotient and the retained character

In a two-coset model, the represented action has the form

$$j \longmapsto \theta(j) \begin{pmatrix} 1 & 0 \\ 0 & \lambda(j) \end{pmatrix}, \qquad g \longmapsto X = \begin{pmatrix} 0 & 1 \\ a & 0 \end{pmatrix}, \qquad a = \theta(g^2) \neq 0.$$

For an element with $\lambda(j) = -1$, the nonscalar part is $Z = \mathrm{diag}(1, -1)$. One has

$$X^2 = aI, \qquad Z^2 = I, \qquad XZX^{-1} = -Z.$$

The scalar preimage is exactly H_0, so $P/H_0 \simeq \mathbb{F}_2^2$ is the actual projective quotient. Its commutator is nondegenerate. This is more information than the dimension of Ξ alone.

The scalar a remains the value of the original character. It need not equal one, and the matrices do not automatically match an independently prescribed native square marking. A source scalar-pushout identification can carry its marked finite extension, but equality with a fixed native quadratic refinement requires the separate compatibility condition from Section 9.

The carrier occurs in $\pi|_P$ in the functions supported on $\mathcal{N}P$. Explicitly, a function f in $\mathrm{Ind}_J^P \theta$ extends there by $f(np) = \Lambda(n)f(p)$. The equality $\mathcal{N} \cap P = J$ proves that this is well defined. The construction uses the same Λ and does not change its noncompact root [21].

10.3 Rank four: the actual final-field approximation

Use the rank-four dyadic constructor with odd $h_1 < h_2$. Put

$$q_i = 3h_i, \qquad n = 2q_2 = 2a, \qquad r = q_1 = 2m + 1, \qquad m \geq 1.$$

Then $a > r$ and $n > 2r$. Let γ be the degree-two outer minimal approximation, $E = F[\gamma]$, $B = \mathrm{End}_E(F^4) = M_2(E)$, and $\mathfrak{B} = \mathfrak{A} \cap B$. Its radical is $\mathfrak{Q}$. Write $\beta = \gamma + \delta$ and $c = s(\delta)$. The derived parameter c is minimal of degree two over E and has level r. The actual final field is $K = F[\beta]$ of degree four. We do not assume $E \subseteq K$.

The recursive group definitions give

$$H = H^1(\beta) = U_K^1 U_{\mathfrak{B}}^{m+1} U_{\mathfrak{A}}^{a+1},$$
$$J = J^1(\beta) = U_K^1 U_{\mathfrak{B}}^{m+1} U_{\mathfrak{A}}^{a}.$$

Thus J/H has dimension two and its Heisenberg representation η has dimension two. The existing θ on H is not replaced.

Lemma 10.1 (Approximation by the outer centralizer)**.** *For $j \geq 1$ and $k \in K \cap \mathfrak{P}^j$, the minimal corestriction exact sequence gives*

$$k = b + e, \qquad b \in \mathfrak{Q}^j, \qquad e \in \mathfrak{P}^{j+n-r}. \tag{9}$$

These leading $\mathfrak{B}$-grades form precisely the centralizing line of the leading derived parameter c.

Proof. Since k commutes with β, one has $[\gamma, k] = -[\delta, k] \in \mathfrak{P}^{j-r}$. The exact commutator sequence for the minimal parameter γ, BK 1.4.10, removes this commutator by an error in grade $j + n - r$. The remainder centralizes γ and lies in B, proving (9).

Apply s to $[\beta, k] = 0$. The corestriction kills $[\gamma, e]$, and its bimodule property gives $s([\delta, b]) = [c, b]$. The remaining error has grade at least $j + n - 2r$, so

$$[c, b] \in \mathfrak{Q}^{j+n-2r}.$$

Because $n > 2r$, the leading grade of b centralizes that of c. In the complete period-two E-chain, choose the shift R with $R^2 = zI$. The leading parameter is R^{-r}, with r odd. Its centralizer in grade j is the line spanned by R^j. The actual maximal totally ramified field K contributes a one-dimensional grade. Estimate (9) is injective on that grade, so its image is exactly the line. □

This is where a literal arithmetic attachment is needed. A dimension comparison between two abstract graded spaces would not identify the line belonging to the actual final field. The use of the minimal exact sequence is recorded separately in the input audit.

10.4 An actual index-two enlargement and a nontrivial character orbit

Set $P = JU_{\mathfrak{B}}^m$. The field approximation implies

$$[U_{\mathfrak{B}}^m, U_K^1] \subseteq U_{\mathfrak{B}}^{m+1} U_{\mathfrak{A}}^{m+n-r+1} \subseteq H.$$

Together with the ordinary filtration estimates, this proves that P is a compact open group and $J \triangleleft P$. The grade of $U_{\mathfrak{B}}^m$ has dimension two. Its intersection with J has exactly the one-dimensional centralizing

line of Lemma 10.1. The converse containment uses $m + n - r > a$, which puts the approximation error inside $U^a_{\mathfrak{A}} \subseteq J$. Hence $[P : J] = 2$.

Take $D = \operatorname{diag}(1, 0)$ and $u = 1 + R^m D$. Its leading grade is outside that line, so it represents the nontrivial coset. Also $u^2 \in U^{2m}_{\mathfrak{B}} \subseteq H$. The character

$$\lambda(h) = \theta(uhu^{-1})/\theta(h), \qquad h \in H,$$

is J-invariant and has values in $\{1, -1\}$. To see the value rather than guess it, use the defining-sequence formula $\theta = \theta_\gamma \psi_\delta$ on the relevant commutator domain. The outer character is invariant under $U^1_{\mathfrak{B}}$, leaving the actual correction ψ_δ.

Lemma 10.2 (The leading dyadic commutator)**.** *Let $R = \left(\begin{smallmatrix} 0 & z \\ 1 & 0 \end{smallmatrix}\right)$ over $\mathbb{F}_2[z, z^{-1}]$. For $m \geq 1$, $X = R^m \operatorname{diag}(A, B)$ and $Y = R^{m+1} \operatorname{diag}(C, D)$ with $A, B, C, D \in \mathbb{F}_2$, one has*

$$\operatorname{tr}\big(R^{-2m-1}(XY - YX)\big) = (A + B)(C + D).$$

Proof. If $m = 2k$, factor z^k from both matrices and use $R^2 = zI$. After multiplication by R^{-2m-1}, the trace reduces to $D(A + B) + C(A + B)$. If $m = 2k + 1$, the roles of the two diagonal entries are exchanged and the same expression results. In characteristic two the subtraction is addition. Both cases give the displayed product. □

Choose $X = R^m D$ and $Y = R^{m+1} D$. The two sums in the lemma are one. The group-commutator remainder has $\mathfrak{B}$-grade at least $3m + 1$. Multiplication by the leading parameter of grade $-2m - 1$ leaves grade at least $m \geq 1$, so the remainder is invisible. The difference between c and R^{-r} also contributes only positive grade. The normalized character satisfies

$$\psi_E(1) = \psi_F(\operatorname{Tr}_{E/F} \alpha) = \psi_F(1) = -1.$$

Thus $\lambda(1 + Y) = -1$. In particular $\theta^u \neq \theta$ and $\eta^u \not\simeq \eta$. Finite-index induction gives the irreducible four-dimensional carrier $\Xi = \operatorname{Ind}_J^P \eta$.

10.5 The rank-four scalar preimage and quotient

Put $H_0 = \ker \lambda$ inside H. In the two-coset model, $h \in H$ acts by

$$\operatorname{diag}\big(\theta(h) I_2, \theta^u(h) I_2\big).$$

Since $u^2 \in J$ and λ has order two, $\lambda^u = \lambda$, so $H_0 \triangleleft P$. The common character $\theta_0 = \theta|_{H_0}$ is P-invariant.

The depth estimates give $\lambda = 1$ on $U^{a+1}_{\mathfrak{A}}$ because $a + 1 + m - r = a - m > 0$. Write $J = HU^a_{\mathfrak{A}}$. The character λ kills $[H, H]$ and $[H, U^a_{\mathfrak{A}}]$ by J-invariance. It kills $[U^a_{\mathfrak{A}}, U^a_{\mathfrak{A}}]$ and squares from that subgroup because they lie in $U^{2a}_{\mathfrak{A}}$. It kills squares from H because it has order two. Finally $[u, U^a_{\mathfrak{A}}] \subseteq U^{a+m}_{\mathfrak{A}}$ and $\lambda([u, H]) = 1$, while $\lambda(u^2) = 1$ since u commutes with its own square. Consequently

$$[P, P] \subseteq H_0, \qquad g^2 \in H_0 \ (g \in P), \qquad [P : H_0] = 2 \cdot 4 \cdot 2 = 16.$$

An element of J acting scalarly on Ξ must act scalarly on its first η block. Nondegeneracy of the original J/H forces it into H, and then scalar equality of the two blocks forces it into H_0. An element outside J exchanges the two blocks and cannot act scalarly. Therefore H_0 is the entire scalar preimage.

It follows that $P/H_0 \simeq \mathbb{F}_2^4$ with actual commutator form $(gH_0, kH_0) \mapsto \theta_0([g,k])$. If a class belongs to the radical, its represented operator commutes with the irreducible image and is scalar by Schur's lemma. It therefore belongs to H_0. The form is nondegenerate.

The finite scalar center is $\theta_0(H_0)$, which can have order larger than two. The result is an actual Heisenberg central extension with a four-dimensional binary projective quotient. It is not automatically a literal extraspecial group of order 32, nor an identification with an independently marked native quadratic space [22].

10.6 The final field normalizes the enlarged rank-four group

To obtain an inducing datum rather than only compact occurrence, one must prove $kPk^{-1} = P$ for every $k \in K^\times$. Units of K lie in $U_K^1 \subseteq J$ because the residue field is $\mathbb{F}_2$, so they normalize P. Let k be a uniformizer of K. Lemma 10.1 at grade one gives

$$k = b + e, \qquad b = Rv \quad (v \in 1 + \mathfrak{Q}), \qquad e \in \mathfrak{P}^{n-r+1}.$$

Hence $k = b(1+\varepsilon)$ with $\varepsilon \in \mathfrak{P}^{n-r}$. Conjugation by b preserves the two chain filtrations. For $w \in U_{\mathfrak{B}}^m$, the error from $1+\varepsilon$ lies in

$$U_{\mathfrak{A}}^{n-r+m} \subseteq U_{\mathfrak{A}}^a \subseteq J, \qquad n - r + m = 2a - m - 1 > a.$$

Thus $kU_{\mathfrak{B}}^m k^{-1} \subseteq P$. Also $kJk^{-1} = J$ by normalization of the actual simple-character groups. The subgroup kPk^{-1} contains J with the same index two as P, so it equals P. This proof does not insert the unjustified assumption $E \subseteq K$.

10.7 One finite-index recovery theorem for both low ranks

Proposition 10.3 (Enlargement retaining the full inducing representation)**.** *Let $G = \mathrm{GL}_n(F)$, let K/F be a maximal field in $M_n(F)$, and suppose $J \triangleleft P$ are compact open subgroups with $[P : J] = 2$. Assume $K^\times$ normalizes both groups, $K^\times \cap P \subseteq J$, and*

$$\mathcal{N} = K^\times J, \qquad \pi = \mathfrak{c}\text{-}\mathrm{Ind}_{\mathcal{N}}^G \Lambda, \qquad \Lambda|_J = \eta$$

with η irreducible. If $\eta^u \not\simeq \eta$ for $u \in P \setminus J$, then $M = K^\times P$ is open and compact modulo the center,

$$\mathcal{N} \triangleleft M, \quad [M : \mathcal{N}] = 2, \quad \Sigma = \mathrm{Ind}_{\mathcal{N}}^M \Lambda \text{ is irreducible},$$

and

$$\Sigma|_P \simeq \mathrm{Ind}_J^P \eta, \qquad \mathfrak{c}\text{-}\mathrm{Ind}_M^G \Sigma \simeq \pi. \tag{10}$$

The central character and the noncompact extension in Λ are unchanged.

Proof. Normalization makes M a group. The action of $K^\times$ on P/J is trivial because that quotient has order two. Thus $[K^\times, P] \subseteq J$, proving normality of $\mathcal{N}$ in M. The intersection hypothesis gives index two. The group is open because it contains J, and compact modulo $F^\times$ because both $K^\times/F^\times$ and P are compact.

The restriction of Σ to $\mathcal{N}$ is the sum of Λ and Λ^u. They are irreducible and nonisomorphic, since their restrictions to J already differ. Any $\mathcal{N}$-subrepresentation is a sum of these distinct irreducible summands. The element u exchanges them, leaving no nonzero proper M-stable subrepresentation. This proves irreducibility without applying a finite-group semisimplicity assertion to an arbitrary noncompact group.

There is a single $\mathcal{N}$–P double coset in M and $\mathcal{N} \cap P = J$, giving the first isomorphism in (10). Finite-index induction from $\mathcal{N}$ to M is compact induction. Induction in stages proves the second. No character in Λ has been rescaled. □

For rank two, $K^\times$ normalizes $P = U_K^1 U_{\mathfrak{A}}^m$ directly. For rank four the preceding subsection provides the missing arithmetic normalization. The resulting Σ has dimension two or four, respectively, and induces to the original candidate. This consolidates the rank-four induction note and makes the same finite-index argument explicit for rank two.

The word "same" in this result is literal. It refers to the previously fixed Λ, not merely to an unramified-twist class or to a representation with matching conductor. It supplies the enlarged actual inducing data required by this restricted realization contract. It does not prove that their source packet has a nontrivial prescribed stabilizer action or the full same-candidate H5 factors.

10.8 A conditional induction statement with actual groups

Let $G = \mathrm{GL}_n(F)$, and fix

$$\pi = \mathfrak{c}\text{-}\mathrm{Ind}_{\mathcal{N}}^{G} \Lambda, \qquad \eta = \Lambda|_J,$$

where J is compact open and normal in $\mathcal{N}$, η is irreducible and $\mathcal{N}/J$ is infinite cyclic. Suppose P is an actual compact open subgroup containing J, with $P \cap \mathcal{N} = J$, and put $\Xi = \mathrm{Ind}_J^P \eta$.

Proposition 10.4 (Occurrence and compact recovery)**.** *The subspace of $\pi|_P$ supported on $\mathcal{N}P$ is isomorphic to Ξ. Moreover,*

$$\mathfrak{c}\text{-}\mathrm{Ind}_P^G \Xi \simeq \mathfrak{c}\text{-}\mathrm{Ind}_J^G \eta.$$

If the actual intertwining set of η in G is $\mathcal{N}$, then Ξ is irreducible.

Proof. Restriction to the support $\mathcal{N}P$ identifies its functions with induction from $\mathcal{N} \cap P = J$, using $\Lambda|_J = \eta$. The displayed isomorphism is induction in stages. For irreducibility, the finite Mackey endomorphism formula has no contribution from nontrivial double cosets in P, since every intertwiner lies in $\mathcal{N}$ and $P \cap \mathcal{N} = J$. Its endomorphism space is therefore $\mathrm{End}_J(\eta) = \mathbb{C}$. Finite-dimensional smooth representations of compact P factor through finite groups and are semisimple over $\mathbb{C}$. A semisimple representation with scalar endomorphisms is irreducible. □

The original Λ, its central character and its noncompact root are unchanged. The equality above does not yet recover the chosen π from the compact carrier alone. It recovers the larger compactly induced representation from J.

10.9 The noncompact relation is a quotient before central normalization

Choose $u \in \mathcal{N}$ generating $\mathcal{N}/J$. Use left-covariant functions $f(jg) = \eta(j)f(g)$ with right translation by G, and write $X = \mathfrak{c}\text{-}\mathrm{Ind}_J^G \eta$. Define

$$(Tf)(g) = \Lambda(u)^{-1} f(ug).$$

Normality and the actual intertwining identity imply that T preserves covariance. The normalization remembers the chosen noncompact extension.

Proposition 10.5 (Recovery of the fixed noncompact extension)**.** *There is a natural isomorphism*

$$X/(T-1)X \simeq \pi.$$

Proof. Define

$$(Qf)(g) = \sum_{k\in\mathbb{Z}} \Lambda(u)^{-k} f(u^k g).$$

The sum is finite at each argument because f has compact support modulo J, while the cosets indexed by k are discrete. Reindexing shows $Qf(ug) = \Lambda(u)Qf(g)$, and the J-covariance gives full $\mathcal{N}$-covariance. The support is compact modulo $\mathcal{N}$, so $Qf \in \pi$. Choosing representatives of the finitely many relevant $\mathcal{N}$-cosets proves surjectivity.

On each $\mathcal{N}/J$-orbit, the kernel statement is the augmentation identity for finitely supported sequences $(v_k)_{k\in\mathbb{Z}}$. If $\sum_k v_k = 0$, the partial sums are finitely supported and write (v_k) as consecutive differences. These differences are exactly $(T-1)X$ in the normalized coordinates. Finally $Q(Tf) = Qf$, so the induced quotient map is the required isomorphism. □

There need not be a splitting $\pi \to X$ before fixing the center. In this general-linear setting, determinant valuations of the support of a nonzero vector in X form a finite set. A scalar uniformizer shifts that set by n, so such a vector cannot be an eigenvector for the scalar uniformizer. Every vector of π has the fixed central character. Thus a nonzero G-map $\pi \to X$ is impossible in this setting [20].

10.10 After retaining the center, a finite projector is available

Let $Z = F^\times I_n$, $J_Z = ZJ$, and suppose $\mathcal{N}/J_Z$ is cyclic of order ℓ. Retain $\Lambda_0 = \Lambda|_{J_Z}$. On $Y = \mathrm{c\text{-}Ind}_{J_Z}^{G} \Lambda_0$, the same normalized shift satisfies $T^\ell = 1$. Hence

$$e_\Lambda = \frac{1}{\ell} \sum_{j=0}^{\ell-1} T^j$$

is an idempotent. Its image is the space of $\mathcal{N}$-covariant functions, which is precisely the fixed π. Division by ℓ takes place in $\mathbb{C}$, not in the residue field. Choosing another noncompact root changes T and can select another extension.

This projector is a finite Hecke/induction projector after central normalization. It is not the older image-algebra projector onto a sector of the original η. The distinction is part of the result, not a cosmetic change of notation. The low-rank enlargement repairs a particular assembly failure while preserving the obstruction for the original carrier.

11 The native stabilizer obstruction and a scalar positive case

11.1 The obstruction belongs to a specified mechanism

For the totally ramified degree-$n = p^r$ host in the stated residue-rank-one setting, the retained native character-preserving normalizer calculation gives

$$\mathcal{N} = K^\times J^1, \qquad \mathcal{N}/(F^\times J^1) \simeq \mathbb{Z}/n\mathbb{Z}.$$

This is an arithmetic input of the source, not a fact about every possible host. Its immediate finite consequence is decisive [23].

Proposition 11.1 (Prime-to-p inner actions vanish in this host)**.** *If a finite group C of order prime to p acts on the host symplectic quotient through the displayed native-normalizer quotient, its action is trivial. If the action on the full scalar extension fixes the scalar center pointwise and lies over that trivial quotient action, that action is also trivial.*

Proof. The image in the cyclic p-group has order dividing both $|C|$ and p^r, so it is trivial. The remaining automorphisms over quotient and center form $\mathrm{Hom}(W, \mathbb{C}^\times)$, a finite p-group. The same coprimality argument makes the remaining image trivial. □

Honest operators on an irreducible Heisenberg carrier are then scalar by Schur's lemma. Matching the source therefore requires the source extension action to be trivial and its scalar character to equal the host character. A general source with a nontrivial prescribed symplectic action cannot be forced into this native mechanism by adjusting a scalar cochain.

The source record gives a nontrivial prime-to-p symplectic action on separate admitted break targets. That is consistent with the original H2 input and inconsistent with the stronger claim that the current totally ramified host can implement every such action by its own inner normalizer. The obstruction refutes the stronger proposed mechanism, not the original finite hypothesis and not Conjecture 66.

11.2 A genuine scalar inertia phase can descend from a cover

Note [24] also proves a restricted positive extension beyond $C = 1$. Assume C is prime to p, acts trivially on the source scalar extension, and its prescribed honest operators on the entire rank-N Weil carrier are $\chi(c) I_N$ in the actual inertia image. Fix the same-source determinant central character under the specified reciprocity convention.

Let the residue cardinality be q and write $B = \omega(\mathrm{Teich}(k^\times))$. The stated actual-inertia and determinant hypotheses give $\chi(C)^N \subseteq B$. Since N is a power of p and the order of $\chi(C)$ is prime to p, taking N th powers permutes $\chi(C)$, so $\chi(C) \subseteq B$. Choose $k_0 N \equiv 1 \pmod{q-1}$ and set

$$\delta(a) = \omega(\mathrm{Teich}(a))^{k_0}, \qquad \widetilde{C} = \{(c, a) \in C \times k^\times : \chi(c) = \delta(a)\}.$$

The projection $\widetilde{C} \to C$ is onto. The central matrix $\mathrm{Teich}(a^{k_0}) I_N$ acts by $\chi(c)$. Its kernel acts trivially on the entire candidate, so the represented action descends to C [24].

A direct matrix homomorphism from C itself need not exist. An arbitrary scalar phase on a multiplicity subspace is not the required actual-inertia phase on the whole source carrier. The argument also does not rescale a fixed canonical beta extension to manufacture a desired character. These qualifications are essential to the positive result.

12 What has been gained and what remains

12.1 The restricted positive result is substantial

The recovered ordinary constructions now address arbitrary odd-prime and dyadic lengths in their stated families. They build actual parameter fields, control native traces or trace visibility, extend full characters through the required shallow intersections, and retain central and noncompact completion choices. For the one-vertex trivial-stabilizer scalar-pushout contract, the dyadic low-rank repair allows an actual enlarged assembly carrier where the original maximal-type quotient is too small. These are mathematical results with explicit hypotheses, not merely numerical suggestions.

At the same time, these families do not supply an actual witness for every d_ρ admitted by the general conjecture. The same-candidate H5 comparison also remains a separate conditional step. These distinctions keep the restricted results separate from the general realization and comparison obligations.

12.2 The remaining obligations are different kinds of problems

Admission and canonical source data. Broader source classes need intrinsic envelopes, the declared equivariant bridges, and the requisite odd or dyadic phase and marking data. Conjecture 66, as stated, starts after H1–H4 have been supplied. Extending the supply of those inputs is related work, but it is not the same quantified task.

A common arithmetic host. Separate breakwise targets do not yet give one actual inducing system of the total rank. The direct families establish some hosts. Their existence does not prove that the host can realize every admitted source action, transport graph or numerical layer placement.

Nontrivial stabilizer actions. The native inner-normalizer obstruction forces a change of host or a different arithmetic action mechanism for genuinely nontrivial prime-to-p symplectic actions. Abstract cohomological lifting has already removed some finite scalar defects. Repeating that lifting does not solve the missing arithmetic attachment.

Independent native markings and fixed embeddings. Transporting a marked source extension is not the same as matching a separately specified native quadratic refinement. A proposed match must respect the actual fixed embedding, rather than merely an abstract isometry class.

General transport diagrams. A one-vertex packet has no nontrivial global transport constraint. Finite holonomy theorems compare supplied source and target systems. A realization theorem for arbitrary admitted diagrams must construct an arithmetic system and show that its maps recover those particular transports.

Same-candidate factors. H5 remains about the actual candidate that realizes the packet. Conductor, torsion and self-pair identities are useful tests, but they do not supply the full characterizing family. A program seeking an independent factor comparison must construct that proof rather than invoke the reference correspondence through an implicit definition.

12.3 The route through the remaining constructions

The preceding sections develop parameter recursion, normalized corestriction, native traces, ordered full characters, and both low-rank enlarged carriers. They retain actual finite-index inducing data alongside compact quotients and central projectors. Section 13 states the principal classical inputs and their audit boundaries.

Sections 14–21 compare the earlier monomial, Clifford-descent, and binder constructions with the direct constructor. They retain both failed mechanisms and surviving positive results, distinguishing source-defined arithmetic induction from direct construction. Sections 22–30 examine the conductor and marking questions under their stated hypotheses.

The research atlas illustrates these constructions, obstructions, and evidence boundaries. The formal companion records the associated verification scope. Together they distinguish the available restricted results from the general realization and same-candidate comparison obligations.

13 Classical inputs and the proof boundary

13.1 What was read directly

The primary reference for the direct construction is Bushnell and Kutzko, The Admissible Dual of GL(N) via Compact Open Subgroups [25]. The retained book and its extracted text were available, and the cited passages were checked against page images. The PDF page number is twelve greater than the printed page number for the passages listed below. This locator offset is recorded rather than guessed from a search hit.

The audit establishes which published statement is used and how the current construction meets its stated interface. It does not turn a citation into a Lean proof or reprove the entire classical theory. Nor does it certify unrelated source-dependent conductor formulas simply because they occur in the same archive.

Input	Printed pages	Application and restriction
BK 1.4.10	38	Minimal commutator exactness used for the actual rank-four field approximation. It is not an integral bimodule splitting in wild ramification.
BK 2.2.6–7	55	Noncommuting refinement. A pure maximal derived field and the strict inequality $-2R > k_0(\gamma) + k_0(C)$ are supplied. The conclusion concerns the literal final sum.
BK 3.1.7–10	90–94	Recursive H and J groups, with distinct floor and ceiling thresholds. These identify the actual congruence subgroups and surviving high layer.
BK 3.2.1–5	98–100	Minimal and nonminimal character definitions, then extension from depth $\lfloor r/2 \rfloor + 1$. The determinant and normalization conditions are retained.
BK 3.3.2	105	Full-character intertwining. At the minimal rank-two full level it reduces to $K^{\times} J$, so the new coset has a genuine nontrivial character orbit.
BK 3.3.18	112–113	Correction at index $m = \lfloor r/2 \rfloor$ with $r < n$. The map is $\theta \mapsto \theta\psi_{\delta}^{-1}$, so the new character is $\Theta\psi_{\delta}$.
BK 3.4.1	114–116	Nondegenerate alternating form for the actual simple-character quotient. It does not assign a native quadratic refinement.
BK 3.6.13(ii), 3.6.14	136–137	Full-character transfer preserves the beta-determinant character. Multiplicity p gives $\chi(u^p)$. Both normalized full indices are zero.
BK 6.2.2–3	203–204	An actual extended maximal simple type induces to the supercuspidal candidate. The compact type alone fixes only an unramified-twist class.
BK 6.2.5	204–205	Torsion number and principal-order normalization. The period must be attached to the same actual type, not to an abstract lattice with the same index.

13.2 The exact refinement substitution

The refinement step is worth displaying as a substitution table. It is the arithmetic bridge between the explicit matrix recursion and a maximal field.

Published requirement	Value in the recursive construction
Initial simple stratum	$[\mathfrak{A}, n, R, \gamma]$
Perturbation	$b = \mathrm{Lift}_E(C) \in \mathfrak{P}^{-R}$
Derived parameter	$s(b) = C$
Pure maximal derived field	$E[C] \subseteq M_{p^{d-1}}(E)$ by induction
Outer critical exponent	$k_0(\gamma, \mathfrak{A}) = -n$
Inner critical exponent	$k_0(C, \mathfrak{B}) = -q_1$
Strict inequality	$-2R > -n - q_1$
Literal output field	$K = F[\gamma + b]$

For odd p, $n > pR > 2R$. For $p = 2$, $n > 2R$. Both imply the required strict inequality. Maximality of the derived field is a genuine induction hypothesis. It is not implied merely by counting its matrix entries.

13.3 Inputs that remain separately audited or deferred

The conductor formulas in Sections 6–8 retain their pinned arithmetic sources. Displayed numerical specializations are checked where indicated. This is not an end-to-end audit of every conductor theorem. The source-defined Bushnell–Henniart induction and descent branch remains separate from the direct matrix constructor and cannot serve as an unstated existence step in its proof.

The marked source attachment remains a scalar-pushout contract. Transporting the whole source marking does not identify it with an independently fixed host marking. The full finite-image center in low rank can exceed order two. Finally the native stabilizer obstruction and the same-candidate factor requirement are unchanged. These boundaries remain visible even though the parameter, character, and enlarged-induction arguments are now developed at greater length.

13.4 Distinguishing evidence from completion

Written proofs, finite tests and formal declarations have distinct scopes. None supplies a missing arithmetic input.

B Finite checks and their limits

The recorded replay covers four bounded gates. The initial evidence suite (Block 01) passed twelve test methods. The direct-construction suite passed eleven methods covering uniformizer identities, native and amplified traces, correction-depth endpoints, character extension, determinant transfer, low-rank commutators, filtration inequalities, and noncompact recovery. The exact rank-nine odd and rank-eight dyadic matrix scripts also passed unchanged in isolated working directories.

The direct-construction suite includes 256 binary leading-commutator cases, 153 low-rank normalization parameter pairs, 2,211 finite cyclic character-extension cases, and 972 finite Laurent-kernel cases. These are distinct finite model families, not a count of universal theorems. Other bounds and negative controls are listed separately in the JSON receipts. No aggregate assertion count is used as a measure of progress toward Conjecture 66.

The two matrix scripts use sparse Laurent arithmetic in positive characteristic. They check actual recursive matrix identities and grades, not merely floating-point approximations. They do not prove field maximality, simple-character existence, source attachment, the H5 factor comparison, or a universal realization statement. A proof reconstruction in prose is likewise not a Lean compilation.

The arithmetic proof route is complemented by the monomial, Clifford-descent, and binder analysis in Sections 14–21, the conductor and marking analysis in Sections 22–30, and the research atlas. These developments preserve the separate source and verification limits of each argument. They do not claim an exhaustive reconciliation of every archived construction.

Monomial and Clifford constructions

The first part of this research companion records the direct odd-prime and weighted dyadic constructions, including the low-rank carrier repairs. This continuation returns to the earlier monomial and Clifford branches. Their conclusions are not interchangeable. A genuine arithmetic candidate may have the right dimension and conductor while failing to contain a prescribed full simple character. A finite carrier may have enough dimension while failing the required action or marking. A compatible formal interface may still await its actual arithmetic input.

The purpose here is to make those distinctions mathematically usable. We extract the precise restricted implications, explain the negative results, and retain the surviving routes. The presentation includes elementary proofs where the argument is short enough to expose completely. Deeper arithmetic statements are identified as inherited ordinary arguments with named classical inputs. No theorem in this continuation is newly certified by Lean.

Source provenance. The first part of this companion is retained from a preserved reader. Its complete editable source archive was not recovered. The monomial and Clifford analysis uses the retained initial research archive. The source map records this boundary. This book does not claim to reconstruct unavailable source files.

Mathematical scope. Conjecture 66 remains the general arithmetic realization obligation, with the same-candidate H5 comparison kept separate. The positive constructions below are restricted statements. The negative examples refute specified shortcuts or prescribed candidates, not the entire conjecture.

14 The monomial branch and its actual scope

14.1 Three statements that must remain separate

The recovered monomial materials contain three different layers. The oldest manuscript develops a classical-input route from rank-one characters through automorphic induction. The finite formal core describes packet and group-theoretic interfaces. The ActualMonomialLocalGlobal module [M45] connects a degree-induced representation to actual residue and Frobenius data. None of these layers, merely by existing, constructs every arithmetic witness demanded by Conjecture 66.

A useful way to read the branch is to keep its input on the left:

rank-one character and admissible induction tower $\Longrightarrow$ the corresponding induced candidate.

This is not the implication from an arbitrary packet diagram to an admissible tower. The tower and its compatibility are substantial data. Nor does an equality between finite conductor profiles identify the full local factors of a separately specified candidate.

14.2 Actual Weil pullback and inertia invariants

Source [M45] uses an actual residue-action pullback and a degree homomorphism

$$\delta : W \longrightarrow \mathbf{Z}.$$

The subgroup for an induction of degree n is $\delta^{-1}(n\mathbf{Z})$. A finite cyclic inertia map is explicitly required to have image $\ker \delta$. This equality is a hypothesis to verify for the actual pullback, not an arbitrary identification of an abstract cyclic group with inertia.

Lemma 14.1 (Fixed vectors depend on the actual image)**.** *Let $\rho : G \to \mathrm{GL}(V)$ and let $\iota : H \to G$ be a homomorphism with image I. Then*

$$V^{\rho\circ\iota(H)} = V^{\rho(I)}.$$

Proof. A vector fixed by every $\rho(\iota(h))$ is fixed by every element of $\rho(I)$ because every element of I has a preimage. The reverse inclusion follows immediately from $\iota(H) \subseteq I$. □

The hypothesis that the first positive ramification group acts trivially places the representation in the tame sector. In the degree-induced examples, triviality of the inertia character gives an n-dimensional fixed space. A nontrivial inertia character, together with its Frobenius conjugates, gives no inertia-fixed coordinate. Thus the Artin conductor is respectively 0 or n, while the Swan conductor is zero. The local polynomial is formed on V^I, not automatically on the entire induced vector space.

This is meaningful arithmetic comparison at a finite interface. It is not a formal definition of an automorphic representation and does not import every analytic identity attached to one. The finite Gauss phase also does not become a fully normalized local epsilon factor without the relevant additive character and measure conventions.

Source and status. [M45] supplies the production module and binder conditions. The fixed-image lemma is proved here. The full module's terminal compilation receipt was not recovered. Earlier checked finite-tame comparison is not thereby invalidated.

14.3 Admissible towers rather than a blanket solvability claim

Let Ω be a finite group and $\Lambda \leq \Omega$. For the induction route in the older manuscript, the required condition is a chain

$$\Lambda = \Lambda_r \triangleleft \Lambda_{r-1} \triangleleft \cdots \triangleleft \Lambda_0 = \Omega, \qquad \Lambda_{i-1}/\Lambda_i \text{ cyclic of prime order.} \tag{14.1}$$

For a finite Galois extension with group Ω, the reversed fixed-field chain has cyclic prime-degree steps. The rank-one character can then be processed through the stated cyclic automorphic-induction inputs. Induction in stages identifies the resulting Weil parameter with the original induced character. Cuspidality requires the appropriate regularity and stabilizer conditions. Without them, an isobaric conclusion must not be relabeled as a cuspidal one.

Proposition 14.2 (Subnormality is the subgroup requirement)**.** *For a finite solvable group Ω, a subgroup Λ admits a chain of the form (14.1) if and only if it is subnormal in Ω.*

Proof. One direction is the definition of subnormality. Conversely, start with a subnormal chain. Each quotient of two successive terms is finite and solvable. A composition series of that quotient has cyclic prime-order factors. Pulling it back refines the original step into the required chain. Repeat for all steps. □

The statement does not say that every subgroup of a solvable group is subnormal. In S_3, a subgroup generated by a transposition has order two. There is no proper intermediate subgroup between it and S_3, and it is not normal in S_3. It cannot serve as the bottom of (14.1). Yet S_3 is solvable through its normal subgroup of order three. The exact finite suite includes this countercontrol.

The recovered manuscript has valid additional sectors, including nilpotent groups and suitable abelian-by-cyclic constructions. Their usefulness comes from producing the needed subgroup chain, not from deleting that requirement. Similarly, a virtual character identity with negative coefficients is not itself a construction of an actual automorphic representation. Dividing products of L-functions does not, by that operation alone, prove the required holomorphy.

14.4 What cyclic Clifford extension really supplies

Proposition 14.3 (Extension across a cyclic quotient)**.** *Let $H \triangleleft G$ be finite, with G/H cyclic of order d. Let τ be an irreducible complex representation of H whose isomorphism class is G-invariant. Then τ extends to G. Any two extensions differ by a character of G/H.*

Proof. Choose $t \in G$ lifting a generator. Invariance supplies an invertible operator A with

$$A\tau(h)A^{-1} = \tau(tht^{-1}).$$

The operator $A^d\tau(t^d)^{-1}$ commutes with $\tau(H)$ and is scalar by irreducibility. The same conjugation identity shows that A commutes with $\tau(t^d)$. Rescale A by a complex dth root so that $A^d = \tau(t^d)$. Sending t to this rescaled operator and retaining τ on H respects the defining conjugation and power relations, hence defines an extension.

After identifying the two restrictions with τ, the ratio between two possible operators at t commutes with $\tau(H)$ and is scalar. Its dth power is one. These scalars are precisely the characters of the cyclic quotient. □

This finite representation argument removes a cyclic projective extension obstruction. It does not prove automorphic descent for an arbitrary local-field construction. The latter needs an actual descent theorem and its hypotheses. This distinction explains why the monomial branch supplies a strong restricted route while leaving the general assembly problem intact.

Source and status. [M] contains the admissible tower and classical-input completion. [BH99, §2.3] supplies a characteristic-zero monomial route through cyclic prime-degree steps. The finite group proofs above are ordinary proofs, independent of any unverified compilation claim.

15 Genuine Clifford multiplicities and dimension

15.1 A projective multiplicity is not a decorative label

Clifford theory produces projective representations on multiplicity spaces. The carrier, cocycle, and actions must all come from the same actual restriction problem. Attaching several abstract Heisenberg packets to a diagram does not ensure that their multiplicities multiply inside a prescribed source representation.

Here is the dimension mechanism in a form that exposes its hypotheses. Let A be a finite abelian group and let $c : A \times A \to \mathbf{C}^\times$ be a unit-modulus multiplier. Write

$$T_a T_b = c(a,b) T_{a+b}, \qquad b_c(a,b) = \frac{c(a,b)}{c(b,a)}, \qquad R = \{a : b_c(a,b) = 1 \text{ for every } b \in A\}.$$

The commutator pairing b_c is alternating. The group R is its radical.

Proposition 15.1 (Degree of an irreducible projective multiplicity)**.** *For an irreducible projective representation with multiplier c on M,*

$$(\dim M)^2 = |A/R|. \tag{15.1}$$

Every finite-dimensional module with this same multiplier has dimension divisible by $\sqrt{|A/R|}$.

Proof. Average an inner product over the operators T_a. Unit modulus of the multiplier makes the averaged inner product invariant, so the operators may be taken unitary. If $r \in R$, then T_r commutes with all T_a and is scalar. Thus $|\operatorname{Tr} T_r| = \dim M$. If $a \notin R$, some T_b conjugates T_a to a nontrivial scalar multiple of itself, so $\operatorname{Tr} T_a = 0$.

The operators $\operatorname{Ad}(T_a)$ form an ordinary representation on $\operatorname{End}(M)$. Their average is the projection onto the commutant, which has dimension one by irreducibility. Its trace is

$$1 = \frac{1}{|A|} \sum_{a \in A} |\operatorname{Tr} T_a|^2 = \frac{|R|}{|A|} (\dim M)^2.$$

This proves (15.1). Finally, averaging a projection gives invariant complements, so modules with the fixed unitary multiplier are semisimple. Every irreducible summand has the degree already computed, proving divisibility. □

No assertion that the original cocycle literally descends to A/R is needed. Choosing a section can produce a quotient multiplier after incorporating the scalar radical action, but that is additional bookkeeping, not a reason to suppress the radical.

If A is elementary abelian of order a power of p and the nondegenerate quotient has dimension $2r$ over $\mathbf{F}_p$, the multiplicity degree is p^r. Independent commuting projective factors on the *same* carrier produce the tensor product degree $p^{r_1+\cdots+r_s}$. Both independence and commutation matter. A single Weyl plane represented twice on a p-dimensional space still has noncommuting cross-operators. It does not force divisibility by p^2.

15.2 The orbit factors in a normal chain

Proposition 15.2 (A compatible Clifford chain)**.** *Suppose*

$$G_0 \lhd G_1 \lhd \cdots \lhd G_s$$

is a chain of finite groups. Choose irreducible ρ_i of G_i such that ρ_{i-1} occurs in $\rho_i|_{G_{i-1}}$. Let I_i be the inertia group of ρ_{i-1} in G_i and $t_i = [G_i : I_i]$. For a fixed prime p, if G_i/G_{i-1} is elementary abelian of p-power order and the actual projective multiplicity pairing has rank $2r_i$, then

$$\dim \rho_s = \dim \rho_0 \prod_{i=1}^{s} t_i p^{r_i}. \tag{15.2}$$

Proof. The restriction at the ith step is a sum of t_i distinct conjugate irreducibles, each of dimension $\dim \rho_{i-1}$ and common multiplicity e_i. The corresponding isotypic component is irreducible as a module for I_i. Factoring out the G_{i-1} representation leaves the actual irreducible projective multiplicity representation of I_i/G_{i-1}. Its commutator rank is $2r_i$, so Proposition 15.1 gives $e_i = p^{r_i}$. Hence $\dim \rho_i = t_i p^{r_i} \dim \rho_{i-1}$. Multiplication over the chain proves the formula. □

For example, the two-dimensional irreducible of S_3 restricts to the two nontrivial characters of C_3. The multiplicity is one, and the orbit factor is two. Dropping the orbit factor would give the wrong dimension. A valid source bound therefore uses a genuine compatible chain, not merely a list of candidate projective factors.

A consequence is $p^{\sum r_i} \le \dim \rho_s$. This can make a proposed target dimensionally plausible. The fixed-rank constructor recorded in [C161] has, under its selected even-depth hypotheses, a host half-dimension $fn(n-1)/2$ over $\mathbf{F}_p$, where the residue field has order p^f. The source bound then provides enough room for the relevant abstract symplectic dimensions. That conclusion remains dimensional. It does not identify a native subgroup, prescribe its quadratic marking, make the action equivariant, or prove full-character occurrence.

Source and status. [C161] gives the genuine-chain constraint and the constructor comparison. Propositions 15.1 and 15.2 are expanded ordinary arguments. The executed finite models check small Weyl carriers, radicals, repeated-factor failures, and the S_3 orbit example, not arbitrary arithmetic host realization.

16 Genuine source fixtures and what they rule out

16.1 Two packets inside a single-slope source

The continuation112 fixtures were designed to avoid an artificial diagram whose purported packets did not come from one irreducible source. In the first construction, over $\mathbf{F}_5((t))$, an unramified degree-four extension supports two independent Heisenberg extensions. Their Schrödinger representations have dimensions five and five. The tensor product is irreducible of dimension 25.

For one factor the finite group law is

$$(u,v,w)(u',v',w') = (u+u', v+v', w+w'+uv'),$$

and its action on functions on $\mathbf{F}_p$ is

$$(T_{u,v,w}f)(s) = \zeta_p^{w+vs} f(s+u). \tag{16.1}$$

Direct composition verifies the group law. In the tensor product, the kernel on the product of the two centers is the anti-diagonal

$$\{(z(w), z(-w)) : w \in \mathbf{F}_5\}.$$

The quotient is the balanced central product, not the direct product with two unrelated central characters. The common central character remains the prescribed character.

The two arithmetic factors have ramification breaks $\{2,4\}$ and $\{3,6\}$. Their disjointness supplies the product filtration. After quotienting by the balanced central kernel, the ambient jumps are $2,3,6$. The ordinary representation has only slope six, while the actual chain still contains two multiplicity-five packets. Thus a sole ordinary slope does not force a single projective multiplicity packet.

The unramified Frobenius action has order four. It must not be relabeled as a nontrivial tame inertia group. The source fixture has trivial tame stabilizer in this sense, even though Frobenius acts nontrivially on the finite operators.

16.2 The symmetric characteristic-three fixture

The refined example works over $F = \mathbf{F}_3((t))$, with unramified degree-four field $K = \mathbf{F}_{81}((t))$. Choose a with $a^9 = -a \neq 0$ and put $b = a^3$. For $m = 1, 5$, write $f = at^{-m}$ and $g = bt^{-m}$. The torsor equations are

$$\wp x = f, \qquad \wp y = g, \qquad \wp z = \frac{fy - gx}{2}, \qquad \wp u = u^3 - u. \tag{16.2}$$

With $w = z + xy/2$, the final equation becomes

$$\wp w = fy + \frac{fg}{2}.$$

For the ramification formula in [Imai, Example 3.8(1)], the differential term is

$$d(fg/2) - f\,dg = \frac{g\,df - f\,dg}{2} = 0.$$

The pole orders m and $2m$ are prime to three. The leading coefficients a, b are linearly independent over $\mathbf{F}_3$: otherwise $a^2 = b/a$ would lie in $\mathbf{F}_3$, contradicting $a^8 = -1$. The total ramification condition is supplied by the torsor argument in [C112]. Under these hypotheses, the two remaining terms give

central break $4m/3$, not $2m$. The latter would correspond to the different unsymmetrized data. The factor break sets are therefore

$$\{1, 4/3\}, \qquad \{5, 20/3\}.$$

The balanced quotient has ambient jumps $1, 5, 20/3$. Its nine-dimensional irreducible source has a sole slope $20/3$, hence Swan conductor 60, and two genuine multiplicity-three packets. Native subgroup labels in a different upper-numbering convention cannot be substituted for these ambient labels.

The ordinary arithmetic analysis of [C112] then tests whether suitable per-break simple-character targets can all be replaced by one raw maximal-type quotient of the same rank. Its rank-nine spectrum excludes the exponent needed for a degree-nine Heisenberg representation in that specific quotient. The result is a failure of that proposed one-quotient shortcut. It does not show that the source satisfies every strengthened realization hypothesis and then violates Conjecture 66.

16.3 Why the fixtures and later enlargements coexist

The first fixture did not even supply all the per-break arithmetic targets required by the stronger test. The second supplies a sharper comparison but still does not provide the full native marking, Frobenius action, and same-candidate H5 package. These omissions are part of its status, not cosmetic caveats.

The later low-rank enlargements in Part I do not retroactively make a forbidden original quotient large enough. They change the assembly carrier while preserving a specified inducing representation. That is a different construction. The argument therefore separates: genuine source multiplicities, obstruction to a particular arithmetic quotient, and a search for a permitted larger carrier with explicit recovery. It is not a sequence of contradictory claims about one unchanged quotient.

Source and status. [C112] records the two arithmetic source fixtures and the maximal-type spectrum test. [Imai] supplies the selected ramification formula. The finite operator law and balanced-center kernel have been rerun here. The local-field torsor construction and full spectrum theorem remain inherited ordinary arguments, not newly formalized results.

17 An actual arithmetic candidate and a failed prescribed character

17.1 The field, the character, and the source

Continuation113 supplies an explicit characteristic-zero experiment in which the source representation and the arithmetic candidate exist independently of the proposed packet attachment. Fix compatible roots of unity and write

$$F = \mathbf{Q}_3(\zeta_3), \qquad L = \mathbf{Q}_3(\zeta_9), \qquad \varpi = \zeta_9 - 1, \qquad \varpi_F = \zeta_3 - 1, \qquad \beta = \varpi^{-10}. \tag{17.1}$$

The extension L/F is cyclic and totally ramified of degree three. The polynomial of ϖ over F is

$$X^3 + 3X^2 + 3X - \varpi_F.$$

Normalize $v_L(\varpi) = 1$. Then $v_L(3) = 6$ and the relative different has exponent six. Let $\psi_{\mathbf{Q}_3}$ have kernel $\mathbf{Z}_3$, and set

$$\psi_F(x) = \psi_{\mathbf{Q}_3}\left(\frac{\mathrm{Tr}_{F/\mathbf{Q}_3}(x)}{3}\right), \qquad \psi_L = \psi_F \circ \mathrm{Tr}_{L/F}.$$

The conductor ideal of ψ_L is $\mathfrak{p}_L^{-3}$. This negative exponent is essential. Replacing it by $\mathfrak{p}_L$ shifts the unit thresholds and changes the calculation.

Every element of $L^\times$ has a unique expression $\varpi^a \epsilon u$ with $a \in \mathbf{Z}$, $\epsilon \in \{1, -1\}$, and $u \in U_L^1$. Define the full multiplicative character by

$$\chi(\varpi^a \epsilon u) = \psi_L(\beta \log u). \tag{17.2}$$

The logarithm converges on U_L^1. At depths at least four, logarithm and exponential are inverse, so the conductor condition becomes $r - 10 \geq -3$. Consequently $a_L(\chi) = 7$. With $\sigma(\zeta_9) = \zeta_9^4$, one obtains

$$v_L(\sigma\beta - \beta) = -8.$$

The nontrivial conjugate quotient has conductor five. In particular, the three conjugate characters are distinct on the actual unit group, not merely assigned distinct labels.

The recorded exact program operates in the rational algebra

$$\mathbf{Q}[X]/(X^6 + 6X^5 + 15X^4 + 21X^3 + 18X^2 + 9X + 3).$$

It verifies field identities and rational trace witnesses such as

$$\chi(\exp(\varpi^6)) = \zeta_3, \qquad \psi_L((\sigma\beta - \beta)\varpi^4) = \zeta_3^2.$$

The program does not approximate a 3-adic exponential. The analytic reduction $\log(\exp x) = x$ is used only in its valid range, after which the calculation is exact rational arithmetic.

Regularity makes

$$\rho = \mathrm{Ind}_{W_L}^{W_F} \chi$$

irreducible of dimension three. The argument in [C113] excludes unramified self-twists. Under [BH99, Proposition 2.2], its wild restriction is irreducible as well. The conductor induction calculation gives

$$a_F(\rho) = 7 + 6 = 13, \qquad \mathrm{Sw}_F(\rho) = 10, \qquad \mathrm{slope}(\rho) = 10/3. \tag{17.3}$$

These are conductor statements for this source, not proofs of the general pair-factor contract.

17.2 The source-defined automorphic-induction candidate

The characteristic-zero cyclic monomial route in [BH99, §2.3] gives an actual supercuspidal candidate

$$\pi_F^0 = \mathrm{AI}_{L/F}(\chi). \tag{17.4}$$

The central character is $\omega = \chi|_{F^\times}$. This construction imports the established automorphic-induction input. It is not described as selecting a representation with convenient packet coordinates and then declaring it to be the realizer of ρ.

For this monomial subcase, the classical source has stronger comparison properties than conductor equality alone. Those properties belong to this actual candidate and its stated local correspondence. They do not prove that an independently specified simple character occurs in π_F^0, nor do they automatically identify a different constructed type with it.

17.3 The independent simple character

Let $\mathfrak{a}$ stabilize the $\mathcal{O}_L$ lattice chain on the three-dimensional F-space L. The element $\beta = \varpi^{-10}$ is minimal over F. The actual full-degree simple stratum gives

$$H^1 = U_L^1 U_{\mathfrak{a}}^6, \qquad J^1 = U_L^1 U_{\mathfrak{a}}^5.$$

The proposed simple character is

$$\theta_0(uy) = \chi(u)\,\psi_F\Big(\mathrm{tr}_{M_3(F)/F}(\beta(y-1))\Big). \tag{17.5}$$

The intersection condition is checked on U_L^6. The quotient J^1/H^1 has order nine, and the associated Heisenberg representation η_0 has dimension three. The type-side conductor agrees with 13.

At this point two real objects have been produced: the source-defined π_F^0 and the independently defined θ_0. The missing assertion is not their existence but

$$\mathrm{Hom}_{H^1}(\theta_0, \pi_F^0) \neq 0. \tag{17.6}$$

The computation below shows that this assertion is false. This is more informative than marking it as an unfinished verification.

17.4 A nontrivial tame descent that still leaves occurrence open

Set $K = F(i)$, the unramified quadratic extension, and $M = KL$. The character $\chi_M = \chi \circ N_{M/L}$ gives $\pi_K^0 = \mathrm{AI}_{M/K}(\chi_M)$. Its Galois invariance and norm-compatible central character meet the tame descent conditions in [BH99, Theorem 1.3]. In characteristic zero, Theorem 1.8 identifies the tame algebraic lift with cyclic base change here, since $[K : F] = 2$ is prime to three.

Lemma 17.1 (Central normalization of a descent fiber)**.** *Suppose a nonempty descent fiber is a torsor under a group X of characters of exponent dividing d. Suppose twisting a rank-n representation by $\alpha \in X$ multiplies its central character by α^n. If $\gcd(n, d) = 1$, every compatible specified central character selects a unique member of the fiber.*

Proof. Choose integers r, s with $rn + sd = 1$. The power map $\alpha \mapsto \alpha^n$ on X has inverse $\alpha \mapsto \alpha^r$, because $\alpha^d = 1$. Relative to one member of the nonempty fiber a compatible desired central-character ratio is an element of X. There is exactly one twisting character with that nth power. □

Here $n = 3$ and $d = 2$, so the central-normalized descent is unique. This is a useful actual descent conclusion. The applicable type transfer also provides an occurrence equivalence between F and K. It does not manufacture the initial occurrence. In this example $\dim \eta_K = 9$ while $\dim \eta_F = 3$, and the literal restriction of the transferred simple character has the power $\theta_K|_{H_F^1} = \theta_F^2$. Equating the two entire finite carriers would erase the actual transfer mechanism.

Source and status. [C113] contains the field, character, conductor, type and descent arguments. The cyclotomic identities and normalization checks were replayed in the recorded audit. [BH99, printed pp. 15, 18, 24 and 48] was read at the selected descent, monomial and central-type statements. A full new audit of every conductor formula and every transfer theorem is not claimed.

18 A full-extension scalar obstruction and its limited repair

18.1 Why testing only the compact part can mislead

The cubic norm-kernel character of $F^\times$ defines $\Omega = \omega_{\mathrm{cub}} \circ \det$ on $\mathrm{GL}_3(F)$. The source-defined automorphic induction satisfies

$$\pi_F^0 \otimes \Omega \simeq \pi_F^0.$$

If a proposed extended type is to induce to π_F^0, it must be compatible with this self-twist. Continuation113 finds an element g that makes the compact character appear compatible, but whose action on a noncompact generator leaves a nontrivial scalar.

In the basis $1, \varpi, \varpi^2$, multiplication by ϖ is

$$\Pi = \begin{pmatrix} 0 & 0 & \varpi_F \\ 1 & 0 & -3 \\ 0 & 1 & -3 \end{pmatrix}, \qquad g = I + \mathrm{diag}(0,1,2)\Pi^4. \tag{18.1}$$

The actual group calculations show that g normalizes the relevant compact groups and the extended inducing group $N = L^\times J^1$. With the convention $\tau^g(x) = \tau(g^{-1}xg)$, one has

$$\theta_0^g = \theta_0\Omega, \qquad \eta_0^g = \eta_0\,\Omega|_{J^1}$$

in the literal operator models used in the audit. The check includes H^1 and two quotient generators, not merely a comparison of characters on an abstract vector space.

The decisive element is

$$h = g^{-1}\Pi g \Pi^{-1} \in H^1, \qquad \theta_0(h) = \zeta_3. \tag{18.2}$$

In contrast, $\Omega(\Pi) = 1$. For every full central extension Λ of η_0, the compact equality remains true but

$$\Lambda^g(\Pi) = \zeta_3\Lambda(\Pi).$$

Lemma 18.1 (A generator scalar cannot be removed by an intertwiner rescaling)**.** *Let two representations of $N = \langle J, z\rangle$ have the same irreducible restriction to J. Suppose their operators at z differ by a scalar $\xi \neq 1$. Then they have no nonzero intertwiner.*

Proof. Any intertwiner on J is scalar by irreducibility. A nonzero scalar commutes with the invertible operator at z. The intertwining equation at z then forces $\xi = 1$, a contradiction. □

The inherited all-extension argument combines this lemma with the actual simple-character intertwining group and the maximal-type induction theorem. Those inputs rule out an alternative intertwining coset elsewhere in the ambient group. The conclusion is therefore not based on numerically searching a finite list of matrices. Every supercuspidal containing this prescribed θ_0 fails the required cubic self-twist, whereas π_F^0 has it. Thus

$$\boxed{\mathrm{Hom}_{H^1}(\theta_0, \pi_F^0) = 0.} \tag{18.3}$$

This is the exact failure of the original prescribed-character route. Neither rescaling an intertwiner nor assigning a different scalar to the same noncompact extension repairs it.

18.2 The independent check and its scope

The replay includes the original rational cyclotomic program, the central-type calculations for three corrections, and an independent scalar calculation. The independent program decomposes the actual matrix h into an L-unit and a deep congruence term. It computes the phase in (18.2) directly, without searching for a Weyl intertwiner or choosing a favored central-type matrix.

This is strong evidence for the specific scalar input. The universal phrase "every extension" still uses the ordinary group-generation and irreducibility argument. Finite arithmetic alone is not being credited with proving a quantified representation-theoretic statement.

18.3 A different full character passes the obstruction

Define a wild unit correction on U_L^1 by

$$\nu(u) = \psi_L(\varpi^{-8} \log u). \tag{18.4}$$

Its conductor is five. It leaves the deeper part used in (17.5) unchanged but changes the full unit character. The corrected simple character is

$$\theta'(uy) = \chi(u)\nu(u)\, \psi_F\Big(\mathrm{tr}_{M_3(F)/F}(\beta(y-1))\Big). \tag{18.5}$$

It is not merely a different choice of basis for θ_0.

The noncompact normalization matters here too. The correction is trivial on U_F^1, but that alone does not establish $\nu|_{F^\times} = 1$. The inherited calculation gives

$$\nu(\varpi_F/\varpi^3) = \zeta_3^2.$$

Assigning $\nu(\varpi) = 1$ would therefore change the central character. Instead set

$$\nu(\varpi) = \zeta_9, \qquad \nu(-1) = 1.$$

Then

$$\nu(\varpi_F) = \nu(\varpi)^3 \nu(\varpi_F/\varpi^3) = \zeta_3\zeta_3^2 = 1. \tag{18.6}$$

This is the required full central normalization.

For the correction powers ν^a, $a = 0, 1, 2$, the remaining generator scalars are respectively

$$\zeta_3, \qquad 1, \qquad \zeta_3^2. \tag{18.7}$$

The middle choice passes the cubic self-twist obstruction. The two other choices remain negative controls. The exact programs reproduce all three cases.

18.4 The remaining selection problem

Passing the self-twist test is necessary but not sufficient to establish

$$\mathrm{Hom}_{H^1}(\theta', \pi_F^0) \neq 0. \tag{18.8}$$

The fixed-center central-type family still has three members in this rank-three setting. [BH99, §5.4] describes such a family as a torsor under the corresponding unramified characters. The corrected scalar does not single out the source-defined automorphic-induction member of that family.

A successful next argument would have to establish occurrence in the actual π_F^0 and then identify the appropriate noncompact extension. Producing some representation containing θ' is a weaker statement. Replacing π_F^0 by a convenient member of the family would abandon the same-candidate obligation. Replacing the supplied canonical extension would be another change of input and would have to be tracked explicitly.

Source and status. [C113-S] gives the all-extension obstruction, central-type hypotheses and corrected character. The recorded exact checks establish the named cyclotomic identities and phases. Equations (18.3) and (18.7) retain that ordinary proof boundary. Equation (18.8) remains an unresolved occurrence assertion, not a failed theorem-prover build.

19 Compact binders and exact fixed-template obstructions

19.1 The compactness theorem with all hypotheses visible

A binder is intended to turn compatible finite constraints into one actual object. Compactness can perform that passage only after the admissible space and simultaneous finite solvability have been supplied. It does not supply those premises from unrelated finite successes.

Let $\mathcal{O}$ be the valuation ring of a nonarchimedean local field, with uniformizer π. Fix a closed subset $X \subseteq \mathcal{O}^m$ and continuous functions $f_j : X \to \mathcal{O}$, $j \geq 1$. Put

$$C_n = \{x \in X : f_j(x) \in \pi^n \mathcal{O} \text{ for } 1 \leq j \leq n\}. \tag{19.1}$$

Proposition 19.1 (A compact simultaneous binder)**.** *If $C_n \neq \varnothing$ for every n, there is $x \in X$ with $f_j(x) = 0$ for every j.*

Proof. The ring $\mathcal{O}$ is compact, so X is compact. Each C_n is closed by continuity and the closedness of $\pi^n\mathcal{O}$. Also $C_{n+1} \subseteq C_n$. The finite intersection property gives an element x in their intersection. For fixed j, the value $f_j(x)$ lies in every $\pi^n\mathcal{O}$ with $n \geq j$, so it is zero. □

The proof requires no surjectivity of the reduction maps between the sets of finite roots. For instance, $x^2 = 17$ has roots modulo every power of two, while the roots 3 and 5 modulo eight do not lift to roots modulo sixteen. An arbitrarily chosen partial root can fail even when the compactness conclusion applies to the full nested solution sets.

Separate solvability is also insufficient. The equations $x \equiv 0$ and $x \equiv 1 \pmod{3^n}$ each have solutions, but never have a common solution. Every finite test in the binder premise concerns one tuple satisfying all current constraints together.

19.2 What makes the result an arithmetic certificate

The variables must remain in one bounded, closed model with fixed rank, lattice data and normalizations. A nonzero leading coefficient should be encoded by a fixed nonzero residue class, a closed and open condition. Merely deleting the zero point from a compact model need not leave a compact space. In $\mathcal{O} \setminus \{0\}$, all congruences $x \equiv 0 \pmod{\pi^n}$ are solvable, but the only exact zero has been removed.

For polynomial constraints, or rational constraints with denominators uniformly units, congruence evaluation is compatible with a fixed finite quotient. An exhaustive empty solution set at some precision then certifies failure of that template. Conversely, if the exact common zero set in the compact template

is empty, Proposition 19.1 implies that some finite stage is empty. This is a finite obstruction certificate for the *specified model*, not automatically for every possible arithmetic host.

Full characters require particular care. The domain on which their finite quotients are modeled must be fixed, and the evaluation constraints must be jointly continuous or uniformly descend to the claimed finite quotient. Values on noncompact generators require their own normalization constraints. A binder that only remembers the deep commutator form does not recover a full inducing representation for free.

19.3 A corner with a complete leading-profile test

The earlier cubic calculation is a concrete example of a useful finite obstruction. Work over $\mathbf{F}_3((t))$. The four-coordinate square-zero corner has entries in positions $05, 16, 27, 38$. Adding a fifth coordinate gives

$$x = t(aE_{05} + bE_{16} + cE_{27} + dE_{38}) + \frac{t^2 h}{1+t} E_{08}. \tag{19.2}$$

The relevant leading target parameters are

$$u = cd, \qquad v = ab, \qquad w = bc, \qquad p = c(a-h), \qquad q = b(d-h). \tag{19.3}$$

Here u, v, w are required nonzero. The structural calculation for the larger sixteen-coordinate corner shows that its remaining entries and higher series terms do not change these decisive coefficients. That leading-coefficient fact is an inherited symbolic calculation, not an assumption that five coordinates exhaust all matrices.

Proposition 19.2 (The nonzero-leading corner spectrum)**.** *For targets in* $\mathbf{F}_3$ *with* $uvw \neq 0$*, equations* (19.3) *have a solution if and only if*

$$p + u = w(q + v). \tag{19.4}$$

Exactly 24 of the 72 possible targets satisfy this condition.

Proof. Write $b = s \neq 0$. The first three equations force

$$a = v/s, \qquad c = w/s, \qquad d = us/w.$$

The equation for q then forces $h = us/w - q/s$. Substitution into the equation for p gives

$$s^2(p + u) = w(q + v).$$

Over $\mathbf{F}_3$, every nonzero s has square one. This proves necessity and, by the displayed formulas, sufficiency. There are eight choices of the nonzero triple (u, v, w) and three choices of q, after which p is forced. Hence there are 24 admissible targets among $8 \cdot 3 \cdot 3 = 72$. □

The target $(u, v, w, p, q) = (1, 1, 1, 2, 2)$ cannot be obtained with $h = 0$ but is obtained by the fifth coordinate. The target $(1, 1, 1, 1, 2)$ violates (19.4), so the larger corner still cannot produce it. The first is an actual repair of an overly small ansatz. The second is an obstruction that survives that repair.

Neither conclusion proves that the target is impossible in every arithmetic matrix realization. It identifies precisely where this corner stops. That distinction is what makes the negative result reusable rather than misleading.

19.4 The arbitrary-exponent cubic family

The later general binder165 changes the construction rather than repeatedly enlarging the same leading-coefficient search. Let E/F be the cubic field with

$$e^3 + te + t = 0, \qquad \pi = -t = \frac{e^3}{1+e}.$$

On the period-nine lattice of E^3, let S denote the lattice shift. For $h > 0$ with $3 \nmid h$, set $a = 4h$ and take $x_h = S^a D$, where D is supported in residue positions

$$\{0, 2a, 4a, 6a\} \pmod 9.$$

Its image positions are $\{a, 3a, 5a, 7a\}$, disjoint from its support, so $x_h^2 = 0$. In the expression $x_h S^{-3a} x_h$, exactly the three starting positions $2a, 4a, 6a$ survive. Modulo three they meet the three E-lanes once each. This support statement holds for every allowed h because a is invertible modulo nine.

Under the retained normalized corestriction calculation, these paths give a primitive whole cycle. The full ordinary binder proposition then constructs the stated degree-nine parameter for allowed $h_1 < h_2$, with critical data $n = 12h_2$ and $k_0 = -4h_1$. The finite replay of support residues illustrates this mechanism. The universal construction rests on the symbolic proof and its refinement hypotheses, not on testing the first hundred exponents.

This family is genuine progress beyond one failed corner. It is still a parameter construction. It does not, without the additional full-character, stabilizer and transport arguments, provide the universal realization endpoint. The later direct odd-prime and dyadic constructions in Part I should be read as further developments of the constructor side, not as proofs that all the earlier occurrence obstructions disappeared.

Source and status. [B162] supplies the fixed compact model theorem. [Corner] supplies the whole-corner leading calculation. [B165] supplies the arbitrary-exponent construction. The compactness and finite profile implications are proved here. The underlying full arithmetic refinement is retained as an ordinary source argument.

20 What this synthesis closes and what it does not

20.1 A single dependency chain

The source history supports a coherent sequence. A monomial source with an admissible induction tower gives a restricted actual candidate through stated classical inputs. Genuine Clifford theory limits the multiplicities that can occur inside one source. Explicit fixtures show that ordinary slope data do not determine the packet chain. An independently constructed simple character then poses a genuine occurrence problem for that candidate.

Continuation113 resolves one such occurrence question negatively and repairs one necessary self-twist condition only by changing the full character. Compactness explains how a fixed admissible template can be assembled from simultaneous finite solutions, while explicit corners give finite obstruction certificates. The later constructor families enlarge what can be built but do not identify every constructed object with an independently fixed candidate.

Statement available	Obligation that is not supplied by it
Admissible monomial induction	An admissible tower for every allowed source.
Projective dimension bound	A native, marked and equivariant arithmetic embedding.
Conductor agreement	Occurrence of the prescribed full character.
Cubic self-twist compatibility	Selection of the source-defined central type.
Nonempty finite systems	A limit unless the model is fixed, compact and simultaneous.
A repaired matrix corner	Success in every target profile or arithmetic host.
A larger inducing carrier	Permission to change the original candidate or extension.

20.2 The most precise surviving occurrence task

For the characteristic-zero experiment, the next question is (18.8), followed by the identification of the full noncompact central-type extension. It is not sensible to retry the already excluded (17.6) by changing only an intertwiner scalar. Its negative proof addresses every such rescaling.

A future attempt at (18.8) should use an invariant or a theorem that actually distinguishes the three fixed-center possibilities. The cubic self-twist test does not do so. An explicit comparison of the full inducing character under the monomial construction, or a sufficiently discriminating normalized local-factor calculation for the same candidate, would address a stronger question than the conductor comparisons already performed. Neither is asserted as completed here.

20.3 Conductor and marking boundaries

The selected characteristic-zero monomial and tame-descent inputs do not by themselves audit the separate conductor arguments used by the direct positive-characteristic constructors. The hypotheses of [BH99, §2.3], where characteristic zero is explicit, cannot be imported into that route. The conductor analysis in Sections 22–25 keeps the source-defined and direct constructions separate.

A scalar pushout can carry a supplied subgroup and phase convention without matching a fixed native quadratic refinement. Sections 26–30 examine this distinction and the equal-capacity dyadic boundary.

21 Evidence and reproducibility

21.1 Evidence classes

Four positive evidence classes accompany the open obligations. An ordinary proof displays its hypotheses. An inherited arithmetic argument retains its classical dependencies. A bounded check records an executed program, input hash, and scope. A formal receipt records a compiler execution. None of these classes converts an open obligation into an established result or implies a new compiler run.

The selected [BH99] page images were inspected at printed pages 15, 18, 24 and 48. These passages establish the scope of the tame descent fiber, the characteristic-zero comparison with base change, the monomial route, and the residual fixed-center ambiguity. The Imai ramification formula is used only at the specialized Heisenberg example. Neither source audit is a claim to have checked the entirety of either work.

21.2 Executable checks

The tests are deliberately bounded. For example, the corner test checks all 72 leading target profiles, while the symbolic statement explaining why the other corner coordinates cannot change the decisive coefficients is a separate ordinary input. The Weyl tests verify finite operators, not local automorphic induction. The exact field calculation verifies rational identities, not a new proof of the classical type theory.

The corrected-character occurrence question remains

a separate mathematical obligation. It is not a premise silently supplied by the conductor or marking audits.

Conductors and native markings

A conductor is an arithmetic invariant, not a complete description of a type. A nondegenerate commutator is a finite invariant, not a complete description of a marked dyadic extension. These two distinctions meet at the smallest dyadic construction with exactly enough quotient capacity. There is no spare plane there to absorb an independently prescribed quadratic mismatch.

This continuation audits the conductor and marking branches at their actual hypotheses. It keeps three arithmetic objects separate: the original source representation, a candidate obtained through a classical induction and descent construction, and a candidate arising from a proposed explicit type. A comparison is useful only when its normalization and its subject are fixed.

The resulting finite theory is also constructive. A discrepancy between two quadratic refinements becomes one linear functional. When its Arf obstruction vanishes, a single transvection gives an explicit correction. For a supplied odd-order finite action, a further central discrepancy can be removed by a finite sum. Neither correction constructs the missing arithmetic action, chooses a canonical native marking, or settles the full phase equations.

Source provenance. The conductor and marking analysis retains the earlier parts' source and review boundaries. In particular, the first part is available as a preserved reader, while its complete editable archive remains unavailable. The evidence ledger distinguishes this documentary gap from the unresolved mathematical obligations.

Evidence status. Ordinary finite proofs are supplied below. The arithmetic construction records are inherited arguments audited against selected primary statements. Recorded exact tests check finite models and normalizations. No fresh Lean compilation or external peer review is claimed.

22 Conductor conventions and actual order periods

22.1 Four numbers, not one unnamed conductor

Let F be a nonarchimedean local field with residue cardinality q, and let V be a complex Weil representation with finite inertia image and no monodromy. Its ordinary Artin conductor is

$$a(V) = \dim V - \dim V^{I_F} + \mathrm{Sw}(V). \tag{22.1}$$

We use a for the ordinary conductor throughout this part. The Swan conductor, an epsilon exponent for a specified additive character, and the self-pair conductor of a representation of $\mathrm{GL}_n(F)$ are named separately.

Take ψ_0 trivial on $\mathcal{O}_F$ and nontrivial on $\mathfrak{p}_F^{-1}$. Set $\psi_1(x) = \psi_0(\varpi^{-1}x)$, so ψ_1 is trivial on $\mathfrak{p}_F$ and nontrivial on $\mathcal{O}_F$. Under the usual change-of-character convention, if the exponent is written as a power of $q^{1/2-s}$, then

$$f_{\psi_0}(V) = a(V), \qquad f_{\psi_1}(V) = a(V) - \dim V. \tag{22.2}$$

The determinant factor in the change of additive character is independent of s and does not alter this exponent shift. Equation (22.2) is a convention translation, not an assertion that every epsilon constant is determined by its exponent.

For a wild-irreducible source ρ of dimension nine with Swan conductor 24, the audited self-pair calculation below gives the following table.

Object	Dimension	I_F-fixed	Swan	Ordinary a	f_{ψ_1}
ρ	9	0	24	33	24
$\mathrm{End}\,\rho$	81	1	152	232	151
$\mathrm{Ad}\,\rho$	80	0	152	232	152

Here $\mathrm{Ad}\,\rho$ denotes the trace-zero summand. Over the complex coefficient field, $\mathrm{End}\,\rho = \mathbf{C} \oplus \mathrm{Ad}\,\rho$. The scalar summand is unramified, so it contributes no ordinary conductor. It does contribute one to the dimension shift in (22.2).

Thus the number 152 has two legitimate appearances, as the Swan conductor of $\mathrm{End}\,\rho$ and as the ψ_1 exponent of its trace-zero summand. It is *not* the ψ_1 exponent of all of $\mathrm{End}\,\rho$, which is 151. The formal-degree parameter δ appearing later also equals 152 in this particular torsion-one comparison. That numerical equality does not make the three definitions interchangeable.

Source and status. The ordinary conductor identity and additive-character normalization underlie [Q14]. [BH99, §6.1, Lemma 2] uses the character trivial on $\mathfrak{p}_F$. [AP, §4 and Theorem 6.9] uses the ordinary pair conductor. The table is an explicit reconciliation of these conventions using the source calculation in Section 23.

22.2 The principal-order grade calculation

Let $\mathfrak{A} \subset M_n(F)$ be a principal hereditary order of period e, with radical $\mathfrak{P}$. Then $e \mid n$. A principal lattice chain has e successive residue pieces, each of dimension $m = n/e$ over k_F. The homogeneous grade of an endomorphism is the direct sum of e matrix blocks of size m by m. Consequently

$$\dim_{k_F} \mathfrak{P}^j/\mathfrak{P}^{j+1} = em^2 = \frac{n^2}{e}, \qquad [\mathfrak{P}^{-r} : \mathfrak{A}] = q^{rn^2/e} \quad (r > 0). \tag{22.3}$$

The period here belongs to the actual principal order carrying the stratum. It cannot be borrowed from a more convenient unrelated lattice chain.

For an actual cuspidal representation containing a positive-level simple stratum $[\mathfrak{A}, r, 0, \beta]$ to which the hypotheses of [BH99, §6.1] apply, its Lemma 2 gives

$$f_{\psi_1}(\pi) = \frac{nr}{e}, \qquad a(\pi) = n + \frac{nr}{e}. \tag{22.4}$$

The first number is the native epsilon exponent in that source. In the present wild cuspidal comparison it equals the standard Swan conductor. A lower critical level in a defining sequence is not the leading level r in this formula.

Proposition 22.1 (Necessary period divisibility)**.** *Suppose an actual type satisfying* (22.4) *is to match a source of rank n and Swan conductor $w > 0$. Its principal period and leading level necessarily satisfy*

$$\frac{n}{\gcd(n, w)} \mid e \mid n, \qquad r = \frac{ew}{n}. \tag{22.5}$$

These are necessary numerical conditions, not sufficient existence conditions for a simple stratum or a prescribed source.

Proof. The principal condition gives $e \mid n$. Matching (22.4) gives $nr = ew$, so $n \mid ew$. Writing $n = gn'$ and $w = gw'$ with $\gcd(n', w') = 1$ yields $n' \mid e$. Conversely these divisibilities make ew/n an integer. They say nothing about purity, intertwining, or character occurrence. □

For $(n, w) = (9, 24)$, the possibilities are $(e, r) = (3, 8)$ and $(9, 24)$. A maximal-order construction of period one cannot have this conductor. The earlier unramified depth-two candidate has Swan conductor 18 and ordinary conductor 27, not 24 and 33. Replacing the leading level eight by a critical level four in the period-three formula would create a different error rather than repair the comparison.

22.3 A self-twist check that survives conductor matching

A second necessary invariant is the torsion number, the order of the group of unramified self-twists. For a maximal simple type, the exact formula in [BK, 6.2.5] identifies that number as $n/e(K/F)$ for the actual parameter field K.

Lemma 22.2(Wild irreducibility excludes nontrivial unramified self-twists) **.** *If $\rho|_{P_F}$ is irreducible and an unramified character χ satisfies $\rho \otimes \chi \simeq \rho$, then $\chi = 1$.*

Proof. An intertwiner restricts to an endomorphism of the irreducible P_F representation, because χ is trivial on P_F. It is therefore a nonzero scalar. Applying the intertwining identity to every Weil element forces χ to be one. □

A period-three candidate may pass the first conductor test and still fail this second test if its actual maximal parameter field has residue degree three. That candidate has torsion number three, while the prescribed source has torsion number one. This is why the conductor equation does not complete the arithmetic attachment.

23 The rank-nine self-pair filter

23.1 The source invariant is computed on the unchanged source

Retain the original source over $F = \mathbf{F}_3((t))$,

$$\rho = A_1 \otimes A_2,$$

where A_h is the rank-three Heisenberg representation attached to the projective equation $x_h^9 + x_h = t^{-h}$, for $h = 1, 2$. The independent quotient and wild-irreducibility hypotheses are those of the original source fixture, not newly imposed conditions on a replacement source.

After the recorded unramified splitting extension, take a nonzero constant v satisfying $v^9 = -v$. Then

$$y_v = v^3 x_h^3 + v x_h, \qquad y_v^3 - y_v = -v t^{-h}. \tag{23.1}$$

The cancellation follows by substituting $v^9 = -v$ and $x_h^9 + x_h = t^{-h}$. As v varies, these give the eight distinct nontrivial characters of the nine-element translation quotient. A nonzero polynomial of degree three cannot vanish on all nine translations, so the corresponding additive functional is nonzero. Distinct v give distinct functionals by the same argument. Since each pole order h is prime to three, the inherited Artin–Schreier ramification calculation gives break h.

Proposition 23.1 (Self-pair conductor of the retained source)**.** *Under the original independence and wild-irreducibility hypotheses,*

$$\mathrm{Sw}(\mathrm{End}\,\rho) = 152, \qquad a(\mathrm{End}\,\rho) = 232. \tag{23.2}$$

Proof. Conjugation on the nine Weyl operators for A_h gives each character of its projective quotient once. Hence $\mathrm{End}\,A_h$ is the sum of the trivial character and eight characters of break h after the unramified extension. In $\mathrm{End}\,A_1 \otimes \mathrm{End}\,A_2$, there is one trivial character, eight characters of break one, and $9 \cdot 8 = 72$ characters of break two. Independence retains the distinct quotient characters. Products of characters of different breaks have the larger break. The Swan conductor is therefore $8 + 2 \cdot 72 = 152$.

Unramified restriction preserves the Swan conductor. By wild irreducibility, the inertia-fixed endomorphisms are exactly the scalars. Thus $\dim(\mathrm{End}\,\rho)^{I_F} = 1$ over F itself. Equation (22.1) gives $81 - 1 + 152 = 232$. No assumption that Frobenius fixes each individual quotient character is needed. □

Source and status. [Q13] supplies the source and the ramification inputs. The displayed Artin–Schreier cancellation, character separation and conductor addition are expanded here. The recorded exact suite constructs the nine-element constant subspace inside $\mathbf{F}_{81}$ and checks all its additive character values. This finite check does not reprove local ramification theory.

23.2 The actual-order certificate cannot be omitted

The proposed rank-nine family has a complete principal order $\mathfrak{A}$ of period nine, radical $\mathfrak{P}$, and an actual final parameter field K/F that is totally ramified of degree nine. An intermediate field E has degree three. Write $\mathfrak{Q}$ for the radical of the induced order in the E-centralizer. The retained candidate certificate specifies leading level 24, lower even critical level $c = 2a$, and actual simple-character orders

$$\mathfrak{J} = \mathcal{O}_K + \mathfrak{Q}^a + \mathfrak{P}^{12}, \qquad \mathfrak{H} = \mathcal{O}_K + \mathfrak{Q}^{a+1} + \mathfrak{P}^{13}. \tag{23.3}$$

These equations are substantive construction inputs. The mere pair of numbers $(24, c)$ does not imply them.

The certificate also ensures that the K contribution adds no unexpected graded direction. In the notation of [Q13], minimality of the leading approximation γ and the correction $d \in \mathfrak{P}^{-c}$ give

$$[\gamma, \mathcal{O}_K] \subseteq \mathfrak{P}^{-c}, \qquad \mathcal{O}_K \subseteq \mathfrak{B} + \mathfrak{P}^{24-c}. \tag{23.4}$$

For $c = 4$ and $c = 10$, the last exponent exceeds thirteen. Through the grades used in (23.3), the K-grade therefore lies inside the three-dimensional centralizer grade. Its own dimension is one. This is an actual commutator and approximation assertion, not an integral splitting of a wildly ramified bimodule.

Let $I^+ = 1 + \mathfrak{P}$. The successive full grades have dimension nine, the intermediate grades dimension three, and the K-grades dimension one. It follows that

$$I = \log_q [I^+ : J^1] = 8(a-1) + 6(12-a) = 2a + 64, \tag{23.4a}$$

$$D = \log_q [J^1 : H^1] = 2 + 6 = 8, \qquad \dim \eta = q^4. \tag{23.4b}$$

Since K/F is totally ramified of full degree, the finite residue factor is a character and the torsion number is one.

23.3 A formal-degree calculation with its measure fixed

Normalize Haar measure by $\mathrm{vol}(\mathrm{GL}_9(\mathcal{O}_F)) = 1$ and central measure by $\mathrm{vol}(\mathcal{O}_F^\times) = 1$. Then

$$\mathrm{vol}(I^+) = \prod_{j=1}^{9} (q^j - 1)^{-1}, \qquad \mathrm{vol}(J) = (q-1) q^{-I} \mathrm{vol}(I^+).$$

The compact-induction degree formula and the Steinberg degree formula give

$$d(\pi) = \frac{q^4}{9 \,\mathrm{vol}(J)}, \qquad d(\mathrm{St}_9) = \frac{1}{9} \prod_{j=1}^{8} (q^j - 1),$$

so that

$$\frac{d(\pi)}{d(\mathrm{St}_9)} = \frac{q^9 - 1}{q - 1} \, q^{2a+68}. \tag{23.5}$$

The factor nine comes from the actual noncompact extension $[K^\times J : F^\times J]$. Omitting it before comparing the measures changes the calculation. An unramified twist may be used to unitarize solely for the formal-degree comparison. That operation does not replace the prescribed noncompact scalar in the construction, and it leaves the self-pair unchanged.

The primary all-characteristic formulas [AP, Theorems 6.5 and 6.9] use the torsion number r_t and an auxiliary parameter δ:

$$\frac{d(\pi)}{d(\mathrm{St}_n)} = r_t \frac{q^n - 1}{q^{r_t} - 1} q^{(r_t - n + \delta)/2}, \qquad a(\pi^\vee \times \pi) = \delta + n^2 - r_t. \tag{23.6}$$

These are the named classical inputs here, not a conjectural formal-degree identity. With $n = 9$, $r_t = 1$ and (23.5), they give

$$\delta = 4a + 144 = 2c + 144, \qquad a(\pi^\vee \times \pi) = 4a + 224 = 2c + 224. \tag{23.7}$$

Certified profile	c	I	D	δ	Self-pair a
First repair	10	74	8	164	244
Deeper repair	4	68	8	152	232
Retained source	—	—	—	152	232

Corollary 23.2 (A necessary selection inside this certified family)**.** *If a member of the actual-order family* (23.3) *is to match the retained source under a correspondence preserving the ordinary self-pair conductor, then* $c = 4$*. The critical-ten member is excluded. The critical-four member passes this invariant, but the comparison does not prove its source identification.*

Proof. Equate (23.2) and (23.7). The equation $2c + 224 = 232$ has the unique solution $c = 4$. Equality of this one invariant supplies no character-occurrence or full pair-factor identity. □

Source and status. [Q13] provides the conditional actual-order certificate and calculation. [AP, formulas (17), (29), Theorems 6.5, 6.9 and Remark 6.7] were checked at their stated measures and field scope. This block does not newly construct the proposed corrected stratum. It preserves the requirement to prove its actual order equations before applying the numerical filter.

24 Why pair conductors do not determine quotient capacity

24.1 The parity identity behind the graded calculation

For the totally ramified complete-order sector, write the successive graded centralizer dimensions as

$$n = b_0 > b_1 > \cdots > b_r = 1, \qquad \Delta_j = b_{j-1} - b_j.$$

Assume the actual defining sequence and its order compatibility give descending positive levels ν_j and the grade-count formulas

$$I = \sum_{j=1}^{r} \Delta_j \left\lceil \frac{\nu_j}{2} \right\rceil - (n-1), \qquad D = \sum_{j=1}^{r} \Delta_j \, \mathbf{1}_{2|\nu_j}. \tag{24.1}$$

Here $I = \log_q[I^+ : J^1]$, while $D = \dim_{k_F}(J^1/H^1)$. The different floor and ceiling thresholds are essential. In this torsion-one, full-degree setting, the same formal-degree calculation gives

$$a(\pi^\vee \times \pi) = n^2 - 1 + (n-1) + 2I + D. \tag{24.2}$$

As in the rank-nine calculation, the formula is conditional on actual orders. A list of integers is not a certificate of their arithmetic existence.

Proposition 24.1 (Parity cancellation)**.** *Under* (24.1),

$$(n-1) + 2I + D = \sum_{j=1}^{r} \Delta_j \nu_j.$$

Consequently

$$a(\pi^\vee \times \pi) = n^2 - 1 + \sum_{j=1}^{r} \Delta_j \nu_j. \tag{24.3}$$

The weighted conductor expression does not retain D as a separate invariant.

Proof. For each integer ν,

$$2\left\lceil \frac{\nu}{2} \right\rceil + \mathbf{1}_{2|\nu} = \nu + 1.$$

Insert this identity into $2I + D$. Since $\sum_j \Delta_j = n - 1$, the constant terms reduce to $-(n-1)$. Adding $n - 1$ proves the result. □

This cancellation explains a concrete limitation of the earlier numerical tests. Changing the parity distribution can change the Heisenberg quotient while preserving the weighted sum in (24.3). More conductor precision does not recover information that the formula has already summed away.

24.2 The full dyadic profile

In the inherited full dyadic construction, let $n = 2^d$ and let $h_1 < \cdots < h_d$ be positive odd integers satisfying its separation requirements. The consecutive profile has

$$b_j = 2^{d-j}, \qquad \nu_j = 2^{d-j}\,3h_{d-j+1} \quad (1 \le j \le d). \tag{24.4}$$

All levels except the final one are even. Therefore

$$D = 2^d - 2. \tag{24.5}$$

A source consisting of d nondegenerate binary symplectic planes has dimension $2d$. The original maximal-type quotient is too small for $d = 1, 2$, has exactly the needed dimension for $d = 3$, and has strict surplus for $d \ge 4$.

d	Arithmetic rank	Source dimension	Host D	Original quotient
1	2	2	0	Insufficient
2	4	4	2	Insufficient
3	8	6	6	Equal capacity
4	16	8	14	Strict surplus
5	32	10	30	Strict surplus

The rank-two and rank-four enlarged arithmetic carriers in Part I are not counterexamples to the first two rows. Those repairs change the assembly carrier while recovering the same inducing representation. The table concerns the original specified maximal-type quotient. At rank eight, dimension ceases to be the obstruction, but an independently fixed quadratic marking can still obstruct the attachment.

24.3 Two rank-eight profiles with the same conductor data

The retained dyadic capacity note contains a sharper example. In the full rank-eight profile, use

$$(b_0, b_1, b_2, b_3) = (8, 4, 2, 1), \qquad (\nu_1, \nu_2, \nu_3) = (12h_3, 6h_2, 3h_1).$$

A second constructed profile skips the intermediate quartic field and has

$$(b_0, b_1, b_2) = (8, 4, 1), \qquad (\nu_1, \nu_2) = (12h_3, c), \qquad c = h_1 + 4h_2. \tag{24.6}$$

The integer c is odd and $2c < 12h_3$. Its last jump therefore contributes nothing to D. The full profile has $D = 6$, while the skipped profile has $D = 4$.

Both weighted sums in (24.3) equal

$$48h_3 + 12h_2 + 3h_1. \tag{24.7}$$

Indeed, the full profile has jump sizes $4, 2, 1$, and the skipped profile has jump sizes $4, 3$. The identity $3(h_1 + 4h_2) = 3h_1 + 12h_2$ supplies the equality.

For $(h_1, h_2, h_3) = (1, 3, 5)$, the distinction becomes completely explicit.

Quantity	Full profile	Skipped profile
Levels	$(60, 18, 3)$	$(60, 13)$
Principal period	8	8
Standard Swan conductor	60	60
Standard ordinary conductor	68	68
Torsion number	1	1
I	133	134
D	6	4
Heisenberg dimension over $q = 2$	8	4
δ	279	279
Ordinary self-pair conductor	342	342

The compensating change is $2 \cdot 133 + 6 = 2 \cdot 134 + 4$. The three necessary invariants (standard conductor, torsion number, and self-pair conductor) cannot force the full six-dimensional quotient.

24.4 The skipped profile is an arithmetic claim, not an integer exercise

The inherited construction of (24.6) works over $\mathbf{F}_2((t))$. Put $q_0 = 3h_3$ and take $\alpha^2 + \alpha = t^{-q_0}$. A uniformizer z of the quadratic field satisfies

$$z = t^{(q_0+1)/2}\alpha, \qquad z^2 + t^{(q_0+1)/2}z + t = 0.$$

Over that field, the polynomial $X^4 + zX + z$ is separable and Eisenstein. If e is a root, let C be multiplication by e^{-c} in the quartic centralizer. The source note chooses a filtered corestriction lift using $s(P_0) = t/z^2$ and takes a correction proportional to $(z^2/t)CP_0$. The actual final parameter is the literal corrected matrix, not an assumed compositum of the approximation fields.

The conclusion that the correction supplies the claimed defining sequence uses the retained refinement argument and its strict inequalities. The conductor and capacity audit examines the consequences of that argument. It does not rerun a skipped-profile matrix constructor or recertify every step of the arithmetic field proof. Exact integer tests do not replace those inputs.

Nor is this example a counterexample to H5. Full local pair factors are substantially more information than the three numbers above. The example shows that those three numbers alone do not identify the needed native quotient, and that a source-defined candidate cannot be declared to contain a prescribed type merely because it passes them.

Source and status. [D167] supplies the actual profile formulas and the skipped-degree construction. Proposition 24.1 and the numerical comparison are fully derived here. The exact suite checks the example and 120 positive odd triples, as well as general parity identities. The underlying arithmetic construction retains its inherited ordinary-proof status.

25 The source-defined arithmetic candidate and its inputs

25.1 Do not identify two candidates by notation

The direct matrix recursion and the classical arithmetic-induction route start with different data. The first produces an actual type satisfying a restricted construction contract. The second starts from a complete source character and uses classical induction and descent to produce a candidate. We denote the latter

by $\pi_{\mathrm{BH}}(\rho)$ in this section. This notation does not identify it with any independently constructed compact type.

The audited source-defined branch is over $\mathbf{F}_3((t))$ and uses the original rank-nine source. It is distinct from the characteristic-zero continuation113 candidate discussed in Part II. Its existence through classical inputs does not prove the corrected-character occurrence question for that other candidate, nor does it place a prescribed native packet in the current one.

25.2 The degree-nine field does not initially have a cubic tower

Write $A = t^{-1}$, and retain the projective coordinates

$$x^9 + x = A, \qquad z^9 + z = A^2.$$

The source record takes $y = z - x - x^3$. In characteristic three,

$$y^9 + y = B, \qquad B = A^2 - A - A^3. \tag{25.1}$$

The inducing field is $M = F(y)$, of degree nine. Its retained Galois-closure analysis has translation subgroup $(C_3)^2$ with a C_4 action irreducible over $\mathbf{F}_3$. There is no cubic intermediate field over F. Therefore one cannot simply write two cubic automorphic-induction arrows over F without changing the base and proving the resulting descent.

Pass to the unramified extension $K = \mathbf{F}_{81}((t))$. The constants now split the translation action. Choose $a \neq 0$ with $a^9 = -a$ and i with $i^2 = -1$, and define

$$r = \frac{y^3 - a^2 y}{ia^3}, \qquad s = \frac{y}{a}.$$

Then

$$r^3 - r = \frac{B}{ia}, \qquad s^3 - s = ir. \tag{25.2}$$

To check the first equality, use $a^8 = -1$ to cancel the y^3 terms after cubing. The second follows immediately from the definition of r. These identities exhibit the cyclic cubic tower over K used by the classical route. The degree and regularity assertions remain part of the complete source fixture.

The inducing character is also complete data. The inherited record includes its ramified Artin–Schreier class, its unit values, and an actual unramified factor. Replacing it by the displayed pole order alone loses information needed for the resulting central character and noncompact extension. The noted different exponent 24 and character conductor 9 give the expected induced ordinary conductor 33, but do not determine that full character.

25.3 Induction, invariance, and descent are separate gates

The construction first applies cyclic automorphic induction at the two cubic stages over K. Cuspidality requires the recorded regularity at each relevant step. It then invokes the classical independence of the resulting monomial construction from an admissible cyclic tower. This independence is a substantial theorem. It is what allows a Galois conjugate of the chosen tower to be compared with the original construction. The fact that the source descends to F does not, by itself, make each arbitrarily chosen tower Frobenius invariant.

Only after this invariance has been established does the tame descent theorem apply. Its fiber is described by the unramified characters associated to K/F, here of order dividing four. The specified determinant of ρ selects the desired central character.

Lemma 25.1 (Unique fixed-center member in a coprime descent fiber)**.** *Let a descent fiber be a torsor for a cyclic character group X of order m. Suppose twisting a rank-n member by $\alpha \in X$ changes its central character by α^n. If $\gcd(n, m) = 1$, then any attainable target central character selects exactly one member of the fiber.*

Proof. The map $\alpha \mapsto \alpha^n$ is a bijection of X because multiplication by n is invertible modulo m. Given one member and an attainable ratio of central characters, there is a unique twist realizing that ratio. □

For $n = 9$ and $m = 4$, the required power map is invertible. This explains the central normalization rather than silently selecting a preferred descent. It differs from the fixed-center cubic ambiguity in continuation113, where the relevant power map is not coprime. The finite tests include both the unique and the nonunique cases.

25.4 What the primary-source audit does and does not establish

The selected audit uses three layers of classical material.

The 1999 monomial and tame-lifting statements. In [BH99], the definitions and properties of tame lifting in Section 1.3 identify the descent fiber twisting and central-character behavior. Section 2.3 gives the characteristic-zero monomial construction, including standard epsilon factors and pair L-functions and conductors. Its proof uses independence of the cyclic construction. Section 1.7 supplies the tame transfer rule for the relevant pair-conductor exponent. These statements were read at their exact pages.

The positive-characteristic extension. The original Section 2.3 is explicitly characteristic zero. The characteristic note on page eight of [BH13] states that the relevant earlier results extend to positive characteristic through the base-change and automorphic-induction work of [HL11]. That note was checked directly. The present audit therefore retains the positive-characteristic induction theorem as a named external input, rather than erasing the original hypothesis.

Source-access boundary. The full text of [HL11] was unavailable during the recorded audit. Its bibliographic record was checked, but the detailed statements III.2 and III.3, including the restricted pair statement used in [Q15], retain their inherited source-audit status. Other selected inputs to the arithmetic route were checked separately.

There are two particularly important exclusions. First, the effective construction in [BH13] treats the totally wildly ramified correspondence as given at its starting point. That paper's effective theorem is not used here as an independent solution of the same totally wild input. Its characteristic note is the relevant source interface. Second, the equality of pair conductors in [BH99, 2.3(vii)] is not equality of full pair epsilon factors. The introduction expressly distinguishes these strengths.

25.5 The resulting candidate still has a native attachment obligation

Under the recorded source and classical hypotheses, [Q15] obtains a cuspidal candidate with prescribed central character, torsion number one, standard Swan conductor 24, ordinary conductor 33, and ordinary self-pair conductor 232. The characterizing information required by the paper's H5 contract is not replaced by that list.

In particular, one cannot compare an essentially discrete source with a reducible principal series outside the scope of a pair theorem and then divide a pair conductor by nine to manufacture the standard conductor. Nor can a first-order calculation of a parameter on deep field units establish occurrence of a full simple character. The original unit domain and noncompact extension remain part of that question.

The next useful arithmetic step is therefore not another calculation of the same three integers. It is to construct or identify the actual native type and its complete character in this fixed candidate, and to compare its marked data with the supplied source. That is the boundary the finite results in the following sections are designed to expose.

Source and status. [Q15] records the source-defined constructor audit. The selected passages [BH99, printed pp. 15, 18, 23–25, 53–54] and [BH13, p. 8] were checked directly. [HL11] remains a named classical input whose detailed application has inherited audit status. A source-defined candidate cannot replace a separately fixed type candidate without proof of their identification.

26 A full dyadic extension remembers its squares

26.1 The extension criterion with a fixed center

Let V be a finite-dimensional vector space over $\mathbf{F}_2$, and consider central extensions

$$1 \longrightarrow Z \longrightarrow E_s \longrightarrow V \longrightarrow 1, \qquad 1 \longrightarrow Z \longrightarrow E_t \longrightarrow W \longrightarrow 1,$$

where Z is the same abelian group in both extensions. In applying this to a represented scalar pushout, the represented center and any already factored character kernel must be recorded. A pushout cannot recover invisible elements that have been discarded.

Fix a linear isomorphism $f : V \to W$. Choose an ordered basis (e_i) of V, lifts a_i in E_s, and lifts b_i of $f(e_i)$ in E_t. Put $s_i = a_i^2$ and $t_i = b_i^2$ in Z. Write $Z^2 = \{z^2 : z \in Z\}$.

Proposition 26.1 (Center-fixing extension criterion)**.** *There exists an isomorphism $F : E_s \to E_t$ fixing Z pointwise and lifting f if and only if*

$$[a_i, a_j] = [b_i, b_j] \quad \text{for every } i, j, \qquad s_i t_i^{-1} \in Z^2 \quad \text{for every } i. \tag{26.1}$$

If the generating lifts themselves must satisfy $F(a_i) = b_i$, then their squares must agree exactly, not merely modulo Z^2.

Proof. Any such isomorphism has $F(a_i) = \lambda_i b_i$ for some $\lambda_i \in Z$. Preservation of squares gives $s_i = \lambda_i^2 t_i$, while preservation of commutators gives the first condition. Conversely choose λ_i with these squares. Every group element has a unique normal form

$$z a_1^{v_1} \cdots a_r^{v_r}, \qquad z \in Z, \quad v_i \in \{0, 1\}.$$

Define its image by replacing a_i with $\lambda_i b_i$. Normal-form multiplication uses only central elements, the displayed squares, and the pairwise commutators. The conditions ensure that each multiplication relation is preserved. The inverse construction proves bijectivity. Taking every $\lambda_i = 1$ gives the final assertion. □

For $Z = \mu_2 = \{1, -1\}$, each square is independent of the chosen lift and defines a quadratic refinement

$$a_v^2 = (-1)^{q(v)}, \qquad q(v + w) = q(v) + q(w) + B(v, w), \tag{26.2}$$

where $(-1)^{B(v,w)}$ is the commutator. A center-fixing extension isomorphism over a specified symplectic map f therefore requires $q_t \circ f = q_s$.

For $Z = \mu_4$, multiplying a lift by i changes its square by -1. The unmarked square classes are consequently weaker. For $Z = \mathbf{C}^\times$, every square ratio has a square root and the unmarked criterion reduces to matching commutators. Neither statement deletes a separately marked finite subgroup or a prescribed family of lifts.

26.2 Dihedral and quaternion data inside the same scalar pushout

Consider the matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z_0 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

They satisfy $X^2 = Z_0^2 = 1$ and $XZ_0 = -Z_0X$. The group $H_D = \langle X, Z_0 \rangle$ is a dihedral group of order eight. The group $H_Q = \langle iX, iZ_0 \rangle$ is a quaternion group of order eight. On their common two-dimensional quotient, their refinements are

$$q_D(x,y) = xy, \qquad q_Q(x,y) = xy + x + y. \tag{26.3}$$

Marked extension	$q(e)$	$q(f)$	$q(e+f)$	$\sum_v (-1)^{q(v)}$
H_D	0	0	1	$+2$
H_Q	1	1	1	-2

Both scalar pushouts identify with

$$P = \{zX^aZ_0^b : z \in \mathbf{C}^\times,\ a,b \in \{0,1\}\}.$$

The marked subgroups remain different. In H_D, five nonidentity elements have order two and two have order four. In H_Q, only the central nonidentity element has order two and six have order four. In particular, no center-fixing automorphism of P carries one marked subgroup onto the other.

On the other hand, a quaternion source does embed into P through iX, iZ_0. Its squares and negative Gauss sign are retained. What fails is identifying it with the *independently chosen* dihedral subgroup H_D. This example separates a successful transported marking from a false assertion about a fixed native marking.

26.3 A full marked section contains more than a square table

Suppose a normalized section $\sigma : V \to E$ has product cocycle

$$\sigma(v)\sigma(w) = c(v,w)\sigma(v+w).$$

If an isomorphism must send each specified section element to a specified target section element, the two cocycles must agree after transport on *every* pair. Equality of polar forms and square tables only identifies their extension class, not all those pointwise choices.

For example, start with the ordered section $\sigma(x,y) = X^xZ_0^y$ of H_D. Change only the value at $e+f$ by a central sign. Every square remains unchanged because $(-1)^2 = 1$, but the product of the specified lifts at e and f now differs by a sign from the specified lift at $e+f$. An isomorphism fixing the two generating lifts cannot also implement this changed third assignment. The recorded tests retain this counterexample rather than treating it as a harmless convention change.

26.4 The valid scalar-pushout attachment contract

There is nevertheless a positive attachment statement in the restricted sector used in Chapter XI. Suppose an actual type, its full character and its Heisenberg representation η_W are supplied. Suppose the graph has one vertex, the prescribed stabilizer is trivial, and the source scalar pushout has the same

faithfully represented center and a nondegenerate quotient V with $\dim V \le \dim W$. No conflicting host-native finite marking is independently fixed.

Choose a symplectic embedding $f : V \to W$. Proposition 26.1 gives a scalar-center-fixing isomorphism of the source pushout with the restriction over $f(V)$. Carry every supplied finite subgroup, every lift, and every section value through this *same* isomorphism. Since it is a homomorphism fixing scalars, it preserves all squares, commutators and section products, not just their classes.

Writing $W = f(V) \perp U$, the finite Heisenberg representation factors as $\eta_V \otimes \eta_U$. A projector onto a line in the second factor gives the source Heisenberg sector inside the actual image algebra. The inherited recovery theorem then returns the unchanged full inducing representation and candidate, under its stated recovery hypotheses. There is no stabilizer equation because the stipulated stabilizer is trivial.

This is an attachment into the scalar extension of the actual type. It is not a claim that every source lift is an unscaled arithmetic matrix, nor that the transported subgroup equals an independently specified canonical host subgroup. A package that records only the unmarked ambient scalar extension loses precisely the information displayed by (26.3).

Source and status. [Q16] contains the complete marked scalar-pushout contract. The normal-form proof, the dihedral/quaternion calculations and the section counterexample are explicit finite arguments. Their application uses the already supplied actual type and inherited recovery theorem. This section does not construct those arithmetic inputs.

27 The sharp boundary for a fixed native refinement

27.1 Quadratic type and its Gauss sign

Let (V, B) be a nondegenerate alternating space over $\mathbf{F}_2$ of dimension $2s$, and let q be a quadratic refinement. In a symplectic basis (e_j, f_j),

$$q(x_1, y_1, \dots, x_s, y_s) = \sum_{j=1}^{s} \big(x_j y_j + q(e_j)x_j + q(f_j)y_j\big).$$

Its Gauss sum factors into plane sums:

$$G(q) = \sum_{v \in V} (-1)^{q(v)} = 2^s (-1)^{\sum_j q(e_j)q(f_j)}. \tag{27.1}$$

Define $\mathrm{Arf}(q) \in \mathbf{F}_2$ by the sign in (27.1). The sum is intrinsic, so this definition is independent of the symplectic basis.

Lemma 27.1 (Classification in the required finite setting)**.** *Two nondegenerate quadratic spaces over $\mathbf{F}_2$ are isometric if and only if they have the same dimension and Arf invariant. Both Arf types occur in every positive even dimension.*

Proof. The coordinate expression decomposes a space into nondegenerate planes. On a plane, the three refinements with Arf invariant zero are isometric to $H(x, y) = xy$, while the unique remaining type is $A(x, y) = xy + x + y$. This can be checked on its three nonzero vectors.

Two anisotropic planes can be replaced by two hyperbolic planes. In $A \perp A$, the vector $u = e_1 + e_2$ is nonzero and isotropic. Choose v with $B(u, v) = 1$. Replacing v by $v + q(v)u$ makes it isotropic too, so u, v span a hyperbolic plane. Its nondegenerate orthogonal complement is a plane whose Gauss sign is

positive by multiplicativity of (27.1). It too is hyperbolic. Pairing all anisotropic planes leaves either only hyperbolic planes or one anisotropic plane. The parity of the remaining anisotropic planes is exactly the Arf invariant. These normal forms prove classification and existence. □

In dimension six, (27.1) gives $G(q) = +8$ or -8. The numbers of zero values are respectively

$$\#q^{-1}(0) = \frac{64+8}{2} = 36, \qquad \#q^{-1}(0) = \frac{64-8}{2} = 28. \tag{27.2}$$

These counts include the zero vector. The suite verifies both refinements on all 64 vectors. They supply a concrete signature of the distinction that dimension alone misses.

27.2 One spare plane is enough only when the embedding is free

Proposition 27.2 (Fixed-refinement capacity criterion)**.** *Let (V, q_V) and (W, q_W) be nondegenerate quadratic spaces of dimensions $2s$ and $2R$. When the embedding may be freely chosen, a quadratic isometric embedding $V \to W$ exists exactly when*

$$R \geq s, \qquad R = s \implies \operatorname{Arf}(q_V) = \operatorname{Arf}(q_W). \tag{27.3}$$

If $R > s$, either source Arf type embeds.

Proof. An embedding requires the dimension inequality. At equal dimension it is an isometry, so the Arf invariants must agree. If $R > s$, choose a nondegenerate space U of dimension $2(R-s)$ with

$$\operatorname{Arf}(q_U) = \operatorname{Arf}(q_W) + \operatorname{Arf}(q_V).$$

Both choices are available by Lemma 27.1. The space $V \perp U$ has the same dimension and Arf invariant as W, hence is isometric to W. Restrict that isometry to V. The equal-dimension sufficiency follows from the same classification. □

After the quadratic embedding is chosen, the preimage of its image in an actual μ_2-central extension of W has the source refinement. Proposition 26.1 supplies the extension isomorphism. Exact generating lifts can be matched when their squares and commutators match. Additional prescribed section values retain the stronger product condition from Section 26.

The spare-plane argument does not permit arbitrary changes to fixed depth placements, transport maps, or stabilizer actions. It proves existence of an otherwise free quadratic embedding. These qualifications are important even when the host has a large dimensional surplus.

27.3 The exact rank-eight consequence

At $d = 3$ in (24.5), the full dyadic quotient and the three-plane source both have dimension six. Suppose the source planes have Arf invariants $\epsilon_1, \epsilon_2, \epsilon_3$, and a particular actual host refinement has Arf invariant ϵ_W. Then the necessary and sufficient test for a *free* quadratic embedding is

$$\epsilon_W = \epsilon_1 + \epsilon_2 + \epsilon_3 \quad \text{in } \mathbf{F}_2. \tag{27.4}$$

There is no orthogonal complement in which to absorb a mismatch. For $d \geq 4$, the full profile has a positive-dimensional complement, so the finite Arf obstruction disappears when the embedding is free and the actual host refinement is supplied.

This does not compute ϵ_W for a particular arithmetic type. A Heisenberg commutator quotient alone supplies an alternating form. It does not canonically select a finite two-element-center extension or its square refinement. Determining an independently required native refinement and its lift phases for the chosen candidate remains an arithmetic obligation.

Conversely, under the transported scalar-pushout contract of Section 26, the host refinement is part of the transported output rather than independently prescribed. Every source marking can then be retained in the same scalar extension once the symplectic capacity and recovery hypotheses hold. There is no contradiction between this positive statement and (27.4). They quantify over different data.

Source and status. The capacity criterion was recorded in [Q16]. Its finite classification input and proof are made explicit here. The rank-eight deduction combines that criterion with [D167]. The block does not assign a canonical native Arf invariant to an actual arithmetic candidate that has not supplied one.

28 An explicit correction for a quadratic marking defect

28.1 The discrepancy is a vector, not an unspecified search

Suppose a symplectic embedding $f : (V, B_V) \to (W, B_W)$ has already been supplied. Put

$$q'(v) = q_W(f(v)), \qquad \ell(v) = q'(v) + q_V(v).$$

The two quadratic refinements have the same polar form, so ℓ is linear. Nondegeneracy gives a unique vector $a \in V$ such that

$$\ell(v) = B_V(a, v) \quad \text{for every } v. \tag{28.1}$$

Finding a is therefore one linear solve over $\mathbf{F}_2$. It is not a search over all symplectic automorphisms.

Lemma 28.1 (Arf change under a linear defect)**.** *With the above notation,*

$$\mathrm{Arf}(q') + \mathrm{Arf}(q_V) = q_V(a). \tag{28.2}$$

Proof. The refinement identity gives

$$q_V(v) + B_V(a, v) = q_V(v + a) + q_V(a).$$

Translation by a permutes V. Summing the signs yields $G(q') = (-1)^{q_V(a)} G(q_V)$. Equation (27.1) identifies the two signs. □

Define the transvection

$$T_a(v) = v + B_V(a, v)a. \tag{28.3}$$

It is symplectic and satisfies $T_a^2 = 1$. For $a = 0$ it is the identity.

Proposition 28.2 (One-transvection quadratic correction)**.** *If $q_V(a) = 0$, then*

$$q'\big(T_a(v)\big) = q_V(v) \quad \textit{for every } v. \tag{28.4}$$

Thus $f \circ T_a$ is a quadratic isometric embedding. The condition is equivalent to equality of the Arf invariants of q' and q_V.

Proof. Write $b = B_V(a, v)$. Since $B_V(a, a) = 0$, one has $B_V(a, T_a v) = b$. Also

$$q_V(T_a v) = q_V(v) + b q_V(a) + b.$$

Adding the defect term $B_V(a, T_a v) = b$ gives

$$q'(T_a v) = q_V(v) + b q_V(a).$$

This equals $q_V(v)$ when $q_V(a) = 0$. For completeness, expanding the two cross terms in $B_V(T_a v, T_a w)$ makes them cancel in characteristic two, so T_a is symplectic. Its square is the identity because the same value b is added twice. Lemma 28.1 proves the equivalence of the condition. □

This yields a direct finite correction procedure: compute the defect on a basis, solve for a, test $q_V(a)$, and, when it vanishes, compose with T_a. The test suite verifies the formula for all linear defects in dimensions two, four and six, against every vector. It also checks the symplectic identities independently.

28.2 Equal Arf does not validate an unchanged map

Take $V = \mathbf{F}_2^2$ with $q_V(x, y) = xy$ and $q'(x, y) = xy + y$. Both refinements have Arf invariant zero. The identity map does not preserve the refinement: their values differ at $(0, 1)$. The defect vector is $a = (1, 0)$, which is isotropic. The transvection $(x, y) \mapsto (x + y, y)$ repairs the map.

The existence of that repair is not a proof that the original identity map worked. When the original embedding is prescribed by an arithmetic placement or a transport diagram, replacing it must itself be permitted and checked. A corrected finite embedding must never be silently substituted into a diagram that required the old map.

28.3 Preserving a previously correct subspace

There is a useful relative form. Let $U \subseteq V$ be a subspace on which the defect vanishes. Then $a \in U^{\perp}$, so $T_a(u) = u$ for every $u \in U$. Under the isotropy condition $q_V(a) = 0$, the correction therefore fixes all previously correct vectors in U pointwise.

This statement does not automatically preserve selected lifts above those vectors. It fixes the quotient embedding there. The extension criterion and section products still determine whether the chosen lifted correction respects all prescribed elements. The distinction is small in notation but essential in a full marked packet.

28.4 A conditional equivariant form

Suppose a finite group C acts symplectically on V and W, preserves q_V and q_W, and the supplied f is C-equivariant. Then

$$\ell(gv) = q_W(f(gv)) + q_V(gv) = \ell(v).$$

Nondegeneracy in (28.1) implies $ga = a$ for every $g \in C$. Consequently T_a commutes with the action. When $q_V(a) = 0$, the correction $f \circ T_a$ remains equivariant.

Corollary 28.3 (Equivariant finite correction under supplied actions)**.** *A supplied equivariant symplectic embedding with matching pullback Arf invariant admits the explicit equivariant quadratic correction of Proposition 28.2. If its defect already vanishes on a prescribed subspace, the correction fixes that subspace pointwise.*

Proof. The invariance computation makes a fixed by C. Inserting $ga = a$ into (28.3) gives $gT_a = T_a g$. The subspace assertion follows from $a \in U^{\perp}$. □

These are finite statements conditional on actual actions and an initial equivariant embedding. They do not construct such actions inside an arithmetic normalizer. Nor do they show that an arbitrary quadratic embedding provided by a dimension argument can be chosen equivariantly. The relative correction is a way to finish a specified finite step, not a replacement for its input.

Source and status. This section develops an explicit elementary repair from the linear-defect observation in [Q16]. The Gauss-sum proof and transvection computation are given in full. They are ordinary finite deductions, not a new Lean result or a claim of a new general arithmetic existence theorem.

29 Full extension actions and the remaining phase boundary

29.1 The discrepancy after a quadratic embedding has been chosen

Even an equivariant quadratic embedding need not give the required equivariant isomorphism of the full finite extensions with every prescribed marking. This section isolates a conditional repair of the group-action part.

For clarity, take nondegenerate μ_2-central extensions of the same quotient dimension, or restrict a larger target extension to a supplied invariant quadratic subspace. Suppose C acts genuinely on the source and target extensions, fixes the center, and induces the same quotient action after identification. Let $F : E_s \to E_t$ be a center-fixing extension isomorphism inducing that quotient identification. Such an F is supplied by the quadratic extension criterion, but is not initially assumed equivariant.

An automorphism of E_t that fixes its center and quotient is uniquely of the form

$$u_\chi(x) = (-1)^{\chi(\bar{x})} x, \qquad \chi \in \mathrm{Hom}_{\mathbf{F}_2}(W, \mathbf{F}_2). \tag{29.1}$$

Indeed, the ratio between the image of x and x is a central sign depending only on $\bar{x}$, and preservation of multiplication makes it additive. Conversely every such character gives an automorphism. These vertical automorphisms form the additive $\mathbf{F}_2$-space $A = \mathrm{Hom}(W, \mathbf{F}_2)$.

Write α_g for the target action and $\beta_g = F\alpha_g^s F^{-1}$ for the transported source action. Their discrepancy is

$$\delta_g = \beta_g \alpha_g^{-1} \in A.$$

The action of C on A is conjugation by α_g. Using additive notation in A, the genuine group-action equations give

$$\delta_{gh} = \delta_g + g\delta_h. \tag{29.2}$$

Thus the discrepancy is an actual 1-cocycle, rather than an arbitrary list of mismatched operators.

29.2 An explicit odd-order averaging repair

Proposition 29.1 (Conditional odd-action correction)**.** *Under the preceding hypotheses, suppose $|C|$ is odd. Put*

$$u = \sum_{h \in C} \delta_h \in A. \tag{29.3}$$

Replacing F by $u \circ F$, with u interpreted through (29.1), makes the extension isomorphism C-equivariant.

Proof. Sum (29.2) over h. Left multiplication permutes C, so

$$u = |C|\delta_g + gu = \delta_g + gu$$

in the characteristic-two space A. Thus $\delta_g = u + gu$. Replacing F by uF changes its discrepancy to $u + \delta_g + gu$, which is zero for every g. □

The correction fixes the scalar center and the quotient. It may change a chosen lift by a central sign. If all generating lifts or an entire section are independently fixed, that modification may be forbidden. The valid conclusion is existence of an equivariant center-fixing extension isomorphism when these vertical corrections are allowed. It is not an equivariant pointwise identification of every independently prescribed lift.

The exact suite enumerates all cocycles for a C_3 action on a two-dimensional $\mathbf{F}_2$ module and checks the averaging formula. It also retains the opposite boundary. For C_2 acting trivially on a nonzero $\mathbf{F}_2$ module, a nonzero value of δ at its generator is a 1-cocycle. Every coboundary $u + gu$ is zero. No correction of this form can remove it. Odd order is therefore a substantive hypothesis, not a cosmetic simplification.

29.3 What has not been repaired by this finite sum

The result starts after the arithmetic host, actual finite actions, invariant quotient subspace and an extension isomorphism have been supplied. It does not construct them from the original source diagram. A native-normalizer obstruction can prevent the initial action from existing even when the later finite cocycle would be removable.

There is a further representation-level boundary. An action by automorphisms of a Heisenberg group preserves the isomorphism class of its representation with fixed nontrivial central character. Intertwining operators can therefore be chosen, but their products can carry scalar discrepancies. Passing from those projective operators to the specified honest phase operators is a separate problem. Proposition 29.1 removes a discrepancy in the action on the *finite extension group*. It does not prove vanishing of every scalar projective obstruction in a representation, nor does it choose the prescribed Gauss or transport phases.

The original full character domain, its central quasicharacter, the noncompact root, and all specified section products remain part of the arithmetic contract. A finite central sign correction is acceptable only when its interaction with that contract is explicitly checked.

29.4 A useful order of operations

The two finite corrections should be used in the following order. First supply an actual invariant host and a quotient-equivariant symplectic embedding. Compare its pulled-back quadratic refinement with the source. When the defect vector is isotropic, use the equivariant transvection to correct the quadratic embedding. Next lift to the finite extensions and compute the resulting action cocycle. When the stipulated hypotheses permit it, use the odd-order sum to correct that action. Finally check every prescribed lift, section, and honest representation phase that these corrections were not authorized to change.

Each stage has a certificate and a failure mode. Combining them into one phrase such as “equivariant attachment” would hide which input is missing. In the equal-capacity case, even the first quadratic stage needs the native Arf match. In a source problem with a fixed embedding, the transvection may be disallowed. In even-order action problems, the 1-cocycle may survive. In all cases, the same arithmetic candidate must remain the subject of the construction.

Source and status. The source marking note leaves nontrivial stabilizer actions outside its established attachment theorem. This section adds a conditional finite group-action correction with a complete averaging proof and explicit even-order control. It is not a claim to have constructed general arithmetic stabilizers or solved the honest phase equations.

30 Audited claim map and the next arithmetic obligations

30.1 What this continuation adds to the progress paper

The conductor branch has a single normalization map, a proof of the necessary period divisibility, an expanded source self-pair calculation, and a precise conditional rank-nine filter. The actual-order certificate remains visible. The full and skipped rank-eight profiles show why even several correct conductor invariants cannot establish the needed native quotient.

The classical induction branch identifies its field tower, tower-independence input, tame descent, and central normalization separately. It names the characteristic extension and the source-access limitation affecting its positive-characteristic input. This supports a selected input audit, not a new proof of all local correspondence results.

The marking branch contains the full center-fixing extension criterion, the dihedral/quaternion distinction, the sharp free-embedding capacity test, and the exact rank-eight Arf condition. The explicit transvection and odd-order action corrections then explain how two later finite discrepancies can be repaired when their actual inputs and allowed changes are supplied.

Statement	Status here	Boundary retained
Principal period and conductor	Ordinary derivation with [BH99] input	Does not construct a stratum
Source self-pair 232	Expanded inherited ramification argument	Original source independence is retained
Critical-four filter	Conditional actual-order consequence	Passing is not type occurrence or H5
Rank-eight profile separation	Audited inherited construction and exact grade calculations	Three invariants do not force the native profile
Classical source candidate	Selected primary-interface audit	Detailed [HL11] reread unavailable in this block
Fixed-center extension criterion	Complete finite proof	Full sections can impose additional pointwise relations
Free quadratic embedding	Complete finite proof	Requires the actual host refinement as input
Transvection correction	Complete finite proof and exact tests	May change a preassigned embedding
Odd-action correction	Complete conditional finite proof	Does not construct arithmetic actions or honest phases
Native rank-eight refinement	Explicit remaining arithmetic obligation	Its value is not inferred from capacity
General realization and H5	Open in this project	No replacement by profile matching

30.2 A certificate for the next native-marking step

For a fixed arithmetic candidate at rank eight, a useful next certificate must identify the actual finite central extension or specified finite subgroup in its scalar pushout. It must give the chosen lift of each of the six symplectic basis vectors and compute its square under the full central character. These six bits determine the refinement once the commutator basis is fixed. Their three pairwise products determine the Arf sign through (27.1).

The certificate must also say whether the target marking is independently fixed or is allowed to be transported from the source. If it is fixed, compute the defect of the required quotient map. A vanishing defect directly certifies that map. An isotropic nonzero defect supplies the transvection only when changing the map is allowed. Every additional section element and transport lift then needs its actual multiplication check.

This task cannot be replaced by evaluating a canonical-looking quadratic form on an abstract six-dimensional space. That would choose new data rather than calculate the data of the specified arithmetic object. It also cannot be replaced by choosing another candidate with the same conductors. The candidate and full inducing character must remain fixed throughout.

30.3 Verification and reproducibility boundaries

The conductor and marking exact suite contains 23 test methods. Its conductor tests cover the normalization table, grade dimensions, period divisibility, the rank-nine formal-degree identities, the rank-eight profiles and the finite field used for the projective source characters. Its marking tests cover extension group laws, square and section controls, Gauss signs, all labeled symplectic bases in dimension four, all linear-defect corrections through dimension six, relative and equivariant corrections, and odd versus even action cocycles.

The research atlas follows these distinctions through normalized conductors, rejected and surviving rank-nine candidates, equal invariants with unequal capacity, transported and native markings, the equal-capacity Arf obstruction, and the two conditional finite corrections. These illustrations retain the general conjecture as an open realization problem.

FURTHER STUDY

Visual studies of arithmetic realization

The plates give worked constructions and counterexamples, with explicit links to the corresponding arguments and their hypotheses.

READING ROUTES

Research atlas 1552

A reading route through the argument

This atlas accompanies the arithmetic realization studies following Chapter XI. Each plate combines a mathematical graphic with an explicit claim boundary and a route back to those studies. The figure data are exact finite calculations or labeled dependency diagrams, not invented measurements.

The central distinction. A finite packet, an actual arithmetic host, and the same candidate's full local factors are different objects. The first two plates fix the quantifiers. The rest explain what can be constructed, what a particular obstruction rules out, and which stronger inputs are still required.

Route	Mathematical purpose
XI.1–XI.5	The witness contract, balanced products, compact binders, and a complete finite corner obstruction.
XI.6–XI.10	Whole characters, trace depth, defining sequences, native quotient capacity, and enlarged-carrier recovery.
XI.11–XI.19	Conductor conventions, candidate filters, information loss, descent, Clifford dimensions, and occurrence/action obstructions.
XI.20–XI.26	Full finite extensions, section products, quadratic markings, the equal-capacity boundary, and transvection repair.
XI.27–XI.32	Genuine action correction, its even-order limit, honest phases, the missing native certificate, and the remaining endpoints.

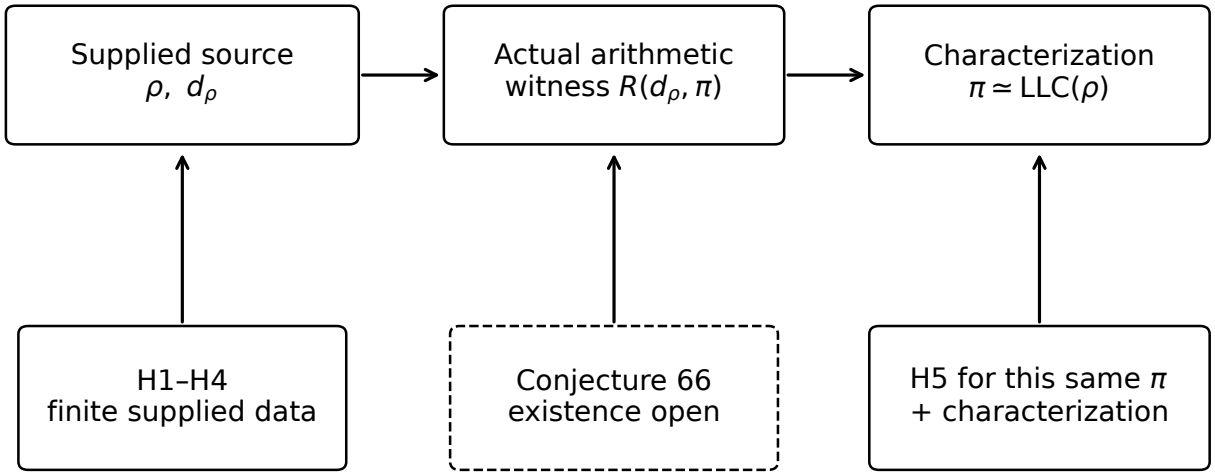
Evidence labels. An exact finite model verifies the displayed example. A conditional profile comparison retains the actual arithmetic construction hypotheses. A contract diagram records dependencies but does not supply their inputs. Ordinary general proofs remain in the companion study. None of these labels denotes a new Lean run or external peer review.

The source line on each plate links to its argument in this book. The final concordance lists the corresponding book pages.

PLATE XI.1

The three obligations

Finite comparison, arithmetic existence, and characterization have different inputs.



The source is a wild Weil atom ρ with the supplied datum d_ρ. H1–H4 include the declared finite break decomposition, symplectic bridges, central extensions and phases, and coherent transports. They describe an admitted input. They do not assert that every possible source arrives with canonical data.

The middle box is an actual arithmetic witness $R(d_\rho, \pi)$. It carries the prescribed packet and recovers the same supercuspidal representation, including the central character and noncompact completion. The final arrow additionally needs H5 for that particular π and the applicable characterization theorem.

$$\forall \rho \in \mathcal{A} \quad \exists \pi\ R(d_\rho, \pi) \qquad \text{is the general realization endpoint.}$$

An arrow is a dependency, not a declaration that its input has been constructed. Restricted witnesses and finite comparison theorems remain useful even while this quantified endpoint is open. Historical obligation 66 is not a measure of the amount of mathematics still required.

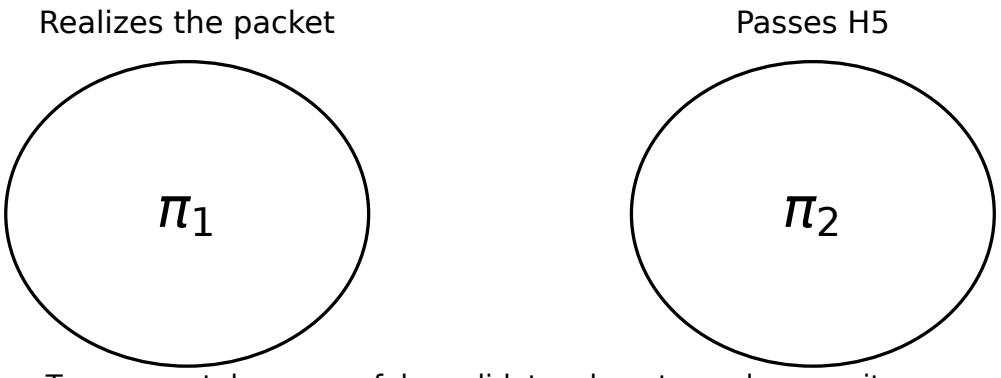
Evidence: Contract diagram.

Companion source: Part I, Section 1. Book page 1486.

PLATE XI.2

One candidate must satisfy both conditions

Separate existential statements cannot be joined by changing the witness.



Two separately successful candidates do not supply one witness.

$$\{\pi_1\} \cap \{\pi_2\} = \varnothing$$

Let $A = \{\pi : R(d, \pi)\}$ and $B = \{\pi : H5(\pi)\}$. The desired comparison uses a member of $A \cap B$. The diagram is an exact two-element countermodel: $A = \{\pi_1\}$ and $B = \{\pi_2\}$, with $\pi_1 \neq \pi_2$. Each condition has a witness. Their conjunction has none.

$$(\exists \pi\, R(d, \pi)) \wedge (\exists \pi\, H5(\pi)) \quad \not\Rightarrow \quad \exists \pi\, (R(d, \pi) \wedge H5(\pi)).$$

There is a second pitfall. An implication about every realizer can be true when there are no realizers. A comparison interface whose structure already contains an existence field does not construct that field merely by using it.

These finite sets are logical illustrations, not a claim that the arithmetic realization and H5 classes are actually disjoint. Their purpose is to make a change of candidate visible before it enters a proof.

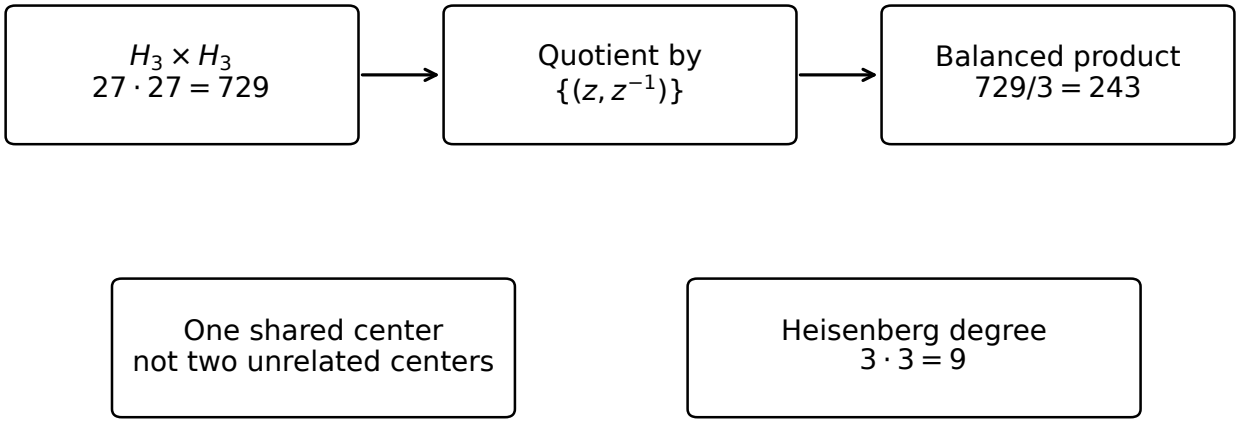
Evidence: Exact logical countermodel.

Companion source: Part I, Section 1.2. Book page 1486.

PLATE XI.3

A balanced central product

The shared center is identified once, while the orthogonal packet factors remain.



Orthogonal factors commute. The actual extensions are inputs.

Take two Heisenberg groups of order $3^3 = 27$. Their direct product has order 729. Quotient by the central subgroup $\{(z, z^{-1})\}$ of order three. The balanced central product has order $3^5 = 243$, with a four-dimensional quotient over $\mathbf{F}_3$ and one common center.

$$(H_3 \times H_3)/\{(z, z^{-1})\} \quad \text{has Heisenberg degree} \quad 3^2 = 9.$$

The figure's order calculation is checked by enumerating every fiber of the multiplication model. Each fiber has exactly three elements. The representation degree agrees with the tensor product of two degree-three Heisenberg representations whose central characters match.

The general argument requires the actual factor extensions and commutation between distinct factors. A list of break labels does not supply either condition. Once those conditions hold, the balanced product removes the proposed extra mixed extension ambiguity. It does not construct an arithmetic host or identify independently fixed lifts.

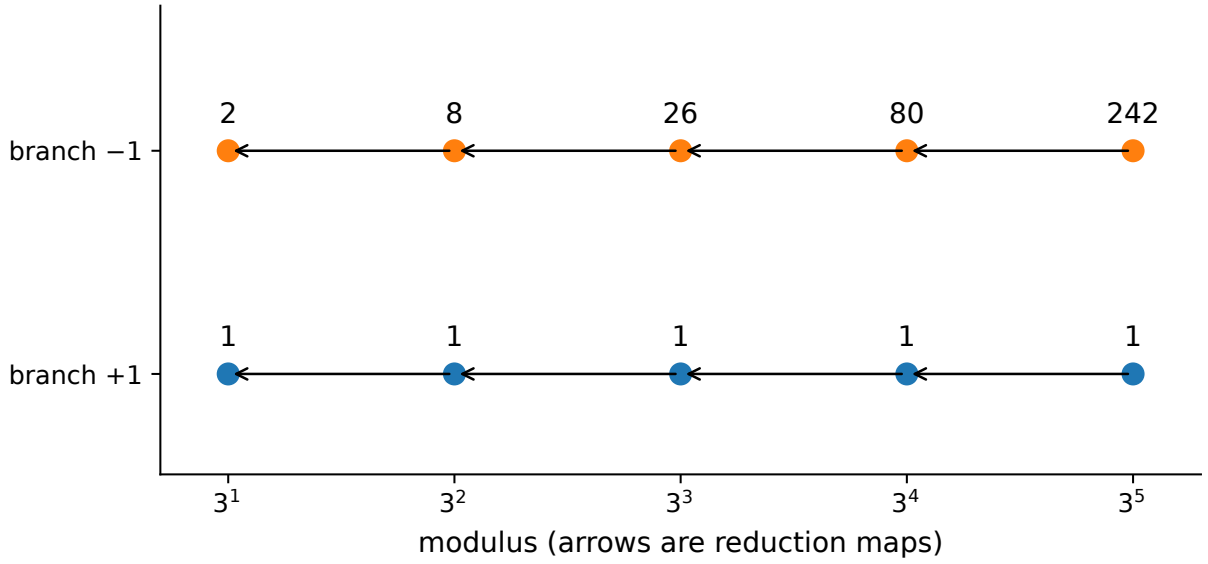
Evidence: Exact finite model.

Companion source: Part I, Section 3. Book page 1488.

PLATE XI.4

What a compact binder can do

Compatible solutions in one fixed compact space can survive every precision.



The points are all roots of $x^2 - 1$ modulo 3^n, for $1 \le n \le 5$. The labels are canonical residue representatives. The arrows reduce a root modulo 3^n to a root modulo 3^{n-1}. Both branches persist, corresponding to 1 and -1 in $\mathbf{Z}_3$.

For the general binder, fix a closed $X \subseteq \mathcal{O}^m$ and continuous constraints f_j. The sets

$$C_n = \{x \in X : f_j(x) \in \varpi^n \mathcal{O} \text{ for every } j \le n\}$$

are closed and nested. If every C_n is nonempty, compactness supplies a point satisfying every exact equation.

The example does not prove simultaneous finite solvability for the arithmetic problem. Even $x = 0$ and $x = 1$ have separately solvable congruences but no joint solution modulo three. Allowing the number of coefficients or pole bounds to grow also changes the compact model. Those missing hypotheses cannot be replaced by more isolated successful tests.

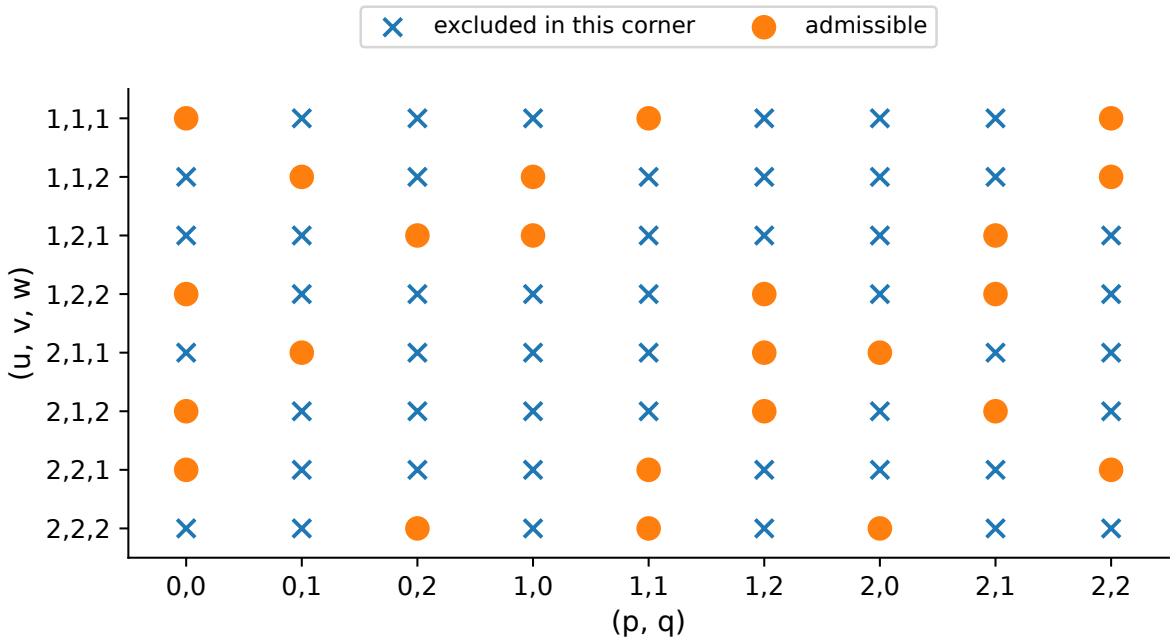
Evidence: Exact finite illustration.

Companion source: Part I, Section 5; Part II, Section 19.1. Book page 1490.

PLATE XI.5

The complete cubic corner spectrum

A finite obstruction can exclude a template without excluding every host.



Rows run through the eight nonzero triples $(u, v, w) \in \mathbf{F}_3^3$. Columns run through all nine pairs (p, q). A circle marks a solvable target in the specified corner. A cross marks an excluded target.

$$u = cd, \quad v = ab, \quad w = bc, \quad p = c(a - h), \quad q = b(d - h).$$

Elimination gives the exact criterion $p + u = w(q + v)$. Thus precisely 24 of the 72 targets are admissible. The data enumerate all five coefficients over $\mathbf{F}_3$ and independently compare their image with this criterion.

The target $(1, 1, 1, 2, 2)$ becomes possible when the fifth coefficient h is allowed. The target $(1, 1, 1, 1, 2)$ remains impossible. The inherited structural calculation explains why the larger corner has the same decisive leading condition. Its scope is that corner. The later arbitrary-exponent binder changes the construction rather than claiming that this negative result vanished.

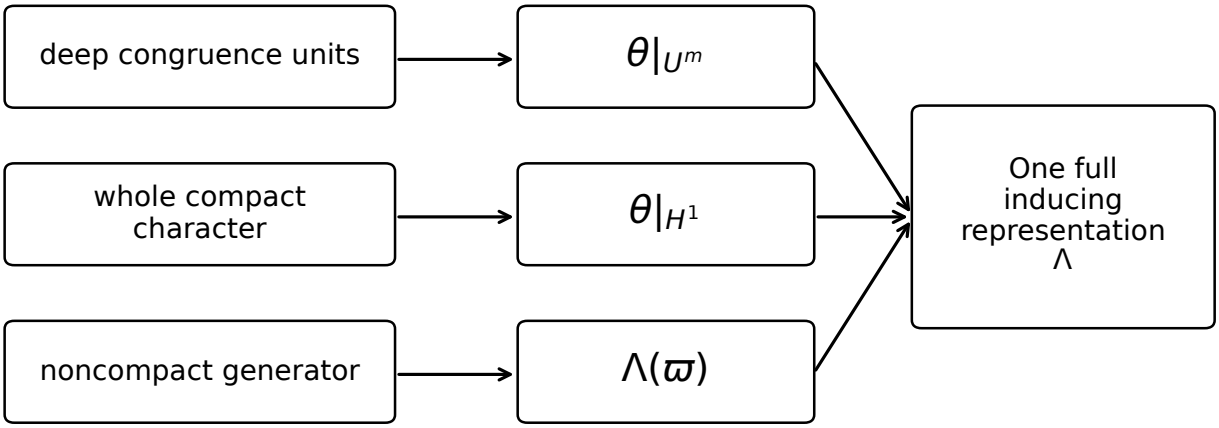
Evidence: Exhaustive finite spectrum.

Companion source: Part II, Section 19.2. Book page 1527.

PLATE XI.6

Do not truncate the full character

Deep values, the full compact domain, and a noncompact generator are distinct certificates.



A deep equality cannot fill in the two larger-domain certificates.

A trace expression first describes a character on sufficiently deep congruence units. The full simple character also has values on the larger native unit domain. A full inducing representation retains further central and noncompact data. These are successive obligations, not three interchangeable notations for the same calculation.

The direct character construction transfers the whole preceding character, applies the actual permitted correction, and then checks the intersection needed for shallow extension. The correction exhibits why noncompact normalization matters. In exponents of ninth roots of unity,

$$\nu(\varpi) = \zeta_9, \qquad \nu(\varpi_F/\varpi^3) = \zeta_3^2 \quad \Longrightarrow \quad \nu(\varpi_F) = 1.$$

Setting $\nu(\varpi) = 1$ instead gives ζ_3^2, changing the central character. The accompanying finite check verifies this root-of-unity identity only. The full unit character, intertwining, and type-extension statements retain the arithmetic hypotheses developed in the progress paper.

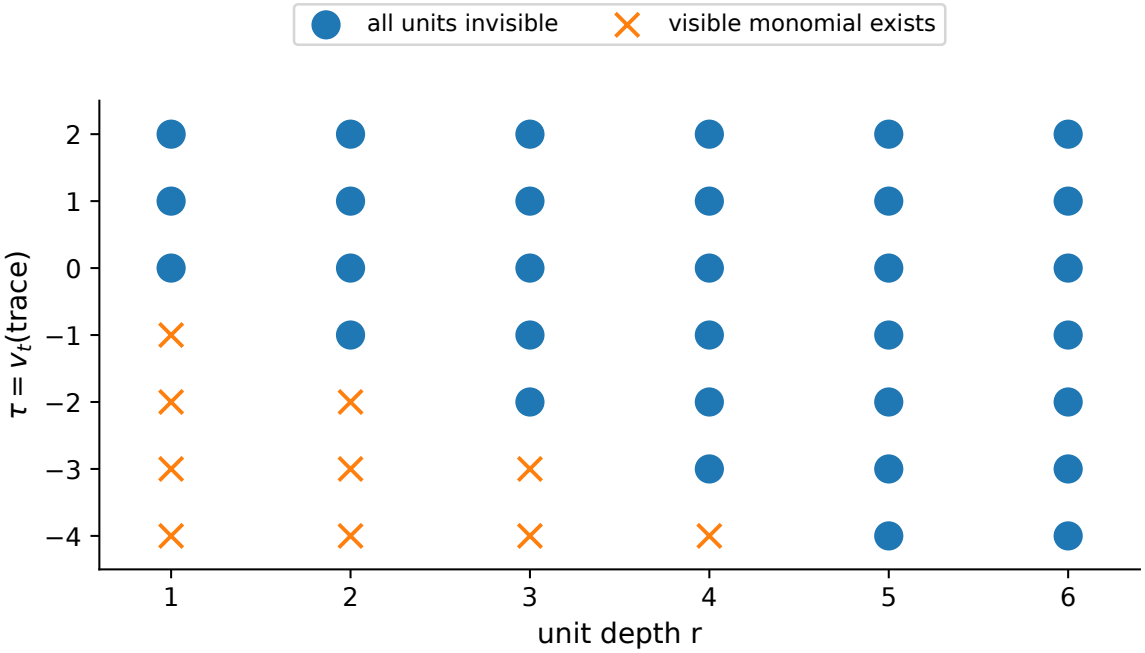
Evidence: Domain diagram.

Companion source: Part I, Sections 7.5–8.5; Part II, Section 18. Book page 1495.

PLATE XI.7

Depth can make native trace one invisible

The valuation inequality is the mechanism, not an assertion that the trace is zero.



In $\mathbf{F}_2((t))$, let the additive character read the coefficient of t^0. Model the trace coefficient by t^τ, and let the variable range over $t^r\mathcal{O}$. A circle means every such product has positive valuation and is invisible to the character:

$$\tau + r \geq 1.$$

At a cross, $t^{-\tau}$ belongs to the allowed domain and produces the visible product 1. This gives an explicit negative control, rather than treating failure of a sufficient bound as proof of visibility without a witness.

The native trace-one row is $\tau = 0$. It is invisible at every positive unit depth in the displayed range. Native trace one, zero trace after amplification, and the trace of the correction remain different quantities in the arithmetic recursion. The exact floor in the correction depth is retained in the written proof. This monomial picture explains the inequality but does not replace the full shallow-character extension argument.

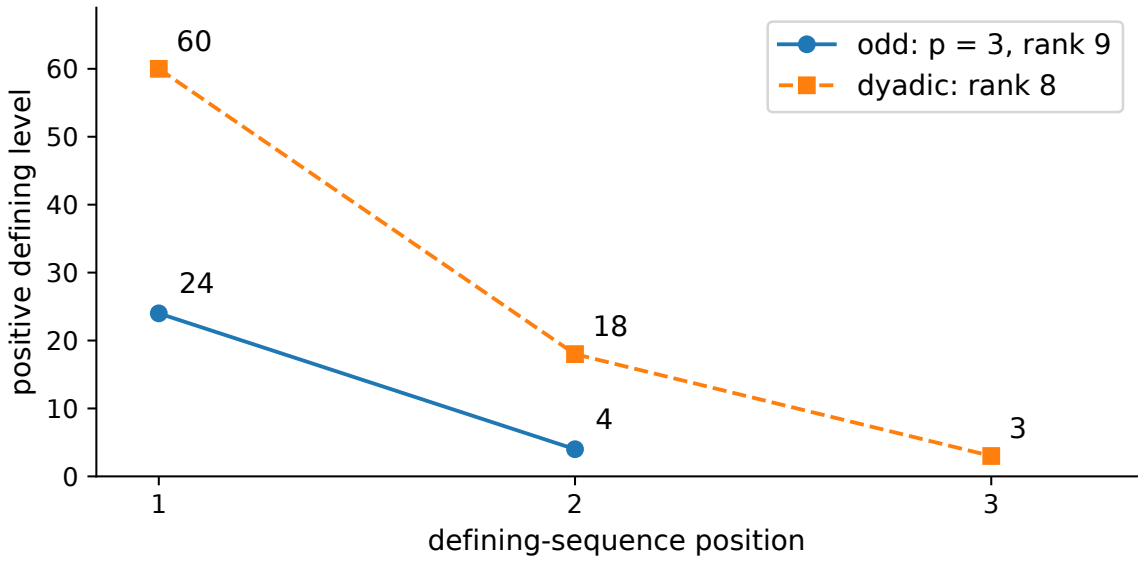
Evidence: Exact Laurent-monomial model.

Companion source: Part I, Section 8. Book page 1498.

PLATE XI.8

Two defining-sequence examples

The recursion records levels and field refinements, not just its final matrix size.



The odd example uses $p = 3$, $d = 2$, and $(h_1, h_2) = (1, 2)$. Its displayed levels are $(24, 4)$. The dyadic example uses $d = 3$ and $(h_1, h_2, h_3) = (1, 3, 5)$, giving $(60, 18, 3)$. Each position refers to that example's own defining sequence.

$$\nu_j = p^{d-j}(p+1)h_{d-j+1} \quad \text{or} \quad \nu_j = 2^{d-j}3h_{d-j+1}.$$

The two plotted lines are not empirical fits or samples of one common arithmetic family. They make the inherited level formulas legible. In the odd example, $24 > 3 \cdot 4 > 2 \cdot 4$ illustrates the separated scales used in refinement.

Supplying the levels does not prove that the literal corrected matrix generates the asserted maximal field. That conclusion uses the filtered corestriction lift and the precise refinement theorem. In the dyadic argument, separability from native trace one is established only after the full field degree has been obtained.

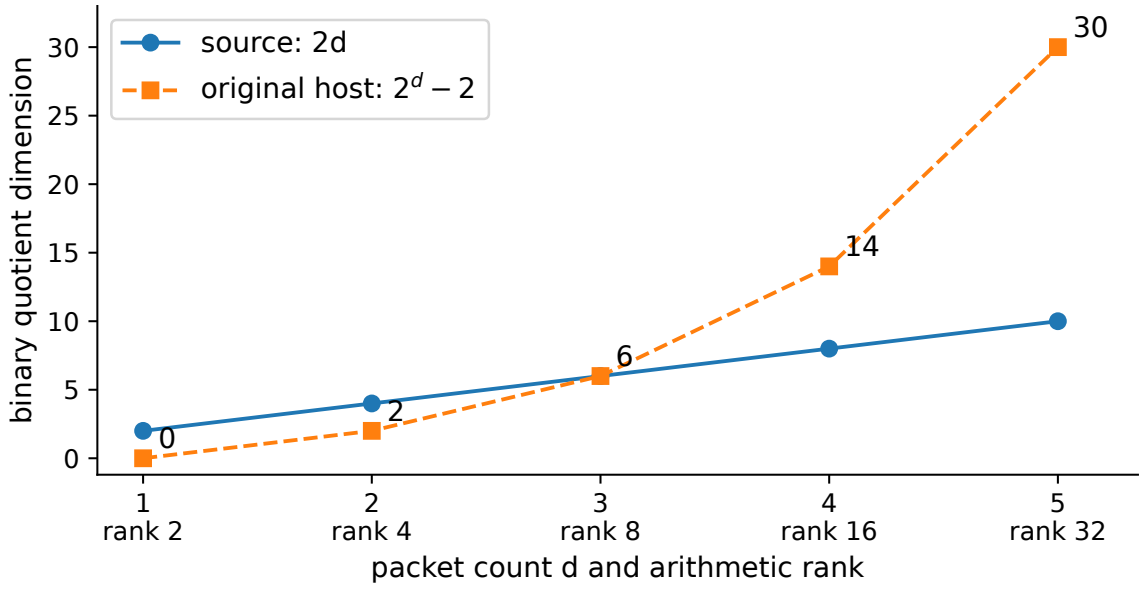
Evidence: Inherited formulas, exact evaluation.

Companion source: Part I, Sections 7.1–7.4 and 8.1–8.3. Book page 1492.

PLATE XI.9

Where the original dyadic quotient runs out of room

The rank-eight case is the first equality, not a low-rank surplus.



For d source planes, the binary source quotient has dimension $2d$. In the full dyadic profile the specified original maximal-type quotient has dimension $D = 2^d - 2$. The first five values are shown without normalizing away their difference.

d	1	2	3	4	5
$2d$	2	4	6	8	10
$2^d - 2$	0	2	6	14	30

Ranks two and four are too small in that original quotient. Rank eight has exactly six dimensions for six required dimensions. Only from rank sixteen is there a strict surplus in this family.

The later low-rank repair uses a larger actual assembly carrier and recovers the same representation. It does not change the first two numbers in this table. At equality, a fixed native quadratic refinement can still obstruct attachment. With spare dimension, the free quadratic embedding problem is easier, but prescribed actions, lifts, and transports remain separate constraints.

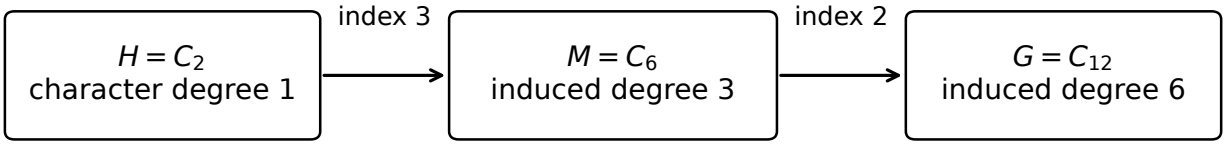
Evidence: Exact capacity formula.

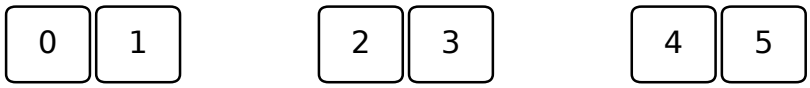
Companion source: Part I, Section 9; Part III, Section 24.2. Book page 1501.

PLATE XI.10

A larger inducing carrier can return the same object

Induction in stages changes the intermediate group without changing the result.





Six cosets, whether induced directly or in two stages.

The diagram is a finite analogue, not the actual local-field construction. Use $H = C_2 \subset M = C_6 \subset G = C_{12}$ and a character of H. Induction to M has dimension three, and induction onward to G has dimension six. Direct induction has the same six cosets, represented by $i + 2j$ with $i \in \{0, 1\}$ and $j \in \{0, 1, 2\}$.

$$\operatorname{Ind}_M^G \operatorname{Ind}_H^M \lambda \simeq \operatorname{Ind}_H^G \lambda.$$

The arithmetic recovery uses the corresponding compact-induction transitivity with the original full inducing representation Λ. Its larger intermediate carrier must be an actual permitted subgroup, and the original central character and noncompact extension must be retained.

The rank-two and rank-four proofs supply additional group, normalization, and character-orbit arguments before transitivity applies. The finite coset calculation isolates the recovery mechanism. It neither constructs those arithmetic groups nor proves irreducibility of a new local-field representation.

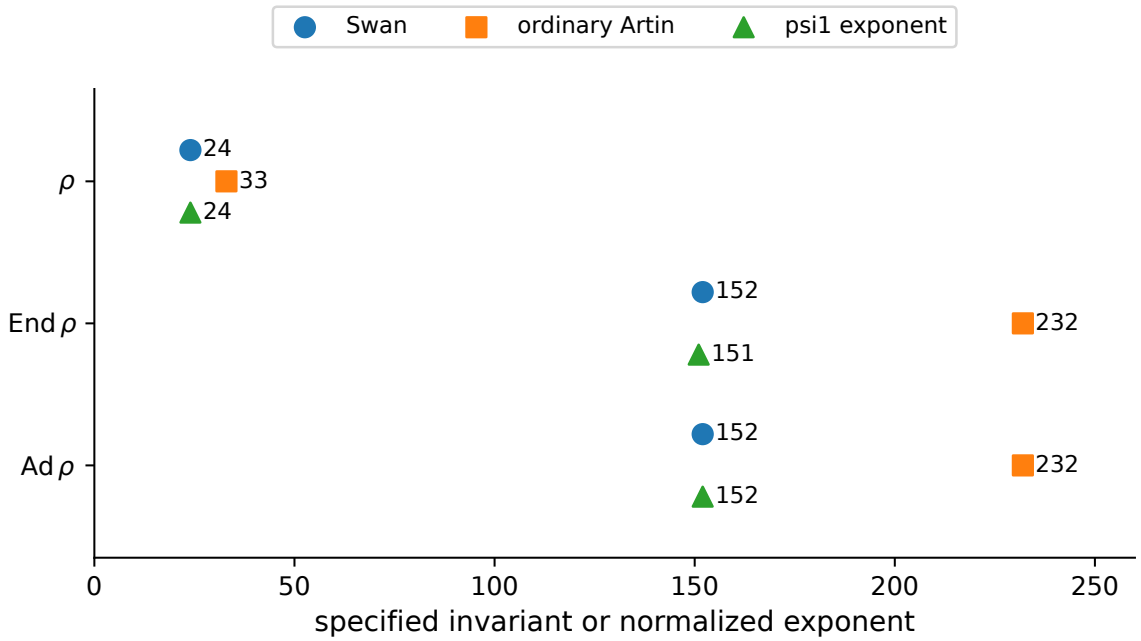
Evidence: Exact finite analogue of recovery.

Companion source: Part I, Section 10. Book page 1502.

PLATE XI.11

Three conductor columns are not interchangeable

The scalar summand changes an epsilon exponent even when the ordinary conductor is unchanged.



The ordinary conductor is $a(V) = \dim V - \dim V^{I_F} + \mathrm{Sw}(V)$. For the additive character ψ_1 trivial on $\mathfrak{p}_F$ but not on $\mathcal{O}_F$, the stated normalization gives $f_{\psi_1}(V) = a(V) - \dim V$.

V	$\mathrm{Sw}(V)$	$a(V)$	$f_{\psi_1}(V)$
ρ	24	33	24
$\mathrm{End}\,\rho$	152	232	151
$\mathrm{Ad}\,\rho$	152	232	152

Here $\mathrm{Ad}\,\rho$ means the trace-zero summand. The scalar summand in $\mathrm{End}\,\rho = \mathbf{C} \oplus \mathrm{Ad}\,\rho$ has zero ordinary conductor, but it contributes one to the dimension subtracted in the epsilon normalization.

The plot keeps the three definitions separate even when values happen to coincide. An exponent is still not the full epsilon factor. The arithmetic conductor inputs and additive-character convention are inherited from the audited progress paper, not freshly established by drawing these numbers.

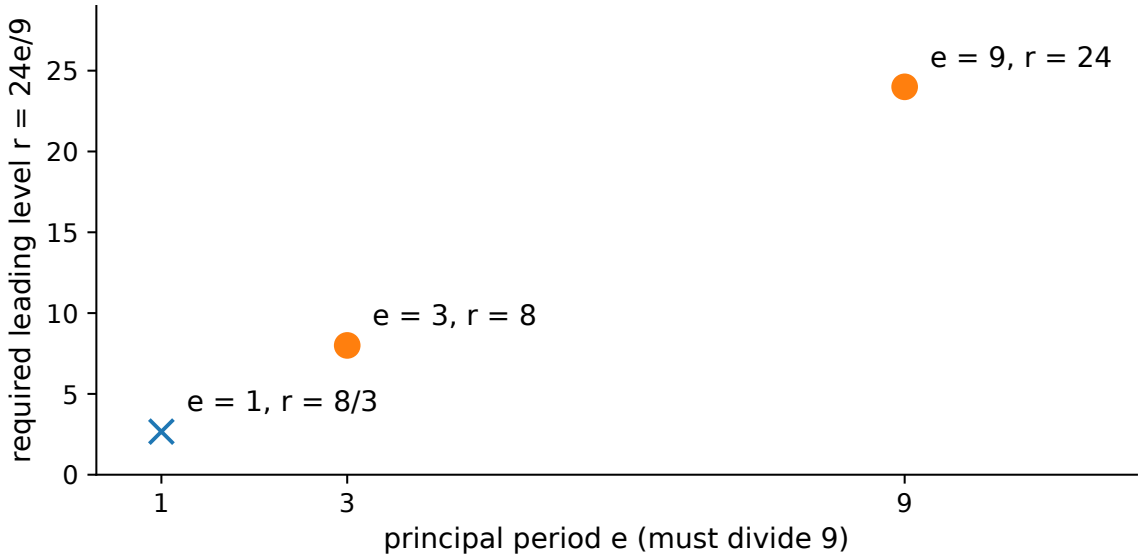
Evidence: Exact normalization worksheet.

Companion source: Part III, Section 22.1. Book page 1532.

PLATE XI.12

The principal-period filter

The period belongs to the actual order, and the level must be integral.



Under the stated actual-type conductor formula, a rank-n source with Swan conductor w requires $nr = ew$, where the principal period satisfies $e \mid n$. Therefore

$$\frac{n}{\gcd(n,w)} \mid e \mid n, \qquad r = \frac{ew}{n}.$$

For $(n,w) = (9,24)$, period one would demand the nonintegral level $8/3$. Periods three and nine leave $(e,r) = (3,8)$ and $(9,24)$. The cross is excluded numerically. The circles pass only this integrality condition.

A period-three candidate can still fail the independent torsion-number test when its actual maximal parameter field has residue degree three. The prescribed wild-irreducible source has no nontrivial unramified self-twist. Neither a convenient lattice period nor a lower critical level can be substituted for the actual period and leading level in the formula. A necessary numerical pass supplies no simple stratum or character occurrence by itself.

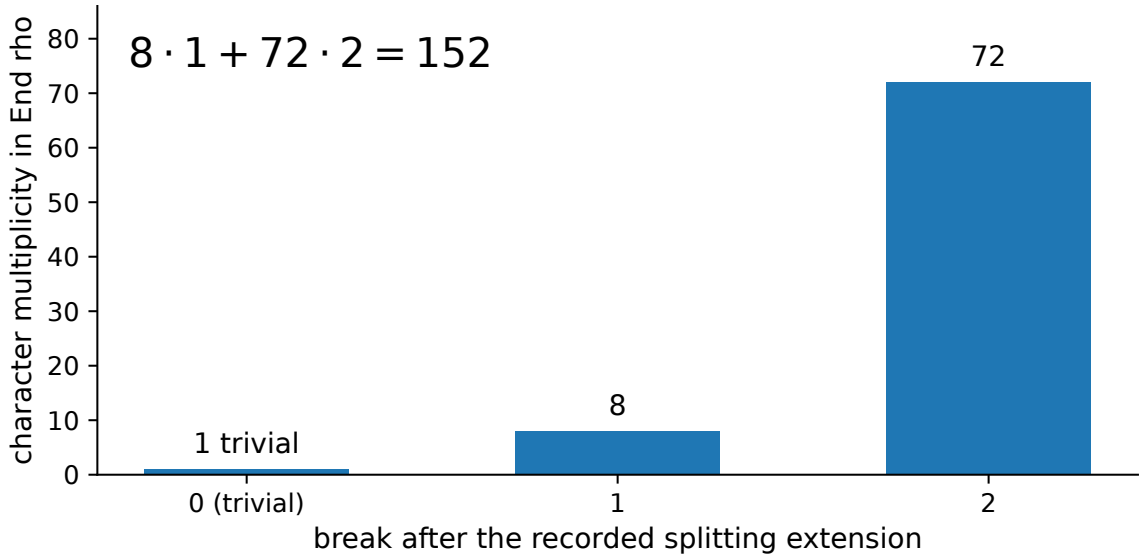
Evidence: Exact necessary filter.

Companion source: Part III, Section 22.2–22.3. Book page 1532.

PLATE XI.13

The rank-nine source has a visible break count

The self-pair conductor comes from the unchanged source and its actual independence assumptions.



For the recorded source $\rho = A_1 \otimes A_2$, each degree-three Heisenberg factor contributes one trivial conjugation character and eight nontrivial characters, at break one or two respectively. Under the original independence hypotheses, the tensor product contributes one trivial character, eight at break one, and seventy-two at break two.

$$\mathrm{Sw}(\mathrm{End}\,\rho) = 8 + 2 \cdot 72 = 152, \qquad a(\mathrm{End}\,\rho) = 81 - 1 + 152 = 232.$$

The last subtraction uses wild irreducibility: the invariant endomorphisms are exactly the scalars. It does not require Frobenius to fix every individual character after the splitting extension.

The histogram is computed by enumerating the 9×9 pairs and taking the larger of the two unequal breaks. The arithmetic assertions that these are the source's genuine characters, that their breaks are as stated, and that the quotient factors are independent remain the inherited source hypotheses. Their role is exposed rather than replaced with a finite counting experiment.

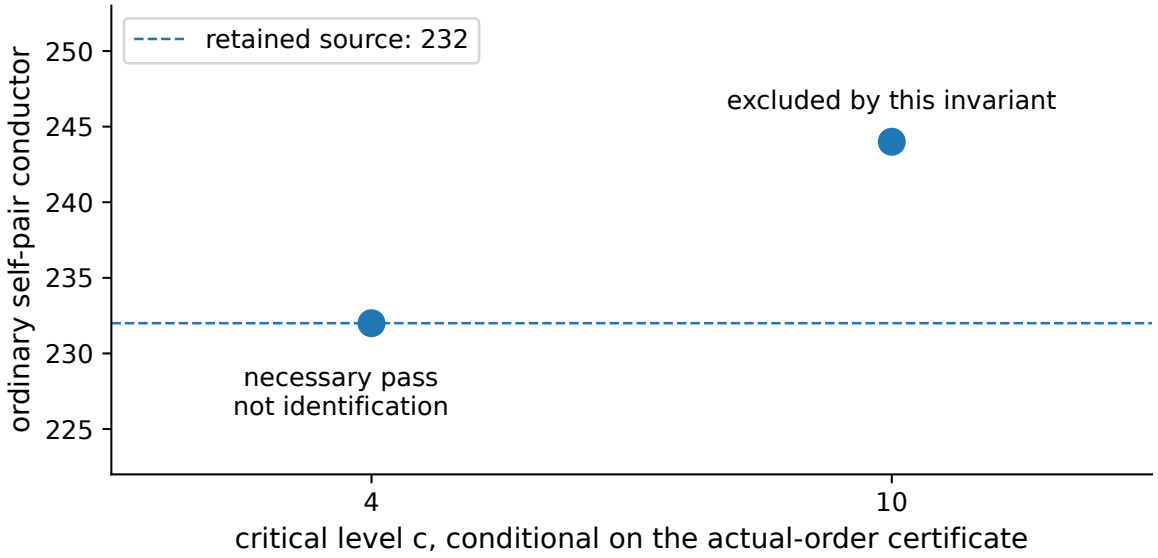
Evidence: Exact character-count model.

Companion source: Part III, Section 23.1. Book page 1533.

PLATE XI.14

A conductor filter is not an identification

The critical-four candidate passes one invariant. It is not thereby identified with the source.



The two points use the recorded period-nine actual-order certificate. It gives leading level 24, an even critical level c, index exponent $I = c + 64$, and quotient dimension $D = 8$. With the stipulated measures and torsion number one, the self-pair conductor is

$$a(\pi^{\vee} \times \pi) = 2c + 224.$$

At $c = 10$ this is 244, contradicting the unchanged source value 232. At $c = 4$ it is 232, so this necessary comparison no longer excludes the candidate. The figure shows only these two conditional evaluations, not an asserted family of actual constructions for all intermediate values of c.

The order equations are substantive inputs. The pair $(24, c)$ does not imply them. Passing the horizontal line neither constructs the corrected stratum nor establishes occurrence of the complete prescribed character. Full pair-factor equality for the same candidate remains a stronger obligation than this conductor comparison.

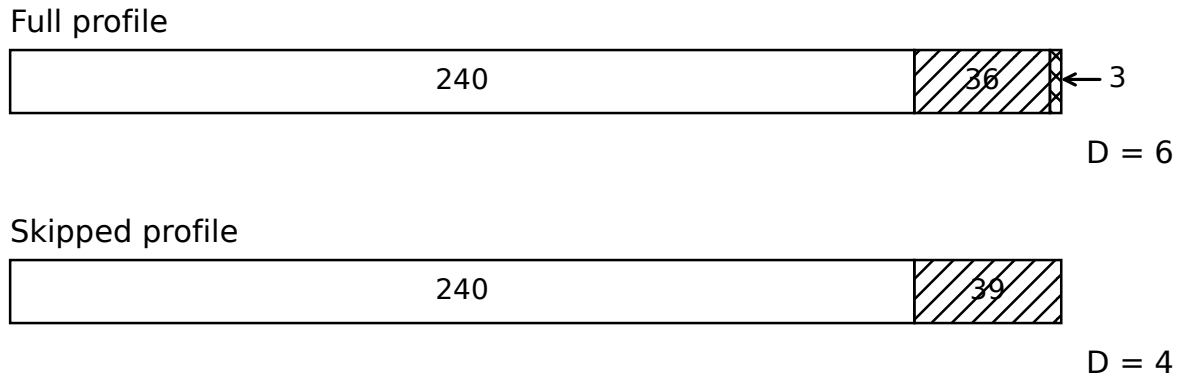
Evidence: Conditional profile comparison.

Companion source: Part III, Section 23.2–23.3. Book page 1534.

PLATE XI.15

Equal weighted sums can conceal different capacity

The conductor formula has already discarded the parity information needed for the quotient.



$$240 + 36 + 3 = 240 + 39 = 279$$

$$2 \cdot 133 + 6 = 2 \cdot 134 + 4 = 272$$

The full rank-eight profile has centralizer dimensions $(8, 4, 2, 1)$ and levels $(60, 18, 3)$. The skipped profile has dimensions $(8, 4, 1)$ and levels $(60, 13)$. The bar segments are the products $\Delta_j \nu_j$. Both sum to 279, giving ordinary self-pair conductor $63 + 279 = 342$.

$$(I, D)_{\text{full}} = (133, 6), \qquad (I, D)_{\text{skipped}} = (134, 4).$$

The compensation $2 \cdot 133 + 6 = 2 \cdot 134 + 4$ explains the equality. The identity $2\lceil \nu/2 \rceil + \mathbf{1}_{2 \mid \nu} = \nu + 1$ removes the parity distribution from the weighted conductor sum, while D retains it.

Under their inherited arithmetic construction hypotheses, both profiles also have standard conductor 68 and torsion number one. Yet only the full profile has room for the six-dimensional source quotient in this specified host. The recorded tests verify the grade consequences. They do not construct the skipped-profile parameter or turn three agreeing integers into H5.

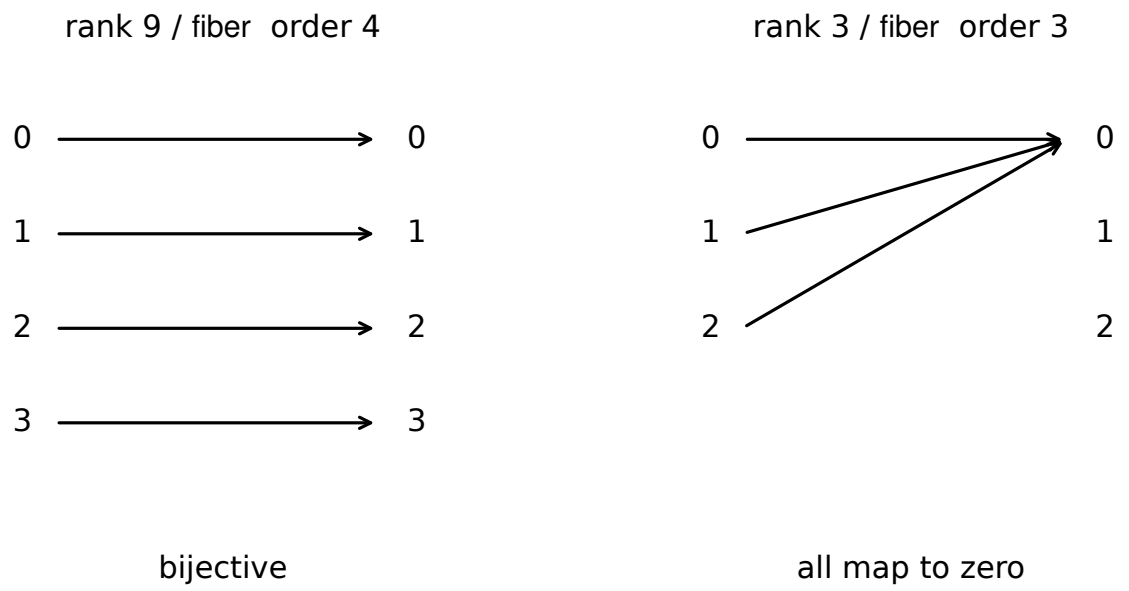
Evidence: Exact information-loss example.

Companion source: Part III, Section 24. Book page 1536.

PLATE XI.16

Central normalization of a descent fiber

A power map can be bijective in one fiber and collapse an entire family in another.



A descent fiber that is a torsor for a cyclic character group X of order m is normalized by the central-character power map $\alpha \mapsto \alpha^n$. Additive exponents identify it with $x \mapsto nx \pmod{m}$.

On the left, $n = 9$ and $m = 4$. The map is the identity modulo four, so every attainable central-character ratio selects one twist. On the right, $n = 3$ and $m = 3$. Every exponent maps to zero, so the fixed-center condition leaves all three twists available.

$$\gcd(n, m) = 1 \quad \Longrightarrow \quad \text{unique member for each attainable central ratio.}$$

The two examples correspond to different source branches. The rank-nine positive-characteristic descent and the rank-three fixed-center ambiguity are not one candidate under two names. Uniqueness in the coprime fiber does not prove a native character occurs in that candidate. Ambiguity in the cubic fiber does not disappear because a scalar obstruction has been removed.

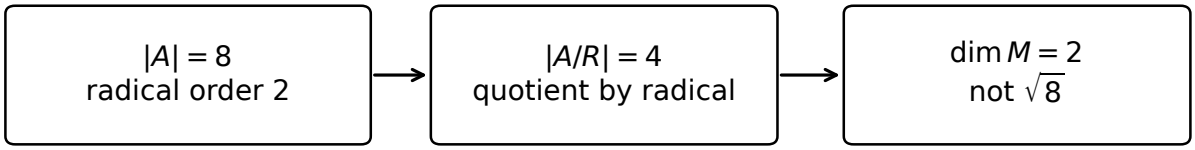
Evidence: Exact finite character-group model.

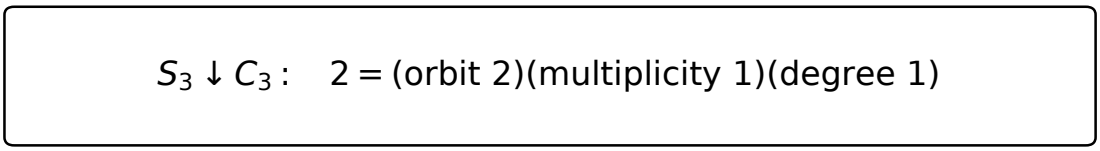
Companion source: Part II, Sections 17.3 and 18.4; Part III, Section 25.3. Book page 1523.

PLATE XI.17

Clifford dimensions keep the radical and orbit factors

An abstract packet count must match an actual restriction problem.





A second copy of one noncommuting Weyl plane is not an independent factor.

For a finite abelian projective quotient A, let R be the radical of its commutator pairing. An irreducible projective multiplicity has degree

$$(\dim M)^2 = |A/R|.$$

The first example uses $A = \mathbf{F}_2^3$, with the standard symplectic pairing on the first two coordinates and the third coordinate radical. Thus $|A| = 8$, $|R| = 2$, and $\dim M = 2$. Suppressing the radical would incorrectly produce $\sqrt{8}$.

The second example retains the inertia orbit. The two-dimensional irreducible of S_3 restricts to two distinct nontrivial characters of C_3. Each multiplicity is one, but the orbit has size two. Along a genuine compatible chain, every orbit factor and every projective multiplicity must be retained.

Independent commuting projective factors on the same carrier multiply their degrees. Representing one Weyl plane twice on the same space does not create independent commuting factors. The finite matrix checks include the S_3 relations, while the general restriction argument remains in the paper.

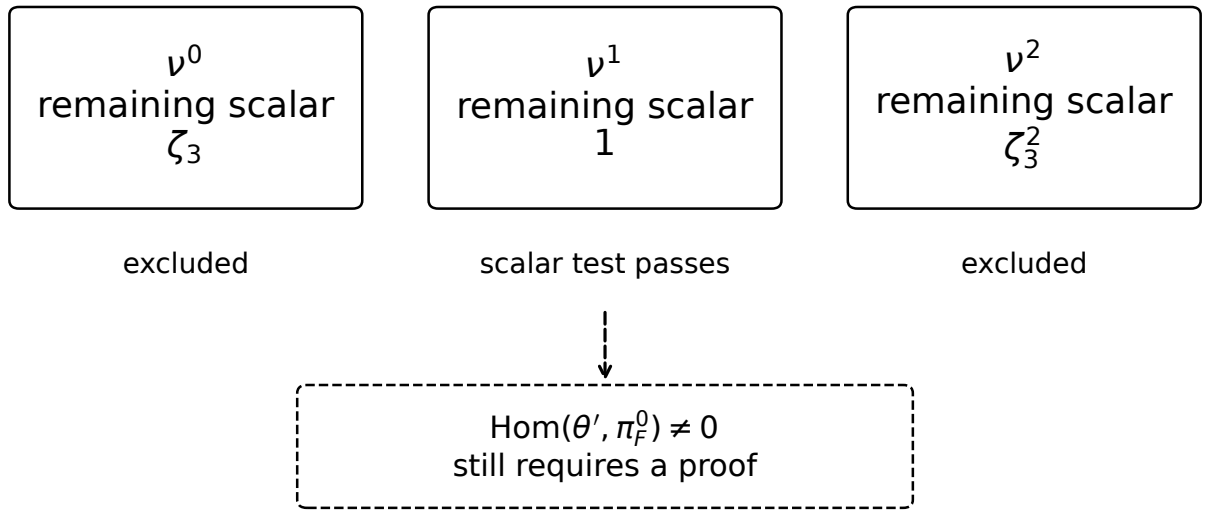
Evidence: Exact projective and orbit models.

Companion source: Part II, Section 15. Book page 1518.

PLATE XI.18

The noncompact scalar obstruction survives compact agreement

Changing the full character removes one obstruction, not the occurrence problem.



In the descent study, the original θ_0 is compatible with the tested compact action but produces the wrong scalar at a noncompact generator. The retained intertwining and maximal-type arguments yield $\mathrm{Hom}_{H^1}(\theta_0, \pi_F^0) = 0$ for the prescribed source-defined candidate.

The correction powers ν^a, for $a = 0, 1, 2$, leave generator scalars $\zeta_3, 1, \zeta_3^2$. Only the middle correction passes this test. It changes the full character, rather than merely changing a basis. Its uniformizer value must also preserve the central normalization shown on Plate XI.6.

$$\mathrm{Hom}_{H^1}(\theta', \pi_F^0) \neq 0 \quad \text{is still not established.}$$

The atlas records the exact scalar exponents. It does not rerun the earlier local-field matrix programs or supply a new occurrence proof. Choosing some other representation that contains θ' would abandon the fixed-candidate question. Rescaling an intertwiner cannot remove a nontrivial scalar between two representations whose common compact restriction is irreducible.

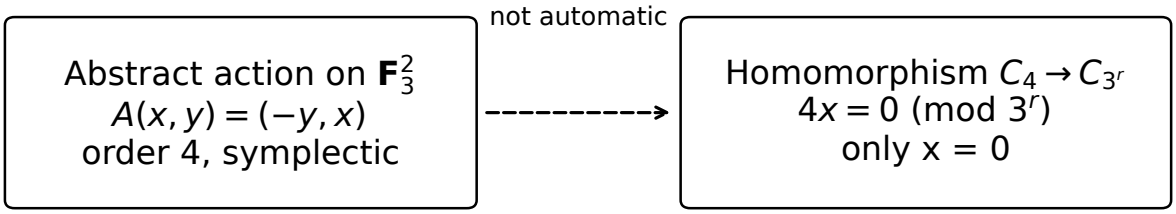
Evidence: Inherited arithmetic obstruction with exact scalar model.

Companion source: Part II, Section 18. Book page 1524.

PLATE XI.19

Abstract actions and native normalizers

Abstract lifting and arithmetic realization are different existence statements.



A nontrivial finite action may fail in a specified native normalizer.

This excludes that action mechanism, not every allowed arithmetic host.

The linear map $A(x,y) = (-y,x)$ on $\mathbf{F}_3^2$ is symplectic and has order four. Thus there is a genuine nontrivial abstract C_4 action. In contrast, every homomorphism $C_4 \to C_{3^r}$ is trivial: an image x satisfies $4x = 0$, and multiplication by four is invertible modulo 3^r.

The arithmetic obstruction uses its stated native-normalizer quotient and inner-action requirements. When that permitted quotient is a p-group, a prime-to-p subgroup cannot implement the desired nontrivial action through it. The cyclic groups in the drawing isolate the elementary coprimality mechanism. They are not asserted to be the entire normalizer of every type.

Finite central-extension actions can still lift under their own coprimality hypotheses, and projective intertwiners can sometimes be linearized. Neither result supplies the missing arithmetic conjugating matrices. An obstruction to this normalizer mechanism therefore remains compatible with success through a different allowed host or a different stated attachment contract.

Evidence: Exact coprimality analogue.

Companion source: Part I, Sections 4 and 11. Book page 1489.

PLATE XI.20

The finite subgroup remembers its squares

Dihedral and quaternion markings survive inside the same complex scalar extension.

$$[X, Z] = [iX, iZ] = -I$$

finite subgroup	q(e)	q(f)	q(e+f)	order 2	order 4
D8	0	0	1	5	2
Q8	1	1	1	1	6

Same scalar pushout, different marked finite subgroups.

Let $X^2 = Z^2 = I$ and $XZ = -ZX$. The subgroup generated by X, Z is dihedral of order eight. The subgroup generated by iX, iZ is quaternion of order eight. Both have the same quotient polarization and commutator $-I$.

$$q_D(x, y) = xy, \qquad q_Q(x, y) = xy + x + y.$$

Their lift squares differ. The order columns count elements of exact order two and four, excluding the identity. They distinguish the finite groups even though adjoining all complex scalars gives the same ambient scalar extension.

For a fixed two-element center, a center-fixing extension map over a prescribed symplectic map must preserve the quadratic refinement. With complex scalar center, square ratios have square roots, but this freedom does not erase a separately prescribed finite subgroup. Transporting a quaternion marking into the common scalar extension is valid. Identifying it with the independently fixed dihedral marking is not.

Evidence: Exact finite-group enumeration.

Companion source: Part III, Section 26.1–26.2. Book page 1541.

PLATE XI.21

A square table does not specify an entire section

Changing one section value can preserve every square and still change products.



Start from the ordered section $\sigma(x,y) = X^x Z^y$ in the dihedral example. Change only $\sigma(e+f)$ by a central sign. Write the sign function as $a(v) = \mathbf{1}_{v=e+f}$. The displayed entries are the resulting cocycle differences

$$\Delta c(v,w) = a(v) + a(w) + a(v+w) \in \mathbf{F}_2.$$

Every diagonal entry is zero, so every square is unchanged. But $\Delta c(e,f) = 1$. A map fixing the specified generating lifts cannot also implement the changed assignment at $e+f$. The cross marks a nonzero product discrepancy, not failure of the two extensions to be isomorphic in some other way.

This is why the native certificate needs all specified section products, not only a quadratic table. When the entire section is transported through one homomorphism, those products are retained automatically. When the target section is independently prescribed, its pointwise assignments must be checked. The distinction concerns what is fixed, not a contradiction in extension theory.

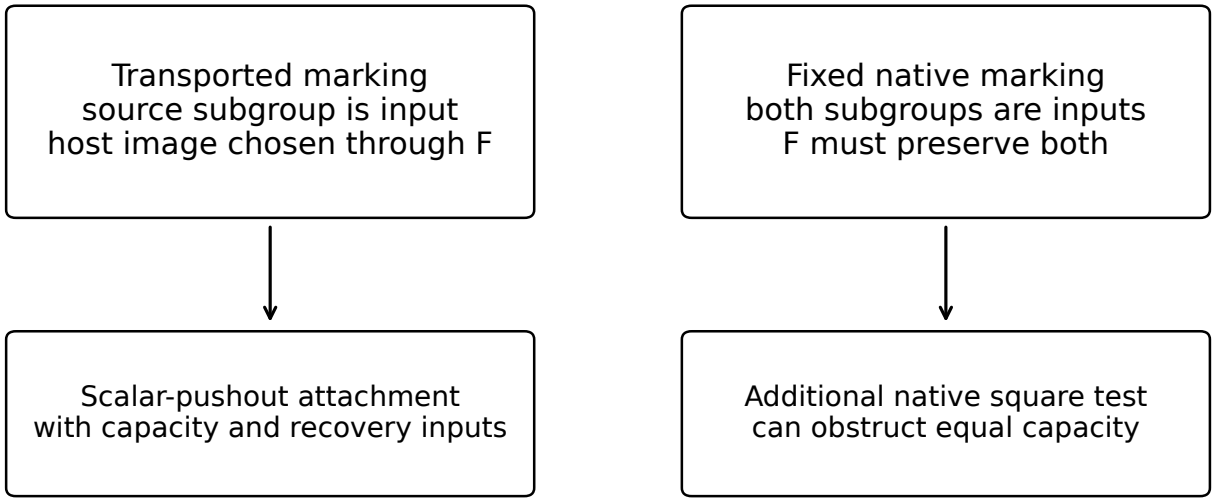
Evidence: Exact cocycle counterexample.

Companion source: Part III, Section 26.3. Book page 1542.

PLATE XI.22

Two different marking contracts

A positive attachment and a native obstruction can both be correct.



In the transported-marking contract, the source marking is supplied. After a permitted symplectic embedding and scalar-extension isomorphism have been chosen, the host marking is their image. A homomorphism fixes the represented center and carries squares, products, and specified lifts together.

In the independently fixed native contract, the target finite subgroup and its refinement are already given. The comparison must hit that subgroup with its existing marking. A scalar rescaling that changes it may be disallowed.

$$\text{choose a compatible target marking} \quad \neq \quad \text{match this prescribed target marking.}$$

The restricted one-vertex, trivial-stabilizer attachment retains its actual-type, capacity, full-character, and recovery hypotheses. It should not acquire an extra native-marking premise merely because a stronger problem is also studied. Conversely, that positive theorem cannot certify the stronger problem without computing the additional datum. The two branches in the figure remain separate throughout the atlas.

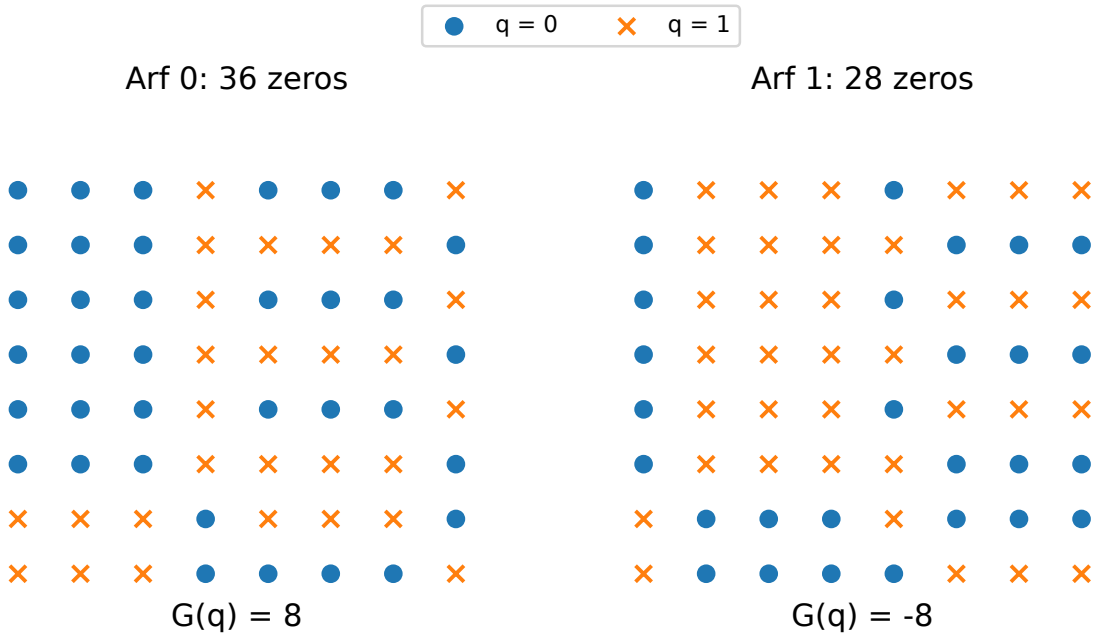
Evidence: Contract comparison.

Companion source: Part III, Sections 26.4 and 27.3. Book page 1542.

PLATE XI.23

Two six-dimensional refinements with different signs

The alternating form and dimension do not determine the Arf invariant.



Each grid contains all 64 binary vectors in dimension six. Columns encode the lower three bits and rows the upper three bits, with row zero at the top. Circles have quadratic value zero. Crosses have value one. The two forms share the standard polarization:

$$q_+ = x_1y_1 + x_2y_2 + x_3y_3, \qquad q_- = q_+ + x_1 + y_1.$$

Their Gauss sums are $+8$ and -8. Consequently the zero counts, including the zero vector, are $(64 + 8)/2 = 36$ and $(64 - 8)/2 = 28$. The diagrams use exact binary evaluation at every vector.

These are explicit abstract refinements, not a claim that either has been computed on the actual rank-eight arithmetic candidate. The example shows the datum still required when the host marking is independently fixed. At equal capacity, an Arf mismatch has no orthogonal complement in which to be absorbed.

Evidence: Exhaustive 64-vector displays.

Companion source: Part III, Section 27.1. Book page 1543.

PLATE XI.24

The equal-capacity test is sharp

One spare quadratic plane removes the free Arf obstruction, not every attachment constraint.

Source dimension is 6. The embedding may be freely chosen.

host dim	host Arf	source Arf 0	source Arf 1
4	0	no	no
4	1	no	no
6	0	yes	no
6	1	no	yes
8	0	yes	yes
8	1	yes	yes

Fix a nondegenerate source of dimension six. The table varies the host dimension and its supplied Arf invariant. The embedding may be freely chosen. A four-dimensional host cannot contain the source. At dimension six, the map is an isometry and the Arf signs must agree. At dimension eight, the complementary plane can have either sign.

$$\mathrm{Arf}(U) = \mathrm{Arf}(W) + \mathrm{Arf}(V) \quad \text{for } W \simeq V \perp U.$$

This is the finite classification argument, not a numerical search over arithmetic matrices. It combines with the original full dyadic capacity formula to isolate the rank-eight equality case.

The word “free” matters. A prescribed depth placement, an already fixed quotient embedding, or a stabilizer-equivariance condition may forbid the required change of subspace. The table assumes the actual quadratic host has been supplied. It does not compute that host’s native marking, choose its lifts, or authorize modifications to the transport diagram.

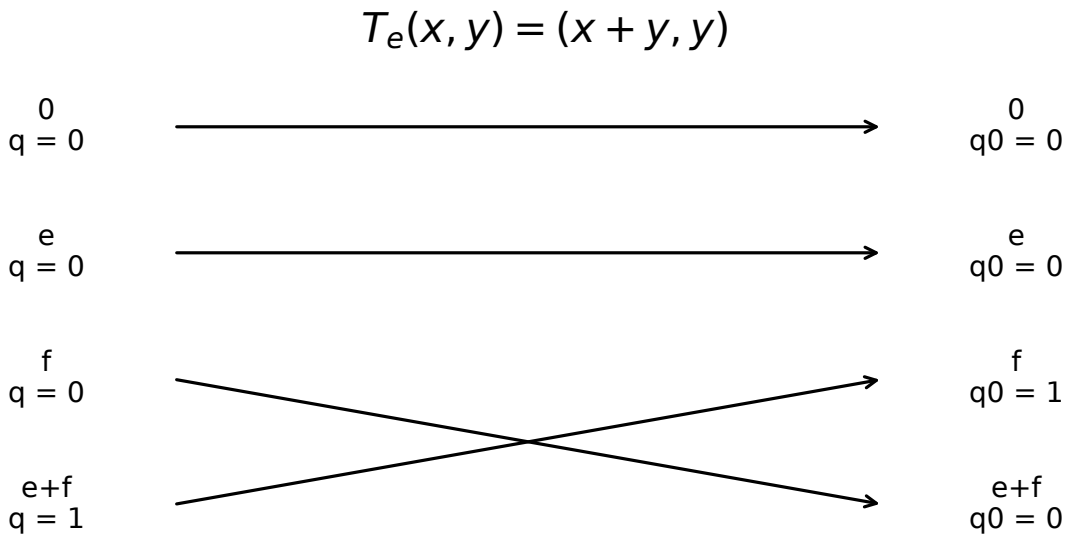
Evidence: Finite classification criterion.

Companion source: Part III, Section 27.2–27.3. Book page 1543.

PLATE XI.25

A single transvection repairs an isotropic defect

The repair is explicit, and the previously correct line stays fixed.



0 and e stay fixed. f and e+f are exchanged.

Take $q(x,y) = xy$ and $q_0(x,y) = xy + y$. Their discrepancy is $B(e,-)$, with $e = (1,0)$ and $q(e) = 0$. The transvection is

$$T_e(x,y) = (x+y,y), \qquad q_0(T_e v) = q(v).$$

The arrows show its action on every vector. Zero and e remain fixed. The vectors f and $e+f$ exchange places. Each arrow therefore lands on the same quadratic value shown at its source.

The original identity map did not work: $q(f) = 0$ while $q_0(f) = 1$. Matching Arf invariants guarantees a correction here, not the validity of an unchanged map. More generally, a defect vanishing on a subspace U gives a defect vector in $U^{\perp}$, so the transvection fixes U pointwise.

The correction fixes quotient vectors, not automatically independently fixed lifts above them. Its use in an arithmetic diagram requires permission to change the embedding and a separate check of the relevant section and phase data.

Evidence: Exact four-vector correction.

Companion source: Part III, Section 28.1–28.3. Book page 1545.

PLATE XI.26

The defect vector supplies a decision procedure

The same formula displays both the successful correction and its genuine obstruction.

$$q_0 = q + B(a, -)$$

defect a	q(a)	mismatched vectors	T_a correction
0	0	0	works
e	0	0	works
f	0	0	works
e+f	1	2	fails

When q(a) = 1, Arf changes and the obstruction is genuine.

For the fixed hyperbolic plane $q = xy$, every refinement with the same polar form is $q_0 = q + B(a, -)$ for one vector a. The table lists all four defect vectors. Its mismatch column counts the vectors at which $q_0(T_a v) \neq q(v)$.

$$q_0(T_a v) = q(v) + B(a, v)q(a), \qquad \mathrm{Arf}(q_0) + \mathrm{Arf}(q) = q(a).$$

When $q(a) = 0$, the correction works everywhere. For $a = e + f$, the Arf sign changes and two vectors still fail. This is not an unlucky choice of basis. In equal dimension, the two forms are not isometric.

In a larger host this test applies to the pullback along the supplied embedding. A different permitted subspace may behave differently. Under genuine supplied actions preserving both refinements and an initially equivariant embedding, the defect vector is fixed by the group. The same transvection is then equivariant. Those initial actions are hypotheses, not outputs of this linear solve.

Evidence: Exact defect enumeration.

Companion source: Part III, Section 28. Book page 1545.

PLATE XI.27

An odd-order action discrepancy can be summed away

The correction acts on a supplied finite extension, not on an unconstructed arithmetic host.

$$u = \sum_{h} \delta_h = e$$

group element	delta_g	g u	u + delta_g + g u
1	0	e	0
g	e+f	f	0
g²	f	e+f	0

$$g(x, y) = (y, x + y) \quad \text{on } \mathbf{F}_2^2$$

Let C_3 act on $A = \mathbf{F}_2^2$ by $g(x, y) = (y, x + y)$. Use the cocycle values $\delta_1 = 0$, $\delta_g = e + f$, and $\delta_{g^2} = f$. Their sum is $u = e$. The final column displays the corrected discrepancy, which is zero at every group element.

$$\delta_{gh} = \delta_g + g\delta_h, \qquad u = \sum_h \delta_h, \qquad \delta_g = u + gu.$$

The last identity follows by summing the cocycle equation and using that the group order is odd in characteristic two. Replacing the comparison by the vertical sign correction u makes it equivariant.

In the extension theorem, the source and target already have genuine actions and an initial center-fixing extension isomorphism. The cocycle lives in the character space $\mathrm{Hom}(W, \mathbf{F}_2)$. The diagram is an exact module example of that calculation, not a new construction of those actions. A vertical correction may also be forbidden when particular lifts must remain unchanged.

Evidence: Exact C3-module model.

Companion source: Part III, Section 29.1–29.2. Book page 1547.

PLATE XI.28

The even-order negative control

A nonzero cocycle survives when every permitted coboundary vanishes.

C2 acts trivially. The cocycle has delta_s = e.

candidate u	u + s u	remaining delta_s
0	0	e
e	0	e
f	0	e
e+f	0	e

Every permitted coboundary is zero. None removes this cocycle.

Let $C_2 = \{1, s\}$ act trivially on the nonzero module $A = \mathbf{F}_2^2$. The assignment $\delta_1 = 0$, $\delta_s = e$ satisfies the cocycle equation, since $e + e = 0$. For every proposed correction u,

$$u + su = u + u = 0, \qquad u + \delta_s + su = e.$$

The four rows exhaust the module. None removes the discrepancy. Thus the odd-order hypothesis of the preceding plate is mathematically active.

This does not say that every even-order action problem is obstructed. It supplies one exact counterexample to an unrestricted averaging claim. Other actions and cocycles may admit corrections. In the arithmetic problem, it remains necessary to compute the actual discrepancy and respect the permitted transformations. Declaring every projective or equivariance mismatch removable would erase precisely the information this example retains.

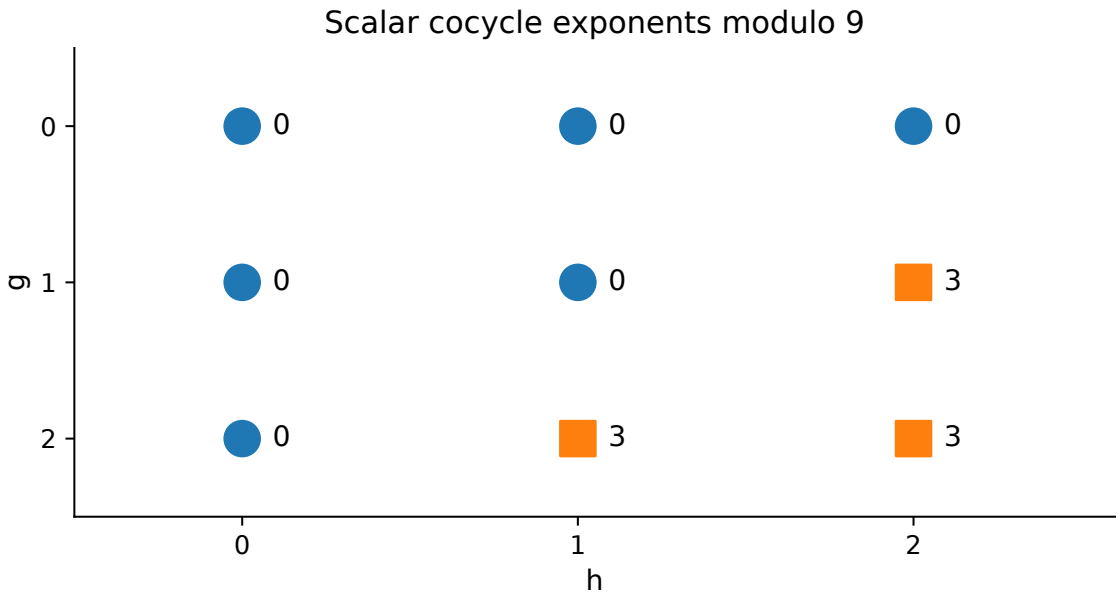
Evidence: Exact obstruction model.

Companion source: Part III, Section 29.2. Book page 1548.

PLATE XI.29

Honest phases remain a separate layer

An action on the extension group is not yet the specified action on its representation.



Let $g, h \in \{0, 1, 2\}$ represent C_3, and take the one-dimensional operators $U_g = \zeta_9^g$. Their multiplication has scalar discrepancy

$$U_g U_h = \zeta_9^{c(g,h)} U_{g+h \bmod 3}, \qquad c(g,h) = 3 \left\lfloor \frac{g+h}{3} \right\rfloor .$$

The figure records its exponents modulo nine. Replacing U_g by $\zeta_9^{-g} U_g$ gives an honest action. Yet the three honest characters $g \mapsto \zeta_3^{kg}$ are still distinct choices.

This scalar toy model isolates two separate questions: whether a projective multiplier can be removed, and which honest linearization the contract prescribes. It is not the phase system of the actual arithmetic candidate. The finite extension-action correction on Plate XI.27 addresses a different discrepancy, in the group automorphisms. Canonical extension phases, Gauss conventions, and coherent transports must still be checked on the same full representation.

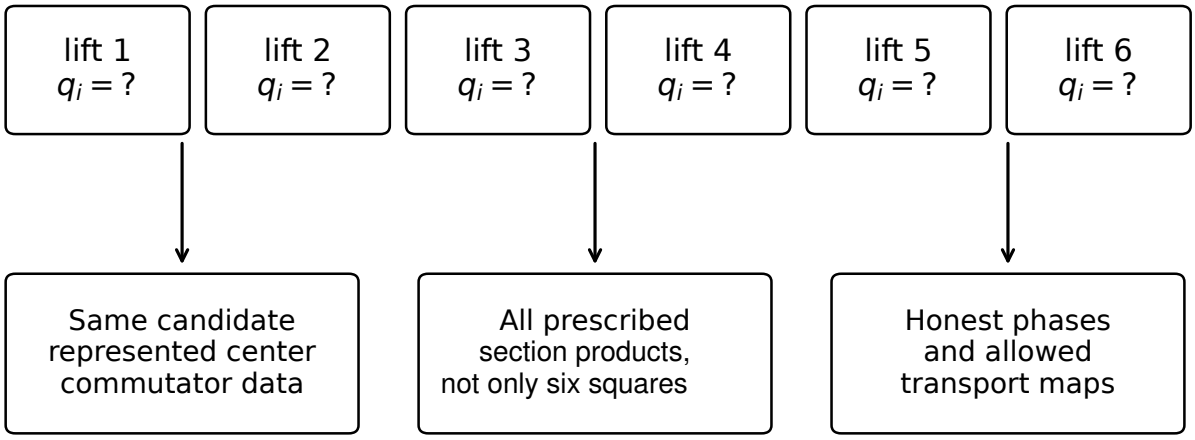
Evidence: Exact scalar projective toy model.

Companion source: Part I, Section 4.2; Part III, Section 29.3. Book page 1489.

PLATE XI.30

The missing native certificate has explicit fields

Question marks are not zeros and must not be filled with a convenient abstract form.



The six question marks are deliberate: no native bit is fabricated.

When the stronger contract independently fixes a native rank-eight marking, the certificate must name the actual candidate, the relevant finite subgroup and represented center, and the six genuine lifts of a symplectic basis. Their squares determine six bits only after those lifts and the two-element central representation have been established.

$$q_W(v_i) \in \mathbf{F}_2 \quad \text{must come from the actual lift square.}$$

All prescribed section products must also be verified. Genuine extension actions do not by themselves choose the required honest representation phases. The certificate must state which maps or lifts may change and verify the resulting transport conditions.

No native square bit is supplied by this atlas. The data file records these fields as null, and the tests ensure they remain uncomputed rather than silently assigning zero. This is a conditional stronger obligation. It is not an extra premise inserted into the already valid transported-marking contract, where the finite marking is carried through the chosen comparison map.

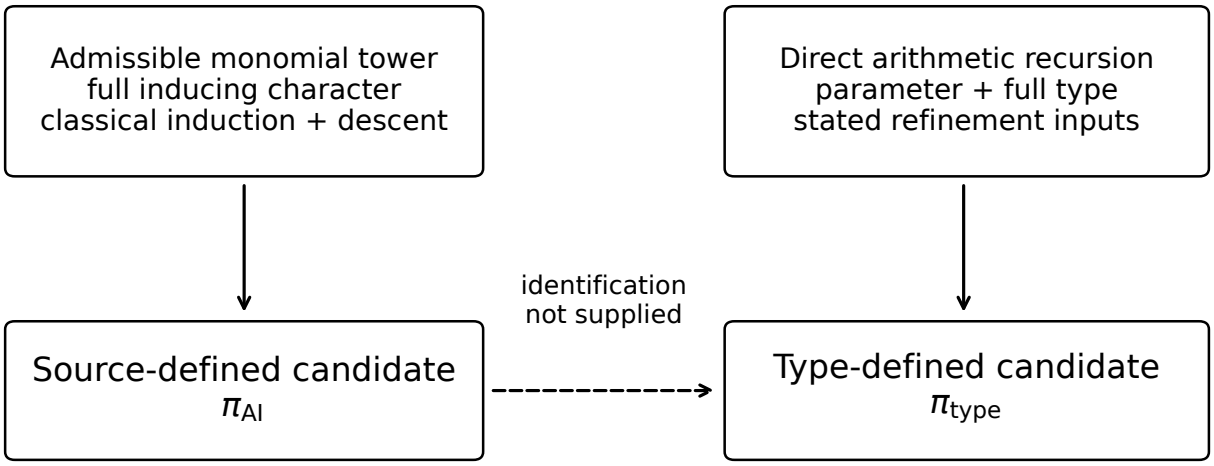
Evidence: Unresolved arithmetic certificate.

Companion source: Part III, Section 30.2. Book page 1549.

PLATE XI.31

Keep the two construction routes distinct

A source-defined candidate and a type-defined candidate require an identification theorem.



The monomial route begins with a full inducing character and an admissible tower. Under the stated classical induction and descent inputs it produces a source-defined candidate. Solvability of an ambient finite group does not make every proposed subgroup admissible for a prime-cyclic tower.

The direct route begins with an explicit parameter and type construction. Its refinement and full-character arguments produce an actual candidate under a restricted contract. Even when both routes match rank, standard conductor, torsion number, and self-pair conductor, those invariants do not identify their outputs.

$$\pi_{\mathrm{AI}} \simeq \pi_{\mathrm{type}} \quad \text{needs its own argument on the same source and field.}$$

The positive-characteristic rank-nine source-defined route and the characteristic-zero example in the descent study remain separate. The complete Henniart–Lemaire text was unavailable during the recorded conductor audit, so its detailed applications retain inherited audit status. The visual argument supplies no new classical-source certification.

Evidence: Source-lineage diagram.

Companion source: Part II, Sections 14 and 17; Part III, Section 25. Book page 1516.

PLATE XI.32

The current frontier

Completed finite and restricted work remains reusable without relabeling the open endpoints.

One proof ledger, not a percentage of a conjecture.

reusable result or endpoint	current boundary
restricted constructions	written arguments retained
finite corrections	conditional proofs, exact models
general realization	OPEN
same-candidate H5	OPEN
corrected-character occurrence	OPEN
fixed native marking	OPEN when independently required

The progress record has a distinct visual reading route through the restricted constructors, finite comparison mechanisms, exact obstructions, and conditional corrections. These are substantive results and reusable arguments. Their existence does not make the remaining arithmetic statements interchangeable.

General realization requires an actual common host with the admitted actions and transports. Same-candidate H5 requires the characterizing factors for that very realizer. The corrected-character problem requires occurrence in the original specified induction candidate. An independently fixed native marking, when required, needs the actual certificate on Plate XI.30.

The atlas accompanies the arithmetic research companion and its evidence record. Finite illustrations, ordinary proofs, classical inputs, and historical formal receipts retain their separate scopes. Neither the atlas nor its assembly into this book supplies a fresh Lean compilation or external peer review, and the recorded checks do not measure a percentage of conjecture completion.

Evidence: Claim and source map.

Companion source: Part I, Section 12; Part II, Section 20; Part III, Section 30. Book page 1509.

Plate concordance

The page numbers below refer to the arithmetic studies in this book. Each plate also has a direct link to its source argument.

Plate	Argument	Book page
XI.1	The three obligations	1486
XI.2	One candidate must satisfy both conditions	1486
XI.3	A balanced central product	1488
XI.4	What a compact binder can do	1490
XI.5	The complete cubic corner spectrum	1527
XI.6	Do not truncate the full character	1495
XI.7	Depth can make native trace one invisible	1498
XI.8	Two defining-sequence examples	1492
XI.9	Where the original dyadic quotient runs out of room	1501
XI.10	A larger inducing carrier can return the same object	1502
XI.11	Three conductor columns are not interchangeable	1532
XI.12	The principal-period filter	1532
XI.13	The rank-nine source has a visible break count	1533
XI.14	A conductor filter is not an identification	1534
XI.15	Equal weighted sums can conceal different capacity	1536
XI.16	Central normalization of a descent fiber	1523

Scope of the source routes

The concordance links each plate to its argument in the arithmetic studies. Hypotheses, arithmetic inputs and verification scope remain those of the cited statements.

Concordance and replay boundary

Plate	Argument	Book page
XI.17	Clifford dimensions keep the radical and orbit factors	1518
XI.18	The noncompact scalar obstruction survives compact agreement	1524
XI.19	Abstract actions and native normalizers	1489
XI.20	The finite subgroup remembers its squares	1541
XI.21	A square table does not specify an entire section	1542
XI.22	Two different marking contracts	1542
XI.23	Two six-dimensional refinements with different signs	1543
XI.24	The equal-capacity test is sharp	1543
XI.25	A single transvection repairs an isotropic defect	1545
XI.26	The defect vector supplies a decision procedure	1545
XI.27	An odd-order action discrepancy can be summed away	1547
XI.28	The even-order negative control	1548
XI.29	Honest phases remain a separate layer	1489
XI.30	The missing native certificate has explicit fields	1549
XI.31	Keep the two construction routes distinct	1516
XI.32	The current frontier	1509

Executable checks and scope

The atlas suite contains 28 test methods. It checks the displayed finite data and deliberate false controls, including the 72-target corner, conductor translations, cocycle products, all 64 vectors in each display of dimension six, and explicit correction formulas. These finite tests do not measure theorem completion. The native certificate remains uncomputed.

Formal companions

The formal companions connect the mathematical chapters to their Lean declarations, hypotheses, source correspondences and verification evidence. Read each statement with the coverage described by its companion.

READING ROUTES

I Projector Resolved Seed Calculus I 1588

II Projector Resolved Seed Calculus II 1721

III Projector Resolved Seed Calculus III 1768

IV Projector Resolved Seed Calculus IV 1815

V Multiresolution Signal Calculus 1936

VI Transfer Semigroups and Memory 1992

VII Quantum Seeds and Effective Dynamics 2054

VIII Packet Operator Algorithms 2132

IX Euler Congruences 2220

X Universal Residual Seeds 2350

XI Tame and Wild Packet Geometry 2384

XII Patching Support and Hecke Packets 2617

Projector Resolved Seed Calculus I

The formal companion to Chapter I explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Computational formulation 1589

Formal companion 1683

Return to Chapter I

Computational records and interfaces

This computational companion represents Chapter I as a collection of mathematical records that can be inspected and queried, with a finite subset supported by executable interfaces. The statement inventory is formal/coverage/COVERAGE_108.json. The executable runtime has a separate source entry in the source register. The catalogue and runtime have the distinct scopes explained below.

The catalogue contains 108 mathematical records, including 69 theorem, proposition, corollary, and lemma records with source proofs. The formal correspondence is reported alongside each record. The preserved runtime interface has twenty typed signatures, and its separate Lean runtime map has eight mappings. Those inherited interfaces and receipts retain their original scope.

The mathematical source supplies the statements and proofs, including the split-semisimple hypotheses and explicit map qualifications. The supplementary worked models remain in the mathematical text. Stable identifiers connect the mathematical, formal, and computational material. Statement hashes and source locators distinguish the catalogue from the archived runtime payloads.

The verification record distinguishes evidence for the mathematical correspondence from the runtime receipts. Its source inventory and formal correspondence map are available under `formal/coverage/`, and compiler and replay receipts are under `formal/evidence/`. The Lean guide explains the formal sources, their hypotheses, and how to interpret those checks.

How to use this volume. Read Sections 1–3 for mathematical and evidence semantics. Sections 4–6 give the executable contracts and an exact worked example. Section 7 describes reproduction. Appendix A reproduces every canonical source statement and every attached proof body, with source locators and machine metadata. Later appendices give signatures, action coverage, operation vocabulary, and the runtime Lean map.

1 The mathematical object behind the interface

1.1 Operator systems, generated hulls, and seed rank

A seed problem consists of an acting algebra, a module, and a family of seed vectors. The algebra must be fixed before the generation question is asked. For a unital algebra A acting on a module M and a finite seed family $S = \{s_1, \dots, s_r\}$, the algebraic generated hull is

$$\mathrm{Hull}_A(S) = \sum_{j=1}^{r} As_j.$$

A one-seed system is cyclic. Seed rank measures the minimum cardinality of a generating family, under the regime declared in the source. Enlarging the acting algebra changes the question. Replacing algebraic span by a closure changes it again. The machine record therefore keeps the acting regime and hypotheses beside the statement.

Chapter I begins with typed operator grammars and their realizations, then passes through free presentations, generator number, packet coordinates, block algebras, transfer graphs, and realization. The resulting record graph covers the whole expanded foundation. It is not limited to the finite matrix examples that the runtime executes.

For finite families of nonnegative integers, the ambient convention is $\max \varnothing = 0$. The height of an empty finite poset is zero. A nonempty poset with no strict comparable pair also has height zero, so height counts strict steps. These conventions clarify the empty cases without adding stable records.

The source separates algebraic, linear, and topological claims. A finite-dimensional matrix computation can exercise a linear identity. It cannot establish the Banach space result in the countable setting or a generator formula over every division ring. Those claims retain their own source proofs and dependencies.

1.2 Exact packet coordinates

For a finite index set I, a packet system on X consists of linear maps P_α satisfying

$$P_\alpha^2 = P_\alpha, \qquad P_\beta P_\alpha = 0 \quad (\beta \neq \alpha), \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

The packet spaces are the ranges $X_\alpha = P_\alpha X$. Analysis and synthesis are

$$\mathcal{A}_P x = (P_\alpha x)_{\alpha \in I}, \qquad \mathcal{S}_P((x_\alpha)_\alpha) = \sum_\alpha x_\alpha.$$

Their inverse laws are exact. No assumption of inner product orthogonality is needed. This distinction is visible in the oblique two-dimensional example later in this companion.

Machine execution represents each packet as an ambient vector. It does not construct a basis of every range or replace the ambient representation by smaller coordinate arrays. This choice makes exact identities easy to compare with the source formulas while retaining the packet index structure.

1.3 Transfer blocks and composition

For an endomorphism T, the transfer block from packet α to packet β is

$$T_{\beta\alpha} = P_\beta T P_\alpha.$$

Its source is X_α and its target is X_β. In the finite runtime it is stored as a full ambient matrix. Completeness gives

$$T = \sum_{\alpha,\beta} T_{\beta\alpha}, \qquad (ST)_{\gamma\alpha} = \sum_{\beta} S_{\gamma\beta} T_{\beta\alpha}.$$

The order matters: the rightmost factor acts first. The runtime uses the same convention. Its nested block array is indexed first by target and then by source.

The graph records source dependencies for these identities. It does not infer every implicit mathematical dependency. The packet-system hypotheses remain required even when a compact source statement inherits them from the surrounding section.

1.4 Intertwining and reconstruction

For packet systems P on X and Q on Y, an intertwiner $U : X \to Y$ satisfies $UT = SU$. Resolving both sides yields

$$\sum_{\beta} U_{\gamma\beta} T_{\beta\alpha} = \sum_{\delta} S_{\gamma\delta} U_{\delta\alpha}, \qquad U_{\gamma\alpha} = Q_\gamma U P_\alpha.$$

All local equations together are equivalent to the global equation under the packet hypotheses. The runtime returns the global Boolean, every local residual, and the conjunction of the local Booleans. It provides the data needed to inspect agreement or failure in a concrete exact case.

1.5 The boundary of finite execution

Countable packet sums require a topology and a convergence condition. Quotient descent requires invariance. Coordinate transport requires a suitable invertible map. Packet-pure seed counts differ from unrestricted seed counts because one mixed-support seed can contribute to several source components. These are mathematical hypotheses, not formatting conventions.

The source records seven explicit boundaries. Their presence is part of the interface. Runtime action names do not imply that every associated theorem has an implementation, and metadata listing a boundary does not itself execute that boundary check.

2 Canonical machine representation

2.1 Entrypoint and package identity

The mathematical records are located at `formal/coverage/COVERAGE_108.json`. Appendix A is generated directly from those records. Each includes the statement, source proof where present, source location, hashes, formal status, and declaration correspondence. The archived mathematical source identified by those hashes is the statement authority.

The source register identifies the executable runtime. Runtime commands assume its working directory. Its entrypoint, PSC_I.machine.json, points to the preserved record collection, statement AST, dependency graph, action registry, and evidence index. A

runtime result therefore retains the exact interface scope explained below. It is not a certificate for every current manuscript statement.

The preserved runtime source digest is:

```
37bf8f03dbbf66ce41f8b07a8e7be8fa5fca44176bdd56dd25b483f72736ff77
```

That digest identifies the archived runtime source, including explicit nonzero-packet assumptions for two corollaries and the empty-family and poset-height conventions. The mathematical inventory has its own statement and proof hashes, printed in Appendix A. Runtime receipts remain attached to their original bytes. They do not transfer automatically to the mathematical inventory.

2.2 Record identity and source provenance

Each current record has a stable ID, kind, title, statement text, optional proof text, source locator, hashes, formal status, declaration targets, and a scope note. These fields identify the actual current statement and the proof text that accompanies it. The preserved executable payload additionally supplies its own dependency annotations, semantic profiles, evidence coordinates, and fingerprints for its source.

There are 28 definitions, 28 theorems, 22 corollaries, 16 propositions, three lemmas, three classical inputs, seven boundaries, and one remark. The 69 theorem/proposition/corollary/lemma records have attached proof bodies. Definitions and boundary statements are not counted as proved theorems merely because they occupy records.

> A fingerprint in the preserved runtime provides a content identity under that payload's generating convention. It is not a proof certificate. The runtime verifier checks the source digest, ID count, graph, AST coverage, selected API behavior, and exact fixtures. It does not establish all mathematical content through independent formal reconstruction.

2.3 Statement AST and typed signatures

The runtime statement AST has one entry per stable ID in its source. The statuses are 20 TYPED_CORE, 81 STRUCTURED_SYMBOLIC, and seven TEXTUAL_SEMANTIC. A typed-core record points to a signature declaring parameters, requirements where supplied, and a return description.

The AST segments statements into prose and mathematical surfaces. Some expressions are recognized as relations or direct-sum expressions. Others retain `latex_expr` leaves. Even a typed-core record can contain these leaves. Consequently, "typed core" refers to the attached interface signature, not to a complete formal elaboration or a proof term.

The signature for block intertwining describes a logical equivalence between the global and local equations. Its source proof remains attached separately. The runtime can evaluate a finite instance. These are three different representations with three different evidentiary roles.

2.4 Dependencies and closure

The preserved runtime dependency graph contains 108 nodes and 97 internal edges. The runtime verifier confirms that the recorded internal dependencies resolve and that the graph is acyclic. Direct dependencies are stored on each record. The API can also return a transitive closure.

Closure queries return sorted identifiers. They do not produce a proof order, carry out the proof, or resolve external classical inputs. A consumer can use the graph to inspect a prerequisite set and then obtain source statements and evidence for each record.

2.5 Operation profiles and actions

Each preserved runtime record carries source-derived operation metadata, with a transitive profile alongside the direct profile. The ontology distinguishes structural operations from hypothesis classes. These annotations support planning and inspection. They should not be interpreted as a verified extraction of every semantic operation used by an arbitrary proof.

The action registry contains 37 action names associated with 75 records. Eight names are implemented in the runtime. Other entries are marked planner-only or fixture-only. The source has broader mathematical scope than the executable subset.

3 Evidence semantics and certification scope

3.1 Independent evidence coordinates

The preserved runtime evidence object separates source, KIR, Python, and Lean. Its source coordinate identifies that payload's source, and its KIR coordinate records validation of that source. Python evidence reports exact finite fixtures for 26 records. Its Lean coordinate points to eight runtime source mappings.

The catalogue uses the formal correspondence map instead. Each of the 108 records carries a status and scope note, and each proof-level statement has its own correspondence assessment. The verification index distinguishes the execution evidence attached to each component.

No coordinate upgrades another automatically. Exact arithmetic removes floating-point ambiguity from a finite computation. It does not quantify over arbitrary fields, modules, dimensions, or topological spaces. A mapped Lean declaration proves its own formal statement under its dependencies. Its relation to the source statement must still be read from the mapping.

3.2 Formal correspondence and checks

The correspondence map classifies the 69 proof-level records as 69 covered, 0 partial, and 0 not mapped. The classifications describe the complete statements at their recorded scopes. They do not count independent theorems or executable actions. Every statement, source proof, and correspondence scope is reproduced in Appendix A. The exact declaration lists are in `formal/coverage/COVERAGE_108.json`.

The recorded audit checks 116 formal source modules plus the retained boundary suite. Including that suite, it inventories 3513 source-owned entries and checks 3500 logical declarations. Of these, 3499 are checked by logical replay and one declaration sharing an upstream name receives a fresh-name kernel body check after exact type/value comparison. It inventories 13 runtime helpers separately and records 12 rejected negative controls. The allowed transitive axiom set is `Classical.choice`, `Quot.sound`, `propext`. The receipt is `formal/evidence/RELEASE_VERIFICATION.json`. Its detailed compiler and logical-replay evidence is under `formal/evidence/whole_closure/`.

The recorded computational checks pass for 96 exact finite or symbolic visual assertions, 33 companion tests, and 10 regenerated PDF/PNG figure pairs. These bounded computations have their own scope and do not replace the formal declarations.

The pinned upstream dependency cache is reused. Logical replay uses Lean's official kernel. It is not an independently implemented checker. The source-to-manuscript correspondence is a separate reviewed mathematical artifact.

3.3 Runtime Lean linkage

The attached seven-module formalization uses the environment specified in the accompanying source records. Its eight mappings include definitions, generalized theorems, one-directional correspondence, and a related sufficient direction. The support-pattern algebra mapping is explicitly partial. It must not be counted as a full formal certificate for every direction of the source criterion.

A separate predecessor report records 35 of 35 obligations for its identified source. That lineage is not the canonical correspondence for the expanded mathematical inventory. The predecessor count, the eight attached runtime mappings, and the expanded correspondence map describe different scopes.

3.4 Recorded runtime checks

The documented verifier and exact fixture suite were run from the archived runtime payload. Both passed. The verification log is attached in the companion's `build/` directory. All 65 files extracted from the original machine archive retain their original bytes.

An initial invocation used an interpreter without SymPy and failed when the blockification action imported that dependency. The successful replay used an existing environment containing SymPy. No runtime action implementation was changed to resolve that environment failure. The two audited statement corrections were applied separately and have their own provenance record.

> Typesetting this companion does not execute Lean. Lean checks are reported in the verification index and explained in the formal companion. The runtime status file is retained with its original source identity.

3.5 What PASS means

A runtime-verifier pass means the named package checks succeeded. A fixture pass means a concrete exact computation satisfied its assertion. A transport-gate pass means all operation flags that this runtime checked were accepted. None of these strings, taken alone, means that the entire 108-record foundation has been formally proved by the runtime.

The full source proof catalog is included so that a reader can inspect mathematical arguments beyond the runtime. Editorial reproduction of those arguments is not an adversarial proof audit or a new formal verification claim.

4 Executable action contracts

4.1 Input and output conventions

The API is a Python command that emits JSON on successful calls. Exact arithmetic actions use SymPy matrices. Integers and rational strings such as `"3/7"` preserve rational values. JSON floating-point numbers should be avoided when the intended claim is an exact rational identity. Returned scalar entries are strings.

Payload shapes are conventional rather than comprehensively validated at the API boundary. Supply nonempty compatible square projector matrices and compatible operators or vectors. Malformed inputs can raise Python exceptions. The command parser does not present a universal contract for structured errors.

Most computational actions do not call packet validation automatically. Validate the packet family before interpreting block or analysis output through the theorems. With invalid projectors, a matrix expression may still be computable while its theorem interpretation fails.

4.2 The six exact finite actions

Action	Payload	Returned information
`VALIDATE_PACKET_SYSTEM`	`projectors`	Idempotence, annihilation, completeness, and their conjunction.
`ANALYZE_PACKETS`	`projectors`, `vector`	Ambient packet vectors and a reconstruction Boolean.
`BLOCKIFY_OPERATOR`	`projectors`, `operator`	Target/source array of ambient transfer matrices.
`RECONSTRUCT_OPERATOR`	`blocks`	Sum of every supplied block. Compatibility with a packet family is not independently checked.
`COMPOSE_TRANSFER_BLOCKS`	`projectors`, `S`, `T`	Array of block products and equality with directly resolved ST.
`LOWER_INTERTWINER`	`P`, `Q`, `T`, `S`, `U`	Global equation, local equations, residuals, and local conjunction.

4.3 The two semantic gate actions

`CHECK_REALIZATION_PROFILE` and `TRANSPORT_THEOREM` share the same runtime gate. Their payload contains a record ID and a profile. The gate reads that record's direct `action_required_operations` and checks the corresponding profile flags.

The accepted flags are true, `VERIFIED`, or `PRESERVED`. False and `NOT_PRESERVED` produce a negative list. Other or missing values are unknown. Any negative or unknown required operation blocks the gate.

This archived implementation does not evaluate the profile's `verified_hypotheses`, does not use the transitive operation closure as its required set, and does not execute the returned boundary guards. It also does not construct a target theorem, a proof object, or an independently verified realization functor. The preservation flags are declarations supplied by the caller.

Thus the action named `TRANSPORT_THEOREM` is an operation-profile gate. A consumer must separately justify target hypotheses, source-to-target interpretation, and any constructions used by the proof. Unknown flags fail closed only within the operation list actually inspected by the function.

4.4 Planner-only and fixture-only actions

The full registry in Appendix C makes execution status explicit. A planner-only action can identify a mathematically meaningful next step without implementing it. A fixture-only association records that a finite check exercises related mathematics. Calling an unimplemented action through `execute` exits with an unsupported-action error.

5 An exact worked execution

5.1 Oblique packets and reconstruction

Use

$$P_0 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}, \qquad P_1 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \qquad x = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$

These are complementary idempotents but are not self-adjoint under the usual Euclidean inner product. The example therefore exercises the algebraic packet hypotheses without adding orthogonality.

```
python runtime/machine_api.py execute VALIDATE_PACKET_SYSTEM tests/packet_input.json
python runtime/machine_api.py execute ANALYZE_PACKETS tests/analyze_input.json
```

```
{
  "complete": true,
  "idempotent": true,
  "pairwise_annihilating": true,
  "valid": true
}
```

The analysis output is exact:

```
{
  "packets": [
    [
      [
        "-2"
      ],
      [
        "0"
      ]
    ],
    [
      [
        "5"
      ],
      [
        "5"
      ]
    ]
  ],
  "reconstructs": true
}
```

The two vectors $(-2, 0)^t$ and $(5, 5)^t$ sum to x. Each belongs to its own projector range. The runtime's zero-based packet indexing corresponds to the displayed P_0, P_1.

5.2 Resolving and rebuilding an operator

Take

$$T = \begin{pmatrix} 2 & 3 \\ 5 & 7 \end{pmatrix}.$$

The command `BLOCKIFY_OPERATOR` returns four ambient 2×2 matrices. In target/source order, they are

$$T_{00} = \begin{pmatrix} -3 & 3 \\ 0 & 0 \end{pmatrix}, \quad T_{01} = \begin{pmatrix} 0 & -7 \\ 0 & 0 \end{pmatrix}, \quad T_{10} = \begin{pmatrix} 5 & -5 \\ 5 & -5 \end{pmatrix}, \quad T_{11} = \begin{pmatrix} 0 & 12 \\ 0 & 12 \end{pmatrix}.$$

Their sum is T. The exact reconstruction API returns:

```
{
  "operator": [
    [
      "2",
      "3"
    ],
    [
      "5",
      "7"
    ]
  ]
}
```

The input and output JSON files for this reconstruction are supplied in `examples/`. The reconstruction action performs the block sum. Its interpretation as recovery of the original operator uses the validated packet system and the way those blocks were formed.

5.3 Composing resolved operators

Now set

$$S = \begin{pmatrix} 11 & 13 \\ 17 & 19 \end{pmatrix}, \qquad ST = \begin{pmatrix} 87 & 124 \\ 129 & 184 \end{pmatrix}.$$

The runtime forms the blocks of S and T, then sums over the middle packet index. It compares the resulting array with the directly resolved product. The supplied worked output has `equals_direct: true`. The full block arrays are retained in `examples/composition.json`.

The comparison checks a concrete instance of the multiplication identity. The general identity follows from inserting the complete family between the factors, as in the source proof reproduced in Appendix A.

5.4 An intertwining computation

Let

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad Q_i = UP_iU^{-1}, \qquad S' = UTU^{-1} = \begin{pmatrix} 7 & 3 \\ 5 & 2 \end{pmatrix}.$$

The target packets are $Q_0 = \left(\begin{smallmatrix} 1 & -2 \\ 0 & 0 \end{smallmatrix}\right)$ and $Q_1 = \left(\begin{smallmatrix} 0 & 2 \\ 0 & 1 \end{smallmatrix}\right)$. The global identity is $UT = S'U$.

The recorded execution of `LOWER_INTERTWINER` returns true for both the global identity and the conjunction of the four local equations. Each local residual is the zero matrix. One complete local result is:

```
{
  "holds": true,
  "lhs": [
    [
      "-3",
      "3"
    ],
    [
      "0",
      "0"
    ]
  ],
  "residual": [
    [
```

```
      "0",
      "0"
    ],
    [
      "0",
      "0"
    ]
  ],
  "rhs": [
    [
      "-3",
      "3"
    ],
    [
      "0",
      "0"
    ]
  ],
  "source": 0,
  "target": 0
}
```

This example deliberately changes packet coordinates as well as the operator. The complete input and four outputs are retained in `examples/intertwining_input.json` and `examples/intertwining.json`.

5.5 Inspecting the dependency set

The API exposes the dependencies used by the intertwining record linked to the source:

```
python runtime/machine_api.py deps PSC-I-T-BLOCK-INTERTWINING --transitive
```

```
{
  "dependencies": [
    "PSC-I-D-ANALYSIS-SYNTHESIS",
    "PSC-I-D-PACKET-SYSTEM",
    "PSC-I-D-TRANSFER-BLOCK",
    "PSC-I-T-BLOCK-MULTIPLICATION",
    "PSC-I-T-EXACT-PACKET-COORDINATES",
    "PSC-I-T-OPERATOR-RECONSTRUCTION"
  ],
  "record": "PSC-I-T-BLOCK-INTERTWINING"
}
```

This is a dependency closure, not a newly generated proof. Reading each named record supplies its exact source statement, proof, and evidence coordinates.

6 Transport profiles in use

6.1 Finite packet multiplication

The packet-linear profile declares preservation of the direct operations required by packet multiplication. The command

```
python runtime/machine_api.py transport PSC-I-T-BLOCK-MULTIPLICATION profiles/packet_linear.json
```

returns:

```
{
```

```
  "boundary_guards": [],
  "not_preserved": [],
  "record": "PSC-I-T-BLOCK-MULTIPLICATION",
  "required_operations": [
    "composition",
    "finite_sum",
    "packet_idempotents",
    "transfer_blocks"
  ],
  "status": "PASS",
  "unknown": []
}
```


The pass applies to the listed operation flags. It does not construct the target packet system or independently check a concrete functor's preservation laws.

6.2 A countable result is blocked

Using the same profile for the countable unconditional packet theorem returns:


```
{
  "boundary_guards": [],
  "not_preserved": [
    "completion",
    "topology"
  ],
  "record": "PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS",
  "required_operations": [
    "completion",
    "packet_idempotents",
    "topology"
  ],
  "status": "BLOCKED",
  "unknown": []
}
```


The profile does not preserve the topological and completion operations required by this gate. A finite linear profile therefore cannot be used to pass this countable request.

The source theorem additionally requires a real or complex Banach space, unconditional reconstruction, and a uniform multiplier bound. These hypotheses remain visible in its statement. Supplying permissive operation flags would not prove those hypotheses. The richer source contract and the narrower executable gate must be kept distinct.

6.3 Changing a profile without changing evidence

A profile file is a declaration supplied to the runtime. Editing its flags changes the gate's output. It does not create mathematical preservation evidence. A downstream system that wants a transport certificate must retain the arguments or formal certificates justifying those flags and the target hypotheses.

To use the gate, identify the exact record, inspect its source and dependency context, establish a target realization, justify the relevant preservation properties and hypotheses, then use the gate as an explicit consistency check. Its output can be attached to the resulting evidence record as one component.

7 Reproducing the computational material

7.1 Documented payload checks

From the runtime directory identified in the source register, use an interpreter containing SymPy:

```
python runtime/verify_release.py
python runtime/exact_fixture_suite.py
python runtime/nonzero_packet_regression.py
```

The first script checks entrypoint identity, record IDs, graph acyclicity and resolution, AST coverage, consistency of the action registry, source hash, selected API queries and actions, structured transport response, and fixture replay. The second emits the detailed fixture results. The latter does not replace a theorem prover.

7.3 A source change is a new evidence task

If a source statement changes, its source and statement identity change. The KIR, AST, dependency annotations, operation profile, formal map, and tests must be reviewed for impact. A recorded pass cannot be assigned to a new statement solely because the stable ID stayed the same.

The catalogue is generated from the archived mathematical source and records each statement and proof hash. The archived runtime payloads retain their own evidence. The additional traditional models and visual appendix are outside the 108-record inventory. They are not silently counted as extra machine records.

7.4 Runtime interfaces and mathematical statements

The source archive retains the computational source records. This companion supplies the statement catalogue and the runtime's exact transport-gate specification. Planner metadata remains distinct from executed proof machinery.

A Mathematical record catalogue

This catalogue is generated from the archived mathematical inventory's 108 stable records. Every statement and attached proof comes from that inventory. The source locator and statement hash identify the exact archived wording. Cross-references are linked to stable records in this catalog.

The formal-status line reports the reviewed correspondence map, not the older machine payload's eight-entry Lean map. Scope notes are part of each correspondence: "partial" is not a formal certificate for the complete source statement. The full declaration lists and source hashes are preserved in `formal/coverage/COVERAGE_108.json`.

The executable runtime contracts and runtime maps remain separate evidence for their stated scopes. Reproducing a source proof in this catalog does not give that proof a Lean certificate.

A.1 Foundations of seed generation

1. Coefficient and module regimes

Stable ID. `PSC-I-D-COEFFICIENT-REGIME`

Kind. definition

Statement.

All modules are left modules unless stated otherwise. The algebraic regime consists of a unital ring A and a unital left module M over A. When linear dimensions, tensor products, quivers, or operator blocks are used, A is a unital algebra over $\mathbb{k}$, where $\mathbb{k}$ is a field, and M is a vector space over $\mathbb{k}$. The topological regime consists of a Hausdorff topological vector space over a complete valued field, together with a unital algebra of continuous linear operators. Every infinite sum, completion, quotient topology, and closure below is interpreted only in the regime explicitly declared in its statement.

Formal correspondence. documentary. Regime declarations are distributed through Lean typeclass hypotheses, not one encoded regime object.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 7–10.

Statement SHA-256. `3350ba8d143f60c74ec0f3c1ea24b7208cc0346c6f292b8590f68a9520efd7cb`

2. Canonical theorem record

Stable ID. `PSC-I-D-THEOREM-RECORD`

Kind. definition

Statement.

A canonical theorem record is the tuple

$$(\mathrm{ID}, \mathrm{status}, \mathrm{hypotheses}, \mathrm{statement}, \mathrm{dependencies}, \mathrm{proof}, \mathrm{exports}, \mathrm{boundary}).$$

A downstream paper cites the stable record ID. A domain specialization may add domain hypotheses, but it does not repeat a generic proof from this paper.

Formal correspondence. documentary. Publication metadata. JSON records are not Lean proof declarations.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 14–22.

Statement SHA-256. `cd1763a1091beb27f77276664bfaa37d65f73f2b14b41b36f4b9692284853761`

3. Symbolic scope

Stable ID. `PSC-I-B-SYMBOLIC-SCOPE`

Kind. boundary

Statement.

This edition proves the algebraic and topological statements printed below. It does not assert historical novelty, independent external certification, automatic discovery of useful packet systems, nonlinear obstruction theory, multilevel refinement, arithmetic congruences, quantum gravity, automorphy, or local or global Langlands reciprocity.

Formal correspondence. scope boundary. Scope exclusion, not an additional positive theorem.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 25–28.

Statement SHA-256. `de436ff66bb1b236ea21bf9d3cf19c10b67981488cc3ab97eadc5b05708a39ba`

4. Typed operator grammar

Stable ID. `PSC-I-D-TYPED-GRAMMAR`

Kind. definition

Statement.

Let $Q = (Q_0, Q_1, s, t)$ be a quiver with finite vertex set. A typed operator grammar over $\Bbbk$ is a pair

$$\mathfrak{G} = (Q, I),$$

where $I \lhd \Bbbk Q$ is an ideal on both sides generated by linear combinations over $\Bbbk$ of paths having common source and common target. Write

$$\Lambda_{\mathfrak{G}} := \Bbbk Q / I.$$

A realization of $\mathfrak{G}$ is a family $(X_v)_{v \in Q_0}$ and linear maps

$$X_a : X_{s(a)} \longrightarrow X_{t(a)} \qquad (a \in Q_1)$$

that annihilate every relation in I. A morphism $f : X \to Y$ of realizations is a family of linear maps $f_v : X_v \to Y_v$ satisfying

$$f_{t(a)} X_a = Y_a f_{s(a)} \qquad (a \in Q_1).$$

Formal correspondence. definition represented. Finite vertex set, arbitrary arrow set, source/target maps, independent vector spaces and typed arrow maps; relations are finite linear combinations of paths with fixed endpoints. The more general free-algebra relation format is specialized both to homogeneous typed path relations and to all representatives of an actual TwoSidedIdeal.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 37–57.

Statement SHA-256. `282b22656cc22122458c5c399315c5468190e48cc5578ef5258b9a5246a905c9`

5. Collapsed one-space realization

Stable ID. `PSC-I-D-COLLAPSED-REALIZATION`

Kind. definition

Statement.

For a typed realization X, define its realization on a single space by

$$|X| := \bigoplus_{v \in Q_0} X_v.$$

Every vertex idempotent acts as a coordinate projector on $|X|$, and every path acts as a block operator from its source space to its target space.

Formal correspondence. definition represented. Total space is the finite product, canonically the finite direct sum. Vertex operators are single-after-projection maps and arrows are single-after-arrow-after-projection. Induction on the genuinely source/target-indexed TypedPath proves the actual block-composition action, including the zero-length vertex path.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 60–67.

Statement SHA-256. `79fcb85b9e889dae8d8305bb688123c8a54c9d9a18a5b86d37765b76b25a82aa`

6. Typed realization and module equivalence

Stable ID. `PSC-I-T-TYPED-MODULE-EQUIVALENCE`

Kind. theorem

Statement.

The category of realizations of $\mathfrak{G}$ is equivalent to the category of unital left modules over $\Lambda_{\mathfrak{G}}$. Explicitly, a realization X determines the collapsed space $|X|$ of `PSC-I-D-COLLAPSED-REALIZATION` with the vertex idempotent e_v acting as the coordinate projector onto X_v and each arrow acting in the block between its declared source and target. Conversely, a unital module M over $\Lambda_{\mathfrak{G}}$ determines the realization

$$M_v = e_v M, \qquad M_a(x) = ax.$$

The two constructions are inverse up to canonical natural isomorphism.

Proof. Because Q_0 is finite, $1 = \sum_{v \in Q_0} e_v$ in $\Bbbk Q$, the e_v are pairwise orthogonal, and the same identities hold in $\Lambda_{\mathfrak{G}}$. Given a realization, define the action of a path by composition of its arrow maps and extend linearly. Every element of I acts as zero, so the action factors through $\Lambda_{\mathfrak{G}}$. The vertex idempotents give

$$|X| = \bigoplus_v e_v|X| = \bigoplus_v X_v.$$

Conversely, if M is unital, then

$$M = 1M = \sum_v e_v M.$$

Orthogonality makes the sum direct. For an arrow $a : v \to w$, the identities $a = e_w a e_v$ imply

$$a(e_v M) \subseteq e_w M.$$

The relations in I act as zero because M is a module over $\Lambda_{\mathfrak{G}}$. Applying both constructions returns the original vertex spaces and arrow maps, or the original module decomposition, through the evident identity maps. These identifications commute with morphisms, giving an equivalence of categories. □

Formal correspondence. covered. The grammar algebra is constructed by FreeAlgebra and RingQuot, with complete orthogonal vertex idempotents and exact source/target arrow corners. Collapse is full, faithful and essentially surjective, using actual vertex ranges and their analysis/synthesis equivalence. Action objects are then proved equivalent to ordinary ModuleCat for all unital left modules; the coefficient action is recovered through algebraMap. A separate quotient comparison uses the actual TwoSidedIdeal ring congruence and proves both inverse identities. No category equivalence, split decomposition, path action or module structure is supplied as an assumption.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 70–80; proof lines 84–98.

Statement SHA-256. `a7fb9fcf8a2ff9a5b6ca94fba965afd801c3acb93f1fe6e0323b9fd3fab8470b`

7. Realized operator algebra

Stable ID. `PSC-I-D-REALIZED-ALGEBRA`

Kind. definition

Statement.

For a realization X, the realized operator algebra is

$$\mathcal{A}_X := \mathrm{im}(\Lambda_{\mathfrak{G}} \longrightarrow \mathrm{End}_{\Bbbk}(|X|)) .$$

It records the actual operators in the realization and may be a proper quotient of the abstract grammar algebra.

Formal correspondence. definition represented. The realized algebra is literally the range of the constructed algebra homomorphism into Module.End. It is not a supplied equivalent algebra or an abstract surrogate.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 114–121.

Statement SHA-256. `43b505da15a63ccf3534f295a12aaadd6cd3d7d4f7e2ab5dba25d3d5cf63d0bc`

8. Action factorization and hull invariance

Stable ID. `PSC-I-P-ACTION-FACTORIZATION`

Kind. proposition

Statement.

Let $S \subseteq |X|$. The generated submodule under the grammar algebra equals the generated submodule under the realized operator algebra:

$$\Lambda_{\mathfrak{G}} S = \mathcal{A}_X S.$$

Consequently, every statement about seed rank depends only on the realized action, not on redundant relations in a chosen presentation of the grammar.

Proof. The action map $\rho : \Lambda_{\mathfrak{G}} \twoheadrightarrow \mathcal{A}_X$ is surjective by definition and satisfies $\lambda s = \rho(\lambda)s$. Hence every grammar orbit vector lies in $\mathcal{A}_X S$. The reverse inclusion holds because every element of $\mathcal{A}_X$ is $\rho(\lambda)$ for some λ. □

Formal correspondence. covered. The two actual spans are compared by span induction and range-surjectivity. The cardinal seed ranks are equal via the identity semilinear bijection into the actual image algebra. Arbitrary seed sets and zero modules are included; there is no finite-generation assumption.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 124–131; proof lines 135–137.

Statement SHA-256. `472b9117cf7ed5ed76713d4b131141c2e9f00b9b6dd827bcb2515705d21e94f1`

9. Linear seed system

Stable ID. `PSC-I-D-SEED-SYSTEM`

Kind. definition

Statement.

A linear seed system is a tuple

$$\mathfrak{S} = (X, \mathcal{A}, S, \mathcal{P}, \tau),$$

where X is a unital left module over $\mathcal{A}$, $\mathcal{A}$ is fixed independently of the target element, $S = (s_1, \dots, s_r)$ is a finite distinguished seed family, $\mathcal{P}$ is either empty or a declared packet system, and τ is either the algebraic convention or a specified Hausdorff topology in which the declared operator action is continuous.

Formal correspondence. documentary. Algebraic and continuous packet cases are represented by concrete operator algebras, packet objects and arbitrary seed sets, which include every finite distinguished family after passage to its range. The record is a regime/data definition, not an assertion that one bundled structure covers the optional packet-free and topological regimes. Optional fields are represented in separate structures.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 145–152.

Statement SHA-256. `f1b9660868ad56e506bd33c2ab474312509199727cc8c23a64d4334641974e26`

10. Generated hull

Stable ID. `PSC-I-D-GENERATED-HULL`

Kind. definition

Statement.

For $S \subseteq X$, define

$$\mathrm{Hull}_{\mathcal{A}}(S) := \sum_{s \in S} \mathcal{A}s.$$

In the topological regime define

$$\mathrm{Hull}_{\mathcal{A},\tau}(S) := \overline{\mathrm{Hull}_{\mathcal{A}}(S)}^{\tau}.$$

For a singleton $S = \{s\}$, write $\mathrm{Hull}_{\mathcal{A}}(s)$.

Formal correspondence. definition represented. The scalar span of algebra-orbit vectors is proved equal as a set to the module span. The general topological operator hull is an actual Submodule and its topologicalClosure is used in the topological development.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 155–166.

Statement SHA-256. `859743fdac21ebb8ba120148cbf03f5511974f6e3356b5f733a3beea316b55f5`

11. Natural presentation

Stable ID. `PSC-I-D-NATURAL-PRESENTATION`

Kind. definition

Statement.

A seed presentation is natural relative to a declared grammar when the primitive operators and relations are fixed independently of the target, the relation system is finite or effectively enumerable, the topology is intrinsic to the realization, no primitive operator directly stores the complete target, and the presentation exposes a reusable invariant, theorem, or computation not encoded by a primitive chosen for the target.

Formal correspondence. documentary. A methodological criterion rather than a decidable Lean predicate.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 171–174.

Statement SHA-256. `94104ee5b6710359c3fa14a5febde34de8455668f83dfc5f49cf3e5028770033`

12. Algebra generation and module generation are distinct

Stable ID. `PSC-I-P-ALGEBRA-MODULE-GENERATION`

Kind. proposition

Statement.

Every nonzero unital algebra A, regarded as a left module over itself, has

$$\operatorname{srank}_A(A) = 1,$$

with cyclic seed 1_A. This conclusion is independent of the number of algebra generators required to present A as an algebra.

Proof. For every $a \in A$, one has $a = a1_A$. Hence $A1_A = A$. The number of operations needed to generate A as an algebra is a different invariant and does not enter the module calculation. □

Formal correspondence. covered. The regular action is multiplication on A. The singleton unit is proved generating by surjectivity of scalar evaluation; the nontrivial regular module then has seedRank exactly one. No assertion about algebra presentations is used as a premise.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 177–184; proof lines 188–190.

Statement SHA-256. `9ba2f9c1f32694b86a7d9ac63ef11d2b3c46ee5c38307f3abcc3c12048cf91d0`

13. Target-dependent operators

Stable ID. PSC-I-B-TARGET-DEPENDENT-OPERATORS

Kind. boundary

Statement.

For a fixed seed s and target x, the existence of a specially defined operator A_x satisfying $A_x s = x$ does not establish a natural seed calculus unless A_x belongs to a grammar fixed independently of the target satisfying PSC-I-D-NATURAL-PRESENTATION.

Formal correspondence. scope boundary. Target-independence restriction remains documentary.

Source locator. editable/traditional/02_foundations.tex; statement lines 192–195.

Statement SHA-256. aece204c76d6b378ec937375eebd02a43d0a48dbed1f94054b4ed514ef573dd1

14. Seed evaluation map

Stable ID. PSC-I-D-SEED-EVALUATION

Kind. definition

Statement.

Let A be a unital ring, let M be a left module over A, and let $S = (s_1, \dots, s_r) \in M^r$. Define the linear map over A

$$\varepsilon_S : A^{\oplus r} \longrightarrow M, \qquad (a_1, \dots, a_r) \longmapsto \sum_{j=1}^{r} a_j s_j.$$

Formal correspondence. definition represented. multiSeedMap is the actual Finsupp linear combination map; finite families give the stated finite free module.

Source locator. editable/traditional/02_foundations.tex; statement lines 215–223.

Statement SHA-256. db3e882c18dfdf5770d2779811295f6d287e9ffa95dbfb79bec513f96cba185d

15. Finite free presentation

Stable ID. PSC-I-T-FREE-PRESENTATION

Kind. theorem

Statement.

For the map in PSC-I-D-SEED-EVALUATION,

$$\operatorname{im} \varepsilon_S = \operatorname{Hull}_A(S), \qquad N_S := \operatorname{Ker} \varepsilon_S \leq A^{\oplus r},$$

and the induced map is an isomorphism of left modules over A

$$\boxed{A^{\oplus r} / N_S \cong \operatorname{Hull}_A(S).}$$

Consequently, S generates M if and only if ε_S is surjective.

Proof. The image consists exactly of finite sums $\sum_j a_j s_j$, which is $\operatorname{Hull}_A(S)$. Because ε_S is linear over A, its kernel is a left submodule. The first isomorphism theorem gives

$$A^{\oplus r} / N_S \cong \operatorname{im} \varepsilon_S = \operatorname{Hull}_A(S).$$

The final equivalence is the equality $\operatorname{im} \varepsilon_S = M$. □

Formal correspondence. covered. Actual range/span equality, quotient equivalence and generation/-surjectivity equivalence are present.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 226–239; proof lines 244–250.

Statement SHA-256. `6e2db91e9df6b69a04e531d9b552fe7148f878663bcd3600c36c94ac62592ef8`

16. Minimal free-cover characterization

Stable ID. `PSC-I-C-MINIMAL-FREE-COVER`

Kind. corollary

Statement.

For every finitely generated left module M over A,

$$\boxed{\operatorname{srank}_A(M) = \min\{r \geq 0 : \text{there exists a surjection } A^{\oplus r} \twoheadrightarrow M\}.}$$

Proof. A seed family of size r gives a surjection by `PSC-I-T-FREE-PRESENTATION`. Conversely, the images of the standard basis vectors under any surjection $A^{\oplus r} \twoheadrightarrow M$ form a generating family of size at most r. □

Formal correspondence. covered. seedFinrank_isLeast_free_surjection proves the finite minimum, not an executable basis algorithm.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 267–275; proof lines 279–281.

Statement SHA-256. `8e4ae31565265679e9c8bf397291baf65b396ad21bb88e7f00b8d17328b1cb00`

17. One-seed presentation

Stable ID. `PSC-I-C-ONE-SEED-PRESENTATION`

Kind. corollary

Statement.

For $s \in M$, the annihilator

$$\operatorname{Ann}_A(s) := \{a \in A : as = 0\}$$

is a left ideal and

$$\boxed{A/\operatorname{Ann}_A(s) \cong \operatorname{Hull}_A(s).}$$

Proof. The evaluation map for one seed is $a \mapsto as$. Its kernel is $\operatorname{Ann}_A(s)$, which is therefore a left ideal, and `PSC-I-T-FREE-PRESENTATION` gives the quotient isomorphism. □

Formal correspondence. covered. oneSeedMap has the stated annihilator kernel and its quotient is the literal cyclic range.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 283–293; proof lines 297–299.

Statement SHA-256. `92a43462490ec1465fe4129738284f9c3f53e2d8bb7b63f980c1d03a0fbbb2ed`

18. Two-sided annihilator criterion

Stable ID. `PSC-I-P-ANNIHILATOR-TWO-SIDED`

Kind. proposition

Statement.

For $s \in M$, the left ideal $L = \mathrm{Ann}_A(s)$ is an ideal on both sides if and only if

$$L\ \mathrm{Hull}_A(s) = 0.$$

If s is cyclic, these conditions are equivalent to

$$\mathrm{Ann}_A(s) = \mathrm{Ann}_A(M).$$

Proof. The left ideal L is an ideal on both sides exactly when $ab \in L$ for every $a \in L$ and $b \in A$. This is equivalent to

$$0 = (ab)s = a(bs) \qquad (a \in L,\ b \in A).$$

By additivity, the last identity is equivalent to $L\,\mathrm{Hull}_A(s) = 0$.

If s is cyclic, then $\mathrm{Hull}_A(s) = M$. Hence the preceding condition is $L \subseteq \mathrm{Ann}_A(M)$. The reverse inclusion always holds because an operator annihilating all of M annihilates s. Thus equality follows. Conversely, $\mathrm{Ann}_A(M)$ is an ideal on both sides, so equality implies that L is an ideal on both sides. □

Formal correspondence. covered. vectorAnnihilator is the actual kernel of a↦a•s. Right multiplication closure is equivalent to annihilating Submodule.span A {s}; Ideal.IsTwoSided supplies the standard two-sided-ideal interface. Under singleton generation, the actual ideal equals Module.annihilator A M. Noncommutative order is a*b and (a*b)•s=a•(b•s).

Source locator. `editable/traditional/02_foundations.tex`; statement lines 302–312; proof lines 316–325.

Statement SHA-256. `dd991bd6ce28ee7f94302ed9dd3fb00719cf26de3b6ccf740f441b118bdd8e54`

19. Completion of a dense free presentation

Stable ID. `PSC-I-T-COMPLETED-PRESENTATION`

Kind. theorem

Statement.

Let X be a complete Hausdorff locally convex space, let $\mathcal{A} \subseteq \mathrm{End}(X)$ act continuously, and let $S = (s_1, \dots, s_r)$. Put

$$E := \mathcal{A}^{\oplus r} / N_S$$

algebraically and let $j : E \to X$ be the injective map induced by ε_S. Give E the locally convex topology transported from $j(E)$, equivalently the topology generated by

$$\widetilde{p}(e) := p(j(e))$$

for the continuous seminorms p on X. Every $a \in \mathcal{A}$ acts continuously on E in the transported topology and therefore extends to $\widehat{E}$. If $\mathrm{Hull}_{\mathcal{A}}(S)$ is dense in X, then j extends uniquely to a topological isomorphism of modules over $\mathcal{A}$

$$\boxed{\widehat{E} \cong X.}$$

Proof. By PSC-I-T-FREE-PRESENTATION, j is an algebraic isomorphism of modules over $\mathcal{A}$ from E onto $\mathrm{Hull}_{\mathcal{A}}(S)$. The transported seminorms make it a topological isomorphism onto that subspace. For $a \in \mathcal{A}$ and a continuous seminorm p on X, continuity of $a : X \to X$ supplies continuous seminorms controlling $p(aj(e))$, so the induced action on E is continuous and extends uniquely to $\widehat{E}$. The subspace is dense by hypothesis. A Cauchy net in E is mapped to a Cauchy net in X, which converges because X is complete. This defines a continuous extension $\widehat{j} : \widehat{E} \to X$. For any $x \in X$, choose a net (e_λ) in E with $j(e_\lambda) \to x$. The transported topology makes (e_λ) Cauchy, so it converges to some $\xi \in \widehat{E}$, and continuity gives $\widehat{j}(\xi) = x$. Thus $\widehat{j}$ is surjective. If $\widehat{j}(\xi) = 0$, represent ξ by a Cauchy net (e_λ). Then $j(e_\lambda) \to 0$, so every defining seminorm $\widetilde{p}(e_\lambda)$ tends to zero. Thus $e_\lambda \to 0$ in the completion and $\xi = 0$. Hence $\widehat{j}$ is bijective. Its inverse is continuous by the universal property of completion, or directly because the topology on E was transported from X. □

Formal correspondence. covered. The actual finite free evaluation A^(Fin r)→X is used, its actual kernel is quotiented, and UniformSpace.comap pulls back the target uniformity to that quotient. Dense range is derived from dense module span. AbstractCompletion.compareEquiv supplies a uniform equivalence, linearity over A is proved by completion density, and uniqueness follows from equality on the dense image. Constant action uniform continuity is inherited through the actual quotient embedding; scalar action on Completion is the usual extension. The local-convexity pullback lemma uses ordered scalars (in particular ℝ, including underlying real spaces for complex spaces); the stronger uniform-module completion proof does not require local convexity.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 333–347; proof lines 351–353.

Statement SHA-256. `23708a85485e7162e31b64ee0a8fb316bec231a1aadaa0e00f0a08f70dfe1721`

20. Seed rank

Stable ID. PSC-I-D-SEED-RANK

Kind. definition

Statement.

For a unital left module M over A, define

$$\mathrm{srank}_A(M) := \min\{|S| : AS = M\}.$$

For a topological module (X, τ), define

$$\mathrm{srank}_{\mathcal{A},\tau}(X) := \min\{|S| : \overline{AS}^{\tau} = X\}.$$

Set $\mathrm{srank}_A(0) = 0$. When the value is finite, it is the minimal number of module generators.

Formal correspondence. definition represented. The operator hull is the scalar span of actual action values; for an acting algebra this is AS. The cardinal infimum ranges over all densely generating subsets, and existence of a minimum is proved from identity membership and cardinal well ordering.

Empty generators and infinite cardinals are included. The direct moduleSeedRank definition uses Dense of the literal acting-ring module span, and a proved equality identifies it with the integer-linear operator-family definition even for a nonfaithful action.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 370–381.

Statement SHA-256. `81a23b39cf2b6194ae8b0bc84ac270ee5ef86e4703b5fb5411f5fa9519aafae9`

21. Cyclic characterization

Stable ID. `PSC-I-P-CYCLIC-CHARACTERIZATION`

Kind. proposition

Statement.

For $M \neq 0$,

$$\operatorname{srank}_A(M) = 1 \quad \Longleftrightarrow \quad M \text{ contains a cyclic vector.}$$

Proof. Both conditions say that there is $m \in M$ with $Am = M$. □

Formal correspondence. covered. The nontriviality hypothesis agrees with the current publication.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 384–392; proof lines 396–398.

Statement SHA-256. `de62f3fddcf5cefb429536723a8933a119585faa47b6f2e51745ea055ae6a9cc`

22. Monotonicity under enlargement of the acting algebra

Stable ID. `PSC-I-P-ACTING-ALGEBRA-MONOTONICITY`

Kind. proposition

Statement.

Let $B \subseteq A \subseteq \operatorname{End}_{\mathbb{k}}(X)$ be unital operator algebras with the same identity. Then

$$\boxed{\operatorname{srank}_A(X) \leq \operatorname{srank}_B(X).}$$

The same statement holds for topological seed rank when both algebras act continuously.

Proof. For every $S \subseteq X$, one has $BS \subseteq AS$. Hence every generating family over B is also a generating family over A. Taking closures gives the topological statement. □

Formal correspondence. covered. For unital operator algebras, actual family inclusion implies operator-hull inclusion. Closure monotonicity transports a minimum generating subset. This exactly supplies the topological clause, combined with the previously mapped algebraic semilinear rank theorem.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 400–407; proof lines 411–413.

Statement SHA-256. `115d0049078122c1d6c43bb8ec53237a6efae32ef13330b8e31d8bf77e68eb56`

23. Quotient monotonicity

Stable ID. `PSC-I-T-QUOTIENT-MONOTONICITY`

Kind. theorem

Statement.

Let $\varphi : A \to B$ be a unital ring homomorphism and let

$$\Phi : M \twoheadrightarrow N$$

be a surjective additive map satisfying

$$\Phi(am) = \varphi(a)\Phi(m) \qquad (a \in A,\ m \in M).$$

Then

$$\boxed{\operatorname{srank}_B(N) \le \operatorname{srank}_A(M).}$$

If M and N are topological modules, Φ is continuous and surjective, and the actions are continuous, then

$$\operatorname{srank}_{B,\tau_N}(N) \le \operatorname{srank}_{A,\tau_M}(M).$$

Proof. Let S generate M. For every $n \in N$, choose $m \in M$ with $\Phi(m) = n$ and write $m = \sum_i a_i s_i$. Then

$$n = \sum_i \varphi(a_i)\Phi(s_i),$$

so $\Phi(S)$ generates N.

For the topological statement, let S generate M densely. Given $n = \Phi(m)$, choose a net $m_\lambda \in AS$ converging to m. Continuity gives

$$\Phi(m_\lambda) \longrightarrow n,$$

and every $\Phi(m_\lambda)$ lies in $B\Phi(S)$. Thus $B\Phi(S)$ is dense in N. □

Formal correspondence. covered. The new theorem starts with the printed arbitrary unital ring homomorphism, additive map, scalar compatibility, continuity, and surjectivity. Span induction maps the actual source module span into the target module span; continuous surjectivity transports density. An attained minimum generating set maps to an image of no larger cardinality. No ring-homomorphism surjectivity or faithfulness is assumed. The direct module-span rank is proved equal to the earlier operator-family rank using integer-linear acting operators. This supplies the exact topological clause alongside the separately accepted algebraic theorem.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 421–440; proof lines 445–457.

Statement SHA-256. `ccde1353d26e9f7bc7aef341c35be5304729daa69f8f42274ab70e2f9d7016a7`

24. Isomorphism invariance

Stable ID. PSC-I-C-ISOMORPHISM-INVARIANCE

Kind. corollary

Statement.

An intertwining module isomorphism preserves algebraic seed rank. A topological intertwining isomorphism preserves topological seed rank.

Proof. Apply PSC-I-T-QUOTIENT-MONOTONICITY to the isomorphism and its inverse. □

Formal correspondence. covered. A continuous semilinear module equivalence with continuous inverse preserves the direct module-span cardinal rank, by applying the exact semilinear surjection inequality in both directions. The same-ring case uses identity ring maps. The earlier bidirectional operator-family theorem covers intertwining operator-algebra isomorphisms, and the new rank equality bridges these definitions. This supplies the exact topological clause alongside separately accepted algebraic invariance.

Source locator. editable/traditional/02_foundations.tex; statement lines 464–467; proof lines 471–473.

Statement SHA-256. fc01d1773cdeebcaba8346b323df6df33d48e0f6f7adda763b14307b2e902ca1

25. Short exact sequence bounds

Stable ID. PSC-I-T-EXACT-SEQUENCE-BOUNDS

Kind. theorem

Statement.

Let

$$0 \longrightarrow L \xrightarrow{\iota} M \xrightarrow{\pi} N \longrightarrow 0$$

be a short exact sequence of left modules over A. Then

$$\boxed{\operatorname{srank}_A(N) \leq \operatorname{srank}_A(M) \leq \operatorname{srank}_A(L) + \operatorname{srank}_A(N).}$$

Proof. The lower inequality is quotient monotonicity. For the upper inequality, choose generating families $(\ell_i)_{i\in I}$ of L and $(n_j)_{j\in J}$ of N with the least possible cardinalities. Choose a lift $m_j \in M$ of each n_j. For any $m \in M$, a finite subset $F \subseteq J$ and coefficients $a_j \in A$ give

$$\pi(m) = \sum_{j\in F} a_j n_j.$$

The vector $m - \sum_{j\in F} a_j m_j$ belongs to $\iota(L)$, so it is a finite linear combination of the $\iota(\ell_i)$. The union of the lifted family and the image of the generating family of L therefore generates M. Its cardinality is at most $|I| + |J|$, giving the upper bound. This argument uses finite sums for each vector even when either generating family is infinite. □

Formal correspondence. covered. Actual linear f,g with injectivity, range f=ker g and surjectivity; canonical seedRank=top.spanRank. Minimum generators of N are lifted and combined with images of

minimum generators of L. The proof derives generation of M and the cardinal upper bound; the lower bound is surjective monotonicity. No splitting, finite generation or desired bound is assumed.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 475–488; proof lines 492–498.

Statement SHA-256. `2e4e5af8593f49860709b8eb0b2d2b88577c0bc765a8b9e0c086441aaf142348`

26. Shared direct-sum bounds

Stable ID. `PSC-I-P-SHARED-DIRECT-SUM`

Kind. proposition

Statement.

If one algebra A acts diagonally on $M \oplus N$, then

$$\boxed{\max\{\operatorname{srank}_A(M), \operatorname{srank}_A(N)\} \le \operatorname{srank}_A(M \oplus N) \le \operatorname{srank}_A(M) + \operatorname{srank}_A(N).}$$

Proof. The coordinate projections are intertwining quotients and give the lower bound. For the upper bound, embed minimal generating families of M as $(m_i, 0)$ and of N as $(0, n_j)$. Their union generates by finite sums and has cardinality at most the sum of the two seed ranks. □

Formal correspondence. covered. The actual diagonal A-action on M×N realizes the binary direct sum. Projection surjectivity gives both lower bounds; inclusion of generating sets in the two factors yields cardinal addition for the upper bound.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 506–516; proof lines 520–522.

Statement SHA-256. `613d0bed5e9f4890c069d8d37670d0b8c9cb3a981df5bceb95686c08dc5638c6`

27. Independent direct-sum formula

Stable ID. `PSC-I-T-INDEPENDENT-DIRECT-SUM`

Kind. theorem

Statement.

Let M be a left module over A and N a left module over B. Under the action of the product algebra

$$(a, b)(m, n) = (am, bn),$$

one has

$$\boxed{\operatorname{srank}_{A\times B}(M \oplus N) = \max\{\operatorname{srank}_A(M), \operatorname{srank}_B(N)\}.}$$

Proof. The coordinate projections give the lower bound. Let r be the maximum on the right, regarded as a cardinal, and choose an index set I of cardinality r. Choose minimal generating families for M and N, index them by subsets of I, and fill any unused positions with zeros. Write the resulting families as $(m_i)_{i\in I}$ and $(n_i)_{i\in I}$. For any $m \in M$ and $n \in N$, their finite expressions involve a common finite subset $F \subseteq I$ after zero coefficients have been added. Thus

$$m = \sum_{i\in F} a_i m_i, \qquad n = \sum_{i\in F} b_i n_i,$$

and

$$(m, n) = \sum_{i \in F} (a_i, b_i)(m_i, n_i).$$

The family of paired vectors generates and has cardinality at most r. The empty family handles the case in which both modules are zero. □

Formal correspondence. covered. Actual product-algebra action and the maximum formula are proved.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 524–538; proof lines 542–554.

Statement SHA-256. `b6f497941ccc0be69e3780d70cd1d79cb39363dedb591b81527183724fc58748`

28. Tensor-product bound

Stable ID. `PSC-I-P-TENSOR-BOUND`

Kind. proposition

Statement.

Let A and B be unital algebras over $\mathbb{k}$, let M be a left module over A, and let N be a left module over B. Under the natural $A \otimes_{\mathbb{k}} B$ action,

$$\boxed{\operatorname{srank}_{A \otimes B}(M \otimes_{\mathbb{k}} N) \leq \operatorname{srank}_A(M) \operatorname{srank}_B(N).}$$

Proof. Choose minimal generating families $(m_i)_{i \in I}$ for M and $(n_j)_{j \in J}$ for N. Each vector in either module is a finite linear combination of its generators. Bilinearity therefore expresses every simple tensor as a finite sum of terms

$$(am_i) \otimes (bn_j) = (a \otimes b)(m_i \otimes n_j).$$

Simple tensors span the algebraic tensor product, so the family $(m_i \otimes n_j)_{(i,j) \in I \times J}$ generates it. This family has cardinality at most $|I||J|$, which is the claimed bound. The proof also covers an empty generating family, when the corresponding module and tensor product are zero. □

Formal correspondence. covered. The genuine tensor-product action and cardinal product upper bound are proved.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 557–567; proof lines 571–577.

Statement SHA-256. `b1b31d79b3ffc3d73319922c4a3f2f1dde12a0f774ceda1ae29227e9c63a6d4e`

29. Tensor rank is not generally multiplicative

Stable ID. PSC-I-B-TENSOR-NONEQUALITY

Kind. boundary

Statement.

The inequality in PSC-I-P-TENSOR-BOUND can be strict. Equality requires additional hypotheses on the acting algebras and their module multiplicities and is not exported as a universal law.

Formal correspondence. scope boundary. No universal tensor equality is exported. This boundary is not counted as a positive theorem.

Source locator. editable/traditional/02_foundations.tex; statement lines 584–587.

Statement SHA-256. 9328501ddd219b9e0defabc4233a3e0c6437a8890f6920385de876bc5223e204

30. Scalar-extension monotonicity

Stable ID. PSC-I-P-SCALAR-EXTENSION-MONOTONICITY

Kind. proposition

Statement.

Let $K/\Bbbk$ be a field extension, let A be a unital algebra over $\Bbbk$, and let M be a finitely generated left module over A. Set

$$A_K := K \otimes_{\Bbbk} A, \qquad M_K := K \otimes_{\Bbbk} M.$$

Then

$$\boxed{\operatorname{srank}_{A_K}(M_K) \le \operatorname{srank}_A(M).}$$

Every generating family $S \subseteq M$ extends to the generating family $1 \otimes S \subseteq M_K$.

Proof. Let $S = \{s_1, \dots, s_r\}$ generate M. Every pure tensor $\lambda \otimes m \in M_K$ has an expression

$$m = \sum_{i=1}^{r} a_i s_i,$$

so

$$\lambda \otimes m = \sum_{i=1}^{r} (\lambda \otimes a_i)(1 \otimes s_i).$$

Pure tensors span M_K, hence $1 \otimes S$ generates M_K over A_K. Taking the minimum over S proves the inequality. □

Formal correspondence. covered. The source constructs the natural (K⊗k A)-action on K⊗k M using commuting tensor-factor actions and proves (c⊗a)•(d⊗m)=(c*d)⊗(a•m). Tensor and span induction prove the literal image (1⊗-) of any generating set generates. Minimum-cardinality generation plus cardinality of an image gives the rank inequality.

Source locator. editable/traditional/02_foundations.tex; statement lines 592–607; proof lines 612–624.

Statement SHA-256. 89013f407adf17424cd55db8a6f59a4a9141e34d39a1d350731d9e18d542a3e4

31. Seed-rank witness

Stable ID. `PSC-I-D-SEED-RANK-WITNESS`

Kind. definition

Statement.

A surjective intertwiner $\Phi : M \twoheadrightarrow N$ is a witness for seed rank of order r when $\operatorname{srank}(N) = r$. It certifies $\operatorname{srank}(M) \geq r$.

Formal correspondence. documentary. A named use of the quotient lower bound, not a new proof target.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 632–635.

Statement SHA-256. `81381ceea3d420ff6ba2f0b9378863598308dc392cf597cc1dc0b4d7f84f7cc0`

32. Scalar quotient witness

Stable ID. `PSC-I-P-SCALAR-QUOTIENT-WITNESS`

Kind. proposition

Statement.

Let M be a module over A with coefficient field $\Bbbk$. Suppose M has a quotient N of dimension r on which A acts through a character

$$\chi : A \longrightarrow \Bbbk.$$

Then

$$\boxed{\operatorname{srank}_A(M) \geq r.}$$

Proof. For every $n \in N$, the cyclic submodule An is contained in the scalar span $\Bbbk n$. Therefore q seed images span a subspace of dimension at most q. Generating the quotient of dimension r requires at least r seeds. Quotient monotonicity transfers the lower bound to M. □

Formal correspondence. covered. The character, surjective semilinear map and finite scalar dimension match the printed hypotheses.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 638–648; proof lines 652–654.

Statement SHA-256. `92e35ac7e226b34fce4aa0e1e40a198c1273cc439ac42a8ea2f437f75965a632`

33. Nakayama lemma

Stable ID. `PSC-I-CI-NAKAYAMA`

Kind. classicalinput

Statement.

Let A be a unital ring, let $J \subseteq \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. If $M = N + JM$ for a submodule $N \leq M$, then $M = N$.

Formal correspondence. classical input integrated. The noncommutative input is integrated at the printed finitely generated left-module scope. For any left ideal I contained in Ring.jacobson R, the actual submodule equality N ⊔ I • top = top implies N = top. No commutativity, Noetherian, semisimplicity, finite length, or two-sidedness of I is required for this lifting theorem. The proof derives the radical lifting result from maximal submodules and the ring-radical action inclusion, rather than assuming the desired conclusion.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 669–672.

Statement SHA-256. `68901a557c34762568711f0e6c8b400485a0c4b66477b8f231e899d9a48f5878`

34. Generator number is detected modulo the Jacobson radical

Stable ID. `PSC-I-T-MODULE-TOP`

Kind. theorem

Statement.

Let A be a unital ring, let $J = \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. A family $S \subseteq M$ generates M if and only if its image generates M/JM as a module over A/J. When A/J is semisimple Artinian, this quotient is the usual semisimple top. No semisimplicity of A/J is required for the general equality. Consequently,

$$\boxed{\mathrm{srank}_A(M) = \mathrm{srank}_{A/J}(M/JM).}$$

Proof. If S generates M, its image generates every quotient. Conversely, suppose the image of S generates M/JM and put $N = AS$. Then

$$M = N + JM.$$

Nakayama's lemma gives $M = N$. Thus the same finite families generate before and after passage to the top, and the minimal cardinalities agree. □

Formal correspondence. covered. The complete generating-set/family equivalence and canonical cardinal seedRank equality are proved for arbitrary unital rings and finitely generated left modules using the literal M / (Jac(R) • top), its actual scalar action over R / Jac(R), and an exact quotient-map kernel. Any lifts of a generating quotient family generate upstairs, and both minimum-cardinality inequalities are established. The natural seedFinrank equality is proved as well. No semisimplicity hypothesis is needed. The descriptive semisimple-top sentence is covered by the pinned general IsSemisimpleRing.isSemisimpleModule instance applied to this actual quotient module. This is not an algorithm for computing the radical or finding generators.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 675–683; proof lines 688–694.

Statement SHA-256. `8e3170bbaf10331d7d774260fbdf4b7df0ca59f1c23116f008a1aa5ac2bf07a8`

35. Local algebra formula

Stable ID. `PSC-I-C-LOCAL-ALGEBRA-FORMULA`

Kind. corollary

Statement.

Let A be a local ring with Jacobson radical J and residue division ring $D = A/J$. For every finitely generated left module M over A,

$$\boxed{\operatorname{srank}_A(M) = \dim_D(M/JM),}$$

where the dimension is that of a left vector space over D.

Proof. The module M/JM over A/J is a left vector space over D. Its minimal number of generators is its dimension over D. □

Formal correspondence. covered. The local formula is proved for noncommutative local rings and finitely generated left modules. From IsLocalRing R the source proves uniqueness of the maximal left ideal, identifies the actual Jacobson radical as maximal, constructs DivisionRing (R / Jac(R)) internally, and proves the canonical seedRank equals the left vector-space Module.rank of the actual radical quotient. The natural finite-rank equality is also proved. No commutative residue-field surrogate or caller-supplied division-ring certificate is used.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 709–718; proof lines 722–724.

Statement SHA-256. `50bd885a16ef40862a42e9ddd15b7dd03758b5ee514ee98aab28acf22c1e26ed`

36. Wedderburn–Artin decomposition

Stable ID. `PSC-I-CI-WEDDERBURN-ARTIN`

Kind. classicalinput

Statement.

Every semisimple Artinian ring is isomorphic to a finite product

$$A \cong \prod_{i=1}^{q} M_{d_i}(D_i)$$

of full matrix rings over division rings.

Formal correspondence. classical input integrated. The actual arbitrary-division-ring Wedderburn-Artin existence theorem is now integrated at its semisimple-ring scope. The finite-algebra radical quotient is proved semisimple and its actual decomposition is constructed. These depend on proved pinned Mathlib results, not a custom existence axiom. Compatible finite module decompositions are separately constructed in ProductPresentation and SemisimpleExistence.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 727–734.

Statement SHA-256. `effdfe288b991fcc5cdf7549800403bbb6fd698be57f017880428042cc23a058`

37. Regular-module multiplicities

Stable ID. `PSC-I-L-REGULAR-MODULE-MULTIPLICITIES`

Kind. lemma

Statement.

Let

$$A = \prod_{i=1}^{q} M_{d_i}(D_i)$$

and let $S_i = D_i^{d_i}$ denote the natural simple left module for the ith factor, extended by zero across the other factors. Then, as a left module over A,

$$\boxed{A \cong \bigoplus_{i=1}^{q} S_i^{\oplus d_i}.}$$

Proof. For one factor $M_d(D)$, let e_{jj} be the diagonal matrix units. The left regular module decomposes as

$$M_d(D) = \bigoplus_{j=1}^{d} M_d(D)e_{jj}.$$

Each minimal left ideal $M_d(D)e_{jj}$ is isomorphic to the natural column module D^d. Taking the direct product of factors gives the stated decomposition. □

Formal correspondence. covered. The actual left regular module is explicitly equivalent to its natural columns, then to the displayed finite direct sum of natural-module copies over arbitrary division rings. Coordinate ring homomorphisms define the inflated actions, other factors act by zero, and positive-dimensional natural factors are proved simple. No transpose-as-ring-map or commutative-scalar shortcut is used. Finite products for the inner repeated copies are the canonical finite direct-sum model.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 737–747; proof lines 751–757.

Statement SHA-256. `cd1bf9aa7ec7ad71f24491e1969c318f330a268fb73a29873ff2e9494eb91799`

38. Semisimple seed-rank formula over arbitrary division rings

Stable ID. PSC-I-T-SEMISIMPLE-SEED-RANK

Kind. theorem

Statement.

Let

$$A \cong \prod_{i=1}^{q} M_{d_i}(D_i)$$

be semisimple Artinian and let

$$M \cong \bigoplus_{i=1}^{q} S_i^{\oplus m_i}$$

be a finitely generated left module over A. Then

$$\boxed{\operatorname{srank}_A(M) = \max_{1 \le i \le q} \left\lceil \frac{m_i}{d_i} \right\rceil .}$$

If A is any algebra of finite dimension over a field, the same formula applied to the decomposition of $M/\operatorname{Jac}(A)M$ over $A/\operatorname{Jac}(A)$ computes $\operatorname{srank}_A(M)$.

Proof. By PSC-I-C-MINIMAL-FREE-COVER, M is generated by r elements exactly when it is a quotient of $A^{\oplus r}$. By PSC-I-L-REGULAR-MODULE-MULTIPLICITIES,

$$A^{\oplus r} \cong \bigoplus_{i} S_i^{\oplus r d_i}.$$

Every short exact sequence of modules over a semisimple ring splits. Therefore a quotient isomorphic to M exists if and only if

$$m_i \le r d_i \qquad \text{for every } i.$$

The smallest such r is the displayed maximum. For a nonsemisimple algebra of finite dimension, PSC-I-T-MODULE-TOP reduces the calculation to the semisimple module $M/\operatorname{Jac}(A)M$ over $A/\operatorname{Jac}(A)$. □

Formal correspondence. covered. Actual product matrix ring with left multiplication and natural column modules acted on through the matching factor. Semilinear eM along eR enforces compatibility. DivisionRows proves left generation iff row spanning and m<=r*d; standard rows prove sufficiency. Product assembly, column rearrangement and seed-rank invariance prove the formula. RadicalTop is literally M/(Jac(R) smul top); existence constructs decompositions.

Source locator. editable/traditional/02_foundations.tex; statement lines 759–779; proof lines 784–795.

Statement SHA-256. c604c6473ceeb6b651f37301bd3e719d69a555f04c9e4aea5fa6ad49bcd35de4

39. Algebraically closed multiplicity formula

Stable ID. PSC-I-C-ALGEBRAICALLY-CLOSED-FORMULA

Kind. corollary

Statement.

Let $\Bbbk$ be algebraically closed, let A be a unital algebra of finite dimension over $\Bbbk$, put $J = \mathrm{Jac}(A)$, and let M be a finitely generated left module over A. Assume compatible decompositions indexed by an integer $q \geq 0$:

$$A/J \cong \prod_{i=1}^{q} M_{d_i}(\Bbbk), \qquad M/JM \cong \bigoplus_{i=1}^{q} (\Bbbk^{d_i})^{\oplus m_i},$$

where the d_i are positive integers and the m_i are nonnegative integers. The first isomorphism is an isomorphism of algebras over $\Bbbk$. The second is an isomorphism of left modules over A/J, using the natural column module for each matrix factor and zero action from the other factors. Then

$$\mathrm{srank}_A(M) = \max_{1 \leq i \leq q} \left\lceil \frac{m_i}{d_i} \right\rceil,$$

with $\max \varnothing = 0$ when $q = 0$.

Proof. By PSC-I-T-MODULE-TOP, $\mathrm{srank}_A(M) = \mathrm{srank}_{A/J}(M/JM)$. Apply PSC-I-T-SEMISIMPLE-SEED-RANK to the displayed decompositions of A/J and M/JM, with every $D_i = \Bbbk$. This gives the stated maximum. If $q = 0$, the unital A/J-module M/JM is zero, and PSC-I-T-MODULE-TOP gives seed rank zero. □

Formal correspondence. covered. Correct public map uses decomposition of A/J, not A. The supplied algebra equivalence forgets to a ring equivalence, and the module decomposition stays compatible. Specializing the exact radical-top theorem to all D_i=k gives the current statement.

Source locator. editable/traditional/02_foundations.tex; statement lines 809–822; proof lines 826–828.

Statement SHA-256. 20c07c2db5ab4c4a956355fe5e45008edff100a4b279dd27a37ddff90871fcbf

40. Full matrix algebra

Stable ID. PSC-I-C-FULL-MATRIX

Kind. corollary

Statement.

For the natural module $S = D^d$ over $M_d(D)$,

$$\mathrm{srank}_{M_d(D)}(S) = 1.$$

More strongly, every nonzero vector of S is cyclic.

Proof. The formula gives $\lceil 1/d \rceil = 1$. Directly, let $v \neq 0$ and choose a coordinate $v_j \neq 0$. For any target $w = (w_i)$, the matrix whose jth column has entries $w_i v_j^{-1}$ and whose other columns vanish sends v to w. □

Formal correspondence. covered. For every division ring and positive matrix size, the natural module has canonical seed rank one and every nonzero vector generates. The proof uses entries w_i / v_j with the inverse on the correct right side. The positive-size convention of the surrounding natural-simple-module discussion is essential: size zero has seed rank zero. This guard remains explicit in Lean and is tested, not silently discarded.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 830–837; proof lines 841–843.

Statement SHA-256. `f22f88ae7b9fe79b50f6c82420d301d784c941d6340fc700723841d724826055`

41. Multiplicity obstruction

Stable ID. `PSC-I-C-MULTIPLICITY-OBSTRUCTION`

Kind. corollary

Statement.

For the diagonal action of $M_d(D)$ on $S^{\oplus m}$,

$$\boxed{\operatorname{srank}_{M_d(D)}(S^{\oplus m}) = \left\lceil \frac{m}{d} \right\rceil .}$$

Proof. Apply `PSC-I-T-SEMISIMPLE-SEED-RANK` to the single simple type with multiplicity m. □

Formal correspondence. covered. For arbitrary division rings and positive block size, the actual shared matrix action on m natural-module copies has canonical rank ceil(m/d). The rectangular-matrix, natural-column-product and finite-direct-sum formulations are connected by actual module equivalences. Zero multiplicity is included. This is not the independent-factor action, whose bound is a maximum across distinct factors.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 845–855; proof lines 859–861.

Statement SHA-256. `df87d9465fd05e359ca1a96ba5b69e501f5e6b9b93b2d4e1b555a84476ec84ef`

42. Primary decomposition for polynomial modules

Stable ID. `PSC-I-CI-PID-PRIMARY-DECOMPOSITION`

Kind. classicalinput

Statement.

Every module X over $\mathbb{k}[t]$ with finite dimension over the coefficient field decomposes as

$$X = \bigoplus_{p \in \operatorname{Irr}(\mathbb{k}[t])} X_p,$$

where $\operatorname{Irr}(\mathbb{k}[t])$ denotes the monic irreducible polynomials, only finitely many components are nonzero, and every nonzero X_p is a finite direct sum of modules $\mathbb{k}[t]/(p^e)$.

Formal correspondence. classical input integrated. For an existing polynomial module, multiplication by X constructs its evaluation operator and an actual polynomial-linear equivalence; finite coefficient dimension gives torsion. PID decomposition constructs positive elementary divisors, while actual

intrinsic primary submodules are proved equal to ideal primary components. PolynomialMonic proves a bijection between monic irreducibles and nonzero prime ideals via p↦(p), deriving injectivity from monicity and surjectivity from a normalized PID prime generator. It reindexes the internal sum and finite nonzero support to precisely the manuscript index set. intrinsic_primary_finite_power_sum is a literal finite DirectSum of k[X]/(p^e), with positive e, for each specified irreducible p.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 883–890.

Statement SHA-256. `446dc677476b213cc4daac554de2b3f15ab816c9d0d6eeff98a67965eea843a9`

43. Single-operator seed rank

Stable ID. `PSC-I-T-SINGLE-OPERATOR`

Kind. theorem

Statement.

Let X have finite dimension over $\mathbb{k}$ and let $T \in \mathrm{End}_{\mathbb{k}}(X)$. Regard X as a module over $\mathbb{k}[t]$ by $t \cdot x = Tx$. For every monic irreducible p, let X_p be the primary component for p and set

$$\mu_p(T) := \dim_{\mathbb{k}[t]/(p)}(X_p / p(T)X_p).$$

Then

$$\boxed{\mathrm{srank}_{\mathbb{k}[T]}(X) = \max_p \mu_p(T).}$$

Moreover, $\mu_p(T)$ is the number of cyclic primary summands for p in the decomposition into elementary divisors of X_p.

Proof. Let the minimal polynomial be

$$m_T(t) = \prod_p p(t)^{e_p}.$$

The acting algebra is

$$\mathbb{k}[T] \cong \mathbb{k}[t]/(m_T) \cong \prod_p R_p, \qquad R_p := \mathbb{k}[t]/(p^{e_p}),$$

by the Chinese remainder theorem. The factor R_p acts on X_p and annihilates every other primary component. Hence the formula for a direct sum with independent actions gives

$$\mathrm{srank}_{\mathbb{k}[T]}(X) = \max_p \mathrm{srank}_{R_p}(X_p).$$

Each R_p is local with maximal ideal generated by the class of p and residue division field $K_p = \mathbb{k}[t]/(p)$. By `PSC-I-C-LOCAL-ALGEBRA-FORMULA`,

$$\mathrm{srank}_{R_p}(X_p) = \dim_{K_p} X_p / p(T)X_p = \mu_p(T).$$

Finally, every summand $\mathbb{k}[t]/(p^e)$ contributes one copy of K_p after quotienting by p, so the dimension equals the number of summands in the elementary divisor decomposition. □

Formal correspondence. covered. The full printed statement is now mapped. The actual power-annihilation submodule is proved equal to the ideal primary component. Its inclusion into V/pV

has exact kernel pX_p and is onto from the internally discharged Artinian hypothesis. The induced quotient equivalence is residue-field linear. Every intrinsic multiplicity is identified with every actual presentation fibre. The grouped factors are proved to be the literal X_p, and any positive primary decomposition of X_p has the same intrinsic count. The presentation-free maximum over all monic irreducibles is proved bounded and attained, and equals both natural-valued and cardinal-valued seed rank. No caller-supplied decomposition or seed-count conclusion is needed by the public maximum theorem. These are noncomputable mathematical constructions. The printed proof uses a different route through local-module/Nakayama machinery; that separately listed machinery is not thereby marked complete.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 893–907; proof lines 914–939.

Statement SHA-256. `e96051e5d8333ea1ba1efe94f70b260ad356a41bc5ecad49525ec1df9c759967`

44. Invariant-factor count

Stable ID. `PSC-I-C-INVARIANT-FACTOR-COUNT`

Kind. corollary

Statement.

If $X \neq 0$ and the rational canonical decomposition is written with nonconstant invariant factors

$$X \cong \bigoplus_{j=1}^{r} \mathbb{k}[t]/(f_j), \qquad f_1 \mid f_2 \mid \cdots \mid f_r,$$

then

$$\boxed{\operatorname{srank}_{\mathbb{k}[T]}(X) = r.}$$

Proof. For each irreducible p, the number of invariant factors divisible by p equals the number of elementary divisors for p. Because $f_1 \mid \cdots \mid f_r$ and $f_1 \neq 1$ after zero summands are removed, some irreducible divisor of f_1 divides all r invariant factors. Hence $\max_p \mu_p(T) = r$. □

Formal correspondence. covered. Complete at the displayed conditional scope: an actual polynomial-module equivalence to the finite product of nonconstant factors and their divisibility chain imply least seed number, natural seed rank and cardinal seed rank equal the factor count. The theorem also handles the empty product. Finite products and finite direct sums represent the same finite decomposition. Existence of an ordered invariant-factor presentation is not asserted by this declaration, and is not an additional hypothesis-free claim of the printed conditional corollary.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 951–963; proof lines 967–969.

Statement SHA-256. `c26c3a48531f1cf7f79090c82c8db185af548fc1137f1c427ac815d1d8120e89`

45. Geometric-multiplicity formula

Stable ID. `PSC-I-C-ALGEBRAICALLY-CLOSED-SINGLE-OPERATOR`

Kind. corollary

Statement.

If $\mathbb{k}$ is algebraically closed, then

$$\boxed{\operatorname{srank}_{\mathbb{k}[T]}(X) = \max_{\lambda \in \operatorname{Spec}(T)} \dim_{\mathbb{k}} \operatorname{Ker}(T - \lambda I).}$$

Proof. The irreducibles are $p = t - \lambda$. On the primary component for λ, the quotient by $(T - \lambda I)X_\lambda$ has dimension equal to the number of Jordan blocks, which is also $\dim \operatorname{Ker}(T - \lambda I)$. □

Formal correspondence. covered. Complete at the finite-dimensional algebraically closed scope inherited from the single-operator theorem. The actual intrinsic multiplicity for X-C(a) is identified with the dimension of the actual eigenspace, without diagonalizability. The canonical natural and cardinal seed ranks equal the maximum over the actual spectrum. Off-spectrum dimensions are zero, the empty-spectrum/zero-space maximum is defined as zero, and the spectral maximum is attained on nonzero spaces. No executable spectral algorithm is claimed.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 971–982; proof lines 986–988.

Statement SHA-256. `babe43b45ebd6abb23a0e31ed41c0918a4babb97578847c852f2c9d11e69515f`

46. Cyclic-matrix criterion

Stable ID. `PSC-I-C-CYCLIC-MATRIX`

Kind. corollary

Statement.

For $X \neq 0$ of finite dimension, the following are equivalent:

(i) T has a cyclic vector.

(ii) $\operatorname{srank}_{\mathbb{k}[T]}(X) = 1$.

(iii) Every nonzero primary component for p is cyclic.

(iv) The rational canonical form has one invariant factor.

(v) The minimal polynomial equals the characteristic polynomial.

(vi) After scalar extension to an algebraic closure, every eigenvalue has geometric multiplicity one.

Proof. The equivalence of (i) and (ii) is `PSC-I-P-CYCLIC-CHARACTERIZATION`. By `PSC-I-T-SINGLE-OPERATOR`, (ii) holds exactly when every $\mu_p(T)$ belonging to a nonzero primary component equals one, which is (iii). Relatively prime primary cyclic modules combine into one cyclic module by the Chinese remainder theorem, so (iii) is equivalent to (iv). In rational canonical form, one invariant factor means that the unique invariant factor is both the minimal and characteristic polynomial, proving (iv)⇔(v). Scalar extension preserves the equality of minimal and characteristic polynomials. Over an algebraic closure, that equality is equivalent to one Jordan block for each eigenvalue, which is (vi). □

Formal correspondence. covered. Block 06 closes the exact six-way criterion for a nonzero finite-dimensional operator over an arbitrary field. The final List.TFAE includes actual cyclicity, canonical seed rank one, cyclicity of every nonzero intrinsic primary component, an actual monic positive-degree one-invariant-factor presentation, minpoly=charpoly, and one-dimensional eigenspaces at every spectral value of the literal tensor extension to AlgebraicClosure k. Minimal-polynomial preservation and cyclicity preservation/descent are proved, not supplied as hypotheses. The one-factor presentation has a constructed basis in which the original operator is the standard companion matrix; its factor is proved to be the minimal polynomial. A separate theorem proves that any actual nonconstant invariant-factor chain has exactly one factor iff the operator is cyclic. This is a mathematical existence and equivalence theorem, not an executable normal-form algorithm or a full sorted invariant-factor construction for arbitrary noncyclic operators. The nonzero-space guard is retained in the six-way statement. Legacy conditional wrappers are preserved unchanged for compatibility, but the new public criterion needs no decomposition, multiplicity, polynomial-preservation, spectral, or seed-number certificates.

Source locator. `editable/traditional/02_foundations.tex`; statement lines 990–1001; proof lines 1005–1007.

Statement SHA-256. `d418194e254211c8f964adf5796418ad7aaff14e630caffc05cb8d935de9803a`

A.2 Packets and transfer graphs

47. Finite packet system

Stable ID. `PSC-I-D-PACKET-SYSTEM`

Kind. definition

Statement.

A finite packet system on a vector space X over $\mathbb{k}$ is a family

$$\mathcal{P} = (P_\alpha)_{\alpha \in I} \subseteq \mathrm{End}_{\mathbb{k}}(X)$$

satisfying

$$P_\alpha^2 = P_\alpha, \qquad P_\alpha P_\beta = 0 \quad (\alpha \neq \beta), \qquad \sum_{\alpha \in I} P_\alpha = I_X.$$

Set $X_\alpha = P_\alpha X$. Zero packet spaces may be removed without changing any identity. No orthogonality is assumed unless stated.

Formal correspondence. definition represented. LinearProjectorFamily encodes the actual finite completeness, idempotence and annihilation laws.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 10–25.

Statement SHA-256. `251dfa1ba22ba22eaf53fdcc7880aa1fe05933738f362ef3f6ae72583f539459`

48. Packet support and packet-pure vectors

Stable ID. `PSC-I-D-PACKET-SUPPORT`

Kind. definition

Statement.

For a finite packet system $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ and $x \in X$, define

$$\mathrm{Supp}_{\mathcal{P}}(x) := \{\alpha \in I : P_\alpha x \neq 0\}.$$

The vector x is *packet pure* if $|\mathrm{Supp}_{\mathcal{P}}(x)| \leq 1$. A finite family is packet pure if each of its members is packet pure, although different members may lie in different packets.

Formal correspondence. definition represented. packetSupport is exactly {i | P_i x≠0}. PacketPure is Set.Subsingleton of that support, equivalent for finite support to cardinality at most one. packetPure_iff_zero_or_range identifies purity with zero or membership in one actual packet range; a pure family is pointwise this predicate.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 28–37.

Statement SHA-256. `d95377c31eb89901afc468fd8f0ba293fee6865b0ae3b22eff99df183cc9eb9e`

49. Finite packet aggregation

Stable ID. `PSC-I-L-PACKET-AGGREGATION`

Kind. lemma

Statement.

Let $\sigma : I \twoheadrightarrow J$ be a surjection and define

$$Q_j := \sum_{\sigma(\alpha)=j} P_\alpha.$$

Then $(Q_j)_{j\in J}$ is a finite packet system and

$$Q_j X = \bigoplus_{\sigma(\alpha)=j} X_\alpha.$$

Proof. Pairwise annihilation of the P_α gives

$$Q_j^2 = Q_j, \qquad Q_j Q_k = 0 \quad (j \neq k).$$

Summing over j recovers $\sum_\alpha P_\alpha = I$. The range statement follows from the directness forced by pairwise annihilation. □

Formal correspondence. covered. Q_j is the actual filtered sum of P_i. Computing P_a Q_j proves the laws. Actual range analysis/synthesis over the literal fibre subtype are inverse, and the finite product/direct-sum equivalence gives the printed conclusion.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 45–55; proof lines 59–67.

Statement SHA-256. `504cbd4eec7a5b5e416f91dcfd13e06586e5c1360990ed480424f4a85dfb9195`

50. Packet analysis and synthesis

Stable ID. `PSC-I-D-ANALYSIS-SYNTHESIS`

Kind. definition

Statement.

For a finite packet system, define

$$\mathcal{A}_{\mathcal{P}} : X \longrightarrow \bigoplus_{\alpha\in I} X_\alpha, \qquad x \longmapsto (P_\alpha x)_\alpha,$$

and

$$\mathcal{S}_{\mathcal{P}} : \bigoplus_{\alpha\in I} X_\alpha \longrightarrow X, \qquad (x_\alpha)_\alpha \longmapsto \sum_\alpha x_\alpha.$$

Formal correspondence. definition represented. Analysis and synthesis use genuine projector ranges, not unconstrained packet slots.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 69–83.

Statement SHA-256. `f364fcc94b357d761a0bb880e5a2d5761fe7bfc687b768b0f4f7cc261a5ce820`

51. Exact packet-coordinate isomorphism

Stable ID. `PSC-I-T-EXACT-PACKET-COORDINATES`

Kind. theorem

Statement.

For every finite packet system,

$$\boxed{\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}} = I_X, \qquad \mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}} = I_{\oplus X_\alpha}.}$$

Hence

$$\boxed{X \cong \bigoplus_{\alpha \in I} X_\alpha.}$$

Proof. For $x \in X$,

$$\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}}x = \sum_\alpha P_\alpha x = x.$$

If $x_\alpha \in X_\alpha$, then $P_\beta x_\alpha = \delta_{\alpha\beta} x_\alpha$, so

$$\mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}}(x_\alpha)_\alpha = \left(P_\beta \sum_\alpha x_\alpha\right)_\beta = (x_\beta)_\beta.$$

Thus analysis and synthesis are inverse linear maps. □

Formal correspondence. covered. Both inverse identities and the linear equivalence are proved.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 86–100; proof lines 105–121.

Statement SHA-256. `d50988e4b39d3f37422e1d0c2704aca3b8acd89beca5afd9d07ad6bc326fc24f`

52. Packetwise equality

Stable ID. `PSC-I-C-PACKETWISE-EQUALITY`

Kind. corollary

Statement.

For $x, y \in X$,

$$x = y \quad \Longleftrightarrow \quad P_\alpha x = P_\alpha y \text{ for every } \alpha.$$

Proof. The forward implication is immediate. For the reverse implication, sum the packet equalities and use $\sum_\alpha P_\alpha = I$. □

Formal correspondence. covered. Exact equality iff every projected coordinate agrees.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 128–136; proof lines 140–142.

Statement SHA-256. `61919a93863e4ac0a41653a394e44cd8f701eac05335283ac8cf73983da43859`

53. Universal mapping property of finite packet coordinates

Stable ID. `PSC-I-C-PACKET-UNIVERSAL-PROPERTY`

Kind. corollary

Statement.

Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ be a finite packet system on X, and let Y be a vector space over $\mathbb{k}$.

1. Every family of linear maps $f_\alpha : X_\alpha \to Y$ determines a unique linear map $f : X \to Y$ with $f|_{X_\alpha} = f_\alpha$, namely
$$f = \sum_\alpha f_\alpha P_\alpha.$$
2. Every family of linear maps $g_\alpha : Y \to X_\alpha$ determines a unique linear map $g : Y \to X$ with $P_\alpha g = g_\alpha$, namely
$$g = \sum_\alpha g_\alpha.$$

Equivalently,

$$\mathrm{Hom}(X, Y) \cong \prod_\alpha \mathrm{Hom}(X_\alpha, Y), \qquad \mathrm{Hom}(Y, X) \cong \prod_\alpha \mathrm{Hom}(Y, X_\alpha),$$

where the products are finite.

Proof. For $x = \sum_\alpha P_\alpha x$, any extension of the f_α must satisfy

$$f(x) = \sum_\alpha f_\alpha(P_\alpha x),$$

which proves both existence and uniqueness of the first formula. For the second, the sum $g = \sum_\alpha g_\alpha$ satisfies

$$P_\beta g = \sum_\alpha P_\beta g_\alpha = g_\beta$$

because $g_\alpha(Y) \subseteq X_\alpha$. Uniqueness follows from $\sum_\alpha P_\alpha = I$. □

Formal correspondence. covered. Outgoing extension is the finite sum of f_i after actual rangeRestrict projections. Incoming assembly is actual rangeSynthesis after LinearMap.pi. Packet laws give restrictions/projections and completeness gives uniqueness. These universal properties induce the displayed Hom equivalences.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 146–166; proof lines 171–183.

Statement SHA-256. `3136b91ef330400f72205441080d679179476d4620287e1bd3658a36dca7a4be`

54. Tensor-product packet systems

Stable ID. `PSC-I-T-TENSOR-PACKET-SYSTEM`

Kind. theorem

Statement.

Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ be a finite packet system on X, and let $\mathcal{Q} = (Q_\gamma)_{\gamma \in J}$ be a finite packet system on Y. Then

$$\mathcal{P} \boxtimes \mathcal{Q} := (P_\alpha \otimes Q_\gamma)_{(\alpha,\gamma) \in I \times J}$$

is a finite packet system on the algebraic tensor product $X \otimes_{\mathbb{k}} Y$, with packet spaces

$$(P_\alpha \otimes Q_\gamma)(X \otimes Y) = X_\alpha \otimes Y_\gamma.$$

Hence

$$\boxed{X \otimes Y \cong \bigoplus_{(\alpha,\gamma) \in I \times J} X_\alpha \otimes Y_\gamma.}$$

Proof. For every pair of packet indices,

$$(P_\alpha \otimes Q_\gamma)^2 = P_\alpha^2 \otimes Q_\gamma^2 = P_\alpha \otimes Q_\gamma.$$

If $(\alpha, \gamma) \neq (\beta, \delta)$, then either $P_\alpha P_\beta = 0$ or $Q_\gamma Q_\delta = 0$, so the corresponding tensor idempotents annihilate each other. Finally,

$$\sum_{\alpha,\gamma} P_\alpha \otimes Q_\gamma = \left(\sum_\alpha P_\alpha \right) \otimes \left(\sum_\gamma Q_\gamma \right) = I_X \otimes I_Y.$$

The range is generated by pure tensors $P_\alpha x \otimes Q_\gamma y$, hence equals $X_\alpha \otimes Y_\gamma$. Exact reconstruction follows from `PSC-I-T-EXACT-PACKET-COORDINATES`. □

Formal correspondence. covered. Actual TensorProduct.map projectors satisfy all laws by tensor induction. Tensor products of range inclusions have the tensor products of projections as left inverses; range_map identifies the actual packet range. This proves the canonical embedded interpretation of the printed range equality. Exact coordinates then give the literal DirectSum.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 188–210; proof lines 215–232.

Statement SHA-256. `afea7128594f24b1249382b028d823051a35d9bd3a3473c247a7b9bd63c5ba47`

55. Dual packet systems

Stable ID. `PSC-I-P-DUAL-PACKET-SYSTEM`

Kind. proposition

Statement.

Let X be finite dimensional and let $\mathcal{P} = (P_\alpha)$ be a finite packet system. The algebraic dual maps

$$P_\alpha^* : X^* \to X^*, \qquad P_\alpha^*(\varphi) = \varphi \circ P_\alpha,$$

form a finite packet system on X^*. Its packet with index α is canonically isomorphic to X_α^*, and

$$X^* \cong \bigoplus_\alpha X_\alpha^*.$$

Proof. Because dualization reverses products,

$$P_\alpha^* P_\beta^* = (P_\beta P_\alpha)^* = \delta_{\alpha\beta} P_\alpha^*.$$

Also $\sum_\alpha P_\alpha^* = I_{X^*}$. Thus the dual maps form a packet system. Restriction gives a map from $P_\alpha^* X^*$ to X_α^*. It is injective because a covector in the range of P_α^* vanishes on every other packet, and it is surjective because any covector on X_α extends to X through the direct sum decomposition. □

Formal correspondence. covered. Actual dualMap means precomposition. Composition reverses indices and orthogonality uses the reversed inequality correctly. Restriction and precomposition with the actual packet projection are inverse, giving the stated packet duals and literal finite DirectSum.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 234–246; proof lines 251–258.

Statement SHA-256. `ade34fd8142ece280ab2ae85913a943cb28a3661978659fde4a644eb5018d119`

56. Finite Banach packet isomorphism

Stable ID. `PSC-I-T-BANACH-PACKET-ISOMORPHISM`

Kind. theorem

Statement.

Let X be Banach and let $(P_\alpha)_{\alpha \in I}$ be a finite family of bounded pairwise annihilating idempotents with sum I_X. Then every range $X_\alpha = P_\alpha X$ is closed. For any finite product norm equivalent to the maximum norm on $\bigoplus_\alpha X_\alpha$, packet analysis and synthesis are bounded inverse operators.

Proof. The range of a bounded idempotent is its space of fixed points,

$$P_\alpha X = \operatorname{Ker}(I - P_\alpha),$$

which is closed. Because I is finite, analysis and synthesis are bounded for the maximum and sum norms, respectively, and therefore for every equivalent finite product norm. Their inverse identities are algebraic and were proved in `PSC-I-T-EXACT-PACKET-COORDINATES`. □

Formal correspondence. covered. The actual finite projectors are continuous and explicitly bounded as IsBoundedLinearMap over an arbitrary NormedField, including a trivial valuation. Individual bounds

produce one finite uniform bound; analysis has that bound and synthesis has the finite-cardinality bound. Any coordinate norm equivalent to the canonical maximum norm is represented by a linear identification and its literal two-sided norm inequalities. Composition gives bounded actual analysis and synthesis, both inverse identities, and closed packet ranges. No conclusion-shaped hypotheses occur.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 267–270; proof lines 274–280.

Statement SHA-256. `3fd1f470c1a5dcd6273739c6f6cb95e5fbb8f8c95e4fc4c76e010430a72afa3e`

57. Packet-coordinate condition number

Stable ID. `PSC-I-D-PACKET-CONDITION-NUMBER`

Kind. definition

Statement.

Fix a finite product norm on $\bigoplus_\alpha X_\alpha$ equivalent to the maximum norm, as in `PSC-I-T-BANACH-PACKET-ISOMORPHISM`. Define

$$\kappa(\mathcal{P}) := \|\mathcal{A}_\mathcal{P}\|\,\|\mathcal{S}_\mathcal{P}\|.$$

Formal correspondence. definition represented. The definition is the product of the ordinary operator norms of the actual bounded analysis and synthesis. The generic bounded additive homomorphism norm is defined as the infimum of nonnegative valid operator bounds; linearity is retained by ofBoundedLinear. Equivalent-norm coordinates have explicit actual-map equalities.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 282–290.

Statement SHA-256. `7c67c989c536ab3eecad8fdf8c19b0ed6eef7c164c569d69a0e04de101b2460d`

58. Conditioned coordinate transport

Stable ID. `PSC-I-P-CONDITIONED-COORDINATES`

Kind. proposition

Statement.

For a finite Banach packet system with $X \neq 0$,

$$\kappa(\mathcal{P}) \geq 1.$$

If $T \in \mathcal{B}(X)$ and

$$[T]_\mathcal{P} := \mathcal{A}_\mathcal{P} T \mathcal{S}_\mathcal{P},$$

then

$$\boxed{\|[T]_\mathcal{P}\| \leq \kappa(\mathcal{P})\|T\|, \qquad \|T\| \leq \kappa(\mathcal{P})\|[T]_\mathcal{P}\|.}$$

Proof. Since $I = \mathcal{S}_\mathcal{P}\mathcal{A}_\mathcal{P}$,

$$1 = \|I\| \leq \|\mathcal{S}_\mathcal{P}\|\|\mathcal{A}_\mathcal{P}\|.$$

The first displayed estimate follows from submultiplicativity. The inverse identity

$$T = \mathcal{S}_\mathcal{P}[T]_\mathcal{P}\mathcal{A}_\mathcal{P}$$

gives the second estimate. □

Formal correspondence. covered. For nonzero X the identity has norm one. The actual inverse relation S∘A=id and operator norm submultiplicativity give $\kappa \geq 1$. The coordinate operator is literally A∘T∘S, and S∘[T]∘A=T gives the reverse estimate. The final wrapper constructs both bounded actual coordinate maps for every equivalent finite-product norm and proves both inequalities for every bounded linear T over any NormedField.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 293–313; proof lines 317–327.

Statement SHA-256. `6799946dd2c88a65a635b418c7d0a74e94a9af4d1dc88df66cdd3b05c4ebc180`

59. Completion preserves a finite packet split

Stable ID. `PSC-I-C-COMPLETION-PACKET-SPLIT`

Kind. corollary

Statement.

Let M be a dense normed subspace of a Banach space X. Suppose every bounded P_α leaves M invariant and the restricted projectors form a packet system on M. Then

$$P_\alpha X = \overline{P_\alpha M}$$

and the packet decomposition of M extends uniquely to the completion X.

Proof. For $x \in X$, choose $m_n \in M$ with $m_n \to x$. Continuity gives $P_\alpha m_n \to P_\alpha x$, so $P_\alpha X \subseteq \overline{P_\alpha M}$. The reverse inclusion follows because $P_\alpha X$ is closed and contains $P_\alpha M$. The bounded extensions remain pairwise annihilating idempotents whose finite sum is the identity. □

Formal correspondence. covered. Over an arbitrary NormedField, the actual dense-submodule restrictions determine the ambient continuous projector identities by equality on a dense set. Each ambient range equals closure of the corresponding dense-submodule image, and two continuous ambient projector extensions agreeing on the dense submodule coincide. This is stronger than the bounded-input manuscript statement; boundedness of the given ambient projectors is available from the printed premise and is not needed for the density/uniqueness argument.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 334–341; proof lines 345–347.

Statement SHA-256. `67202410fc4580e865ea492dba6cf8dff7e0f296d89ac886c96188a0169a6b70`

60. Unconditional packet sum space

Stable ID. `PSC-I-D-UNCONDITIONAL-PACKET-SPACE`

Kind. definition

Statement.

Let $(X_n)_{n\ge 1}$ be closed subspaces of a real or complex Banach space X. Define

$$\bigoplus_{n\ge 1}^{u} X_n$$

to be the space of sequences $x = (x_n)$ with $x_n \in X_n$ for which $\sum_n x_n$ converges unconditionally in X. Equip it with

$$\|x\|_u := \sup_{F\in\mathbb{N}} \sup_{|a_n|\le 1} \left\| \sum_{n\in F} a_n x_n \right\|.$$

Formal correspondence. definition represented. The carrier is exactly sequences x_n in the specified closed subspaces for which Summable x holds; mathlib Summable is the finite-subset net (unconditional summation). The norm is exactly sSup of norms of finite sums with coefficients in the closed scalar unit ball. Type synonyms avoid inherited product topology. Finiteness is derived, not assumed. RCLike covers real and complex scalars as stated.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 352–366.

Statement SHA-256. `bcbeba58302101658c66251d7b338a6a3932fb07b059e77931ffd063b5303b4f`

61. Completeness of the unconditional packet sum

Stable ID. `PSC-I-L-UNCONDITIONAL-COMPLETENESS`

Kind. lemma

Statement.

The normed space $\bigoplus_{n\ge 1}^{u} X_n$ in `PSC-I-D-UNCONDITIONAL-PACKET-SPACE` is Banach.

Proof. Let $x^{(k)} = (x_n^{(k)})$ be Cauchy in $\|\cdot\|_u$. For every fixed n,

$$\|x_n^{(k)} - x_n^{(\ell)}\| \le \|x^{(k)} - x^{(\ell)}\|_u,$$

so $x_n^{(k)}$ converges to some $x_n \in X_n$ because X_n is closed. Given $\varepsilon > 0$, choose k_0 such that

$$\|x^{(k)} - x^{(\ell)}\|_u < \varepsilon \qquad (k, \ell \ge k_0).$$

For finite F and $|a_n| \le 1$, pass $\ell \to \infty$ in the finite sum to obtain

$$\left\| \sum_{n\in F} a_n (x_n^{(k)} - x_n) \right\| \le \varepsilon \qquad (k \ge k_0).$$

Hence $\|x^{(k)} - x\|_u \le \varepsilon$ once x is known to be unconditionally summable.

Choose such a $k \geq k_0$. Since $x^{(k)}$ is unconditionally summable, there is a finite F_0 such that every finite $F \subseteq \mathbb{N} \setminus F_0$ and every $|a_n| \leq 1$ satisfy

$$\left\| \sum_{n \in F} a_n x_n^{(k)} \right\| < \varepsilon.$$

The preceding uniform estimate then gives

$$\left\| \sum_{n \in F} a_n x_n \right\| < 2\varepsilon.$$

Thus the tails of $\sum_n x_n$ are uniformly small under bounded scalar changes, which is the unconditional Cauchy criterion. Therefore $x \in \bigoplus^u X_n$, and the earlier estimate proves $x^{(k)} \to x$ in $\| \cdot \|_u$. □

Formal correspondence. covered. Uniform boundedness on the complete continuous dual is justified by finite-dimensional real absolute summability of each scalar image; the isometric bidual inclusion yields finite multiplier bounds. The signed norm controls each coordinate; coordinate limits stay in closed subspaces. Each fixed finite multiplier estimate passes to the limit, uniformly in the finite set and coefficients. Unweighted tails prove Summable for the limit, and the same uniform estimate gives convergence in the signed norm. Metric completeness is then obtained without a projector hypothesis.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 369–372; proof lines 376–405.

Statement SHA-256. `339adef1c7737cb8adc8a4bc76b3a5a2f95a575ffb576a34404d8da60bfe0b62`

62. Countable unconditional packet isomorphism

Stable ID. `PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS`

Kind. theorem

Statement.

Let X be a real or complex Banach space and let $(P_n)_{n \geq 1} \subseteq \mathcal{B}(X)$ satisfy

$$P_n P_m = \delta_{nm} P_n$$

and

$$x = \sum_{n \geq 1} P_n x$$

unconditionally for every $x \in X$. Assume

$$K := \sup_{F \Subset \mathbb{N}} \sup_{|a_n| \leq 1} \left\| \sum_{n \in F} a_n P_n \right\| < \infty.$$

Then analysis and synthesis define inverse Banach space isomorphisms

$$\boxed{X \cong \bigoplus_{n \geq 1}^{u} P_n X.}$$

More precisely,

$$\|\mathcal{A}_{\mathcal{P}}\| \le K, \qquad \|\mathcal{S}_{\mathcal{P}}\| \le 1.$$

Proof. Each $P_nX = \operatorname{Ker}(I - P_n)$ is closed. For $x \in X$,

$$\|\mathcal{A}_{\mathcal{P}}x\|_u = \sup_{F,a} \left\| \sum_{n\in F} a_n P_n x \right\| \le K\|x\|.$$

For $y = (y_n) \in \bigoplus^u P_nX$, define

$$\mathcal{S}_{\mathcal{P}}y = \sum_n y_n.$$

Since the finite partial sums with every coefficient equal to one are among those used in $\|y\|_u$,

$$\|\mathcal{S}_{\mathcal{P}}y\| \le \|y\|_u.$$

The assumed expansion gives $\mathcal{S}_{\mathcal{P}}\mathcal{A}_{\mathcal{P}}x = x$. Conversely, boundedness of P_m permits termwise application to the unconditionally convergent series:

$$P_m\mathcal{S}_{\mathcal{P}}y = \sum_n P_m y_n = y_m.$$

Thus $\mathcal{A}_{\mathcal{P}}\mathcal{S}_{\mathcal{P}} = I$. Completeness of the coordinate space is PSC-I-L-UNCONDITIONAL-COMPLETENESS. □

Formal correspondence. covered. ofSupBound constructs K as the literal supremum of finite multiplier operator norms from BddAbove. The carriers use Summable sequences in actual projector ranges. Continuous projectors commute with the sum, so annihilation proves both inverse identities. The literal signed norm yields analysis bound K and synthesis bound 1. A proved linear isometry identifies the projector-specific carrier and norm with the general ClosedPackets.Space. Completeness follows either from the general theorem or from the constructed uniform equivalence with Banach X.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 407–436; proof lines 441–467.

Statement SHA-256. `7579699c5a34468a6f9df136e7c897641c5409e5f5e597cd849348bddc92a469`

63. Uniform boundedness

Stable ID. PSC-I-R-UNIFORM-BOUNDEDNESS

Kind. remark

Statement.

If every expansion $\sum_n P_n x$ is unconditional, the operators obtained from finite partial sums with signs form a pointwise bounded family. The uniform boundedness principle therefore yields a finite constant K. We retain this constant explicitly in PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS because later estimates use it to bound the coordinate map.

Formal correspondence. documentary. A uniform-boundedness explanation. No instance for the signed packet partial-sum family is mapped.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 469–472.

Statement SHA-256. `2ae822d9b50496c6462b94007b5ff1a8d083e7109e6da7d0c2453ca40b80373c`

64. Infinite packet sums

Stable ID. `PSC-I-B-INFINITE-PACKET-TOPOLOGY`

Kind. boundary

Statement.

For an infinite family, an identity such as $\sum_\alpha P_\alpha = I$ requires a declared convergence mode and the corresponding uniform boundedness statement. Pointwise, strong, norm, weak, and unconditional convergence are not interchangeable.

Formal correspondence. scope boundary. Convergence modes remain separate; finite identities are not used as infinite sums.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 475–478.

Statement SHA-256. `305cbc8493dffcc2407698c4df7a3f54fecc76bb57e10c75390b0cedc5bce53d`

65. Transfer block

Stable ID. `PSC-I-D-TRANSFER-BLOCK`

Kind. definition

Statement.

For $T \in \mathrm{End}_{\Bbbk}(X)$ and a finite packet system $\mathcal{P}$, define

$$T_{\beta\alpha} := P_\beta T P_\alpha \in \mathrm{Hom}_{\Bbbk}(X_\alpha, X_\beta).$$

Formal correspondence. definition represented. Ambient compositions and their genuinely range-restricted packet Hom maps are both encoded.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 492–499.

Statement SHA-256. `1b27b671e8c43628db2f3e157df05e01a63658c32e64a6ae2a94fdc69fd5ad73`

66. Exact operator reconstruction

Stable ID. `PSC-I-T-OPERATOR-RECONSTRUCTION`

Kind. theorem

Statement.

Every $T \in \mathrm{End}_{\Bbbk}(X)$ satisfies

$$\boxed{T = \sum_{\alpha,\beta \in I} T_{\beta\alpha}.}$$

Proof. Insert the finite packet identity on both sides:

$$T = \left(\sum_\beta P_\beta\right) T \left(\sum_\alpha P_\alpha\right) = \sum_{\alpha,\beta} P_\beta T P_\alpha.$$

□

Formal correspondence. covered. The existing result is more general: source and target can have different finite packet systems.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 502–508; proof lines 512–522.

Statement SHA-256. `f8c54ea5f0aa3075a42556f7d9b9b29fd9399d88e9de974c6eb098fceb2a7abb`

67. Packet matrix multiplication

Stable ID. `PSC-I-T-BLOCK-MULTIPLICATION`

Kind. theorem

Statement.

For $S, T \in \mathrm{End}_{\Bbbk}(X)$,

$$\boxed{(ST)_{\gamma\alpha} = \sum_{\beta \in I} S_{\gamma\beta} T_{\beta\alpha}.}$$

Proof. Insert $I = \sum_{\beta} P_{\beta}$ between S and T:

$$P_{\gamma} S T P_{\alpha} = \sum_{\beta} P_{\gamma} S P_{\beta} T P_{\alpha}.$$

□

Formal correspondence. covered. The output-first block product is proved for three possibly different modules.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 529–539; proof lines 544–552.

Statement SHA-256. `a7aaf1cde8454ca239c380ca964f5b13f20d39baaf62b7a859caf9a849378314`

68. Packet powers and paths

Stable ID. `PSC-I-C-BLOCK-POWERS`

Kind. corollary

Statement.

For $m \geq 1$,

$$(T^{m})_{\beta_m \beta_0} = \sum_{\beta_1, \ldots, \beta_{m-1}} T_{\beta_m \beta_{m-1}} \cdots T_{\beta_1 \beta_0}.$$

The same formula holds for a product of nonidentical operators with the corresponding blocks in temporal order.

Proof. Induct on m using `PSC-I-T-BLOCK-MULTIPLICATION`. For a product of different operators, use the same induction with the specified operator at each step. □

Formal correspondence. covered. The power theorem expands transferBlock(P,P,T^m) using packetPathSum. The independent operatorWord/packetWordSum induction expands every finite list of

possibly different operators, with composition order matching rightmost-acts-first temporal order. Each summation is over actual intermediate packet labels.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 559–571; proof lines 575–577.

Statement SHA-256. `dbaa3f1f55517bd6366bcf856dc8eb6836a642144d805f9bed4899e5bba7038f`

69. Adjoint reversal

Stable ID. `PSC-I-C-ADJOINT-REVERSAL`

Kind. corollary

Statement.

If X is Hilbert, every P_α is orthogonal, and $T \in \mathcal{B}(X)$, then

$$\boxed{(T_{\beta\alpha})^* = (T^*)_{\alpha\beta}.}$$

Proof. Because $P_\alpha^* = P_\alpha$,

$$(P_\beta T P_\alpha)^* = P_\alpha T^* P_\beta.$$

□

Formal correspondence. covered. The result is proved for bounded self-adjoint outer maps, more generally than orthogonal packet projectors.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 579–585; proof lines 589–594.

Statement SHA-256. `8e978a295a29a3eac3d1c4bec1a8e3f4e05a2684fdbce8e80a357aa7891e8f79`

70. Tensor-product transfer blocks

Stable ID. `PSC-I-C-TENSOR-TRANSFER-BLOCKS`

Kind. corollary

Statement.

Let $S \in \mathrm{End}(X)$ and $T \in \mathrm{End}(Y)$, with packet systems $\mathcal{P}$ and $\mathcal{Q}$. Relative to the tensor product packet system of `PSC-I-T-TENSOR-PACKET-SYSTEM`,

$$\boxed{(S \otimes T)_{(\beta,\delta),(\alpha,\gamma)} = S_{\beta\alpha} \otimes T_{\delta\gamma}.}$$

Proof. By definition,

$$\begin{aligned}(S \otimes T)_{(\beta,\delta),(\alpha,\gamma)} &= (P_\beta \otimes Q_\delta)(S \otimes T)(P_\alpha \otimes Q_\gamma) \\ &= (P_\beta S P_\alpha) \otimes (Q_\delta T Q_\gamma) \\ &= S_{\beta\alpha} \otimes T_{\delta\gamma}.\end{aligned}$$

□

Formal correspondence. covered. Actual transfer blocks factor by tensor-map composition in exactly the (target,source) order printed. Both sides are actual endomorphisms of X tensor Y; no factorization witness is an input.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 598–608; proof lines 613–621.

Statement SHA-256. `ec0bc2d2c85cdf55ea2f27003457a653f8f2426fe97dafd6c918cbda54874878`

71. Blockwise intertwining equations

Stable ID. `PSC-I-T-BLOCK-INTERTWINING`

Kind. theorem

Statement.

Let $T \in \mathrm{End}(X)$, $S \in \mathrm{End}(Y)$, let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ and $\mathcal{Q} = (Q_\gamma)_{\gamma \in J}$ be finite packet systems, and let $U : X \to Y$ be linear. Define

$$T_{\beta\alpha} = P_\beta T P_\alpha, \qquad S_{\gamma\delta} = Q_\gamma S Q_\delta, \qquad U_{\gamma\alpha} = Q_\gamma U P_\alpha.$$

Then

$$\boxed{UT = SU}$$

if and only if, for every $\gamma \in J$ and $\alpha \in I$,

$$\boxed{\sum_{\beta \in I} U_{\gamma\beta} T_{\beta\alpha} = \sum_{\delta \in J} S_{\gamma\delta} U_{\delta\alpha}.}$$

Proof. Apply Q_γ on the left and P_α on the right. Inserting $I_X = \sum_\beta P_\beta$ gives

$$Q_\gamma U T P_\alpha = \sum_\beta Q_\gamma U P_\beta T P_\alpha = \sum_\beta U_{\gamma\beta} T_{\beta\alpha}.$$

Inserting $I_Y = \sum_\delta Q_\delta$ gives

$$Q_\gamma S U P_\alpha = \sum_\delta Q_\gamma S Q_\delta U P_\alpha = \sum_\delta S_{\gamma\delta} U_{\delta\alpha}.$$

Thus $UT = SU$ implies every block equation. Conversely, if every block equation holds, then all blocks of $UT - SU$ vanish. Exact operator reconstruction yields $UT - SU = 0$. □

Formal correspondence. covered. U is any linear map, not an intertwiner structure. Forward direction inserts middle projector sums; reverse direction reconstructs UT and SU from all equal blocks. Distinct source/target index sets are retained.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 623–645; proof lines 650–668.

Statement SHA-256. `558adbc4eead21c3f371918f40b9f462000c7b26a7234864ee44c103e09cf1e6`

72. Full block-algebra representation

Stable ID. PSC-I-T-BLOCK-ALGEBRA

Kind. theorem

Statement.

The map

$$\Theta_{\mathcal{P}} : \mathrm{End}_{\Bbbk}(X) \longrightarrow \prod_{\beta,\alpha\in I} \mathrm{Hom}_{\Bbbk}(X_\alpha, X_\beta), \qquad T \longmapsto (T_{\beta\alpha})_{\beta,\alpha},$$

is an algebra isomorphism when the codomain is equipped with matrix multiplication whose entries are operators.

Proof. Injectivity follows from exact reconstruction. Given arbitrary linear maps $B_{\beta\alpha} : X_\alpha \to X_\beta$, define

$$T\left(\sum_\alpha x_\alpha\right) := \sum_{\alpha,\beta} B_{\beta\alpha} x_\alpha.$$

The packet decomposition is unique and the index set is finite, so this defines a linear operator whose blocks are the prescribed maps. Thus $\Theta_{\mathcal{P}}$ is surjective. Compatibility with multiplication is PSC-I-T-BLOCK-MULTIPLICATION. □

Formal correspondence. covered. Full dependent packet Hom families and their multiplication are identified with the actual endomorphism algebra.

Source locator. editable/traditional/03_packets_and_graphs.tex; statement lines 675–686; proof lines 691–699.

Statement SHA-256. a66e4216289ceb5e48d1ccd41d269b656c6c93ed462a47f68cbd0172c9360bda

73. Propagation of packet support

Stable ID. PSC-I-C-PACKET-SUPPORT-PROPAGATION

Kind. corollary

Statement.

Let $T \in \mathrm{End}(X)$ and $x \in X$. Then

$$\boxed{\mathrm{Supp}_{\mathcal{P}}(Tx) \subseteq \left\{\beta : \exists \alpha \in \mathrm{Supp}_{\mathcal{P}}(x) \text{ with } T_{\beta\alpha} \neq 0\right\}.}$$

Equality holds whenever, for every target β in the set on the right, the active contributions

$$(T_{\beta\alpha} P_\alpha x : \alpha \in \mathrm{Supp}_{\mathcal{P}}(x),\ T_{\beta\alpha} \neq 0)$$

have nonzero sum. In particular, equality holds if at most one active source packet feeds each such β and that source coordinate does not lie in the kernel of the corresponding block.

Proof. For every target packet β,

$$P_\beta T x = \sum_{\alpha\in\mathrm{Supp}_{\mathcal{P}}(x)} T_{\beta\alpha} P_\alpha x.$$

If every block in this sum is zero, then the coordinate with index β vanishes. This proves the inclusion. Equality is exactly the assertion that the displayed active sum is nonzero for every target in the set on the right. The final sufficient condition leaves at most one nonzero summand at each target, and kernel avoidance makes that summand nonzero. □

Formal correspondence. covered. The source expands each target coordinate as the finite sum of the actual transfer-block contributions and filters to active sources. This yields the support inclusion; nonzero active sums give equality. The single-active theorem uses activeSources={a} plus nonzero transfer contribution, which is equivalent to at-most-one active source plus kernel avoidance on a target that is active.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 703–720; proof lines 725–733.

Statement SHA-256. `f05b8bf4828b88f6020e2e0b824e228acffcb77d7a514ced31ec90c64e6d4437`

74. Dimension count

Stable ID. `PSC-I-C-DIMENSION-COUNT`

Kind. corollary

Statement.

If X is finite dimensional and $d_\alpha = \dim X_\alpha$, then

$$\dim \mathrm{End}_{\Bbbk}(X) = \sum_{\alpha,\beta} d_\alpha d_\beta = \left(\sum_{\alpha} d_\alpha \right)^2 .$$

Proof. Take dimensions in the block algebra isomorphism and use $\dim \mathrm{Hom}(X_\alpha, X_\beta) = d_\alpha d_\beta$. □

Formal correspondence. covered. The actual End space is linearly equivalent to the heterogeneous packet Hom product. Its finite dimension equals the sum of products of actual packet-range dimensions, and finite distributivity gives the displayed square. Both printed equalities occur in one conjunction.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 736–746; proof lines 750–752.

Statement SHA-256. `85dd27ca5c20eeb56683b4b3d9b9093503e2c3c2fdc4a07c1aaf33eefab855c8`

75. Block-diagonal projection and packet commutant

Stable ID. `PSC-I-P-BLOCK-DIAGONAL-COMMUTANT`

Kind. proposition

Statement.

Define

$$\Delta_{\mathcal{P}}(T) := \sum_{\alpha} P_\alpha T P_\alpha .$$

Then $\Delta_{\mathcal{P}}$ is a linear idempotent on $\mathrm{End}(X)$ whose range is the block diagonal subalgebra. For $T \in \mathrm{End}(X)$, the following are equivalent:

(i) $TP_\alpha = P_\alpha T$ for every α.

(ii) $T_{\beta\alpha} = 0$ whenever $\beta \neq \alpha$.

(iii) $\Delta_{\mathcal{P}}(T) = T$.

Proof. Pairwise annihilation gives $\Delta_{\mathcal{P}}^2 = \Delta_{\mathcal{P}}$ and identifies its range with diagonal block families. If T commutes with all P_α, then for $\beta \neq \alpha$,

$$P_\beta T P_\alpha = P_\beta P_\alpha T = 0.$$

Thus (i) implies (ii). Exact reconstruction shows (ii) implies (iii). If $T = \sum_\alpha P_\alpha T P_\alpha$, then

$$T P_\gamma = P_\gamma T P_\gamma = P_\gamma T,$$

so (iii) implies (i). □

Formal correspondence. covered. diagonalProjection is the actual K-linear map T↦Σ_i P_i T P_i. The proofs derive idempotence, equivalence of commutation with every projector and off-diagonal block vanishing, equivalence to fixed points of this projection, and equality of its actual linear range to the equality-supported subalgebra.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 754–766; proof lines 771–781.

Statement SHA-256. `613238e060365a7c00244f44abb552ada9f54933d8483382f259d3f32843eae5`

76. Support relation and support space

Stable ID. `PSC-I-D-SUPPORT-RELATION`

Kind. definition

Statement.

For a relation $R \subseteq I \times I$, with (β, α) interpreted as an allowed transfer $\alpha \to \beta$, define

$$\mathfrak{M}_R := \{T \in \mathrm{End}(X) : T_{\beta\alpha} = 0 \text{ whenever } (\beta, \alpha) \notin R\}.$$

Formal correspondence. definition represented. OperatorSupported is initially defined via actual restricted packet maps; operatorSupported_iff_ambient proves its exact equivalence to ambient transfer-block vanishing. supportSubmodule is the corresponding actual linear subspace of End(X). Relation dictionary: formal rel(a,b) means manuscript (b,a)∈R, with a source and b target.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 789–797.

Statement SHA-256. `babc2ddcee7669992338d1b7741bd50fab01175a280772ab40df14dcc814ebf3`

77. Support-pattern algebra criterion

Stable ID. `PSC-I-T-SUPPORT-PATTERN-ALGEBRA`

Kind. theorem

Statement.

Assume $X_\alpha \neq 0$ for every $\alpha \in I$. Then:

(i) $I_X \in \mathfrak{M}_R$ if and only if R is reflexive.

(ii) $\mathfrak{M}_R\mathfrak{M}_R \subseteq \mathfrak{M}_R$ if and only if R is transitive.

(iii) $\mathfrak{M}_R$ is a unital subalgebra if and only if R is a preorder.

Proof. The identity has diagonal blocks I_{X_α} and no blocks outside the diagonal, so it belongs to $\mathfrak{M}_R$ exactly when $(\alpha, \alpha) \in R$ for every α.

Assume R is transitive. If a summand

$$S_{\gamma\beta}T_{\beta\alpha}$$

in the (γ, α) block of ST is nonzero, then (γ, β) and (β, α) lie in R. Transitivity gives $(\gamma, \alpha) \in R$, so every forbidden block of ST vanishes.

Conversely, suppose transitivity fails. Choose α, β, γ with

$$(\beta, \alpha) \in R, \qquad (\gamma, \beta) \in R, \qquad (\gamma, \alpha) \notin R.$$

Choose nonzero vectors $u \in X_\alpha$, $v \in X_\beta$, and $w \in X_\gamma$. Choose linear functionals $f \in X_\alpha^*$ and $g \in X_\beta^*$ with $f(u) = g(v) = 1$. Let T have the sole block $x \mapsto f(x)v$ from X_α to X_β, and let S have the sole block $y \mapsto g(y)w$ from X_β to X_γ. Then $S, T \in \mathfrak{M}_R$, but ST has a nonzero forbidden (γ, α) block. This proves necessity. Combining reflexivity and transitivity gives (iii). □

Formal correspondence. covered. The two difficult converses use arbitrary maps between the actual nonzero packet ranges. Identity support is equivalent to reflexivity; multiplication closure is equivalent to transitivity. support_mul_closed_iff transfers the block statement to actual ambient operators, and support_pattern_algebra_criterion characterizes existence of a Subalgebra whose carrier is precisely that support space.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 800–808; proof lines 812–830.

Statement SHA-256. `fcd8af3b6718c7460c4e58639b8e942c24ede23b8673f9811f193cab4ccd3b20`

78. $*$-closed bounded support patterns

Stable ID. `PSC-I-C-SUPPORT-PATTERN-STAR`

Kind. corollary

Statement.

Let X be Hilbert, let the packet projectors be orthogonal, assume $X_\alpha \neq 0$ for every $\alpha \in I$, and put

$$\mathfrak{M}_R^{\mathrm{b}} := \mathfrak{M}_R \cap \mathcal{B}(X).$$

Then $\mathfrak{M}_R^{\mathrm{b}}$ is closed under adjoints if and only if R is symmetric. Hence $\mathfrak{M}_R^{\mathrm{b}}$ is a unital $*$ subalgebra of $\mathcal{B}(X)$ if and only if R is an equivalence relation.

Proof. Adjoint reversal sends an allowed (β, α) block to an (α, β) block. Symmetry is therefore sufficient. If symmetry fails, choose a nonzero map of rank one in an allowed nonsymmetric block. Its adjoint occupies the forbidden reverse block, proving necessity. A reflexive, transitive, symmetric relation is an equivalence relation. □

Formal correspondence. covered. Supported uses literal continuous transfer blocks and is proved equivalent to the underlying algebraic support predicate. Converse symmetry and transitivity use actual bounded rank-one operators between nonzero ranges. Identity support forces reflexivity. Sufficiency constructs an actual StarSubalgebra. RCLike covers the real and complex Hilbert settings; no finite-dimensional restriction or blanket bounded-block assumption is used.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 832–839; proof lines 843–845.

Statement SHA-256. `89dfdc356ea3dbbf9ee8018c66d8307f445d19078e0d7226372aae44c1872333`

79. Preorder support algebra decomposition

Stable ID. `PSC-I-T-PREORDER-ALGEBRA-DECOMPOSITION`

Kind. theorem

Statement.

Let R be a preorder on the finite set I. Define

$$\alpha \sim \beta \quad \Longleftrightarrow \quad (\beta, \alpha) \in R \text{ and } (\alpha, \beta) \in R.$$

Let $\overline{I} = I/\!\sim$, which carries the induced partial order, and put

$$X_C := \bigoplus_{\alpha \in C} X_\alpha.$$

Let $\mathfrak{N}_R \subseteq \mathfrak{M}_R$ consist of operators whose blocks vanish whenever source and target belong to the same equivalence class. Then:

(i) $\mathfrak{N}_R$ is a nilpotent ideal on both sides.

(ii) If h is the maximal number of strict steps in a chain of $\overline{I}$, then

$$\mathfrak{N}_R^{h+1} = 0.$$

(iii) The map to the diagonal classes induces an algebra isomorphism

$$\boxed{\mathfrak{M}_R/\mathfrak{N}_R \cong \prod_{C\in\overline{I}} \mathrm{End}_{\Bbbk}(X_C).}$$

Proof. A nonzero block in $\mathfrak{N}_R$ moves from one equivalence class to a strictly larger class in the quotient partial order. The block multiplication law shows that products concatenate such moves. Therefore left or right multiplication by an element of $\mathfrak{M}_R$ preserves the property of having no blocks within a single class, so $\mathfrak{N}_R$ is an ideal on both sides. A product of $h+1$ elements of $\mathfrak{N}_R$ would require a strict chain of $h+1$ steps in $\overline{I}$, which does not exist. This proves nilpotence.

Let $P_C = \sum_{\alpha\in C} P_\alpha$. Define

$$\rho : \mathfrak{M}_R \longrightarrow \prod_C \mathrm{End}(X_C), \qquad \rho(T) = (P_C T P_C)_C.$$

If an intermediate class D contributes to $P_C S T P_C$, then the preorder contains transfers $C \to D$ and $D \to C$, forcing $C = D$ in the quotient poset. Hence

$$P_C S T P_C = (P_C S P_C)(P_C T P_C),$$

so ρ is an algebra homomorphism. Every collection of endomorphisms of the X_C defines an allowed operator that is block diagonal with respect to the classes, so ρ is surjective. Its kernel is exactly $\mathfrak{N}_R$. The first isomorphism theorem gives the quotient formula. □

Formal correspondence. covered. The support algebra is constructed from the actual preorder. strictSupportIdeal is an actual TwoSidedIdeal of it, with membership proved equivalent to vanishing within every mutual-order class. Nonzero product blocks force strict paths; finite strict-path height is bounded and attained on nonempty indices. strictSupportIdeal_pow_eq_bot uses Mathlib standard ideal powers, with the span-of-words surrogate bridged by strictIdeal_power_mem. Antisymmetrization supplies the actual quotient partial order; class_maximalStrictSteps proves its height equals the exponent parameter. Grouped class projectors are actual sums of original projectors. The diagonal-class homomorphism is surjective with kernel equal to the strict ideal, and preorderQuotientDirectSumEquiv is the actual quotient by that ideal to the product of End of the literal direct sums of class packet ranges.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 847–875; proof lines 880–894.

Statement SHA-256. `89f93d5bdea5558d5ca9fef72742db6339af268623a37bedb7baabb777e4452a`

80. Transfer graph of an algebra-generating family

Stable ID. `PSC-I-D-TRANSFER-GRAPH`

Kind. definition

Statement.

Let $\mathcal{P} = (P_\alpha)_{\alpha \in I}$ be a finite packet system on X with every $P_\alpha \in \mathcal{A}$. Let $\mathcal{G} \subseteq \mathcal{A}$ satisfy

$$\mathcal{A} = \mathrm{alg}_{\Bbbk}(\{P_\alpha : \alpha \in I\} \cup \mathcal{G}),$$

where $\mathrm{alg}_{\Bbbk}(E)$ denotes the unital subalgebra over $\Bbbk$ generated by E. The transfer graph $\Gamma(\mathcal{G}, \mathcal{P})$ has vertex set I and an arrow

$$\alpha \longrightarrow \beta$$

when

$$P_\beta G P_\alpha \neq 0$$

for at least one $G \in \mathcal{G}$. Reachability includes paths of length zero.

Formal correspondence. definition represented. packetGeneratedAlgebra is Algebra.adjoin K (range P.proj ∪ ops), exactly the declared algebra in the manuscript. TransferEdge is witnessed by an actual member of ops with nonzero ambient transfer block. ReachableFrom uses ReflTransGen and hence includes length zero.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 916–931.

Statement SHA-256. `f6ddc4edec01afcf027a48300c34e66e9d1209d4f56bce17e3b9d00f88cd62cf`

81. Reachability support of the generated algebra

Stable ID. `PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT`

Kind. theorem

Statement.

For every $A \in \mathcal{A}$,

$$P_\beta A P_\alpha = 0 \qquad \text{whenever } \beta \notin \mathrm{Reach}(\alpha).$$

Equivalently,

$$\mathcal{A} \subseteq \mathfrak{M}_{R_\Gamma}, \qquad R_\Gamma := \{(\beta, \alpha) : \beta \in \mathrm{Reach}(\alpha)\}.$$

Proof. It is enough to treat a word in the generators and packet projectors. Insert

$$I = \sum_{\gamma \in I} P_\gamma$$

between consecutive nonprojector generators. Every resulting (β, α) block is a finite sum of products

$$P_\beta G_m P_{\gamma_{m-1}} G_{m-1} \cdots P_{\gamma_1} G_1 P_\alpha.$$

A nonzero product requires every consecutive block to be nonzero, hence gives a directed path

$$\alpha \to \gamma_1 \to \cdots \to \gamma_{m-1} \to \beta.$$

Packet projectors only insert stationary steps. Thus a word has no block from α to an unreachable β. Linear combinations preserve the same zero pattern. □

Formal correspondence. covered. Each projector and declared generator lies in the actual support algebra of ReflTransGen(TransferEdge). Algebra.adjoin_le therefore proves containment of the entire generated algebra. generatedAlgebra_block_zero is the literal ambient P_b A P_a=0 consequence outside reachability.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 934–947; proof lines 951–967.

Statement SHA-256. `7496f96565d405f8fe6b80491655dffe3b2ddad07b104c1131704464fb8ad031`

82. Reachability obstruction for generated hulls

Stable ID. `PSC-I-P-REACHABILITY-OBSTRUCTION`

Kind. proposition

Statement.

Let $I_0 \subseteq I$ and let

$$S \subseteq \bigoplus_{\alpha \in I_0} X_\alpha.$$

Then

$$\boxed{\mathrm{Hull}_{\mathcal{A}}(S) \subseteq \bigoplus_{\beta \in \mathrm{Reach}(I_0)} X_\beta.}$$

In particular, no seed supported outside a source strongly connected component can generate a nonzero vector in that source component.

Proof. For $s \in S$, write $s = \sum_{\alpha \in I_0} P_\alpha s$. If β is unreachable from I_0, then for every $A \in \mathcal{A}$,

$$P_\beta A s = \sum_{\alpha \in I_0} P_\beta A P_\alpha s = 0$$

by PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT. The same holds for every finite linear combination of orbit vectors. □

Formal correspondence. covered. The actual reachable-support subspace is preserved by every element of the generated algebra. Span of actual algebra orbits yields the hull inclusion for arbitrary seed sets. The source SCC is the actual mutual-reachability quotient, with source status equivalent to no incoming condensation edge. generatedAlgebra_source_obstruction separately assembles the printed final clause: seeds outside that source cannot generate any nonzero vector supported in it. It obtains a nonzero coordinate by packet completeness and applies no_reach_into_source and the unreachable-coordinate obstruction.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 977–992; proof lines 996–1005.

Statement SHA-256. `a46f6c292b4825e63689b6302077e4db2b2a8fe38f9958fc9018a623e7b8a239`

83. Packet-pure seed rank

Stable ID. PSC-I-D-DIRECTED-SEED-RANK

Kind. definition

Statement.

A vector is packet pure when it is zero or lies in one packet space X_α, equivalently when its packet support has cardinality at most one. Define

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) := \min\{|S| : S \text{ is packet-pure and } \mathcal{A}S = X\}.$$

For $C \subseteq I$, put

$$P_C := \sum_{\alpha\in C} P_\alpha, \qquad X_C := P_C X, \qquad \mathcal{A}_C := P_C \mathcal{A} P_C,$$

and define the local packet pure rank

$$r_C := \operatorname{psrank}_{\mathcal{A}_C,\mathcal{P}|_C}(X_C).$$

Formal correspondence. definition represented. The cardinal infimum is over all actual pure generating subsets, its indexing type is nonempty, and its infimum is attained by well-ordering. Corner compression is e*T on range(e), hence literally e*T*e there. Closure and the identity restriction are proved. A local family retains outside indices as zero maps; local_proj_zero_iff identifies these precisely for nonzero original packets, and zero deletion preserves purity and rank. The corrected zero-or-range wording matches empty-index conventions.

Source locator. editable/traditional/03_packets_and_graphs.tex; statement lines 1011–1031.

Statement SHA-256. 2a2a8b2073162ddfce0cf7462765d5453176d369133cc86dc699c8d038e8fee8

84. Basic packet-pure rank bounds

Stable ID. PSC-I-P-PACKET-PURE-RANK-BOUNDS

Kind. proposition

Statement.

For every A module resolved into packets with $P_\alpha \in A$,

$$\boxed{\operatorname{srank}_A(X) \le \operatorname{psrank}_{A,\mathcal{P}}(X) \le \sum_{\alpha\in I} \operatorname{srank}_{P_\alpha A P_\alpha}(X_\alpha).}$$

The corner algebra on the right is unital on X_α with identity P_α.

Proof. Every packet pure generating family is an unrestricted generating family, which gives the first inequality. For each α, choose a minimal generating family $S_\alpha \subseteq X_\alpha$ for the corner action $P_\alpha A P_\alpha$. The union $S = \bigcup_\alpha S_\alpha$ is packet pure. Since every $x \in X$ has the unique decomposition $x = \sum_\alpha x_\alpha$ and every x_α lies in $(P_\alpha A P_\alpha)S_\alpha \subseteq AS$, the union generates X. Its cardinality is at most the displayed sum. □

Formal correspondence. covered. The left inequality is direct inclusion of admissible generating sets. The upper inequality chooses attained minimal generating sets under each genuine corner algebra, embeds them into X, and proves their union generates by projector completeness and cor-

ner_action_ambient. The union cardinal is bounded by the cardinal sum. This works for arbitrary infinite dimensions and arbitrary cardinal ranks, including zero packets.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1034–1048; proof lines 1052–1054.

Statement SHA-256. `11a3ad129e84e51edb000d4027f785874b59f3867639a7813ab9c8d944bf81c0`

85. Edge-surjective packet system

Stable ID. `PSC-I-D-EDGE-SURJECTIVE`

Kind. definition

Statement.

The module resolved into packets is surjective along edges relative to $\Gamma(\mathcal{G}, \mathcal{P})$ when, for every graph edge $\alpha \to \beta$, there exists $B \in \mathcal{A}$ such that the pure block

$$P_\beta B P_\alpha : X_\alpha \twoheadrightarrow X_\beta$$

is surjective.

Formal correspondence. definition represented. The edge is TransferEdge P ops, not a supplied surrogate graph. The witness T belongs to Algebra.adjoin of the projectors and declared operators; restrictBlocks is the literal map from the source range to the target range.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1056–1063.

Statement SHA-256. `46dd6f245aadc93dd24ac5505cd37893d5474fb31208bab5babfacf0fa6d1ce4`

86. Source-component formula for packet-pure seed rank

Stable ID. `PSC-I-T-SOURCE-COMPONENT-SEED-RANK`

Kind. theorem

Statement.

Assume I is finite, every packet space is nonzero, and the module resolved into packets is surjective along edges. Let $\mathfrak{C}_0$ be the set of source strongly connected components of the condensation graph of $\Gamma(\mathcal{G}, \mathcal{P})$. Then

$$\boxed{\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = \sum_{C \in \mathfrak{C}_0} r_C.}$$

Proof. Let S be a packet pure generating family for X. Discard zero seeds, which do not change the generated hull, and fix a source component C. Seeds outside C cannot reach C by `PSC-I-P-REACHABILITY-OBSTRUCTION`. Moreover, for $s \in X_C$ and $A \in \mathcal{A}$,

$$P_C A s = (P_C A P_C) s \in \mathcal{A}_C s.$$

Thus the seeds of S lying in X_C must generate X_C under $\mathcal{A}_C$, and there are at least r_C of them. Summing over disjoint source components gives

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) \geq \sum_{C \in \mathfrak{C}_0} r_C.$$

For the reverse inequality, choose for each source component C a packet pure family S_C of size r_C generating X_C under $\mathcal{A}_C$, and put $S = \bigcup_C S_C$. Then every source component lies in $\mathcal{A}S$. Traverse the finite condensation graph in a topological order. Suppose all components preceding a component D that is not a source have already been shown to lie in $\mathcal{A}S$. Choose an incoming condensation edge from such a component to D, represented by a graph edge $\alpha \to \beta$ with α already generated and $\beta \in D$. Surjectivity along edges gives a pure block

$$B_{\beta\alpha} : X_\alpha \twoheadrightarrow X_\beta.$$

Because $X_\alpha \subseteq \mathcal{A}S$ and $\mathcal{A}S$ is invariant under $\mathcal{A}$, surjectivity implies $X_\beta \subseteq \mathcal{A}S$. For any other vertex $\gamma \in D$, choose a directed path inside the strongly connected component from β to γ. Successive surjective edge blocks propagate the full packet space along the path, so $X_\gamma \subseteq \mathcal{A}S$. Hence $X_D \subseteq \mathcal{A}S$. Induction over the condensation graph gives $X = \mathcal{A}S$, proving the upper bound. □

Formal correspondence. covered. SCC is the quotient by mutual reflexive-transitive reachability. Source is proved equivalent to absence of incoming condensation edges. Finite minimality proves every vertex is reached from a source. Generated-algebra support proves source projection compression. The lower bound removes zero seeds, proves those actually lying in each source generate its corner module, and injects the dependent union of these nonzero seed sets into the original generating set. The upper bound chooses attained local minima; actual surjective blocks propagate range containment along actual paths. No source coverage, compression identity, generating conclusion, or graph certificate is assumed in the final theorem. The theorem in fact does not require nonzero original packets, so it includes the printed hypotheses.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1066–1076; proof lines 1081–1098.

Statement SHA-256. `0be97dd456719faad7e8e6ba4091ee877f58e7c7c900c8ae79a7b8385c19daea`

87. Packet-propagating system

Stable ID. `PSC-I-D-PACKET-PROPAGATING`

Kind. definition

Statement.

A packet system that is surjective along edges is called packet propagating when every packet space X_α is cyclic under its diagonal corner algebra

$$P_\alpha \mathcal{A} P_\alpha.$$

Formal correspondence. definition represented. Cyclicity is the existing module Cyclic predicate for the genuine packetCornerAlgebra acting on its actual range. There are no extra local-rank-one certificates supplied.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1105–1111.

Statement SHA-256. `4e9bfc875f14eeec60ebfab1d4906a00230861dfe0e37ea46c90639653fbc111`

88. Unweighted source-component count

Stable ID. `PSC-I-C-SOURCE-COMPONENT-COUNT`

Kind. corollary

Statement.

Assume the module resolved into packets is packet propagating and $X_\alpha \neq 0$ for every $\alpha \in I$. If c_0 denotes the number of source strongly connected components, then

$$\boxed{\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = c_0.}$$

Proof. Fix a source component C and a vertex $\alpha \in C$. Choose a cyclic vector $s \in X_\alpha$ for $P_\alpha \mathcal{A} P_\alpha$. It generates all of X_α. Strong connectivity and surjectivity along edges then propagate the whole packet space to every vertex of C. Thus $r_C = 1$ for every source component. Apply `PSC-I-T-SOURCE-COMPONENT-SEED-RANK`. □

Formal correspondence. covered. The upper bound chooses a cyclic vector at one actual representative vertex of each source SCC and propagates full packet ranges to every vertex. The lower bound proves every generating set has a nonzero source projection, then uses purity and orthogonality to inject Sources into the chosen pure generating set. Nonzero packet range hypotheses are explicit. The finite cardinal-number form is proved; no finite generation or dimension assumption is introduced.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1114–1122; proof lines 1126–1128.

Statement SHA-256. `f7d07d0e6a080b4b92c8a401d9663246fb377ecc6d930b8ecdb94ec07ea7e5dc`

89. Mixed-support seeds can merge source requirements

Stable ID. `PSC-I-B-PACKET-PURE-BOUNDARY`

Kind. boundary

Statement.

The source component formulas concern packet pure seeds. They do not give the unrestricted seed rank. For example, let

$$\mathcal{A} = \mathbb{k} \times \mathbb{k}, \qquad X = \mathbb{k} \oplus \mathbb{k}$$

with the coordinate action and the two coordinate packet projectors. The transfer graph has two source components and

$$\operatorname{psrank}_{\mathcal{A},\mathcal{P}}(X) = 2,$$

but the mixed vector $(1, 1)$ is cyclic, so

$$\operatorname{srank}_{\mathcal{A}}(X) = 1.$$

Formal correspondence. scope boundary. The unrestricted mixed-seed example is consistent with the proved product-algebra formula. The packet-pure rank calculation is not separately formalized.

Source locator. `editable/traditional/03_packets_and_graphs.tex`; statement lines 1130–1146.

Statement SHA-256. `a3610fb8adcedb4fb91006a77f357f3f18c57678d044c21b8522730a456b0006`

A.3 Transport and realization

90. Coordinate transport

Stable ID. `PSC-I-T-COORDINATE-TRANSPORT`

Kind. theorem

Statement.

Let $U : X \to Y$ be a vector space isomorphism and let $\mathcal{P} = (P_\alpha)$ be a packet system on X. Define

$$Q_\alpha := UP_\alpha U^{-1}.$$

Then $\mathcal{Q} = (Q_\alpha)$ is a packet system on Y. For $\widetilde{T} = UTU^{-1}$,

$$\boxed{Q_\beta \widetilde{T} Q_\alpha = U(P_\beta T P_\alpha)U^{-1}.}$$

Hence packet dimensions, zero patterns, and support preorders are invariant under coordinate transport. If a declared algebra generating family $\mathcal{G}$ is transported to $U\mathcal{G}U^{-1}$, then its transfer graph and reachability relation are also preserved.

Proof. Conjugation preserves products and sums, so

$$Q_\alpha Q_\beta = UP_\alpha P_\beta U^{-1} = \delta_{\alpha\beta} Q_\alpha, \qquad \sum_\alpha Q_\alpha = I_Y.$$

The block identity is direct multiplication:

$$Q_\beta \widetilde{T} Q_\alpha = UP_\beta T P_\alpha U^{-1}.$$

All listed invariants are determined by these transported blocks. □

Formal correspondence. covered. conjugate constructs the actual family U P_i U^{-1} and proves all projector identities. conjugate_transferBlock proves the exact displayed equality. packetRangeEquiv restricts U to each actual packet range and yields cardinal dimension equality with explicit universe lifts. support_pattern_invariant proves preservation for every relation. TransferEdge and ReachableFrom are unchanged when the actual generator set is transported by conjugation.

Source locator. `editable/traditional/04_transport.tex`; statement lines 7–22; proof lines 26–41.

Statement SHA-256. `ae5e550ba65822235a1f8ab84c76a5a3ac876b28334523c9d845ece2af2d2d5a`

91. Packet-compatible operator morphism

Stable ID. `PSC-I-D-PACKET-MORPHISM`

Kind. definition

Statement.

Let $(X, \mathcal{A}, \mathcal{P})$ and $(Y, \mathcal{B}, \mathcal{Q})$ be packet resolved operator modules with packet index sets I and J. A packet compatible operator morphism is a triple

$$(\Phi, \varphi, \sigma)$$

consisting of a unital algebra homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$, a linear map $\Phi : X \to Y$ satisfying

$$\Phi(Ax) = \varphi(A)\Phi(x),$$

and a map $\sigma : I \to J$ satisfying

$$Q_j \Phi = \Phi P_{\sigma^{-1}(j)}, \qquad P_{\sigma^{-1}(j)} := \sum_{\sigma(\alpha)=j} P_\alpha.$$

An empty fiber is interpreted as the zero projector.

Formal correspondence. definition represented. The source and target have actual concrete unital operator subalgebras and nonzero complete orthogonal packet projectors. The linear map, unital algebra map and entirely arbitrary index map are separately retained. Fiber projectors are literal finite sums; no injectivity or surjectivity condition is imposed on the index map.

Source locator. `editable/traditional/04_transport.tex`; statement lines 48–68.

Statement SHA-256. `e0e522a401795ef9ecf2497c058baf522801bf33dd581a837deaabd9cbe887bf`

92. Morphism of seeded systems

Stable ID. `PSC-I-D-SEED-MORPHISM`

Kind. definition

Statement.

A morphism

$$(X, \mathcal{A}, S, \mathcal{P}) \longrightarrow (Y, \mathcal{B}, S', \mathcal{Q})$$

is a packet compatible operator morphism satisfying the seed condition

$$\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S').$$

In the topological category all maps and actions are required to be continuous and the seed condition uses the topological hull when declared.

Formal correspondence. definition represented. The algebraic condition uses the actual target algebra span, with arbitrary seed sets. The continuous variant separately retains Hausdorff topological vector-space structure, actual continuity of every algebra operator and packet projector, and actual continuity of the carrier morphism. Its seed condition is membership in the literal topologicalClosure of AlgebraGeneratedHull. The paper specifies an algebra of continuous operators, not a separate topology on that algebra, and the code states this convention explicitly.

Source locator. `editable/traditional/04_transport.tex`; statement lines 73–84.

Statement SHA-256. `88c7e9a865f68bac1df1727bd7cfc8c66e5b11d2e8be77ada048a83aaafd5637`

93. Naturality of generated hulls and transfer blocks

Stable ID. `PSC-I-T-NATURALITY`

Kind. theorem

Statement.

Let (Φ, φ, σ) be a packet compatible operator morphism. For every subset $S \subseteq X$,

$$\boxed{\Phi(\mathrm{Hull}_{\mathcal{A}}(S)) = \mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S)) \subseteq \mathrm{Hull}_{\mathcal{B}}(\Phi(S)).}$$

For $T \in \mathcal{A}$ and $i, j \in J$,

$$\boxed{Q_j \varphi(T) Q_i \Phi = \sum_{\substack{\sigma(\beta)=j \\ \sigma(\alpha)=i}} \Phi P_\beta T P_\alpha.}$$

If Φ is surjective and S generates X, then $\Phi(S)$ generates Y under $\varphi(\mathcal{A})$ and hence under $\mathcal{B}$. If, in addition, the morphism is seeded, then S' generates Y.

Proof. Every element of $\Phi(\mathrm{Hull}_{\mathcal{A}}(S))$ is a finite sum of vectors

$$\Phi(As) = \varphi(A)\Phi(s),$$

so it belongs to $\mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S))$. Conversely, every generator $\varphi(A)\Phi(s)$ equals $\Phi(As)$, giving equality.

For the block formula, use intertwining and packet compatibility twice:

$$\begin{aligned} Q_j \varphi(T) Q_i \Phi &= Q_j \varphi(T) \Phi P_{\sigma^{-1}(i)} \\ &= Q_j \Phi T P_{\sigma^{-1}(i)} \\ &= \Phi P_{\sigma^{-1}(j)} T P_{\sigma^{-1}(i)} \\ &= \sum_{\substack{\sigma(\beta)=j \\ \sigma(\alpha)=i}} \Phi P_\beta T P_\alpha. \end{aligned}$$

If Φ is surjective and $\mathrm{Hull}_{\mathcal{A}}(S) = X$, the hull equality gives $Y = \mathrm{Hull}_{\varphi(\mathcal{A})}(\Phi(S))$. If $\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S')$, then $\mathrm{Hull}_{\mathcal{B}}(\Phi(S)) \subseteq \mathrm{Hull}_{\mathcal{B}}(S')$, so S' generates Y. □

Formal correspondence. covered. Hull naturality is actual Submodule.map equality with the span of image-algebra orbits; target-algebra inclusion is derived. The block formula is the exact double sum over possibly many-to-one fibers, at the level of actual linear maps. Surjective generation is derived from the map of top, then the seeded condition implies generation by the target seeds.

Source locator. `editable/traditional/04_transport.tex`; statement lines 87–109; proof lines 114–131.

Statement SHA-256. `e7fb174db6031054bdf06996ebad915dfaf307ad38b27fceb17442c46128863b`

94. Category of seeded packet systems

Stable ID. PSC-I-P-SEED-SYSTEM-CATEGORY

Kind. proposition

Statement.

Packet compatible operator morphisms compose, seeded morphisms compose, and identity morphisms satisfy both definitions. Thus packet resolved operator modules form a category **PktMod** and seeded packet systems form a category **SeedSys**.

Proof. Let $(\Phi, \varphi, \sigma) : (X, \mathcal{A}, \mathcal{P}) \to (Y, \mathcal{B}, \mathcal{Q})$ and $(\Psi, \psi, \tau) : (Y, \mathcal{B}, \mathcal{Q}) \to (Z, \mathcal{C}, \mathcal{R})$ be morphisms. Intertwining is preserved because

$$\Psi\Phi(Ax) = \Psi(\varphi(A)\Phi x) = (\psi \circ \varphi)(A)\Psi\Phi x.$$

For k in the target packet set,

$$\begin{aligned} R_k \Psi\Phi &= \Psi Q_{\tau^{-1}(k)}\Phi \\ &= \Psi\Phi \sum_{\tau(\sigma(\alpha))=k} P_\alpha. \end{aligned}$$

Thus the composite packet map is $\tau \circ \sigma$. If the morphisms are seeded, then

$$\Phi(S) \subseteq \mathrm{Hull}_{\mathcal{B}}(S'), \qquad \Psi(S') \subseteq \mathrm{Hull}_{\mathcal{C}}(S'').$$

By PSC-I-T-NATURALITY,

$$\Psi\Phi(S) \subseteq \Psi(\mathrm{Hull}_{\mathcal{B}}(S')) \subseteq \mathrm{Hull}_{\mathcal{C}}(\Psi(S')) \subseteq \mathrm{Hull}_{\mathcal{C}}(S'').$$

Identity maps satisfy the definitions with identity algebra and packet maps. □

Formal correspondence. covered. Composition explicitly aggregates composite fibers by a finite-sum rearrangement and composes operator intertwiners. Algebraic seed closure is proved using span transport. The continuous variant proves invariance of the actual closed algebraic hull from individual operator continuity, maps the source algebraic hull into the target closed hull, then derives map_closedHull from carrier continuity and closure minimality. Seeded composition applies this proved closure transport; identity seed membership and all category laws are proved. The actual algebraic and topological Category instances therefore cover the full stated category claim.

Source locator. `editable/traditional/04_transport.tex`; statement lines 138–141; proof lines 147–180.

Statement SHA-256. `3cc5c23c5df9094cc9c18b7b9539bf0cadbbe2e74dd249651920fd5034f347a1`

95. Packet exact sequence for an invariant subspace

Stable ID. `PSC-I-T-PACKET-EXACT-SEQUENCE`

Kind. theorem

Statement.

Let $K \leq X$ satisfy $P_\alpha K \subseteq K$ for every α. Define

$$K_\alpha := P_\alpha K = K \cap X_\alpha$$

and

$$\overline{P}_\alpha(x + K) := P_\alpha x + K.$$

Then:

(i) The nonzero restrictions $P_\alpha|_K$ form a packet system on K.

(ii) The nonzero $\overline{P}_\alpha$ form a packet system on X/K.

(iii) For every α there is a short exact sequence

$$0 \longrightarrow K_\alpha \longrightarrow X_\alpha \longrightarrow \overline{P}_\alpha(X/K) \longrightarrow 0,$$

and

$$\overline{P}_\alpha(X/K) \cong X_\alpha/(X_\alpha \cap K) \cong (X_\alpha + K)/K.$$

(iv) The full exact sequence decomposes packetwise as the direct sum of these short exact sequences.

Proof. Restriction preserves idempotency, pairwise annihilation, and the identity sum on K. If $x \in K \cap X_\alpha$, then $x = P_\alpha x \in P_\alpha K$, proving $K_\alpha = K \cap X_\alpha$.

The quotient operator is well defined because replacing x by $x + k$ changes $P_\alpha x$ by $P_\alpha k \in K$. The idempotent and annihilation identities descend, and the sum is the identity on X/K. The map $X_\alpha \to X/K$ has image $(X_\alpha + K)/K$ and kernel $X_\alpha \cap K$, giving the short exact sequence and the quotient isomorphisms. Exact packet coordinates identify the original sequence with the direct sum over α. □

Formal correspondence. covered. K is an actual invariant submodule. Restriction/quotient projectors are constructed; restricted range equals K intersect X_i. Actual packet inclusion is injective, its range equals the actual quotient-map kernel, and quotient map is onto. First/second quotient isomorphisms use denominator comaps in the correct ambient type. Pointwise coordinate maps intertwine with original inclusion/quotient and form a short exact sequence; exact direct-sum coordinates give the full decomposition.

Source locator. `editable/traditional/04_transport.tex`; statement lines 185–216; proof lines 221–225.

Statement SHA-256. `7311937b07e98adf3e6f91ad665358380cab6ce26e3daeffdbf4787d7a01de56`

96. Packet descent through an invariant quotient

Stable ID. PSC-I-T-QUOTIENT-PACKET-DESCENT

Kind. theorem

Statement.

Under the hypotheses of PSC-I-T-PACKET-EXACT-SEQUENCE, the operators

$$\overline{P}_\alpha(x+K) = P_\alpha x + K$$

are well defined pairwise annihilating idempotents with sum $I_{X/K}$, and the quotient map satisfies

$$\boxed{\pi P_\alpha = \overline{P}_\alpha \pi.}$$

After zero packets are removed, they form the canonical quotient packet system.

Proof. All assertions are the quotient part of PSC-I-T-PACKET-EXACT-SEQUENCE. The intertwining identity follows directly from the definition:

$$\overline{P}_\alpha \pi(x) = P_\alpha x + K = \pi P_\alpha(x).$$

□

Formal correspondence. covered. Actual mapQ uses invariance of K. Quotient representative calculations prove the laws and commuting identity. Zero removal preserves completeness. Quotient surjectivity proves uniqueness of every descended operator, so canonicity is established.

Source locator. editable/traditional/04_transport.tex; statement lines 232–243; proof lines 247–252.

Statement SHA-256. b51391a00350d220fc92610f351826489ca69d166e27e266a1ecc6ecc7a2ffc7

97. Restriction and quotient of transfer blocks

Stable ID. PSC-I-C-QUOTIENT-PACKET-DESCENT

Kind. corollary

Statement.

Under the hypotheses of PSC-I-T-PACKET-EXACT-SEQUENCE, let $T \in \mathrm{End}(X)$ preserve K. Then every block $T_{\beta\alpha}$ maps K_α into K_β, and the induced quotient operator $\overline{T}$ satisfies

$$\boxed{\overline{T}_{\beta\alpha}\,\pi = \pi T_{\beta\alpha},}$$

where $\pi : X \twoheadrightarrow X/K$.

Proof. If $x \in K_\alpha$, then $Tx \in K$, so

$$T_{\beta\alpha}x = P_\beta T x \in K \cap X_\beta = K_\beta.$$

On the quotient,

$$\overline{T}_{\beta\alpha}\pi = \overline{P}_\beta \overline{T} \overline{P}_\alpha \pi = \pi P_\beta T P_\alpha.$$

□

Formal correspondence. covered. Invariance under P_i,T,P_j proves preservation of K and membership in the target packet. Tbar is actual K.mapQ K T hT, and representative calculation proves the quotient-block identity. Lean orientation is the symmetric equality of the printed formula.

Source locator. `editable/traditional/04_transport.tex`; statement lines 254–265; proof lines 269–282.

Statement SHA-256. `4fa0ae8effe1e5472dcde2be67829cba4ae586df9853d252baff055f3ee7fd46`

98. Topological packet exact sequence

Stable ID. `PSC-I-C-TOPOLOGICAL-PACKET-EXACT-SEQUENCE`

Kind. corollary

Statement.

If X is Banach, the P_α are bounded, and K is closed and packet invariant, then every packet sequence in `PSC-I-T-PACKET-EXACT-SEQUENCE` is a short exact sequence of Banach spaces with bounded maps, and the quotient projectors are bounded.

Proof. Closedness of K makes X/K Banach. The restricted projectors are bounded, and the quotient projectors are bounded because

$$\|\overline{P}_\alpha(x+K)\|_{X/K} \le \|P_\alpha\|\,\|x+K\|_{X/K}.$$

The packet spaces are closed ranges of bounded idempotents. □

Formal correspondence. covered. The generic theorem works over any NormedField, including a trivial valuation, with explicit boundedness of the original projectors. Restriction and quotient CLMs are the actual algebraic packet maps. Near-minimal quotient representatives transfer every original bound to the quotient; the norm bound uses the usual infimum-of-bounds norm of the bounded additive projector. Actual packet inclusion and quotient maps are bounded with constant one and retain injectivity, exactness, and surjectivity. Closed K and closed bounded-idempotent ranges make all three terms complete.

Source locator. `editable/traditional/04_transport.tex`; statement lines 284–287; proof lines 292–300.

Statement SHA-256. `2590b3fce057e3d054522a58a728f46dbf3c609c95ff6ab061ff654ffd72fee1`

99. Dense realization theorem

Stable ID. `PSC-I-T-DENSE-REALIZATION`

Kind. theorem

Statement.

Let X and Y be Hausdorff topological vector spaces, let $\Phi : X \to Y$ be continuous, surjective, and intertwining with respect to φ, and let S topologically generate X. Then $\Phi(S)$ topologically generates Y. If finite continuous packet systems satisfy

$$Q_j \Phi = \Phi P_{\sigma^{-1}(j)},$$

then

$$\boxed{Q_j Y = \Phi(P_{\sigma^{-1}(j)} X)}$$

for every j.

Proof. Given $y = \Phi(x)$, choose a net $x_\lambda \in \mathrm{Hull}_{\mathcal{A}}(S)$ converging to x. Then

$$\Phi(x_\lambda) \to y,$$

and naturality places every $\Phi(x_\lambda)$ in $\mathrm{Hull}_{\mathcal{B}}(\Phi(S))$. Hence the latter hull is dense. For packets, one inclusion follows from compatibility. Conversely, if $y = Q_j y$, choose x with $\Phi(x) = y$. Then

$$y = Q_j \Phi(x) = \Phi P_{\sigma^{-1}(j)} x.$$

□

Formal correspondence. covered. For the actual continuous linear intertwiner in the packet-morphism convention, the image of the source algebraic operator hull is contained in the target hull, and continuous surjectivity transports density. The actual finite grouped projector is the sum over the σ-fibre, including empty fibres. Surjectivity and the given intertwining identity prove exact equality of image and target range. No norm or metrizability assumptions occur.

Source locator. `editable/traditional/04_transport.tex`; statement lines 305–318; proof lines 322–331.

Statement SHA-256. `58bf2b9bfb4ab3ae9f3309b03cc231e0b59c81b7fe2b6c99e4b74884aaf15bca`

100. Quotient topology and completed realization

Stable ID. `PSC-I-T-QUOTIENT-TOPOLOGY-COMPLETION`

Kind. theorem

Statement.

Let $\Phi : X \twoheadrightarrow Y$ be a continuous linear quotient map between Hausdorff topological vector spaces, and put $K = \operatorname{Ker} \Phi$. Then the induced map

$$\widetilde{\Phi} : X/K \longrightarrow Y$$

is a topological isomorphism. If $D \subseteq X$ is a linear subspace whose image $\Phi(D)$ is dense in a complete space Y, then the completion of

$$D/(D \cap K)$$

in the topology transported from Y is canonically isomorphic to Y.

Proof. The algebraic first isomorphism theorem gives a bijection $\widetilde{\Phi}$. By definition of the quotient topology, a set $U \subseteq X/K$ is open exactly when its preimage in X is open. Because Φ is a quotient map, this is equivalent to $\widetilde{\Phi}(U)$ being open in Y. Thus $\widetilde{\Phi}$ is a homeomorphism. The second statement applies the dense completion argument of PSC-I-T-COMPLETED-PRESENTATION to the injective map $D/(D \cap K) \to Y$. □

Formal correspondence. covered. The actual first-isomorphism linear equivalence is continuous both ways using IsQuotientMap, not continuity plus surjectivity alone. The restriction to D uses its actual kernel, definitionally comap(D inclusion,ker Φ); density is exactly Φ(D). The same explicit transported-uniformity quotient completion gives the canonical target equivalence and agrees with Φ on represented classes. No original-domain quotient topology is silently substituted for the printed transported topology.

Source locator. `editable/traditional/04_transport.tex`; statement lines 333–344; proof lines 348–350.

Statement SHA-256. `2e986d76e93d67446708502bfae8275f63d8ab0eaf8dcc3e3630ee7f6b3fa077`

101. Continuous surjections and quotient topology

Stable ID. PSC-I-B-TOPOLOGICAL-QUOTIENT

Kind. boundary

Statement.

Continuity and surjectivity transport density as in PSC-I-T-DENSE-REALIZATION. They do not by themselves identify the given topology on Y with the quotient topology from X. The stronger conclusion in PSC-I-T-QUOTIENT-TOPOLOGY-COMPLETION requires the quotient map hypothesis or an equivalent open mapping condition.

Formal correspondence. scope boundary. The quotient-map requirement is preserved; no open-map theorem is inferred from continuity alone.

Source locator. `editable/traditional/04_transport.tex`; statement lines 352–355.

Statement SHA-256. `855989a0fadfaf7174d83889d9a2c4208bc722f866b35c5da99de3e7ecf1b9f7`

102. Realization functor and preservation profile

Stable ID. `PSC-I-D-REALIZATION-FUNCTOR`

Kind. definition

Statement.

Let $\mathbf{SeedSys}_{\mathrm{abs}}$ be an abstract category of seed systems and let $\mathbf{C}_D$ be a domain category. A realization functor is a functor

$$\mathsf{Ev}_D : \mathbf{SeedSys}_{\mathrm{abs}} \longrightarrow \mathbf{C}_D$$

with a declared preservation profile specifying which of the following structures are preserved:

$$0,\ I,\ \text{finite sums},\ \text{composition},\ \text{biproducts},$$
$$\text{kernels},\ \text{cokernels},\ \text{quotients},\ \text{tensor products},$$
$$\text{packet idempotents},\ \text{transfer blocks},\ \text{topological completions}.$$

Preservation of a structure means preservation of its defining identity or universal property, not merely a similarity of notation.

Formal correspondence. documentary. The profile is represented by actual additive/linear functor and PreservesLimit/PreservesColimit instances, with typed syntax for equational operations. Universal constructions also use explicit structured categories, an induced functor, an underlying comparison isomorphism, and actual initial/terminal preservation. No Boolean profile field is accepted in place of preservation evidence.

Source locator. `editable/traditional/04_transport.tex`; statement lines 382–397.

Statement SHA-256. `152d8ee6b5bb1735f7655c6eeee72210d213959c00c9e6168d554eee25b7c5fe`

103. Equational realization principle

Stable ID. `PSC-I-T-REALIZATION-EXPORT`

Kind. theorem

Statement.

Let $F : \mathbf{C} \to \mathbf{D}$ be an additive functor between preadditive categories. Any equality of morphisms constructed from declared generating morphisms by finite sums, additive inverses, identities, zero maps, and composition is preserved by F. Scalar multiplication by a coefficient ring is also preserved when the categories are enriched over that ring and F is linear for the declared enrichment. If a proof additionally uses a chosen biproduct, kernel, cokernel, quotient, tensor product, or completion and F preserves that construction in its declared profile, then the corresponding proof diagram is transported to $\mathbf{D}$.

Proof. Functoriality gives

$$F(I) = I, \qquad F(g \circ f) = F(g) \circ F(f),$$

and additivity gives

$$F(0) = 0, \qquad F(f+g) = F(f) + F(g), \qquad F(-f) = -F(f).$$

If the declared enrichment is present, linearity additionally gives $F(af) = aF(f)$ for every coefficient a. A structural induction on the construction of a morphism expression therefore carries every permitted

equational proof to its image under F. If a step invokes a universal construction, preservation means that the image diagram satisfies the same universal property. The uniquely induced morphism in the source proof is therefore carried to the uniquely induced morphism in the target proof. Repeating this argument through the finite proof diagram transports the conclusion. □

Formal correspondence. covered. Structural induction on well-typed morphism expressions proves transport of identities, zero, addition, negation, scalar multiplication and composition. The additive-only statement uses the canonical integer enrichment, and no extra scalar-linearity assumption is inserted. Actual limit/colimit diagrams are transported under the matching preservation instance. The conditional universal-construction clause is explicit: a declared profile supplies the structured categories of the universal problem, an induced functor, compatibility with the underlying realization functor, and actual initial/terminal preservation. CategoryUniversal derives the mapped initial/terminal property, equality of the uniquely induced arrows, and the underlying commuting comparison squares. Tensor and completion preservation are legitimate conditional inputs, not falsely inferred from additivity.

Source locator. `editable/traditional/04_transport.tex`; statement lines 400–403; proof lines 408–424.

Statement SHA-256. `e0352bdb7ba75e24d4c7e6e0259802f47800c4945c13efe4b3c68edcacef678d`

104. Realization export principle

Stable ID. `PSC-I-P-REALIZATION-EXPORT`

Kind. proposition

Statement.

Every canonical Paper I identity whose proof uses only the operations listed in a realization functor's verified preservation profile remains valid after applying that functor. If the proof uses a domain specific hypothesis not encoded by the functor, that hypothesis must be verified separately before the transported statement is promoted to a theorem.

Proof. The first assertion is `PSC-I-T-REALIZATION-EXPORT`. The second is a logical boundary: a functor transports established identities and preserved universal constructions, but it does not create an unverified analytic, topological, positivity, finiteness, or integrality hypothesis. □

Formal correspondence. covered. This is the methodological specialization of record 103 to already established Paper I identities. The formal equations are transported from a proved source equality, not assumed as the desired target equality. The universal-property step uses the explicit structured-category profile of CategoryUniversal. No new analytic/topological/positivity/finiteness/integrality assumption is generated.

Source locator. `editable/traditional/04_transport.tex`; statement lines 432–435; proof lines 439–441.

Statement SHA-256. `bf39b86447c8332d60216e48d23129b6ae5b096611e3febd9d09a699f1a9a557`

105. Semantic realization criterion

Stable ID. PSC-I-D-SEMANTIC-REALIZATION

Kind. definition

Statement.

A domain theorem is a semantic realization of a canonical seed calculus theorem only when:

(i) An explicit realization functor or intertwining construction is given.

(ii) The preservation profile covers every construction used in the canonical proof.

(iii) Every additional analytic, finiteness, convergence, positivity, or integrality hypothesis is verified in the domain.

(iv) The resulting conclusion is stated in the domain's native objects.

Reuse of the words "seed", "packet", or "transfer" without these data is only syntactic reuse.

Formal correspondence. documentary. A methodology and scope checklist, not automatic formal certification.

Source locator. editable/traditional/04_transport.tex; statement lines 443–453.

Statement SHA-256. 7bfb521bf39d1f519716ff37b9ffcf3ae76ce4f62fe121586daa29e66e7223df

106. Reserved example slots

Stable ID. PSC-I-ALG-EXAMPLE-SLOTS

Kind. definition

Statement.

The foundational calculus is illustrated by four canonical algebraic realizations:

(i) A typed quiver reachability model.

(ii) An algebraic coordinate ring presentation.

(iii) A companion and repeated primary single operator model.

(iv) Path and cycle adjacency models.

The worked model sections below instantiate these four interfaces. A bilateral shift example records the analytic completion mechanism also used in Paper V. Computational specifications connect the proofs with the verification companion and are not premises of the foundational theorems.

Formal correspondence. documentary. Example slots are editorial. Existing examples and algorithm specifications remain outside the 69 proved-record denominator.

Source locator. editable/traditional/04_transport.tex; statement lines 470–480.

Statement SHA-256. a870d22072364c0a743aede20db499ccda1ce4c62a009a82dbaefd76f525375d

107. Paper I export object

Stable ID. `PSC-I-EXPORT`

Kind. definition

Statement.

The canonical Paper I export is

$$\mathsf{PSC}_I = \left(\begin{array}{c} \text{typed grammars and realized operator algebras,} \\ \text{seed systems, free presentations, and seed rank,} \\ \text{module-top, semisimple, and single-operator formulas,} \\ \text{finite and unconditional packet coordinates,} \\ \text{transfer blocks and support algebras,} \\ \text{reachability and packet-pure source generation,} \\ \text{morphisms, invariant exact sequences, quotients, completions,} \\ \text{realization functors and equational theorem transport} \end{array} \right).$$

Formal correspondence. documentary. An export catalogue, not one additional kernel theorem or a completed formal package object.

Source locator. `editable/traditional/04_transport.tex`; statement lines 483–503.

Statement SHA-256. `5c58d6e198263b11ae3e1f98701de3cf1b70b5bc5da229d1ed705b607efe2aa7`

108. Exact limits of Paper I

Stable ID. `PSC-I-B-FINAL-CLAIM-BOUNDARY`

Kind. boundary

Statement.

The existence of an exact packet decomposition does not imply that the decomposition is computationally discoverable, numerically well conditioned, statistically identifiable, dynamically invariant, or mathematically useful. The condition number in `PSC-I-D-PACKET-CONDITION-NUMBER` quantifies only coordinate stability once the packets are supplied. Refinement towers, filtered transfer, nonlinear obstruction objects, automatic packet recovery, continuum limits, arithmetic lifts, and automorphic occupancy require additional theorems in later papers.

Formal correspondence. scope boundary. No discoverability, conditioning, identifiability or later-paper existence claim follows automatically from finite packet identities.

Source locator. `editable/traditional/04_transport.tex`; statement lines 532–535.

Statement SHA-256. `d059c43a34200d583173f81a335b2acb713d6ddbf0d46054763a575b1fdef0c7`

B Typed core signatures

These twenty signatures are reproduced from `kir/core_typed_signatures.json`. Unicode mathematical characters are represented with standard JSON escapes in the listings. Their meanings are unchanged. The schema does not make these strings into independently checked proof terms.

Linear seed system

PSC-I-D-SEED-SYSTEM

```
{
  "params": [
    "A:OperatorAlgebra",
    "M:Module(A)",
    "S:FiniteFamily(M)"
  ],
  "returns": "SeedSystem(A,M,S)"
}
```

Generated hull

PSC-I-D-GENERATED-HULL

```
{
  "params": [
    "A:OperatorAlgebra",
    "M:Module(A)",
    "S:Family(M)"
  ],
  "returns": "Submodule(M)"
}
```

Seed rank

PSC-I-D-SEED-RANK

```
{
  "params": [
    "A:OperatorAlgebra",
    "M:Module(A)"
  ],
  "returns": "Cardinal"
}
```

Finite packet system

PSC-I-D-PACKET-SYSTEM

```
{
  "params": [
    "K:Field",
    "X:VectorSpace(K)",
    "I:FiniteIndexSet",
    "P:I\u2192End(X)"
  ],
  "requires": [
    "idempotent(P)",
    "pairwise_annihilating(P)",
```

```
    "sum(P)=id_X"
  ],
  "returns": "PacketSystem(X,I)"
}
```

Packet analysis and synthesis

PSC-I-D-ANALYSIS-SYNTHESIS

```
{
  "params": [
    "P:PacketSystem(X,I)"
  ],
  "returns": [
    "Analyze_P:X\u2192DirectSum(I,PacketSpace(P,i))",
    "Synthesize_P:DirectSum(I,PacketSpace(P,i))\u2192X"
  ]
}
```

Exact packet-coordinate isomorphism

PSC-I-T-EXACT-PACKET-COORDINATES

```
{
  "params": [
    "P:PacketSystem(X,I)"
  ],
  "returns": "Equivalence(X,DirectSum(I,PacketSpace(P,i)))",
  "certificate": "exact_inverse"
}
```

Transfer block

PSC-I-D-TRANSFER-BLOCK

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "T:End(X)",
    "alpha:I",
    "beta:I"
  ],
  "returns": "LinearMap(PacketSpace(P,alpha),PacketSpace(P,beta))"
}
```

Exact operator reconstruction

PSC-I-T-OPERATOR-RECONSTRUCTION

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "T:End(X)"
  ],
  "returns": "Equality(T,Sum(alpha,beta,TransferBlock(P,T,beta,alpha)))"
}
```

Packet matrix multiplication

PSC-I-T-BLOCK-MULTIPLICATION

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "S:End(X)",
    "T:End(X)",
    "alpha:I",
    "gamma:I"
  ],
  "returns": "Equality(Block(S\u2218T,gamma,alpha),Sum(beta,Block(S,gamma,beta)\u2218Block(T,beta,alpha)))"
}
```

Packet powers and paths

PSC-I-C-BLOCK-POWERS

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "T:End(X)",
    "m:Nat"
  ],
  "returns": "PacketPathExpansion(T,m)"
}
```

Blockwise intertwining equations

PSC-I-T-BLOCK-INTERTWINING

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "Q:PacketSystem(Y,J)",
    "T:End(X)",
    "S:End(Y)",
    "U:LinearMap(X,Y)"
  ],
  "returns": "Iff(U\u2218T=S\u2218U,LocalBlockIntertwining(P,Q,T,S,U))"
}
```

Full block-algebra representation

PSC-I-T-BLOCK-ALGEBRA

```
{
  "params": [
    "P:PacketSystem(X,I)"
  ],
  "returns": "AlgebraEquiv(End(X),DependentBlockAlgebra(P))"
}
```

Support relation and support space

PSC-I-D-SUPPORT-RELATION

```
{
  "params": [
    "I:FiniteIndexSet",
    "R:Relation(I)"
  ],
  "returns": "SupportSpace(R)"
}
```

Coordinate transport

PSC-I-T-COORDINATE-TRANSPORT

```
{
  "params": [
    "U:LinearEquiv(X,Y)",
    "P:PacketSystem(X,I)",
    "T:End(X)"
  ],
  "returns": "TransportedPacketOperator(U,P,T)"
}
```

Packet exact sequence for an invariant subspace

PSC-I-T-PACKET-EXACT-SEQUENCE

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "K:InvariantSubspace(P)"
  ],
  "returns": "PacketwiseShortExactSequence(K,X,X/K)"
}
```

Packet descent through an invariant quotient

PSC-I-T-QUOTIENT-PACKET-DESCENT

```
{
  "params": [
    "P:PacketSystem(X,I)",
    "K:InvariantSubspace(P)"
  ],
  "returns": "PacketSystem(X/K,I)"
}
```

Naturality of generated hulls and transfer blocks

PSC-I-T-NATURALITY

```
{
  "params": [
    "Phi:PacketCompatibleOperatorMorphism",
    "S:SeedSet"
  ],
```

```
  "returns": "NaturalityCertificate(Phi,S)"
}
```

Realization functor and preservation profile

PSC-I-D-REALIZATION-FUNCTOR

```
{
  "params": [
    "F:AdditiveFunctor(C,D)",
    "profile:PreservationProfile"
  ],
  "returns": "RealizationFunctor(C,D,profile)"
}
```

Equational realization principle

PSC-I-T-REALIZATION-EXPORT

```
{
  "params": [
    "Theta:TheoremRecord",
    "F:RealizationFunctor"
  ],
  "requires": [
    "operations_used(Theta)\u2286verified_preservation(F)",
    "target_hypotheses_verified"
  ],
  "returns": "TransportCertificate(Theta,F)"
}
```

Semantic realization criterion

PSC-I-D-SEMANTIC-REALIZATION

```
{
  "params": [
    "Theta:TheoremRecord",
    "F:RealizationFunctor"
  ],
  "returns": "SemanticRealizationCriterion(Theta,F)"
}
```

C Complete action registry

The registry below contains all 37 action names. “Implemented” describes the runtime dispatcher. It does not assert that all hypotheses of every linked source record are automatically validated. Record associations are links for inspection.

AGGREGATE_PACKETS

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-L-PACKET-AGGREGATION

ANALYZE_PACKETS

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_EXACT_FINITE

Records. PSC-I-C-PACKET-UNIVERSAL-PROPERTY; PSC-I-C-PACKETWISE-EQUALITY; PSC-I-D-ANALYSIS-SYNTHESIS; PSC-I-T-EXACT-PACKET-COORDINATES

ASSEMBLE_BLOCK_ALGEBRA

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-DIMENSION-COUNT; PSC-I-T-BLOCK-ALGEBRA

BLOCKIFY_OPERATOR

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_EXACT_FINITE

Records. PSC-I-D-TRANSFER-BLOCK; PSC-I-T-OPERATOR-RECONSTRUCTION

BUILD_SEED_HULL

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-D-GENERATED-HULL; PSC-I-D-SEED-EVALUATION

BUILD_SUPPORT_ALGEBRA

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-SUPPORT-PATTERN-STAR; PSC-I-T-PREORDER-ALGEBRA-DECOMPOSITION; PSC-I-T-SUPPORT-PATTERN-ALGEBRA

BUILD_TRANSFER_GRAPH

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-D-TRANSFER-GRAPH

CHANGE_PACKET_COORDINATES

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY

Records. PSC-I-T-COORDINATE-TRANSPORT

CHECK_REALIZATION_PROFILE

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_SEMANTIC_GATE

Records. PSC-I-D-REALIZATION-FUNCTOR; PSC-I-D-SEMANTIC-REALIZATION

COMBINE_SEED_SYSTEMS

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-P-SHARED-DIRECT-SUM; PSC-I-P-TENSOR-BOUND; PSC-I-T-EXACT-SEQUENCE-BOUNDS; PSC-I-T-INDEPENDENT-DIRECT-SUM

COMPLETE_PACKET_SPLIT

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-COMPLETION-PACKET-SPLIT; PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS

COMPLETE_TOPOLOGICAL_REALIZATION

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-TOPOLOGICAL-PACKET-EXACT-SEQUENCE; PSC-I-T-DENSE-REALIZATION; PSC-I-T-QUOTIENT-TOPOLOGY-COMPLETION

COMPOSE_TRANSFER_BLOCKS

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_EXACT_FINITE

Records. PSC-I-C-TENSOR-TRANSFER-BLOCKS; PSC-I-T-BLOCK-MULTIPLICATION

COMPUTE_PACKET_PURE_RANK

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-C-SOURCE-COMPONENT-COUNT; PSC-I-D-DIRECTED-SEED-RANK; PSC-I-P-PACKET-PURE-RANK-BOUNDS; PSC-I-T-SOURCE-COMPONENT-SEED-RANK

COMPUTE_REACHABILITY

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-P-REACHABILITY-OBSTRUCTION; PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT

COMPUTE_SEED_RANK

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-C-ALGEBRAICALLY-CLOSED-SINGLE-OPERATOR; PSC-I-C-INVARIANT-FACTOR-COUNT; PSC-I-D-DIRECTED-SEED-RANK; PSC-I-D-SEED-RANK; PSC-I-P-CYCLIC-CHARACTERIZATION; PSC-I-T-MODULE-TOP; PSC-I-T-SEMISIMPLE-SEED-RANK; PSC-I-T-SINGLE-OPERATOR; PSC-I-T-SOURCE-COMPONENT-SEED-RANK

DECOMPOSE_PRIMARY

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-ALGEBRAICALLY-CLOSED-SINGLE-OPERATOR; PSC-I-C-CYCLIC-MATRIX; PSC-I-C-INVARIANT-FACTOR-COUNT; PSC-I-T-SINGLE-OPERATOR

DESCEND_PACKET_QUOTIENT

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY

Records. PSC-I-C-QUOTIENT-PACKET-DESCENT; PSC-I-T-QUOTIENT-PACKET-DESCENT

DUALIZE_PACKETS

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-P-DUAL-PACKET-SYSTEM

EVALUATE_PACKET_CONDITIONING

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-D-PACKET-CONDITION-NUMBER; PSC-I-P-CONDITIONED-COORDINATES

EXPAND_OPERATOR_PATHS

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY

Records. PSC-I-C-BLOCK-POWERS

EXPORT_PSCI_INTERFACE

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-EXPORT

EXTRACT_PACKET_COMMUTANT

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-P-BLOCK-DIAGONAL-COMMUTANT

FACTOR_ACTION

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-P-ACTION-FACTORIZATION

LOWER_INTERTWINER

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_EXACT_FINITE

Records. PSC-I-T-BLOCK-INTERTWINING

PRESENT_AS_FREE_QUOTIENT

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-C-MINIMAL-FREE-COVER; PSC-I-C-ONE-SEED-PRESENTATION; PSC-I-T-FREE-PRESENTATION

PROPAGATE_PACKET_SUPPORT

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY

Records. PSC-I-C-PACKET-SUPPORT-PROPAGATION

PUSH_GENERATORS_THROUGH_QUOTIENT

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-P-SCALAR-QUOTIENT-WITNESS; PSC-I-T-QUOTIENT-MONOTONICITY

REALIZE_GRAMMAR

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-D-COLLAPSED-REALIZATION; PSC-I-T-TYPED-MODULE-EQUIVALENCE

REDUCE_TO_SEMISIMPLE_TOP

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-LOCAL-ALGEBRA-FORMULA; PSC-I-T-MODULE-TOP; PSC-I-T-SEMISIMPLE-SEED-RANK

RESOLVE_PACKET_EXACT_SEQUENCE

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-C-TOPOLOGICAL-PACKET-EXACT-SEQUENCE; PSC-I-T-PACKET-EXACT-SEQUENCE

TENSOR_PACKETS

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY

Records. PSC-I-T-TENSOR-PACKET-SYSTEM

TRANSPORT_MORPHISM

Runtime implemented. No

Modes. EXACT_FIXTURE_ONLY, PLANNER_ONLY

Records. PSC-I-D-PACKET-MORPHISM; PSC-I-D-SEED-MORPHISM; PSC-I-P-SEED-SYSTEM-CATEGORY; PSC-I-T-NATURALITY

TRANSPORT_SEED_RANK

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-C-ISOMORPHISM-INVARIANCE; PSC-I-P-SCALAR-EXTENSION-MONOTONICITY

TRANSPORT_THEOREM

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_SEMANTIC_GATE

Records. PSC-I-P-REALIZATION-EXPORT; PSC-I-T-REALIZATION-EXPORT

VALIDATE_PACKET_SYSTEM

Runtime implemented. Yes

Modes. RUNTIME_EXECUTABLE_EXACT_FINITE

Records. PSC-I-D-PACKET-SYSTEM

VERIFY_PROPAGATION_HYPOTHESES

Runtime implemented. No

Modes. PLANNER_ONLY

Records. PSC-I-D-EDGE-SURJECTIVE; PSC-I-D-PACKET-PROPAGATING

D Operation and hypothesis vocabulary

The runtime ontology separates structural operations from hypothesis classes. These names form a controlled vocabulary for profiles and source-derived annotations. They are not a complete formal semantics for every mathematical construction named.

Name	Class
`identity`	structural
`zero`	structural
`finite_sum`	structural
`additive_inverse`	structural
`composition`	structural
`scalar_action`	structural
`direct_sum`	structural
`finite_product`	structural
`biproduct`	structural
`kernel`	structural
`cokernel`	structural
`image`	structural
`quotient`	structural
`tensor_product`	structural
`dual`	structural
`packet_idempotents`	structural
`transfer_blocks`	structural
`support_relation`	structural
`path_composition`	structural
`polynomial_action`	structural
`jacobson_radical`	structural
`semisimple_decomposition`	structural
`primary_decomposition`	structural
`universal_property`	structural
`functoriality`	structural
`exact_sequence`	structural
`closure`	structural
`completion`	structural
`topology`	structural
`norm`	structural
`adjoint`	structural
`dimension`	structural
`finite_indexing`	hypothesis
`surjectivity`	hypothesis
`injectivity`	hypothesis
`isomorphism`	hypothesis
`continuity`	hypothesis
`boundedness`	hypothesis
`closedness`	hypothesis
`density`	hypothesis
`orthogonality`	hypothesis
`self_adjointness`	hypothesis
`positivity`	hypothesis
`integrality`	hypothesis

Name	Class
unconditional_convergence	hypothesis
hausdorff	hypothesis
complete_field	hypothesis
invariant_subspace	hypothesis
commutation	hypothesis
transitivity	hypothesis
edge_surjectivity	hypothesis
packet_propagation	hypothesis
nonzero_packet_spaces	hypothesis

E Runtime Lean record map

This is the eight-entry mapping attached to the archived runtime. The evidence descriptions and source-match categories are reproduced from that map. A fresh build performed elsewhere must be documented separately.

Finite packet system

PSC-I-D-PACKET-SYSTEM

Source match. `structural_exact`

Stored certificate status. `compiled_in_verified_tranche`

Declarations. `Komorebi.Foundation.LinearProjectorFamily`

In the runtime formalization, ProjectorCore.lean supplies structure axioms for idempotence, pairwise annihilation, and completeness.

Packet analysis and synthesis

PSC-I-D-ANALYSIS-SYNTHESIS

Source match. `stronger_typed_realization`

Stored certificate status. `kernel_accepted`

Declarations. `Komorebi.Foundation.LinearProjectorFamily.PacketSpace`; `Komorebi.Foundation.LinearProjectorFamily.rangeAnalysis`; `Komorebi.Foundation.LinearProjectorFamily.rangeSynthesis`

ExactCoordinates.lean defines the packet space as the product of projector ranges, so both transforms carry range certificates.

Exact packet-coordinate isomorphism

PSC-I-T-EXACT-PACKET-COORDINATES

Source match. `stronger_typed_realization`

Stored certificate status. `kernel_accepted`

Declarations. `Komorebi.Foundation.LinearProjectorFamily.exactPacketCoordinates`

Explicitly listed as Kernel accepted in VERIFICATION_REPORT.md; proves both inverse laws for range-restricted packet coordinates.

Packetwise equality

PSC-I-C-PACKETWISE-EQUALITY

Source match. `one_direction_plus_trivial_reverse`

Stored certificate status. `compiled_in_verified_tranche`

Declarations. `Komorebi.Foundation.LinearProjectorFamily.eq_of_projectors_eq`

The nontrivial direction is proved in ProjectorCore.lean; the reverse direction follows by congruence from equality.

Transfer block

PSC-I-D-TRANSFER-BLOCK

Source match. `generalized_domain_codomain`

Stored certificate status. `compiled_in_verified_tranche`

Declarations. `Komorebi.Foundation.transferBlock`

Definition allows distinct source and target packet systems; the PSC-I endomorphism case is a specialization.

Exact operator reconstruction

PSC-I-T-OPERATOR-RECONSTRUCTION

Source match. `generalized_domain_codomain`

Stored certificate status. `kernel_accepted`

Declarations. `Komorebi.Foundation.transferBlock_reconstruction`

Explicitly listed as Kernel accepted; reconstructs any linear map from all source-target blocks using completeness of both packet families.

Packet matrix multiplication

PSC-I-T-BLOCK-MULTIPLICATION

Source match. `generalized_composition_form`

Stored certificate status. `kernel_accepted`

Declarations. `Komorebi.Foundation.transferBlock_comp`

Explicitly listed as Kernel accepted; inserts the complete middle packet family and yields the local composition sum.

Support-pattern algebra criterion

PSC-I-T-SUPPORT-PATTERN-ALGEBRA

Source match. `partial_sufficient_direction`

Stored certificate status. `kernel_accepted_related`

Declarations. `Komorebi.Algebra.MatrixSupported.mul`

Verified tranche proves multiplicative closure for transitive support relations; it is related evidence, not a source-equivalent certificate for every direction of the PSC-I criterion.

Reading scope

This formal companion connects Chapter I to its Lean sources, correspondence map, and verification evidence. Section 2 identifies the source tree and distinguishes complete correspondences from partial or unmapped records.

> The tutorial and its complete 35-row contract describe the foundational formalization: 20 source files, 84 mapped targets, and 1,126 recorded namespace entries. Those counts belong to that component. The later continuation has a larger source closure and a separate map to the expanded manuscript's 108 records. The two inventories are retained with their exact scopes.

Evidence for the 108-record correspondence is indexed separately from the foundational receipts. Rendering this guide is a presentation check. Compiler acceptance, logical kernel replay, source identity, and source-to-statement correspondence are reported as distinct checks.

1 Reading the mathematical, formal, and computational texts together

The mathematical text develops the language and its examples. The computational companion gives stable records, typed dependencies, semantic profiles, and executable validation interfaces. This guide makes the Lean development usable: where the definitions live, what each theorem actually accepts, how its proof works, and what evidence connects the delivered source to an accepted kernel run.

The foundational formalization in Sections 3–14 has twenty Lean source files.
The source register identifies that formalization. The mathematical text has an expanded core of 108 stable records, an inventory distinct from the 35 foundational Lean contract rows.

> The guide retains both the scope and the exclusions of every contract row. For example, foundational row 10 proves a split-semisimple formula after algebra and module equivalences are supplied. The later continuation constructs compatible decompositions and establishes the general division-ring formula. The new result is documented through its own declarations and evidence. It does not change the inputs of foundational row 10.

1.1 What the numbers count

Quantity	Meaning
20 source files	Local Lean files, including the import aggregator, focused Mathlib import file, and standalone audit.
35 contract rows	Mathematical statement families and specifications in the foundational contract map.
84 mapped targets	Distinct Lean declaration names used by those rows. They include definitions and equivalences as well as theorems.
1,126 entries	Entries in the project namespace inspected by the recorded audit, including generated helpers.
108 stable records	The expanded manuscript inventory. The current correspondence map and the machine payload's separate eight-record map have different evidence scopes.

None of these figures is a count of 1,126 independent new theorems. The useful unit of mathematical coverage is a statement together with its hypotheses, its precise conclusion, its declaration target, and the evidence attached to that target.

1.2 Suggested reading routes

A reader new to Lean can begin with Sections 3–5, which unpack the syntax through exact packet coordinates. A mathematician checking coverage can go directly to the exhaustive map in Section 14, then inspect the relevant source type and proof. A contributor preparing a build should read Sections 10 and 11 before interpreting any successful command.

The displayed Lean listings are excerpts from the preserved source. Their surrounding imports, namespace declarations, and section variables remain in the original files. They are reading excerpts, not advertised standalone files. The editable companion includes their source ranges and hashes in `generated/excerpts.json`.

2 Formal sources, correspondence, and evidence

The mathematical text and both companions share one statement inventory. The traditional manuscript supplies the statements and proofs. The machine catalog reproduces those statements with stable identifiers and source locators. The formal correspondence map records the exact Lean support for each statement. The source archive uses the following stable paths:

File or directory	Purpose
`formal/current/`	Flat formal source closure, including the continuation modules and their earlier dependencies.
`formal/coverage/COVERAGE_108.json`	All 108 statements, source proofs and hashes, declaration mappings, scopes, and record classifications.
`formal/evidence/`	Compiler, replay, audit, identity, and boundary-check receipts, indexed by the verification record.
`editable/traditional/`	Mathematical source, including the stated hypotheses of the split-semisimple corollary.

The inventory has 69 proof-level records. Definitions, classical inputs, boundaries, and documentary records are listed separately in the map and are excluded from this proof-level count.

Status	Covered	Partial	Not mapped
Proof-level records	69	0	0

"Covered" means a complete correspondence at the recorded scope. "Partial" includes substantive missing proofs and consequences not yet assembled from existing declarations. "Not mapped" means that this source closure has no established complete declaration for the record. These statuses are not a percentage of independent theorems: the records differ in scope and may share mathematical dependencies.

2.1 Primary and semisimple constructions

The continuation proves the intrinsic primary quotient and generator count, the geometric formula, scalar extension, and the six-way cyclic criterion. It also proves noncommutative module-top reduction and the nonsplit semisimple formula, including actual compatible decompositions from the Wedderburn–Artin existence theorem. The final existence theorems do not take a desired rank equality or a compatible decomposition as an input. They are noncomputable mathematical constructions, not executable decomposition algorithms.

Source layer	Mathematical entry points
`PaperICompletion/`, `PaperICompletion3/`	Spectral certificates, invariant factors, primary seed formulas, and operator-level bridges.
`PaperICompletion4/`	Intrinsic primary decomposition, residue quotients, and the operator seed formula.
`PaperICompletion5/`	Linear primary eigenspaces, the geometric seed formula, and its cyclic criterion.
`PaperICompletion6/`	Scalar extension, one invariant factor, and the six-way cyclic criterion.
`PaperICompletion7/`	Nakayama, module-top reduction, and the local residue division-ring case.
`PaperICompletion8/`	Division-ring row generation, products, regular multiplicities, compatible presentations, and semisimple existence/-formulas.

The original `KomorebiVerification/` modules remain the foundation for packet coordinates, transport, seed rank, matrix models, and bounded-packet analysis. Their explanations occupy the following tutorial sections. Current declarations may strengthen those earlier results. The original theorem signatures retain their own input boundaries.

2.2 Source modules and presentation conventions

The `PaperIRelease/` layer extends the preceding continuation. Its source files are grouped by mathematical construction:

Source family	Construction
`GrammarAlgebra`, `GrammarTyped`, `GrammarModules`, `GrammarEquivalence`, `GrammarIdeal`	Typed realizations, module categories, comparison functors, and quotient-algebra presentations.
`SeedOperations`, `ExactSequences`, `ScalarExtension`, `PolynomialModule`, `PolynomialMonic`	Generator bounds, exact sequences, scalar extension, and canonical polynomial-module wrappers.
`PacketAggregation`, `PacketDescent`, `PacketExactSequence`, `PacketUniversal`, `TensorPackets`	Actual range packets, quotient and restriction maps, exactness, universal properties, and tensor constructions.
`AlgebraSupport`, `AlgebraPreorder`, `AlgebraQuotient`, `PacketCorners`, `BoundedStar`	Support algebras, diagonal and quotient structure, local actions, and bounded adjoints.
`GraphReachability`, `GraphSources`, `GraphSourceCount`, `PacketPureRank`, `PacketSourceRank`	Transfer reachability, source components, and packet-pure generator counts.
`TopologySeedRank`, `TopologyPresentations`, `TopologyCountablePackets`, `TopologyCategories`	Closed hulls, locally convex presentations, unconditional packets, and continuous categorical maps.
`CategoryPackets`, `CategoryUniversal`, `CategoryExport`	Packet-system categories, naturality, and preservation of equations under functors.

These entries support source navigation. The correspondence map supplies the exact declaration names and hypotheses. Its verification receipt determines the accepted scope.

For the typed-grammar correspondence, the finite-vertex path algebra uses its standard vertex-and-arrow presentation: vertex idempotents are pairwise annihilating and sum to one, and each arrow lies in its target/source corner. The formal `PathAlgebra` names that presentation. `GrammarIdeal` takes an

arbitrary two-sided ideal and constructs the comparison between the iterated ideal quotient and the combined presentation. Its two comparison maps are proved inverse. Thus the presentation convention is explicit when reading the manuscript notation kQ/I. It is not an assumed module-category equivalence or a supplied conclusion about generators.

2.3 Verification scope

The verification receipt reports complete correspondences for all 69 proof-level records in the 108-record inventory. Each conclusion retains the hypotheses and presentation conventions recorded in the correspondence map. Definitions, boundaries, and classical inputs are classified separately. The record includes targeted semantic reviews for 47 proof-level records and accepted predecessor correspondences for 22 others. The total of 69 accepted correspondences does not mean that all records received independent reviews in the same verification event.

The recorded audit checks 116 formal source modules plus the retained boundary suite. Including that suite, it inventories 3513 source-owned entries and checks 3500 logical declarations. Of these, 3499 are checked by logical replay and one declaration sharing an upstream name receives a fresh-name kernel body check after exact type/value comparison. It inventories 13 runtime helpers separately and records 12 rejected negative controls. The allowed transitive axiom set is `Classical.choice`, `Quot.sound`, `propext`. The receipt is `formal/evidence/RELEASE_VERIFICATION.json`. Its detailed compiler and logical-replay evidence is under `formal/evidence/whole_closure/`.

The recorded computational checks pass for 96 exact finite or symbolic visual assertions, 33 companion tests, and 10 regenerated PDF/PNG figure pairs. These bounded computations have their own scope and do not replace the formal declarations.

The pinned upstream dependency cache is reused. Logical replay uses Lean's official kernel. It is not an independently implemented checker. The source-to-manuscript correspondence is a separate reviewed mathematical artifact.

2.4 Proof-level correspondence index

The following index is generated from the correspondence map. The machine companion supplies the source statements, and the JSON map supplies every declaration name and exact scope note. Every covered status must be read with that scope note before reusing a theorem.

Stable record	Recorded status
`PSC-I-T-TYPED-MODULE-EQUIVALENCE`	covered
`PSC-I-P-ACTION-FACTORIZATION`	covered
`PSC-I-P-ALGEBRA-MODULE-GENERATION`	covered
`PSC-I-T-FREE-PRESENTATION`	covered
`PSC-I-C-MINIMAL-FREE-COVER`	covered
`PSC-I-C-ONE-SEED-PRESENTATION`	covered
`PSC-I-P-ANNIHILATOR-TWO-SIDED`	covered
`PSC-I-T-COMPLETED-PRESENTATION`	covered
`PSC-I-P-CYCLIC-CHARACTERIZATION`	covered
`PSC-I-P-ACTING-ALGEBRA-MONOTONICITY`	covered
`PSC-I-T-QUOTIENT-MONOTONICITY`	covered
`PSC-I-C-ISOMORPHISM-INVARIANCE`	covered
`PSC-I-T-EXACT-SEQUENCE-BOUNDS`	covered

Stable record	Recorded status
PSC-I-P-SHARED-DIRECT-SUM	covered
PSC-I-T-INDEPENDENT-DIRECT-SUM	covered
PSC-I-P-TENSOR-BOUND	covered
PSC-I-P-SCALAR-EXTENSION-MONOTONICITY	covered
PSC-I-P-SCALAR-QUOTIENT-WITNESS	covered
PSC-I-T-MODULE-TOP	covered
PSC-I-C-LOCAL-ALGEBRA-FORMULA	covered
PSC-I-L-REGULAR-MODULE-MULTIPLICITIES	covered
PSC-I-T-SEMISIMPLE-SEED-RANK	covered
PSC-I-C-ALGEBRAICALLY-CLOSED-FORMULA	covered
PSC-I-C-FULL-MATRIX	covered
PSC-I-C-MULTIPLICITY-OBSTRUCTION	covered
PSC-I-T-SINGLE-OPERATOR	covered
PSC-I-C-INVARIANT-FACTOR-COUNT	covered
PSC-I-C-ALGEBRAICALLY-CLOSED-SINGLE-OPERATOR	covered
PSC-I-C-CYCLIC-MATRIX	covered
PSC-I-L-PACKET-AGGREGATION	covered
PSC-I-T-EXACT-PACKET-COORDINATES	covered
PSC-I-C-PACKETWISE-EQUALITY	covered
PSC-I-C-PACKET-UNIVERSAL-PROPERTY	covered
PSC-I-T-TENSOR-PACKET-SYSTEM	covered
PSC-I-P-DUAL-PACKET-SYSTEM	covered
PSC-I-T-BANACH-PACKET-ISOMORPHISM	covered
PSC-I-P-CONDITIONED-COORDINATES	covered
PSC-I-C-COMPLETION-PACKET-SPLIT	covered
PSC-I-L-UNCONDITIONAL-COMPLETENESS	covered
PSC-I-T-COUNTABLE-UNCONDITIONAL-PACKETS	covered
PSC-I-T-OPERATOR-RECONSTRUCTION	covered
PSC-I-T-BLOCK-MULTIPLICATION	covered
PSC-I-C-BLOCK-POWERS	covered
PSC-I-C-ADJOINT-REVERSAL	covered
PSC-I-C-TENSOR-TRANSFER-BLOCKS	covered
PSC-I-T-BLOCK-INTERTWINING	covered
PSC-I-T-BLOCK-ALGEBRA	covered
PSC-I-C-PACKET-SUPPORT-PROPAGATION	covered
PSC-I-C-DIMENSION-COUNT	covered
PSC-I-P-BLOCK-DIAGONAL-COMMUTANT	covered
PSC-I-T-SUPPORT-PATTERN-ALGEBRA	covered
PSC-I-C-SUPPORT-PATTERN-STAR	covered
PSC-I-T-PREORDER-ALGEBRA-DECOMPOSITION	covered
PSC-I-T-ALGEBRA-REACHABILITY-SUPPORT	covered
PSC-I-P-REACHABILITY-OBSTRUCTION	covered
PSC-I-P-PACKET-PURE-RANK-BOUNDS	covered
PSC-I-T-SOURCE-COMPONENT-SEED-RANK	covered
PSC-I-C-SOURCE-COMPONENT-COUNT	covered
PSC-I-T-COORDINATE-TRANSPORT	covered
PSC-I-T-NATURALITY	covered
PSC-I-P-SEED-SYSTEM-CATEGORY	covered
PSC-I-T-PACKET-EXACT-SEQUENCE	covered
PSC-I-T-QUOTIENT-PACKET-DESCENT	covered
PSC-I-C-QUOTIENT-PACKET-DESCENT	covered

Stable record	Recorded status
PSC-I-C-TOPOLOGICAL-PACKET-EXACT-SEQUENCE	covered
PSC-I-T-DENSE-REALIZATION	covered
PSC-I-T-QUOTIENT-TOPOLOGY-COMPLETION	covered
PSC-I-T-REALIZATION-EXPORT	covered
PSC-I-P-REALIZATION-EXPORT	covered

3 The type system as mathematical bookkeeping

Lean's types carry assumptions that would otherwise be easy to leave implicit. A statement over a semiring and a module is not automatically a statement about a Hilbert space. A linear equivalence is not automatically a continuous linear equivalence. An endomorphism over the ground field is not automatically linear over a larger acting algebra.

Lean expression	Mathematical role
`[SemiringR]`	An instance supplying the semiring laws for the coefficient object. Square brackets request typeclass synthesis.
`[ModuleRM]`	The scalar action and its laws. The module is over the named scalar object, not an unspecified field.
`M→ₗ[R]N`	An R-linear map, with additive and scalar compatibility bundled into the value.
`Module.EndRM`	The endomorphism type $\mathrm{End}_R(M)$.
`M≃ₗ[R]N`	A linear equivalence with inverse laws.
`X→L[k]Y`	A continuous linear map between normed spaces. The uppercase L is significant.
`X≃L[k]Y`	A continuous linear equivalence.
`M→ₛₗ[σ]N`	A semilinear map along the specified scalar homomorphism σ.
`LinearMap.rangef`	The range as a submodule. An element includes a witness that it is attained by f.
`[Fintypeι]`	A finite enumeration of the index type, allowing finite sums over all indices.
`ι→₀A`	Finitely supported coefficient families, the free A-module on the label type.
`IsLeastSR`	Both $R \in S$ and $R \leq r$ for every $r \in S$. This packages upper and lower bounds.

Arguments in braces are implicit parameters inferred from use. Arguments in parentheses are explicit inputs. A section's `variable` declarations can supply parameters to all later declarations that use them. Therefore a theorem copied without its section context can look less constrained than its actual elaborated type. The preserved `mapped_declarations.log` prints every mapped type with universe information and is the right companion when checking the complete signature.

The package disables automatic invention of implicit variables with `autoImplicit=false` and `relaxedAutoImplicit=false`. This makes a misspelled or undeclared variable an error instead of silently introducing a new parameter. It does not remove explicit hypotheses, nor prove that a chosen formal statement is the intended prose statement.

3.1 Proof terms and the tactics used here

`exacth` closes the current goal with an existing proof of its type. `rfl` proves a definitional equality after reduction. `rw` substitutes proved equalities. `simp` applies a collection of simplification lemmas and produces a proof term. `funext` reduces equality of functions to equality at each input. `Subtype.ext` reduces equality of range elements to equality of their underlying vectors.

The command `classical` makes classical instances available during elaboration. It is not a placeholder. Some resulting declarations depend on `Classical.choice`. The audit records this explicitly. A `noncomputabledef` can define a mathematically precise object through choice or noncomputable library constructions. It does not by itself provide executable numerical code.

These distinctions matter particularly for graph rank. A formula defined using a finite exact spectrum may be mathematically well defined and formally verified while still not being an extracted eigensolver.

4 The source architecture

The project is organized around a finite algebraic core, seed generation, concrete matrix models, operator and packet completion, and a separate analytic component. MathlibImports.lean centralizes the focused external imports.

Layer	Main responsibility
Projector core	`ProjectorCore`, `ExactCoordinates`, `FibreRefinement`, and `ResidueProjectors`: finite masks, range coordinates, exact reconstruction, and refinement.
Transport and blocks	`PaperITransport` and `PaperIBlockCompletion`: conjugation, naturality, operator-level block identities, and support patterns.
Seed rank	`SeedRankBasics` and `SeedRankOperations`: generation, free presentations, cardinal minima, semilinear transport, products, tensor bounds, and scalar quotients.
Matrix realizations	`SemisimpleSeedRank`, `FullMatrixCyclicity`, `SemisimpleCompletion`: concrete row-space bounds and their connection to canonical seed rank.
One operator	`PathGraphKrylov` and `SingleOperatorGraph`: polynomial orbits, exact minimum certificates, paths, and cycles.
Packet completion	`PacketNaturalityCompletion`: dependent block algebra, support, reachability, strict morphisms, and the typed-quiver certificate.
Analysis	`TopologicalCompletion`: bounded projectors, closure, dense realization, completion, and the bilateral shift.
Models and specifications	`EllipticAlgorithms`: the elliptic quotient model and the three finite mathematical algorithm specifications.
Audit	`KernelAudit`: concrete instantiations, the exact mapped-target list, and the namespace-wide axiom gate.

The accompanying source records contain the direct-import table. File imports do not identify every logical dependency of each declaration. A declaration's transitive axiom set is obtained separately from the Lean environment.

5 Exact packet coordinates, from assumptions to inverse laws

The starting object is a finite family of linear projectors. Its structure records three separate laws: each map is idempotent, different maps annihilate one another, and their finite sum is the identity.

Source: KomorebiVerification/ProjectorCore.lean.

```
structure LinearProjectorFamily
    (R : Type u) (M : Type v) (ι : Type w)
    [Semiring R] [AddCommMonoid M] [Module R M] [Fintype ι] where
  proj : ι → Module.End R M
  idempotent : ∀ i x, proj i (proj i x) = proj i x
  orthogonal : ∀ {i j}, i ≠ j → ∀ x, proj i (proj j x) = 0
  complete : ∀ x, (∑ i, proj i x) = x
```

Here orthogonality means pairwise annihilation. An inner product is not among the assumptions, so this word is algebraic at this stage. Self-adjointness appears only in the later adjoint result.

Write P_i for the maps. The intended coordinates are

$$\mathcal{P} = \prod_{i \in I} \mathrm{range}(P_i), \qquad \mathsf{A}(x) = (P_i x)_i, \qquad \mathsf{S}((x_i)_i) = \sum_i x_i.$$

The type of the packet space is crucial. It is the product of independently defined ranges. It is not defined as the image of analysis, and synthesis is the ordinary sum. Thus the reverse inverse law must be proved.

Source: KomorebiVerification/ExactCoordinates.lean.

```
/-- The explicit product of the projector ranges. -/
abbrev PacketSpace (P : LinearProjectorFamily R M ι) : Type max v w :=
  (i : ι) → LinearMap.range (P.proj i)

/-- Analysis, with each projected value certified to lie in its projector range. -/
def rangeAnalysis (P : LinearProjectorFamily R M ι) : M →ₗ[R] PacketSpace P where
  toFun x i := ⟨P.proj i x, ⟨x, rfl⟩⟩
  map_add' x y := by
    funext i
    apply Subtype.ext
    exact (P.proj i).map_add x y
  map_smul' c x := by
    funext i
    apply Subtype.ext
    exact (P.proj i).map_smul c x

/-- Synthesis is the direct finite sum of packet values. -/
def rangeSynthesis (P : LinearProjectorFamily R M ι) : PacketSpace P →ₗ[R] M where
  toFun packets := ∑ i, (packets i : M)
  map_add' packets packets' := by
    simp only [Pi.add_apply, Submodule.coe_add, Finset.sum_add_distrib]
  map_smul' c packets := by
    simp only [Pi.smul_apply, SetLike.val_smul, Finset.smul_sum, RingHom.id_apply]
```

The term `⟨P.projix,⟨x,rfl⟩⟩` has two layers. The outer pair contains the vector and its proof of range membership. The inner pair gives the preimage x and the equality showing that projecting it gives the required vector. The coercion `(packetsi:M)` later forgets this proof component and returns the underlying vector for summation.

5.1 Why the two directions use different hypotheses

Synthesis after analysis is exactly completeness:

$$\mathsf{SA}(x) = \sum_i P_i x = x.$$

Analysis after synthesis uses the range conditions as well. If $x_j \in \mathrm{range}(P_j)$, choose a preimage y_j. Idempotence gives $P_j x_j = x_j$, while pairwise annihilation gives $P_i x_j = 0$ for $i \neq j$. Consequently

$$P_i\Big(\sum_j x_j\Big) = \sum_j P_i x_j = x_i.$$

The source follows this reasoning directly.

Source: `KomorebiVerification/ExactCoordinates.lean`.

```
/-- Completeness proves that synthesizing analyzed coordinates returns the input. -/
@[simp] theorem rangeSynthesis_rangeAnalysis
    (P : LinearProjectorFamily R M ι) (x : M) :
    P.rangeSynthesis (P.rangeAnalysis x) = x := by
  change (∑ i, P.proj i x) = x
  exact P.complete x

/-- Idempotence and orthogonality prove that analysis recovers every range packet. -/
@[simp] theorem rangeAnalysis_rangeSynthesis
    (P : LinearProjectorFamily R M ι) (packets : PacketSpace P) :
    P.rangeAnalysis (P.rangeSynthesis packets) = packets := by
  classical
  funext i
  apply Subtype.ext
  change P.proj i (∑ j, (packets j : M)) = (packets i : M)
  rw [map_sum, Finset.sum_eq_single i]
  · exact P.proj_packet_self packets i
  · intro j _ hji
    exact P.proj_packet_eq_zero packets hji.symm
  · simp
```

After `funexti`, the goal concerns one packet. `Subtype.ext` removes the irrelevant distinction between the proof components of range elements. `map_sum` distributes the linear projector over the finite sum. `Finset.sum_eq_singlei` separates the surviving diagonal term from the annihilated off-diagonal terms. The final `simp` handles the finite-set side condition.

Once the two laws are established, the linear equivalence merely packages them.

Source: `KomorebiVerification/ExactCoordinates.lean`.

```
/-- Exact linear coordinates in the independent product of projector ranges. -/
def exactPacketCoordinates
    (P : LinearProjectorFamily R M ι) : M ≃ₗ[R] PacketSpace P where
  toFun := P.rangeAnalysis
  invFun := P.rangeSynthesis
  left_inv := P.rangeSynthesis_rangeAnalysis
  right_inv := P.rangeAnalysis_rangeSynthesis
  map_add' := (P.rangeAnalysis).map_add
  map_smul' := (P.rangeAnalysis).map_smul
```

There is no inverse selected by choice at this step. Both maps are already explicit. The proof is stronger than recovery of vectors that happen to lie in the image of analysis: it covers every tuple of valid range packets.

5.2 Four coordinates and two parity packets

The standalone audit specializes the construction to $\mathbb{Q}^4$. Fine labels are $0, 1, 2, 3$ and the fine address is the identity. The coarse address is reduction modulo two. For

$$x = (a, b, c, d), \qquad Q_0 x = (a, 0, c, 0), \qquad Q_1 x = (0, b, 0, d),$$

the fine projections isolate one coordinate, and

$$Q_0 = P_0 + P_2, \qquad Q_1 = P_1 + P_3.$$

The general fiber constructor proves its laws from masks, rather than requiring the user to supply them for each instance.

Source: `KomorebiVerification/FibreRefinement.lean`.

```
/-- Every finite address map induces a complete orthogonal projector family. -/
def fibreLinearProjectorFamily
    [Semiring R] [AddCommMonoid M] [Module R M]
    [DecidableEq β] [Fintype β]
    (address : α → β) :
    Komorebi.Foundation.LinearProjectorFamily R (α → M) β where
  proj := fibreLinearProj (R := R) address
  idempotent b x := fibreProj_idempotent address b x
  orthogonal hbc x := fibreProj_orthogonal address hbc x
  complete x := fibreProj_reconstruction address x
```

The audit then proves the parity formula by specializing the independent coarse/fine refinement theorem.

Source: `KomorebiVerification/KernelAudit.lean`.

```
theorem parity_refinement (b : Fin 2) (x : Fin 4 → ℚ) :
    fibreProj parityAddress b x =
      ∑ c, if parityAddress c = b then fibreProj fineAddress c x else 0 := by
  simpa [fineAddress] using
    (fibreProj_refinement fineAddress parityAddress b x)
```

This is an instance of the theorem for an arbitrary input vector, not a comparison of a few sampled rational vectors. The same audit instantiates reconstruction of an arbitrary linear operator on $\mathbb{Q}^4$ from all sixteen fine blocks.

6 Transfer blocks, support, and transport

For source projectors P_i, target projectors Q_j, and a linear map $T : M \to N$, the output-first convention is

$$T_{ji} = Q_j T P_i.$$

The declaration `Komorebi.Foundation.transferBlock` has arguments source family, target family, map, output label, input label. Getting the last two arguments reversed can produce a well-typed but different map when source and target spaces coincide. The guide uses output-first indices throughout.

6.1 Reconstruction and multiplication

Completeness of both families gives

$$T = \sum_j \sum_i Q_j T P_i.$$

For $U : N \to L$ and a third family S_k, completeness of the middle family gives

$$(UT)_{ki} = \sum_j U_{kj} T_{ji}.$$

The earlier core proves pointwise identities. `PaperIBlockCompletion` promotes them to equality of linear maps using extensionality. Contract rows 19 and 20 are classified as stronger than the single-space manuscript instances because the compiled maps may have different source, middle, and target modules.

The block algebra development then restricts each map to the actual source packet and gives it codomain the actual target packet. This yields a dependent family

$$B_{ji} : \text{range}(P_i) \longrightarrow_R \text{range}(P_j),$$

which is the full packet-to-packet Hom space. Its multiplication is the finite sum of composable maps. The equivalence with ambient endomorphisms is not a definition of the block space as the image of ambient operators. Both restriction and synthesis are separately constructed and proved inverse.

6.2 Why nonzero packets appear in converse statements

If a support relation is transitive, supported blocks are closed under multiplication. Conversely, to detect a missing compositional edge from universal multiplication closure, one needs blocks whose composite is nonzero. Contract row 25 works over a field, with every packet range nontrivial and arbitrary packet-Hom blocks available. Those hypotheses provide the witnesses used by the converse.

A zero packet can make a relation edge mathematically invisible. The mathematical text therefore explicitly includes the necessary nonzero-packet assumptions in its $*$-support and source-component-count corollaries. Row 25 has its own nontriviality scope. The twenty-file foundational formalization retains that scope.

6.3 A reachability obstruction is a certificate for a lower bound

The reachability module first shows that word orbits of packet-supported seeds remain in the support reachable from those seeds. If a nonzero packet is unreachable, the generated algebraic hull is proper. The conclusion depends on the seed support as well as the graph. It is not a general rule saying that the number of graph sources equals unrestricted seed rank.

The typed quiver model makes the distinction concrete. It uses $U = k^2$, $V = k^3$, the embedding $A(u_1, u_2) = (u_1, u_2, 0)$, and the projection $B(v_1, v_2, v_3) = (v_1, v_2)$. Although $BA = 1_U$, the vector $(0, 0, 1)$ in V is outside range(A). No family supported solely at U generates the whole representation. A source seed and the missing V-direction give the displayed generating pair. Contract row 30 is the certificate for this model, not a classification of arbitrary quivers.

6.4 Coordinate transport and strict naturality

A linear equivalence $C : M \simeq N$ transports projectors by $P_i \mapsto CP_iC^{-1}$ and operators by $T \mapsto CTC^{-1}$. Blocks and their zero/nonzero pattern then transport exactly. The inverse matters: a noninjective realization can kill a nonzero block.

The strict naturality module also handles semilinear realized actions, with explicit algebra maps and maps of packet labels. Its statement about the generated hull uses the exact image algebra. Surjectivity gives the appropriate rank inequality. Equality and complete edge correspondence require the stronger bijectivity hypotheses recorded in row 28. The injective label map excludes packet merging from that particular strict interface.

7 Generation and seed rank

Seed rank depends on an acting algebra. It is not simply vector space dimension. One vector generates the natural module of a full matrix algebra, whereas a scalar action on k^n needs n independent generators.

The foundational definitions use the existing span-rank API for the top submodule.

Source: `KomorebiVerification/SeedRankBasics.lean`.

```
variable (A : Type u) (M : Type v)
variable [Semiring A] [AddCommMonoid M] [Module A M]

/-- A set of seeds generates `M` when its `A`-linear span is all of `M`. -/
def Generates (s : Set M) : Prop :=
  Submodule.span A s = ⊤

/-- A family generates when its range is a generating set. -/
def FamilyGenerates {ι : Type w} (v : ι → M) : Prop :=
  Generates A M (Set.range v)

/-- A module is cyclic when one element generates it. -/
def Cyclic : Prop :=
  ∃ x : M, Generates A M {x}

/--
The seed rank is the least cardinality of a set whose `A`-span is `M`.
```

```
Because it is cardinal-valued, it remains meaningful for modules that are not
finitely generated.  In particular, the zero module has seed rank zero.
-/
noncomputable def seedRank : Cardinal :=
  (⊤ : Submodule A M).spanRank

/-- The natural-valued minimum used for finite-generation corollaries. -/
noncomputable def seedFinrank : ℕ :=
  (⊤ : Submodule A M).spanFinrank
```

The cardinal-valued rank makes sense without finite generation. The natural-valued `seedFinrank` is the finite-generation minimum only under the hypotheses used by the finite-rank theorems. The zero module has rank zero. The cyclic/rank-one equivalence therefore includes a nontriviality hypothesis.

7.1 The one-seed presentation

For a seed $s \in M$, the presentation map is $a \mapsto a \cdot s$. Its range is exactly the cyclic submodule. Therefore s generates if and only if this map is surjective.

Source: `KomorebiVerification/SeedRankBasics.lean`.

```
/-- The universal one-seed presentation map, `a ↦ a • x`. -/
def oneSeedMap (x : M) : A →ₗ[A] M :=
  LinearMap.toSpanSingleton A M x

@[simp]
theorem oneSeedMap_apply (x : M) (a : A) :
    oneSeedMap A M x a = a • x := by
  rfl

/-- The image of the one-seed map is exactly the cyclic submodule it spans. -/
theorem range_oneSeedMap (x : M) :
    LinearMap.range (oneSeedMap A M x) = Submodule.span A {x} := by
  exact LinearMap.range_toSpanSingleton x

/-- A seed generates exactly when its presentation map is surjective. -/
theorem generates_singleton_iff_surjective (x : M) :
    Generates A M {x} ↔ Function.Surjective (oneSeedMap A M x) := by
  rw [Generates, ← range_oneSeedMap A M x, LinearMap.range_eq_top]
```

The first isomorphism theorem then supplies the quotient-to-range equivalence.

Source: `KomorebiVerification/SeedRankBasics.lean`.

```
/-- First-isomorphism presentation for one seed, with its exact range. -/
noncomputable def oneSeedQuotientEquivRange (x : M) :
    (A ⧸ LinearMap.ker (oneSeedMap A M x)) ≃ₗ[A]
      LinearMap.range (oneSeedMap A M x) :=
  (oneSeedMap A M x).quotKerEquivRange

/--
```

```
The range-valued quotient equivalence is the canonical presentation map:
the class of `a` is sent to the range element represented by `a • x`.
This is the explicit glue between the first-isomorphism construction and the
formula used in the manuscript.
-/
@[simp]
theorem oneSeedQuotientEquivRange_apply_mk (x : M) (a : A) :
    (oneSeedQuotientEquivRange A M x) (Submodule.Quotient.mk a) =
      ⟨a • x, by
        exact ⟨a, rfl⟩⟩ := by
  rfl
```

The displayed application theorem is an important connection to the mathematical text: the equivalence sends the class of a to the actual vector $a \cdot s$. It is not merely an unspecified isomorphism between abstract objects with the right dimensions.

For a noncommutative acting algebra, the kernel here is a left submodule of the regular module. The theorem does not promote it to a two-sided ideal. Likewise, no norm or bounded completion is manufactured by this algebraic construction. The analytic completion result needs a supplied continuous isometry with dense range.

For several seeds, `Finsupp.linearCombination` evaluates a finitely supported coefficient family. This works for arbitrary label types. The finite-minimum theorem states that, when the module is finitely generated, its seed finrank is the least n admitting a surjection from $\texttt{Finn}\to_0\texttt{A}$.

7.2 Surjections, image algebras, and two product actions

Under a surjective semilinear map $f : M \to N$ along $\sigma : A \to B$, an A-generating set maps to a B-generating set. Coefficients in the image of σ are already available in B, so the rank inequality does not require σ itself to be onto. An equality statement requires more: either the scalar map is surjective or the target action is literally restricted to the image algebra.

For products there are two different actions. A shared algebra acts diagonally on $M \times N$, producing an upper bound by the sum of ranks. Independent actions use $A \times B$ and yield the maximum of the two ranks. Coordinate idempotents in $A \times B$ separate the two components, which is precisely what a general shared action may fail to do.

The tensor theorem proves an upper bound under its stated commuting-action and scalar-tower laws. The scalar-quotient theorem gives a lower bound when the action on a finite-dimensional quotient factors through a character. A dimension bound for an arbitrary quotient would be a different statement.

7.3 A complete constructive example: full matrix cyclicity

Suppose $x \neq 0$ in k^n. Choose a coordinate j with $x_j \neq 0$. For any target vector y, make a matrix with only column j nonzero and entries $a_{ij} = y_i/x_j$. Then $Ax = y$. The source is short enough to read in full.

Source: `KomorebiVerification/FullMatrixCyclicity.lean`.

```
/-- A vector is cyclic for the natural action of the full matrix algebra. -/
def IsCyclic (x : ι → K) : Prop :=
```

```
  ∀ y : ι → K, ∃ A : Matrix ι ι K, Matrix.mulVec A x = y

/-- Every nonzero vector is cyclic under the full matrix algebra. -/
theorem isCyclic_of_ne_zero {x : ι → K} (hx : x ≠ 0) : IsCyclic x := by
  obtain ⟨j, hj⟩ : ∃ j, x j ≠ 0 := by
    by_contra h
    apply hx
    funext i
    by_contra hi
    exact h ⟨i, hi⟩
  intro y
  let A : Matrix ι ι K := fun i k ↦ if k = j then y i / x j else 0
  refine ⟨A, ?_⟩
  funext i
  simp [A, Matrix.mulVec, dotProduct, hj]
```

The initial contradiction argument finds the nonzero coordinate. The matrix is then explicitly defined and supplied as the witness to the existential goal. Function extensionality reduces $Ax = y$ to its coordinates. Simplification evaluates the finite sum and cancels the nonzero denominator. Later declarations connect this concrete cyclicity predicate to `Generates` and canonical seed rank. Row 11 records the result for $n > 0$.

8 Certificate inputs and what their proofs establish

Several statements require explicit certificates. Their hypotheses encode mathematical data that a separate construction or classical theorem would supply. The Lean proof checks the implication from those data. The certificate is not a project axiom, and it is not a proof of its own existence from bare input.

8.1 The split-semisimple formula

The concrete block module consists of $d \times m$ matrices under left multiplication by $M_d(k)$. Each seed supplies at most d independent row directions. The exact count is

$$\left\lceil \frac{m}{d} \right\rceil \qquad (d > 0).$$

For a finite product of blocks, independent actions combine by the maximum. `SemisimpleCompletion` connects these row-space arguments to `seedRank`. The abstract theorem consumes an algebra equivalence to the product of full matrix algebras and a compatible semilinear module equivalence to the product of matrix modules.

The output is a rank formula for the supplied realization. This preserved 20-file component does not derive a Wedderburn decomposition from an arbitrary semisimple algebra, nor the nonsplit divisionalgebra form. The later continuation supplies that existence integration and the nonsplit formula, as described in Section 2. A reader applying foundational row 10 should still identify its supplied equivalences before quoting its conclusion.

8.2 Single-operator seed number: the upper and lower halves

For an operator T, the formalization first works with polynomial orbits $p(T)v$. It then proves a bridge to the genuine image algebra $k[T] = \text{Algebra.adjoin}_k\{T\}$. This prevents an informal change of acting algebra from being hidden in the notation for the seed number.

The primary decomposition theorem uses an attained bound R on block multiplicities. For each seed index $a < R$, it adds the ath block generator from each primary packet, inserting zero when that packet has fewer blocks. Polynomial extraction certificates recover the individual block generators from these padded sums. Thus R seeds suffice. A surjective quotient on which T acts as a scalar gives the opposite bound.

Source: `KomorebiVerification/SingleOperatorGraph.lean`.

```
theorem singleOperator_seedNumber_isLleast_of_primary_decomposition
    (T : Module.End k X) (R : ℕ)
    (multiplicity : Λ → ℕ) (eigenvalue : Λ → k)
    (blockGenerator : ∀ eigen : Λ, Fin (multiplicity eigen) → X)
    (hbound : ∀ eigen, multiplicity eigen ≤ R)
    (maxEigen : Λ) (hmax : multiplicity maxEigen = R)
    (hblocks : FamilyGenerates T
      (fun z : Σ eigen : Λ, Fin (multiplicity eigen) =>
        blockGenerator z.1 z.2))
    (hextract : ∀ (eigen : Λ) (j : Fin (multiplicity eigen)),
      ∃ p : k[X],
        aeval T p
          (paddedBlockSum R multiplicity blockGenerator
            ⟨j.val, lt_of_lt_of_le j.isLt (hbound eigen)⟩) =
          blockGenerator eigen j)
    (q : ∀ eigen : Λ, X →ₗ[k] (Fin (multiplicity eigen) → k))
    (hq : ∀ eigen, Function.Surjective (q eigen))
    (hqT : ∀ (eigen : Λ) (p : k[X]) (x : X),
      q eigen (aeval T p x) = p.eval (eigenvalue eigen) • q eigen x) :
    IsLeast {r : ℕ | SeedNumberLE T r} R := by
  constructor
  · exact ⟨paddedBlockSum R multiplicity blockGenerator,
      paddedBlockSums_generate T R multiplicity blockGenerator hbound hblocks hextract⟩
  · intro r hr
    obtain ⟨v, hv⟩ := hr
    have hle := multiplicity_le_of_generates T v (multiplicity maxEigen)
      (q maxEigen) (hq maxEigen) (eigenvalue maxEigen) (hqT maxEigen) hv
    simpa [hmax] using hle
```

The signature has three groups of data. First, `hbound` and `hmax` make R an attained maximum. Second, `hblocks` and `hextract` supply the information about orbit generation and extraction for the upper bound. Third, `q`, `hq`, and `hqT` supply the scalar quotient and polynomial intertwining law for the lower bound.

The proof constructs an `IsLeast` pair. Its first component is the padded generating family. Its second component takes an arbitrary generating family and uses the quotient at `maxEigen` to force its cardinality to be at least R. The existence of these quotient and extraction certificates is an input. The file does not independently construct general Jordan form or Chinese-remainder primary projectors from a bare matrix.

8.3 The cyclic matrix bridge

Row 14 derives the four-way cyclicity statement after two canonical-form bridges have been supplied: geometric multiplicity agrees with block multiplicity, and equality of minimal and characteristic polynomials is equivalent to one block per eigenvalue. Lean verifies their use in the equivalence chain. It does not reconstruct those bridges from arbitrary matrix data in this component.

The row is marked stronger because its compiled implication omits a separately inherited finite-label hypothesis from the manuscript. Stronger refers to this precise change in the typed statement. It does not mean that the supplied canonical-form hypotheses disappeared.

8.4 Graphs, paths, and cycles

The general graph formula takes an eigenbasis grouped by multiplicity, an attained maximum, and polynomial projectors or distinct eigenvalue labels. It does not infer an eigenbasis from the word "symmetric." The source also proves two concrete families independently: a path endpoint generates over any field, and a complex simple cycle has minimum two.

For the path, the endpoint Krylov matrix has zero entries below its diagonal and ones on the diagonal: a walk of length r cannot reach a vertex farther than r, and exactly one length-r walk reaches vertex r. Hence its determinant is one. This yields cyclicity without a numerical eigendecomposition.

For cycles, two adjacent delta vectors generate by the adjacency recurrence. Over $\mathbb{C}$, two distinct Fourier characters with the same adjacency eigenvalue provide a two-dimensional scalar quotient, supplying the lower bound. The resulting minimum-two theorem applies to the simple cycle on $n + 3$ vertices.

9 Topology and mathematical algorithms

9.1 Bounded packets and closed-hull obstructions

The analytic structure replaces linear maps by continuous linear maps. Bounded idempotents have closed ranges. Finite analysis and synthesis become continuous linear maps and then a continuous linear equivalence. The algebraic pattern of inverse laws remains visible, while continuity is carried in the types.

For a dense submodule $M \subseteq X$ and a bounded idempotent P, the closure of $P(M)$ is the full range of P. The theorem does not require that M be invariant under P. This is why the source map records the exact scope rather than summarizing the statement only as "completion preserves packets."

The topological obstruction is especially transparent:

Source: `KomorebiVerification/TopologicalCompletion.lean`.

```
/-- If a continuous packet projector annihilates an algebraic generated
hull, then it also annihilates its closed topological hull. -/
theorem closure_le_ker_of_le_ker
    (Q : X →L[𝕜] X) (H : Submodule 𝕜 X) (hH : H ≤ Q.ker) :
    H.topologicalClosure ≤ Q.ker :=
  Submodule.topologicalClosure_minimal H hH (ContinuousLinearMap.isClosed_ker Q)

/-- A nonzero unreachable packet prevents topological generation. -/
theorem topological_reachability_obstruction
```

```
    (Q : X →L[𝕜] X) (H : Submodule 𝕜 X)
    (hQ : Q ≠ 0) (hH : H ≤ Q.ker) :
    H.topologicalClosure ≠ ⊤ := by
  intro htop
  apply hQ
  apply ContinuousLinearMap.ext
  intro x
  have hx : x ∈ H.topologicalClosure := by rw [htop]; trivial
  exact (closure_le_ker_of_le_ker Q H hH hx : Q x = 0)
```

The kernel of a continuous linear map is closed. If an algebraic hull lies in that kernel, so does its closure. If the map is nonzero, the closure cannot be the whole space. This analytic theorem uses the kernel inclusion. It does not independently derive graph reachability. The algebraic and analytic arguments join at that explicit inclusion.

9.2 Dense realizations and completed presentations

A continuous surjection intertwining the stated operator families sends a dense source hull to a dense target hull. Dense range already suffices for that density conclusion. Exact equality of a source packet's image with a target packet requires the surjectivity used by the packet-range theorem. Neither conclusion needs an unspoken open-mapping argument.

For completed free presentations, the supplied map is a continuous linear isometry with dense range into a complete target. Its completion extension is an isometry with closed and dense range, hence surjective. This gives a continuous linear equivalence. Further theorems prove intertwining of supplied bounded operators and uniqueness of the continuous extension.

The bilateral shift result uses an actual reindexing representation on $\ell^2(\mathbb{Z})$, not an assumed abstract shift action. Its translates of δ_0 recover all delta vectors, whose linear span is dense. The final statement says that one is the least finite seed cardinality for dense generation by this representation. It does not establish an unrestricted cardinal-valued Laurent-algebra rank.

9.3 Orbit closure: why termination is not code extraction

The finite algorithm specification begins with the span of the seeds and repeatedly adds the images under the input operators:

$$W_0 = \operatorname{span}(S), \qquad W_{n+1} = W_n + \sum_i T_i(W_n).$$

Each iterate lies in the least invariant hull. If two successive iterates agree, the stopped subspace is invariant and contains the seeds, so it equals that hull. Finite dimensionality makes the ascending chain eventually constant.

Source: KomorebiVerification/EllipticAlgorithms.lean.

```
/-- A stopped closure pass is exactly the operator hull. -/
theorem orbitClosureIter_eq_orbitHull_of_stop
    (operators : ι → Module.End k V) (seeds : Set V) (n : ℕ)
    (hStop : orbitClosureIter operators seeds (n + 1) =
      orbitClosureIter operators seeds n) :
    orbitClosureIter operators seeds n = orbitHull operators seeds := by
```

```
  apply le_antisymm
  · exact orbitClosureIter_le_orbitHull operators seeds n
  · apply orbitHull_le operators seeds
    · exact seeds_subset_orbitClosureIter operators seeds n
    · apply invariant_of_orbitClosureStep_eq operators
      simpa [orbitClosureIter] using hStop

/--
In finite dimension the ascending closure algorithm stops after finitely many
passes, and its stopped output is exactly the least invariant seed hull.
-/
theorem orbitHull_algorithm_terminates_and_correct
    [FiniteDimensional k V]
    (operators : ι → Module.End k V) (seeds : Set V) :
    ∃ n : ℕ,
      orbitClosureIter operators seeds (n + 1) =
        orbitClosureIter operators seeds n ∧
      orbitClosureIter operators seeds n = orbitHull operators seeds := by
  have hEventually := eventuallyConst_of_isNoetherian
    (orbitClosureChain operators seeds)
  rw [Filter.eventuallyConst_atTop] at hEventually
  obtain ⟨n, hn⟩ := hEventually
  have hStop : orbitClosureIter operators seeds (n + 1) =
      orbitClosureIter operators seeds n := by
    exact hn (n + 1) (Nat.le_succ n)
  exact ⟨n, hStop,
    orbitClosureIter_eq_orbitHull_of_stop operators seeds n hStop⟩
```

The proof's termination argument uses the Noetherian property of the submodule lattice. It establishes existence of a stopping index and correctness of the stopped subspace. It does not extract a basis-maintenance loop, a decidable rank test, or a complexity bound. Those belong to an implementation refinement that would require its own proof.

The transfer-block specification is similarly exact at the mathematical level: its table consists of the blocks, summing it gives the original map, and its support predicate is block nonvanishing. It does not verify floating-point tolerances or a sparse edge-list generator.

The exact graph-rank specification takes the finite-spectrum maximum of exact eigenspace dimensions. Its correctness theorem consumes an eigenbasis, projector-polynomial data, spectrum coverage, eigenspace dimensions, and an attained maximum. Its definition is marked `noncomputable`. A Python experiment and this specification can support the same formula while remaining different forms of evidence.

9.4 The elliptic example and scalar discipline

The same source module includes a concrete algebraic model for

$$k[x, y]/(y^2 - (x^3 - x)).$$

It constructs quotient and quadratic normal-form realizations, explicit two-coefficient packet coordinates, and multiplication operators. It proves the relation $M_y^2 = M_x^3 - M_x$ and one-seed generation of the regular module.

The coefficient projectors are linear over the ground field, but not over the full coordinate algebra. The file contains explicit witnesses to that failure. This is an instructive type boundary: the acting algebra for generation may be larger than the scalar algebra over which packet maps are linear. These additional checked declarations are part of the namespace audit but are not counted as extra rows in the 35-row contract.

10 Replaying the preserved verification

The source records specify the verification environment.
Replaying the verification requires the specified compiler and matching dependency cache. These prerequisites are supplied separately from the source archive.

The following commands are quoted from the component's documented workflow, adapted to the source archive layout. They were not run as part of authoring this guide. Use a working copy for new output and keep the archived source intact.

10.1 A normal provisioned Lake environment

The standalone audit is separate from the library build. The source records give both commands and the full verifier's additional identity checks and negative controls.

Keep the manifest unchanged when reproducing the recorded verification. A successful build with another compiler or Mathlib dependency constitutes a separate verification event.

10.2 The verifier for the portable kit

The source records provide the portable kit workflow and its prerequisites.

The kit is supplied separately. The verifier checks compiler and cached dependency identities against the recorded environment. A different cache does not reproduce that run.

The verifier discovers all local Lean files, resolves their local import graph, and compiles them in dependency order to a fresh `objects/` directory. It refuses an output directory that already contains local `.olean` files. That restriction makes the local compilation claim stronger than a build that could reuse prior project objects.

Compiler invocations use:

```
lean -j1 -DwarningAsError=true \
  -DautoImplicit=false -DrelaxedAutoImplicit=false \
  -DmaxSynthPendingDepth=3
```

Each invocation has a recorded command, return code, duration, and log hash. The project modules are newly compiled in that workflow, while matching upstream cached dependencies are reused. A report of this local fresh build does not claim that Mathlib or Lean itself was independently rebuilt.

10.3 The two deliberate rejection controls

After compilation, map reconciliation, and the audit, the verifier asks Lean to reject two small examples. The first asserts a false equality and tries to prove it by decision. The second uses a placeholder, which warnings-as-errors must reject.

```
-- FalseEquality.lean: expected rejection
import Mathlib.Data.Nat.Basic
example : (0 : Nat) = 1 := by decide

-- Placeholder.lean: expected rejection
import Mathlib.Data.Nat.Basic
example : (0 : Nat) = 1 := by sorry
```

These files are negative controls under verification directories. They are not part of the main library and are not unfinished proofs in it. A nonzero exit code alone is insufficient for a passing control: the verifier also checks for the intended diagnostic and rejects timeouts, unknown imports, missing objects, and unsupported-option failures. This ensures that a broken compiler setup is not mistaken for logical rejection.

The preserved final run recorded both controls with exit code 1 and the intended diagnostic. The larger archived build-job totals elsewhere in the distribution describe other runs and must not be reused as this run's local-module count.

11 Interpreting the evidence

The source records include the foundational formalization's execution receipt.
The receipt identifies source hashes, logs and exact commands for that verification.

The preserved guide data's static check compares the delivered twenty source hashes against that record, compares all twenty-four execution-log hashes, reconciles the 35 rows and 84 distinct targets with KernelAudit.lean, and checks the saved printed axiom-report inventory. These checks passed. They establish identity and internal consistency of the delivered evidence. They do not perform another elaboration or kernel run.

11.1 The namespace-wide gate

The audit enumerates imported declarations owned by the two project namespaces. It rejects unsafe declarations, custom axiom declarations, and opaque project declarations. For every enumerated declaration it collects transitive axioms and accepts only `propext`, `Classical.choice`, and `Quot.sound`.

Source: `KomorebiVerification/KernelAudit.lean`.

```
run_cmd do
  let env ← getEnv
  let mut audited : Nat := 0
  for (name, info) in env.checked.get.constants.toList do
    if Name.isPrefixOf `Komorebi name || Name.isPrefixOf `KomorebiVerification name then
      audited := audited + 1
      if info.isUnsafe then
        throwError "unsafe project declaration: {name}"
      match info with
      | .axiomInfo _ =>
          throwError "project axiom declaration: {name}"
      | .opaqueInfo _ =>
          throwError "opaque project declaration: {name}"
      | _ => pure ()
      let axioms ← Lean.collectAxioms name
      for axiomName in axioms do
        unless axiomName == ``propext || axiomName == ``Classical.choice ||
            axiomName == ``Quot.sound do
          throwError "unexpected axiom {axiomName} in project declaration {name}"
  if audited == 0 then
    throwError "kernel audit found no project declarations"
  logInfo m!"FAIL-CLOSED KERNEL AUDIT PASS: {audited} project declarations; allowed axioms only:
  ↪ propext, Classical.choice, Quot.sound"
```

The zero-entry check matters: an audit that accidentally imports nothing must fail. The final count is a count of entries in the environment, not a count of handwritten mathematical theorems. The separate mapped-target pass resolves each of the eighty-four names, prints its type, and checks its axiom report.

The source-policy scanner also rejects a specified list of forbidden constructs after removing nested comments and string contents. This catches prohibited source forms early. Its role is complementary to the environment audit. Neither a string scan nor a metadata label alone replaces kernel acceptance.

11.2 What each layer establishes

Evidence layer	Interpretation
File hash identity	The present bytes equal the bytes identified by a record. This says nothing by itself about the correctness of those bytes.
Compiler acceptance	The declarations elaborate and their proof terms are accepted in the stated environment. The exact hypotheses remain part of the result.
Axiom audit	The enumerated declarations have only the allowed transitive foundational dependencies.
Normative scope map	The editorial bridge between prose claims and particular compiled declarations, including exclusions. This bridge still requires mathematical reading.

Evidence layer	Interpretation
Negative controls	The tested setup rejects the selected false proof and placeholder for the intended reasons. This is a useful pipeline check, not a proof of every possible failure mode.
Python tests	Finite implementation behavior checked by those tests. They are not a universal Lean theorem.
Machine record map	An explicit association between selected stable record IDs and attached evidence. A record's existence is not its proof.

The allowlisted principles are standard Lean/Mathlib foundations. The audit does not claim axiom-free mathematics in an absolute sense. It also does not prove historical priority, performance claims, or the absence of defects in the compiler, trusted kernel, or dependency cache. These are the ordinary boundaries of the recorded verification method, not gaps secretly filled by the count of mapped declarations.

12 The expanded chapter and computational crosswalk

The computational companion carries a separate seven-source-file formalization and an explicit eight-record map. That tranche shares some material on finite projectors with the larger twenty-file formal component, but it is a separate packaged evidence scope.

The computational source and its 108 stable statement records include the two explicit nonzero-packet hypotheses required by the corresponding corollaries. These source hypotheses do not enlarge the attached eight-record Lean map. The following eight mappings describe the attached machine tranche's own evidence allocation.

Stable machine record	Attached Lean subject
`PSC-I-D-PACKET-SYSTEM`	Finite linear projector structure.
`PSC-I-D-ANALYSIS-SYNTHESIS`	Range packet space and the two coordinate maps.
`PSC-I-T-EXACT-PACKET-COORDINATES`	Range-restricted linear equivalence.
`PSC-I-C-PACKETWISE-EQUALITY`	Equality detected by the finite complete family.
`PSC-I-D-TRANSFER-BLOCK`	Definition of the typed block.
`PSC-I-T-OPERATOR-RECONSTRUCTION`	Pointwise finite reconstruction.
`PSC-I-T-BLOCK-MULTIPLICATION`	Pointwise finite composition identity.
`PSC-I-T-SUPPORT-PATTERN-ALGEBRA`	Transitive ordinary-matrix support is multiplicatively closed. This is not the whole expanded iff criterion.

The larger component can inform a broader semantic crosswalk. Rows 1–2 concern free presentations, rows 3–9 concern seed-rank operations, rows 10–14 concern split-semisimple and operator formulas, rows 15–29 concern packets, blocks, support, and transport. Rows 30–35 concern the supplementary worked models and algorithm specifications. This is a navigation aid. Where the expanded statement adds a division-ring generalization, an infinite family, an existence theorem, or a different categorical interface, the old scoped result must not be silently relabeled as a proof of the expanded statement.

The source archive's `COMPANION_CROSSWALK.json` is the navigation record for the series, and `formal/coverage/COVERAGE_108.json` is the current formal correspondence map. The machine-readable

`generated/theorem_map.json` accompanying this guide preserves all thirty-five contract scopes and their exact declaration lists. Unmapped expanded statements remain unmapped until an explicit type-level comparison and accepted proof support a new mapping.

13 How to extend the formal development

A useful extension adds a theorem with a clear source statement, a type that exposes its genuine inputs, and evidence that can be replayed. The easiest way to lose that clarity is to hide the desired conclusion inside an unnamed certificate and then announce that the original unconditional theorem has been formalized.

1. Identify the exact new statement and its relation to the current contract. Decide whether it is a new row, a stronger replacement, or a helper used by an existing row. 2. Inspect the current types before designing another abstraction. For finite packets, reuse range subtypes and actual Hom blocks. For topology, use continuous linear maps and state completeness where it is needed. 3. Separate supplied data from proved consequences. A new canonical-form constructor should have its own existence theorem, not only a theorem taking a canonical-form certificate as an argument. 4. Prove or import the bridge from the new representation to the manuscript's object. A result about a bespoke rank predicate needs a theorem relating it to canonical seed rank before it can replace the existing formula. 5. Compile the changed modules in the pinned environment, update the normative map deliberately, and update the audit inventory to exactly the desired mapped surface. 6. Run the explicit audit and the full verifier with a new output directory. Preserve prior records and retain source hashes and execution records for each run. 7. Review the new source-to-prose correspondence. Kernel acceptance proves the typed theorem. The review establishes that this is the intended theorem and that its public summary retains the hypotheses.

13.1 Continuing the correspondence work

The continuation has established the primary decomposition and quotient constructions, the geometric formula, scalar extension, the six-way cyclic criterion, module-top reduction, and the nonsplit semisimple formula. Further work is determined by the current coverage map in Section 2. These established results must not be treated as missing certificate constructors. For graph rank, an implementation refinement would relate executable spectral or algebraic data to the exact certificates, with a separate proof of each algorithmic step. For the expanded packet theory, countable unconditional sums require a topology and convergence interface different from the finite `Fintype` API.

These are different tasks from increasing a declaration count. Each closes a named input boundary or extends an exact scope. A completed extension should say which boundary has changed and exhibit the theorem that changed it.

13.2 Troubleshooting a replay

An unknown module or missing object indicates an environment or import-path problem, not a false mathematical theorem. A pin mismatch means the cached dependency state is different from the recorded source. A refusal of nonempty output means the verifier is protecting the local rebuild claim.

Choose a new output directory. A warning-as-error failure requires reading the diagnostic before deciding whether the source, compiler, or style policy changed.

A mapped-name failure after a rename is not solved by silently removing the map entry. Either preserve the public declaration name, or update the map and record the new correspondence. A new unexpected axiom requires tracing its transitive dependency before acceptance. A changed source hash is expected after a deliberate edit, but it ends byte-identity with the old accepted source and requires fresh evidence for the changed file.

14 The complete 35-row contract

This section reproduces the normative map's statement titles, declaration names, exact scopes, and exclusions. The numbering belongs to the corrected journal formalization contract. It is not the numbering of the expanded mathematical text or the stable inventory of machine records.

"Exact" means exact at the scope printed in the row. "Stronger" means the mapped declaration removes a specified hypothesis, treats a more general typed setting, or includes an additional endpoint. Rows 10, 13, 14, 32, and 35 are certificate-explicit. The precise certificate boundaries appear both here and in Section 8.

The long qualified names are intentionally retained. They allow a reader or a tool to search the source, issue `#check`, and compare the printed type without guessing a namespace.

14.1 Row 1: Universal one-seed presentation

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankBasics.oneSeedQuotientEquivRange`

`Komorebi.SeedRankBasics.oneSeedQuotientEquivRange_apply_mk`

`Komorebi.SeedRankBasics.generates_singleton_iff_surjective`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.completionEquivOfIsometryDense`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.completionEquivOfIsometryDense_intertwines`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.completionEquivOfIsometryDense_unique`

Exact scope. For the module map `a↦a•s`, the quotient by its kernel is linearly equivalent to its exact range; the glue theorem verifies that the class of `a` maps to `a•s`; and singleton generation is equivalent to surjectivity. If the induced map from the normed quotient is supplied as a continuous linear isometry with dense range into a Banach target, it extends uniquely to a continuous linear equivalence from the completion and intertwines supplied bounded operators.

Boundary. No norm, isometry, density, or bounded extension is constructed automatically on the algebraic quotient. The annihilator is only the typed left submodule/left-ideal object; no two-sided ideal claim is made.

14.2 Row 2: Multi-seed presentation

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankBasics.multiSeedQuotientEquivRange`

`Komorebi.SeedRankBasics.multiSeedQuotientEquivRange_apply_mk`

`Komorebi.SeedRankBasics.multiSeedQuotientEquiv`

`Komorebi.SeedRankOperations.seedFinrank_isLeast_free_surjection`

Exact scope. An arbitrary Finsupp-indexed family has the free-module quotient presentation; the glue theorem verifies that a coefficient family maps to its finite linear combination; and a generating family

gives the quotient onto the whole module. For a finite module, `seedFinrank` is the least `n` admitting a surjection from $\texttt{Finn}\to_0\texttt{A}$.

Boundary. The natural-valued minimum is asserted only under module finiteness. No finite basis or executable presentation matrix is produced.

14.3 Row 3: Cyclic characterization

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.cyclic_iff_seedRank_eq_one`

Exact scope. For every nontrivial module over a semiring, unrestricted cardinal-valued seed rank equals one exactly when a cyclic vector exists.

Boundary. The zero module is excluded here and is governed separately by `seedRank_eq_zero_iff`.

14.4 Row 4: Quotient monotonicity

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.seedRank_le_of_surjective_semilinear`

Exact scope. For an arbitrary ring homomorphism `σ:A→+*B`, a surjective σ-semilinear map gives `seedRankBN≤seedRankAM`; σ need not be onto.

Boundary. No equality is claimed from surjectivity of the module map alone.

14.5 Row 5: Image-algebra invariance

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.seedRank_imageAlgebra_eq`

Exact scope. For a `k`-algebra homomorphism $\texttt{φ:A}\to_{\texttt{a}}\texttt{[k]B}$ and a bijective φ-semilinear map, the target action is restricted literally to `φ.range`, and its seed rank equals the source seed rank.

Boundary. Equality for the full target algebra `B` requires φ to be surjective; it is not inferred otherwise.

14.6 Row 6: Shared direct-sum bound

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.seedRank_prod_le_add`

Exact scope. One semiring acting diagonally on `M×N` gives the cardinal inequality `seedRankA(M×N)≤ seedRankAM+seedRankAN`.

Boundary. This is only an upper bound; it does not identify when equality holds.

14.7 Row 7: Independent direct-sum formula

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.seedRank_prod_productAlgebra`

Exact scope. With the coordinatewise action of `A×B` on `M×N`, the cardinal seed rank is the maximum of the two component ranks.

Boundary. This formula is not for a single shared diagonal action.

14.8 Row 8: Tensor-product bound

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.tensorOperator_smul_tmul`

`Komorebi.SeedRankOperations.seedRank_tensorProduct_le_mul`

Exact scope. Over a commutative semiring `k`, with the stated algebra/module, scalar-tower, and commuting-action laws, the action of `A⊗[k]B` on `X⊗[k]Y` satisfies the cardinal product bound and realizes `(a⊗b)(x⊗y)=ax⊗by`.

Boundary. No equality, completed tensor product, or alternative topological tensor norm is claimed.

14.9 Row 9: Scalar quotient obstruction

Contract classification: Exact.

Declaration targets

`Komorebi.SeedRankOperations.finrank_le_seedRank_of_scalar_quotient`

Exact scope. For a field `k`, a character $\chi : A \to_a [k]k$, a finite-dimensional scalar module `Y`, and a surjective χ-semilinear map `X→Y`, `finrankkY` is a cardinal lower bound for `seedRankAX`.

Boundary. The target action must factor through the stated character; a general quotient does not yield the dimension bound.

14.10 Row 10: Semisimple multiplicity formula

Contract classification: Exact.

Declaration targets

`Komorebi.SemisimpleCompletion.semisimple_seedRank_eq_maxCeilDiv_of_equivalences`

`Komorebi.SemisimpleCompletion.splitProduct_seedRank_eq_maxCeilDiv`

Exact scope. For a finite product, positive block sizes `di`, a supplied algebra equivalence to the product of full matrix algebras, and a compatible supplied semilinear module equivalence to the product of `di×mi` matrix modules, seed rank is `max_i⌈mi/di⌉`.

Boundary. Lean does not prove the existence of the Wedderburn/module decompositions, nor the nonsplit division-algebra variant.

14.11 Row 11: Full matrix algebra

Contract classification: Exact.

Declaration targets

`Komorebi.SemisimpleCompletion.fullMatrix_every_nonzero_vector_generates`

`Komorebi.SemisimpleCompletion.fullMatrix_seedRank_eq_one`

Exact scope. Over a field and for `n>0`, every nonzero vector in the natural `M_n(k)`-module generates and the canonical cardinal seed rank is one.

Boundary. No statement is made for `n=0` or for a smaller acting subalgebra.

14.12 Row 12: Multiplicity obstruction

Contract classification: Exact.

Declaration targets

`Komorebi.SemisimpleCompletion.multiplicity_obstruction`

`Komorebi.SemisimpleCompletion.multiplicity_seedRank_formula`

Exact scope. For `d>0`, the left action of `M_d(k)` on the `d×m` matrix realization of `m` copies has seed rank `⌈m/d⌉`; every generating `r`-family satisfies `⌈m/d⌉≤r`.

Boundary. No general simple-module formula over a nontrivial endomorphism division algebra is included.

14.13 Row 13: Single-operator seed rank

Contract classification: Exact.

Declaration targets

`Komorebi.SingleOperatorGraph.singleOperator_seedNumber_isLeast_of_primary_decomposition`

`Komorebi.SingleOperatorGraph.singleOperator_operatorAlgebra_seedRanks_eq_of_primary_decomposition`

Exact scope. Certificate-explicit: a finite label family supplies block generators, an attained bound `R`, joint polynomial-orbit generation, polynomial extraction from padded seeds, and surjective scalar quotient maps with the evaluation intertwining law. Then `R` is the least polynomial seed number and both `seedFinrank` and cardinal `seedRank` over the genuine image algebra `k[T]=Algebra.adjoink{T}` equal `R`.

Boundary. Lean does not construct a primary/Jordan decomposition, Chinese-remainder projectors, or identify the supplied multiplicities with `dimker(T-λI)` from bare hypotheses.

14.14 Row 14: Cyclic matrix criterion

Contract classification: Stronger.

Declaration targets

`Komorebi.SingleOperatorGraph.cyclic_matrix_criterion_of_primary_decomposition`

`Komorebi.SingleOperatorGraph.cyclic_matrix_four_way_iff_of_primary_decomposition`

Exact scope. Certificate-explicit and finite-dimensional: from an exact positive seed-number certificate, a supplied equality between geometric and block multiplicities, and the supplied minpoly/charpoly canonical-form bridge, Lean proves the four-way equivalence among cyclicity, all geometric multiplicities one, one block per eigenvalue, and `minpoly=charpoly`. The compiled implication does not require the manuscript's separately inherited finite-label hypothesis.

Boundary. Neither canonical-form bridge is derived by Lean from an arbitrary matrix; the theorem verifies the equivalence after those bridges are supplied.

14.15 Row 15: Exact packet coordinates

Contract classification: Exact.

Declaration targets

`Komorebi.Foundation.LinearProjectorFamily.rangeSynthesis_rangeAnalysis`

`Komorebi.Foundation.LinearProjectorFamily.rangeAnalysis_rangeSynthesis`

`Komorebi.Foundation.LinearProjectorFamily.exactPacketCoordinates`

Exact scope. A finite complete, idempotent, pairwise-annihilating linear projector family gives mutually inverse analysis and finite-sum synthesis maps between the module and the independently defined dependent product of projector ranges.

Boundary. No infinite summation or topological convergence assertion is present in this algebraic declaration.

14.16 Row 16: Packetwise equality

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.Packets.eq_iff_projectors_eq`

Exact scope. For a finite linear projector family, `x=y` iff every projected packet coordinate agrees.

Boundary. Finiteness and completeness of the packet family remain essential hypotheses.

14.17 Row 17: Topological packet isomorphism

Contract classification: Exact.

Declaration targets

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.topological_packet_isomorphism`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.packetEquiv`

Exact scope. For a finite bounded projector family on a Banach space over a nontrivially normed field, all packet ranges are closed and analysis/synthesis form a continuous linear equivalence with the canonical finite dependent-product norm.

Boundary. No infinite packet family, unbounded projector, or assertion about every arbitrarily chosen equivalent product norm is included.

14.18 Row 18: Completion preserves a finite packet split

Contract classification: Exact.

Declaration targets

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.idempotentRange_eq_closure_map`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.packetRange_eq_closure_map`

Exact scope. If `M` is a dense submodule of a normed space and `P_i` is any bounded idempotent, the closure of `P_i(M)` equals the full range `P_i(X)`. The packet-family theorem is the direct specialization. No invariance of `M`, surrounding finite packet family, or completeness of `X` is required by the manuscript or the compiled declaration.

Boundary. Boundedness is required; no unbounded-projector or infinite conditional-sum claim is made.

14.19 Row 19: Exact operator reconstruction

Contract classification: Stronger.

Declaration targets

`Komorebi.Foundation.transferBlock_reconstruction_linearMap`

`Komorebi.Foundation.transferBlock_reconstruction_end`

Exact scope. The sum of all transfer blocks reconstructs an arbitrary linear map between modules carrying possibly different finite projector families; the endomorphism formula is the manuscript instance.

Boundary. This is algebraic finite reconstruction, not convergence of an infinite block expansion.

14.20 Row 20: Packet matrix multiplication

Contract classification: Stronger.

Declaration targets

`Komorebi.Foundation.transferBlock_comp_linearMap`

`Komorebi.Foundation.transferBlock_comp_end`

`Komorebi.Foundation.end_comp_eq_sum_transferBlock_comps`

Exact scope. The block of a composite is the finite sum of composite blocks, even for three different source/intermediate/target modules and projector families; the single-space operator identity is the manuscript instance.

Boundary. No infinite matrix multiplication or operator-norm convergence claim is made.

14.21 Row 21: Packet powers

Contract classification: Stronger.

Declaration targets

`Komorebi.PacketNaturalityCompletion.Packets.transferBlock_pow_eq_packetPathSum`

Exact scope. For every natural power `m`, including `m=0`, the transfer block of `T^m` equals the recursively defined finite sum over length-`m` packet paths.

Boundary. Only finite powers and finite packet sets are covered.

14.22 Row 22: Adjoint reversal

Contract classification: Stronger.

Declaration targets

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.adjoint_transferBlock`

Exact scope. On a complete real or complex inner-product space, bounded self-adjoint maps satisfy `(PβT Pα)†=PαT†Pβ`. This is stronger than the packet-projector formulation because no idempotence, mutual annihilation, or completeness of a projector family is used.

Boundary. Self-adjointness of both outer maps and boundedness/completeness needed by the adjoint API are explicit.

14.23 Row 23: Algebraic full block representation

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.FullBlocks.fullBlockAlgEquiv`

`Komorebi.PacketNaturalityCompletion.FullBlocks.restrictBlocks_synthesizeBlocks`

`Komorebi.PacketNaturalityCompletion.FullBlocks.synthesizeBlocks_restrictBlocks`

`Komorebi.PacketNaturalityCompletion.FullBlocks.packetBlockAlgebra_mul_eq_blockMul`

Exact scope. A finite projector family over a commutative semiring identifies ambient endomorphisms with the genuine dependent family of all packet-to-packet linear maps; transported multiplication is exactly the output-first block product.

Boundary. This is the algebraic endomorphism algebra. Arbitrary algebraic blocks are not asserted to be bounded in a Banach realization.

14.24 Row 24: Dimension count

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.FullBlocks.end_finrank_eq_packet_sum_eq_square`

`Komorebi.PacketNaturalityCompletion.FullBlocks.finrank_eq_sum_packet_finrank`

Exact scope. For a finite-dimensional vector space over a field, endomorphism dimension equals the double sum of packet-dimension products, which equals the square of the packet-dimension sum.

Boundary. No dimension statement for infinite-dimensional or topological dimensions is made.

14.25 Row 25: Algebraic packet support pattern

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.FullBlocks.identity_blocks_supported_iff_reflexive`

`Komorebi.PacketNaturalityCompletion.FullBlocks.blockMulClosed_iff_transitive`

Exact scope. Over a field, with every packet range nontrivial and arbitrary packet-Hom blocks allowed, the identity is supported exactly for reflexive relations and universal multiplication closure holds exactly for transitive relations.

Boundary. The converse is not promoted to arbitrary bounded/continuous categories without enough nonzero composable maps.

14.26 Row 26: Reachability obstruction

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.Reachability.algebraicOrbitHull_le_reachable_support`
`Komorebi.PacketNaturalityCompletion.Reachability.unreachable_nonzero_packet_obstruction`
`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.closure_le_ker_of_le_ker`
`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.topological_reachability_obstruction`

Exact scope. Word orbits from packet-supported seeds stay in the reflexive-transitive reachable support, and a nontrivial unreachable packet makes the algebraic hull proper. Separately, if the algebraic hull lies in the kernel of a nonzero bounded projector, its closure also lies there and is not dense.

Boundary. The analytic theorem consumes the algebraically proved kernel inclusion; it does not infer graph reachability on its own. No general source-count lower bound is claimed for mixed-packet seeds.

14.27 Row 27: Coordinate transport

Contract classification: Exact.

Declaration targets

`Komorebi.Foundation.LinearProjectorFamily.conjugate`
`Komorebi.Foundation.conjugate_transferBlock`
`Komorebi.PacketNaturalityCompletion.Reachability.conjugate_transferEdge_iff`
`Komorebi.PacketNaturalityCompletion.Reachability.conjugate_reachableFrom_iff`

Exact scope. A linear equivalence transports a finite projector family and conjugates every transfer block. For the conjugated operator family, every zero/nonzero edge and every reflexive-transitive reachability statement is preserved.

Boundary. The full invariance assertion requires invertible coordinate transport, not an arbitrary non-injective map.

14.28 Row 28: Naturality of generation and packets

Contract classification: Stronger.

Declaration targets

`Komorebi.PacketNaturalityCompletion.StrictNaturality.generatedHull_natural`
`Komorebi.PacketNaturalityCompletion.StrictNaturality.strict_naturality_package`
`Komorebi.PacketNaturalityCompletion.StrictNaturality.strict_seedRank_le`
`Komorebi.PacketNaturalityCompletion.StrictNaturality.strict_seedRank_eq_of_bijective`
`Komorebi.PacketNaturalityCompletion.StrictNaturality.full_transfer_edge_zero_pattern_correspondence`

Exact scope. For a strict morphism over a field, generated hulls are natural over the exact image algebra, packet analysis and blocks intertwine, a surjective realization gives the cardinal rank inequality, bijective realization/algebra maps give equality, and bijective packet transport gives complete edge/zero-pattern correspondence. These conclusions do not require the manuscript's finite-dimensional hypothesis.

Boundary. Packet merging is excluded by the injective label map. Full target-algebra rank equality and full target edge coverage require the stated surjectivity/bijectivity hypotheses.

14.29 Row 29: Dense realization theorem

Contract classification: Exact.

Declaration targets

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.dense_operatorHull_image`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.map_packetRange_eq`

Exact scope. For normed spaces and bounded operator families, a continuous surjection that sends each source operator into the target family and intertwines it sends a dense source operator hull to a dense target hull. A surjective map intertwining bounded projectors maps the source range onto the target range.

Boundary. No algebra-homomorphism law, openness, or completeness is needed or concluded. Surjectivity is sufficient for the stated conclusions; dense range alone suffices for the density conclusion, but not for exact range equality. The algebraic hulls need not be equal as sets. The range equality itself needs only an intertwining surjective map; it does not require idempotence.

14.30 Row 30: Typed reachability certificate

Contract classification: Exact.

Declaration targets

`Komorebi.PacketNaturalityCompletion.TypedQuiver.typed_reachability_certificate`

Exact scope. For the displayed `U=k`^{2}, `V=k`^{3} quiver realization, Lean proves `BA=id`, exhibits the nonzero missing direction in `kerB` but outside `rangeA`, proves every `U`-supported family insufficient, and proves that the displayed two seeds generate.

Boundary. This is a certificate for the stated finite model, not a classification or minimum-seed theorem for arbitrary quivers.

14.31 Row 31: Bilateral-shift seed

Contract classification: Exact.

Declaration targets

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.BilateralShiftRepresentation.concretePower_apply`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.concreteBilateralShift_topologicallyCyclic`

`KomorebiVerification.TopologicalCompletion.BoundedProjectorFamily.concreteBilateralShift_seedCardinality_isLeast`

Exact scope. For the explicit reindexing representation on ℓ^2`(ℤ,𝕂), (S_mx)(n)=x(n-m)`, the orbit of δ_0 is dense and one is the least finite cardinality of a densely generating seed set.

Boundary. This does not define or equate an unrestricted cardinal-valued Laurent-algebra rank, and it does not formalize the later residue-packet discussion.

14.32 Row 32: Graph seed rank, path, and cycle

Contract classification: Exact.

Declaration targets

`Komorebi.SingleOperatorGraph.graph_seedNumber_eq_max_eigenspace_multiplicity`

`Komorebi.SingleOperatorGraph.graph_seedNumber_eq_max_eigenspace_multiplicity_of_distinct`

`Komorebi.SingleOperatorGraph.operatorAlgebra_seedFinrank_eq_of_seedNumber_isLeast`

`Komorebi.SingleOperatorGraph.operatorAlgebra_seedRank_eq_of_seedNumber_isLeast`

`Komorebi.SingleOperatorGraph.path_endpoint_seedNumber_isLeast_one`

`Komorebi.SingleOperatorGraph.complex_cycle_seedNumber_isLeast_two`

Exact scope. Certificate-explicit general formula: a finite eigenbasis grouped by multiplicity, an attained maximum `R`, and either explicit polynomial projectors or distinct eigenvalue labels prove that `R` is the least operator-seed number; the bridge declarations identify it with both canonical ranks over `k[A]`. Independently, every nonempty path (`n+1` vertices) has minimum one from an endpoint over any field, and every complex simple cycle (`n+3` vertices) has minimum two.

Boundary. Lean does not derive the eigenbasis from the bare assertion that a matrix is symmetric, nor identify an arbitrary graph's algebraic multiplicities without supplied spectral data. The unqualified spectral theorem remains cited classical input.

14.33 Row 33: Orbit-hull algorithm

Contract classification: Exact.

Declaration targets

`Komorebi.FiniteAlgorithms.orbitHull_algorithm_terminates_and_correct`

`Komorebi.FiniteAlgorithms.orbitClosureIter_eq_orbitHull_of_stop`

Exact scope. For a finite operator index on a finite-dimensional vector space over a field and any seed set, the ascending subspace iteration stops after finitely many passes, and the stopped subspace equals the least invariant subspace containing the seeds.

Boundary. The proof is a mathematical termination/correctness specification; it does not extract the paper's basis-maintenance/rank-test implementation or prove a complexity bound.

14.34 Row 34: Transfer-block algorithm

Contract classification: Exact.

Declaration targets

`Komorebi.FiniteAlgorithms.transferBlockTable`

`Komorebi.FiniteAlgorithms.assembleTransferBlockTable`

`Komorebi.FiniteAlgorithms.transferBlock_algorithm_correct`

`Komorebi.FiniteAlgorithms.transferGraph_iff_block_ne_zero`

Exact scope. For finite source and target projector families, the table is exactly all transfer blocks, summing it reconstructs the input linear map, and its support predicate is exactly block nonvanishing.

Boundary. No decidable zero-test, sparse edge-list extractor, floating-point tolerance, or runtime bound is verified.

14.35 Row 35: Exact graph seed-rank algorithm

Contract classification: Exact.

Declaration targets

`Komorebi.FiniteAlgorithms.exactGraphSeedRank`

`Komorebi.FiniteAlgorithms.exactGraphSeedRank_isLeast`

Exact scope. For a finite square matrix over a field, supplied exact eigenbasis, polynomial-projector, spectrum-coverage, eigenspace-dimension, and attained-maximum certificates prove that the finite-spectrum maximum computed by `exactGraphSeedRank` is the least operator-seed number.

Boundary. `exactGraphSeedRank` is noncomputable in Lean. No verified eigensolver, symbolic factorization routine, symmetry-to-diagonalization step, code extraction, or performance guarantee is claimed.

Projector Resolved Seed Calculus II

The formal companion to Chapter II explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 1722

Return to Chapter II

How to use this volume

This formal companion separates the chapter-wide correspondence from the foundational 25-row contract. The overview records exact source scope and verification evidence. The tutorial explains the foundational source architecture, representative proofs, and evidence with their original boundaries. **Formal correspondence.** The correspondence records 81 source items, including

53 proof-level items. Its status is FINAL_SOURCE_BOUND_CORRESPONDENCE_53_OF_53. The tables retain the recorded states and verification checks. The inherited 25-row map remains a separate foundational contract.

53 of 53 proof-level source records have kernel checked, semantically accepted scope. No partial or pending proof records are listed in this correspondence. These are source-record counts under their stated hypotheses, not counts of independent discoveries.

1 Formal scope and evidence

The Lean project is formal/current/. Its correspondence records exact statement and proof text, source hashes, declaration targets and execution evidence. The source records include the complete machine-readable correspondence.

Formal correspondence. The correspondence records 81 source items, including 53 proof-level items. Its status is FINAL_SOURCE_BOUND_CORRESPONDENCE_53_OF_53. The tables retain the recorded states and verification checks. The inherited 25-row map remains a separate foundational contract.

53 of 53 proof-level source records have kernel checked, semantically accepted scope. No partial or pending proof records are listed in this correspondence. These are source-record counts under their stated hypotheses, not counts of independent discoveries.

1.1 Proof inventory

Recorded status	Items
Kernel checked, scope accepted	53

Exact statement comparison, semantic acceptance, aggregate compilation, and kernel replay provide distinct kinds of evidence. The table records the results of those checks. Typesetting does not repeat them.

1.2 Verification checks

Gate	Recorded state
exact source aggregate compile	PASS
owned kernel type body replay	PASS
standard axiom and runtime helper closure	PASS
manuscript semantic acceptance	53_COMPLETE_SCOPES_ACCEPTED

1.3 The rectifiable analytic endpoints

The rectifiable analytic endpoints use a representation whose velocity length is bounded by the original geometric length. This inequality supplies the manuscript estimate. Equality of the two lengths is not required. The accepted correspondence identifies the exact canonical endpoints and their hypotheses.

1.4 Statement map

The following table lists the proof-level source records. The correspondence gives exact declaration names, statement and proof hashes, hypotheses and inherited evidence. Mathematical definitions marked `noncomputable` do not supply an executable solver.

Source label	Recorded status
`thm:children_transitivity`	Kernel checked, scope accepted

Source label	Recorded status
thm:coordinate_coherence	Kernel checked, scope accepted
cor:orthogonal_isometry	Kernel checked, scope accepted
thm:block_refinement	Kernel checked, scope accepted
cor:block_support_refinement	Kernel checked, scope accepted
thm:common_refinement	Kernel checked, scope accepted
prop:coarsest_common_refinement	Kernel checked, scope accepted
prop:residue_algebra	Kernel checked, scope accepted
thm:root-of-unity	Kernel checked, scope accepted
thm:character_realization	Kernel checked, scope accepted
thm:residue_refinement	Kernel checked, scope accepted
thm:crt_factorization	Kernel checked, scope accepted
cor:dyadic_odd_coordinates	Kernel checked, scope accepted
thm:weighted_inverse	Kernel checked, scope accepted
thm:block_path_formula	Kernel checked, scope accepted
cor:minimum_path_weight	Kernel checked, scope accepted
cor:acyclic_inverse	Kernel checked, scope accepted
cor:inverse_stability	Kernel checked, scope accepted
cor:finite_precision	Kernel checked, scope accepted
thm:weighted-inverse	Kernel checked, scope accepted
cor:finite-precision	Kernel checked, scope accepted
thm:transfer_rationality	Kernel checked, scope accepted
cor:path_recurrence	Kernel checked, scope accepted
cor:multigraded_series	Kernel checked, scope accepted
thm:transfer-zeta	Kernel checked, scope accepted
cor:zeta-recurrence	Kernel checked, scope accepted
thm:bezout_projector	Kernel checked, scope accepted
thm:several_projectors	Kernel checked, scope accepted
cor:resultant_bound	Kernel checked, scope accepted
thm:bezout-projector	Kernel checked, scope accepted
thm:precision-delay	Kernel checked, scope accepted
thm:gluing_weight	Kernel checked, scope accepted
cor:length_and_weight	Kernel checked, scope accepted
cor:binary_gluing	Kernel checked, scope accepted
cor:integral_split	Kernel checked, scope accepted
thm:gluing_refinement	Kernel checked, scope accepted

Source label	Recorded status
thm:precision_delay	Kernel checked, scope accepted
prop:sign_projectors	Kernel checked, scope accepted
prop:character_integrality	Kernel checked, scope accepted
prop:dyadic-sign	Kernel checked, scope accepted
prop:weight_from_inclusion	Kernel checked, scope accepted
thm:smith_congruence	Kernel checked, scope accepted
cor:smith_obstruction	Kernel checked, scope accepted
thm:smith-solve	Kernel checked, scope accepted
thm:power_defect	Kernel checked, scope accepted
cor:power_norm	Kernel checked, scope accepted
cor:defect-norm	Kernel checked, scope accepted
cor:polynomial_defect	Kernel checked, scope accepted
thm:resolvent_defect	Kernel checked, scope accepted
thm:holomorphic_defect	Kernel checked, scope accepted
thm:exponential_defect	Kernel checked, scope accepted
thm:multilevel_defect	Kernel checked, scope accepted
cor:exact_compatibility	Kernel checked, scope accepted

1.5 Reading the formal walkthrough

The following tutorial explains the foundational 25-row Lean contract. Its counts, excerpts, and certificate boundaries refer to the archived source identified in the orientation. Use the chapter-wide map above for the broader constructions and their verification evidence. Each report retains its own source identity and evidence scope.

2 From original contours to canonical values

The analytic development follows the contours that occur in the mathematical text. A contour is a continuous complex path of bounded variation on the closed unit interval. A cycle is a finite family of closed contours. An occurrence in that family contributes once to every sum, so a repeated contour retains its multiplicity. The construction permits constant paths, retracing, and an empty family.

The proof has three distinct tasks. It identifies the classical contour integral, controls the original geometric length, and proves that admissible contours give the same functional calculus value. Each task has an explicit endpoint in the current sources. The identity of a chosen integral is kept separate from independence of that choice.

2.1 The original tagged integral

For a path γ and a continuous function F on its trace with values in a complex Banach space, the tagged sums are

$$S(\gamma, F; P, \tau) = \sum_{j=0}^{m-1} (\gamma(t_{j+1}) - \gamma(t_j)) F(\gamma(\tau_j)), \qquad t_j \le \tau_j \le t_{j+1}.$$

The factor on the left acts by complex scalar multiplication. The partition is monotone, begins at 0, and ends at 1. Repeated partition points are allowed. The definition of the integral quantifies over every such tagged partition with sufficiently small mesh. Arbitrarily fine partitions are constructed, and the value is proved unique. The definition therefore expresses an actual limit of the original path increments.

For an absolutely continuous path, the difference between a tagged increment and its velocity integral is an integral of the oscillation of F along that subinterval. Its norm is bounded by that oscillation times the integral of the speed. Summing over the partition and using uniform continuity on the compact trace proves convergence of all the tagged sums to the Bochner velocity integral.

An arbitrary continuous rectifiable path need not be absolutely continuous in its original parameter. The project constructs its normalized arclength path and a continuous monotone map from the original parameter interval. Tagged partitions are transported through this map. Their increments and tagged function values agree exactly. Uniform continuity controls the transported mesh, including flat portions of the parameter map. The zero length case is handled separately. This proves equality between the original contour integral and the represented velocity integral without assuming a comparison certificate.

The source `ReleaseRectifiableRSError.lean` contains the tagged definition, existence and uniqueness results, monotone transport, and the finite cycle comparison. In particular, the original winding expression obtained by integrating $(z-w)^{-1}$ agrees with the represented winding whenever w avoids the trace.

2.2 The geometric length used in the estimate

Write $L(\gamma)$ for the total variation of the original path. The normalized arclength path is Lipschitz with constant $L(\gamma)$. Its coordinate velocity consequently satisfies

$$\int_0^1 \|v_\gamma(t)\|\, dt \le L(\gamma).$$

For a cycle, these inequalities add over its occurrences. This is the inequality needed by the manuscript. The proof does not require an additional theorem asserting equality between integrated speed and variation.

Combining this inequality with the actual cycle integral estimate gives

$$\left\| \int_\Gamma F(z)\,dz \right\| \le \operatorname{length}(\Gamma) \sup_{z\in\Gamma} \|F(z)\|,$$

where the length on the right is the original geometric length with multiplicities. Each supremum of norms is defined to be zero when its trace is empty. The separate supremum factors in Theorem 11.6 are retained by `ReleaseRectifiableDefect.lean`. They are not replaced by an unspecified product bound.

2.3 General Cauchy comparison

The represented cycle has integrable velocities. Its curves are absolutely continuous, and composition with a holomorphic scalar function on a neighborhood of the compact trace is absolutely continuous. The chain rule and the fundamental theorem of calculus therefore give

$$\int_\Gamma G'(z)\,dz = 0$$

for a supplied holomorphic primitive G. This scalar primitive result also proves local constancy of the actual normalized winding integral away from the trace. Applying it to an inverse kernel on an enclosing ball proves that winding vanishes sufficiently far from the trace.

For a holomorphic function F with values in a complex Banach space on an arbitrary open set U, define

$$H_{1,\Gamma}(w) = \int_\Gamma [w,z]F\,dz, \qquad H_{2,\Gamma}(w) = \int_\Gamma (z-w)^{-1}F(z)\,dz.$$

The divided difference $[w,z]F$ takes the value $F'(w)$ on the diagonal. The first function is holomorphic on U. The second is holomorphic away from the trace. Compact trace bounds justify differentiation under the actual Banach integral in both cases.

Suppose that two cycles Γ and Λ lie in U and have equal winding outside U. Use $H_{1,\Gamma} - H_{1,\Lambda}$ on U and $H_{2,\Gamma} - H_{2,\Lambda}$ outside U. At an exterior point, both traces are absent from a neighborhood. Local constancy of winding and the divided difference identity make the two expressions agree there. The resulting function is entire, even when U is disconnected or unbounded. Far from the traces it tends to zero. Banach valued Liouville then makes it identically zero.

To recover the integral of F, replace it by $(z-a)F(z)$ for any $a \in U$. Its divided difference at a equals $F(z)$, including at $z = a$. Thus no point outside the traces has to be chosen. The empty domain case is handled directly by the zero bound on an empty trace. The resulting theorem is

$$\int_\Gamma F(z)\,dz = \int_\Lambda F(z)\,dz.$$

Its only geometric inputs are the actual finite cycles and equality of winding outside the open domain. It assumes neither a general Cauchy conclusion nor contour independence.

2.4 The canonical defect and compatibility

For a bounded operator T, admissibility records an open holomorphy domain, a genuine cycle in that domain and the resolvent set, winding one on the spectrum, and winding zero outside the domain. These are input data. No functional calculus value or independence equation is a field of that record.

Two admissible choices are compared on the union of their holomorphy domains intersected with the resolvent set. Outside this comparison domain, their windings are both zero or both one. General Cauchy comparison proves equality of their normalized Dunford integrals. A canonical value can therefore be selected from any nonempty family of admissible data and then represented by every admissible original contour.

The original contour supplied in Theorem 11.6 provides this witness for both operators. After rewriting the two canonical values through that contour, the actual resolvent defect identity proves the manuscript's original tagged integral formula and its geometric length bound. No extra existence assumption is needed for this theorem.

For compatibility, the two endpoint operators may use different admissible contours. When the operator defect is zero, their two intertwined resolvents agree on the overlap of their resolvent sets. They glue to one holomorphic map on the union. General Cauchy comparison again identifies the integrals. Applying this naturality result to the total map of a finite tower proves Corollary 11.10 whenever its endpoint expressions are defined. The formal witnesses express this definedness. They do not assert compatibility as a premise.

The current coverage statement concerns the manuscript's enumerated proof scopes. It does not assert a separate theorem constructing an admissible contour for every possible operator and holomorphic germ, or an identification with a different software library's calculus API.

The relevant final endpoints are in `ReleaseCanonicalHFC.lean`, `ReleaseCanonicalNaturality.lean`, and `ReleaseCanonicalDefect.lean`. Their exact source identity and execution receipts are recorded separately from this exposition.

3 Reading the inherited formal walkthrough

The mathematical text develops the mathematical theory and its written proofs. The visual guide provides finite models for its identities, precision statements, and boundaries. This guide explains how to read and use the preserved Lean component: the types accepted by each theorem, the objects actually constructed, the obligations supplied by a caller, and the evidence attached to the source. These three companions have complementary roles.

This walkthrough describes the foundational 25-row formal contract. Section 1 records the chapter-wide correspondence and its boundaries.

The source register identifies the foundational formalization. The mathematical text includes broader written proofs of lattice gluing, necessary precision delay, inverse-limit convergence, general DVR arithmetic, holomorphic and exponential defects, and multilevel identities. Their appearance beside mapped results does not give them Lean proofs.

The fixed formal contract contains 25 rows: 21 numbered canonical statements, one Smith phase identity, and three mathematical algorithm specifications. The appendix crosswalk identifies their locations in the mathematical text. Throughout this guide, "row" means a row of the canonical formal map, never a count of every theorem in the mathematical text.

Quantity	What is counted
15 Lean source files	Four shared foundation modules, eight substantive Chapter II modules, two aggregate imports, and one audit module.
5,603 source lines	Physical lines in those files, including comments and blank lines.
25 map rows	Statement families, constructions, and specifications in the foundational contract.
160 unique targets	Distinct mapped declarations. Not all are theorem declarations.
171 target occurrences	Targets counted again when reused by another contract row.
1,034 namespace entries	Entries reported by the recorded compiled namespace scan, includ- ing generated auxiliaries.

A name count is not a measure of mathematical depth. A short wrapper can collect several substantial conclusions. A large structure can expose a difficult hypothesis without constructing its witness. The useful unit is a complete type together with its proof, its dependencies, and an accurate comparison with the prose statement.

3.1 Three useful reading routes

For a first reading, follow Sections 4–12: exact coordinates lead to weighted transfer, then arithmetic and typed defects. For a coverage review, use the crosswalk to the mathematical text in Appendix A, read the corresponding row in Section 15, and inspect its source type. For a reproduction attempt, read Section 13 before running any build command. The source checksum check and the compiler audit establish different things.

The main walkthroughs use exact source excerpts, with explanatory equations outside the listings. Names are case sensitive. `PowerSeries`, `Polynomial`, `LinearMap`, and `ContinuousLinearMap` are different types. Similar notation is not a license to replace one with another in a theorem application.

4 Lean types as mathematical bookkeeping

The basic theorem syntax places assumptions before a colon and a conclusion after it. A term after `:=` supplies the definition or proof. A `by` block is tactic syntax that elaborates a proof term. A successful elaboration still leaves a semantic question for the reader: does this exact type express the intended mathematical assertion?

Expression	Reading in this component
`[Semiring R]`	A typeclass instance supplying semiring operations and laws. Subtraction need not exist.
`[Ring R]`	A ring, possibly noncommutative. Multiplicative order remains significant.
`[CommSemiring R]`	Commutative multiplication is available, as in the general Fourier certificate.
`[Module R M]`	A scalar action with its laws. The scalar type is explicitly named.
`M →ₗ[R] N`	An R-linear map from M to N, including additive and scalar compatibility.
`Module.End R M`	The endomorphism ring or semiring of a single module.
`X →L[k] Y`	A continuous linear map with operator norm in the relevant normed setting.
`M ≃ₗ[R] N`	A linear equivalence, including both inverse laws.
`R →+* S`	A ring or semiring homomorphism. It preserves the operations used to transport an inverse.
`[Fintype E]`	A finite enumeration of all elements of E.
`Fin n`	Natural numbers strictly below n, bundled with their bound.
`ZMod n`	Integers modulo n. The nonzero modulus hypotheses matter for finite cardinality.
`AddSubgroup M`	An additive subgroup, without automatically being a valuation-ring lattice.
`Nonempty A`	Existence of a term of type A. It does not by itself supply a computational search procedure.
`Prop`	The sort of propositions, used for certificates whose fields are proof obligations.

Parenthesized arguments are normally supplied explicitly. Braced arguments can be inferred. Arguments in square brackets request typeclass synthesis. Section variables become parameters of later declarations when used. Therefore a source theorem must be read with its section context. In this project, `autoImplicit = false` and `relaxedAutoImplicit = false` prevent undeclared identifiers from silently becoming new implicit parameters.

The common `constructor` tactic splits a structure or conjunction into its fields. `rcases` opens a supplied witness into its data and proofs. `rw` rewrites using proved equalities. `simp` applies simplification rules and produces a proof term. `ext` reduces equality of structured maps to pointwise equality. `exact` supplies a term of the required type. `rfl` uses definitional equality. None of these words is a substitute for a missing mathematical hypothesis.

4.1 Definitions, certificates, and algorithms

A structure such as `TwoSidedInverse x y` records equations about already supplied x and y. A theorem that accepts this structure does not discover y. By contrast, the rationality branch for finite graphs

defines a denominator and numerator from the graph and proves their properties without accepting rationality as an input.

`noncomputable` does not mean unproved. It marks a definition that is not offered as executable code through Lean's ordinary computational evaluation. This component uses noncomputable choices from finite sets, instances for finite types, determinants, adjugates, and constructions of least exponents. Their mathematical properties can be proved even though a standalone executable implementation is not supplied.

The three algorithm rows certify mathematical contracts relating inputs to outputs. They do not certify the companion Python package, its storage layout, its numerical residuals, or its performance. Even where a finite enumeration is executable in principle, the contract contains no general complexity estimate or extraction theorem.

5 Source architecture and entry points

In the foundational project, the main entry point is import KomorebiVerification.PaperII. It imports the eight substantive modules but not the audit command. The root module KomorebiVerification.lean imports the aggregate for Lake's default library target. The standalone audit imports the substantive modules explicitly and inspects the resulting environment.

Module or layer	Responsibility
Shared foundation	`ProjectorCore`, `ExactCoordinates`, `FibreRefinement`, and `ResidueProjectors`: complete finite projector families, range coordinates, masks, and refinement.
`NestedRefinement`	Parent maps, absorption, child uniqueness, exact coordinate equivalences, cocycles, naturality, and block aggregation.
`ResidueFourierCRT`	Division-free masks, finite Fourier certificates, primitive-root specialization, polyphase refinement, and CRT labels.
`WeightedTransfer`	Ordered geometric inverses, normed Neumann inverses, multiplicative filtrations, and precise word pruning.
`TransferZeta`	Finite weighted walks, exact enumeration, first-edge recurrence, determinant rationality, and coefficient recurrence.
`ResultantSeparation`	Evaluated Bézout identities, numerator and kernel actions, denominator clearing, and certificate-based precision delay.
`SmithDyadic`	Concrete modular fibers, abstract certified products, rational sign matrices, local denominator weights, and the Smith-phase formula.
`TypedLinearDefects`	Fine-to-coarse maps between distinct spaces, power and resolvent defects, polynomial compatibility, and operator-norm estimates.
`LinearDefects`	Ring-valued counterparts and auxiliary inverse identities.
`KernelAudit`	Representative axiom reports and the fail-closed namespace gate.

The main mathematical dependencies are modular. Coordinate refinement does not require a valuation. Weighted pruning needs a filtration but not a graph incidence law. Transfer zeta counts graph walks but uses integer formal power series, not operator norms. Smith fibers work in finite modular arithmetic. The typed defect module is independent of a particular packet construction: a caller may use a coarsening map, but the theorem accepts any map of the stated type.

6 Exact packets and refinement

Rows 1–5 start from finite complete projector families. A family $P = (P_i)$ satisfies $P_i^2 = P_i$, $P_i P_j = 0$ for $i \neq j$, and $\sum_i P_i = I$. Orthogonality here is algebraic annihilation. An inner product, self-adjointness, and topology are absent from these assumptions. The canonical formal family also permits zero projectors, whereas the expanded traditional packet terminology requires nonzero members.

The packet space is the product of actual ranges,

$$\mathcal{P}(P) = \prod_i \text{range}(P_i).$$

Its elements carry their proofs of membership in a range. Analysis sends x to $(P_i x)_i$. Synthesis sums the underlying vectors. Completeness proves synthesis after analysis. Pairwise annihilation and idempotence prove analysis after synthesis. Both maps are independently defined, so the second inverse law has mathematical content: every tuple in the product of ranges is accepted.

6.1 What a refinement witness supplies

A parent map $\pi : J \to I$ assigns fine labels to coarse labels. The finite set `children parent a` is the fiber $\pi^{-1}(a)$. The refinement witness records surjectivity and exact equality of coarse maps with sums of their fine children.

Source: `KomorebiVerification/PaperII/NestedRefinement.lean`.

```
structure Refines
    (coarse : LinearProjectorFamily R M ι)
    (fine : LinearProjectorFamily R M κ)
    (parent : κ → ι) : Prop where
  surjective : Function.Surjective parent
  coarse_eq_sum_children :
    ∀ a, coarse.proj a = ∑ b ∈ children parent a, fine.proj b
```

The structure does not construct a parent map between arbitrary families. To apply its theorems, a caller supplies a map and proves these two fields. The resulting absorption identities are genuine consequences. If $\pi(j) = i$, then

$$P_i Q_j = Q_j = Q_j P_i;$$

if $\pi(j) \neq i$, both products vanish. Expand P_i as the sum of its children, distribute composition, and use orthogonality of the fine family. Transitivity follows by applying this reasoning through two successive parent maps.

6.2 A proof walkthrough: recovering a selected finite decomposition

The most general uniqueness theorem in row 2 does not use the refinement witness. Suppose a selected finite set S of packet vectors sums to x, with $v_j \in \text{range}(P_j)$ for $j \in S$. Applying P_i kills all terms except v_i, so $v_i = P_i x$. The exact proof shows which hypotheses enter.

Source: KomorebiVerification/PaperII/NestedRefinement.lean.

```
theorem finitePacket_decomposition_unique
    (P : LinearProjectorFamily R M κ) (s : Finset κ) (x : M)
    (packets : κ → M)
    (hrange : ∀ b ∈ s, packets b ∈ LinearMap.range (P.proj b))
    (hsum : ∑ b ∈ s, packets b = x) :
    ∀ b ∈ s, packets b = P.proj b x := by
  classical
  intro b hb
  have hself : P.proj b (packets b) = packets b := by
    rcases hrange b hb with ⟨y, hy⟩
    rw [← hy]
    exact P.idempotent b y
  have hproject : P.proj b (∑ j ∈ s, packets j) = packets b := by
    rw [map_sum]
    calc
      (∑ j ∈ s, P.proj b (packets j)) = P.proj b (packets b) := by
        apply Finset.sum_eq_single_of_mem b hb
        intro j hj hjb
        rcases hrange j hj with ⟨y, hy⟩
        rw [← hy]
        exact P.orthogonal hjb.symm y
      _ = packets b := hself
  calc
    packets b = P.proj b (∑ j ∈ s, packets j) := hproject.symm
    _ = P.proj b x := congrArg (P.proj b) hsum
```

First `rcases hrange b hb with` extracts a preimage y for the selected packet. Rewriting the packet as $P_b y$ makes idempotence applicable. Next `map_sum` distributes P_b across the finite sum. `Finset.sum_eq_single_of_mem` leaves the diagonal term. Each other term vanishes after its own range witness is opened. Finally `congrArg` applies P_b to the supplied sum equality.

The conclusion is restricted to $b \in S$. No conclusion is drawn about values of the packet function outside that finite set. The source family is complete as a bundled object, but the proof of this result never uses its completeness field. This explains the map's "Stronger" classification without suggesting a theorem about arbitrary infinite sums.

6.3 Coordinate changes need no parent map

Let A_P and S_P denote analysis and synthesis. For any two complete finite families on the same module, define

$$R_{P\to Q} = \mathsf{A}_Q \mathsf{S}_P, \qquad R_{Q\to P} = \mathsf{A}_P \mathsf{S}_Q.$$

The source calls these maps refinement and coarsening.

Source: KomorebiVerification/PaperII/NestedRefinement.lean.

```
variable (coarse : LinearProjectorFamily R M ι)
variable (fine : LinearProjectorFamily R M κ)

/-- Resolve coarse range coordinates by synthesis followed by fine analysis. -/
```

```
def refineCoordinates : PacketSpace coarse →ₗ[R] PacketSpace fine :=
  fine.rangeAnalysis.comp coarse.rangeSynthesis

/-- Regroup fine range coordinates by synthesis followed by coarse analysis. -/
def coarsenCoordinates : PacketSpace fine →ₗ[R] PacketSpace coarse :=
  coarse.rangeAnalysis.comp fine.rangeSynthesis
```

These maps are inverses because $\mathsf{S}_Q\mathsf{A}_Q = I$ and $\mathsf{A}_P\mathsf{S}_P = I$. The proof uses the earlier simplification rules for exact coordinates.

Source: `KomorebiVerification/PaperII/NestedRefinement.lean`.

```
@[simp] theorem coarsenCoordinates_refineCoordinates
    (packets : PacketSpace coarse) :
    coarsenCoordinates coarse fine (refineCoordinates coarse fine packets) =
      packets := by
  simp [coarsenCoordinates, refineCoordinates]

/-- Refinement after coarsening is the identity on fine range coordinates. -/
@[simp] theorem refineCoordinates_coarsenCoordinates
    (packets : PacketSpace fine) :
    refineCoordinates coarse fine (coarsenCoordinates coarse fine packets) =
      packets := by
  simp [coarsenCoordinates, refineCoordinates]

/-- The exact linear equivalence between the two independent range products. -/
def coordinateRefinementEquiv : PacketSpace coarse ≃ₗ[R] PacketSpace fine where
  toFun := refineCoordinates coarse fine
  invFun := coarsenCoordinates coarse fine
  left_inv := coarsenCoordinates_refineCoordinates coarse fine
  right_inv := refineCoordinates_coarsenCoordinates coarse fine
  map_add' := (refineCoordinates coarse fine).map_add
  map_smul' := (refineCoordinates coarse fine).map_smul
```

The short `simp` proofs rest on established inverse laws. They are not assumptions that refinement loses no data. The two packet products are full exact presentations of the same module. Counting the number of labels alone does not measure their dimension: every label contains an entire projector range.

A parent witness adds component formulas. The fine component is obtained by applying the fine projector to its coarse parent packet. The coarse component is the sum of its fine children. These are additional identities about the already defined coordinate equivalence. They do not turn it into averaging, sampling, or truncation.

With a third complete family H, the same inverse laws give the cocycle

$$R_{Q\to H}R_{P\to Q} = R_{P\to H}.$$

The theorem `refineCoordinates_comp` first proves the equality on each packet and then uses extensionality of linear maps. A reader can thus separate the simple conceptual proof from the bookkeeping needed to state it as equality of bundled maps.

6.4 Naturality and independently refined blocks

For $T : M \to N$ with source family P and target family Q, the exact coordinate operator is $\mathsf{A}_Q T \mathsf{S}_P$. Its composition law cancels the middle pair of synthesis and analysis maps. Changing the source and target coordinate systems independently gives the corresponding commuting naturality identity.

The block convention places the output index first:

$$T_{ji} = Q_j T P_i.$$

If source and target families refine separately, a coarse block is the double sum over fine descendants of its source and target labels. The mapped theorem permits different source and target modules, which is stronger than the endomorphism special case. Its composite traceability wrapper also records reconstruction of T from all coarse blocks and from all fine blocks.

The finite algebraic result does not establish sparsity of a computed representation, a block extraction routine with a finite complexity bound, or convergence of an infinite block matrix. These need separate hypotheses and, where intended, separate formal developments.

7 Residue selectors, Fourier certificates, and CRT

Rows 6–9 act on coefficient functions $x : \mathbb{N} \to A$. At a positive modulus N, the residue projector keeps $x(n)$ when n mod $N = j$ and returns zero otherwise. Since exactly one residue label matches each n, idempotence, pairwise annihilation, and completeness follow pointwise. No division or roots of unity are required.

This representation is a sequence module, not a formal power series object with a multiplication law. It is sufficient for coefficient masks. A user wishing to apply the result to a particular function space must supply the connection to that space and any closure or convergence facts.

7.1 The finite Fourier table is an input over general semirings

A Fourier realization uses scalars ω, $\omega^\vee$, and η. The weighted residue kernel is

$$\eta \sum_{k=0}^{N-1} (\omega^\vee)^{jk} \omega^{rk}.$$

Over a general commutative semiring the source does not assume that $\omega^\vee$ is an inverse or that $\eta = 1/N$. Instead it accepts a finite orthogonality table with exactly the required conclusion at residue addresses.

Source: `KomorebiVerification/PaperII/ResidueFourierCRT.lean`.

```
structure FiniteRingFourierOrthogonalityCertificate
    (R : Type u) [CommSemiring R] (N : Nat)
    (omega omegaDual eta : R) : Prop where
  omega_pow_order : omega ^ N = 1
  weighted_residue_sum_eq (j r : Fin N) :
    eta * ringFourierResidueKernel N omega omegaDual j r =
      if r = j then 1 else 0
```

The substantive bridge is from that finite table to every coefficient address. The equation $\omega^N = 1$ reduces the exponent n to n mod N. The last line below uses `Fin.ext_iff` to equate residue labels through their underlying natural numbers.

Source: `KomorebiVerification/PaperII/ResidueFourierCRT.lean`.

```
theorem FiniteRingFourierOrthogonalityCertificate.toRingCertificate
    {R : Type u} [CommSemiring R]
    (N : Nat) (hN : 0 < N) (omega omegaDual eta : R)
    (certificate :
      FiniteRingFourierOrthogonalityCertificate R N omega omegaDual eta) :
    RingFourierOrthogonalityCertificate R N omega omegaDual eta := by
  refine (?_)
  intro j n
  rw [ringFourierKernel_eq_residueKernel N hN omega omegaDual
    certificate.omega_pow_order j n]
  rw [certificate.weighted_residue_sum_eq]
  congr 1
  simp only [Fin.ext_iff]
```

After this bridge, the average and the mask are equal coefficient by coefficient: the normalized scalar is one at the selected address and zero elsewhere. The proof works with a finite table rather than an infinite family of independent premises.

There is also a stronger field branch. A primitive root of positive order supplies the normalized orthogonality certificate for the inverse root internally. Its field hypothesis is essential to that branch's scope. Root order by itself is not a replacement for the table over arbitrary rings with zero divisors.

7.2 Refining residues and factoring their labels

For positive N and k, the modulus-N selector is the sum of all modulus-Nk selectors whose labels reduce to it. This is exact regrouping of masks. No packet is discarded and there is no approximation error to estimate.

For positive coprime m, n, CRT gives an equivalence between labels modulo mn and pairs of labels modulo m and n. The corresponding selector is the composite of the two independent selectors in either order. For $m = 2^a$ and positive odd n, the factorization into dyadic and odd moduli is a specialization.

These are coordinate facts. They do not prove that rational spectral projectors have integral denominators, that a completed lattice splits, or that solving a lifting problem is possible. The later arithmetic modules use different data for those questions. This separation is useful when reading the same terminology about dyadic and odd moduli in the mathematical text: residue factorization and spectral denominator clearing are distinct statements.

8 Weighted inversion: three separate branches

Rows 10–11 concern $D + H$ in a possibly noncommutative ring. A supplied inverse D^{-1} that works on both sides permits the factorization

$$D + H = D(1 + A), \qquad A = D^{-1}H.$$

Thus an inverse of $1 + A$ is multiplied by D^{-1} on the right. Reversing this order changes the formula in a noncommutative ring. The source parameter named `K` in these declarations means the perturbation H. It is not a fraction field.

Three hypotheses lead to three different results. Nilpotence gives a finite ring identity. A homomorphism that kills a filtration level gives a finite identity after passing to precision. A strict norm bound in a complete normed ring gives an infinite Neumann sum. The formal component does not merge these into an unrestricted adic convergence theorem.

8.1 The finite geometric identity

For $A^N = 0$, define $B = \sum_{n=0}^{N-1} (-A)^n$. The product formulas for finite geometric sums prove both $(1 + A)B = 1$ and $B(1 + A) = 1$.

Source: `KomorebiVerification/PaperII/WeightedTransfer.lean`.

```
def finiteGeometricInverse {R : Type u} [Ring R] (a : R) (N : ℕ) : R :=
  ∑ n ∈ Finset.range N, (-a) ^ n

/-- If `a ^ N = 0`, the finite geometric candidate is a genuine two-sided inverse of `1 + a`. -/
theorem finiteGeometricInverse_spec_of_pow_eq_zero {R : Type u} [Ring R]
    (a : R) (N : ℕ) (hN : a ^ N = 0) :
    TwoSidedInverse (1 + a) (finiteGeometricInverse a N) := by
  have hneg : (-a) ^ N = 0 := by
    rw [neg_pow, hN, mul_zero]
  constructor
  · simpa [finiteGeometricInverse, hneg] using geom_sum_mul_neg (-a) N
  · simpa [finiteGeometricInverse, hneg] using mul_neg_geom_sum (-a) N
```

The source separately invokes the two ordered geometric sum lemmas. This is why `TwoSidedInverse` has two fields. An inverse from one side does not automatically supply the other in a general noncommutative ring. Once this result is available, the theorem for a diagonal part with transfers transports the candidate through the factorization of $D + H$.

The cutoff N and its nilpotence proof are inputs. The theorem is not a procedure for deciding whether an arbitrary ring element is nilpotent. Its finite sum is explicit, but a concrete matrix application must still identify the matrix ring, supply D^{-1}, and prove the required equations.

8.2 The filtration is deliberately minimal

A multiplicative filtration records additive closure, decreasing levels, and multiplicative addition of indices.

Source: `KomorebiVerification/PaperII/WeightedTransfer.lean`.

```
structure MultiplicativeFiltration (R : Type u) [Ring R] where
  Mem : ℕ → R → Prop
  zero_mem : ∀ n, Mem n 0
  add_mem : ∀ {n x y}, Mem n x → Mem n y → Mem n (x + y)
  neg_mem : ∀ {n x}, Mem n x → Mem n (-x)
  one_mem : Mem 0 1
  antitone : ∀ {m n x}, m ≤ n → Mem n x → Mem m x
  mul_mem : ∀ {m n x y}, Mem m x → Mem n y → Mem (m + n) (x * y)
```

Write F^n for the predicate `Mem n`. Multiplicativity is $F^m F^n \subseteq F^{m+n}$. Antitonicity sends membership at a larger index to membership at a smaller one. Completeness, separation, and a topology for an inverse limit are not fields of this structure. Thus the word "filtration" in its name does not imply any convergence statement.

A precision certificate for a ring homomorphism $q : R \to S$ says $q(F^L) = 0$. It does not claim that S is a particular quotient or that $\ker q = F^L$. Killing the level is sufficient for the theorem. The proof below first puts a word into its cumulative filtration level and then uses antitonicity to reach F^L.

Source: `KomorebiVerification/PaperII/WeightedTransfer.lean`.

```
structure PrecisionCertificate {R : Type u} {S : Type v} [Ring R] [Ring S]
    (filtration : MultiplicativeFiltration R) (q : R →+* S) (L : ℕ) : Prop where
  map_eq_zero : ∀ {x : R}, filtration.Mem L x → q x = 0

/-- A transfer word whose cumulative weight reaches `L` vanishes in every realization carrying a
precision-`L` certificate. -/
theorem wordProduct_vanishes_at_precision {R : Type u} {S : Type v}
    [Ring R] [Ring S] (filtration : MultiplicativeFiltration R)
    (q : R →+* S) (L : ℕ) (certificate : PrecisionCertificate filtration q L)
    {E : Type w} (weight : E → ℕ) (edge : E → R)
    (hedge : ∀ e, filtration.Mem (weight e) (edge e))
    (word : List E) (hweight : L ≤ wordWeight weight word) :
    q (wordProduct edge word) = 0 := by
  apply certificate.map_eq_zero
  apply filtration.antitone hweight
  exact filtration.wordProduct_mem weight edge hedge word
```

The caller supplies the facts about edge membership $A_e \in F^{w(e)}$. Induction on a list gives $A_{e_1} \cdots A_{e_r} \in F^{w(e_1)+\cdots+w(e_r)}$. All words of weight at least L therefore disappear after applying q.

8.3 Written products and chronological paths

The definition `wordProduct edge word` is `(word.map edge).prod`. For the list $[e_1, \dots, e_r]$, it gives the written product $A_{e_1} \cdots A_{e_r}$. When these are operators, the rightmost factor acts first. This is the convention in the canonical formal branch for weighted words.

The expanded traditional discussion of block paths writes a chronologically traversed path $e_1, \dots, e_r$ as $A_{e_r} \cdots A_{e_1}$. To translate an indexed term, reverse the list. The full sums over all words of a fixed length agree under this bijection. The sums truncated by weight also agree because reversal preserves cumulative weight. Individual indexed products need not agree. An equation for one labeled path must therefore retain its order convention.

The generic formal sum over weighted words ranges over edge lists without imposing graph adjacency. In a concrete block realization, incompatible products may vanish for structural reasons, but those incidence facts are not fields of `MultiplicativeFiltration`. They must be supplied when interpreting the sum as a sum over directed block paths.

8.4 What is actually retained

The pruned expression contains only words with total weight strictly below L. Its length cutoff N is a separate parameter.

Source: KomorebiVerification/PaperII/WeightedTransfer.lean.

```
def retainedFixedWordSum {E : Type u} {R : Type v} [Fintype E] [Ring R]
    (weight : E → ℕ) (edge : E → R) (L n : ℕ) : R :=
  ∑ word : Fin n → E with wordWeight weight (List.ofFn word) < L,
    wordProduct edge (List.ofFn word)

/-- The explicit signed, weight-pruned word candidate through word length `N - 1`.

Unlike `finiteGeometricInverse`, this expression visibly contains only retained transfer words:
every summand has cumulative weight strictly below `L`.  The independent length cutoff `N` permits
the sharper cutoff obtained from a lower bound on all edge weights. -/
def prunedWordInverse {E : Type u} {R : Type v} [Fintype E] [Ring R]
    (weight : E → ℕ) (edge : E → R) (L N : ℕ) : R :=
  ∑ n ∈ Finset.range N, (-1 : R) ^ n * retainedFixedWordSum weight edge L n
```

If each edge weight is at least m, a word of length N has weight at least mN. Hence $L \le mN$ makes the Nth mapped power vanish. For $m > 0$, the proved ceiling inequality allows $N = \lceil L/m \rceil$. With merely positive integral weights, $N = L$ is a valid coarse cutoff.

The equality between the mapped full geometric sum and the mapped sum of retained words is proved by splitting the finite word sums into retained and killed terms. The inverse conclusion then follows from mapped nilpotence. This order of reasoning matters: pruning is justified by exact zero in the precision realization, not by a small numerical residual.

The full theorem for a diagonal part with transfers exposes all remaining data. It accepts the inverse of D, the normalized edge expansion, the precision certificate, the lower bound on the weights, the cutoff inequality, and the proofs about edge membership.

Source: KomorebiVerification/PaperII/WeightedTransfer.lean.

```
theorem finitePrecisionPrunedDiagonalTransferInverse_of_minWeight
    {R : Type u} {S : Type v} [Ring R] [Ring S]
    (filtration : MultiplicativeFiltration R)
    (q : R →+* S) (L m N : ℕ) (certificate : PrecisionCertificate filtration q L)
    (D Dinv K : R) (hD : TwoSidedInverse D Dinv)
    {E : Type w} [Fintype E] (weight : E → ℕ) (edge : E → R)
    (htransfer : Dinv * K = ∑ e : E, edge e)
    (hmin : ∀ e, m ≤ weight e) (hcutoff : L ≤ m * N)
    (hedge : ∀ e, filtration.Mem (weight e) (edge e)) :
    TwoSidedInverse
      (q (D + K))
      (q (prunedDiagonalTransferInverse weight edge Dinv L N)) := by
  have hpruned := prunedWordInverse_spec_of_minWeight
    filtration q L m N certificate weight edge hmin hcutoff hedge
  have htransferMapped : q Dinv * q K = ∑ e : E, q (edge e) := by
    simpa only [map_mul, map_sum] using congrArg q htransfer
  have hnormalized :
      TwoSidedInverse (1 + q Dinv * q K) (q (prunedWordInverse weight edge L N)) := by
    simpa only [map_add, map_one, map_sum, htransferMapped] using hpruned
  have hdiagonal := diagonal_inverse_of_perturbation_inverse
    (q D) (q Dinv) (q K) (q (prunedWordInverse weight edge L N)) (hD.map q) hnormalized
  simpa only [prunedDiagonalTransferInverse, map_add, map_mul] using hdiagonal
```

The proof first obtains the pruned inverse of $1+\sum_e A_e$. It maps the supplied equality for normalized transfers, rewrites the inverse target, and applies the diagonal factorization lemma. The final `simpa` only unfolds the candidate and the homomorphism's preservation of addition and multiplication. No inverse of D or quotient construction is hidden in that final simplification.

8.5 The analytic branch and the algorithm contract

In a complete normed ring, $\|D^{-1}H\| < 1$ supplies a separate inverse given by a Neumann series. The source defines the analytic candidate noncomputably and proves both inverse laws using the normed summability theory. This is neither the finite nilpotence premise nor completeness of an arbitrary filtered ring. An application using a valuation needs a bridge to one of the formal settings.

Row 24 packages the bounded geometric/pruned expression as a mathematical algorithm specification. It proves that the chosen ceiling cutoff meets the required inequality and that the mapped result is an inverse on both sides. It does not construct a modular inverse routine in a chosen basis, decide symbolic zero, certify the Python implementation, or establish a runtime bound.

9 Transfer zeta: from independent walks to rationality

Rows 12–13 and 25 describe the finite graph construction. The input is a finite directed multigraph with positive natural edge weights. Parallel edges are distinct edge values and therefore contribute their multiplicities automatically. Repeated vertices and edges are allowed: the objects counted are walks.

Source: `KomorebiVerification/PaperII/TransferZeta.lean`.

```
structure PositiveWeightedMultigraph (V : Type u) (E : Type v) where
  source : E → V
  target : E → V
  weight : E → ℕ
  weight_pos : ∀ e, 0 < weight e

namespace PositiveWeightedMultigraph

variable {V : Type u} {E : Type v}

/-- The recursive endpoint condition for an ordered list of edges. -/
def IsWalk (G : PositiveWeightedMultigraph V E) (s t : V) : List E → Prop
  | [] => s = t
  | e :: es => G.source e = s ∧ G.IsWalk (G.target e) t es
```

The recursion `IsWalk` says an empty edge list starts and ends at the same vertex. A nonempty list starts with an edge whose source is the current vertex, then continues from that edge's target. The total weight is the sum of edge weights.

9.1 Why a coefficient has a finite enumeration

Positivity implies that a walk of total weight n has length at most n. The enumerator can therefore range over lengths $0, \dots, n$ and all edge tuples of each length, then filter by endpoint validity and exact total weight.

Source: KomorebiVerification/PaperII/TransferZeta.lean.

```
/--
The exact finite enumeration used for the coefficient of weight `n`.
-/
def walkTuplesAtWeight [Fintype E] [DecidableEq E] [DecidableEq V]
    (G : PositiveWeightedMultigraph V E) (s t : V) (n : ℕ) :
    Finset (BoundedEdgeTuple E n) :=
  Finset.univ.filter fun x ↦
    G.IsWalk s t (tupleEdges x) ∧ G.pathWeight (tupleEdges x) = n

/-- Membership in the finite enumerator has exactly the advertised meaning. -/
theorem mem_walkTuplesAtWeight_iff [Fintype E] [DecidableEq E] [DecidableEq V]
    (G : PositiveWeightedMultigraph V E) (s t : V) (n : ℕ)
    (x : BoundedEdgeTuple E n) :
    x ∈ G.walkTuplesAtWeight s t n ↔
      G.IsWalk s t (tupleEdges x) ∧ G.pathWeight (tupleEdges x) = n := by
  simp [walkTuplesAtWeight]
```

The module also defines an independent semantic type of finite edge lists having the requested endpoints and weight, without building the enumeration bound into that definition. The capture theorem uses positivity to connect the unrestricted semantic representation to the bounded tuple enumeration. The counting theorem and the trace wrapper identify the returned natural number with the cardinality of this semantic walk type.

This avoids defining correctness merely as equality with the same algorithm under another name. The finite enumeration and the semantic list predicate are separate constructions connected by equivalences. The implementation remains a potentially large exhaustive enumeration. No efficiency theorem is claimed.

9.2 The recurrence from the first edge

Fix a target t and let $a_{st}(n)$ count walks from s to t of weight n. The empty walk contributes 1 exactly when $s = t$ and $n = 0$. Every other walk has a unique first edge. Removing it gives

$$a_{st}(n) = \mathbf{1}_{s=t,\, n=0} + \sum_{\substack{e:\, s\to u \\ w(e)\le n}} a_{ut}(n - w(e)).$$

The source constructs the corresponding equivalence from the first edge and obtains the recurrence by cardinality. Positivity ensures the recursive weight decreases. The bound $w(e) \le n$ is explicit, avoiding an incorrect use of truncated natural subtraction when the first edge is too heavy.

Define $Z_{st}(X) = \sum_{n\ge 0} a_{st}(n) X^n$. This is a formal power series over $\mathbb{Z}$. The weighted adjacency entry $A_{su}(X)$ sums $X^{w(e)}$ over all edges $s \to u$. The recurrence from the first edge becomes

$$(I - A(X))Z(X) = I.$$

The row/column orientation here follows traversal from the first edge. When comparing it with operator blocks with the output index first, read the definitions rather than transferring an index convention by analogy.

9.3 A second construction: determinant and adjugate

Finite vertex data define the polynomial matrix $B(X) = I - A(X)$ and

$$Q(X) = \det B(X), \qquad P_{st}(X) = \operatorname{adj}(B(X))_{st}.$$

Positive edge weights give $A(0) = 0$, hence $B(0) = I$ and $Q(0) = 1$. Thus Q is a unit in $\mathbb{Z}[\![X]\!]$. The adjugate identity constructs a matrix Z^{mat} that is an inverse on both sides.

The proof identifies the independently enumerated series matrix with this rational matrix. It uses associativity and the two inverse equations.

Source: KomorebiVerification/PaperII/TransferZeta.lean.

```
theorem enumeratedPathZetaMatrix_eq_matrixPathZetaMatrix :
    G.enumeratedPathZetaMatrix = G.matrixPathZetaMatrix := by
  calc
    G.enumeratedPathZetaMatrix = 1 * G.enumeratedPathZetaMatrix :=
      (Matrix.one_mul _).symm
    _ = (G.matrixPathZetaMatrix * G.transferSeriesMatrix) *
        G.enumeratedPathZetaMatrix := by
      rw [G.matrixPathZetaMatrix_mul_transferSeriesMatrix]
    _ = G.matrixPathZetaMatrix *
        (G.transferSeriesMatrix * G.enumeratedPathZetaMatrix) :=
      Matrix.mul_assoc _ _ _
    _ = G.matrixPathZetaMatrix * 1 := by
      rw [G.transferSeriesMatrix_mul_enumeratedPathZetaMatrix]
    _ = G.matrixPathZetaMatrix := Matrix.mul_one _
```

This is the uniqueness calculation $Z = IZ = (Z^{\mathrm{mat}}B)Z = Z^{\mathrm{mat}}(BZ) = Z^{\mathrm{mat}}$. Rationality was not assumed for the enumerated coefficients. The final existential theorem packages the already constructed numerator and denominator.

Source: KomorebiVerification/PaperII/TransferZeta.lean.

```
theorem finite_graph_pathZeta_rationality (s t : V) :
    ∃ P Q : Polynomial ℤ,
      Q.coeff 0 = 1 ∧ Q ≠ 0 ∧
        (Q : PowerSeries ℤ) * G.pathZeta s t = (P : PowerSeries ℤ) := by
  refine ⟨G.transferNumerator s t, G.transferDenominator,
    G.transferDenominator_coeff_zero, G.transferDenominator_ne_zero, ?_⟩
  exact G.transferDenominator_mul_pathZeta s t
```

The older `RationalCertificate` interface remains available for an externally supplied pair P, Q. Its theorem is explicit about its certificate. The canonical branch for finite graphs does not take that certificate. It derives its own data from the graph. The map preserves both interfaces so users can choose one without confusing their hypothesis burdens.

9.4 The recurrence is extracted from the proved identity

Write $Q(X) = \sum q_j X^j$. Taking coefficient n in $QZ = P$ gives an antidiagonal convolution. Once $n >$ deg P, the coefficient on the right is zero.

Source: KomorebiVerification/PaperII/TransferZeta.lean.

```
theorem finite_graph_constant_coefficient_recurrence (s t : V)
    {n : ℕ} (hn : (G.transferNumerator s t).natDegree < n) :
    ∑ p ∈ Finset.HasAntidiagonal.antidiagonal n,
        G.transferDenominator.coeff p.1 *
          (G.coefficientAlgorithm s t p.2 : ℤ) = 0 := by
  have hcoeff := congrArg (PowerSeries.coeff n)
    (G.transferDenominator_mul_pathZeta s t)
  rw [PowerSeries.coeff_mul] at hcoeff
  simpa [Polynomial.coeff_eq_zero_of_natDegree_lt hn] using hcoeff
```

Because $q_0 = 1$, this can be read as an eventual recurrence with constant coefficients for $a_{st}(n)$. The source's antidiagonal form states the identity without needing to rearrange its leading term. It does not guess a recurrence from sampled coefficients.

The determinant denominator may have canceling factors for a particular matrix entry, so this construction does not prove minimality of Q or of the recurrence order. The determinant and adjugate definitions are noncomputable here. The result concerns formal power series and supplies no disk of analytic convergence, extracted symbolic determinant program, or rationality theorem for infinite graphs.

10 Evaluated separation and lattice precision

Rows 14–16 start with endomorphisms F, G, U, V of a vector space over a field K. The central premise is an evaluated Bézout equality

$$UF + VG = rI.$$

The source does not construct polynomials whose evaluation gives these maps, compute a general resultant, factor a characteristic polynomial, or instantiate Cayley–Hamilton. These operations may justify the premise in a mathematical application, but they remain upstream of this formal interface.

Let $N = VG$. The identity without denominators is $N^2 = rN$. After adding $r \neq 0$, the normalized map $E = r^{-1}N$ is idempotent. The bundled `EvaluatedBezoutCertificate` contains six pairwise commutations as well as nonvanishing, Bézout, and annihilation. The mapped minimal theorems need fewer fields and state this directly.

10.1 A proof walkthrough: the square identity without division

Only $FG = GF$, $GU = UG$, and $FG = 0$ are needed to kill the cross term $VGUF$. Using $VG = rI - UF$ on one occurrence of the numerator then proves the square identity.

Source: KomorebiVerification/PaperII/ResultantSeparation.lean.

```
theorem separatorNumerator_mul_self_of_bezout
    (bezout : U * F + V * G = r • (1 : End (K := K) (M := M)))
    (commute_FG : Commute F G) (commute_GU : Commute G U) (annihilate : F * G = 0) :
    separatorNumerator V G * separatorNumerator V G = r • separatorNumerator V G := by
  calc
    separatorNumerator V G * separatorNumerator V G =
        separatorNumerator V G *
```

```
        (r • (1 : End (K := K) (M := M)) - U * F) := by
          exact congrArg (separatorNumerator V G * ·)
            (separatorNumerator_eq_scalar_sub_of_bezout bezout)
_ = separatorNumerator V G * (r • (1 : End (K := K) (M := M))) -
      separatorNumerator V G * (U * F) := by rw [mul_sub]
_ = r • separatorNumerator V G := by
  rw [separatorNumerator_mul_complement_eq_zero_of_commute
    commute_FG commute_GU annihilate]
  simp only [sub_zero, mul_smul_comm, mul_one]
```

The proof rewrites just the second factor, distributes multiplication over subtraction, and invokes vanishing of the cross term. It needs neither $r \neq 0$ nor four unused commutations from the stronger bundle. Consequently the theorem without denominators remains meaningful even when normalization is unavailable. The later normalized theorem uses scalar associativity and field simplification with the explicit nonzero hypothesis.

On $\ker F$, the evaluated Bézout equation gives $Nx = rx$ directly. On $\ker G$, $Nx = V(Gx) = 0$. After normalization, E acts as identity and zero respectively. A statement that F is nilpotent on some larger subspace would not imply $Fx = 0$ there. The formal theorem uses exact kernel membership. This prevents an intuition about primary summands from silently strengthening the proved conclusion.

10.2 What the formal lattice means

The lattice parameter is an `AddSubgroup M`. Preservation by a map means that each member is sent back into this additive subgroup. A general valuation ring, finite generation, full rank, completeness, and a chosen basis are not built into its type.

A clearing certificate at exponent s states that $\pi^s E$ preserves the lattice. The source names this `IsSeparationWeight`. If multiplication by π also preserves the lattice, the set of clearing exponents is upward closed. Given existence of some clearing exponent, `Nat.find` defines a least one and proves its minimality property. This is a noncomputable construction of a least witness, not a search algorithm over matrix entries.

If $r = u\pi^s$, then

$$\pi^s E = u^{-1} V G.$$

To turn this algebraic identity into lattice preservation, one must know that u^{-1} preserves the chosen lattice, together with V and G. An abstract field unit need not preserve an arbitrary additive subgroup. The formal certificate for the unit action on the lattice makes this assumption visible. The resulting exponent bounds the least separation weight from above. It need not be the minimal exponent.

The expanded traditional theory identifies exact weights through a lattice gluing module and gives necessary as well as sufficient delay statements. Those ordinary written proofs are beyond these mapped consequences of clearing certificates. They should not be attributed to `separationWeight` merely because the names describe related mathematics.

10.3 Precision delay as transport of a witness

Formal congruence at depth n means an explicit equation $x - y = \pi^n z$ with z in the lattice. If $\pi^s E$ preserves the lattice, input congruence at $m + s$ becomes output congruence at m.

Source: KomorebiVerification/PaperII/ResultantSeparation.lean.

```
/-- Congruence modulo the scalar lattice `pi^n lattice`, expressed with an explicit witness. -/
def CongruentModLattice (pi : K) (lattice : AddSubgroup M) (n : ℕ) (x y : M) : Prop :=
  ∃ z : M, z ∈ lattice ∧ x - y = (pi ^ n) • z

/-- An `s`-digit denominator certificate turns `m+s` input digits into `m` output digits. -/
theorem projector_precision_delay {pi : K} {E : End (K := K) (M := M)}
    {lattice : AddSubgroup M} {s m : ℕ}
    (hweight : IsSeparationWeight pi E lattice s) {x y : M}
    (hxy : CongruentModLattice pi lattice (m + s) x y) :
    CongruentModLattice pi lattice m (E x) (E y) := by
  rcases hxy with ⟨z, hz, hxy⟩
  refine ⟨((pi ^ s) • E) z, hweight z hz, ?_⟩
  rw [← map_sub, hxy, map_smul]
  simp only [LinearMap.smul_apply, pow_add, smul_smul]
```

The proof opens the original witness z. It chooses the new witness $z' = (\pi^s E)z$, whose lattice membership is exactly the clearing hypothesis. Linearity and $\pi^{m+s} = \pi^m \pi^s$ establish the new difference equation. No digit expansion, topology, or valuation API enters this proof. It is sufficient delay from a supplied certificate, without a claim that fewer input digits must fail.

11 Concrete Smith fibers and the dyadic sign model

Row 18 proves a concrete arithmetic statement over `ZMod (p^depth)` for $p > 1$. The hypothesis is not that p is prime. Since the modulus and coefficient are powers of the same integer, the needed gcd and divisibility identities hold at this stated scope.

11.1 The solution fiber is an actual type of solutions

For a weight s and depth L, multiplication by p^s is an additive endomorphism of $\mathbb{Z}/p^L\mathbb{Z}$. Its fiber over b is the subtype of elements that map to b.

Source: KomorebiVerification/PaperII/SmithDyadic.lean.

```
def smithZModMulHom (p weight depth : ℕ) :
    ZMod (p ^ depth) →+ ZMod (p ^ depth) :=
  nsmulAddMonoidHom (p ^ weight)

/-- The actual solution fibre of the one-coordinate Smith congruence
`p ^ weight * x = target` modulo `p ^ depth`. -/
abbrev SmithZModFiber (p weight depth : ℕ) (target : ZMod (p ^ depth)) :=
  ↥(smithZModMulHom p weight depth ⁻¹' {target})

/-- The concrete fibre is finite, inherited from the finite quotient `ZMod (p ^ depth)`. -/
noncomputable instance smithZModFiberFintype (p weight depth : ℕ) [NeZero (p ^ depth)]
    (target : ZMod (p ^ depth)) : Fintype (SmithZModFiber p weight depth target) :=
  Fintype.ofFinite _

/-- Concrete divisibility admissibility for a one-coordinate Smith congruence. -/
```

```
def SmithZModAdmissible (p weight depth : ℕ) (target : ZMod (p ^ depth)) : Prop :=
  p ^ min weight depth | target.val
```

The criterion is

$$p^{\min(s,L)} \mid \mathrm{val}(b),$$

where `target.val` is the canonical natural representative. The source proves that this is equivalent to nonemptiness of the actual fiber. If $s \leq L$, an admissibility witness supplies a quotient of the target representative and hence a solution. If $L \leq s$, admissibility forces the target to be zero. The zero residue supplies a solution.

The forward direction uses the modular representative of a supplied solution and divisibility by both the coefficient and modulus. Thus solvability is proved from concrete modular arithmetic, rather than inserted as a certificate field.

11.2 A proof walkthrough: counting an admissible fiber

Once a solution x_0 exists, translation by x_0 identifies the fiber with the kernel of multiplication by p^s. The kernel cardinality in this finite cyclic additive group is the gcd of its order with p^s.

Source: `KomorebiVerification/PaperII/SmithDyadic.lean`.

```
theorem smithZModFiber_card_of_admissible {p weight depth : ℕ} [NeZero (p ^ depth)] (hp : 1 < p)
    (target : ZMod (p ^ depth))
    (hadmissible : SmithZModAdmissible p weight depth target) :
    Fintype.card (SmithZModFiber p weight depth target) = p ^ min weight depth := by
  obtain ⟨x, hx⟩ := (smithZModFiber_nonempty_iff hp target).mpr hadmissible
  let f : ZMod (p ^ depth) →+ ZMod (p ^ depth) := smithZModMulHom p weight depth
  have hxEq : f x = target := Set.mem_singleton_iff.mp hx
  have e : SmithZModFiber p weight depth target ≃ f.ker := by
    simpa only [SmithZModFiber, f, hxEq] using f.fiberEquivKer x
  calc
    Fintype.card (SmithZModFiber p weight depth target) =
        Nat.card (SmithZModFiber p weight depth target) := Nat.card_eq_fintype_card.symm
    _ = Nat.card f.ker := Nat.card_congr e
    _ = (Nat.card (ZMod (p ^ depth))).gcd (p ^ weight) := by
      simpa [f, smithZModMulHom] using
        IsAddCyclic.card_nsmulAddMonoidHom_ker (ZMod (p ^ depth)) (p ^ weight)
    _ = (p ^ depth).gcd (p ^ weight) := by rw [Nat.card_zmod]
    _ = p ^ min weight depth := by
      rcases Nat.le_total weight depth with hwd | hdw
      · rw [Nat.min_eq_left hwd, Nat.gcd_eq_right_iff_dvd.mpr (Nat.pow_dvd_pow p hwd)]
      · rw [Nat.min_eq_right hdw, Nat.gcd_eq_left_iff_dvd.mpr (Nat.pow_dvd_pow p hdw)]
```

The final two cases use the divisibility of one power of p by the other to obtain

$$\#\{x : p^s x = b \pmod{p^L}\} = p^{\min(s,L)}.$$

This is an exact cardinality of a finite type. It does not enumerate the solutions or provide a particular coordinate normal form for every one of them.

For a finite diagonal system, the global fiber is a dependent function type: one solution in each coordinate. Products of finite cardinalities give the count, while one inadmissible coordinate makes the product type empty.

Source: KomorebiVerification/PaperII/SmithDyadic.lean.

```
theorem smithZMod_global_conditional_count {p depth : ℕ} [NeZero (p ^ depth)] (hp : 1 < p)
    (weight : κ → ℕ) (target : κ → ZMod (p ^ depth)) :
    Fintype.card (GlobalFiber (fun i => SmithZModFiber p (weight i) depth (target i))) =
      if ∀ i, SmithZModAdmissible p (weight i) depth (target i)
      then p ^ ∑ i, min (weight i) depth else 0 := by
  classical
  exact smith_conditional_product_count
    (fiber := fun i => SmithZModFiber p (weight i) depth (target i))
    (admissible := fun i => SmithZModAdmissible p (weight i) depth (target i))
    p depth weight (fun i => smithZModCoordinateCertificate hp (target i))
```

If all coordinates are admissible, the cardinality is $p^{\sum_i \min(s_i, L)}$. Otherwise it is zero. At depth zero, the quotient has one element and the exponent is zero, consistent with the stated formula. A literal zero diagonal coefficient can be represented at precision L by any weight at least L, since p^s then vanishes in the quotient.

11.3 The abstract extension keeps its certificate boundary

The same module has separate `SmithSolvabilityCertificate` and `SmithCardinalityCertificate` structures. For arbitrary finite coordinate fibers, these supply the equivalence and count for each coordinate, with an externally chosen base q. The global product theorems then assemble them.

In a general discrete valuation ring with residue field of cardinality q, the familiar coordinate count has the analogous form. The formal component does not construct that DVR arithmetic or derive its cardinality theorem. Likewise it does not calculate Smith normal form of an input matrix, find matrices for a change of basis, or extract diagonal weights. The automatic result for row 18 begins with a diagonal system over the concrete modular quotient. The abstract extension begins with certificates.

11.4 Rational sign projectors and local denominator weight

Row 17 fixes the rational matrices

$$J = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \qquad P_+ = \begin{pmatrix} 1 & 1/2 \\ 0 & 0 \end{pmatrix}, \qquad P_- = \begin{pmatrix} 0 & -1/2 \\ 0 & 1 \end{pmatrix}.$$

Entrywise proofs establish $J^2 = I$, projector idempotence, mutual annihilation, and $P_+ + P_- = I$. The local integrality predicate says that a rational entry's reduced denominator is not divisible by p. Each projector has exact local denominator weight zero at every prime other than two and exact weight one at two.

The local word matters. Global integrality demands integer entries. Integrality at p permits denominators prime to p. A global denominator weight and a local denominator weight are defined separately in the source. The resultant of the sign factors is explicitly computed as $\mathrm{Res}(X-1, X+1) = 2$, and its dyadic valuation is matched to the exact local weights of these matrices.

This is a model with rational entries using valuations with natural number values. It does not construct $\mathbb{Q}_p$ or its valuation ring, identify arbitrary completed lattices, or prove a general equivalence between all rational integrality criteria and lattice preservation. Reduction modulo two applies to the characteristic factors. The rational matrices containing $1/2$ do not have an entrywise reduction modulo two.

11.5 The Smith phase identity has only algebraic scope

Row 22 checks the fractional expression `smithPhase`, obtained by pulling back the action, at phase -1. Over a field with $2 \neq 0$ and $z \neq 0$, its value is z^{-1}. The extra manuscript condition $z \neq -1$ is unnecessary for this algebraic identity and is removed by the formal theorem.

The result does not formalize the complex Smith disk, an interpretation on a Möbius domain, radius preservation, or physical impedance. In the traditional companion, see the subsection on Smith chart coordinates and its unnumbered phase identity. The subsection counter is not an equation number.

12 Typed defects: domain and codomain stay visible

Rows 19–21 keep the domain and codomain explicit. Let X_f and X_c be fine and coarse modules, $C : X_f \to X_c$, and let T_f and T_c act on their respective spaces. The defect for one step is

$$\Delta = CT_f - T_cC : X_f \longrightarrow X_c.$$

The source does not assume that the two spaces coincide.

Source: `KomorebiVerification/PaperII/TypedLinearDefects.lean`.

```
variable {R : Type u} [Semiring R]
variable {Xf : Type v} {Xc : Type w}
variable [AddCommGroup Xf] [AddCommGroup Xc]
variable [Module R Xf] [Module R Xc]

/-! ## Typed power defects -/

/-- The typed one-step defect `C ∘ Tf - Tc ∘ C`. -/
def defect (C : Xf →ₗ[R] Xc)
    (Tf : Module.End R Xf) (Tc : Module.End R Xc) : Xf →ₗ[R] Xc :=
  C.comp Tf - Tc.comp C
```

Notice that the scalar type is a semiring while the spaces have additive group structures, allowing subtraction of maps. Scalar subtraction is not needed for the basic power identity. This is one example of why the complete type is more informative than a loose description such as “linear algebra over a ring.”

12.1 The telescoping mechanism

Define $E_n = CT_f^n - T_c^nC$. The induction uses

$$E_{n+1} = T_cE_n + \Delta T_f^n.$$

Expanding E_n gives

$$CT_f^n - T_c^nC = \sum_{k=0}^{n-1} T_c^{n-1-k}\Delta T_f^k.$$

Every summand is a map $X_f \to X_c$: the right power acts in X_f, the defect crosses to X_c, and the left power acts there. Treating C as an endomorphism would hide this type constraint.

The theorem for all powers includes $n = 0$. The left side is $C - C = 0$ and the finite sum is empty. The theorem for positive powers is a wrapper around this stronger statement. In the successor proof, `Finset.sum_range_succ` separates the new last term. Arithmetic of natural numbers verifies the shifted exponents. The exponent $n - 1 - k$ occurs only at $k < n$, where it has the intended value.

If the defect for one step is zero, every term in the sum vanishes.

Source: `KomorebiVerification/PaperII/TypedLinearDefects.lean`.

```
theorem powers_compatible_of_defect_eq_zero
    (C : Xf →ₗ[R] Xc) (Tf : Module.End R Xf) (Tc : Module.End R Xc)
    (hdefect : defect C Tf Tc = 0) (n : ℕ) :
    C.comp (Tf ^ n) = (Tc ^ n).comp C := by
  apply sub_eq_zero.mp
  rw [power_defect_identity_all]
  simp [hdefect]
```

This extends to finite polynomial functional calculus by summing the corresponding power identities with scalar coefficients. The definition of polynomial action explicitly carries the commutative scalar assumptions needed for those coefficients to act centrally. The result does not establish holomorphic, exponential, or other infinite functional calculus without further analytic work.

12.2 The operator norm estimate

In the branch for continuous linear maps, the same finite identity yields

$$\|CT_f^n - T_c^n C\| \le \|\Delta\| \sum_{k=0}^{n-1} \|T_c\|^{n-1-k} \|T_f\|^k.$$

The triangle inequality bounds the norm of the sum. Submultiplicativity bounds each composition, and the endomorphism power estimate bounds each power. Completeness of the spaces is unnecessary because this is a finite sum.

If both operator norms are at most ρ, the component derives the following majorant

$$\|CT_f^n - T_c^n C\| \le n\rho^{n-1}\|\Delta\|.$$

The source states the endpoint for natural exponents carefully, so the theorem for all n also includes zero. The hypotheses supply ρ and the two norm bounds. A bound on the spectral radius is not interchangeable with an operator norm bound here, and the majorant is not asserted to be sharp.

For example, exact intertwining $\Delta = 0$ forces every defect for finite powers to vanish. A small nonzero defect may still be amplified by large powers. The theorem gives an upper bound with the stated norm data. It does not identify the actual error for a particular trajectory or nonlinear dynamics.

12.3 Resolvents from the inverse equations actually used

Let R_f and R_c be supplied inverse candidates for the scalar shifts. The continuous theorem needs only

$$(zI - T_f)R_f = I, \qquad R_c(zI - T_c) = I.$$

The fine witness is a right inverse. The coarse witness is a left inverse. They are sufficient for the ordered sandwich calculation

$$CR_f - R_cC = R_c\Delta R_f.$$

Source: KomorebiVerification/PaperII/TypedLinearDefects.lean.

```
theorem continuous_scalarShift_resolvent_identity
    (z : 𝕜) (C : Ef →L[𝕜] Ec)
    (Tf Rf : Ef →L[𝕜] Ef) (Tc Rc : Ec →L[𝕜] Ec)
    (hf : (continuousScalarShift z Tf).comp Rf = 1)
    (hc : Rc.comp (continuousScalarShift z Tc) = 1) :
    C.comp Rf - Rc.comp C =
      (Rc.comp (continuousDefect C Tf Tc)).comp Rf := by
  calc
    C.comp Rf - Rc.comp C =
        (Rc.comp
          ((continuousScalarShift z Tc).comp C -
            C.comp (continuousScalarShift z Tf))).comp Rf :=
      continuous_inverse_sandwich_identity
        (continuousScalarShift z Tc) Rc
        (continuousScalarShift z Tf) Rf C hc hf
    _ = (Rc.comp (continuousDefect C Tf Tc)).comp Rf := by
      rw [continuousScalarShift_comp_sub_comp_continuousScalarShift]
```

The source first applies a general sandwich identity for inverse candidates. The difference of shifted compositions simplifies to $CT_f - T_cC$ because the two scalar terms cancel. The sign is therefore positive on the right for this convention of Δ. Reversing the defect definition would change that sign.

Applying the operator norm gives

$$\|CR_f - R_cC\| \le \|R_c\|\,\|\Delta\|\,\|R_f\|.$$

No theorem here constructs the inverse candidates or proves that z lies in a resolvent set. The algebraic typed branch uses explicit structures certifying inverse equations on both sides. The continuous branch needs only the two inverse equations displayed above. A user should choose the theorem matching the available data rather than infer inverse existence from the word “resolvent.”

12.4 What remains in the written companion

The mathematical text also proves holomorphic functional calculus, exponential, and multilevel defect identities with their own analytic or compositional hypotheses. The preserved Lean map records finite polynomial compatibility, the power estimates, and the resolvent identities with inverse witnesses. Those additional results are outside this inherited map. The chapter-wide correspondence and the analytic constructions are recorded in Section 1.

13 Replaying the formal project

Reproduction has two separate layers. The lightweight guide checks confirm that the supplied source matches its checksum manifest, that all mapped names can be found lexically, and that the printed

excerpts are exact slices of those files. The Lean replay builds the project, resolves names in the compiled environment, and traverses axiom dependencies. Only the latter executes Lean.

13.1 Start with the preserved pins

`Use a writable copy of the formal source directory.`
Keep lean-toolchain, lakefile.toml and lake-manifest.json together. The source records specify the compiler and dependency pins.

A working Lean/Elan installation, network access for dependencies or a suitable local cache, and sufficient disk space are prerequisites. The source records provide the replay procedure.

The cache command obtains compiled dependency artifacts. The build checks the aggregate library against the pinned environment. The Python map checker parses exactly one specified theorem map, expands its namespace prefixes, requires rows 1 through 25, and writes a temporary Lean command that checks every mapped name in the compiled environment. The script removes that temporary file afterward. Its environment check is stronger than this guide's lexical source lookup.

The final command separately executes the audit that rejects violations. Warning escalation is part of the documented invocation. Preserve the exit status and full output of each command. A report written after the fact should not substitute for those transcripts in a new replay.

13.2 The audit gate and its trust boundary

The audit has two parts. `#print axioms` displays dependencies for a representative public sample. A subsequent command traverses every compiled declaration in the imported `Komorebi` namespace and rejects unsafe entries or dependencies outside the allowed basis.

Source: `KomorebiVerification/PaperII/KernelAudit.lean`.

```
run_cmd do
  let env ← getEnv
  let mut audited : Nat := 0
  for (name, info) in env.checked.get.constants.toList do
    if Name.isPrefixOf `Komorebi name then
      audited := audited + 1
      if info.isUnsafe then
        throwError "unsafe project declaration: {name}"
      let axioms ← Lean.collectAxioms name
      for axiomName in axioms do
        unless axiomName == ``propext || axiomName == ``Classical.choice ||
            axiomName == ``Quot.sound do
          throwError "unexpected axiom {axiomName} in project declaration {name}"
  if audited == 0 then
```

```
    throwError "kernel audit found no Komorebi declarations"
  logInfo m!"FAIL-CLOSED KERNEL AUDIT PASS: {audited} Komorebi declarations; allowed axioms only:
   ↪ propext, Classical.choice, Quot.sound"
```

The approved foundations are `propext`, `Classical.choice`, and `Quot.sound`. The gate rejects any other axiom dependency, including `sorryAx`, and explicitly fails if no project declaration was inspected. A successful audit is a property of the imported compiled environment at that run. It is not inferred solely from the absence of the word `sorry` in source text.

The sample of printed axioms is not the complete traversal. Conversely, the namespace count includes generated declarations and helper entries, so it is not a count of independent mathematical theorems. The map checker and namespace gate also do not decide whether a theorem's type faithfully represents a prose theorem. That requires scope comparison such as the canonical map and this guide.

13.3 A minimal session for source inspection

After a successful build, create a scratch Lean file beside the project's aggregate import. The following inspection commands ask Lean to display two representative declarations and the axiom dependencies of the automatic rationality result.

```
import KomorebiVerification.PaperII
#check Komorebi.PaperII.NestedRefinement.coordinateRefinementEquiv
#check Komorebi.PaperII.SmithDyadic.smithZMod_global_conditional_count
#print axioms
 ↪ Komorebi.PaperII.TransferZeta.PositiveWeightedMultigraph.finite_graph_pathZeta_rationality
```

Run it with `lake env lean YourInspection.lean`. These are suggested inspection commands, not a new executed transcript. Use `#print` or editor hover to inspect elaborated types, including inferred section parameters and typeclass assumptions. The printed source excerpt alone may omit surrounding binders. The full declaration type is decisive.

When applying a theorem, first fix the intended scalar type and domain/codomain. Then assemble the structural object and prove its fields. For a weighted inverse, this means the filtration, map q, precision certificate, diagonal inverse, and edge expansion. For a finite graph, it means finite vertex/edge types, endpoint maps, and positive weights. For a Smith system, it means a diagonal weight function and modular target. This preparation makes failures informative: a missing instance differs from a missing mathematical witness.

13.4 Common failure modes

Symptom	What to check
Toolchain mismatch	Read lean-toolchain. Use the compiler specified by lean-toolchain.
Missing imported module	Confirm dependencies and cache availability, then build from the Lean project root.
Unknown mapped name	Run the map checker only after a successful aggregate build. Verify source and manifest identity.
Unsolved typeclass	Inspect the theorem's complete type and the scalar/module, finite-type, or normed-space assumptions.

Symptom	What to check
Missing inverse proof	Supply the actual equation required by the chosen theorem. Naming a candidate does not prove it invertible.
Unexpected axiom	Preserve the failing declaration and transcript. Do not weaken the allowed-axiom gate to make the run pass.
Different namespace count	Compare toolchain, dependencies, source hashes, and imports before treating the run as a replay of the recorded source.
Successful Python tests	Interpret them as tests of the Python component. They are not Lean kernel evidence.

A new successful run should record the source hashes, dependency pins, commands, exit codes, output hashes, and environment. If any formal source changes, the old report no longer supplies execution evidence for that changed source, even if the filenames remain the same.

14 Evidence retained and evidence newly checked

The foundational `VERIFICATION_REPORT.md` reports successful aggregate build, theorem-map resolution of 160 unique declarations, and a successful scan of 1,034 compiled namespace entries with a gate that rejects violations. It records the approved axiom basis and states that no audited declaration reports `sorryAx`. These are the claims of that report.

The inspected canonical component contains no raw transcript of those Lean commands and no log hash attestation for each command. Consequently the foundational component alone does not establish a fresh compiler replay, a fresh environment scan, or an identity chain to missing transcripts. Execution evidence for the chapter-wide correspondence is reported separately in Section 1. The report itself is checksummed, so its preservation is verifiable. Checksumming a report does not independently establish its execution claims.

The new lightweight check establishes the following facts directly from the delivered files:

1. All 22 files under the preserved formal component match their entries in the archived `SHA256SUMS.txt` manifest.
2. The same 22 files match the separately preserved baseline copy in this editing workspace.
3. The normative map has exactly rows 1–25 and 160 distinct expanded declaration names, with 171 occurrences across rows.
4. Every mapped name has a lexical declaration location in the preserved source. This is navigation evidence, not compiled name resolution.
5. The 15 Lean source files contain 5,603 physical lines. A scan that accounts for comments and strings finds no active token from the documented placeholder/unsafe source policy list.
6. Every exact excerpt matches its recorded inclusive source slice. Each source and excerpt has a SHA-256 digest.

The scan policy checks active occurrences of `sorry`, `admit`, `sorryAx`, `axiom`, `unsafe`, `native_decide`, and named shortcuts for dynamic declaration. It strips nested block comments, line comments, and strings while retaining line positions. This is a useful static check, not a replacement for Lean elaboration or the environment's transitive axiom inspection.

The guide audit records PDF and static source checks, including lean_compiler_run: false. PDF checks and Lean replay are separate evidence records. A document build verifies rendering and navigation. Formal coverage depends on the exact source types and their own compiler evidence.

14.1 A practical rule for reporting success

Report the strongest evidence actually obtained. After running only the data generator, say that static source identity and map reconciliation passed. After a new successful Lean replay, cite that replay's own logs and environment. When quoting the preserved count of 1,034 entries, attribute it to the historical report. When using an expanded traditional theorem, identify its ordinary written proof unless a precise Lean target with matching hypotheses has been supplied.

This discipline makes the component usable: readers can trust an automatic construction where it exists, supply a certificate where it is required, and locate the written mathematics that remains beyond the formal contract.

15 The inherited contract of 25 rows

The following map preserves the foundational contract's classifications, checked scopes, and explicit boundaries. The targets use the namespace abbreviations below. These abbreviations are only typography: the companion JSON expands every name and records its source file and line. A declaration may occur in several rows. There are 171 occurrences and 160 distinct names. Definitions, structures, equivalences, and theorems all count as declaration targets.

Prefix	Full namespace
PC	Komorebi.Foundation.LinearProjectorFamily
NR	Komorebi.PaperII.NestedRefinement
R	Komorebi.Residue
WT	Komorebi.PaperII.WeightedTransfer
TZ	Komorebi.PaperII.TransferZeta.PositiveWeightedMultigraph
TZC	Komorebi.PaperII.TransferZeta.PositiveWeightedMultigraph.RationalCertificate
RS	Komorebi.PaperII.ResultantSeparation
SD	Komorebi.PaperII.SmithDyadic
LD	Komorebi.PaperII.LinearDefects
TLD	Komorebi.PaperII.TypedLinearDefects

15.1 Row 1: Refinement map

Classification: Exact.

Declaration targets

`NR.children`

`NR.mem_children`

`NR.Refines`

Checked scope. For finite complete projector families over a semiring module, `Refines` records a surjective parent map and equality of every coarse projector with the finite sum over its fine children.

Boundary. No refinement is constructed from arbitrary projector families; no infinite or topological sum is asserted.

15.2 Row 2: Nested refinement theorem

Classification: Stronger.

Declaration targets

`NR.finitePacket_decomposition_unique`

`NR.finitePacket_decomposition_ext`

`NR.Refines.coarse_comp_fine_eq_self`

`NR.Refines.coarse_comp_fine_eq_zero`

`NR.Refines.fine_comp_coarse_eq_self`

`NR.Refines.fine_comp_coarse_eq_zero`

`NR.Refines.coarse_packet_reconstruction`

`NR.Refines.child_decomposition_unique`

`NR.Refines.trans`

Checked scope. Parent/child absorption holds in both composition orders and refinements compose. More generally, any finite sum of vectors lying in mutually orthogonal packet ranges has uniquely determined components, without a refinement or completeness hypothesis.

Boundary. Uniqueness is only asserted on the selected finite set and requires each component to lie in its stated projector range. All index types and sums are finite.

15.3 Row 3: Exact coordinate refinement

Classification: Stronger.

Declaration targets

`NR.refineCoordinates`

`NR.coarsenCoordinates`

`NR.refineCoordinates_self`

`NR.refineCoordinates_comp`

`NR.coordinateRefinementEquiv_trans`

`NR.Refines.refineCoordinates_component`

`NR.Refines.coarsenCoordinates_component`

Checked scope. Analysis after synthesis defines cross-level linear maps for any two finite complete projector families; these maps obey identity and cocycle laws through every intermediate family. A `Refines` witness gives the parent/child component formulas.

Boundary. These are coherent changes of exact coordinates, not truncation or dimensional reduction operators; the cocycle uses exact completeness at each finite level.

15.4 Row 4: Perfect reconstruction across levels

Classification: Stronger.

Declaration targets

`NR.coarsenCoordinates_refineCoordinates`

`NR.refineCoordinates_coarsenCoordinates`

`NR.coordinateRefinementEquiv`

`NR.coordinateOperator`

`NR.coordinateOperator_comp`

`NR.coordinateOperator_naturality`

Checked scope. Cross-level refinement and coarsening are mutual inverses over a semiring module. The typed coordinate representation of every linear map preserves identities and composition and commutes with independent exact coordinate changes on source and target.

Boundary. The inverse laws need completeness of the projector families but no parent map. This is finite algebraic naturality, not convergence for an infinite packet tower.

15.5 Row 5: Transfer-block refinement

Classification: Stronger.

Declaration targets

`NR.transferBlock_refinement_between`

`NR.transferBlock_refinement`

`NR.transferBlock_refinement_and_reconstruction`

Checked scope. Source and target projector families may live on different modules and refine independently; every coarse linear-map block is the double finite sum of its fine descendant blocks. The direct traceability wrapper also records reconstruction of the same map from all coarse blocks and from all fine blocks. The endomorphism formula is a specialization.

Boundary. No sparse extraction algorithm, infinite block matrix, or norm convergence is proved here.

15.6 Row 6: Residue projector algebra

Classification: Exact.

Declaration targets

`R.residueProj_idempotent`

`R.residueProj_orthogonal`

`R.residueProj_complete`

Checked scope. For every positive modulus, coefficient residue selectors on `Nat -> A` are idempotent, mutually annihilating, and complete over an arbitrary semiring and module.

Boundary. This is the division-free coefficient selector, not a character-average formula.

15.7 Row 7: Root-of-unity realization

Classification: Stronger / certificate-explicit.

Declaration targets

`R.ringFourierResidueKernel`

`R.ringFourierKernel_eq_residueKernel`

`R.FiniteRingFourierOrthogonalityCertificate`

`R.FiniteRingFourierOrthogonalityCertificate.toRingCertificate`

`R.ringFourierAverage`

`R.ringFourierAverage_eq_residueProj_of_finite_certificate`

`R.primitiveRoot_fourierOrthogonalityCertificate`

`R.fourierAverage_eq_residueProj_of_primitiveRoot`

Checked scope. Over any commutative semiring, `omega^N = 1` reduces the arbitrary coefficient address to a finite residue address. Consequently an explicit finite `N x N` orthogonality table identifies the division-free Fourier average with the residue selector. Over a field, a positive-order primitive root automatically supplies the normalized inverse-root certificate.

Boundary. The general semiring branch still requires the finite orthogonality table; root-of-unity order alone is not enough in rings with zero divisors. The primitive-root construction is field-only.

15.8 Row 8: Polyphase refinement

Classification: Exact.

Declaration targets

`R.polyphase_refinement`

Checked scope. For positive `N` and `k`, the modulus-`N` selector is the finite sum of exactly those modulus-`N*k` selectors that merge to it, over any semiring module.

Boundary. The theorem concerns exact coefficient masks; it gives no approximation or discarded-packet estimate.

15.9 Row 9: Dyadic-odd packet factorization

Classification: Exact.

Declaration targets

`R.crtEquiv`

`R.crtComponents_bijective`

`R.residueProj_crt`

`R.residueProj_crt_comm`

`R.dyadicOddEquiv`

`R.dyadicOddComponents_bijective`

`R.residueProj_dyadicOdd`

Checked scope. Positive coprime moduli give a bijection of residue labels and the composite selector in both composition orders. For positive odd `M`, this specializes to `2^a * M`.

Boundary. This is coordinate CRT factorization. It does not identify the denominator or lifting behavior of spectral projectors.

15.10 Row 10: Certified weighted inverse

Classification: Stronger / certificate-explicit.

Declaration targets

`WT.TwoSidedInverse`

`WT.finiteGeometricInverse_spec_of_pow_eq_zero`

`WT.diagonalTransferInverse_spec`

`WT.analyticGeometricInverse_spec`

`WT.analyticDiagonalTransferInverse_spec`

Checked scope. In a possibly noncommutative ring, an explicit two-sided inverse for `D` and nilpotence of the normalized perturbation `Dinv*H` give a finite two-sided inverse of `D+H`. In a complete normed ring, `norm (Dinv*H) < 1` gives the separate infinite Neumann inverse.

Boundary. The manuscript perturbation is `H`; the Lean declaration's parameter name `K` has no fraction-field meaning. There is no unrestricted convergence theorem for an abstract complete filtered ring. The analytic candidate is noncomputable.

15.11 Row 11: Finite precision and weighted pruning

Classification: Stronger / certificate-explicit.

Declaration targets

`WT.MultiplicativeFiltration`

`WT.PrecisionCertificate`

`WT.retainedFixedWordSum`

`WT.prunedWordInverse`

`WT.map_finiteGeometricInverse_eq_prunedWordInverse`

`WT.minWeight_mul_length_le_wordWeight`

`WT.map_sum_pow_eq_zero_at_scaled_cutoff`

`WT.prunedWordInverse_spec_of_minWeight`

`WT.finitePrecisionPrunedInverse_of_weightedEdges`

`WT.finitePrecisionPrunedDiagonalTransferInverse_of_minWeight`

`WT.finitePrecisionPrunedDiagonalTransferInverse_of_weightedEdges`

Checked scope. The mapped finite geometric inverse equals an explicit noncommutative word sum retaining exactly the words of cumulative weight below `L`. If every edge weight is at least `m` and `L <= m*N`, the `N`th mapped power vanishes and the pruned word sum is a two-sided inverse; the corresponding diagonal-plus-transfer formula follows from supplied `Dinv` and `Dinv*H = sum edge`.

Boundary. The filtration, precision quotient map, edge-membership data, diagonal inverse, and edge expansion remain inputs. No DVR quotient, quotient map, basis-level modular matrix routine, or extracted pruning implementation is constructed.

15.12 Row 12: Finite-graph transfer-zeta rationality

Classification: Automatic.

Declaration targets

`TZ.coefficientAlgorithm_correct`

`TZ.coefficientAlgorithm_firstEdge`

`TZ.weightedAdjacency`

`TZ.transferMatrix`

`TZ.transferDenominator`

`TZ.transferNumerator`

`TZ.transferDenominator_coeff_zero`

`TZ.pathZeta_eq_matrixPathZeta`

`TZ.transferDenominator_mul_pathZeta`

`TZ.finite_graph_pathZeta_rationality`

`TZ.finite_graph_transferZeta_trace`

`TZC.certificate_explicit_rationality`

Checked scope. For every finite positive-weight directed multigraph and source/target pair, finite walk enumeration satisfies the first-edge recurrence. Lean constructs `Q = det(I-A(X))` and `P` as the corresponding adjugate entry, proves `Q(0)=1`, identifies the independently enumerated `pathZeta` with the matrix inverse entry, and proves `Q*Z=P` without a rationality certificate. A direct wrapper packages

the manuscript's enumeration, recurrence, normalization, nonvanishing, and rational identity in one declaration. The older coefficient-certificate route remains available.

Boundary. Walks may repeat vertices and edges; no simple-path restriction is imposed. Parallel edges are distinct edge values. Determinant and adjugate are noncomputable Lean definitions here; no denominator cancellation, minimality, extracted symbolic implementation, or analytic convergence claim is made.

15.13 Row 13: Linear recurrence

Classification: Automatic.

Declaration targets

`TZ.finite_graph_constant_coefficient_recurrence`

`TZC.constant_coefficient_recurrence`

Checked scope. The automatically constructed determinant denominator gives an antidiagonal constant-coefficient recurrence for every `n` beyond the degree of the canonical adjugate numerator. A separately supplied `P,Q` certificate yields the same form through the legacy route.

Boundary. The canonical recurrence is not claimed minimal, and no recurrence is guessed or certified from finitely many sampled coefficients alone.

15.14 Row 14: Evaluated Bezout spectral projector

Classification: Stronger / certificate-explicit.

Declaration targets

`RS.EvaluatedBezoutCertificate`

`RS.separatorNumerator_eq_scalar_sub_of_bezout`

`RS.separatorNumerator_mul_self_of_bezout`

`RS.separator_idempotent_of_bezout`

`RS.separatorNumerator_apply_of_mem_ker_F_of_bezout`

`RS.separator_apply_of_mem_ker_F_of_bezout`

`RS.separator_apply_of_mem_ker_G`

`RS.scalar_mul_separator_of_ne_zero`

`RS.scalar_mul_separator_preservesLattice_of_ne_zero`

Checked scope. Over a field, the evaluated equality `U*F + V*G = r*1`, `F*G=0`, and only the two commutations actually used prove the denominator-free identity `(VG)^2=r(VG)`; adding `r!=0` gives idempotence of `r^-1*VG`. The kernel actions need only the displayed evaluated Bezout or kernel hypotheses, and lattice preservation of the numerator follows when `V,G` preserve the lattice.

Boundary. The evaluated Bezout equality and annihilation are supplied. No polynomial Bezout identity, resultant, characteristic factorization, or Cayley-Hamilton instance is constructed; nilpotence alone is not used to infer exact kernel action.

15.15 Row 15: Separation weight

Classification: Stronger / certificate-explicit.

Declaration targets

`RS.ScalarPreservesLattice`
`RS.IsLatticeUnit`
`RS.IsSeparationWeight`
`RS.IsSeparationWeight.mono`
`RS.separationWeight`
`RS.separationWeight_clears`
`RS.separationWeight_le_of_clears`
`RS.pow_smul_separator_eq_unit_inv_smul_separatorNumerator`
`RS.separator_isSeparationWeight_of_unit_mul_pow_eq_of_inv_preserves`
`RS.separator_hasFiniteSeparationWeight_of_eq_unit_mul_pow`
`RS.separator_separationWeight_le_of_eq_unit_mul_pow`

Checked scope. `IsSeparationWeight pi E lattice s` means that `(pi^s)E` preserves the chosen lattice; weights are upward closed when scalar multiplication by `pi` preserves it. If `r=u*pi^s`, then `pi ^s E=u^-1 VG`; preservation by the inverse unit and numerator proves clearing, while a bundled lattice-unit certificate gives existence and the least-weight upper bound `separationWeight<=s`.

Boundary. The unit action on the chosen lattice is explicit data. The least weight is noncomputable via `Nat.find`; no valuation, polynomial resultant construction, search procedure, or claim that the displayed exponent is minimal is supplied.

15.16 Row 16: Projector precision delay

Classification: Exact from certificate.

Declaration targets

`RS.CongruentModLattice`
`RS.projector_precision_delay`

Checked scope. For an arbitrary field scalar `pi`, additive lattice, and clearing-weight certificate, an explicit congruence witness at depth `m+s` maps to one at depth `m`.

Boundary. Congruence is witness-based scalar-lattice membership; no topology, completion, digit representation, or valuation API is required or concluded.

15.17 Row 17: Odd-adic versus dyadic sign behavior

Classification: Stronger / scoped realization.

Declaration targets

`SD.signInvolution_mul_self`
`SD.positiveSignProjector_idempotent`
`SD.negativeSignProjector_idempotent`
`SD.signProjectors_complete`
`SD.HasExactPLocalDenominatorWeight`

SD.positiveSignProjector_exact_pLocal_weight_zero_of_prime_ne_two

SD.negativeSignProjector_exact_pLocal_weight_zero_of_prime_ne_two

SD.signFactors_resultant

SD.signProjectors_resultant_dyadic_weight_endpoint

SD.padicValNat_two_at_odd_prime

SD.oddAdicVersusDyadicSignBehavior_trace

Checked scope. The displayed rational `2 x 2` involution and sign projectors are checked entrywise. Each projector has exact local denominator weight zero at every prime other than `2` and exact `2`-local weight one. Lean also computes `resultant(X-1,X+1)=2` and identifies its `2`-adic valuation with the exact local weight of both projectors. The direct trace wrapper combines both local regimes and the resultant statements.

Boundary. This is an explicit rational-entry, natural-valuation model. Reduction modulo `2` applies to the characteristic factors, not to the rational projector matrices containing `1/2`. Lean does not construct `Q_p`, its valuation ring, or a general equivalence between rational `p`-integrality and preservation of an arbitrary completed lattice.

15.18 Row 18: Concrete ZMod Smith-weighted congruence solve

Classification: Automatic at concrete scope / certificate-explicit extension.

Declaration targets

SD.smithZModMulHom

SD.SmithZModAdmissible

SD.smithZModFiber_nonempty_iff

SD.smithZModFiber_card_of_admissible

SD.smithZModCoordinateCertificate

SD.smithZMod_global_solvable_iff

SD.smithZMod_global_conditional_count

SD.SmithSolvabilityCertificate

SD.SmithCardinalityCertificate

SD.smith_conditional_product_count

Checked scope. For `p>1`, Lean proves directly over `ZMod(p^depth)` that `p^weight*x=target` is solvable exactly when `p^min(weight,depth)` divides `target.val`, and every admissible fibre has cardinality `p^min(weight,depth)`. It then proves the coordinatewise criterion and the exact conditional count `p^(sum min(weight,depth))` for finite diagonal systems, including cardinality zero when a coordinate is inadmissible. The split certificate API gives the analogous product statement for abstract finite fibres, with the residue cardinality supplied as `q`.

Boundary. The counted manuscript theorem is the concrete `ZMod(p^L)` result with `q=p`. The arbitrary-DVR, residue-size-`q` extension is classical/certificate-explicit, not derived by Lean. Diagonal weights and targets are inputs; Lean does not construct Smith normal form. A literal zero diagonal factor is represented at precision `L` by a weight at least `L`, or handled by an explicit abstract coordinate certificate.

15.19 Row 19: Power-defect identity

Classification: Stronger.

Declaration targets

`TLD.defect`

`TLD.power_defect_identity_all`

`TLD.power_defect_identity`

`TLD.powers_compatible_of_defect_eq_zero`

`TLD.polynomialAction`

`TLD.polynomialAction_compatible_of_defect_eq_zero`

`TLD.continuousPolynomialAction_compatible_of_defect_eq_zero`

`LD.power_defect_identity_all`

Checked scope. The primary theorem is typed for distinct fine and coarse semiring modules and proves the ordered telescoping identity for every natural `n`, including zero. Zero one-step defect gives exact compatibility with every power and every finite polynomial functional calculus; the continuous version is also proved.

Boundary. This is linear algebra and finite polynomial calculus; no nonlinear higher-jet, operator-tree defect expansion, or analytic functional calculus is asserted.

15.20 Row 20: Power-defect norm bound

Classification: Stronger.

Declaration targets

`TLD.continuousDefect`

`TLD.continuous_power_defect_identity_all`

`TLD.continuous_end_pow_norm_le`

`TLD.continuous_norm_power_defect_le`

`TLD.continuous_norm_power_defect_le_geometric_majorant`

`LD.norm_power_defect_le_raw`

`LD.norm_power_defect_le`

Checked scope. Over a nontrivially normed field, continuous maps between distinct normed fine and coarse spaces satisfy the exact finite-sum operator-norm bound for every natural `n`, including zero. If both dynamics have norm at most `rho`, Lean derives the closed majorant `n*rho^(n-1)*norm(defect)`. No completeness assumption on the spaces is needed.

Boundary. These are operator-norm estimates, not spectral-radius estimates, equality statements, or nonlinear defect bounds. The majorant takes `rho` and both norm inequalities as hypotheses.

15.21 Row 21: Resolvent-defect identity

Classification: Stronger / certificate-explicit.

Declaration targets

`TLD.TwoSidedInverse`

`TLD.scalarShift_resolvent_identity`

`TLD.resolvents_compatible_of_defect_eq_zero`

`TLD.continuousScalarShift`

`TLD.continuous_scalarShift_resolvent_identity`

`TLD.continuous_norm_resolvent_defect_le`

`TLD.continuous_resolvents_compatible_of_defect_eq_zero`

`LD.resolvent_defect_identity`

Checked scope. Algebraically, supplied two-sided inverse witnesses give the typed resolvent identity on distinct modules. For continuous maps, the proof needs only `(zI-Tf)Rf=I` and `Rc(zI-Tc)=I` and yields `norm(C Rf-Rc C) <= norm(Rc)*norm(defect)*norm(Rf)`; zero defect gives exact intertwining.

Boundary. All inverse equations are supplied witnesses. Lean does not construct resolvents, prove spectral membership, or infer inverse existence from analytic hypotheses.

15.22 Row 22: Smith phase inversion identity

Classification: Stronger / scoped realization.

Declaration targets

`SD.smithPhase`

`SD.smithPhase_neg_one_eq_inv`

Checked scope. Over any field in which `2` is nonzero, the pulled-back phase formula at `omega = -1` equals `z^-1` for `z != 0`; this includes `Q`, `R`, and `C` and does not need the manuscript's additional `z != -1` condition.

Boundary. Only the algebraic formula is checked. No Smith disk, Mobius-domain, radius-preservation, or physical impedance theorem is formalized.

15.23 Row 23: Residue analysis and synthesis algorithm

Classification: Specification.

Declaration targets

`PC.rangeSynthesis_rangeAnalysis`

`PC.rangeAnalysis_rangeSynthesis`

`PC.exactPacketCoordinates`

`NR.coordinateRefinementEquiv`

Checked scope. For finite complete projector families, range-restricted analysis and finite-sum synthesis are inverse linear maps; cross-level analysis/synthesis gives an exact coordinate equivalence.

Boundary. This is the mathematical contract behind the array routine, not extracted interleaving code or a proof about Python storage shapes.

15.24 Row 24: Weighted finite-precision inverse algorithm

Classification: Specification / certificate-explicit.

Declaration targets

`WT.finiteGeometricInverse`

`WT.prunedWordInverse`

`WT.map_finiteGeometricInverse_eq_prunedWordInverse`

`WT.le_mul_ceilDiv`

`WT.prunedWordInverse_spec_of_minWeight`

`WT.prunedDiagonalTransferInverse`

`WT.finitePrecisionPrunedDiagonalTransferInverse_of_minWeight`

`WT.finitePrecisionPrunedDiagonalTransferInverse_of_weightedEdges`

Checked scope. The bounded geometric loop maps exactly to an explicit signed sum of retained noncommutative words. For `m>0`, Lean proves `L <= m * ceil(L/m)`, so the pseudocode choice `N=ceil(L/m)` meets the cutoff premise. Filtration data and `L<=m*N` prove that the pruned expression is a two-sided inverse; supplied `Dinv` and a normalized-transfer edge expansion lift it to the explicit pruned diagonal-plus-transfer inverse after applying `q`.

Boundary. This is a mathematical specification. The manuscript calls the perturbation `H`; the Lean parameter name `K` has no fraction-field meaning. The result does not provide an extracted modular matrix inverse, basis representation, zero test, complexity bound, or proof of equivalence to the companion Python implementation.

15.25 Row 25: Transfer-zeta walk construction algorithm

Classification: Specification / automatic.

Declaration targets

`TZ.walkTuplesAtWeight`

`TZ.mem_walkTuplesAtWeight_iff`

`TZ.tuple_captured_iff`

`TZ.coefficientAlgorithm`

`TZ.coefficientAlgorithm_correct`

`TZ.exactWalkFirstEdgeEquiv`

`TZ.coefficientAlgorithm_firstEdge`

`TZ.pathZetaAlgorithm_trace`

`TZ.weightedAdjacency`

`TZ.weightedAdjacency_coeff`

`TZ.enumeratedPathZetaMatrix_eq_matrixPathZetaMatrix`

`TZ.pathZeta_eq_inverse_mul_transferNumerator`

`TZ.finite_graph_pathZeta_rationality`

`TZ.finite_graph_constant_coefficient_recurrence`

Checked scope. The coefficient loop is a finite `Finset` enumeration with exact capture and cardinality theorems. A direct trace wrapper pairs its semantic walk count with the first-edge recurrence. Finite matrix determinant/adjugate data are then constructed and proved equal to the enumerated se-

ries, yielding the normalized rational identity and eventual recurrence without a supplied rationality certificate.

Boundary. The objects counted are directed walks, with repetitions allowed. The Lean polynomial matrix definitions are noncomputable, and no extracted determinant/adjugate program, cancellation or minimal-denominator routine, performance bound, or analytic convergence theorem is claimed.

A Navigation in the mathematical text

The locators below use statement and section numbers in the Chapter II mathematical text. A shared label is a navigation aid. It does not extend the formal scope to additional claims in the surrounding section. In particular, the Smith phase display has no equation number.

Row	Canonical label	Chapter II locator
1	`def:refinement-map`	Statement 3.2
2	`thm:nested-refinement`	Statement 3.3
3	`def:coordinate-refinement`	Statement 3.4
4	`prop:perfect-reconstruction`	Statement 3.5
5	`thm:transfer-block-refinement`	Statement 3.8
6	`prop:residue-algebra`	Statement 4.2
7	`thm:root-of-unity`	Statement 4.3
8	`thm:polyphase-refinement`	Statement 4.6
9	`thm:dyadic-odd`	Statement 4.9
10	`thm:weighted-inverse`	Statement 5.10
11	`cor:finite-precision`	Statement 5.11
12	`thm:transfer-zeta`	Statement 6.6
13	`cor:zeta-recurrence`	Statement 6.7
14	`thm:bezout-projector`	Statement 7.5
15	`def:separation-weight`	Statement 7.6
16	`thm:precision-delay`	Statement 7.7
17	`prop:dyadic-sign`	Statement 9.3
18	`thm:smith-solve`	Statement 10.6
19	`thm:power-defect`	Statement 11.1
20	`cor:defect-norm`	Statement 11.3
21	`thm:resolvent-defect`	Statement 11.5
22	`eq:smith-inversion`	Section 12.3, Smith-phase identity
23	`alg:residue`	Statement B.1
24	`alg:weighted-inverse`	Statement B.2
25	`alg:path-zeta`	Statement B.3

Projector Resolved Seed Calculus III

The formal companion to Chapter III explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 1769

Return to Chapter III

How to use this volume

A guide to the finite algebraic core, its analytic continuation, and the complete 41-row finite contract. Read exact excerpts from the proofs, follow the direction of each map, distinguish exact coordinate transport from independent connection data, and locate every formal target.

Formal evidence. The verification record includes a pinned compilation, a compiled ownership and axiom audit, upstream-only kernel replay, and adverse controls. The opening chapter gives their exact scope. The preserved finite map still has 143 targets and its recorded namespace audit has 468 entries. Those counts describe the finite baseline, not the enlarged ownership closure.

The formal core and analytic continuation

The formal development combines finite algebraic constructions, exact nonlinear packet coordinates, and an analytic packet interface. Its verification record includes compilation in the pinned environment, an audit of the compiled ownership boundary and transitive axioms, replay of the owned logical declarations into an environment initially containing only upstream dependencies, and rejection controls.

The mathematical objects are the actual ranges

The input space for a jet is the dependent finite product of the submodules ran P_{α_i}. The output is ran Q_γ. These submodules carry their inherited norms. The continuous jet uses the actual inclusions and output restriction. Its underlying multilinear map is the previously defined algebraic range jet.

$$\|J_{\gamma|\alpha}B\| \le \|Q_\gamma\|\,\|B\|.$$

The product of norms of the input projectors belongs to the estimate for the ambient extension. It is unnecessary in the sharper bound for the literal restricted jet. Closed ranges and the continuous coordinate equivalence are proved for continuous projectors.

Source module	Formal content
`PacketCoordinates.lean`	Analysis and synthesis for a finite complete pairwise annihilating
`PacketDirectSum.lean`	idempotent family, with range-valued coordinates and the direct-sum comparison.
`NonlinearPacketCoordinates.lean`	Exact coordinate representation of arbitrary functions, composition and coordinate changes.
`MultilinearPacketJets.lean`	Complete finite multilinear expansion, literal range jets, and reconstruction. Mixed input labels are retained.
`BoundedPacketJetNorm.lean`	The ambient continuous multilinear extension and its product norm bound.
`ContinuousPacketJets.lean`	Closed ranges, continuous packet coordinates, literal continuous range jets, the sharper norm bound, and specialization to `iteratedFDeriv`.
`PacketJetComposition.lean`	Composition over all finite ordered partitions and intermediate labels, structural vanishing, and one-way support inclusion.

All paths in this table are relative to `formal/KomorebiPaperIII/`. The formal sources state their own field, norm, finiteness and regularity assumptions. The composition theorem assumes `ContDiffAt K n f x` and `ContDiffAt K n g (f x)`. The intermediate and output projectors are continuous. Input range inclusions are continuous without an additional continuity hypothesis on the input family.

What the derivative notation commits to

Mathlib defines `iteratedFDeriv` as a total function. Reconstructing its chosen value is unconditional. That identity alone does not assert differentiability of an arbitrary map. The composition theorem supplies local differentiability hypotheses explicitly. Its `OrderedFinpartition n` uses the canonical order of the blocks by their greatest elements. Each original slot is selected by the partition embedding, and every intermediate label assignment is summed. In the real or complex smooth setting, symmetry identifies this formula with the manuscript's unordered-partition notation.

The support theorem says that a nonzero composite jet has at least one decorated partition with a nonzero outer jet and all inner jets nonzero. It gives no converse. Constituent maps can compose to zero, and distinct terms can cancel. The formal composition identity also covers order zero through the unique empty partition. The positive-order manuscript formula requires no new nullary convention.

Compilation and kernel evidence

The verification run compiled **16 production modules** from an initially empty owned object directory. The compiled ownership audit recorded **608 owned declarations**. The clean kernel replay accepted **594 logical declarations**, with **14 generated runtime helpers** separately inventoried. The full build, ownership audit, kernel replay, and control suite all passed.

The exact compiler is Lean `4.34.0-rc2`, commit `6a10ac8c22beadecabdbb0919c2b50214762f91d`. Mathlib is pinned at `aeb72f1e9981b6d35b426b16dca7759be3e7ba06`. The machine-readable summary is `formal/verification/FINAL_VERIFICATION_SUMMARY.json`. The full command transcripts and exact source snapshots accompany every phase. The owned audit examines compiled declarations by module ownership, including names outside the main namespace. It rejects project axioms, unreviewed opaque or partial declarations, unsafe declarations, and transitive axioms outside the standard basis `propext`, `Classical.choice`, and `Quot.sound`.

Generated runtime helpers are inventoried separately from the logical declarations replayed by the kernel. A helper count is not a theorem count. The rejection controls include foreign owned axioms, opaque, unsafe and partial declarations, false arithmetic, a proof placeholder, and a deliberately corrupted replay declaration. A harmless foreign namespace is a positive control. The negative controls pass when the designated defect is rejected for the recorded reason.

The original 41-row map below remains the finite baseline map. Its 143 targets comprise 88 theorem declarations and 55 constructions. The recorded count of 468 namespace entries includes 229 theorem entries. These denominators are different from the enlarged ownership closure and from the manuscript's 98 result environments plus two classical standard facts.

Mathematical and finite evidence

The main text contains 98 result environments and two classical standard facts. Its finite and nonlinear coordinate arguments, the Section 17 analytic arguments, and the 59 later results plus two classical inputs have ordinary mathematical reviews under their stated hypotheses. The statement register identifies their status individually. These review counts do not assert complete Lean coverage.

The accompanying finite evidence records 18,630 exact assertions across three coordinate and calculus suites, 124 visual-model checks, and four rejected false controls. The computation audit specifies the tested ranges and repeated cases. These finite calculations support the examples and implementations. They do not replace the ordinary proofs or extend the formal contract.

The mathematical reading route is the main text, the analytic packet supplement, the 24 worked visual plates with 22 exact model figures, and this declaration-level guide. The supplement supplies the range-space, norm, Taylor, composition, and support arguments, including degenerate cases and the relation to the formal interface. The formal source distribution includes its exact sources, dependency pins, verification transcripts, mathematical reviews, computation scripts, statement register, and SHA-256 manifest.

Reproduce the formal verification

The archive supplies owned source, pins, receipts and scripts. Restore the pinned Lean distribution and the dependency identities using formal/README.md. Build in a separate working copy so the recorded manifest and transcripts stay intact. From formal/, run the four phases with a fresh object directory and unique labels:

```
python3 scripts/run_release_verification.py --lean /absolute/path/to/lean --phase full --label
 ↪  local_full --objects build/local_objects
python3 scripts/run_release_verification.py --lean /absolute/path/to/lean --phase audit --label
 ↪  local_audit --objects build/local_objects
python3 scripts/run_release_verification.py --lean /absolute/path/to/lean --phase replay --label
 ↪  local_replay --objects build/local_objects
python3 scripts/run_release_verification.py --lean /absolute/path/to/lean --phase controls --label
 ↪  local_controls --objects build/local_objects
```

The runner refuses a preexisting owned object directory for the full build. It records source hashes before and after compilation, the exact executable and command, output hashes, exit codes, and the expected diagnostics for adverse controls. Changing a dependency or toolchain creates a new verification event.

The remaining formal boundary

The manuscript's integral Taylor remainder, its factorial estimate, and the Stirling count of decorated terms are supplied as ordinary proofs. The 59 later results and two classical standard facts in Sections 18–27 were reviewed mathematically. Matching Lean formalizations for that expanded portion were not recovered.

The statement register distinguishes fully mapped statements, partial formal correspondence, inherited references, and ordinary proofs.

1 Reading the formal development

Chapter III turns three kinds of data into precise finite statements: directed packet operators, coordinate and connection transport, and local obstruction certificates. The preserved finite predecessor is small enough to read as a connected development. Its seven mathematical modules contain the author's constructions and proofs. An aggregate import exposes them together. A separate audit module checks the compiled namespace. The opening chapter describes the finite core and its analytic continuation.

This guide follows the definitions before using their theorem names. A declaration name is a precise destination, but its type determines what has actually been proved. The preserved finite map has 41 rows and 143 distinct targets. Of these, 88 are theorem declarations and 55 are constructions. A construction can be a structure, inductive type, definition, or abbreviation. The count does not mean 143 independent mathematical theorems.

1.1 Three levels of evidence

The finite algebraic verification report records a successful aggregate build of 3,076 jobs, the gate that checks parity with the theorem map, and a namespace audit of 468 declarations including 229 entries of kind `theoremInfo`. Chapter III also preserves the kernel audit transcript. That transcript contains one axiom report for each of the 143 mapped targets and the final namespace PASS line.

A static reconciliation matches all 41 map rows with their lexical source locations, all 143 explicit `#print axioms` directives, and all 143 transcript records. It checked the 16 files in the preserved formal component against the original checksum manifest. The theorem map and transcript hashes also match the values printed in the inherited verification report. These are static checks of the preserved evidence. They do not replace a Lean execution. The opening chapter separately records the compiler and replay runs.

The expanded mathematical text includes additional ordinary mathematical proofs. The finite algebraic formal contract stays fixed. A section of the main text may combine a formally covered finite result with a stronger ordinary theorem. The presence of a related Lean name does not make every statement in that section part of the contract.

1.2 A route through the source

Module	Reading objective
`EdgeOperators`	Orientation, broken-word vanishing, and finite walk expansion.
`EdgeExtensions`	Endpoint blocks, reachability, and heterogeneous factor types.
`Transport`	Exact frames, the minimal connection, and a nontrivial loop.
`CoordinateNaturality`	Exact coordinate representation of arbitrary functions.
`ReversibleConnections`	Dependent fibers, defect, holonomy, gauges, paths, and blocks.
`HolonomyBlocks`	The commutation hypothesis for packetwise gauge covariance.
`NonlinearObstructions`	Tree locality and candidate-specific obstruction certificates.

The name of a file and the namespace of its declarations need not coincide. For example, the basic packet resolution declarations occur at the end of `ReversibleConnections.lean` under `Komorebi.Pap`

`erIII.HolonomyBlocks`. The complete map gives both the fully qualified name and the file and line. Follow that location when searching a source checkout.

1.3 Notation that carries mathematical content

Lean writes a linear map as `M →ₗ[R] N` and a linear equivalence as `M ≃ₗ[R] N`. The latter includes an inverse and its inverse laws. An arbitrary equivalence is `X ≃ Y`. A function `X → Y` carries no linear structure. Replacing one of these types with another changes the claim.

Bracketed arguments such as `[Semiring R]` are typeclass hypotheses. They supply operations and laws, not merely notation. Braced arguments such as `\{R : Type u\}` are implicit parameters inferred from use. `Type u` allows the object to live in a universe. Different universe names allow different input, fiber, and coefficient types. `autoImplicit false` requires undeclared identifiers to be rejected rather than silently generalized.

`Fintype V` supplies finite enumeration. `DecidableEq V` supplies a decision procedure for equality. Some definitions use `classical` so that a finite sum can be filtered by a proposition without demanding a decision procedure supplied by the user. This logical choice does not turn an abstract graph theorem into an efficient graph algorithm.

Three composition conventions must remain distinct:

Written expression	Execution order
`B * A`	Matrix A acts first, then B.
`g.comp f`	Linear map f acts first, then g.
`e.trans f`	Equivalence e acts first, then f.

A calculation translated from matrices to `LinearEquiv.trans` therefore reverses the written composition pattern. The underlying maps still execute in the same order.

1.4 What a short proof is doing

`rfl` closes a definitional equality: the two sides reduce to the same expression. `ext x` or `LinearEquiv.ext` reduces equality of maps to equality at an arbitrary input. `simp` uses definitions and registered simplification rules, including inverse laws and zero laws. `rw` rewrites by an explicit equality. `induction` exposes a finite recursive object, such as a list or tree, and proves the conclusion from the smaller objects.

The printed excerpts are exact source selections, not standalone files. Implicit variables, local instances, namespaces, and earlier lemmas may be outside a selection. Every exact UTF-8 excerpt has a source path, inclusive line range, and hashes in `generated/excerpts.json`. Copy the adjacent `.lean.txt` file when exact tokens matter. The PDF display uses font fallbacks for otherwise unsupported characters.

2 Edges, words, and reachable packets

2.1 Graph orientation and finite enumeration

`PacketGraph V E` has two fields, `source : E → V` and `target : E → V`. Distinct values of `E` may share endpoints, so parallel edges are allowed. The structure itself does not carry a finite enumeration. The theorems that sum over vertices or edges request the appropriate `Fintype` instances separately. This makes the boundary visible in each declaration.

The elementary scalar block is the matrix whose only potentially nonzero entry is in target row and source column.

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
def packetMatrix [Zero R] [DecidableEq V]
    (source target : V) (coefficient : R) : Matrix V V R :=
  fun row column ↦ if row = target ∧ column = source then coefficient else 0

@[simp] theorem packetMatrix_apply [Zero R] [DecidableEq V]
    (source target row column : V) (coefficient : R) :
    packetMatrix source target coefficient row column =
      if row = target ∧ column = source then coefficient else 0 :=
  rfl
```

For a column vector x, this block sends the source coordinate to the target coordinate and multiplies it by the stored coefficient. `EdgeSystem G R` is a coefficient function `E → R`. It adds weights to graph edges without changing their endpoints.

If $e : a \to b$ has coefficient r and $f : b \to c$ has coefficient s, the product for the traversal e followed by f is $E_f E_e$, and its coefficient is sr. A semiring need not be commutative. That coefficient cannot in general be replaced by rs.

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
theorem edgeOp_mul_edgeOp_of_composable
    [Fintype V] [DecidableEq V] [Semiring R]
    (e f : E) (h : G.target e = G.source f) :
    edgeOp G S f * edgeOp G S e =
      packetMatrix (G.source e) (G.target f)
        (S.coefficient f * S.coefficient e) := by
  ext row column
  by_cases hrow : row = G.target f
  · by_cases hcolumn : column = G.source e
    · simp [edgeOp, packetMatrix, Matrix.mul_apply, h, hrow, hcolumn]
    · simp [edgeOp, packetMatrix, Matrix.mul_apply, h, hcolumn]
  · simp [edgeOp, packetMatrix, Matrix.mul_apply, hrow]
```

The proof examines one matrix entry. The sum over intermediate vertices has at most one surviving term after support conditions are imposed. If the interface does not match, the companion theorem `edgeOp_mul_edgeOp_eq_zero` shows every intermediate term is zero. The zero conclusion depends on support, not on an assumed numerical cancellation.

2.2 Lists record chronological traversal

`Composable G` accepts the empty list and every singleton. A longer list must satisfy equality between the target of one edge and the source of the next at each step. The evaluator then places the first listed edge on the right:

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
def wordOp [Fintype V] [DecidableEq V] [Semiring R]
    (G : PacketGraph V E) (S : EdgeSystem G R) : List E → Matrix V V R
  | [] => 1
  | e :: rest => wordOp G S rest * edgeOp G S e

@[simp] theorem wordOp_nil [Fintype V] [DecidableEq V] [Semiring R] :
    wordOp G S [] = 1 :=
  rfl

@[simp] theorem wordOp_singleton [Fintype V] [DecidableEq V] [Semiring R]
    (e : E) : wordOp G S [e] = edgeOp G S e := by
  simp [wordOp]
```

Thus `[e,f,g]` evaluates to $E_g E_f E_e$. The empty list evaluates to the identity matrix, including when the finite vertex type is empty. The formal definitions cover the empty case directly.

The concatenation law gives a quick way to check the orientation:

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
theorem wordOp_append [Fintype V] [DecidableEq V] [Semiring R]
    (first second : List E) :
    wordOp G S (first ++ second) = wordOp G S second * wordOp G S first := by
  induction first with
  | nil => simp
  | cons e rest ih =>
      simp [wordOp, ih, Matrix.mul_assoc]
```

The operator for `first ++ second` is the operator for `second` multiplied by the operator for `first`. The proof is list induction plus associativity. Its order is essential in a noncommutative coefficient system.

The next theorem removes every word with a broken interface, not just a break at the start.

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
theorem wordOp_eq_zero_of_not_composable
    [Fintype V] [DecidableEq V] [Semiring R]
    (word : List E) (h : ¬ Composable G word) :
    wordOp G S word = 0 := by
  induction word with
  | nil => simp [Composable] at h
  | cons e rest ih =>
      cases rest with
      | nil => simp [Composable] at h
      | cons f tail =>
```

```
          simp only [Composable, not_and_or] at h
          rcases h with hbreak | htail
          · exact wordOp_eq_zero_of_first_break G S e f tail hbreak
          · change wordOp G S (f :: tail) * edgeOp G S e = 0
            rw [ih htail]
            simp
```

The induction distinguishes a mismatch in the first pair from a mismatch in the tail. The former invokes the support result for two edges. The latter uses the induction hypothesis to make the tail operator zero. Empty and singleton words cannot satisfy the noncomposability hypothesis, so those branches eliminate a contradiction.

2.3 From a transfer power to a finite walk sum

The transfer operator is $T = \sum_{e \in E} E_e$. Four related objects are kept separate:

1. `wordExpansion` recursively multiplies on the left by T.
2. `tupleWordOp` converts a tuple `Fin n → E` to an execution list.
3. `explicitWordExpansion` sums over every tuple of length n.
4. `walkExpansion` keeps precisely the composable tuples.

At length zero there is one empty tuple, not zero tuples. Its contribution is the identity. This is why the explicit sum agrees with T^0.

To pass from length n to length $n + 1$, the proof separates a tuple into its first n entries and its final entry. `Fin.snocEquiv` supplies the finite indexing equivalence. Appending the final edge places its operator to the left of the prefix operator.

Source: `KomorebiPaperIII/EdgeOperators.lean`.

```
theorem explicitWordExpansion_succ
    [Fintype E] [Fintype V] [DecidableEq V] [Semiring R]
    (n : ℕ) :
    explicitWordExpansion G S (n + 1) =
      transferOp G S * explicitWordExpansion G S n := by
  unfold explicitWordExpansion
  rw [← (Fin.snocEquiv (fun _ : Fin (n + 1) ↦ E)).sum_comp]
  rw [Fintype.sum_prod_type]
  change (∑ final : E, ∑ prior : Fin n → E,
    tupleWordOp G S (n + 1) (Fin.snoc prior final)) = _
  simp only [tupleWordOp_snoc]
  unfold transferOp
  rw [Finset.sum_mul]
  simp_rw [Finset.mul_sum]
```

This is a finite distributivity argument. It uses neither an infinite geometric series nor a convergence limit. The proof then identifies the recursive expansion with the explicit tuple sum by induction on n.

Pruning the tuple sum uses the theorem for words with a broken interface, term by term:

Source: KomorebiPaperIII/EdgeOperators.lean.

```
theorem explicitWordExpansion_eq_walkExpansion
    [Fintype E] [Fintype V] [DecidableEq V] [Semiring R]
    (n : ℕ) :
    explicitWordExpansion G S n = walkExpansion G S n := by
  classical
  unfold explicitWordExpansion walkExpansion
  apply Finset.sum_congr rfl
  intro word _
  by_cases hwalk : Composable G (List.ofFn word)
  · simp [hwalk]
  · have hzero : tupleWordOp G S n word = 0 :=
      wordOp_eq_zero_of_not_composable G S (List.ofFn word) hwalk
    simp [hwalk, hzero]
```

When a tuple is composable, the filtered term is unchanged. When it is not, its unfiltered operator was already zero. Combining this equality with the recursive power identity gives `transferOp_pow_eq_walkExpansion`.

2.4 Endpoint coefficients retain the empty path

`reverseCoefficient` recursively multiplies coefficients in operator order. `terminalTarget` records the last endpoint of a nonempty word. `Connects G source target word` combines the endpoint conditions with composability. For an empty word it means `source = target`.

Source: KomorebiPaperIII/EdgeExtensions.lean.

```
theorem wordOp_apply_eq_if_connects
    [Fintype V] [DecidableEq V] [Semiring R]
    (word : List E) (source target : V) :
    wordOp G S word target source =
      if Connects G source target word then reverseCoefficient G S word else 0 := by
  cases word with
  | nil =>
      by_cases h : source = target
      · subst target
        simp [wordOp, Connects, reverseCoefficient]
      · have hreverse : target ≠ source := fun h' => h h'.symm
        simp [wordOp, Connects, h, hreverse]
  | cons first rest =>
      by_cases hwalk : Composable G (first :: rest)
      · rw [wordOp_eq_packetMatrix_of_composable G S first rest hwalk]
        simp [packetMatrix, Connects, hwalk, and_comm]
      · rw [wordOp_eq_zero_of_not_composable G S (first :: rest) hwalk]
        simp [Connects, hwalk]
```

This theorem includes both a nonempty composable word and the diagonal entry of the empty identity word. The endpoint sum is consequently valid for every natural length, including zero. The entry of T^n at the chosen endpoints is the sum of the reverse coefficients of exactly those words of length n connecting them.

2.5 What the support theorem implies

`Reachable G source target` means that some finite list connects the vertices. The empty list makes reachability reflexive. Absence of a walk of length n forces the corresponding entry of T^n to vanish. Absence of any walk gives this conclusion at every length.

The source also handles a seed supported on a specified set:

Source: `KomorebiPaperIII/EdgeExtensions.lean`.

```
theorem transferOp_pow_mulVec_apply_eq_zero_of_unreachable_support
    [Fintype E] [Fintype V] [DecidableEq V] [Semiring R]
    (n : ℕ) (x : V → R) (support : Set V) (target : V)
    (hx : ∀ source, source ∉ support → x source = 0)
    (hunreachable : ∀ source, source ∈ support → ¬ Reachable G source target) :
    ((transferOp G S ^ n) *ᵥ x) target = 0 := by
  rw [Matrix.mulVec_apply_eq_sum]
  apply Finset.sum_eq_zero
  intro source _
  by_cases hsource : source ∈ support
  · rw [transferOp_pow_apply_eq_zero_of_not_reachable G S n source target
      (hunreachable source hsource)]
    simp
  · rw [hx source hsource]
    simp
```

For each source vertex, the proof splits on membership in the support. Inside the support, the relevant transfer entry vanishes by unreachability. Outside it, the seed coordinate vanishes. Every term in the sum defining the matrix action is therefore zero.

The direction cannot be reversed without extra assumptions. A graph may contain a path whose coefficient is zero. Distinct paths may cancel over a ring. Thus a zero orbit coordinate does not by itself prove that the target is unreachable. The Lean theorem uses unreachability as a hypothesis and returns zero as a conclusion.

3 Heterogeneous products and exact coordinates

3.1 What varies in a heterogeneous path

There are two different kinds of heterogeneity. The matrix factor may change at every step while the packet index type stays fixed. Alternatively, the source, intermediate, and target packet types themselves may differ. The formal component includes a theorem for arbitrary finite length of the first kind and separate theorems for two and three steps of the second kind.

The statement for two steps displays all three packet types:

Source: `KomorebiPaperIII/EdgeExtensions.lean`.

```
theorem heterogeneousPacketPathExpansion_two
    {I0 : Type u0} {I1 : Type u1} {I2 : Type u2}
    [Fintype I1] [NonUnitalNonAssocSemiring R]
    (first : Matrix I1 I0 R) (second : Matrix I2 I1 R)
    (source : I0) (target : I2) :
    (second * first) target source =
      ∑ middle : I1, second target middle * first middle source := by
  rw [Matrix.mul_apply]
```

Only the intermediate type `I1` needs a finite enumeration. The source and target types are not enumerated in this entrywise identity. The coefficient hypothesis is `NonUnitalNonAssocSemiring R`. No multiplicative unit or associativity is needed for a single entry sum with two factors.

The statement for three steps introduces two finite intermediate types and a semiring:

Source: `KomorebiPaperIII/EdgeExtensions.lean`.

```
theorem heterogeneousPacketPathExpansion_three
    {I0 : Type u0} {I1 : Type u1} {I2 : Type u2} {I3 : Type u3}
    [Fintype I1] [Fintype I2] [Semiring R]
    (first : Matrix I1 I0 R) (second : Matrix I2 I1 R)
    (third : Matrix I3 I2 R) (source : I0) (target : I3) :
    ((third * second) * first) target source =
      ∑ middle2 : I2, ∑ middle1 : I1,
        (third target middle2 * second middle2 middle1) * first middle1 source := by
  rw [Matrix.mul_apply]
  simp_rw [Matrix.mul_apply, Finset.sum_mul]
  rw [Finset.sum_comm]
```

The parentheses on the coefficient product match `(third * second) * first`. The proof expands matrix multiplication twice, distributes multiplication across a finite sum, and exchanges the two finite sums. The packet index types remain separate throughout the calculation.

3.2 Arbitrary finite length on one packet type

`heterogeneousProduct` evaluates a list of square matrices in execution order. The head acts first and appears on the right. `heterogeneousPathSum` independently sums the intermediate packet at each recursive step. Its empty case is the identity coefficient, equal to one when source and target agree and zero otherwise.

Source: `KomorebiPaperIII/EdgeExtensions.lean`.

```
theorem heterogeneousPacketPathExpansion
    [Fintype V] [DecidableEq V] [Semiring R]
    (steps : List (Matrix V V R)) (source target : V) :
    heterogeneousProduct steps target source =
      heterogeneousPathSum steps source target := by
  induction steps generalizing source target with
  | nil =>
      by_cases h : source = target
      · subst target
        simp [heterogeneousProduct, heterogeneousPathSum]
      · have hreverse : target ≠ source := fun h' => h h'.symm
        simp [heterogeneousProduct, heterogeneousPathSum, hreverse]
  | cons step rest ih =>
      simp [heterogeneousProduct, heterogeneousPathSum, Matrix.mul_apply, ih]
```

The induction is on the list of factors. The base case is the matrix identity. The successor case is exactly the entry formula for matrix multiplication, with the induction hypothesis substituted into the remaining product. The theorem quantifies over every finite list, but all factors are `Matrix V V R`.

The mathematical text may treat a fully dependent chain with a new packet index type at every stage. That theorem lies outside the fixed Lean contract of 41 rows. Rows 13 and 14 make the distinction explicit: differing types at lengths two and three, varying factors on one common type at arbitrary finite length.

3.3 Declared support is a certificate

A `SupportedStep` stores an operator, a relation on source and target packets, and a proof that an entry is zero whenever the relation does not hold. The relation need not be the exact nonzero support. It can be a conservative relation of allowed packet transitions. `SupportedPath` says each successive step respects the corresponding relation.

Source: `KomorebiPaperIII/EdgeExtensions.lean`.

```
theorem heterogeneousPathSum_eq_zero_of_no_supportedPath
    [Fintype V] [DecidableEq V] [Semiring R]
    (steps : List (SupportedStep (V := V) (R := R))) (source target : V)
    (h : ¬ SupportedPath steps source target) :
    heterogeneousPathSum (steps.map SupportedStep.operator) source target = 0 := by
  induction steps generalizing source target with
  | nil =>
      have hreverse : target ≠ source := fun h' => h h'.symm
      simp [heterogeneousPathSum, hreverse]
  | cons step rest ih =>
```

```
      simp only [List.map_cons, heterogeneousPathSum]
      apply Finset.sum_eq_zero
      intro middle _
      by_cases hedge : step.relation source middle
      · have hrest : ¬ SupportedPath rest middle target := by
          intro hpath
          exact h ⟨middle, hedge, hpath⟩
        rw [ih middle target hrest]
        simp
      · rw [step.supported source middle hedge]
        simp
```

The proof splits at every candidate intermediate packet. If the first relation fails, the first matrix entry is zero by the stored certificate. If it holds, absence of an entire supported path forces absence of a supported tail, and the induction hypothesis makes the tail sum zero. This is useful even when exact nonzero support would be expensive to decide.

3.4 Exact frames point from coordinates to ambient space

In `Transport.lean`, a frame synthesizes an ambient vector from its coordinate representation. The homogeneous model uses an automorphism of one module, so source and target types look alike, but the direction still matters:

Source: `KomorebiPaperIII/Transport.lean`.

```
def exactTransport (frame : X → M ≃ₗ[R] M) (a b : X) : M ≃ₗ[R] M :=
  (frame a).trans (frame b).symm

@[simp] theorem exactTransport_apply (frame : X → M ≃ₗ[R] M)
    (a b : X) (x : M) :
    exactTransport frame a b x = (frame b).symm (frame a x) :=
  rfl
```

If Ψ_a is the frame, this is $\Psi_b^{-1}\Psi_a$. It first synthesizes through frame a, then analyzes through frame b. Cancellation of the intervening frame proves the cocycle law:

Source: `KomorebiPaperIII/Transport.lean`.

```
theorem exactTransport_cocycle (frame : X → M ≃ₗ[R] M) (a b c : X) :
    (exactTransport frame a b).trans (exactTransport frame b c) =
      exactTransport frame a c := by
  apply LinearEquiv.ext
  intro x
  simp [exactTransport]
```

The proof applies extensionality and simplifies an equivalence followed by its inverse. The results for self transport, return trips, triangles, and finite paths follow from the same cancellation. Exact transport obtained from one frame family cannot acquire a nontrivial loop merely by changing the path.

3.5 Exact coordinate systems point the other way

`CoordinateNaturality.lean` uses an equivalence `X ≃ Source` from an ambient type to a coordinate type. This is an analysis map, inverse in direction to the frame convention above.

Source: `KomorebiPaperIII/CoordinateNaturality.lean`.

```
def exactCoordinateTransport
    {X : Type u} {Source : Type v} {Target : Type w}
    (source : X ≃ Source) (target : X ≃ Target) : Source ≃ Target :=
  source.symm.trans target

@[simp] theorem exactCoordinateTransport_apply
    {X : Type u} {Source : Type v} {Target : Type w}
    (source : X ≃ Source) (target : X ≃ Target) (value : Source) :
    exactCoordinateTransport source target value = target (source.symm value) :=
  rfl
```

For analysis equivalences $\Phi_a : X \to A$ and $\Phi_b : X \to B$, the transport is $\Phi_b \Phi_a^{-1}$. Both conventions express the same operation of synthesizing and then analyzing. Check the direction of the supplied equivalence before comparing the formulas. This avoids an apparent inverse error.

> **Two directions for exact coordinates**
>
> A frame takes coordinates to ambient space. An analysis equivalence takes ambient data to coordinates. In both conventions, transport first returns to the ambient object and then enters the destination coordinates. The different written formulas reflect the different directions of the supplied equivalences.

An arbitrary function $f : X \to Y$ has coordinate representative

$$f_{a,r} = \Phi_r f \Phi_a^{-1}.$$

The type is an ordinary function `Source → Target`. It need not be linear, invertible, continuous, or differentiable. Composition is preserved exactly:

Source: `KomorebiPaperIII/CoordinateNaturality.lean`.

```
theorem exactCoordinateMap_comp
    {X : Type u} {Y : Type v} {Z : Type w}
    {CX : Type x} {CY : Type y} {CZ : Type z}
    (xCoordinate : X ≃ CX) (yCoordinate : Y ≃ CY)
    (zCoordinate : Z ≃ CZ) (first : X → Y) (second : Y → Z) :
    exactCoordinateMap xCoordinate zCoordinate (second ∘ first) =
      exactCoordinateMap yCoordinate zCoordinate second ∘
        exactCoordinateMap xCoordinate yCoordinate first := by
  funext value
  simp [exactCoordinateMap]
```

The middle coordinate analysis and synthesis cancel. The source proves equality of functions with `funext`, then simplifies at one value. No approximation is involved.

3.6 The naturality square is a typed equality

Source: `KomorebiPaperIII/CoordinateNaturality.lean`.

```
theorem exactCoordinateMap_naturality
    {X : Type u} {Y : Type v}
    {XA : Type w} {XB : Type x} {YR : Type y} {YS : Type z}
    (sourceA : X ≃ XA) (sourceB : X ≃ XB)
    (targetR : Y ≃ YR) (targetS : Y ≃ YS) (map : X → Y) :
    exactCoordinateTransport targetR targetS ∘
        exactCoordinateMap sourceA targetR map =
      exactCoordinateMap sourceB targetS map ∘
        exactCoordinateTransport sourceA sourceB := by
  funext value
  simp [exactCoordinateTransport, exactCoordinateMap, Function.comp_def]
```

On the left, represent the map in the first coordinate systems and then change target coordinates. On the right, change source coordinates first and then use the second representative. Both sides map `XA` to `YS`. Expanding them gives the same function.

This exact statement about sets already applies to a nonlinear function. It does not imply differentiability, a chain rule for jets, or convergence of a formal series. Those stronger properties require further structure and separate theorems. The formal contribution here is exact representation and naturality for arbitrary functions under exact equivalences.

4 Connections, defects, and holonomy

4.1 The minimal interface separates a choice from a theorem

The minimal `Connection R M X` stores an equivalence for each ordered pair of vertices, together with the identity law at equal endpoints.

Source: `KomorebiPaperIII/Transport.lean`.

```
structure Connection (R : Type u) (M : Type v) (X : Type w)
    [Semiring R] [AddCommMonoid M] [Module R M] where
  transport : X → X → M ≃ₗ[R] M
  transport_self : ∀ x, transport x x = LinearEquiv.refl R M
```

Neither reversal by inversion nor the cocycle equation is a field. Both could fail for an instance. This makes the distinction from exact frame transport substantive: exact transport has a prescribed construction that forces the cocycle law, while a connection carries separately chosen pair maps.

The minimal connection's defect is an equivalence. It compares the direct map with the route through an intermediate vertex by composing with an inverse:

Source: `KomorebiPaperIII/Transport.lean`.

```
def cocycleDefect (a b c : X) : M ≃ₗ[R] M :=
  (C.transport a c).symm.trans
    ((C.transport a b).trans (C.transport b c))
```

With ordinary function composition, this is

$$D_{a,b,c} = (\Gamma_{b,c}\Gamma_{a,b})\Gamma_{a,c}^{-1}.$$

It is based at the target coordinate space. In this multiplicative model, absence of a defect means that the comparison is the *identity*, rather than an additive zero map. Since the fibers here are homogeneous, the Lean type does not itself display which endpoint is the base. The definition fixes that interpretation.

4.2 A counterexample to automatic flatness in the source

`doubleEquiv` multiplies rational coordinates by two and divides by two for its inverse. The connection on three points chooses it for the directed edge $0 \to 1$ and chooses the identity for every other unequal pair.

Source: `KomorebiPaperIII/Transport.lean`.

```
def twistedThreePointConnection : Connection ℚ ℚ (Fin 3) where
  transport a b :=
    if a = b then LinearEquiv.refl ℚ ℚ
    else if a = (0 : Fin 3) ∧ b = (1 : Fin 3) then doubleEquiv
    else LinearEquiv.refl ℚ ℚ
  transport_self a := by simp
```

The loop $0 \to 1 \to 2 \to 0$ doubles the input one. Equality with the identity would therefore force $2 = 1$. The source proves nonidentity by applying the proposed equality of equivalences to that input:

Source: KomorebiPaperIII/Transport.lean.

```
theorem twisted_triangleHolonomy_ne_refl :
    twistedThreePointConnection.triangleHolonomy
        (0 : Fin 3) (1 : Fin 3) (2 : Fin 3) ≠ LinearEquiv.refl ℚ ℚ := by
  intro h
  have happly := congrArg (fun e : ℚ ≃ₗ[ℚ] ℚ ↦ e (1 : ℚ)) h
  norm_num [Connection.triangleHolonomy, twistedThreePointConnection,
    doubleEquiv] at happly
```

This is an example for the minimal connection interface. The reverse edge $1 \to 0$ is the identity, so it is not the inverse of doubling. The example must not be advertised as an instance of the stronger reversible interface.

4.3 Reversible connections use dependent fibers

The second interface assigns a module $F(a)$ to each vertex. Pair transport is a linear equivalence between the actual endpoint modules, and reverse transport is required to be its inverse.

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
structure ReversibleConnection
    (R : Type u) (X : Type v) (F : X → Type w)
    [CommRing R] [∀ a, AddCommGroup (F a)] [∀ a, Module R (F a)] where
  transport : ∀ a b, F a ≃ₗ[R] F b
  transport_self : ∀ a, transport a a = LinearEquiv.refl R (F a)
  transport_reverse : ∀ a b, transport b a = (transport a b).symm
```

The hypotheses include a commutative ring, additive commutative groups on all fibers, and module structures. These permit subtraction of parallel linear maps. The cocycle law is still not a structural field. Thus reversible connections may have nontrivial triangle holonomy.

The additive defect has a source and target visible in its type:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
def cocycleDefect (a b c : X) : F a →ₗ[R] F c :=
  (C.transport b c).toLinearMap.comp (C.transport a b).toLinearMap -
    (C.transport a c).toLinearMap
```

Write

$$\Delta_{a,b,c} = \Gamma_{b,c}\Gamma_{a,b} - \Gamma_{a,c} \quad : F(a) \longrightarrow F(c).$$

The triangle holonomy is

$$H_{a,b,c} = \Gamma_{c,a}\Gamma_{b,c}\Gamma_{a,b} \quad : F(a) \longrightarrow F(a),$$

and the triangle obstruction is $O_{a,b,c} = H_{a,b,c} - I$. Unlike the minimal multiplicative defect, this Δ is a linear map that need not be invertible. Both APIs use the short name `cocycleDefect`, but they live in different namespaces and have different types.

4.4 The identity between defect and holonomy

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem triangleHolonomy_sub_id (a b c : X) :
    C.triangleObstruction a b c =
      (C.transport c a).toLinearMap.comp (C.cocycleDefect a b c) := by
  ext x
  simp only [triangleObstruction, triangleHolonomy, cocycleDefect,
    LinearMap.sub_apply, LinearMap.id_apply, LinearMap.comp_apply,
    LinearEquiv.coe_toLinearMap, LinearEquiv.trans_apply, map_sub]
  rw [C.transport_reverse a c]
  simp
```

Pointwise, the identity reads

$$(H - I)x = \Gamma_{c,a}(\Gamma_{b,c}\Gamma_{a,b}x - \Gamma_{a,c}x).$$

Linearity moves the subtraction through $\Gamma_{c,a}$. The reverse law makes $\Gamma_{c,a}\Gamma_{a,c}$ the identity. The source proof uses exactly these two steps after unfolding the definitions.

Because $\Gamma_{c,a}$ is injective, the obstruction is zero exactly when the defect is zero. Vanishing of the additive defect is the cocycle equation. Vanishing of holonomy minus identity means the holonomy is the identity. The theorem chain establishes these equivalences at a fixed ordered triple before quantifying over every triple.

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem flat_iff_triangleHolonomy_eq_refl :
    C.Flat ↔ ∀ a b c,
      C.triangleHolonomy a b c = LinearEquiv.refl R (F a) := by
  constructor
  · intro hflat a b c
    rw [← C.cocycleDefect_eq_zero_iff_triangleHolonomy_eq_refl a b c,
      C.cocycleDefect_eq_zero_iff_cocycleAt]
    exact hflat a b c
  · intro hhol a b c
    rw [← C.cocycleDefect_eq_zero_iff_cocycleAt,
      C.cocycleDefect_eq_zero_iff_triangleHolonomy_eq_refl]
    exact hhol a b c
```

Here `Flat` means the pair-cocycle law at every triple of vertices. It is a global algebraic condition on all pair transports. It is not a differential curvature condition obtained from a limiting process or a local chart cover.

4.5 Finite paths and their endpoint types

The path function on lists returns `F start ≃[R] F (pathEndpoint start vertices)`. The return type depends on the endpoint computed from the list. Flatness lets induction replace every remaining segment by direct transport, then collapse the next triangle by the cocycle law.

The indexed alternative packages the endpoints in the path's type. `VertexPath X a b` is a finite path from a to b. An empty path has equal endpoints. A constructor prefixes a step to a path from the intermediate vertex. Its transport theorem is:

Source: `KomorebiPaperIII/ReversibleConnections.lean`.

```
theorem indexedPathTransport_eq_direct_of_flat (hflat : C.Flat) :
    ∀ {a b : X} (path : VertexPath X a b),
      indexedPathTransport C path = C.transport a b := by
  intro a b path
  induction path with
  | nil a => exact (C.transport_self a).symm
  | cons a b c rest ih =>
      simp only [indexedPathTransport]
      rw [ih]
      exact hflat a b c
```

The loop corollary takes `path : VertexPath X a a`. Its output is then already an automorphism of `F a`, and the self law identifies it with the identity. This avoids an extra coercion between a computed endpoint and the initial vertex.

The paths are finite syntax. Nothing here states that an infinite composition exists or that approximating finite paths converge. The finite endpoint theorem is exact on its stated domain.

5 Gauge reconstruction and packet obstruction blocks

5.1 The gauge action fixes its own convention

A gauge is a family of fiber automorphisms $G_a : F(a) \to F(a)$. The source convention is

$$\Gamma^G_{a,b} = G_b \Gamma_{a,b} G_a^{-1}.$$

The definition has to prove that the new pair maps satisfy the self and reverse laws, because its return value is again a reversible connection.

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
def gauge (G : Gauge (R := R) (F := F)) : ReversibleConnection R X F where
  transport a b := ((G a).symm.trans (C.transport a b)).trans (G b)
  transport_self a := by
    apply LinearEquiv.ext
    intro x
    rw [C.transport_self]
    simp
  transport_reverse a b := by
    rw [C.transport_reverse a b]
    apply LinearEquiv.ext
    intro x
    simp
```

The proof of each structural law is pointwise cancellation. Flatness is not needed to define the gauge action.

The defect has different source and target fibers, so it transforms with a gauge at each end:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem gauge_cocycleDefect
    (G : Gauge (R := R) (F := F)) (a b c : X) :
    (C.gauge G).cocycleDefect a b c =
      (G c).toLinearMap.comp
        ((C.cocycleDefect a b c).comp (G a).symm.toLinearMap) := by
  ext x
  simp [gauge, cocycleDefect]
```

Thus $\Delta^G_{a,b,c} = G_c \Delta_{a,b,c} G_a^{-1}$. Holonomy is based at a, so both endpoint gauges are at that vertex:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem gauge_triangleHolonomy
    (G : Gauge (R := R) (F := F)) (a b c : X) :
    (C.gauge G).triangleHolonomy a b c =
      ((G a).symm.trans (C.triangleHolonomy a b c)).trans (G a) := by
  apply LinearEquiv.ext
  intro x
  simp [gauge, triangleHolonomy]
```

The transformed matrix is conjugate to the original. Whether the holonomy is the identity is invariant, and the source also proves that the cocycle condition and flatness are invariant. A coordinate change may alter entries while preserving these exact statements.

5.2 Reconstructing a flat connection from canonical transport

Suppose a family of frames $\Psi_a : F(a) \simeq M$ and a base vertex o are supplied. The canonical connection is $K_{a,b} = \Psi_b^{-1}\Psi_a$. The source defines

$$G_a = \Gamma_{o,a} K_{a,o}.$$

Its type is a gauge on each fiber:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
def flatGauge (frame : ∀ a, F a ≃ₗ[R] M) (o : X) :
    Gauge (R := R) (F := F) :=
  fun a ↦ (canonicalTransport frame a o).trans (C.transport o a)
```

The reconstruction theorem proves $\Gamma_{a,b} = G_b K_{a,b} G_a^{-1}$ under flatness.

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem flat_isGaugeOfCanonical
    (hflat : C.Flat) (frame : ∀ a, F a ≃ₗ[R] M) (o a b : X) :
    C.transport a b =
      ((C.flatGauge frame o a).symm.trans
        ((canonicalConnection frame).transport a b)).trans
          (C.flatGauge frame o b) := by
  apply LinearEquiv.ext
  intro x
  have hg : (C.flatGauge frame o a).symm =
      (C.transport o a).symm.trans (canonicalTransport frame a o).symm := rfl
  rw [hg]
  have hcanonical_symm (p q : X) :
      (canonicalTransport frame p q).symm = canonicalTransport frame q p := rfl
  rw [hcanonical_symm]
  simp only [flatGauge, canonicalConnection, LinearEquiv.trans_apply]
  rw [show (C.transport o a).symm = C.transport a o by
    rw [C.transport_reverse o a]]
  simp only [canonicalTransport, LinearEquiv.trans_apply,
    LinearEquiv.symm_apply_apply, LinearEquiv.apply_symm_apply]
  have hao : (C.transport a o).trans (C.transport o b) = C.transport a b :=
    hflat a o b
  exact (congrArg (fun e : F a ≃ₗ[R] F b ↦ e x) hao).symm
```

The proof first expands the inverse of the defined gauge, then cancels the canonical frame maps. What remains is $\Gamma_{o,b}\Gamma_{a,o} = \Gamma_{a,b}$, the flatness equation at a, o, b.

The theorem is relative to supplied frames and a supplied vertex. It does not silently choose a vertex from an empty type. Nor does this declaration prove a classification theorem for topological bundles. Its result is an explicit algebraic formula for a flat normalized reversible pair connection on the supplied data.

5.3 A packet resolution need not be orthogonal

The formal interface for packet resolution asks for a finite complete family of idempotent endomorphisms.

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
structure PacketResolution (R : Type u) (M : Type v) (I : Type w)
    [CommRing R] [AddCommGroup M] [Module R M] [Fintype I] where
  projector : I → M →ₗ[R] M
  complete : ∑ i, projector i = LinearMap.id
  idempotent : ∀ i, (projector i).comp (projector i) = projector i
```

Completeness is $\sum_i P_i = I$. Idempotence is $P_i^2 = P_i$. The structure does not assert pairwise orthogonality. The block from input packet ν to output packet μ is

$$A_{\mu\nu} = P_\mu A P_\nu.$$

Reconstruction only needs completeness on both sides:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem sum_blocks (A : M →ₗ[R] M) :
    ∑ mu, ∑ nu, P.block A mu nu = A := by
  have hcomplete (x : M) : ∑ i, P.projector i x = x := by
    have hx := congrArg (fun L : M →ₗ[R] M ↦ L x) P.complete
    simpa using hx
  ext x
  simp only [block, LinearMap.sum_apply, LinearMap.comp_apply]
  calc
    (∑ mu, ∑ nu, P.projector mu (A (P.projector nu x))) =
        ∑ mu, P.projector mu (A (∑ nu, P.projector nu x)) := by
          apply Finset.sum_congr rfl
          intro mu _
          calc
            (∑ nu, P.projector mu (A (P.projector nu x))) =
                P.projector mu (∑ nu, A (P.projector nu x)) := by
                  exact (map_sum (P.projector mu)
                    (fun nu ↦ A (P.projector nu x)) Finset.univ).symm
            _ = P.projector mu (A (∑ nu, P.projector nu x)) := by
                  congr 1
                  exact (map_sum A
                    (fun nu ↦ P.projector nu x) Finset.univ).symm
    _ = ∑ mu, P.projector mu (A x) := by rw [hcomplete x]
    _ = A x := hcomplete (A x)
```

The proof applies the completeness equation to a vector. It combines the finite inner sum using linearity, applies completeness at the input, and then uses completeness at the output. No orthogonality cancellation appears.

All blocks vanish if and only if the map vanishes:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem blocks_zero_iff (A : M →ₗ[R] M) :
    (∀ mu nu, P.block A mu nu = 0) ↔ A = 0 := by
  constructor
  · intro h
    rw [← P.sum_blocks A]
    simp [h]
  · intro h mu nu
    simp [block, h]
```

The forward direction rewrites the map as its double block sum. The reverse direction substitutes the zero map into the definition of a block. This result is a detector for the whole map even when the projectors overlap.

5.4 Extracting a witness supported in a packet

Idempotence is needed for the stronger witness statement:

Source: KomorebiPaperIII/ReversibleConnections.lean.

```
theorem exists_packetWitness_of_block_ne_zero
    (A : M →ₗ[R] M) (mu nu : I) (hne : P.block A mu nu ≠ 0) :
    ∃ x, P.projector nu x = x ∧ P.projector mu (A x) ≠ 0 := by
  classical
  have hexists : ∃ y, P.block A mu nu y ≠ 0 := by
    by_contra h
    apply hne
    ext y
    by_contra hy
    exact h ⟨y, hy⟩
  rcases hexists with ⟨y, hy⟩
  refine ⟨P.projector nu y, ?_, ?_⟩
  · have hid := congrArg (fun L : M →ₗ[R] M ↦ L y) (P.idempotent nu)
    simpa using hid
  · simpa [block] using hy
```

A nonzero block supplies some vector y on which that block is nonzero. The proof then takes $x = P_\nu y$. Idempotence ensures $P_\nu x = x$, and the original nonzero block value becomes $P_\mu A x \neq 0$.

The extraction is classical. It proves existence and does not supply a numerical search routine for an arbitrary module. Its input is a proof that a block is nonzero. It does not decide that predicate for the user.

Apply these results to $O = H - I$ on the base fiber of a fixed triangle. The declarations in rows 33 and 34 show

$$\Delta = 0 \iff H = I \iff O = 0 \iff (\forall \mu, \nu)\ P_\mu O P_\nu = 0.$$

The theorem for the whole map, `obstructionDetection_equivalences`, records the three equivalences joining these four criteria. Nonzero obstruction blocks inherit the theorem giving a witness supported in a packet.

5.5 When a fixed packet pattern survives a gauge

Total obstruction always transforms by conjugation at the base vertex. For a *fixed* packet family to transform block by block, the gauge must commute with every projector.

Source: `KomorebiPaperIII/HolonomyBlocks.lean`.

```
def PacketPreserving (gauge : M ≃ₗ[R] M) : Prop :=
  ∀ packet, gauge.toLinearMap.comp (P.projector packet) =
    (P.projector packet).comp gauge.toLinearMap
```

The source first proves that the inverse gauge also commutes with the projectors. This lets the proof move both the gauge and its inverse past the output and input projectors:

Source: `KomorebiPaperIII/HolonomyBlocks.lean`.

```
theorem block_gauge_of_commutes
    (gauge : M ≃ₗ[R] M) (map : M →ₗ[R] M)
    (hpreserves : P.PacketPreserving gauge) (target source : I) :
    P.block (gaugeConjugate gauge map) target source =
      gaugeConjugate gauge (P.block map target source) := by
  ext value
  simp only [block, gaugeConjugate, LinearMap.comp_apply,
    LinearEquiv.coe_toLinearMap]
  have htarget (y : M) :
      P.projector target (gauge y) = gauge (P.projector target y) :=
    (LinearMap.congr_fun (hpreserves target) y).symm
  have hsource :
      gauge.symm (P.projector source value) =
        P.projector source (gauge.symm value) :=
    LinearMap.congr_fun
      (P.symm_commutes_of_packetPreserving gauge hpreserves source) value
  calc
    P.projector target
        (gauge (map (gauge.symm (P.projector source value)))) =
        gauge
          (P.projector target
            (map (gauge.symm (P.projector source value)))) :=
      htarget _
    _ = gauge
          (P.projector target
            (map (P.projector source (gauge.symm value)))) := by
      rw [hsource]
```

Consequently each block is conjugated separately, and each block's zero or nonzero status is invariant. Row 37 specializes this statement to a triangle obstruction and requires `P.PacketPreserving (G a)` at the base vertex.

An arbitrary gauge need not preserve the blocks of a fixed resolution. The unconditional fact is conjugation of the total obstruction. The conditional fact is conjugation of each fixed block. Keeping those hypotheses distinct is essential when interpreting a packet obstruction diagram.

6 Operation trees and certificates for individual candidates

6.1 The grammar does not assume algebraic laws

The nonlinear module starts with a finite syntax tree. Inputs, constants, unary labels, and binary labels can each have their own type.

Source: KomorebiPaperIII/NonlinearObstructions.lean.

```
inductive OperationTree
    (Input : Type u) (Constant : Type v) (Unary : Type w) (Binary : Type x)
    where
  | input : Input → OperationTree Input Constant Unary Binary
  | constant : Constant → OperationTree Input Constant Unary Binary
  | unary : Unary → OperationTree Input Constant Unary Binary →
      OperationTree Input Constant Unary Binary
  | binary : Binary → OperationTree Input Constant Unary Binary →
      OperationTree Input Constant Unary Binary →
      OperationTree Input Constant Unary Binary
```

To evaluate a tree, supply four functions: input values, constant values, unary semantics, and binary semantics. Unary and binary semantics are arbitrary functions on one common value type. They are not assumed additive, multilinear, differentiable, associative, or commutative. This generality is what makes the locality theorem applicable to any chosen semantics.

Every tree has finitely many nodes. The input type itself need not be finite. The function `inputs` computes a finite set of names appearing in the tree:

Source: KomorebiPaperIII/NonlinearObstructions.lean.

```
def inputs [DecidableEq Input] :
    OperationTree Input Constant Unary Binary → Finset Input
  | .input i => {i}
  | .constant _ => ∅
  | .unary _ child => inputs child
  | .binary _ left right => inputs left ∪ inputs right
```

A constant reads no input. A unary node reads exactly the names read by its child. A binary node reads the union of its two children's names. Because this is a set, repeated uses of one input are recorded once. The semantics may still depend nonlinearly on repeated occurrences.

6.2 The locality proof follows the syntax

Source: KomorebiPaperIII/NonlinearObstructions.lean.

```
theorem eval_congr_on_inputs [DecidableEq Input]
    (tree : OperationTree Input Constant Unary Binary)
    (first second : Input → Value)
    (constantValue : Constant → Value)
    (unaryValue : Unary → Value → Value)
    (binaryValue : Binary → Value → Value → Value)
```

```
  (hagree : ∀ i ∈ tree.inputs, first i = second i) :
  tree.eval first constantValue unaryValue binaryValue =
    tree.eval second constantValue unaryValue binaryValue := by
induction tree with
| input i =>
    exact hagree i (by simp [inputs])
| constant c =>
    rfl
| unary op child ih =>
    simp only [eval]
    congr 1
    exact ih fun i hi ↦ hagree i (by simpa [inputs] using hi)
| binary op left right ihLeft ihRight =>
    simp only [eval]
    congr 1
    · exact ihLeft fun i hi ↦ hagree i (by simp [inputs, hi])
    · exact ihRight fun i hi ↦ hagree i (by simp [inputs, hi])
```

In an input case, the agreement hypothesis applies to that input's singleton support. In a constant case, both evaluations are definitionally equal. At a unary node, the induction hypothesis identifies the child values, and applying the same unary function preserves equality. At a binary node, the two child induction hypotheses apply because each child's support lies in the union.

The result states one implication: agreement on the syntactically named inputs gives equal evaluation. It does not claim every named input is semantically influential. For example, an operation may ignore one argument. The syntactic support remains a safe dependency set.

If `tree.inputs = ∅`, agreement on the empty set is vacuous, so the value is independent of the input environment. The source proves this as `eval_eq_of_inputs_eq_empty`. Constants and semantics are held fixed between the two evaluations. Changing the semantics would require a different comparison theorem.

6.3 What the nonlinear extension adds outside this contract

The mathematical text develops stronger ordinary statements about typed or colored trees, multilinear packet expansions, composition defects, and higher-order structures. The finite Lean contract contains the common-value unary/binary grammar and its input-locality theorem. It does not contain the dependent colored-tree development or a tree-level Faà di Bruno theorem.

This boundary is mathematical, not stylistic. A colored tree needs compatibility between node input and output types. A multilinear packet expansion needs algebraic hypotheses absent from arbitrary unary and binary semantics. The existing locality proof cannot certify these statements by a change of terminology.

6.4 Residuals and a fixed proposed solution

The obstruction API asks for a function `residual : Candidate → Packet → Residual`, with only `Zero Residual` assumed. A candidate is a solution when every packet residual is zero. A certificate stores one packet at which its residual is nonzero:

Source: KomorebiPaperIII/NonlinearObstructions.lean.

```
structure ObstructionPacket
    (residual : Candidate → Packet → Residual) (candidate : Candidate) where
  packet : Packet
  nonzero : residual candidate packet ≠ 0
```

The candidate appears in the certificate's type. A certificate for one candidate cannot directly be passed as a certificate for a different one. The theorem `ObstructionPacket.not_isSolution` uses the stored packet to contradict the condition that every packet residual is zero.

Classically, failure of the condition that every packet residual is zero supplies a nonempty certificate type. The combined equivalence is:

Source: KomorebiPaperIII/NonlinearObstructions.lean.

```
theorem not_isSolution_iff_nonempty_obstructionPacket
    {residual : Candidate → Packet → Residual} {candidate : Candidate} :
    ¬ IsSolution residual candidate ↔
      Nonempty (ObstructionPacket residual candidate) := by
  constructor
  · exact obstructionPacket_of_not_isSolution
  · rintro (certificate)
    exact certificate.not_isSolution
```

The output is `Nonempty (ObstructionPacket residual candidate)`. This is a proposition asserting that the certificate type has an inhabitant. It is not a computable function that enumerates packets or finds one within a time bound.

Neither `Packet` nor `Candidate` needs a finite enumeration for this logical equivalence. The adjective "finite" in the overall contract describes the walk sums, paths, packet resolutions, and operation-tree syntax where finiteness is imposed. It should not be misread as an extra `Fintype` assumption on this candidate theorem.

6.5 The quantifiers determine the conclusion

For one proposed candidate c, the theorem says

$$\neg(\forall p,\ r(c,p) = 0) \quad \Longleftrightarrow \quad \text{a certificate of } r(c,p) \neq 0 \text{ exists.}$$

To show that a global solution does not exist, one needs the additional assertion that *every* candidate fails, or an argument producing an obstruction for each candidate. To prove existence of a global solution, one must produce a candidate with every residual zero. Neither conclusion follows merely from the certificate interface.

Once a witness is supplied, the interface makes local failure portable and checkable. The formal type records the residual, the candidate, the packet, and the exact nonzero assertion. A later theorem can build on these data without confusing rejection of one candidate with a universal impossibility result.

7 The finite baseline: evidence, trust, and replay

This chapter explains the finite algebraic core and its verification gates. For the combined core and analytic continuation, use the opening chapter and `formal/README.md`. The finite namespace counts in the excerpts apply only to that finite baseline.

7.1 The pinned environment is part of the claim

The source records specify the compiler and dependency environment.

The recorded compiler controls identify the warning policy actually checked.

The source distribution omits compiled outputs and dependency caches. A fresh environment therefore needs the pinned toolchain and the dependencies named by the manifest. Rebuilding the PDF guide does not need Lean. Replaying the mathematical formalization does.

Descriptive dependency branches such as `master` or `main` are not substitutes for exact manifest revisions. Updating a dependency and obtaining a successful build may be useful, but it is a new verification event. The original event remains tied to the recorded environment.

7.2 A map gate and a namespace gate answer different questions

The authoritative `scripts/verify_theorem_map.py` checks the map row sequence, uniqueness, compiled declaration kinds, manuscript status annotations, inherited references, and explicit audit coverage. In the pinned compiled environment, it distinguishes the 88 theorem targets from the 55 construction targets. A lexical scan can locate these declarations in source, but does not independently elaborate their types.

The kernel audit file asks Lean to print transitive axioms for all mapped targets. It then inspects the compiled `Komorebi.PaperIII` namespace, including generated declarations and implementation auxiliaries. This second step protects the boundary beyond just the manually listed names.

Source: `KomorebiPaperIII/KernelAudit.lean`.

```
run_cmd do
  let env ← getEnv
  let mut audited : Nat := 0
  let mut theorems : Nat := 0
  for (name, info) in env.checked.get.constants.toList do
    if Name.isPrefixOf `Komorebi.PaperIII name then
      audited := audited + 1
      match info with
      | .thmInfo _ => theorems := theorems + 1
      | _ => pure ()
      if info.isUnsafe then
        throwError "unsafe Paper III declaration: {name}"
      match info with
      | .axiomInfo _ =>
          throwError "Paper III axiom declaration: {name}"
      | .opaqueInfo _ =>
```

```
        throwError "opaque Paper III declaration: {name}"
    | _ => pure ()
    let axioms ← Lean.collectAxioms name
    for axiomName in axioms do
      unless axiomName == ``propext || axiomName == ``Classical.choice ||
          axiomName == ``Quot.sound do
        throwError "unexpected axiom {axiomName} in Paper III declaration {name}"
if audited != 468 then
  throwError "Paper III namespace declaration count changed: expected 468, found {audited}"
if theorems != 229 then
  throwError "Paper III namespace theorem count changed: expected 229, found {theorems}"
```

The gate rejects unsafe declarations, project axioms, opaque declarations, and any collected axiom outside propext, Classical.choice, and Quot.sound. It also checks the exact namespace and theorem-kind counts. These expected counts identify the audited declaration set. A deliberate addition requires reviewing the counts while retaining the checks.

The source policy check additionally rejects proof placeholders and specified shortcuts. Text scanning alone cannot establish a theorem's correctness. It is one gate among the build, compiled map checks, axiom traversal, and checks of source integrity.

7.3 Reading the inherited transcript

The saved transcript has 143 named records. Some say that the target depends on no axioms. Others list a subset of the allowed standard basis. The final line records 468 audited namespace declarations, including 229 `theoremInfo` entries, and 143 explicit mapped targets.

The difference between 88 and 229 is not a discrepancy. Eighty-eight counts the theorem declarations deliberately mapped to the chapter. The compiled namespace count includes generated theorems and other supporting theorem entries. Similarly, 143 and 468 refer to different denominators: the explicitly mapped targets versus every compiled entry under the audited namespace.

This guide checked that the transcript's named records agree with the map and the audit directives. It also checked the transcript hash against the inherited report. That establishes consistency and byte identity of the preserved materials. It does not authenticate an external compiler session by itself. A new execution is the direct way to regenerate compiler evidence from those sources.

7.4 The inherited Chapters I–II references

Chapter III records twelve unique declarations from the separately verified foundational Chapters I–II bundle. They appear in fourteen manuscript references because two refinement declarations are used twice. The bundle is pinned by SHA-256:

`a20cf9353f0e882834d31e665dfd819a04e43fa606d402309be63e3b94447052`

The standalone Chapter III project does not import or recompile those declarations. Its gate checks inherited annotations against the dependency manifest. These are inherited references, not additional entries in the 143-target Chapter III namespace audit. Appendix B gives the exact dependency names and roles.

7.5 Replay from a preserved source copy

The source distribution offers an offline integrity mode. From its root, run:

```
bash VERIFY.sh
```

This mode checks source membership, checksums, map and audit consistency, manuscript status references, and the saved transcript. It does not run Lean. It is suitable for checking that a copied source tree still matches its manifest.

For a formal replay, work in a separate copy of the source distribution. From formalization/lean, run the documented commands in order:

```
lake --wfail build
python3 scripts/verify_theorem_map.py
lake env lean -E warning KomorebiPaperIII/KernelAudit.lean
```

Retain the exit code and the complete standard output and error stream for every command. The first command builds the project. The second checks the human map against the compiled environment and the original manuscript. The third regenerates the axiom reports and namespace result. A missing dependency, toolchain failure, timeout, or nonzero exit is not a passing proof run.

The source distribution also supplies bash VERIFY.sh --full. That path creates a temporary copy, regenerates figures, runs Lean and its gates, rebuilds the original finite manuscript, and verifies the original artifact hashes. It requires its documented TeX, graphics, Python, and Lean dependencies. It verifies the finite source distribution. The expanded companion PDF has a separate scope.

7.6 The formal verification boundary

The finite portion of this guide supplies exact source excerpts, the complete 41-row map, the preserved transcript, and static reconciliation. The opening chapter separately reports compiler and kernel verification. The package records separate computational checks and the chapter's earlier bounded mathematical review. These records concern different objects. They do not establish fresh Lean proofs of every ordinary theorem.

The finite algebraic contract has explicit types and evidence. To expand it, add the required types and proofs, update the map, run the pinned or newly documented environment, and retain the new transcript.

8 The complete 41-row formal contract

This index preserves every row, target, status, and boundary of the authoritative finite algebraic map. Fully qualified declaration names are printed in full. File and line locations are a current lexical navigation aid, not a second compiled type audit. Each target appears in exactly one row. The 143 names comprise 88 theorem declarations and 55 constructions.

The finite limits are part of the statements: edge sums use finite packet and edge types, arbitrary-length heterogeneous products keep one common finite packet type, and packet reconstruction uses a finite complete idempotent family. Rows 38–41 have the separate grammar and candidate-specific scopes explained in Section 6.

8.1 Row 1: Packet graph and scalar packet matrix

Status and boundary. Defined; row is target and column is source

`Komorebi.PaperIII.EdgeOperators.PacketGraph`
structure, `KomorebiPaperIII/EdgeOperators.lean`, line 27.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.packetMatrix`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 36.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.packetMatrix_apply`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 40.

8.2 Row 2: Typed edge system, elementary edge operator, and finite transfer operator

Status and boundary. Defined for the finite scalar matrix model

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem`
structure, `KomorebiPaperIII/EdgeOperators.lean`, line 47.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 55.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_apply`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 58.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 162.

8.3 Row 3: Edge support outside the target-source block vanishes

Status and boundary. Proved

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_eq_zero_off_support`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 65.

8.4 Row 4: Scalar packet multiplication and the two-edge composition dichotomy

Status and boundary. Proved; right factor acts first

`Komorebi.PaperIII.EdgeOperators.PacketGraph.packetMatrix_mul_packetMatrix`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 31.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_mul_edgeOp_eq_zero`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 75.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.edgeOp_mul_edgeOp_of_composable`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 92.

8.5 Row 5: Composable edge words and their operators

Status and boundary. Defined and proved

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Composable`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 106.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 112.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_nil`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 117.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_singleton`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 121.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_append`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 126.

8.6 Row 6: Broken and noncomposable words vanish

Status and boundary. Proved

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_zero_of_first_break`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 135.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_zero_of_not_composable`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 144.

8.7 Row 7: Recursive, tuple, explicit-word, and walk expansions

Status and boundary. Defined; all sums are finite

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordExpansion`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 170.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.tupleWordOp`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 177.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.explicitWordExpansion`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 182.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.walkExpansion`
def, `KomorebiPaperIII/EdgeOperators.lean`, line 190.

8.8 Row 8: Successor laws for explicit words

Status and boundary. Proved

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.tupleWordOp_snoc`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 198.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.explicitWordExpansion_succ`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 210.

8.9 Row 9: Transfer powers equal recursive, explicit-word, and composable-walk expansions

Status and boundary. Proved

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordExpansion_eq_explicitWordExpansion`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 226.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.explicitWordExpansion_eq_walkExpansion`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 237.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_wordExpansion`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 252.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_explicitWordExpansion`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 263.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_eq_walkExpansion`
theorem, `KomorebiPaperIII/EdgeOperators.lean`, line 274.

8.10 Row 10: Endpoint block and coefficient of a composable word

Status and boundary. Defined and proved; coefficient order follows operator composition

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.reverseCoefficient`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 49.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.terminalTarget`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 54.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Connects`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 59.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_eq_packetMatrix_of_composable`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 75.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.wordOp_apply_eq_if_connects`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 98.

8.11 Row 11: Fixed-endpoint coefficient sum

Status and boundary. Defined and proved for fixed length

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.endpointWalkCoefficientSum`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 118.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_apply_eq_sum_walks`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 128.

8.12 Row 12: Fixed-length and global reachability obstructions

Status and boundary. Proved; reachability includes the empty path

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_apply_eq_zero_of_no_walk`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 142.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.Reachable`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 153.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_apply_eq_zero_of_not_reachable`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 157.

`Komorebi.PaperIII.EdgeOperators.PacketGraph.EdgeSystem.transferOp_pow_mulVec_apply_eq_zero_of_unreachable_support`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 169.

8.13 Row 13: Fully typed two- and three-step packet expansions

Status and boundary. Proved; intermediate packet types may differ

`Komorebi.PaperIII.EdgeOperators.heterogeneousPacketPathExpansion_two`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 195.

`Komorebi.PaperIII.EdgeOperators.heterogeneousPacketPathExpansion_three`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 205.

8.14 Row 14: Arbitrary-length heterogeneous factor expansion

Status and boundary. Defined and proved for varying factors on one common finite packet type

`Komorebi.PaperIII.EdgeOperators.heterogeneousProduct`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 221.

`Komorebi.PaperIII.EdgeOperators.heterogeneousPathSum`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 231.

`Komorebi.PaperIII.EdgeOperators.heterogeneousPacketPathExpansion`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 242.

8.15 Row 15: Supported heterogeneous paths and vanishing

Status and boundary. Defined and proved; support relations are explicit data

`Komorebi.PaperIII.EdgeOperators.SupportedStep`
structure, `KomorebiPaperIII/EdgeExtensions.lean`, line 258.

`Komorebi.PaperIII.EdgeOperators.SupportedPath`
def, `KomorebiPaperIII/EdgeExtensions.lean`, line 265.

`Komorebi.PaperIII.EdgeOperators.heterogeneousPathSum_eq_zero_of_no_supportedPath`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 273.

`Komorebi.PaperIII.EdgeOperators.supportedHeterogeneousPacketPathExpansion`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 296.

`Komorebi.PaperIII.EdgeOperators.heterogeneousProduct_apply_eq_zero_of_no_supportedPath`
theorem, `KomorebiPaperIII/EdgeExtensions.lean`, line 304.

8.16 Row 16: Canonical homogeneous linear coordinate transport

Status and boundary. Defined from one frame family

`Komorebi.PaperIII.Transport.exactTransport`
def, `KomorebiPaperIII/Transport.lean`, line 35.

`Komorebi.PaperIII.Transport.exactTransport_apply`
theorem, `KomorebiPaperIII/Transport.lean`, line 38.

8.17 Row 17: Exact self, cocycle, round-trip, and triangle laws

Status and boundary. Proved

`Komorebi.PaperIII.Transport.exactTransport_self`
theorem, `KomorebiPaperIII/Transport.lean`, line 44.

`Komorebi.PaperIII.Transport.exactTransport_cocycle`
theorem, `KomorebiPaperIII/Transport.lean`, line 51.

`Komorebi.PaperIII.Transport.exactTransport_roundTrip`
theorem, `KomorebiPaperIII/Transport.lean`, line 59.

`Komorebi.PaperIII.Transport.exactTransport_triangleLoop`
theorem, `KomorebiPaperIII/Transport.lean`, line 65.

8.18 Row 18: Exact finite path transport telescopes

Status and boundary. Defined and proved for arbitrary finite vertex lists

`Komorebi.PaperIII.Transport.pathEndpoint`
def, `KomorebiPaperIII/Transport.lean`, line 71.

`Komorebi.PaperIII.Transport.exactPathTransport`
def, `KomorebiPaperIII/Transport.lean`, line 76.

`Komorebi.PaperIII.Transport.exactPathTransport_eq_endpoint`
theorem, `KomorebiPaperIII/Transport.lean`, line 84.

`Komorebi.PaperIII.Transport.exactPathTransport_loop`
theorem, `KomorebiPaperIII/Transport.lean`, line 96.

8.19 Row 19: Set-level exact coordinate transport

Status and boundary. Defined and proved for arbitrary equivalences, without linearity assumptions

`Komorebi.PaperIII.Transport.exactCoordinateTransport`
def, `KomorebiPaperIII/CoordinateNaturality.lean`, line 24.

`Komorebi.PaperIII.Transport.exactCoordinateTransport_apply`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 29.

`Komorebi.PaperIII.Transport.exactCoordinateTransport_self`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 36.

`Komorebi.PaperIII.Transport.exactCoordinateTransport_symm`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 43.

`Komorebi.PaperIII.Transport.exactCoordinateTransport_cocycle`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 52.

`Komorebi.PaperIII.Transport.exactCoordinateTransport_triangle`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 61.

8.20 Row 20: Representation, composition, and naturality of arbitrary functions

Status and boundary. Defined and proved at set level

`Komorebi.PaperIII.Transport.exactCoordinateMap`
def, `KomorebiPaperIII/CoordinateNaturality.lean`, line 71.

`Komorebi.PaperIII.Transport.exactCoordinateMap_apply`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 77.

`Komorebi.PaperIII.Transport.exactCoordinateMap_id`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 86.

`Komorebi.PaperIII.Transport.exactCoordinateMap_comp`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 93.

`Komorebi.PaperIII.Transport.exactCoordinateMap_naturality`
theorem, `KomorebiPaperIII/CoordinateNaturality.lean`, line 109.

8.21 Row 21: Minimal homogeneous noncanonical connection and its defects

Status and boundary. Defined; only self-transport is structural

`Komorebi.PaperIII.Transport.Connection`
structure, `KomorebiPaperIII/Transport.lean`, line 113.

`Komorebi.PaperIII.Transport.Connection.CocycleAt`
def, `KomorebiPaperIII/Transport.lean`, line 123.

`Komorebi.PaperIII.Transport.Connection.Flat`
def, `KomorebiPaperIII/Transport.lean`, line 127.

`Komorebi.PaperIII.Transport.Connection.cocycleDefect`
def, `KomorebiPaperIII/Transport.lean`, line 133.

`Komorebi.PaperIII.Transport.Connection.triangleHolonomy`
def, `KomorebiPaperIII/Transport.lean`, line 138.

8.22 Row 22: Flat minimal connections have trivial defect and triangle holonomy

Status and boundary. Proved

`Komorebi.PaperIII.Transport.Connection.triangleHolonomy_eq_refl_of_flat`
theorem, `KomorebiPaperIII/Transport.lean`, line 142.

`Komorebi.PaperIII.Transport.Connection.cocycleDefect_eq_refl_of_flat`
theorem, `KomorebiPaperIII/Transport.lean`, line 147.

8.23 Row 23: Explicit three-point connection with nontrivial loop

Status and boundary. Constructed and proved over the rationals

`Komorebi.PaperIII.Transport.doubleEquiv`
def, `KomorebiPaperIII/Transport.lean`, line 160.

`Komorebi.PaperIII.Transport.doubleEquiv_apply`
theorem, `KomorebiPaperIII/Transport.lean`, line 170.

`Komorebi.PaperIII.Transport.twistedThreePointConnection`
def, `KomorebiPaperIII/Transport.lean`, line 177.

`Komorebi.PaperIII.Transport.twisted_triangleHolonomy_apply_one`
theorem, `KomorebiPaperIII/Transport.lean`, line 185.

`Komorebi.PaperIII.Transport.twisted_triangleHolonomy_ne_refl`
theorem, `KomorebiPaperIII/Transport.lean`, line 192.

8.24 Row 24: Heterogeneous reversible connection and gauges

Status and boundary. Defined over a commutative ring and a family of modules

`Komorebi.PaperIII.Transport.ReversibleConnection`
structure, `KomorebiPaperIII/ReversibleConnections.lean`, line 35.

`Komorebi.PaperIII.Transport.ReversibleConnection.Gauge`
abbrev, `KomorebiPaperIII/ReversibleConnections.lean`, line 49.

8.25 Row 25: Reversible cocycle, flatness, additive defect, holonomy, and obstruction

Status and boundary. Defined; theorem arguments are in traversal order

`Komorebi.PaperIII.Transport.ReversibleConnection.CocycleAt`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 52.

`Komorebi.PaperIII.Transport.ReversibleConnection.Flat`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 56.

`Komorebi.PaperIII.Transport.ReversibleConnection.cocycleDefect`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 62.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleHolonomy`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 67.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleObstruction`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 71.

8.26 Row 26: Defect-holonomy identity and equivalent flatness criteria

Status and boundary. Proved exactly

`Komorebi.PaperIII.Transport.ReversibleConnection.cocycleDefect_eq_zero_iff_cocycleAt`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 75.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleHolonomy_sub_id`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 96.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleObstruction_eq_zero_iff_cocycleDefect_eq_zero`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 107.

`Komorebi.PaperIII.Transport.ReversibleConnection.triangleObstruction_eq_zero_iff_triangleHolonomy_eq_refl`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 120.

`Komorebi.PaperIII.Transport.ReversibleConnection.cocycleDefect_eq_zero_iff_triangleHolonomy_eq_refl`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 139.

`Komorebi.PaperIII.Transport.ReversibleConnection.flat_iff_triangleHolonomy_eq_refl`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 146.

8.27 Row 27: Canonical reversible frame connection is flat

Status and boundary. Defined and proved

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalTransport`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 164.

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalConnection`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 171.

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalConnection_flat`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 181.

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalTransport_zero_connectionDefect`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 189.

`Komorebi.PaperIII.Transport.ReversibleConnection.canonicalTransport_triangleHolonomy_eq_refl`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 196.

8.28 Row 28: Every flat reversible connection is gauge-equivalent to canonical transport

Status and boundary. Gauge constructed and identity proved relative to a base vertex and frames

`Komorebi.PaperIII.Transport.ReversibleConnection.flatGauge`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 207.

`Komorebi.PaperIII.Transport.ReversibleConnection.flat_isGaugeOfCanonical`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 215.

8.29 Row 29: List-based reversible path transport is endpoint-only when flat

Status and boundary. Defined and proved for finite lists

`Komorebi.PaperIII.Transport.ReversibleConnection.pathTransport`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 241.

`Komorebi.PaperIII.Transport.ReversibleConnection.pathTransport_eq_direct_of_flat`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 248.

8.30 Row 30: Indexed finite path transport and flat loops

Status and boundary. Defined and proved for dependent fibers

`Komorebi.PaperIII.Transport.ReversibleConnection.VertexPath`
inductive, `KomorebiPaperIII/ReversibleConnections.lean`, line 262.

`Komorebi.PaperIII.Transport.ReversibleConnection.indexedPathTransport`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 267.

`Komorebi.PaperIII.Transport.ReversibleConnection.indexedPathTransport_eq_direct_of_flat`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 274.

`Komorebi.PaperIII.Transport.ReversibleConnection.indexedLoopTransport_eq_refl_of_flat`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 286.

8.31 Row 31: Gauge action, covariance, and invariance

Status and boundary. Defined and proved algebraically

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 294.

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_cocycleDefect`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 308.

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_triangleHolonomy`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 317.

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_cocycleAt_iff`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 326.

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_flat_iff`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 344.

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_triangleHolonomy_eq_refl_iff`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 353.

8.32 Row 32: Finite complete idempotent packet resolution and blocks

Status and boundary. Defined; orthogonality is not required

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution`
structure, `KomorebiPaperIII/ReversibleConnections.lean`, line 379.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.block`
def, `KomorebiPaperIII/ReversibleConnections.lean`, line 391.

8.33 Row 33: Exact block reconstruction, zero detection, and packet witnesses

Status and boundary. Proved; witness extraction is classical

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.sum_blocks`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 395.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.blocks_zero_iff`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 420.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.exists_packetWitness_of_block_ne_zero`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 430.

8.34 Row 34: Triangle-obstruction block reconstruction and four-way detection

Status and boundary. Proved for each fixed triangle and finite packet resolution

`Komorebi.PaperIII.HolonomyBlocks.obstruction_sum_blocks`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 452.

`Komorebi.PaperIII.HolonomyBlocks.obstructionBlocks_zero_iff`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 464.

`Komorebi.PaperIII.HolonomyBlocks.obstructionDetection_equivalences`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 479.

`Komorebi.PaperIII.HolonomyBlocks.exists_packetWitness_of_obstructionBlock_ne_zero`
theorem, `KomorebiPaperIII/ReversibleConnections.lean`, line 495.

8.35 Row 35: Gauge conjugation and packet preservation

Status and boundary. Defined

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.gaugeConjugate`
def, `KomorebiPaperIII/HolonomyBlocks.lean`, line 51.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.PacketPreserving`
def, `KomorebiPaperIII/HolonomyBlocks.lean`, line 55.

8.36 Row 36: Packet-preserving conjugation acts blockwise and preserves vanishing

Status and boundary. Proved

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.symm_commutes_of_packetPreserving`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 60.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.block_gauge_of_commutes`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 70.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.gaugeConjugate_eq_zero_iff`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 99.

`Komorebi.PaperIII.HolonomyBlocks.PacketResolution.block_gauge_eq_zero_iff`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 111.

8.37 Row 37: Gauge covariance of total and packetwise triangle obstructions

Status and boundary. Proved; packetwise result assumes the base gauge commutes with every projector

`Komorebi.PaperIII.Transport.ReversibleConnection.gauge_triangleObstruction`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 27.

`Komorebi.PaperIII.HolonomyBlocks.obstructionBlock_gauge_of_commutes`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 131.

`Komorebi.PaperIII.HolonomyBlocks.obstructionBlock_gauge_eq_zero_iff`
theorem, `KomorebiPaperIII/HolonomyBlocks.lean`, line 145.

8.38 Row 38: Nonlinear operation-tree grammar, evaluation, and input support

Status and boundary. Defined for arbitrary unary and binary semantics

`Komorebi.PaperIII.NonlinearObstructions.OperationTree`
inductive, `KomorebiPaperIII/NonlinearObstructions.lean`, line 34.

`Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval`
def, `KomorebiPaperIII/NonlinearObstructions.lean`, line 51.

`Komorebi.PaperIII.NonlinearObstructions.OperationTree.inputs`
def, `KomorebiPaperIII/NonlinearObstructions.lean`, line 66.

8.39 Row 39: Operation-tree packet locality

Status and boundary. Proved without algebraic assumptions on operations

`Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval_congr_on_inputs`
theorem, `KomorebiPaperIII/NonlinearObstructions.lean`, line 77.

`Komorebi.PaperIII.NonlinearObstructions.OperationTree.eval_eq_of_inputs_eq_empty`
theorem, `KomorebiPaperIII/NonlinearObstructions.lean`, line 102.

8.40 Row 40: Candidate-specific solution and obstruction packet

Status and boundary. Defined

`Komorebi.PaperIII.NonlinearObstructions.IsSolution`
def, `KomorebiPaperIII/NonlinearObstructions.lean`, line 126.

`Komorebi.PaperIII.NonlinearObstructions.ObstructionPacket`
structure, `KomorebiPaperIII/NonlinearObstructions.lean`, line 130.

8.41 Row 41: Candidate failure is equivalent to a local obstruction

Status and boundary. Proved; extraction from nonsolution is classical and candidate-specific

`Komorebi.PaperIII.NonlinearObstructions.ObstructionPacket.not_isSolution`
theorem, `KomorebiPaperIII/NonlinearObstructions.lean`, line 136.

`Komorebi.PaperIII.NonlinearObstructions.obstructionPacket_of_not_isSolution`
theorem, `KomorebiPaperIII/NonlinearObstructions.lean`, line 144.

`Komorebi.PaperIII.NonlinearObstructions.not_isSolution_iff_nonempty_obstructionPacket`
theorem, `KomorebiPaperIII/NonlinearObstructions.lean`, line 156.

A Monograph navigation

The following locators use statement and section numbers in the Chapter III mathematical text. A locator connects related presentations. It does not certify every neighboring ordinary statement in Lean.

Canonical label	Chapter III locator
`sec:purpose`	Section 1
`sec:edge-operators`	Section 2
`def:edge-system`	Definition 2.1
`lem:edge-support`	Lemma 2.2
`thm:edge-annihilation`	Theorem 2.3
`thm:edge-composable`	Theorem 2.4
`sec:words-walks`	Section 3
`def:word-op`	Definition 3.1
`thm:first-break`	Theorem 3.2
`thm:nonwalk-vanish`	Theorem 3.3
`def:recursive-word-expansion`	Definition 3.4
`thm:power-recursive-expansion`	Theorem 3.5
`lem:word-endpoint`	Lemma 3.7
`thm:word-to-walk`	Theorem 3.8
`cor:word-to-walk-entry`	Corollary 3.9
`cor:fixed-reachability`	Corollary 3.10
`cor:orbit-reachability`	Corollary 3.11
`sec:canonical-transport`	Section 4
`thm:exact-frame-transport`	Theorem 4.1
`thm:exact-path-transport`	Theorem 4.2
`def:canonical-J`	Definition 4.3
`thm:J-cocycle`	Theorem 4.4
`thm:canonical-naturality`	Theorem 4.5
`cor:J-flat`	Corollary 4.6
`sec:connection`	Section 5
`def:minimal-connection`	Definition 5.1
`thm:minimal-flatness`	Theorem 5.2
`prop:twisted-loop`	Proposition 5.3
`def:connection`	Definition 5.5
`def:calibration-flatness`	Definition 5.6
`def:cocycle-defect`	Definition 5.8
`def:triangle-holonomy`	Definition 5.9

Canonical label	Chapter III locator
thm:defect-holonomy	Theorem 5.10
cor:canonical-zero-defect	Corollary 5.11
def:path-transport	Definition 5.12
prop:flat-endpoints	Proposition 5.13
sec:gauge	Section 6
def:gauge-change	Definition 6.1
thm:gauge-covariance	Theorem 6.2
thm:flat-gauge-canonical	Theorem 6.3
sec:obstruction-packets	Section 7
def:generic-obstruction	Definition 7.1
thm:generic-obstruction	Theorem 7.2
def:obstruction-packet	Definition 7.4
thm:obstruction-detection	Theorem 7.5
cor:packet-witness	Corollary 7.6
prop:packet-gauge	Proposition 7.7
sec:holonomy-walks	Section 8
prop:heterogeneous-path	Proposition 8.1
sec:example	Section 9
sec:arbitrary-maps	Section 10
prop:set-naturality	Proposition 10.1
sec:operation-trees	Section 11
def:compiled-operation-tree	Definition 11.1
thm:tree-locality	Theorem 11.2
def:signature	Definition 11.4
def:operation-tree	Definition 11.5
sec:ledger	Section 12

B Inherited declaration names

The following twelve names are inherited citations to the separately verified foundational formalizations of Chapters I–II. The Chapter III formal build does not recompile them. Their fourteen manuscript uses are checked against the archived dependency manifest.

1. Reachability support hull

`Komorebi.PacketNaturalityCompletion.Reachability.algebraicOrbitHull_le_reachable_support`

2. Unreachable-packet obstruction

`Komorebi.PacketNaturalityCompletion.Reachability.unreachable_nonzero_packet_obstruction`

3. Nested coordinate refinement

`Komorebi.PaperII.NestedRefinement.refineCoordinates`

4. Refinement equivalence

`Komorebi.PaperII.NestedRefinement.coordinateRefinementEquiv`

5. Identity refinement

`Komorebi.PaperII.NestedRefinement.refineCoordinates_self`

6. Coarsen-after-refine identity

`Komorebi.PaperII.NestedRefinement.coarsenCoordinates_refineCoordinates`

7. Refine-after-coarsen identity

`Komorebi.PaperII.NestedRefinement.refineCoordinates_coarsenCoordinates`

8. Refinement cocycle and functoriality

`Komorebi.PaperII.NestedRefinement.refineCoordinates_comp`

9. Transitivity of refinement equivalences

`Komorebi.PaperII.NestedRefinement.coordinateRefinementEquiv_trans`

10. Coordinate-operator composition

`Komorebi.PaperII.NestedRefinement.coordinateOperator_comp`

11. Coordinate-operator naturality

`Komorebi.PaperII.NestedRefinement.coordinateOperator_naturality`

12. Prior packet-path power identity

`Komorebi.PacketNaturalityCompletion.Packets.transferBlock_pow_eq_packetPathSum`

Projector Resolved Seed Calculus IV

The formal companion to Chapter IV explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 1816

Return to Chapter IV

How to use this volume

A guide to the actual formal development. 119 Lean source files. 151 map rows. 1,164 mapped declarations. All 68 theorem-like manuscript results have full scoped coverage in the recorded formal contract. Read the eight tutorial chapters for the mathematical route, the coverage ledger for complete manuscript results, and the declaration catalog for exact source targets.

This guide relates the source, contract, theorem map, and saved kernel transcript. The verification chapter records the replay using pinned sources and its precise scope. Static reconciliation remains a distinct form of evidence.

1 How to read this formalization

1.1 What has been completed

Chapter IV studies one scalar cyclic family through several realizations. A residue-class projector selects coefficients. Its exponential sections become entire functions. Signed sections produce a distinguished denominator, whose reciprocal coefficients form parameterized polynomial families. The denominator's zeros then control canonical products, a diagonal determinant, coefficient asymptotics, and zeta functions.

The formal development follows these transitions explicitly. It does not identify a formal series with an analytic function by notation alone. It proves summation bridges, matches ordinary and factorial-normalized coefficients, and carries zero multiplicities and logarithm choices into the analytic statements.

The completed formal contract has **68 full scoped manuscript results, zero partial, and zero unmapped**. The 151 map rows name 1,164 distinct declarations: 953 theorems and 211 constructions. The saved fail-closed namespace audit reports 2,234 declarations, including 1,999 theorem declarations. These are different counts. The namespace includes supporting material beyond the explicit map, while the map includes definitions as well as theorems. Neither is a count of independent mathematical discoveries.

1.2 Formal verification and source reconciliation

Evidence	Verified scope
Formal execution	The source contract is supported by the preserved aggregate build, 43 semantic checks, exact target-kind resolution, and kernel audit. The verification chapter records the separate replay using pinned sources for all 118 production modules.
Static reconciliation	This guide checks source checksums, exact map and contract parity, lexical source locations, target sets in the audit source and saved transcript, dependency pins, and excerpt identity.

The transcript permits only the disclosed standard transitive axioms `propext`, `Classical.choice`, and `Quot.sound`. This is a claim about the audited Lean environment and its declarations, not an assertion of assumption-free mathematics. Hypotheses appear in declaration types and remain part of each theorem.

1.3 The main route through the source

Layer	Mathematical work	Main map rows
Formal algebra	Residue filters, signed cycles, reciprocal powers, Appell and theta families	1–46
Entire realization	Finite exponential sums, summation bridges, Wronskians, matrix and ODE laws	47–56
Zero geometry	Dominance, local zeros, exhaustion, multiplicity, exact rays, counting	57–87

Layer	Mathematical work	Main map rows
Zeta continuation	Lattice comparison, finite exceptions, Hurwitz models, branches	88–95
Products and moments	Multiplicity-indexed products, growth rigidity, logarithmic derivatives	96–109
Growth and determinant	Exact indicator, order, trace-class diagonal realization	110–120
Normalized sections	Products, logarithms, analytic powers, Taylor coefficients	121–132
Pole asymptotics	Principal parts, removal, nearest-circle estimates, orbit reduction	133–142
Final closure modules	Pointwise filters and addition, branch laws, operator bridge, order one, determinants, hypergeometric and quotient exponentials	143–151

This ordering is a reading route, not the exact import order. The final nine modules supply 57 mapped declarations linking 15 manuscript records to their analytic or literal displayed forms.

1.4 Reading a Lean declaration

An expression such as `(hN : 0 < N)` is an explicit hypothesis. Braces indicate implicit arguments whose values Lean may infer. They do not erase hypotheses. A proposition such as `HasSum f x` asserts convergence of the indexed sum to the displayed value. A bare `tsum` is an operation with a total definition and does not, by itself, assert that a series converges. The same distinction applies to `Multipliable` and an infinite product.

Formal power series, polynomials, complex functions, and continuous linear maps inhabit different types. The transition between them is usually the mathematically important theorem. In this guide, a short final proof often rests on substantial earlier modules. A proof of the final Hadamard identity in two lines is not the whole product argument.

The code excerpts are literal source intervals, including their proofs. They are reading excerpts inside the original module context, not independent files that can be pasted without imports, variables, namespace declarations, or local instances. The complete source is supplied. The excerpt manifest records each original file, line interval, file hash, excerpt hash, and display hash.

1.5 Boundaries that completion preserves

Imported standard fact 11.1 remains imported and is outside the 68-result contract. The checked order proofs do not need it. The work does not claim universal simplicity of all zeros, positivity of every reduced zero, a full displacement expansion, or a zeta functional equation. Eventual simplicity and exact asymptotic statements are proved with their stated ranges. Nearest-pole results keep their integer reciprocal-order and local pole hypotheses. Complex powers use a formal series, a normalized germ, or one fixed holomorphic logarithm on the stated domain.

Chapter IV's diagonal determinant is a concrete scalar-family specialization. It is not the general bounded-operator theory allocated to Chapter VI, the arithmetic interpolation theory of Chapter IX, or the abstract completion questions of Chapter X.

2 Coefficient algebra and normalization

2.1 The projector acts on coefficients

For a formal series $f(X) = \sum_n a_n X^n$, the cyclic projector retains exactly those coefficients for which $n \equiv j \pmod N$. The implementation uses the total natural-number remainder operation. Idempotence makes sense without a positive-order assumption. The finite resolution over $0 \le j < N$ requires $N > 0$.

Source: KomorebiPaperIV/CyclicFilters.lean.

```
def cyclicProjector (N j : ℕ) (f : R⟦X⟧) : R⟦X⟧ :=
  PowerSeries.mk fun n => if n % N = j % N then PowerSeries.coeff n f else 0
```

Source: KomorebiPaperIV/CyclicFilters.lean.

```
theorem cyclicProjector_idempotent (N j : ℕ) (f : R⟦X⟧) :
    cyclicProjector N j (cyclicProjector N j f) = cyclicProjector N j f := by
  ext n
  by_cases h : n % N = j % N <;> simp [h]
```

The proof of idempotence is coefficientwise extensionality. After selecting an index, a case split determines whether the residue matches. This is a universal identity in the source's context of a commutative semiring, not a computation on a bounded list of coefficients.

Source: KomorebiPaperIV/CyclicFilters.lean.

```
theorem sum_cyclicProjector {N : ℕ} (hN : 0 < N) (f : R⟦X⟧) :
    (∑ j ∈ Finset.range N, cyclicProjector N j f) = f := by
  ext n
  rw [map_sum]
  rw [Finset.sum_eq_single (n % N)]
  · simp [Nat.mod_eq_of_lt (Nat.mod_lt n hN)]
  · intro j hj hne
    have hjN : j < N := Finset.mem_range.mp hj
    have hmod : j % N = j := Nat.mod_eq_of_lt hjN
    have hres : n % N ≠ j % N := by simpa [hmod] using hne.symm
    simp [hres]
  · exact fun hmem => (hmem (Finset.mem_range.mpr (Nat.mod_lt n hN))).elim
```

The positive-order hypothesis ensures that n mod N lies in the finite index set. The sum has exactly one surviving term. This is also why the proof cannot simply discard the $N = 0$ boundary.

2.2 Multiplication adds residue labels

A product of two pure packets has residue equal to the sum of their labels modulo N. The full product law first reconstructs each factor from its packet sum, distributes the product, and projects every pair. Nonmatching label pairs vanish. The displayed Lean proof records this route directly.

Source: KomorebiPaperIV/CyclicFilters.lean.

```
theorem cyclic_product_law {N : ℕ} (hN : 0 < N) (k : ℕ) (f g : R⟦X⟧) :
    cyclicProjector N k (f * g) =
      ∑ i ∈ Finset.range N, ∑ j ∈ Finset.range N,
        if (i + j) % N = k % N then
          cyclicProjector N i f * cyclicProjector N j g
        else 0 := by
  calc
    cyclicProjector N k (f * g) =
        cyclicProjector N k
          ((∑ i ∈ Finset.range N, cyclicProjector N i f) *
            (∑ j ∈ Finset.range N, cyclicProjector N j g)) := by
      rw [sum_cyclicProjector hN, sum_cyclicProjector hN]
    _ = ∑ i ∈ Finset.range N, ∑ j ∈ Finset.range N,
          if (i + j) % N = k % N then
            cyclicProjector N i f * cyclicProjector N j g
          else 0 := by
      rw [Finset.sum_mul]
      simp_rw [Finset.mul_sum]
      rw [cyclicProjector_sum]
      apply Finset.sum_congr rfl
      intro i hi
      rw [cyclicProjector_sum]
      apply Finset.sum_congr rfl
      intro j hj
      by_cases hlabel : (i + j) % N = k % N
      · simp only [hlabel, ↓reduceIte]
        rw [cyclicProjector_eq_of_mod_eq (i := k) (j := (i + j) % N)
          (by simpa using hlabel.symm)]
        exact cyclicProjector_product_packet N i j f g
      · simp only [hlabel, ↓reduceIte]
        exact cyclicProjector_product_other (R := R) (fun h => hlabel h.symm) f g
```

No analytic interchange occurs here. Every coefficient of a formal product is a finite convolution, and the outer packet decomposition is finite. The pointwise counterpart, including covariance on invariant domains, is supplied later by row 143.

2.3 Transfer and signed sections

The transfer and signed coefficient selectors are deliberately separated from their exponential-generating normalization.

Source: KomorebiPaperIV/SectionCycles.lean.

```
def transferSectionCoeff (N j : ℕ) (lambda : R) (n : ℕ) : R :=
  if n % N = j then lambda ^ n else 0
```

Source: KomorebiPaperIV/SectionCycles.lean.

```
def signedSectionCoeff (N j : ℕ) (n : ℕ) : R :=
  if n % N = j then (-1 : R) ^ (n / N) else 0
```

The transfer selector contributes λ^n in one residue class. The signed selector contributes $(-1)^{\lfloor n/N \rfloor}$. The surrounding `exponentialGeneratingSeries` construction divides the nth selected coefficient by $n!$. Thus the signed section is

$$D_{N,j}(z) = \sum_{q \geq 0} \frac{(-1)^q z^{Nq+j}}{(Nq+j)!}, \qquad 0 \leq j < N.$$

Differentiating shifts the residue label downward. At label zero, crossing the period contributes a minus sign.

Source: KomorebiPaperIV/SectionCycles.lean.

```
theorem derivative_signedSection_zero {N : ℕ} (hN : 0 < N) :
    PowerSeries.derivative R (signedSection (R := R) N 0) =
      -signedSection (R := R) N (N - 1) := by
  rw [signedSection, derivative_exponentialGeneratingSeries, signedSection]
  ext n
  simp only [coeff_exponentialGeneratingSeries, map_neg]
  rw [signedSectionCoeff_succ_zero hN]
  ring
```

For a full period, the signed family solves the corresponding negative cyclic differential equation. The formal derivative cycle is proved before the entire-function derivative cycle. The later analytic bridge ensures that the same normalization appears in both.

2.4 Reduced and original variables

The distinguished denominator is the signed section with residue label zero in the original variable. It is written in a reduced variable $w = z^N$ through

$$D_N(z) = \Delta_N(z^N), \qquad \Delta_N(w) = \sum_{n \geq 0} \frac{(-1)^n w^n}{(Nn)!}.$$

Source: KomorebiPaperIV/ReciprocalCore.lean.

```
def reducedDenominator (N : ℕ) : R⟦X⟧ :=
  PowerSeries.mk fun n =>
    algebraMap ℚ R ((-1 : ℚ) ^ n / ((N * n).factorial : ℚ))
```

The reciprocal family is generated formally by the negative logarithm of the unit series Δ_N. Its constant coefficient is one, which is the algebraic condition needed to define the normalized logarithm and reciprocal family.

Source: KomorebiPaperIV/ReciprocalCore.lean.

```
def reciprocalSeries (N : ℕ) (r : R) : R⟦X⟧ :=
  formalExpFamily (negativeLogDenominator (R := R) N) r
```

Two normalizations coexist. Write

$$\Delta_N(w)^{-r} = \sum_{n\ge 0} b_n(r) w^n = \sum_{n\ge 0} P_{N,n}(r) \frac{w^n}{(Nn)!}.$$

Then $P_{N,n}(r) = (Nn)! b_n(r)$. The source's `reciprocalOrdCoeff` denotes the ordinary coefficient, while `reciprocalCoeff` uses `blockCoeff` and the $(Nn)!$ normalization.

Source: KomorebiPaperIV/ReciprocalCore.lean.

```
def reciprocalCoeff (N n : ℕ) (r : R) : R :=
  blockCoeff N n (reciprocalSeries N r)
```

The parameter-addition law has binomial factors at the block indices, not ordinary $\binom{n}{q}$ factors.

Source: KomorebiPaperIV/ReciprocalCore.lean.

```
theorem reciprocalCoeff_addition (N n : ℕ) (r s : R) :
    reciprocalCoeff N n (r + s) =
      ∑ q ∈ Finset.range (n + 1),
        ((N * n).choose (N * q) : R) *
          reciprocalCoeff N q r * reciprocalCoeff N (n - q) s := by
  exact reciprocal_block_convolution
    (constantCoeff_negativeLogDenominator N) N n r s
```

Dividing this identity by $(Nn)!$ recovers the ordinary Cauchy convolution for the b_n. Multiplying by the wrong factorial changes the theorem. The source makes that conversion an explicit named construction.

2.5 Newton recurrence and its inputs

The logarithmic coefficients `kappaCoeff` encode the logarithmic derivative of the reduced denominator. The formal differential identity for the reciprocal series gives a recurrence for its ordinary coefficients.

Source: KomorebiPaperIV/ReciprocalCore.lean.

```
theorem reciprocalOrdCoeff_newton (N : ℕ) (r : R) (n : ℕ) :
    (n + 1 : R) * reciprocalOrdCoeff N (n + 1) r =
      r * ∑ p ∈ Finset.antidiagonal n,
        reciprocalOrdCoeff N p.1 r * kappaCoeff N (p.2 + 1) := by
  simpa [reciprocalOrdCoeff, kappaCoeff] using
    reciprocalSeries_coefficient_recurrence (R := R) N r n
```

The antidiagonal index contains pairs with sum n. Its second component is shifted by one to match the logarithmic derivative. This is the same recurrence later specialized to a literal Hessenberg determinant. It is not necessary to assume that any zero sum converges at this stage. Zero moments are subsequently proved equal to these algebraic coefficients.

2.6 Appell and theta families

The generic Appell polynomial is built from a formal exponential family and then differentiated as a polynomial.

Source: KomorebiPaperIV/Appell.lean.

```
def formalAppellPolynomial (L : R⟦X⟧) (r : R) (m : ℕ) : Polynomial R :=
  appellPolynomial (fun q => egfCoeff q (formalExpFamily L r)) m
```

Source: KomorebiPaperIV/Appell.lean.

```
theorem derivative_formalAppellPolynomial (L : R⟦X⟧) (r : R) (m : ℕ) :
    Polynomial.derivative (formalAppellPolynomial L r (m + 1)) =
      Polynomial.C (m + 1 : R) * formalAppellPolynomial L r m := by
  exact derivative_appellPolynomial _ m
```

The derivative law holds because polynomial differentiation is finite. The residue-specialized Appell family also has parameter addition, covariance, a sparse coefficient formula, and an inverse differential-operator action. Row 146 proves that the ordinary-derivative and factorial/Hasse encodings agree. Both interpretations have an explicit bridge.

The theta construction combines two rescalings of the reciprocal family.

Source: KomorebiPaperIV/ThetaCore.lean.

```
def thetaSeries (N : ℕ) (u : R) : R⟦X⟧ :=
  PowerSeries.rescale (-u ^ N) (reciprocalSeries N 1) *
    PowerSeries.rescale (-1) (reciprocalSeries N u)
```

The rescaling factors $-u^N$ and -1 are part of the definition. Theta reconstruction, parameter factors, polynomial degree, and Newton recurrence are covered by rows 22–23 and 43–45. A later analytic zero-data formula identifies their coefficients with the corresponding Bell expressions. The formal identities remain available without importing that analytic interpretation.

3 From formal series to entire functions

3.1 Finite exponential sums are the analytic realization

The entire transfer function is defined as a finite root-weighted exponential sum.

Source: KomorebiPaperIV/EntireBridge.lean.

```
def entireTransfer (N j : ℕ) (omega lambda : ℂ) (z : ℂ) : ℂ :=
  (N : ℂ)⁻¹ *
    ∑ l ∈ Finset.range N,
      ((omega ^ j)⁻¹) ^ l * Complex.exp ((lambda * omega ^ l) * z)
```

The primitive root, residue index, and spectral parameter appear explicitly. The finite sum is entire because each exponential summand is entire. The crucial next step is a `HasSum` theorem identifying its Taylor coefficients with the formal transfer section.

Source: KomorebiPaperIV/EntireBridge.lean.

```
theorem hasSum_entireTransfer {N j : ℕ} (hN : 0 < N) (hj : j < N)
    {omega : ℂ} (homega : IsPrimitiveRoot omega N) (lambda z : ℂ) :
    HasSum (fun n => PowerSeries.coeff n (transferSection N j lambda) * z ^ n)
      (entireTransfer N j omega lambda z) := by
  simpa only [finiteRootAverage_rescale_exp_eq_transferSection hN hj homega]
    using hasSum_finiteRootAverage N j omega lambda z
```

This theorem has positive order, a residue index in range, and a primitive root as hypotheses. It provides a convergent series at every complex argument. It also permits independence-of-root proofs: two finite sums obtained from allowed root choices have the same summing series, hence the same value by uniqueness of a sum.

The signed analytic realization additionally chooses ρ with $\rho^N = -1$. The factor ρ^{-j} compensates for the residue label. It cannot be removed without changing the normalization of the basis. The independent-root statements compare the resulting functions under exactly the root assumptions they require.

3.2 Initial data and Wronskians

The formal cycle specifies derivatives at zero, while the analytic bridge transfers them to iterated complex derivatives. For the normalized signed basis, the matrix of initial data is the identity. Constancy of the Wronskian then identifies the Wronskian with one for the stated $N > 1$ range.

Source: KomorebiPaperIV/Wronskians.lean.

```
theorem signedWronskian_eq_one {N : ℕ} (hN : 1 < N) {omega rho : ℂ}
    (homega : IsPrimitiveRoot omega N) (hrho : rho ^ N = -1) (z : ℂ) :
    analyticWronskian (fun j : Fin N => entireSigned N j.val omega rho) z = 1 := by
  rw [signedWronskian_constant hN homega]
  have hm : (fun k j : Fin N => deriv^[k.val]
      (entireSigned N j.val omega rho) 0) = (1 : Matrix (Fin N) (Fin N) ℂ) := by
    ext k j
```

```
    rw [entireSigned_initial_data (by omega) j.isLt k.isLt homega hrho]
    simp [Matrix.one_apply, Fin.ext_iff]
  rw [analyticWronskian, hm, Matrix.det_one]
```

The source does not conclude linear independence from an informal appeal to distinct exponentials alone. It factors the exponential jet matrix, handles its determinant, proves the Wronskian law, and identifies the normalized initial matrix. The ODE modules then describe the corresponding realization of the solution space.

3.3 Companion matrices and finite algebras

The signed shift matrix has a negative carry across the cyclic boundary. Its powers encode paths through the residue labels. This provides a companion realization of the signed differential cycle, explicit fundamental-matrix entries, addition and inversion identities, and a spectrum controlled by the chosen roots.

There are two compatible algebra models. In the cyclic quotient algebra the generator satisfies $x^N = 1$. In the signed quotient it satisfies $x^N = -1$. Earlier modules construct the quotient algebras and their power bases, then prove formal exponential identities. Row 151 equips the finite-dimensional quotients with coefficient norm topologies and proves convergence of the actual exponential series. Consequently, the analytic identity in a finite algebra is more than equality of formal coefficient expressions.

3.4 Why the final pointwise addition theorem matters

A bivariate addition law for formal series is an algebraic identity among coefficients. It does not on its own prove equality of evaluated functions at arbitrary complex arguments. The completed source includes both. Row 144 obtains the actual transfer and signed addition formulas, retaining the signed carry factor forced by $\rho^N = -1$.

The useful reading order is: formal product and addition identities, the entire summation bridge, then the pointwise closure. Keeping these stages visible prevents a notation change from hiding the convergence theorem. It also makes the formal statements reusable in contexts where no analytic topology has been chosen.

3.5 Growth order and indicator geometry

The entire functions are finite exponential sums. Their exponents form regular polygon configurations in the complex plane. In directions with a unique exposed vertex, one exponential dominates. On tie directions, cancellation can obstruct a naive pointwise lower bound, so the exact indicator proof uses aligned radii in addition to uniform sector estimates.

Rows 110–115 prove sharp order, exponential type, indicator geometry, uniform dominance in sectors, and the exact directional indicator. These results are proved in the source. Imported standard fact 11.1 remains recorded separately and is not required for the checked order proofs. The guide therefore does not count that imported fact among the 68 completed results.

4 Zero geometry, exhaustion, and counting

4.1 Local zeros are only the start

The two-term dominance analysis separates the leading pair of exponentials from a controlled remainder. Near the cancellation lattice of the leading pair, the perturbation argument constructs nearby zeros and estimates their displacement. The remainder bounds include derivative information, which supports simplicity and the first correction term.

A local construction alone would not justify counting all zeros. The source therefore adds sector confinement, sector exhaustion, rotation, and global exhaustion. Those modules show that sufficiently large zeros belong to the constructed symmetry strings. Finite exceptional zeros remain present throughout the later counting and product arguments.

The following division of labor is useful when reviewing the proof path:

Stage	Output	Map rows
Remainder control	Canonical polygon gap, strip and disk bounds, normalized jets	58–60, 67
Local perturbation	Nearby zeros, displacement, derivative estimates	57, 61
Global coverage	Sector confinement, exhaustion, rotational orbits	62–65
Simplicity and rays	All sufficiently large zeros simple and on exact rays	66, 68
Reduced variable	Zero and multiplicity transfer under $z \mapsto z^N$	69–71
Complete enumeration	Constructed tail plus finite exceptions	75–79, 81
Refined phase	Real leading correction and alternating derivative sign	82–84

4.2 Named enumeration is constructed, not assumed

The `SignedZeroEnumeration` structure packages a positive simple tail, its lattice windows, rotation orbits, and the finite-complement information needed downstream. Existence of this structure is proved for every $N \geq 2$, with the exact order-two cosine model handled separately.

A downstream theorem may accept an enumeration as input because it is reusable for any structure with the required fields. That is different from assuming the final zero theorem. The manuscript specializations instantiate it using `canonicalZeroEnumeration`, constructed from the preceding analytic results. To audit a final theorem, follow that constructor as well as the generic lemma.

4.3 Multiplicity enters the actual counting function

The disk counting function sums `analyticOrderNatAt` over the actual finite zero set. The source proves that these analytic orders are finite for the nonzero entire denominators. Thus replacing an extended order by its natural-number value does not silently discard an infinite order.

The full original-variable counting theorem keeps every exceptional multiplicity:

Source: `KomorebiPaperIV/FullZeroCounting.lean`.

```
theorem canonicalSignedDenominator_zero_counting {N : ℕ} (hN : 2 ≤ N) :
    ∃ K : ℝ, 0 < K ∧ ∀ R : ℝ, 0 ≤ R →
      |(analyticZeroCount (canonicalSignedDenominator N) R : ℝ) -
```

```
      ((N : ℝ) * Real.sin (Real.pi / N) / Real.pi) * R| ≤ K := by
let e := canonicalZeroEnumeration hN
let W := finiteWeight (analyticOrderNatAt (canonicalSignedDenominator N)) e.exceptions
refine ⟨(W : ℝ) + N * ((e.shift : ℝ) + 2) + 1, by positivity, ?_⟩
intro R hR
have hw := e.weighted_count_bounds hN hR
have hl : (N : ℝ) * sequenceCounting e.root R ≤ analyticZeroCount (canonicalSignedDenominator N)
 ↪  R := by
  exact_mod_cast hw.1
have hu : (analyticZeroCount (canonicalSignedDenominator N) R : ℝ) ≤
    N * sequenceCounting e.root R + W := by exact_mod_cast hw.2
have hc := (lattice_window_counting (div_pos Real.pi_pos (signed_sine_pos hN))
  e.shift e.root e.windows hR).2
rw [abs_le] at hc ⊢
have he : ((N : ℝ) * Real.sin (Real.pi / N) / Real.pi) * R =
    N * (R / (Real.pi / Real.sin (Real.pi / N))) := by field_simp
rw [he]
have hn := Nat.cast_nonneg (α := ℝ) N
constructor <;> nlinarith
```

The proof compares the full count with the simple tail count and adds a finite exception weight. The final error constant contains both that weight and the tail shift. It is uniform for all nonnegative radii, not only an eventual asymptotic abbreviation.

The corresponding reduced-variable count is

$$n_{\Delta_N}(R) = \frac{\sin(\pi/N)}{\pi} R^{1/N} + O(1), \qquad N \geq 2.$$

The original-variable constant is $N \sin(\pi/N)/\pi$. These constants differ because one original symmetry orbit collapses to one reduced zero and the radius changes by the Nth power. Both transformations are proved in the counting modules.

4.4 Sharp summability thresholds

Rows 72 and 79 distinguish a tail estimate from the full theorem. The full reciprocal-power sums range over actual zeros with their analytic multiplicities. Their sharp thresholds are $s > 1$ for the original denominator and $s > 1/N$ for the reduced denominator. The boundary divergences are included. Removing finite exceptions is justified by a proved summability equivalence, not by silently redefining the zero set.

For $N \geq 2$, reduced reciprocal zeros are absolutely summable because $1 > 1/N$. This precise consequence feeds the genus-zero product and the trace-class diagonal model. It would fail as a route at $N = 1$, where the denominator is exponential and the analytic situation is handled separately.

4.5 What eventual simplicity leaves open

The source proves eventual simplicity and exact rays for sufficiently large zeros. It does not replace this with a blanket assertion that all zeros are simple. A finite exceptional zero can have multiplicity greater than one, and the canonical product includes it accordingly. A finite exceptional zero can also lie outside the asymptotic positive-ray description. The global product and zero-zeta constructions are designed to retain such exceptions.

The first displacement correction and its exponential error are likewise specific proved statements. They do not constitute an all-orders expansion. The explicit cubic correction in row 84 is a specialization with exact spectral gap, lattice centers, and phase, not a numerical fit to plotted zeros.

5 Canonical products, moments, and the diagonal model

5.1 The index set records multiplicity

The product index is a dependent pair. Its first component chooses an actual zero. Its second chooses one of the copies counted by that zero's analytic multiplicity.

Source: KomorebiPaperIV/FullCanonicalProduct.lean.

```
def ZeroMultiplicityIndex (f : ℂ → ℂ) :=
  Σ z : {z : ℂ // f z = 0}, Fin (analyticOrderNatAt f z)
```

Source: KomorebiPaperIV/FullCanonicalProduct.lean.

```
def fullCanonicalProduct (f : ℂ → ℂ) (w : ℂ) : ℂ :=
  ∏' i : ZeroMultiplicityIndex f, (1 - w / indexedZero i)
```

For a zero of order m, the dependent finite type contributes exactly m copies of its factor. The construction therefore does not require a global simplicity assumption. Its factor uses the reduced variable w.

5.2 Absolute summability precedes the product theorem

The weighted reciprocal-zero sum is converted to a sum over the multiplicity index.

Source: KomorebiPaperIV/FullCanonicalProduct.lean.

```
theorem zeroMultiplicity_reciprocal_summable {f : ℂ → ℂ} (hf : ZeroPowerSummable f 1) :
    Summable (fun i : ZeroMultiplicityIndex f => 1 / ‖indexedZero i‖) := by
  apply (summable_sigma_of_nonneg (fun i : ZeroMultiplicityIndex f => by positivity)).mpr
  constructor
  · intro z
    exact (hasSum_fintype _).summable
  · simpa only [indexedZero, tsum_fintype, Finset.sum_const, Finset.card_univ,
      Fintype.card_fin, nsmul_eq_mul, mul_one_div, ZeroPowerSummable, Real.rpow_one] using hf
```

The next source results prove that the perturbations w/ρ are absolutely summable, the product is multipliable, convergence is uniform on compact sets and locally uniform, and the product defines an entire function. These are needed analytic statements. Writing an infinite product symbol does not establish them.

The product's zeros and multiplicities are then identified. Dividing the denominator by the product gives an entire, nonvanishing quotient. Growth estimates and a sublinear rigidity argument force the normalized quotient to be one. Only after those steps is the final identity short:

Source: KomorebiPaperIV/HadamardIdentification.lean.

```
theorem entireReducedDenominator_eq_fullCanonicalProduct {N : ℕ} (hN : 2 ≤ N) :
    entireReducedDenominator N = fullCanonicalProduct (entireReducedDenominator N) := by
  funext z
  have he := (entireReducedDenominator_product_quotient hN).2.2.2 z
  rw [entireReducedDenominator_quotient_eq_one hN, Pi.one_apply, mul_one] at he
  exact he.symm
```

The proof depends on the product-quotient theorem and the quotient-equals-one theorem. Those in turn depend on convergence, exact multiplicities, growth reduction, and Jensen–Poisson estimates. Rows 96–102 expose these stages in the map. The earlier tail product in row 74 is one ingredient, not the complete identification.

5.3 Recovering coefficients from zeros

After the full product is identified, its logarithmic derivative gives a convergent Mittag–Leffler expansion away from the zeros. Rows 104–109 match derivatives and analytic Taylor coefficients with the actual multiplicity-weighted zero moments. The coefficients previously called `kappaCoeff` are thereby identified with zero sums.

This direction matters. The coefficient recurrence was proved algebraically before any product formula. The analytic argument does not define the denominator's coefficients by a conjectured zero expansion. It proves that the coefficients already defined equal the zero data. Tangent coefficients, reciprocal Bell formulas, and theta Bell formulas then inherit exact normalizations through named bridges.

Normalized nonzero residue sections need their own growth and zero arguments. Rows 121–132 cover their entire realization, zero-growth estimates, summability, product identification, logarithms, analytic powers, and Taylor-differential conversions. They are not inferred merely by relabeling the distinguished denominator.

5.4 A concrete trace-class definition

The source uses a rank-one summability definition for Hilbert trace class.

Source: `KomorebiPaperIV/DiagonalNuclear.lean`.

```
def IsHilbertTraceClass {E : Type*} [NormedAddCommGroup E] [InnerProductSpace ℂ E]
    (T : E →L[ℂ] E) : Prop :=
  ∃ (ι : Type) (x y : ι → E), Summable (fun i => ‖x i‖ * ‖y i‖) ∧
    HasSum (fun i => rankOne ℂ (x i) (y i)) T
```

The witness is a family of rank-one operators with summable products of vector norms and a sum equal to the continuous linear map. For the diagonal model, reciprocal reduced zeros form the eigenvalue family. Their absolute summability is proved from the zero theorem, then used to build the operator and its trace-class witness.

The source also proves the basis action, power traces, finite compression determinants, and their limit. The exact determinant identity is:

Source: `KomorebiPaperIV/DiagonalFredholm.lean`.

```
theorem reducedZeroOperator_fredholm {N : ℕ} (hN : 2 ≤ N) (w : ℂ) :
    diagonalFredholmDeterminant (reducedZeroEigenvalue N) w = entireReducedDenominator N w := by
  calc
    _ = fullCanonicalProduct (entireReducedDenominator N) w := by
      simp only [diagonalFredholmDeterminant, reducedZeroEigenvalue, fullCanonicalProduct,
        ↪ div_eq_mul_inv]
    _ = _ := congrFun (entireReducedDenominator_eq_fullCanonicalProduct hN).symm w
```

Here `diagonalFredholmDeterminant` is the defined diagonal specialization, backed by convergence of finite compressions. The identity follows by matching its factors to the multiplicity-indexed canonical

product. It does not assert a general Fredholm determinant theory for arbitrary operators or conceal such a theorem behind the name.

5.5 Complete symmetric functions

The reciprocal of the diagonal determinant has complete homogeneous symmetric-function coefficients. Rows 119–120 handle the finite combinatorics and the infinite summability argument. The latter cannot be replaced by an unqualified passage from finite-dimensional symmetric polynomials to infinitely many variables. The summable eigenvalue family is the analytic input that makes the coefficient expression meaningful.

This yields three linked descriptions of the same scalar family: formal reciprocal coefficients, zero moments and symmetric functions, and a diagonal determinant model. Each identification is a theorem with a specified source and normalization.

6 Pole coefficients and zero zeta functions

6.1 Nearest-pole estimates retain a local analytic contract

For a positive integral reciprocal order, a simple denominator zero produces a pole of finite order. The source constructs principal parts, proves their uniqueness, removes them, and bounds the Taylor coefficients of the analytic remainder. This produces the nearest-circle coefficient expansion with a controlled error.

The generic coefficient bound below states its local principal-part assumption explicitly. Later source results construct the required principal parts for the manuscript specialization.

Source: `KomorebiPaperIV/NearestPoleExpansion.lean`.

```
theorem reciprocalCoeff_nearest_pole_bound {N : ℕ} (hN : 0 < N)
    {S : Finset ℂ} (hS : ∀ ρ ∈ S, ρ ≠ 0) (a : ℂ → ℕ → ℂ) (r : ℕ)
    {R R₂ : ℝ} (hR : 0 < R) (hR₂ : R < R₂)
    (hzeros : ∀ z : ℂ, ‖z‖ < R₂ → canonicalSignedDenominator N z = 0 → z ∈ S)
    (hlocal : ∀ ρ ∈ S, ∃ C : ℝ, ∀ᶠ z in 𝓝[≠] ρ,
      ‖(canonicalSignedDenominator N z ^ r)⁻¹ - polePrincipalPart a r ρ z‖ ≤ C) :
    ∃ C : ℝ, ∀ n : ℕ,
      ‖reciprocalCoeff N n (r : ℂ) - ((N*n).factorial : ℂ) *
        ∑ ρ ∈ S, ∑ k ∈ Finset.range r, a ρ (k+1) * ((N*n+k).choose k : ℂ) * ρ⁻¹ ^ (N*n)‖ ≤
      C * ((N*n).factorial : ℝ) * R⁻¹ ^ (N*n) := by
  have hf : Differentiable ℂ (canonicalSignedDenominator N) := fun z => (analyticAt_entireSigned
   ↪  ..).differentiableAt
  obtain ⟨C, hC⟩ := nearest_pole_taylor_bound hf (by rw [canonicalSignedDenominator_zero hN];
   ↪  exact one_ne_zero)
    hS a r hR hR₂ hzeros hlocal
  refine ⟨C, fun n => ?_⟩
  rw [reciprocalCoeff_analyticTaylor hN, ← mul_sub, norm_mul, Complex.norm_natCast]
  have h := mul_le_mul_of_nonneg_left (hC (N*n)) (show (0 : ℝ) ≤ (N*n).factorial by positivity)
  simpa only [div_eq_mul_inv, inv_pow, mul_assoc, mul_comm, mul_left_comm] using h
```

The finite set S contains the zeros inside the larger disk. The inner radius R is strictly smaller than the outer radius R_2. The assumption `hlocal` states that after subtracting the proposed principal part, the function is locally bounded near each selected pole. These are the hypotheses needed by the removable-singularity and Taylor-bound route.

The conclusion is in the manuscript's block normalization. Its leading sum and error both carry $(Nn)!$. The factor ρ^{-Nn} arises from extracting the original-variable coefficient at degree Nn. The binomial coefficient records the higher pole order. Omitting any of these factors changes the asymptotic statement.

6.2 Principal parts are constructed before the final specialization

Rows 133–138 provide coefficient bounds, principal-part coefficients, pole removal, Taylor recovery, simple-pole expansions, and uniqueness. Row 139 supplies the full nearest-pole expansion. Row 140 isolates the leading term under its stated dominance hypotheses. Row 141 reduces symmetry orbits on the nearest circle. Row 142 gives the secant specialization.

This route is conditional in the mathematically ordinary sense that the theorem retains its hypotheses on reciprocal order, selected zeros, radii, and simplicity where used. Formal completion proves those

scoped statements. It does not delete the assumptions or claim that every parameter and every zero configuration satisfies them.

6.3 Absolute zero zeta and branch-dependent zeta differ

The absolute zero zeta uses terms of the form

$$m(\rho)\exp(-s\log|\rho|),$$

where $m(\rho)$ is analytic multiplicity and the real logarithm of the norm is unambiguous. The reduced denominator has sharp absolute-convergence half-plane $\operatorname{Re} s > 1/N$. The original denominator has $\operatorname{Re} s > 1$. Local uniform convergence and holomorphy are proved, as is failure of absolute convergence on and to the left of the respective boundaries.

A branch-dependent symmetry-orbit zeta instead assigns complex logarithms. Its terms carry argument information. The source requires compatible logarithm choices and records the phase factors in its continuation. These are related constructions, not interchangeable notations for one function.

6.4 Continuation is an explicit model plus an entire correction

Exponential displacement of the positive zero sequence from a lattice makes the difference of the two zeta sums entire. The source proves a locally uniform majorant on bounded sets of the parameter. It then constructs the lattice correction, finite missing lattice prefix, and finite exceptional-zero contribution.

Source: KomorebiPaperIV/ZeroZetaContinuation.lean.

```
def reducedZetaCorrection {N : ℕ} (hN : 2 ≤ N) (s : ℂ) : ℂ :=
  let e := canonicalZeroEnumeration hN
  e.reducedExceptionZeta (by omega) s + e.latticeCorrection ((N : ℂ) * s) -
    e.latticePrefix ((N : ℂ) * s)
```

The finite exception term retains multiplicity. The correction and prefix use the scaled parameter Ns because reduced roots are Nth powers of the positive original roots. The equality with the Hurwitz lattice model is proved in the initial convergence domain:

Source: KomorebiPaperIV/ZeroZetaContinuation.lean.

```
theorem entireReducedDenominator_zeta_eq_hurwitz {N : ℕ} (hN : 2 ≤ N)
    {s : ℂ} (hs : 1 / (N : ℝ) < s.re) :
    absoluteZeroZeta (entireReducedDenominator N) s =
      hurwitzLatticeModel (Real.pi / Real.sin (Real.pi / N)) N s + reducedZetaCorrection hN s :=
      ↪ by
  let e := canonicalZeroEnumeration hN
  have hsN : 1 < ((N : ℂ) * s).re := by
    simp only [Complex.mul_re, Complex.natCast_re, Complex.natCast_im, zero_mul, sub_zero]
    exact (div_lt_iff₀ (by exact_mod_cast (show 0 < N by omega) : (0 : ℝ) < N)).mp hs |>.trans_eq
    ↪ (mul_comm _ _)
  rw [e.reducedZeta_decomposition hN hs, e.root_zeta_eq_model hN hsN]
  simp only [hurwitzLatticeModel, Nat.cast_one, one_mul, reducedZetaCorrection]
  dsimp [e]
  ring
```

The continuation is then defined by the meromorphic model plus its entire correction.

Source: KomorebiPaperIV/ZeroZetaContinuation.lean.

```
theorem reducedZetaContinuation_meromorphic {N : ℕ} (hN : 2 ≤ N) (s : ℂ) :
    MeromorphicAt (reducedZetaContinuation hN) s :=
  (hurwitzLatticeModel_meromorphicAt _ _ s).add
    ((reducedZetaCorrection_entire hN).analyticAt s).meromorphicAt
```

This theorem is short because its two analytic ingredients have already been proved. The continuation is tied to the original sum by the preceding equality. The source also proves holomorphy off the pole, exact residues, and meromorphic order -1 at the nonzero-residue pole. It does not merely define a meromorphic function and call it a continuation.

6.5 Derivative weights and boundary orders

Derivative-weighted sums use complex derivative values, not only their moduli. A lower bound on the derivative at sufficiently large zeros gives exponential decay of its inverse powers. For $N \geq 3$ and positive integral derivative weight, this dominates every fixed polynomial weight and yields an entire sum, assuming logarithm arguments remain in a fixed bounded interval.

The order-two case is separate: the derivative magnitude does not provide the same exponential gain, and the theorem retains a right half-plane of convergence. Rows 85–87 state and prove this distinction. Exact order-two and order-four zeta models are separately identified in rows 94–95.

7 The nine modules that close the manuscript contract

7.1 Row 143: actual pointwise projectors

The earlier coefficient projector and formal Fourier average are joined to the literal finite average on functions.

Source: KomorebiPaperIV/PointwiseFilters.lean.

```
def pointwiseRootAverage (N j : ℕ) (omega : ℂ) (f : ℂ → ℂ) (z : ℂ) : ℂ :=
  (N : ℂ)⁻¹ * ∑ l ∈ Finset.range N, ((omega ^ j)⁻¹) ^ l * f (omega ^ l * z)
```

The projector law is proved for actual pointwise values:

Source: KomorebiPaperIV/PointwiseFilters.lean.

```
theorem pointwiseRootAverage_projector {N : ℕ} (hN : 0 < N) {omega : ℂ}
    (ho : IsPrimitiveRoot omega N) (i j : ℕ) (f : ℂ → ℂ) (z : ℂ) :
    pointwiseRootAverage N i omega (pointwiseRootAverage N j omega f) z =
      if j % N = i % N then pointwiseRootAverage N j omega f z else 0 := by
  exact pointwiseRootAverage_of_covariant hN ho (mapsTo_univ _ _) i j _
    (fun w _ => pointwiseRootAverage_covariant hN ho j f w) (mem_univ z)
```

On an invariant domain, the range characterization also tracks differentiability and equality restricted to that domain.

Source: KomorebiPaperIV/PointwiseFilters.lean.

```
theorem pointwiseRootAverage_range_iff {N : ℕ} (hN : 0 < N) {omega : ℂ}
    (ho : IsPrimitiveRoot omega N) {s : Set ℂ}
    (hs : MapsTo (fun z => omega * z) s s) (j : ℕ) {f : ℂ → ℂ}
    (hf : DifferentiableOn ℂ f s) :
    (∃ g : ℂ → ℂ, DifferentiableOn ℂ g s ∧ EqOn (pointwiseRootAverage N j omega g) f s) ↔
      ∀ z ∈ s, f (omega * z) = omega ^ j * f z := by
  constructor
  · rintro ⟨g, _, hg⟩ z hz
    rw [← hg (hs hz), ← hg hz]
    exact pointwiseRootAverage_covariant hN ho j g z
  · intro hc
    refine ⟨f, hf, fun z hz => ?_⟩
    simpa using pointwiseRootAverage_of_covariant hN ho hs j j f hc hz
```

The domain condition says multiplication by the primitive root maps the set into itself. This is needed to apply covariance repeatedly along an orbit without evaluating a domain-restricted hypothesis outside its scope. The proof of the reverse implication chooses the function itself as a preimage. The forward implication uses covariance of the finite root average. Row 143 also includes the full pointwise product law and coefficient-series summation, closing the formal-to-holomorphic clauses of results 3.1–3.3.

7.2 Row 144: arbitrary-argument addition

`PointwiseAddition` supplies the actual entire transfer addition theorem and the signed addition theorem for arbitrary complex arguments. The signed carry is derived from the relation $\rho^N = -1$. This completes results 4.4 and 5.5 together with their earlier formal rows. No unspecified analytic evaluation map is left between the bivariate formal identity and the pointwise equality.

7.3 Row 145: one fixed logarithm on the domain

A complex reciprocal power needs a logarithm choice. The source makes it part of the data:

Source: `KomorebiPaperIV/AnalyticReciprocalLaws.lean`.

```
def IsLogarithmOn (f L : ℂ → ℂ) (s : Set ℂ) : Prop :=
  DifferentiableOn ℂ L s ∧ ∀ z ∈ s, Complex.exp (L z) = f z
```

Source: `KomorebiPaperIV/AnalyticReciprocalLaws.lean`.

```
def logarithmPower (L : ℂ → ℂ) (r : ℂ) (z : ℂ) : ℂ :=
  Complex.exp (-r * L z)
```

For one chosen L, the parameter law follows directly from the exponential addition identity.

Source: `KomorebiPaperIV/AnalyticReciprocalLaws.lean`.

```
theorem logarithmPower_add (L : ℂ → ℂ) (r t z : ℂ) :
    logarithmPower L r z * logarithmPower L t z = logarithmPower L (r + t) z := by
  unfold logarithmPower
  rw [← Complex.exp_add]
  congr 1
  ring
```

Differentiation requires an open domain and a holomorphic logarithm of the actual denominator.

Source: `KomorebiPaperIV/AnalyticReciprocalLaws.lean`.

```
theorem reciprocal_fixed_log_derivative {N : ℕ} {L : ℂ → ℂ} {s : Set ℂ}
    (hs : IsOpen s) (hL : IsLogarithmOn (canonicalSignedDenominator N) L s)
    (r : ℂ) {z : ℂ} (hz : z ∈ s) :
    deriv (logarithmPower L r) z =
      r * analyticTransferTangent N z * logarithmPower L r z :=
  (hasDerivAt_logarithmPower hs hL r hz).deriv
```

The source also proves the differential Bell/Riccati identity at points where the denominator is nonzero. These statements close results 7.4, 8.4, and 8.5 alongside their formal coefficient laws. They do not choose unrelated logarithms for different parameters and then assert a branch-independent multiplicative law.

7.4 Row 146: ordinary derivatives and Hasse derivatives

The formal differential operator acts on polynomials, so its apparently infinite series truncates. There are two encodings: ordinary iterated derivatives and factorial-normalized Hasse derivatives. The bridge is explicit:

Source: KomorebiPaperIV/OperatorEncodingBridge.lean.

```
theorem truncatedDifferentialOperator_egfAction (A : R⟦X⟧) (p : Polynomial R) :
    truncatedDifferentialOperator p.natDegree A p =
      egfConstantCoeffAction (fun q => (q.factorial : R) * PowerSeries.coeff q A) p := by
  rw [truncatedDifferentialOperator_apply, egfConstantCoeffAction]
  apply Finset.sum_congr rfl
  intro q _
  have h := congrFun (Polynomial.factorial_smul_hasseDeriv (R := R) q) p
  change q.factorial • (Polynomial.hasseDeriv q p) = Polynomial.derivative^[q] p at h
  rw [← h]
  simp only [nsmul_eq_mul, map_mul, map_natCast]
  ring
```

The proof uses the identity relating the qth Hasse derivative to the ordinary qth derivative after multiplication by $q!$. The source then proves the forward action on the named residue Appell polynomial and its inverse.

Source: KomorebiPaperIV/OperatorEncodingBridge.lean.

```
theorem residueAppell_operator_inverse (N j : ℕ) (hN : 0 < N) (r : R) (m : ℕ) :
    truncatedDifferentialOperator m
        (PowerSeries.expand N hN.ne' (residuePowerSeries N j (-r)))
        (residueAppellPolynomialParam N j hN r m) = Polynomial.X ^ m := by
  rw [← residueAppell_operator_forward N j hN r m]
  exact truncatedDifferentialOperator_residuePowerSeries_inverse N j hN r m
    (by simp)
```

The inverse parameter is $-r$. The expansion of the residue series changes the differential powers by multiples of N. The polynomial degree makes the action finite, so the result does not require a Banach-space operator convergence theorem.

7.5 Row 147: the order-one tangent boundary

The order-one denominator is identified exactly:

Source: KomorebiPaperIV/TangentBoundary.lean.

```
theorem canonicalSignedDenominator_one (z : ℂ) :
    canonicalSignedDenominator 1 z = Complex.exp (-z) := by
  have h := entireSigned_independent_roots (N := 1) (j := 0) (by omega) (by omega)
    (canonicalPrimitiveRoot_isPrimitive (by omega)) IsPrimitiveRoot.one
    (canonicalSignedRoot_pow (by omega)) (show (-1 : ℂ) ^ 1 = -1 by simp) z
  change canonicalSignedDenominator 1 z = _ at h
  simpa [entireSigned, entireTransfer] using h
```

It follows that the transfer tangent is one and its coefficients have the corresponding degenerate form. This boundary case is joined to the earlier $N \geq 2$ convergence theorem to obtain every positive order.

Source: KomorebiPaperIV/TangentBoundary.lean.

```
theorem analyticTransferTangent_hasSum_all_orders {N : ℕ} (hN : 0 < N) {R : ℝ}
    (hn : ∀ w ∈ Metric.ball (0 : ℂ) R, entireReducedDenominator N w ≠ 0)
    {z : ℂ} (hz : z ^ N ∈ Metric.ball (0 : ℂ) R) :
    HasSum (fun n => transferTangentCoeff (R := ℂ) N n /
      ((N * n + N - 1).factorial : ℂ) * z ^ (N * n + N - 1))
      (analyticTransferTangent N z) := by
  by_cases h : N = 1
  · subst N
    exact analyticTransferTangent_one_hasSum z
  · exact analyticTransferTangent_coefficient_hasSum (by omega) hn hz
```

Notice the original-variable exponent $Nn + N - 1$ and factorial $(Nn + N - 1)!$. The nonvanishing hypothesis is stated in the reduced-variable ball, while the evaluation point is tested through z^N. These are explicit normalizations, not cosmetic choices.

7.6 Row 148: literal compact reciprocal polynomials

A recurrence determines low-order coefficients, but the manuscript displays compact binomial/factorial expressions. This module proves the displayed forms over every characteristic-zero field. For the second coefficient:

Source: KomorebiPaperIV/CompactReciprocal.lean.

```
theorem reciprocalCoeff_two_compact {N : ℕ} (hN : 0 < N) (r : K) :
    reciprocalCoeff N 2 r =
      r * (((2*N).choose N : K) / 2 * (r+1) - 1) := by
  have h := (ordinary_low_equations N r).2.1
  change 2 * PowerSeries.coeff 2 (reciprocalSeries N r) = _ at h
  rw [reciprocalCoeff, blockCoeff, Nat.mul_comm N 2,
    Nat.cast_choose K (show N ≤ 2*N by omega), show 2*N-N=N by omega]
  have hf : ((2*N).factorial : K) ≠ 0 := by exact_mod_cast (2*N).factorial_ne_zero
  have hg : (N.factorial : K) ≠ 0 := by exact_mod_cast N.factorial_ne_zero
  field_simp at h ⊢
  linear_combination h
```

The third coefficient is likewise identified by its literal formula. The proof moves between ordinary coefficients and factorial normalization, justifies the nonzero factorial denominators, and uses the earlier recurrence. This closes the displayed-polynomial clause rather than relying on an informal simplification outside Lean.

7.7 Row 149: the displayed Hessenberg determinant

The determinant's matrix is defined entry by entry. Below the diagonal it contains the recurrence coefficients. Its superdiagonal contains negative integers.

Source: KomorebiPaperIV/NewtonDeterminant.lean.

```
def newtonHessenberg (p : ℕ → K) (n : ℕ) : Matrix (Fin n) (Fin n) K :=
  fun i j => (if j.val ≤ i.val then p (i.val+1-j.val) else 0) -
    (if j.val = i.val+1 then (i.val+1 : K) else 0)
```

The generic determinant theorem identifies the determinant with $n!b_n$. The reciprocal specialization is:

Source: KomorebiPaperIV/NewtonDeterminant.lean.

```
theorem reciprocalOrdCoeff_newton_determinant (N n : ℕ) (r : K) :
    reciprocalOrdCoeff N n r = (n.factorial : K)⁻¹ *
      (newtonHessenberg (fun k => r*kappaCoeff N k) n).det := by
  have hb0 : reciprocalOrdCoeff N 0 r = 1 := by simp [reciprocalOrdCoeff]
  have hb (i : ℕ) : ∑ j ∈ range (i+1), reciprocalOrdCoeff N j r *
      (r*kappaCoeff N (i+1-j)) = (i+1 : K)*reciprocalOrdCoeff N (i+1) r := by
    have h := reciprocalOrdCoeff_newton N r i
    rw [Finset.Nat.sum_antidiagonal_eq_sum_range_succ_mk] at h
    rw [h, mul_sum]
    apply sum_congr rfl
    intro j hj
    have he : i-j+1=i+1-j := by have := mem_range.mp hj; omega
    rw [he]
    ring
  rw [newtonHessenberg_det _ _ hb0 hb]
  have hf : (n.factorial : K) ≠ 0 := by exact_mod_cast n.factorial_ne_zero
  field_simp
```

The finite-dimensional convention includes $n = 0$, whose determinant and initial coefficient are one. The formula uses ordinary coefficients. Obtaining the manuscript's block coefficients requires their separate $(Nn)!$ conversion. The negative superdiagonal and the recurrence index are part of the checked matrix, not inferred from a picture.

7.8 Row 150: hypergeometric normalization and summation

The literal denominator parameters are generated by removing the unique residue parameter equal to one.

Source: KomorebiPaperIV/HypergeometricAnalytic.lean.

```
def residueHypergeometricParameters (N j : ℕ) : Multiset ℂ :=
  ((range N).erase (N-j-1)).val.map (fun a => ((j+a+1 : ℕ) : ℂ)/(N : ℂ))
```

There are $N - 1$ remaining parameters, all with positive real part. The source first relates factorials to residue Pochhammer products. The analytic closure then converts the pinned Mathlib regularized hypergeometric series to the ordinary normalization using Gamma factors and proves its sum.

Source: KomorebiPaperIV/HypergeometricAnalytic.lean.

```
theorem residueHypergeometric_hasSum {N j : ℕ} (hN : 0 < N) (hj : j < N) (w : ℂ) :
    HasSum (fun q => (residueZeroFCoeff N j q : ℂ) * w^q)
      (residueHypergeometric N j w) := by
  let b := residueHypergeometricParameters N j
  have hb : ∀ c ∈ b, ∀ k : ℕ, c ≠ -(k : ℂ) := by
    intro c hc k he
    have hp := residueHypergeometricParameters_re_pos hN hc
    rw [he] at hp
    simp only [Complex.neg_re, Complex.natCast_re] at hp
    have hk : (0 : ℝ) ≤ k := Nat.cast_nonneg k
    linarith
  have hs := (Complex.regularizedHGFunSeries 0 b).hasSum (x := w) (by
    rw [Complex.radius_regularizedHGFunSeries_eq_top (by simp)]
    simp)
  have h := hs.mul_left (b.map Complex.Gamma).prod
  change HasSum _ (residueHypergeometric N j w) at h
  apply h.congr_fun
  intro q
  simp only [Complex.regularizedHGFunSeries, FormalMultilinearSeries.ofScalars_apply_eq,
    smul_eq_mul, ← mul_assoc, regularized_zeroF_normalization b hb]
  rw [← residuePochhammerProduct_erase hN hj]
  simp [residueZeroFCoeff]
```

Finally the entire transfer section is identified at arbitrary complex arguments.

Source: KomorebiPaperIV/HypergeometricAnalytic.lean.

```
theorem entireTransfer_hypergeometric {N j : ℕ} (hN : 0 < N) (hj : j < N)
    {omega : ℂ} (ho : IsPrimitiveRoot omega N) (lambda z : ℂ) :
    entireTransfer N j omega lambda z =
      (lambda*z)^j / (j.factorial : ℂ) *
        residueHypergeometric N j ((lambda*z)^N / (N : ℂ)^N) := by
  let f := fun m => PowerSeries.coeff m (transferSection N j lambda) * z^m
  have hi : Function.Injective (fun q : ℕ => N*q+j) := by
    intro a b h
    exact Nat.eq_of_mul_eq_mul_left hN (Nat.add_right_cancel h)
  have hf : ∀ m, m ∉ Set.range (fun q : ℕ => N*q+j) → f m = 0 := by
    intro m hm
    have he : m%N ≠ j := by
      intro he
      apply hm
      use m/N
      change N*(m/N)+j=m
      rw [← he, Nat.add_comm, Nat.mod_add_div]
    simp [f, coeff_transferSection, transferSectionCoeff, he]
  have ha := (hi.hasSum_iff hf).2 (hasSum_entireTransfer hN hj ho lambda z)
  have hh := (residueHypergeometric_hasSum hN hj ((lambda*z)^N / (N : ℂ)^N)).mul_left
    ((lambda*z)^j / (j.factorial : ℂ))
  apply ha.unique
  apply hh.congr_fun
```

```
  intro q
  dsimp [Function.comp_def, f]
  rw [coeff_transferSection, transferSectionCoeff,
    show (N*q+j)%N = j by simp [Nat.add_mod, Nat.mod_eq_of_lt hj]]
  simp only [one_div, map_invв‚Ђ, map_natCast]
  have hn := congrArg (fun a : в„љ => (a : в„‚)) (residueZeroFCoeff_normalization hN j q)
  push_cast at hn
  rw [в†ђ hn]
  simp only [div_eq_mul_inv, mul_pow, inv_pow, в†ђ pow_mul]
  rw [pow_add, pow_add]
  simp only [ite_true]
  ring
```

The proof reindexes the original Taylor sum by $Nq + j$, proves off-residue terms vanish, and matches the factorial/Pochhammer coefficients. The argument retains the scale $(\lambda z)^N/N^N$ and the prefactor $(\lambda z)^j/j!$. For $N = 1$, the erased parameter multiset is empty and the same theorem applies.

7.9 Row 151: convergent quotient-algebra exponentials

The actual exponential series is defined in a topological algebra.

Source: KomorebiPaperIV/QuotientExponential.lean.

```
def finiteAlgebraExponential {A : Type*} [Ring A] [Algebra в„‚ A]
    [TopologicalSpace A] (x : A) (z : в„‚) : A :=
  в€‘' m : в„•, (z^m / (m.factorial : в„‚)) вЂў x^m
```

Its definition alone does not assert summability. The next theorem supplies convergence for the cyclic relation under the surrounding module's topological linear-space assumptions.

Source: KomorebiPaperIV/QuotientExponential.lean.

```
theorem hasSum_cyclicAlgebra_exponential {N : в„•} (hN : 0 < N)
    {omega : в„‚} (ho : IsPrimitiveRoot omega N) (x : A) (hx : x^N = 1)
    (lambda z : в„‚) :
    HasSum (fun m : в„• => ((lambda*z)^m / (m.factorial : в„‚)) вЂў x^m)
      (в€‘ j : Fin N, entireTransfer N j.val omega lambda z вЂў x^j.val) := by
  have h := hasSum_sum (s := Finset.univ) (fun j (_ : j в€€ (Finset.univ : Finset (Fin N))) =>
    (hasSum_entireTransfer hN j.isLt ho lambda z).smul_const (x^j.val))
  apply h.congr_fun
  intro m
  have hxmod : x^m = x^(m%N) := by
    calc
      x^m = x^(m%N + N*(m/N)) := by rw [Nat.mod_add_div]
      _ = x^(m%N) := by rw [pow_add, pow_mul, hx]; simp
  rw [hxmod, Finset.sum_eq_single (вџЁm%N, Nat.mod_lt m hNвџ© : Fin N)]
  В· simp only [coeff_transferSection, transferSectionCoeff, ite_true]
    congr 1
    simp [mul_pow, div_eq_mul_inv, mul_assoc, mul_left_comm, mul_comm]
  В· intro j _ hj
    have hne : m%N в‰  j.val := by intro he; apply hj; exact Fin.ext he.symm
    simp [coeff_transferSection, transferSectionCoeff, hne]
  В· simp
```

The proof groups powers by residue and uses the scalar `HasSum` theorem for each coordinate. The quotient algebras receive their finite-dimensional coefficient norm topologies and compatible normed-space instances. The signed specialization is:

Source: `KomorebiPaperIV/QuotientExponential.lean`.

```
theorem signedAlgebra_pointwise_exponential {N : ℕ} [NeZero N]
    {omega rho : ℂ} (ho : IsPrimitiveRoot omega N) (hr : rho^N = -1) (z : ℂ) :
    finiteAlgebraExponential (signedCyclicGenerator ℂ N) z =
      ∑ j : Fin N, entireSigned N j.val omega rho z • (signedCyclicGenerator ℂ N)^j.val := by
  exact (hasSum_signedAlgebra_exponential (Nat.pos_of_ne_zero (NeZero.ne N)) ho hr
    (signedCyclicGenerator ℂ N) (signedCyclicGenerator_pow ℂ N) z).tsum_eq
```

This is the final analytic link for result 6.2. The inherited formal power-series identity, the finite algebra relation, and the pointwise convergent exponential all retain the same generator and normalization.

8 Reproduction, audit, and evidence handling

8.1 The exact environment

The source records specify the Lean compiler and Mathlib environment used for the verification.

The accompanying lake-manifest.json records all nine dependency pins.

The project has autoImplicit = false and relaxedAutoImplicit = false. The verified replay establishes warning rejection through the explicit control in Section 9, which distinguishes the inherited setting from the corrected invocation. The aggregate import integrates the development, while the explicit kernel audit checks the mapped declarations and the namespace. Build caches are excluded from the source distribution.

8.2 What the saved kernel transcript establishes

The saved transcript contains one axiom record for each of the 1,164 mapped targets and ends with the namespace audit's pass line. The guide generator compares the target sets in four places: the theorem map, the declaration-kind contract, the audit source's explicit directives, and the transcript. It also checks that every recorded axiom belongs to the permitted set of three axioms.

This reconciliation proves consistency of the preserved evidence files with the preserved source identities. It does not repeat elaboration, type checking, or a live environment traversal. The namespace totals of 2,234 declarations and 1,999 theorem declarations are inherited results, not counts newly obtained from a running Lean process. A lexical source parser cannot substitute for elaborated declaration types or compiler-generated declarations.

8.3 Static reconciliation

The generator validates every one of the 119 Lean source files against the canonical SHA-256 manifest. It resolves the 1,164 target names to source declarations and checks the theorem-versus-construction split against the explicit contract. It emits the complete row catalog, 68-result ledger, direct-import inventory, exact pins, and source-excerpt manifest.

The source policy scan excludes comments and string literals before checking for forbidden proof placeholders and unsafe shortcuts. The kernel-audit file is treated separately because its exclusion logic necessarily names prohibited axioms. This is a source-hygiene check. It does not establish that elaboration succeeds.

The 43 declarations in `SemanticChecks.lean` are supported by their source and preserved execution evidence. Static reconciliation does not relabel them as newly executed tests. The separate verification chapter records the production replay and compiler controls. The 119 mathematical source files remained unchanged.

8.5 Failure modes worth recognizing

An environment mismatch invalidates a replay claim. A transcript missing contract targets is incomplete even when its last line says pass. A declaration that fails to elaborate has not passed a formal run. A successfully built PDF does not itself certify Lean proofs.

> A recorded numerical suite accepted mutations of a projector identity and a zero-counting constant. Its passing totals cannot certify theorems. Finite checks can detect selected regressions, but must retain their tested scope and sensitivity. The formal contract and kernel evidence have separate roles.

8.6 How to inspect one result end to end

Choose a manuscript record in the coverage ledger. Read all of its map rows, including later rows that close analytic clauses. Follow each exact declaration name to its source location. Inspect the surrounding variables, typeclass instances, definitions, and imported lemmas. Check the target's kind in the contract and its axiom record in the saved transcript. If a new formal run is needed, use the pinned environment and retain its complete output separately from inherited evidence.

For example, result 4.5 joins row 27's factorial/Pochhammer formal identity with row 150's literal parameter multiset, Gamma normalization, and actual convergent hypergeometric summation. Reviewing only row 27 would miss completed analytic coverage. Reviewing only the final identity would miss the coefficient normalization on which it depends. The contract is designed to make both routes visible.

9 Pinned-source verification and its scope

The 68-result formal core passed a replay using pinned sources, declaration-kind resolution, a transitive axiom audit, and compiler controls. All 119 Lean source files and their mathematical hypotheses remained unchanged. The 151-row reviewed map covers the full scoped statements of the 68 theorem-like manuscript records. This verification checks the exact inherited statement contract. It does not supply a new line-by-line mathematical proof of that correspondence.

9.1 Production compilation and proof evidence

The replay compiled all 118 production modules into an initially empty directory for project objects. Each module ran separately with one Lean worker. The cumulative receipt fingerprints every successful source and output object. The aggregate imports the completed production closure. The original audit source and a generated declaration-kind consumer were then executed.

Compiler and dependency identities were checked against the manifest. Cached external objects were reused. The record does not claim an independent rebuild of Mathlib or a separately implemented proof kernel.

Evidence	Recorded scope
Production compilation	118 of 118 modules, with new project objects
Source preservation	All 119 Lean source files unchanged
Reviewed targets	1,164 targets: 953 theorems and 211 constructions
Namespace audit	2,234 declarations, including 1,999 theorem declarations
Compiler controls	Six accepted outcomes under explicit diagnostics
Verifier tests	42 verification tests and 10 integrity tests

Supporting and generated declarations contribute to these totals. The totals are not counts of independent discoveries.

```
The source records specify the exact compiler and dependency environment. The permitted
transitive axioms are propext, Classical.choice, and Quot.sound.
```

9.2 Controls and their diagnostics

A valid projector-idempotence application is accepted. An explicit false theorem, an admitted proof, an omitted premise, an injected custom axiom, and a definition presented as a theorem must produce their intended rejection diagnostics. A timeout, signal kill, missing import, or unrelated API error does not count as successful rejection.

The warning policy is material to this contract. The inherited `-E warning` setting accepted an isolated deliberately admitted proof with exit code 0. The verified replay instead uses `-DwarningAsError=true`, which rejected that fixture with exit code 1. The production mathematical sources were not weakened or edited.

The verifier tests use the recovered typed source contract and adversarial mutations of rows, targets, evidence labels, axiom reports, and process outcomes. A descendant-process cleanup fixture waits for explicit child readiness before applying a separate bounded execution deadline. An earlier timeout

before readiness did not establish that the descendant had started and is retained as a failed check. The final 42-test suite passes.

The ten integrity tests cover changed bytes, missing files, unexpected files, duplicate records, unsafe paths, symlinks, prohibited font files, and positive and empty-manifest cases. The distribution manifest describes the actual 836-file preserved tree. It does not require the 73 excluded font files listed in an older manifest. The exact earlier unsaved fixture-control failure was not recovered, so the record makes no claim to reproduce that absent script. The replacement controls directly exercise the recovered checker's actual typed records and have their own fingerprints and receipts.

9.3 Finite computations and their limits

The finite computations were executed separately from Lean. Their domains overlap, so the following rows are not added into one total. They support the implementations and worked examples. Universal statements rely on proof terms under their declared hypotheses.

Suite	Recorded result
Exact algebra	1,008 exact rational assertions
Visual examples	147 symbolic, finite, and enclosure checks
Original tables	981 of 981 rows checked
Strengthened-paper suite	1,246 of 1,246 checks passed
Original-paper suite	647 of 647 checks passed
Rational subset and brackets	818 rational table rows and four bracket records

The previously documented weakness of the strengthened numerical suite remains relevant: its historical acceptance of mutations limits what its passing total can establish. It does not replace the formal contract or the targeted compiler controls.

The source records specify the computation environment. All 836 files remained byte-identical before and after the finite replay. Results and logs were written separately.

9.4 Evidence files and reproduction

The formal verification distribution retains compiler logs, generated consumers, fixtures, source hashes, and project output objects. The external runtime and dependency cache are separate requirements. The principal evidence records are:

- `verification/lean_primary/REPLAY.json` and `verification/lean_primary/AXIOM_REPORTS.json`;
- `verification/SOURCE_MAP_CHECK.json` and `verification/SOURCE_CONTRACT.json`;
- `verification/finite_primary/FINITE_REPLAY.json`;
- `verification/VERIFIER_TESTS.json` and `verification/INTEGRITY_TESTS.json`.

The accompanying source records provide the reproduction commands.

Use a new writable working directory outside the preserved source tree. The first command checks stored evidence and file integrity. It does not rerun the kernel. Finite mode reruns the examples and verifier controls. Full mode additionally compiles every production module and repeats the formal audits. These commands do not download or install dependencies.

The complete replay targets Linux or WSL with Python 3.11 or newer, the pinned Lean runtime, and matching cached packages. The recorded run used a four-GiB container, one Lean worker, a 3,072-MiB Lean heap setting, and a 300-second deadline for each command. This is not a claim of reproducibility on native Windows or across platforms.

9.5 Mathematical boundaries

Imported standard fact 11.1 remains outside the 68-result contract. The core does not assert universal simplicity of all zeros, positivity of every reduced zero, a full displacement expansion, or a zeta functional equation. Eventual simplicity, branch choices, multiplicities, and nearest-pole asymptotics retain their stated hypotheses and ranges. The separately preserved visual reader includes a branch-sensitive clarification of reciprocal powers. No older editable source is substituted for that reader.

10 Complete manuscript coverage ledger

Each entry is a full scoped result in the formal contract. Its coverage is the union of the cited rows, with the original hypotheses retained. This ledger has 68 entries. Definitions and the explicitly imported standard fact 11.1 are not added to that count.

Record	Full scoped result	Additive map rows
3.1	Root filter	1, 2, 4, 143
3.2	Covariance characterization	24, 143
3.3	Cyclic product law	3, 143
4.2	Differential cycle and initial data	7, 47, 52, 53
4.3	Wronskian of the transfer basis	48
4.4	Addition law	25, 26, 144
4.5	Hypergeometric form	27, 150
5.2	Relation with the transfer exponential	9, 47
5.3	Signed derivative cycle	10, 47, 53, 56
5.4	Wronskian of the signed basis	48
5.5	Twisted addition law	28, 144
5.6	Finite exponential representation	11, 47
5.8	Symmetries	29, 60, 64
6.2	Finite algebra exponential identities	31, 151
6.4	Companion matrix realization	33, 49, 51
6.5	Entries of the fundamental matrix	33, 50
6.6	Addition and inversion laws for the companion matrix	34, 50, 54
7.2	Bell polynomial formula	46
7.3	Parameter recurrence	35
7.4	Additive parameter law for reciprocal powers	13, 145
7.5	A sequence of binomial type	14
7.6	Logarithmic Newton recurrence	15, 149
7.7	The first three reciprocal polynomials	36, 148
7.8	(untitled)	37, 46
8.2	Logarithmic coefficient formula	15, 38, 147
8.3	Tangent recurrence	15, 38, 147
8.4	Derivative of the reciprocal family	15, 145
8.5	Scalar higher Riccati identity	39, 145
9.2	Triangular inverse system	16, 17
9.4	Parameter-addition law for residue-section reciprocals	41

Record	Full scoped result	Additive map rows
9.6	Appell laws	18, 42
9.7	Explicit Appell polynomial	18, 42
9.8	Joint Appell convolution in the space and order parameters	19, 42
9.9	Formal differential-operator representation on polynomials	20, 42, 146
9.10	Appell symmetry	21, 42
10.2	Theta reconstruction	22, 43
10.3	Trivial parameter factors	23, 44
10.4	Theta Newton recurrence	45
11.2	Orders and exponential type	54, 110, 111
11.4	Indicator polygon and dominant sectors	112, 113, 114, 115
11.5	Two term dominance	58, 59, 60, 62, 67
11.6	Near cancellation of two exponentials	55
11.7	Large zero strings	57, 60, 61, 63, 64, 65, 66, 71, 81
11.8	Leading displacement term and derivative sign	57, 61, 82, 83
11.9	Leading correction for $N = 3$	84
11.10	The case $N = 2$	55, 70
11.11	Eventual simplicity and exact rays	68
11.13	Reduced zeros	69, 70, 71
11.14	Zero counting asymptotics	73, 75, 76, 77, 78
11.15	Zero exponents	72, 79
12.1	Genus zero product	74, 96, 97, 98, 99, 100, 101, 102
12.2	Diagonal Fredholm determinant representation	116, 117, 118
12.3	Complete symmetric function form	119, 120
12.4	Products for the normalized sections	121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 132
12.6	Newton sums for the reduced zeros	104, 105, 109
12.7	Convergent Mittag–Leffler partial-fraction form	103
12.8	Theta Bell polynomial formula	108
12.9	Tangent coefficients as zero sums	106
12.10	Reciprocal coefficients from zero data	107
13.1	Nearest-pole coefficient expansion	131, 132, 133, 134, 135, 136, 137, 138, 139
13.2	Leading dominant zero term	134, 140
13.3	Orbit reduced nearest circle term	141
13.4	Secant asymptotic	142

Record	Full scoped result	Additive map rows
14.2	Abscissae of absolute convergence	80
14.3	Meromorphic continuation of the symmetry-orbit zero zeta	88, 89, 90, 91, 92
14.4	Exact zeta models in orders two and four	94, 95
14.6	Branch-dependent symmetry-orbit continuation	93
14.8	Derivative-weighted convergence	85, 86, 87

11 Complete declaration catalog

The following 151 entries reproduce the exact row scopes and all 1,164 fully qualified declaration names from the canonical theorem map. File names, lexical line locations, declaration kinds, and manuscript labels are reconciled with the source and the formal contract.

Read scope additively. An earlier row may say that pointwise convergence, a branch law, or another clause is outside *that row*. Such wording is preserved here because it precisely describes that contribution. It is not an unresolved gap in the completed result. Consult the 68-result ledger and rows 143–151 for the joined coverage. The present static reconciliation does not rerun Lean's elaborator or kernel.

11.1 Row 1: 3.1 Root filter, coefficient action

Exact row scope. Exact formal coefficient selector.

Manuscript labels. `thm:root_filter`

`Komorebi.PaperIV.cyclicProjector`
definition; `KomorebiPaperIV/CyclicFilters.lean`, line 25.

`Komorebi.PaperIV.coeff_cyclicProjector`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 29.

`Komorebi.PaperIV.cyclicProjector_eq_zero_of_residue_ne`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 47.

`Komorebi.PaperIV.cyclicProjector_eq_of_mod_eq`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 52.

11.2 Row 2: 3.1 Root filter, projector resolution

Exact row scope. Exact over every commutative semiring; completeness assumes `N > 0`.

Manuscript labels. `thm:root_filter`

`Komorebi.PaperIV.cyclicProjector_idempotent`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 34.

`Komorebi.PaperIV.cyclicProjector_orthogonal`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 39.

`Komorebi.PaperIV.sum_cyclicProjector`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 74.

11.3 Row 3: 3.3 Cyclic product law

Exact row scope. Exact pure-packet label addition, off-label vanishing, and full finite double-sum product formula.

Manuscript labels. `thm:cyclic_product`

`Komorebi.PaperIV.cyclicProjector_add`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 58.

`Komorebi.PaperIV.cyclicProjector_sum`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 63.

`Komorebi.PaperIV.cyclicProjector_product_packet`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 88.

`Komorebi.PaperIV.cyclicProjector_product_other`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 114.

`Komorebi.PaperIV.cyclic_product_law`
theorem; `KomorebiPaperIV/CyclicFilters.lean`, line 122.

11.4 Row 4: 3.1 Root filter, explicit Fourier average

Exact row scope. Exact primitive-root finite Fourier average in `ℂ⟦X⟧`, proved equal to the coefficient projector.

Manuscript labels. `thm:root_filter`

`Komorebi.PaperIV.finiteRootAverage`
definition; `KomorebiPaperIV/FiniteAnalytic.lean`, line 30.

`Komorebi.PaperIV.coeff_finiteRootAverage`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 69.

`Komorebi.PaperIV.finiteRootAverage_eq_cyclicProjector`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 83.

11.5 Row 5: 4.1 Transfer exponential basis

Exact row scope. Exact formal realization of the lacunary exponential series.

Manuscript labels. `def:transfer_exponential`

`Komorebi.PaperIV.exponentialGeneratingSeries`
definition; `KomorebiPaperIV/SectionCycles.lean`, line 26.

`Komorebi.PaperIV.coeff_exponentialGeneratingSeries`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 31.

`Komorebi.PaperIV.transferSectionCoeff`
definition; `KomorebiPaperIV/SectionCycles.lean`, line 68.

`Komorebi.PaperIV.transferSection`
definition; `KomorebiPaperIV/SectionCycles.lean`, line 72.

`Komorebi.PaperIV.coeff_transferSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 76.

11.6 Row 6: 4.1 Transfer exponential, root-average representation

Exact row scope. Exact finite root-of-unity representation in `ℂ⟦X⟧`, including rescaling by the spectral parameter.

Manuscript labels. `def:transfer_exponential`

`Komorebi.PaperIV.finiteRootAverage_exp_eq_transferSection`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 90.

`Komorebi.PaperIV.finiteRootAverage_rescale_exp_eq_transferSection`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 104.

11.7 Row 7: 4.2 Differential cycle and initial data

Exact row scope. Exact formal derivative cycle, full-period equation, and initial data. Analytic solution-space dimension and the Wronskian claim are outside this row.

Manuscript labels. `thm:exponential_cycle`

`Komorebi.PaperIV.derivative_exponentialGeneratingSeries`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 37.

`Komorebi.PaperIV.iterate_derivative_exponentialGeneratingSeries`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 55.

`Komorebi.PaperIV.transferSectionCoeff_succ_of_pos`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 84.

`Komorebi.PaperIV.transferSectionCoeff_succ_zero`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 113.

`Komorebi.PaperIV.derivative_transferSection_of_pos`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 225.

`Komorebi.PaperIV.derivative_transferSection_zero`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 237.

`Komorebi.PaperIV.transferSectionCoeff_add_period`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 269.

`Komorebi.PaperIV.iterate_derivative_transferSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 298.

`Komorebi.PaperIV.constantCoeff_iterate_derivative_transferSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 320.

11.8 Row 8: 5.1 Signed cyclic sections

Exact row scope. Exact formal realization of the signed lacunary series.

Manuscript labels. `def:signed_sections`

`Komorebi.PaperIV.signedSectionCoeff`
definition; `KomorebiPaperIV/SectionCycles.lean`, line 137.

`Komorebi.PaperIV.signedSection`
definition; `KomorebiPaperIV/SectionCycles.lean`, line 141.

`Komorebi.PaperIV.coeff_signedSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 145.

11.9 Row 9: 5.2 Relation with the transfer exponential

Exact row scope. Exact formal identity for a chosen `rho` with `rho ^ N = -1`, including the factor $(\texttt{rho}\ \texttt{\^{}}\ \texttt{j})^{-1}$.

Manuscript labels. `prop:signed_exponential_relation`

`Komorebi.PaperIV.signedSection_eq_inv_pow_mul_transferSection`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 143.

11.10 Row 10: 5.3 Signed derivative cycle

Exact row scope. Exact formal derivative cycle, negative wraparound, full-period equation, and initial data.

Manuscript labels. `thm:signed_cycle`

`Komorebi.PaperIV.signedSectionCoeff_succ_of_pos`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 170.

`Komorebi.PaperIV.signedSectionCoeff_succ_zero`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 200.

`Komorebi.PaperIV.derivative_signedSection_of_pos`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 248.

`Komorebi.PaperIV.derivative_signedSection_zero`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 258.

`Komorebi.PaperIV.signedSectionCoeff_add_period`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 284.

`Komorebi.PaperIV.iterate_derivative_signedSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 310.

`Komorebi.PaperIV.constantCoeff_iterate_derivative_signedSection`
theorem; `KomorebiPaperIV/SectionCycles.lean`, line 337.

11.11 Row 11: 5.6 Finite exponential representation

Exact row scope. Exact finite representation in $\mathbb{C}[\![X]\!]$ for every signed section and the distinguished denominator. Pointwise evaluation is supplied by row 47.

Manuscript labels. `thm:finite_exponential_representation`

`Komorebi.PaperIV.signedSection_eq_finiteRootAverage_rescale_exp`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 153.

`Komorebi.PaperIV.signedSection_zero_eq_finite_exponential_average`
theorem; `KomorebiPaperIV/FiniteAnalytic.lean`, line 166.

11.12 Row 12: 5.7 and 7.1 Reduced denominator and reciprocal family

Exact row scope. Exact named formal specialization `Delta_N(w)^(-r) = exp(-r log Delta_N(w))`, with parameters `-1`, `0`, and `1` linked to the denominator, unit, and inverse. No analytic logarithm branch is asserted.

Manuscript labels. `def:reciprocal_family`, `def:reduced_denominator`

`Komorebi.PaperIV.formalExpFamily`
definition; `KomorebiPaperIV/FormalExponential.lean`, line 27.

`Komorebi.PaperIV.formalExpFamily_zero`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 37.

`Komorebi.PaperIV.formalExpFamily_one`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 45.

`Komorebi.PaperIV.reducedDenominator`

definition; KomorebiPaperIV/ReciprocalCore.lean, line 28.

Komorebi.PaperIV.coeff_reducedDenominator
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 33.

Komorebi.PaperIV.constantCoeff_reducedDenominator
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 39.

Komorebi.PaperIV.signedSection_zero_eq_expand_reducedDenominator
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 46.

Komorebi.PaperIV.negativeLogDenominator
definition; KomorebiPaperIV/ReciprocalCore.lean, line 63.

Komorebi.PaperIV.constantCoeff_negativeLogDenominator
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 67.

Komorebi.PaperIV.exp_subst_logOf
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 74.

Komorebi.PaperIV.mul_derivative_logOf
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 88.

Komorebi.PaperIV.formalExpFamily_eq_exp_subst
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 101.

Komorebi.PaperIV.reciprocalSeries
definition; KomorebiPaperIV/ReciprocalCore.lean, line 112.

Komorebi.PaperIV.constantCoeff_reciprocalSeries
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 116.

Komorebi.PaperIV.reciprocalSeries_zero
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 128.

Komorebi.PaperIV.reciprocalSeries_neg_one
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 133.

Komorebi.PaperIV.reciprocalSeries_one_eq_inverse
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 151.

Komorebi.PaperIV.reciprocalCoeff
definition; KomorebiPaperIV/ReciprocalCore.lean, line 161.

Komorebi.PaperIV.reciprocalOrdCoeff
definition; KomorebiPaperIV/ReciprocalCore.lean, line 165.

Komorebi.PaperIV.reciprocalCoeff_zero_parameter
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 169.

Komorebi.PaperIV.reciprocalCoeff_zero_parameter_of_pos
theorem; KomorebiPaperIV/ReciprocalCore.lean, line 173.

11.13 Row 13: 7.4 Additive parameter law for reciprocal powers

Exact row scope. Exact formal parameter law and exact `(Nn choose Nq)` convolution for the named `P_{N,n}` specialization.

Manuscript labels. `thm:reciprocal_parameter_addition`

`Komorebi.PaperIV.formalExpFamily_semigroup`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 30.

`Komorebi.PaperIV.blockCoeff`
definition; `KomorebiPaperIV/FormalExponential.lean`, line 129.

`Komorebi.PaperIV.blockCoeff_mul`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 134.

`Komorebi.PaperIV.reciprocal_block_convolution`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 163.

`Komorebi.PaperIV.reciprocalSeries_add`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 142.

`Komorebi.PaperIV.reciprocalCoeff_addition`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 178.

11.14 Row 14: 7.5 Binomial-type normalization

Exact row scope. Exact exponential-generating normalization and binomial convolution.

Manuscript labels. `cor:binomial_type`

`Komorebi.PaperIV.egfCoeff`
definition; `KomorebiPaperIV/FormalExponential.lean`, line 101.

`Komorebi.PaperIV.egfCoeff_mul`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 106.

`Komorebi.PaperIV.reciprocal_binomial_type`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 174.

11.15 Row 15: 7.6, 8.2, 8.3, and 8.4 Newton, tangent, and reciprocal derivative identities

Exact row scope. Exact formal logarithmic derivative and finite coefficient recurrences in the reduced variable. The displayed z-variable tangent scaling is not separately named.

Manuscript labels. `cor:reciprocal_derivative`, `prop:tangent_kappa`, `thm:logarithmic_newton_recurrence`, `thm:tangent_recurrence`

`Komorebi.PaperIV.derivative_rescale_exp`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 50.

`Komorebi.PaperIV.derivative_formalExpFamily`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 75.

`Komorebi.PaperIV.formalExpFamily_coefficient_recurrence`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 84.

`Komorebi.PaperIV.kappaCoeff`
definition; `KomorebiPaperIV/ReciprocalCore.lean`, line 191.

`Komorebi.PaperIV.reducedTangentSeries`
definition; `KomorebiPaperIV/ReciprocalCore.lean`, line 195.

`Komorebi.PaperIV.coeff_reducedTangentSeries`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 199.

`Komorebi.PaperIV.reducedDenominator_mul_reducedTangentSeries`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 207.

`Komorebi.PaperIV.reducedTangent_coefficient_convolution`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 217.

`Komorebi.PaperIV.derivative_reciprocalSeries`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 228.

`Komorebi.PaperIV.reciprocalSeries_coefficient_recurrence`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 235.

`Komorebi.PaperIV.reciprocalOrdCoeff_newton`
theorem; `KomorebiPaperIV/ReciprocalCore.lean`, line 245.

11.16 Row 16: 9.2 Generic inverse support

Exact row scope. Exact inverse convolution for every unit-constant series over a field.

Manuscript labels. `thm:triangular_inverse`

`Komorebi.PaperIV.inverse_coefficient_convolution`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 188.

`Komorebi.PaperIV.constantCoeff_inverse_of_one`
theorem; `KomorebiPaperIV/FormalExponential.lean`, line 198.

11.17 Row 17: 9.1 and 9.2 Normalized residue section and triangular inverse

Exact row scope. Exact manuscript normalization, factorial-rescaled triangular identity, and solved recurrence over a characteristic-zero field.

Manuscript labels. `def:normalized_section`, `thm:triangular_inverse`

`Komorebi.PaperIV.normalizedResidueSection`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 35.

`Komorebi.PaperIV.coeff_normalizedResidueSection`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 41.

`Komorebi.PaperIV.constantCoeff_normalizedResidueSection`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 47.

`Komorebi.PaperIV.residueInverseCoeff`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 56.

`Komorebi.PaperIV.residueInverseCoeff_triangular`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 61.

`Komorebi.PaperIV.residueInverseCoeff_zero`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 126.

`Komorebi.PaperIV.residueInverseCoeff_recurrence`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 131.

11.18 Row 18: 9.5, 9.6, and 9.7 Appell realization and exact sparse polynomial

Exact row scope. Exact finite Appell realization, lowering and translation laws, and the manuscript's sparse sum with only degrees `Nn`. The residue specialization here is `r = 1`.

Manuscript labels. `def:block_appell`, `prop:explicit_appell`, `thm:appell_laws`

`Komorebi.PaperIV.appellSeries`
definition; `KomorebiPaperIV/Appell.lean`, line 26.

`Komorebi.PaperIV.appellCoeff`
definition; `KomorebiPaperIV/Appell.lean`, line 30.

`Komorebi.PaperIV.egfCoeff_rescale_exp`
theorem; `KomorebiPaperIV/Appell.lean`, line 58.

`Komorebi.PaperIV.appellCoeff_zero_parameter`
theorem; `KomorebiPaperIV/Appell.lean`, line 73.

`Komorebi.PaperIV.appell_addition`
theorem; `KomorebiPaperIV/Appell.lean`, line 80.

`Komorebi.PaperIV.appellCoeff_at_zero`
theorem; `KomorebiPaperIV/Appell.lean`, line 89.

`Komorebi.PaperIV.appellPolynomial`
definition; `KomorebiPaperIV/Appell.lean`, line 97.

`Komorebi.PaperIV.derivative_appellPolynomial`
theorem; `KomorebiPaperIV/Appell.lean`, line 103.

`Komorebi.PaperIV.formalAppellPolynomial`
definition; `KomorebiPaperIV/Appell.lean`, line 133.

`Komorebi.PaperIV.derivative_formalAppellPolynomial`
theorem; `KomorebiPaperIV/Appell.lean`, line 137.

`Komorebi.PaperIV.eval_formalAppellPolynomial`
theorem; `KomorebiPaperIV/Appell.lean`, line 142.

`Komorebi.PaperIV.blockCoefficientSequence`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 155.

`Komorebi.PaperIV.blockCoefficientSequence_mul`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 159.

`Komorebi.PaperIV.appellPolynomial_eq_sparse_sum`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 166.

`Komorebi.PaperIV.appellPolynomial_eq_block_sum`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 185.

`Komorebi.PaperIV.residueAppellPolynomial_explicit`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 282.

11.19 Row 19: 9.8 Joint Appell convolution

Exact row scope. Exact formal product and coefficient convolution in space and order parameters for the generic zero-constant logarithm family.

Manuscript labels. `thm:appell_two_parameter_convolution`

`Komorebi.PaperIV.appellSeries_two_parameter`
theorem; `KomorebiPaperIV/Appell.lean`, line 34.

`Komorebi.PaperIV.appell_two_parameter_convolution`
theorem; `KomorebiPaperIV/Appell.lean`, line 48.

11.20 Row 20: 9.9 Formal differential-operator representation

Exact row scope. Exact finite action on monomials and exact operator inversion on every polynomial of bounded degree; the direct residue inverse corollary is the `r = 1` case.

Manuscript labels. `prop:appell_operator_form`

`Komorebi.PaperIV.egfConstantCoeffAction`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 325.

`Komorebi.PaperIV.egfConstantCoeffAction_X_pow`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 331.

`Komorebi.PaperIV.egfConstantCoeffAction_residue_X_pow`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 345.

`Komorebi.PaperIV.polynomialDifferentialEval`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 363.

`Komorebi.PaperIV.polynomialDifferentialEval_X`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 370.

`Komorebi.PaperIV.truncatedDifferentialOperator`
definition; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 375.

`Komorebi.PaperIV.coeff_trunc_mul_trunc_of_lt`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 379.

`Komorebi.PaperIV.X_pow_dvd_trunc_mul_trunc_sub_one`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 392.

`Komorebi.PaperIV.truncatedDifferentialOperator_mul_apply_of_natDegree_le`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 405.

`Komorebi.PaperIV.truncatedDifferentialOperator_normalizedResidueSection_inverse`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 439.

11.21 Row 21: 9.10 Appell monicity and covariance

Exact row scope. Exact monicity, exact degree, and `omega ^ m` covariance for block-supported Appell polynomials; residue specialization is `r = 1`.

Manuscript labels. `cor:appell_symmetry`

`Komorebi.PaperIV.appellPolynomial_isMonicOfDegree`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 218.

`Komorebi.PaperIV.blockAppellPolynomial_isMonicOfDegree`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 239.

`Komorebi.PaperIV.eval_appellPolynomial_block_covariance`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 248.

`Komorebi.PaperIV.residueAppellPolynomial_isMonicOfDegree`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 293.

`Komorebi.PaperIV.eval_residueAppellPolynomial_covariance`
theorem; `KomorebiPaperIV/AlgebraicCompletion.lean`, line 303.

11.22 Row 22: 10.1 and 10.2 Theta definition and reconstruction

Exact row scope. Exact formal theta series in the reduced variable and exact signed reconstruction from the named reciprocal coefficients.

Manuscript labels. `def:unsigned_denominator`, `thm:theta_reconstruction`

`Komorebi.PaperIV.blockCoeff_rescale`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 26.

`Komorebi.PaperIV.thetaSeries`
definition; `KomorebiPaperIV/ThetaCore.lean`, line 37.

`Komorebi.PaperIV.thetaCoeff`
definition; `KomorebiPaperIV/ThetaCore.lean`, line 42.

`Komorebi.PaperIV.thetaCoeff_reconstruction_unfactored`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 63.

`Komorebi.PaperIV.thetaCoeff_reconstruction`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 77.

11.23 Row 23: 10.3 Theta trivial parameter factors

Exact row scope. Exact vanishing at `u = 0`, and at `u = -1` for even `N`; these are the root statements underlying the polynomial divisibility claim.

Manuscript labels. `prop:theta_factors`

`Komorebi.PaperIV.thetaSeries_zero`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 46.

`Komorebi.PaperIV.thetaCoeff_zero_parameter`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 51.

`Komorebi.PaperIV.thetaCoeff_zero_parameter_of_pos`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 56.

`Komorebi.PaperIV.thetaSeries_neg_one_of_even`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 106.

`Komorebi.PaperIV.thetaCoeff_neg_one_of_even_of_pos`
theorem; `KomorebiPaperIV/ThetaCore.lean`, line 112.

11.24 Row 24: 3.2 Covariance characterization

Exact row scope. Exact formal fixed-space and range/covariance characterizations for a primitive root. Holomorphic-domain and pointwise versions remain outside this row.

Manuscript labels. `prop:covariance`

`Komorebi.PaperIV.cyclicProjector_fixed_iff_rescale_covariant`
theorem; `KomorebiPaperIV/EarlyCompletion.lean`, line 35.

`Komorebi.PaperIV.cyclicProjector_mem_range_iff_rescale_covariant`
theorem; `KomorebiPaperIV/EarlyCompletion.lean`, line 74.

11.25 Row 25: 4.4 Bivariate formal-series setup

Exact row scope. Exact coefficient encodings for substitution into a sum of two formal variables and for separated products. A bridge to a canonical multivariate substitution API is not asserted.

Manuscript labels. `cor:exponential_addition`

`Komorebi.PaperIV.bivariateExponentialGeneratingSeries`
definition; `KomorebiPaperIV/EarlyCompletion.lean`, line 90.

`Komorebi.PaperIV.coeff_bivariateExponentialGeneratingSeries`
theorem; `KomorebiPaperIV/EarlyCompletion.lean`, line 97.

`Komorebi.PaperIV.separatedExponentialGeneratingSeries`
definition; `KomorebiPaperIV/EarlyCompletion.lean`, line 105.

`Komorebi.PaperIV.coeff_separatedExponentialGeneratingSeries`
theorem; `KomorebiPaperIV/EarlyCompletion.lean`, line 112.

11.26 Row 26: 4.4 Addition law

Exact row scope. Exact coefficient-encoded bivariate formal addition law. Pointwise analytic evaluation is outside this row.

Manuscript labels. `cor:exponential_addition`

`Komorebi.PaperIV.transferSection_addition`
theorem; `KomorebiPaperIV/EarlyCompletion.lean`, line 144.

11.27 Row 27: 4.5 Hypergeometric form

Exact row scope. Exact rational factorial/Pochhammer normalization and formal-power-series identity. The parameter is rational; the explicit erased parameter multiset, complex analytic summation, and Mathlib regularized-hypergeometric bridge remain outside this row.

Manuscript labels. `thm:hypergeometric_form`

`Komorebi.PaperIV.residuePochhammerProduct`
definition; `KomorebiPaperIV/HypergeometricCompletion.lean`, line 36.

`Komorebi.PaperIV.factorial_eq_residuePochhammerProduct`
theorem; `KomorebiPaperIV/HypergeometricCompletion.lean`, line 61.

`Komorebi.PaperIV.residueZeroFCoeff`

definition; KomorebiPaperIV/HypergeometricCompletion.lean, line 97.

Komorebi.PaperIV.residueZeroFSeries
definition; KomorebiPaperIV/HypergeometricCompletion.lean, line 101.

Komorebi.PaperIV.coeff_residueZeroFSeries
theorem; KomorebiPaperIV/HypergeometricCompletion.lean, line 105.

Komorebi.PaperIV.residueZeroFCoeff_normalization
theorem; KomorebiPaperIV/HypergeometricCompletion.lean, line 113.

Komorebi.PaperIV.residueParameter_eq_one
theorem; KomorebiPaperIV/HypergeometricCompletion.lean, line 123.

Komorebi.PaperIV.omitted_residuePochhammer_eq_factorial
theorem; KomorebiPaperIV/HypergeometricCompletion.lean, line 130.

Komorebi.PaperIV.transferSection_eq_residueZeroFSeries
theorem; KomorebiPaperIV/HypergeometricCompletion.lean, line 145.

11.28 Row 28: 5.5 Twisted addition law

Exact row scope. Exact coefficient-encoded bivariate formal law, including the carry sign `(i+j)/N`. Canonical substitution and pointwise evaluation remain outside this row.

Manuscript labels. `prop:signed_addition`

Komorebi.PaperIV.signedSection_addition
theorem; KomorebiPaperIV/EarlyCompletion.lean, line 217.

11.29 Row 29: 5.8 Denominator symmetries

Exact row scope. Exact formal rotational invariance and conjugation-fixed coefficients. Pointwise conjugation, zero orbits, and multiplicity preservation remain outside this row.

Manuscript labels. `cor:denominator_symmetries`

Komorebi.PaperIV.signedDenominator_rescale_of_pow_eq_one
theorem; KomorebiPaperIV/EarlyCompletion.lean, line 247.

Komorebi.PaperIV.signedDenominator_map_conj
theorem; KomorebiPaperIV/EarlyCompletion.lean, line 263.

11.30 Row 30: 6.1 Finite cyclic algebras

Exact row scope. Exact `AdjoinRoot` quotient algebras, generator relations, power reductions, and `N`-dimensional power bases for `N > 0`.

Manuscript labels. `def:cyclic_algebras`

Komorebi.PaperIV.cyclicAlgebraPolynomial
definition; KomorebiPaperIV/FiniteModels.lean, line 33.

Komorebi.PaperIV.signedCyclicAlgebraPolynomial
definition; KomorebiPaperIV/FiniteModels.lean, line 37.

Komorebi.PaperIV.CyclicAlgebra
definition; KomorebiPaperIV/FiniteModels.lean, line 41.

`Komorebi.PaperIV.SignedCyclicAlgebra`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 45.

`Komorebi.PaperIV.cyclicGenerator`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 49.

`Komorebi.PaperIV.signedCyclicGenerator`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 53.

`Komorebi.PaperIV.cyclicGenerator_pow`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 57.

`Komorebi.PaperIV.signedCyclicGenerator_pow`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 65.

`Komorebi.PaperIV.cyclicGenerator_pow_eq_mod`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 73.

`Komorebi.PaperIV.signedCyclicGenerator_pow_eq_mod`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 86.

`Komorebi.PaperIV.cyclicAlgebraPowerBasis`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 104.

`Komorebi.PaperIV.signedCyclicAlgebraPowerBasis`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 112.

`Komorebi.PaperIV.cyclicAlgebraPowerBasis_dim`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 119.

`Komorebi.PaperIV.signedCyclicAlgebraPowerBasis_dim`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 127.

11.31 Row 31: 6.2 Finite algebra exponential identities

Exact row scope. Exact identities of formal power series in the quotient algebras. Pointwise quotient-algebra convergence/evaluation and the packaged derivation of prior differential laws are outside this row.

Manuscript labels. `thm:finite_algebra_exponentials`

`Komorebi.PaperIV.cyclicAlgebraExponentialSeries`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 138.

`Komorebi.PaperIV.signedCyclicAlgebraExponentialSeries`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 145.

`Komorebi.PaperIV.cyclicAlgebra_exponential_eq_sections`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 152.

`Komorebi.PaperIV.signedCyclicAlgebra_exponential_eq_sections`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 178.

11.32 Row 32: 6.3 Signed shift matrix

Exact row scope. Exact closed signed-shift construction and additive power semigroup for `N > 0`.

Manuscript labels. `def:signed_shift_matrix`

`Komorebi.PaperIV.signedShiftPowerMatrix`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 213.

`Komorebi.PaperIV.signedShiftMatrix`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 221.

`Komorebi.PaperIV.signedShiftPowerMatrix_apply`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 225.

`Komorebi.PaperIV.signedShiftPowerMatrix_zero`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 232.

`Komorebi.PaperIV.signedShiftPowerMatrix_mul`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 258.

`Komorebi.PaperIV.signedShiftMatrix_pow`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 297.

11.33 Row 33: 6.4 and 6.5 Companion realization and path entries

Exact row scope. Exact formula for every entry of every matrix power and exact `J_N^N = -I`, plus the matrix-exponential construction. The exponential-section entry bridge, first-column and trace formulas, characteristic polynomial, and simple spectrum remain outside this row.

Manuscript labels. `cor:fundamental_matrix_entries`, `thm:companion_matrix_realization`

`Komorebi.PaperIV.signedShiftMatrix_pow_apply`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 305.

`Komorebi.PaperIV.signedShiftMatrix_pow_order`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 328.

`Komorebi.PaperIV.signedFundamentalMatrix`
definition; `KomorebiPaperIV/FiniteModels.lean`, line 337.

11.34 Row 34: 6.6 Companion addition and inversion

Exact row scope. Exact matrix-exponential addition, inverse, cancellation, and unit laws. The determinant-one and signed-section entry clauses remain outside this row.

Manuscript labels. `cor:companion_addition_inversion`

`Komorebi.PaperIV.signedFundamentalMatrix_add`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 341.

`Komorebi.PaperIV.signedFundamentalMatrix_neg`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 349.

`Komorebi.PaperIV.signedFundamentalMatrix_mul_neg`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 355.

`Komorebi.PaperIV.signedFundamentalMatrix_isUnit`
theorem; `KomorebiPaperIV/FiniteModels.lean`, line 361.

11.35 Row 35: 7.3 Parameter recurrence

Exact row scope. Exact formal differential equation, factorial/binomial coefficient equation, solved recurrence, and initial value; the second sum uses the explicit reindexing `k=q+1`.

Manuscript labels. `thm:parameter_recurrence`

`Komorebi.PaperIV.reducedDenominator_reciprocalSeries_differentialEquation`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 32.

`Komorebi.PaperIV.reciprocalOrdCoeff_parameter_equation`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 58.

`Komorebi.PaperIV.reciprocalCoeff_parameter_equation`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 198.

`Komorebi.PaperIV.reciprocalCoeff_parameter_recurrence`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 317.

`Komorebi.PaperIV.reciprocalCoeff_block_zero`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 346.

11.36 Row 36: 7.7 First three reciprocal polynomials

Exact row scope. Exact `P_(N,1)=r` and exact recurrence-normal forms for `P_(N,2)` and `P_(N,3)`. The manuscript's compact factorial rewrites are not separately proved.

Manuscript labels. `cor:first_reciprocal_polynomials`

`Komorebi.PaperIV.reciprocalCoeff_one`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 351.

`Komorebi.PaperIV.reciprocalCoeff_two_recurrence_form`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 360.

`Komorebi.PaperIV.reciprocalCoeff_three_recurrence_form`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 376.

11.37 Row 37: 7.8 Divisibility by the parameter

Exact row scope. Exact divisibility by `X` for the universal polynomial over $\mathbb{Q}$, plus a separately defined generic-field polynomial with exact scalar evaluation. A named base-change theorem connecting those two polynomial lifts, or generic-field divisibility itself, remains outside this row.

Manuscript labels. `cor:factor_r`

`Komorebi.PaperIV.reciprocalCoeffPolynomial`
definition; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 397.

`Komorebi.PaperIV.X_dvd_reciprocalCoeffPolynomial`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 402.

`Komorebi.PaperIV.coeff_zero_reciprocalCoeffPolynomial`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 430.

`Komorebi.PaperIV.reciprocalCoeffPolynomialOver`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 718.

`Komorebi.PaperIV.eval_reciprocalCoeffPolynomialOver`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 721.

11.38 Row 38: 8.1, 8.2, and 8.3 Transfer tangent and recurrence

Exact row scope. Exact formal quotient definition and exact factorial/binomial recurrence for the kappa-scaled sequence. The bridge identifying these named coefficients with the full-variable tangent series remains outside this row.

Manuscript labels. `def:transfer_tangent`, `prop:tangent_kappa`, `thm:tangent_recurrence`

`Komorebi.PaperIV.formalTransferTangent`
definition; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 579.

`Komorebi.PaperIV.transferTangentCoeff`
definition; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 444.

`Komorebi.PaperIV.transferTangentCoeff_eq_kappa`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 448.

`Komorebi.PaperIV.transferTangentCoeff_convolution`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 510.

`Komorebi.PaperIV.transferTangentCoeff_recurrence`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 538.

`Komorebi.PaperIV.transferTangentCoeff_zero`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 566.

11.39 Row 39: 8.5 Scalar higher Riccati identity

Exact row scope. Exact nontrivial formal-germ differential-Bell/Riccati identity. Pointwise analytic validity on the entire zero-free set remains outside this row.

Manuscript labels. `thm:higher_riccati`

`Komorebi.PaperIV.differentialBellSeries`
definition; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 586.

`Komorebi.PaperIV.differentialBellSeries_zero`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 594.

`Komorebi.PaperIV.differentialBellSeries_succ`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 599.

`Komorebi.PaperIV.iterate_derivative_eq_differentialBellSeries_mul`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 607.

`Komorebi.PaperIV.derivative_signedSection_zero_eq_neg_formalTransferTangent_mul`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 621.

`Komorebi.PaperIV.differentialBellSeries_neg_formalTransferTangent_eq_neg_one`
theorem; `KomorebiPaperIV/ReciprocalCompletion.lean`, line 640.

11.40 Row 40: 9.3 Arbitrary-order residue-section powers

Exact row scope. Exact branch-free formal family `Phi_(N,j)^(-r)`, its factorially normalized coefficients, and the `r=1` bridge to the triangular inverse. No analytic logarithm branch is asserted.

Manuscript labels. `def:residue_section_power`

`Komorebi.PaperIV.normalizedResidueSectionAlg`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 33.

`Komorebi.PaperIV.coeff_normalizedResidueSectionAlg`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 38.

`Komorebi.PaperIV.constantCoeff_normalizedResidueSectionAlg`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 45.

`Komorebi.PaperIV.normalizedResidueSectionAlg_eq_normalizedResidueSection`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 53.

`Komorebi.PaperIV.normalizedResidueNegativeLog`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 68.

`Komorebi.PaperIV.constantCoeff_normalizedResidueNegativeLog`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 72.

`Komorebi.PaperIV.residuePowerSeries`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 141.

`Komorebi.PaperIV.constantCoeff_residuePowerSeries`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 145.

`Komorebi.PaperIV.residuePowerSeries_zero`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 157.

`Komorebi.PaperIV.residuePowerSeries_neg_one`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 161.

`Komorebi.PaperIV.residuePowerSeries_one_eq_inverse`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 191.

`Komorebi.PaperIV.residuePowerCoeff`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 202.

`Komorebi.PaperIV.residuePowerCoeff_zero`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 206.

`Komorebi.PaperIV.residuePowerCoeff_one_eq_residueInverseCoeff`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 218.

11.41 Row 41: 9.4 Residue parameter law, polynomiality, and leading term

Exact row scope. Exact parameter addition, scalar evaluation, leading coefficient, and exact degree of the honest polynomial lift. Analytic complex powers remain outside this row.

Manuscript labels. `thm:residue_section_parameter_addition`

`Komorebi.PaperIV.residuePowerSeries_add`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 172.

`Komorebi.PaperIV.residuePowerCoeff_addition`

theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 210.

Komorebi.PaperIV.residuePowerCoeffPolynomial
definition; KomorebiPaperIV/ParameterThetaCompletion.lean, line 232.

Komorebi.PaperIV.eval_residuePowerCoeffPolynomial
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 237.

Komorebi.PaperIV.coeff_residuePowerCoeffPolynomial_parameterDegree
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 463.

Komorebi.PaperIV.natDegree_residuePowerCoeffPolynomial
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 503.

11.42 Row 42: 9.6, 9.7, 9.8, 9.9, and 9.10 Arbitrary-order residue Appell system

Exact row scope. Exact arbitrary-order formal Appell specialization: sparse formula, lowering, translation, joint convolution, monicity, covariance, monomial symbol action, and opposite-order finite operator inversion. The two operator encodings are not yet bridged into the manuscript's single displayed named inverse relation.

Manuscript labels. cor:appell_symmetry, prop:appell_operator_form, prop:explicit_appell, thm:appell_laws, thm:appell_two_parameter_convolution

Komorebi.PaperIV.expandedResidueNegativeLog
definition; KomorebiPaperIV/ParameterThetaCompletion.lean, line 528.

Komorebi.PaperIV.constantCoeff_expandedResidueNegativeLog
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 533.

Komorebi.PaperIV.expand_residuePowerSeries
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 539.

Komorebi.PaperIV.egfCoeff_expand_residuePowerSeries
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 552.

Komorebi.PaperIV.residueAppellPolynomialParam
definition; KomorebiPaperIV/ParameterThetaCompletion.lean, line 564.

Komorebi.PaperIV.residueAppellPolynomialParam_eq_block
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 568.

Komorebi.PaperIV.residueAppellPolynomialParam_explicit
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 581.

Komorebi.PaperIV.derivative_residueAppellPolynomialParam
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 592.

Komorebi.PaperIV.eval_residueAppellPolynomialParam_add
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 601.

Komorebi.PaperIV.eval_residueAppellPolynomialParam_two_parameter
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 613.

Komorebi.PaperIV.residueAppellPolynomialParam_isMonicOfDegree
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 627.

Komorebi.PaperIV.eval_residueAppellPolynomialParam_covariance
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 636.

`Komorebi.PaperIV.egfConstantCoeffAction_residuePowerSeries_X_pow`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 645.

`Komorebi.PaperIV.truncatedDifferentialOperator_residuePowerSeries_inverse`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 660.

11.43 Row 43: 10.2 Theta polynomial and degree

Exact row scope. Exact polynomial lift and scalar evaluation with degree at most `N*n`, completing the formal polynomial clause of theta reconstruction.

Manuscript labels. `thm:theta_reconstruction`

`Komorebi.PaperIV.thetaPolynomial`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 714.

`Komorebi.PaperIV.eval_thetaPolynomial`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 763.

`Komorebi.PaperIV.natDegree_thetaPolynomial_le`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 779.

11.44 Row 44: 10.3 Theta parameter factors

Exact row scope. Exact literal polynomial divisibility by `X`, and by `X+1` when `N` is even, for positive coefficient index.

Manuscript labels. `prop:theta_factors`

`Komorebi.PaperIV.X_dvd_thetaPolynomial`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 831.

`Komorebi.PaperIV.X_add_one_dvd_thetaPolynomial_of_even`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 839.

11.45 Row 45: 10.4 Theta Newton recurrence

Exact row scope. Exact formal logarithmic derivative, Newton recurrence in equivalent antidiagonal indexing, initial value, and `Theta_(N,1)=-(u^N+u)`.

Manuscript labels. `prop:theta_newton_recurrence`

`Komorebi.PaperIV.derivative_rescale_series`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 859.

`Komorebi.PaperIV.rescale_C_series`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 870.

`Komorebi.PaperIV.scaledReciprocalLogDerivative`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 877.

`Komorebi.PaperIV.derivative_rescale_reciprocalSeries`
theorem; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 882.

`Komorebi.PaperIV.thetaLogDerivativeSeries`
definition; `KomorebiPaperIV/ParameterThetaCompletion.lean`, line 895.

`Komorebi.PaperIV.derivative_thetaSeries`

theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 899.

Komorebi.PaperIV.coeff_thetaLogDerivativeSeries
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 911.

Komorebi.PaperIV.thetaOrdCoeff
definition; KomorebiPaperIV/ParameterThetaCompletion.lean, line 925.

Komorebi.PaperIV.thetaOrdCoeff_zero
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 929.

Komorebi.PaperIV.thetaOrdCoeff_newton
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 938.

Komorebi.PaperIV.factorial_mul_kappaCoeff_one
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 952.

Komorebi.PaperIV.thetaCoeff_one
theorem; KomorebiPaperIV/ParameterThetaCompletion.lean, line 979.

11.46 Row 46: 7.2 Bell formula and 7.8 parameter divisibility

Exact row scope. Exact standard Bell generating-function specialization, every reciprocal coefficient, exact polynomial degree and leading coefficient over any characteristic-zero field, and generic-field parameter divisibility. This closes the full formal statement of 7.2; it asserts no global logarithm branch.

Manuscript labels. `cor:factor_r`, `thm:reciprocal_bell`

Komorebi.PaperIV.bellInput
definition; KomorebiPaperIV/BellCompletion.lean, line 21.

Komorebi.PaperIV.completeExponentialBell
definition; KomorebiPaperIV/BellCompletion.lean, line 25.

Komorebi.PaperIV.constantCoeff_bellInput
theorem; KomorebiPaperIV/BellCompletion.lean, line 30.

Komorebi.PaperIV.completeExponentialBell_generating_coefficient
theorem; KomorebiPaperIV/BellCompletion.lean, line 35.

Komorebi.PaperIV.completeExponentialBell_zero
theorem; KomorebiPaperIV/BellCompletion.lean, line 41.

Komorebi.PaperIV.bellInput_reciprocal
theorem; KomorebiPaperIV/BellCompletion.lean, line 50.

Komorebi.PaperIV.reciprocalCoeff_bell_formula
theorem; KomorebiPaperIV/BellCompletion.lean, line 66.

Komorebi.PaperIV.coeff_one_negativeLogDenominator
theorem; KomorebiPaperIV/BellCompletion.lean, line 76.

Komorebi.PaperIV.coeff_reciprocalCoeffPolynomialOver_parameterDegree
theorem; KomorebiPaperIV/BellCompletion.lean, line 86.

Komorebi.PaperIV.natDegree_reciprocalCoeffPolynomialOver
theorem; KomorebiPaperIV/BellCompletion.lean, line 111.

Komorebi.PaperIV.X_dvd_reciprocalCoeffPolynomialOver
theorem; KomorebiPaperIV/BellCompletion.lean, line 124.

11.47 Row 47: 4.1, 4.2, 5.1, 5.2, 5.3, and 5.6 Entire realization

Exact row scope. Actual entire functions, convergence of the existing formal transfer/signed series at every complex argument, independence of root choices, pointwise finite exponential formulas, iterated derivative expressions, full-period ODEs, and exact initial data. Existence of a basis of the full ODE solution space and the one-step section-to-section derivative identities are not asserted by this row.

Manuscript labels. `def:signed_sections`, `def:transfer_exponential`, `prop:signed_exponential_relation`, `thm:exponential_cycle`, `thm:finite_exponential_representation`, `thm:signed_cycle`

`Komorebi.PaperIV.exponentialSumJet`
definition; `KomorebiPaperIV/EntireBridge.lean`, line 21.

`Komorebi.PaperIV.analyticAt_exponentialSumJet`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 24.

`Komorebi.PaperIV.hasDerivAt_exponentialSumJet`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 29.

`Komorebi.PaperIV.iteratedDeriv_exponentialSum`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 43.

`Komorebi.PaperIV.exponentialSum_ode`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 53.

`Komorebi.PaperIV.entireTransfer`
definition; `KomorebiPaperIV/EntireBridge.lean`, line 66.

`Komorebi.PaperIV.analyticAt_entireTransfer`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 71.

`Komorebi.PaperIV.entireTransfer_eq_exponentialSum`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 76.

`Komorebi.PaperIV.hasSum_finiteRootAverage`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 85.

`Komorebi.PaperIV.hasSum_entireTransfer`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 106.

`Komorebi.PaperIV.entireTransfer_eq_tsum`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 113.

`Komorebi.PaperIV.entireTransfer_independent_root`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 120.

`Komorebi.PaperIV.entireTransfer_ode`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 127.

`Komorebi.PaperIV.entireSigned`
definition; `KomorebiPaperIV/EntireBridge.lean`, line 138.

`Komorebi.PaperIV.analyticAt_entireSigned`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 141.

`Komorebi.PaperIV.hasSum_entireSigned`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 146.

`Komorebi.PaperIV.entireSigned_eq_tsum`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 155.

`Komorebi.PaperIV.entireSigned_independent_roots`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 162.

`Komorebi.PaperIV.entireSigned_zero_eq_average`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 171.

`Komorebi.PaperIV.entireTransfer_derivative_at_zero`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 177.

`Komorebi.PaperIV.entireTransfer_initial_data`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 194.

`Komorebi.PaperIV.entireSigned_eq_exponentialSum`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 207.

`Komorebi.PaperIV.entireSigned_ode`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 215.

`Komorebi.PaperIV.entireSigned_derivative_eq_scaled_transfer`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 227.

`Komorebi.PaperIV.entireSigned_initial_data`
theorem; `KomorebiPaperIV/EntireBridge.lean`, line 234.

11.48 Row 48: 4.3 and 5.4 Analytic Wronskians

Exact row scope. Exact analytic Wronskians using actual iterated derivatives: lambda^(N*(N-1)/2) for the ordered transfer family and 1 for the ordered signed family, for N>1. The transfer equality also holds at lambda=0. The proofs derive determinant factorization and constancy; no Wronskian or determinant value is assumed.

Manuscript labels. `cor:signed_wronskian`, `cor:wronskians`

`Komorebi.PaperIV.exponentialJetMatrix`
definition; `KomorebiPaperIV/Wronskians.lean`, line 21.

`Komorebi.PaperIV.exponentialJetMatrix_factor`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 25.

`Komorebi.PaperIV.det_exponentialJetMatrix`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 40.

`Komorebi.PaperIV.det_exponentialJetMatrix_constant`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 51.

`Komorebi.PaperIV.analyticWronskian`
definition; `KomorebiPaperIV/Wronskians.lean`, line 58.

`Komorebi.PaperIV.transferWronskian_constant`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 61.

`Komorebi.PaperIV.transferWronskian_eq`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 73.

`Komorebi.PaperIV.signedWronskian_constant`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 90.

`Komorebi.PaperIV.signedWronskian_eq_one`
theorem; `KomorebiPaperIV/Wronskians.lean`, line 102.

11.49 Row 49: 6.4 Companion first column and trace

Exact row scope. Convergent scalar series for each entry of the actual matrix exponential, exact first-column identification with the signed entire sections, every diagonal entry equal to the denominator, and trace=N*denominator. The full off-diagonal entry formula, characteristic polynomial, spectrum, and determinant-one claim remain outside this row.

Manuscript labels. `thm:companion_matrix_realization`

`Komorebi.PaperIV.hasSum_signedFundamentalMatrix_entry`
theorem; `KomorebiPaperIV/CompanionBridge.lean`, line 21.

`Komorebi.PaperIV.signedFundamentalMatrix_first_column`
theorem; `KomorebiPaperIV/CompanionBridge.lean`, line 35.

`Komorebi.PaperIV.signedFundamentalMatrix_diagonal`
theorem; `KomorebiPaperIV/CompanionBridge.lean`, line 72.

`Komorebi.PaperIV.signedFundamentalMatrix_trace`
theorem; `KomorebiPaperIV/CompanionBridge.lean`, line 85.

11.50 Row 50: 6.5 and 6.6 CompanionCompletion

Exact row scope. Exact convergent derivative-series bridge, every signed off-diagonal entry, and the determinant-one identity derived from the actual Wronskian.

Manuscript labels. `cor:companion_addition_inversion`, `cor:fundamental_matrix_entries`

`Komorebi.PaperIV.hasSum_exponentialSumJet`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 20.

`Komorebi.PaperIV.entireSigned_derivative_at_zero`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 36.

`Komorebi.PaperIV.hasSum_entireSigned_derivative`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 48.

`Komorebi.PaperIV.signedFundamentalMatrix_entry_eq_derivative`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 66.

`Komorebi.PaperIV.signedFundamentalMatrix_det`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 84.

`Komorebi.PaperIV.companionDisplacement`
definition; `KomorebiPaperIV/CompanionCompletion.lean`, line 97.

`Komorebi.PaperIV.companionDisplacement_lt`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 100.

`Komorebi.PaperIV.companionEntrySign`
definition; `KomorebiPaperIV/CompanionCompletion.lean`, line 108.

`Komorebi.PaperIV.signedShiftMatrix_pow_entry_packet`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 112.

`Komorebi.PaperIV.signedFundamentalMatrix_entry`
theorem; `KomorebiPaperIV/CompanionCompletion.lean`, line 159.

11.51 Row 51: 6.4 CompanionSpectrum

Exact row scope. Exact characteristic polynomial X^N+1, spectrum equal to the Nth roots of -1, N roots counted with multiplicity, and no repeated roots. No spectral certificate is assumed.

Manuscript labels. `thm:companion_matrix_realization`

`Komorebi.PaperIV.aeval_signedShiftMatrix_first_column`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 23.

`Komorebi.PaperIV.signedShiftMatrix_no_small_annihilator`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 38.

`Komorebi.PaperIV.aeval_signedShiftMatrix_X_pow_add_one`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 49.

`Komorebi.PaperIV.signedShiftMatrix_charpoly`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 55.

`Komorebi.PaperIV.signedShiftMatrix_mem_spectrum_iff`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 69.

`Komorebi.PaperIV.signedShiftMatrix_charpoly_separable`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 75.

`Komorebi.PaperIV.signedShiftMatrix_charpoly_roots_card`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 82.

`Komorebi.PaperIV.signedShiftMatrix_charpoly_roots_nodup`
theorem; `KomorebiPaperIV/CompanionSpectrum.lean`, line 89.

11.52 Row 52: 4.2 ODESolutions

Exact row scope. Exact uniqueness and explicit expansion for every entire analytic solution of y^(N)=lambda^N y, with unique coefficients when lambda is nonzero. This is the finite-basis assertion for the entire solution space.

Manuscript labels. `thm:exponential_cycle`

`Komorebi.PaperIV.ode_all_derivatives_zero`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 14.

`Komorebi.PaperIV.entire_ode_zero_of_initial_zero`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 30.

`Komorebi.PaperIV.entire_ode_unique`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 43.

`Komorebi.PaperIV.transferCombination`
definition; `KomorebiPaperIV/ODESolutions.lean`, line 60.

`Komorebi.PaperIV.analyticAt_transferCombination`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 63.

`Komorebi.PaperIV.transferCombination_derivative`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 71.

`Komorebi.PaperIV.transferCombination_ode`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 82.

`Komorebi.PaperIV.transferCombination_initial`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 93.

`Komorebi.PaperIV.entireTransfer_solution_expansion`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 108.

`Komorebi.PaperIV.transferCombination_injective`
theorem; `KomorebiPaperIV/ODESolutions.lean`, line 124.

11.53 Row 53: 4.2 and 5.3 AnalyticCycles

Exact row scope. Exact pointwise analytic derivative cycles, including both wraparound cases. The generic bridge derives analytic differential identities from the previously defined formal Taylor coefficients.

Manuscript labels. `thm:exponential_cycle`, `thm:signed_cycle`

`Komorebi.PaperIV.entire_eq_of_all_derivatives`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 15.

`Komorebi.PaperIV.entire_derivative_of_formal`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 31.

`Komorebi.PaperIV.entireTransfer_derivative_of_pos`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 49.

`Komorebi.PaperIV.entireTransfer_derivative_zero`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 61.

`Komorebi.PaperIV.entireSigned_derivative_of_pos`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 72.

`Komorebi.PaperIV.entireSigned_derivative_zero`
theorem; `KomorebiPaperIV/AnalyticCycles.lean`, line 84.

11.54 Row 54: 6.6 and 11.2 ExponentialBounds

Exact row scope. Explicit root witnesses and unconditional companion determinant for N>1; exact global bound norm(D_N(z)) <= exp(norm(z)). The order and matching type lower bound in Proposition 11.2 remain outside this row.

Manuscript labels. `cor:companion_addition_inversion`, `prop:orders`

`Komorebi.PaperIV.canonicalPrimitiveRoot`
definition; `KomorebiPaperIV/ExponentialBounds.lean`, line 12.

`Komorebi.PaperIV.canonicalSignedRoot`
definition; `KomorebiPaperIV/ExponentialBounds.lean`, line 15.

`Komorebi.PaperIV.canonicalPrimitiveRoot_isPrimitive`
theorem; `KomorebiPaperIV/ExponentialBounds.lean`, line 18.

`Komorebi.PaperIV.canonicalSignedRoot_pow`
theorem; `KomorebiPaperIV/ExponentialBounds.lean`, line 22.

`Komorebi.PaperIV.signedFundamentalMatrix_det_one`
theorem; `KomorebiPaperIV/ExponentialBounds.lean`, line 32.

`Komorebi.PaperIV.norm_signedRoot`

theorem; KomorebiPaperIV/ExponentialBounds.lean, line 38.

Komorebi.PaperIV.norm_entireSigned_zero_le
theorem; KomorebiPaperIV/ExponentialBounds.lean, line 46.

11.55 Row 55: 11.6 and 11.10 ZeroGeometry

Exact row scope. Full near-cancellation lemma with universal epsilon=1/2 and C=3/2, including uniqueness, with no strip restriction. Full order-two cosine zero set in integer and positive/negative natural forms, analytic multiplicity one, and exact lattice spacing.

Manuscript labels. cor:case_two_zeros, lem:near_cancellation

Komorebi.PaperIV.entireSigned_two_eq_cos
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 20.

Komorebi.PaperIV.entireSigned_two_zero_iff
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 30.

Komorebi.PaperIV.entireSigned_two_zero_derivative_ne_zero
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 37.

Komorebi.PaperIV.entireSigned_two_zero_iff_nat
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 52.

Komorebi.PaperIV.entireSigned_two_zero_order_one
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 78.

Komorebi.PaperIV.orderTwo_zero_spacing
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 86.

Komorebi.PaperIV.oddExponentialCenter
definition; KomorebiPaperIV/ZeroGeometry.lean, line 90.

Komorebi.PaperIV.near_cancellation_exists
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 93.

Komorebi.PaperIV.oddExponentialCenter_separation
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 114.

Komorebi.PaperIV.near_cancellation_unique
theorem; KomorebiPaperIV/ZeroGeometry.lean, line 129.

11.56 Row 56: 5.3 Signed ODE basis

Exact row scope. Exact spanning and unique coefficients for every entire analytic solution of y^(N)=-y, in the signed-section basis with its stated initial normalization.

Manuscript labels. thm:signed_cycle

Komorebi.PaperIV.signedCombination
definition; KomorebiPaperIV/ODESolutions.lean, line 134.

Komorebi.PaperIV.signedCombination_eq_transfer
theorem; KomorebiPaperIV/ODESolutions.lean, line 137.

Komorebi.PaperIV.entireSigned_solution_expansion
theorem; KomorebiPaperIV/ODESolutions.lean, line 148.

Komorebi.PaperIV.signedCombination_injective
theorem; KomorebiPaperIV/ODESolutions.lean, line 161.

11.57 Row 57: 11.7 Local zero existence and 11.8 leading correction

Exact row scope. General two-exponential perturbation theorem, proved by the actual logarithmic contraction. Explicit remainder and derivative bounds imply unique local zero, multiplicity one, displacement and quadratic correction bounds. These helper theorems do not assume or prove global exhaustion.

Manuscript labels. `thm:first_zero_correction`, `thm:large_zero_strings`

`Komorebi.PaperIV.norm_one_add_lower_half`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 19.

`Komorebi.PaperIV.log_one_add_lipschitz_half`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 25.

`Komorebi.PaperIV.logarithmicZeroMap`
definition; `KomorebiPaperIV/ZeroPerturbation.lean`, line 40.

`Komorebi.PaperIV.logarithmicZeroMap_mapsTo`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 43.

`Komorebi.PaperIV.logarithmicZeroMap_lipschitz`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 53.

`Komorebi.PaperIV.logarithmicZeroMap_fixed_iff`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 71.

`Komorebi.PaperIV.perturbed_exponential_unique_zero`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 113.

`Komorebi.PaperIV.perturbed_exponential_zero_distance`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 139.

`Komorebi.PaperIV.perturbed_exponential_pointwise_distance`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 155.

`Komorebi.PaperIV.hasDerivAt_perturbed_exponential`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 168.

`Komorebi.PaperIV.perturbed_exponential_zero_derivative_lower`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 180.

`Komorebi.PaperIV.perturbed_exponential_zero_order_one`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 194.

`Komorebi.PaperIV.perturbed_exponential_derivative_error`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 207.

`Komorebi.PaperIV.norm_log_one_add_sub_self_half`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 215.

`Komorebi.PaperIV.perturbed_exponential_first_correction`
theorem; `KomorebiPaperIV/ZeroPerturbation.lean`, line 224.

11.58 Row 58: 11.5 Two-term normalization and remainder jets

Exact row scope. Exact normalized finite sum and actual signed remainder, derivative jets of every order, zero equivalence and strip estimate from stated real-part gaps. The sector-to-strip confinement clause is outside this row.

Manuscript labels. `lem:two_term_dominance`

`Komorebi.PaperIV.finiteExponentialRemainder`
definition; `KomorebiPaperIV/ExponentialRemainder.lean`, line 17.

`Komorebi.PaperIV.analyticAt_finiteExponentialRemainder`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 21.

`Komorebi.PaperIV.hasDerivAt_finiteExponentialRemainder`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 27.

`Komorebi.PaperIV.iteratedDeriv_finiteExponentialRemainder`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 42.

`Komorebi.PaperIV.finiteExponentialRemainder_strip_bound`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 53.

`Komorebi.PaperIV.finite_two_term_normalization`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 82.

`Komorebi.PaperIV.signedExponentialRemainder`
definition; `KomorebiPaperIV/ExponentialRemainder.lean`, line 105.

`Komorebi.PaperIV.signed_two_term_normalization`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 109.

`Komorebi.PaperIV.signed_two_term_zero_iff`
theorem; `KomorebiPaperIV/ExponentialRemainder.lean`, line 128.

11.59 Row 59: 11.5 Canonical polygon gap and strip bounds

Exact row scope. The canonical roots have the precise positive gap gamma_N. The actual signed remainder satisfies the explicit strip estimate for every derivative order. No polygon-gap certificate is assumed. Global sector confinement is outside this row and supplied by rows 62–67.

Manuscript labels. `lem:two_term_dominance`

`Komorebi.PaperIV.canonicalSignedVertex`
definition; `KomorebiPaperIV/RegularPolygonGap.lean`, line 12.

`Komorebi.PaperIV.signedPolygonGap`
definition; `KomorebiPaperIV/RegularPolygonGap.lean`, line 15.

`Komorebi.PaperIV.canonicalSignedVertex_exp`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 18.

`Komorebi.PaperIV.canonicalSignedVertex_re`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 27.

`Komorebi.PaperIV.canonicalSignedVertex_im`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 31.

`Komorebi.PaperIV.norm_canonicalSignedVertex`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 35.

`Komorebi.PaperIV.canonicalSignedVertex_last_angle`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 39.

`Komorebi.PaperIV.canonicalSignedVertex_last_re`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 48.

`Komorebi.PaperIV.canonicalSignedVertex_last_im`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 52.

`Komorebi.PaperIV.signedPolygonGap_pos`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 56.

`Komorebi.PaperIV.canonicalSignedVertex_re_gap`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 64.

`Komorebi.PaperIV.canonicalSignedRemainder_strip_bound`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 95.

`Komorebi.PaperIV.canonicalSignedVertex_adjacent_difference`
theorem; `KomorebiPaperIV/RegularPolygonGap.lean`, line 117.

11.60 Row 60: 5.8 Conjugation, 11.5 disk bounds, and 11.7 local zero strings

Exact row scope. For every N >= 3, constructs a unique positive real simple zero near every sufficiently large lattice center with exponential displacement. All smallness hypotheses follow from the actual canonical remainder. Also proves conjugation and local lattice assignment. Exhaustion of all large complex zeros, orbit uniqueness and the original-denominator derivative asymptotic remain outside this row.

Manuscript labels. `cor:denominator_symmetries`, `lem:two_term_dominance`, `thm:large_zero_strings`

`Komorebi.PaperIV.canonicalSignedDenominator`
definition; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 20.

`Komorebi.PaperIV.signedZeroFrequency`
definition; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 23.

`Komorebi.PaperIV.signedZeroCenter`
definition; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 26.

`Komorebi.PaperIV.signed_sine_pos`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 29.

`Komorebi.PaperIV.signedZeroFrequency_ne_zero`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 35.

`Komorebi.PaperIV.signedZeroCenter_cancellation`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 39.

`Komorebi.PaperIV.canonicalSignedRemainder_disk_bound`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 55.

`Komorebi.PaperIV.signedRemainder_eventually_small_on_disks`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 80.

`Komorebi.PaperIV.canonicalSignedDenominator_conj`
theorem; `KomorebiPaperIV/SignedZeroLocalization.lean`, line 119.

`Komorebi.PaperIV.canonicalSignedDenominator_eventual_local_zero`

theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 134.

Komorebi.PaperIV.canonicalSignedDenominator_unique_zero_real
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 161.

Komorebi.PaperIV.canonicalSignedDenominator_local_zero_simple
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 175.

Komorebi.PaperIV.canonicalSignedDenominator_eventual_positive_simple_zeros
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 206.

Komorebi.PaperIV.canonicalSignedDenominator_local_zero_exponential_distance
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 248.

Komorebi.PaperIV.canonicalSignedDenominator_local_strings
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 273.

Komorebi.PaperIV.canonicalSignedDenominator_zero_lattice_distance
theorem; KomorebiPaperIV/SignedZeroLocalization.lean, line 309.

11.61 Row 61: 11.8 Leading correction and 11.7 normalized derivative

Exact row scope. Actual signed-remainder leading correction has an explicit quadratic envelope error, exactly a constant times exp(-2 gamma_N t). Proves normalized derivative error. Assumes the stated envelope smallness and local zero membership. Does not cover global exhaustion, derivative sign, reality of the displayed correction, or the full original-denominator asymptotic.

Manuscript labels. thm:first_zero_correction, thm:large_zero_strings

Komorebi.PaperIV.signedRemainderScale
definition; KomorebiPaperIV/SignedZeroCorrection.lean, line 14.

Komorebi.PaperIV.signedRemainderEnvelope
definition; KomorebiPaperIV/SignedZeroCorrection.lean, line 18.

Komorebi.PaperIV.signedRemainderEnvelope_pos
theorem; KomorebiPaperIV/SignedZeroCorrection.lean, line 21.

Komorebi.PaperIV.signedRemainderEnvelope_bounds
theorem; KomorebiPaperIV/SignedZeroCorrection.lean, line 27.

Komorebi.PaperIV.canonicalSignedDenominator_first_correction
theorem; KomorebiPaperIV/SignedZeroCorrection.lean, line 39.

Komorebi.PaperIV.signedRemainderEnvelope_sq
theorem; KomorebiPaperIV/SignedZeroCorrection.lean, line 62.

Komorebi.PaperIV.canonicalSignedDenominator_normalized_derivative_error
theorem; KomorebiPaperIV/SignedZeroCorrection.lean, line 71.

11.62 Row 62: 11.5 Exact sector confinement

Exact row scope. Exact Cartesian sector and explicit strip width log(N-1)/(2 sin(pi/N)) for every sector zero; derived finite polygon projection inequalities and one-term normalization. No large-radius or projection-gap certificate is assumed.

Manuscript labels. `lem:two_term_dominance`

`Komorebi.PaperIV.signedFundamentalSector`
definition; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 17.

`Komorebi.PaperIV.signed_cosine_pos`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 20.

`Komorebi.PaperIV.cosine_arc_upper`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 28.

`Komorebi.PaperIV.canonicalSignedVertex_upper_boundary`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 38.

`Komorebi.PaperIV.canonicalSignedVertex_upper_projection`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 65.

`Komorebi.PaperIV.canonicalSignedVertex_sector_gap`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 94.

`Komorebi.PaperIV.finite_one_term_normalization`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 120.

`Komorebi.PaperIV.finite_exponential_zero_balance`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 131.

`Komorebi.PaperIV.signedSectorWidth`
definition; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 147.

`Komorebi.PaperIV.canonicalSignedDenominator_upper_sector_zero_bound`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 150.

`Komorebi.PaperIV.canonicalSignedDenominator_sector_zero_bound`
theorem; `KomorebiPaperIV/SignedSectorConfinement.lean`, line 173.

11.63 Row 63: 11.7 Sector exhaustion

Exact row scope. Uniform strip smallness is derived from the actual remainder. Every sufficiently large sector zero belongs to a natural-index local disk, is positive real and simple, and is the only denominator zero in that disk.

Manuscript labels. `thm:large_zero_strings`

`Komorebi.PaperIV.signedSectorWidth_pos`
theorem; `KomorebiPaperIV/SignedSectorExhaustion.lean`, line 13.

`Komorebi.PaperIV.signedRemainder_eventually_small_on_strip`
theorem; `KomorebiPaperIV/SignedSectorExhaustion.lean`, line 18.

`Komorebi.PaperIV.signed_integer_center`
theorem; `KomorebiPaperIV/SignedSectorExhaustion.lean`, line 32.

`Komorebi.PaperIV.canonicalSignedDenominator_sector_exhaustion`
theorem; `KomorebiPaperIV/SignedSectorExhaustion.lean`, line 43.

11.64 Row 64: 5.8 Rotation and 11.7 global exhaustion

Exact row scope. Actual entire rotation invariance under every Nth root of unity; existence of a rotation into the fundamental sector with a real-part/norm bound; every sufficiently large complex zero rotates into a constructed positive real simple local zero.

Manuscript labels. `cor:denominator_symmetries`, `thm:large_zero_strings`

`Komorebi.PaperIV.canonicalSignedDenominator_rotation`
theorem; `KomorebiPaperIV/SignedGlobalExhaustion.lean`, line 14.

`Komorebi.PaperIV.signedFundamentalSector_polar`
theorem; `KomorebiPaperIV/SignedGlobalExhaustion.lean`, line 32.

`Komorebi.PaperIV.exists_signed_sector_rotation`
theorem; `KomorebiPaperIV/SignedGlobalExhaustion.lean`, line 73.

`Komorebi.PaperIV.canonicalSignedDenominator_global_exhaustion`
theorem; `KomorebiPaperIV/SignedGlobalExhaustion.lean`, line 110.

11.65 Row 65: 11.7 Rotation, multiplicity, and orbit parameters

Exact row scope. Exact frequency norm, rotation of analytic multiplicity and derivative, uniqueness of a unit-modulus positive real rotation, disjoint lattice disks, and global exhaustion using the canonical finite powers of omega. The full sequence/Big-O presentation is not encoded as a single theorem.

Manuscript labels. `thm:large_zero_strings`

`Komorebi.PaperIV.signedZeroFrequency_norm`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 11.

`Komorebi.PaperIV.canonicalSignedDenominator_order_rotation`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 19.

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_rotation`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 34.

`Komorebi.PaperIV.positive_real_rotation_unique`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 50.

`Komorebi.PaperIV.signedZeroCenter_disk_index_unique`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 66.

`Komorebi.PaperIV.canonicalSignedDenominator_large_zero_orbits`
theorem; `KomorebiPaperIV/SignedZeroOrbits.lean`, line 102.

11.66 Row 66: 11.7 Original denominator derivative

Exact row scope. Exact product-rule bridge at zeros, quantified relative error for the actual derivative, exact leading modulus (2 sin(pi/N)/N) exp(cos(pi/N) x), and a modulus-ratio error bound. These explicit estimates assume the stated local disk membership and center threshold; signs and the final sequence-level Big-O packaging remain outside this row.

Manuscript labels. `thm:large_zero_strings`

`Komorebi.PaperIV.signedNormalizationFactor`
definition; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 13.

`Komorebi.PaperIV.signedNormalizationFactor_ne_zero`
theorem; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 16.

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_normalization`
theorem; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 22.

`Komorebi.PaperIV.canonicalSignedDenominator_relative_derivative_error`
theorem; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 47.

`Komorebi.PaperIV.signed_derivative_leading_norm`
theorem; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 74.

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_modulus_error`
theorem; `KomorebiPaperIV/SignedDerivativeAsymptotics.lean`, line 86.

11.67 Row 67: 11.5 Full two-term dominance lemma

Exact row scope. Full argument-sector confinement, exact normalization, and a common positive constant for derivative jets q=0,1,2. Every constant is derived; Y=log(N-1)/(2 sin(pi/N)), R=1, C=4(N-2) exp(2Y).

Manuscript labels. `lem:two_term_dominance`

`Komorebi.PaperIV.signedFundamentalSector_of_arg`
theorem; `KomorebiPaperIV/SignedDominance.lean`, line 11.

`Komorebi.PaperIV.canonicalSignedRemainder_three_jet_bound`
theorem; `KomorebiPaperIV/SignedDominance.lean`, line 17.

`Komorebi.PaperIV.canonicalSignedDenominator_two_term_dominance`
theorem; `KomorebiPaperIV/SignedDominance.lean`, line 33.

11.68 Row 68: 11.11 Eventual simplicity and exact rays

Exact row scope. Full statement for every N >= 2: compact zero sets are finite, large zeros are simple and lie on canonical positive rays, and the exceptional set is finite.

Manuscript labels. `cor:eventual_simplicity`

`Komorebi.PaperIV.canonicalSignedDenominator_zero`
theorem; `KomorebiPaperIV/SignedFiniteExceptions.lean`, line 12.

`Komorebi.PaperIV.canonicalSignedDenominator_finite_zeros_on_compact`
theorem; `KomorebiPaperIV/SignedFiniteExceptions.lean`, line 18.

`Komorebi.PaperIV.onSignedZeroRay`
definition; `KomorebiPaperIV/SignedFiniteExceptions.lean`, line 32.

`Komorebi.PaperIV.canonicalSignedDenominator_eventually_simple_on_rays`
theorem; `KomorebiPaperIV/SignedFiniteExceptions.lean`, line 35.

`Komorebi.PaperIV.canonicalSignedDenominator_finite_exceptional_zeros`
theorem; `KomorebiPaperIV/SignedFiniteExceptions.lean`, line 68.

11.69 Row 69: 11.13 Entire reduced denominator and multiplicity

Exact row scope. The exact complex series is entire for N > 0. Convergent reindexing proves D_N(z)=Delta_N(z^N); the power substitution preserves analytic multiplicity away from zero.

Manuscript labels. `cor:reduced_zeros`

`Komorebi.PaperIV.entireReducedDenominator`
definition; `KomorebiPaperIV/ReducedEntire.lean`, line 14.

`Komorebi.PaperIV.reducedDenominator_coeff_complex`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 17.

`Komorebi.PaperIV.reducedDenominator_coeff_norm`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 22.

`Komorebi.PaperIV.reducedDenominator_entire_radius`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 28.

`Komorebi.PaperIV.analyticAt_entireReducedDenominator`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 38.

`Komorebi.PaperIV.hasSum_entireReducedDenominator`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 43.

`Komorebi.PaperIV.entireReducedDenominator_zero`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 50.

`Komorebi.PaperIV.canonicalSignedDenominator_eq_reduced_power`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 54.

`Komorebi.PaperIV.canonicalSignedDenominator_order_eq_reduced`
theorem; `KomorebiPaperIV/ReducedEntire.lean`, line 73.

11.70 Row 70: 11.10 Order two and 11.13 reduced zeros

Exact row scope. Finite exceptional reduced zeros, eventual positivity and simplicity for every N >= 2, and the exact positive half-integer squares with multiplicity one for N=2.

Manuscript labels. `cor:case_two_zeros`, `cor:reduced_zeros`

`Komorebi.PaperIV.exists_complex_nth_root`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 11.

`Komorebi.PaperIV.entireReducedDenominator_zero_ne_zero`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 18.

`Komorebi.PaperIV.entireReducedDenominator_finite_zeros_on_compact`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 22.

`Komorebi.PaperIV.onSignedZeroRay_power_positive`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 35.

`Komorebi.PaperIV.entireReducedDenominator_finite_exceptional_zeros`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 44.

`Komorebi.PaperIV.entireReducedDenominator_eventually_positive_simple`
theorem; `KomorebiPaperIV/ReducedZeros.lean`, line 68.

`Komorebi.PaperIV.entireReducedDenominator_two_zero_iff`

theorem; KomorebiPaperIV/ReducedZeros.lean, line 92.

Komorebi.PaperIV.entireReducedDenominator_two_zero_order_one
theorem; KomorebiPaperIV/ReducedZeros.lean, line 109.

11.71 Row 71: 11.7 Named strings and 11.13 reduced displacement

Exact row scope. Actual choice-defined positive sequences have local and global exhaustion and tail injectivity for N >= 3. The reduced sequence is the Nth power and satisfies the explicit C t_m^(N-1) exp(-gamma t_m) error. Definitions use a documented fallback only before the proved tail. Together with rows 69-70 this covers all clauses of 11.13; the remaining 11.7 assembly is supplied by row 81. Alternating derivative signs belong to 11.8 and are now supplied by rows 82–83.

Manuscript labels. cor:reduced_zeros, thm:large_zero_strings

Komorebi.PaperIV.isLocalizedPositiveZero
definition; KomorebiPaperIV/SignedZeroSequence.lean, line 12.

Komorebi.PaperIV.signedPositiveZero
definition; KomorebiPaperIV/SignedZeroSequence.lean, line 17.

Komorebi.PaperIV.signedPositiveZero_spec
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 21.

Komorebi.PaperIV.signedPositiveZero_eq_of_unique
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 27.

Komorebi.PaperIV.signedPositiveZero_local_strings
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 38.

Komorebi.PaperIV.signedPositiveZero_global_strings
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 59.

Komorebi.PaperIV.signedZeroCenter_ge_radius
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 72.

Komorebi.PaperIV.signedPositiveZero_pos
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 83.

Komorebi.PaperIV.signedPositiveZero_injective_on_tail
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 93.

Komorebi.PaperIV.reducedPositiveZero
definition; KomorebiPaperIV/SignedZeroSequence.lean, line 103.

Komorebi.PaperIV.reducedPositiveZero_global_strings
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 106.

Komorebi.PaperIV.reducedPositiveZero_injective_on_tail
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 127.

Komorebi.PaperIV.reducedPositiveZero_local_strings
theorem; KomorebiPaperIV/SignedZeroSequence.lean, line 138.

11.72 Row 72: 11.15 Sharp summability thresholds

Exact row scope. Actual positive sequences satisfy tau_m ~ t_m and lambda_m ~ t_m^N. Reciprocal real powers are summable exactly for s>1 and N*s>1 respectively, for N >= 3. This row does not assemble the full complex zero multiset, exceptional multiplicities, or the N=2 exponent statement.

Manuscript labels. `cor:exponent_of_convergence`

`Komorebi.PaperIV.signedZeroCenter_tendsto_atTop`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 14.

`Komorebi.PaperIV.signedPositiveZero_sub_center_isBigO_one`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 19.

`Komorebi.PaperIV.signedPositiveZero_equivalent_center`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 29.

`Komorebi.PaperIV.signedPositiveZero_tendsto_atTop`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 40.

`Komorebi.PaperIV.reducedPositiveZero_equivalent_center_power`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 45.

`Komorebi.PaperIV.signedPositiveZero_reciprocal_rpow_summable_iff`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 50.

`Komorebi.PaperIV.reducedPositiveZero_reciprocal_rpow_summable_iff`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 67.

`Komorebi.PaperIV.reducedPositiveZero_reciprocal_summable`
theorem; `KomorebiPaperIV/ZeroSequenceAsymptotics.lean`, line 76.

11.73 Row 73: 11.14 Positive-tail counting

Exact row scope. A finite lattice-window counting lemma is applied to the actual positive zero tail, giving density sin(pi/N)/pi with uniform bounded error for N >= 3. This row is limited to tails; rows 77–78 supply both full multiplicity-weighted counts.

Manuscript labels. `thm:zero_counting`

`Komorebi.PaperIV.sequenceCounting`
definition; `KomorebiPaperIV/ZeroSequenceCounting.lean`, line 14.

`Komorebi.PaperIV.lattice_window_counting`
theorem; `KomorebiPaperIV/ZeroSequenceCounting.lean`, line 17.

`Komorebi.PaperIV.signedPositiveZero_lattice_windows`
theorem; `KomorebiPaperIV/ZeroSequenceCounting.lean`, line 49.

`Komorebi.PaperIV.signedPositiveZero_tail_counting`
theorem; `KomorebiPaperIV/ZeroSequenceCounting.lean`, line 70.

11.74 Row 74: 12.1 Convergent entire tail product

Exact row scope. Reciprocal summability proves compact-uniform convergence and entire analyticity of the genus-zero product over any shifted chosen positive sequence for N >= 3. Its zero set is exactly that sequence. A sufficiently late shift consists of actual simple reduced zeros by row 71. The finite exceptional factor and equality with Delta_N are not asserted.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.reducedTailProduct`
definition; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 16.

`Komorebi.PaperIV.reducedPositiveZero_pos`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 19.

`Komorebi.PaperIV.reducedTail_reciprocal_summable`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 22.

`Komorebi.PaperIV.reducedTail_factors_norm_summable`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 26.

`Komorebi.PaperIV.reducedTailProduct_multipliable`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 32.

`Komorebi.PaperIV.reducedTailProduct_uniform_on_compact`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 36.

`Komorebi.PaperIV.reducedTailProduct_locally_uniform`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 50.

`Komorebi.PaperIV.analyticAt_reducedTailProduct`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 57.

`Komorebi.PaperIV.reducedTailProduct_zero`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 67.

`Komorebi.PaperIV.reducedTailProduct_ne_zero`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 70.

`Komorebi.PaperIV.reducedTailProduct_zero_iff`
theorem; `KomorebiPaperIV/ReducedCanonicalProduct.lean`, line 84.

11.75 Row 75: 11.14 Analytic multiplicity-weighted counting

Exact row scope. Actual disk counts sum analyticOrderNatAt over the finite zero set. The original and reduced entire denominators have finite order at every point, so the Nat-valued order loses no infinite multiplicity. Derived finite-weight comparison lemmas retain all exceptional weights.

Manuscript labels. `thm:zero_counting`

`Komorebi.PaperIV.finiteWeight`
definition; `KomorebiPaperIV/WeightedZeroCounts.lean`, line 13.

`Komorebi.PaperIV.finiteWeight_of_finite`
theorem; `KomorebiPaperIV/WeightedZeroCounts.lean`, line 17.

`Komorebi.PaperIV.finiteWeight_mono`
theorem; `KomorebiPaperIV/WeightedZeroCounts.lean`, line 23.

`Komorebi.PaperIV.finiteWeight_union_le`

theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 29.

Komorebi.PaperIV.finiteWeight_eq_ncard
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 38.

Komorebi.PaperIV.finiteWeight_between_tail_and_exceptions
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 47.

Komorebi.PaperIV.zeroDisk
definition; KomorebiPaperIV/WeightedZeroCounts.lean, line 60.

Komorebi.PaperIV.analyticZeroCount
definition; KomorebiPaperIV/WeightedZeroCounts.lean, line 63.

Komorebi.PaperIV.entire_analyticOrderAt_ne_top
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 67.

Komorebi.PaperIV.canonicalSignedDenominator_order_ne_top
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 75.

Komorebi.PaperIV.entireReducedDenominator_order_ne_top
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 80.

Komorebi.PaperIV.canonicalSignedDenominator_zeroDisk_finite
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 85.

Komorebi.PaperIV.entireReducedDenominator_zeroDisk_finite
theorem; KomorebiPaperIV/WeightedZeroCounts.lean, line 90.

11.76 Row 76: 11.14 Exhaustive zero enumeration

Exact row scope. Constructs a positive simple tail for every N >= 2, including the exact cosine case. Proves injectivity, lattice windows, exhaustive finite root-of-unity orbits, orbit norms and multiplicities. The structure is constructed from the proved zero theorems, never assumed in the final counting results.

Manuscript labels. `thm:zero_counting`

Komorebi.PaperIV.signedZeroEnumeration_exists_ge_three
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 30.

Komorebi.PaperIV.signedZeroEnumeration_exists_two
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 81.

Komorebi.PaperIV.signedZeroEnumeration_exists
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 121.

Komorebi.PaperIV.canonicalZeroEnumeration
definition; KomorebiPaperIV/ZeroEnumeration.lean, line 127.

Komorebi.PaperIV.SignedZeroEnumeration.orbit
definition; KomorebiPaperIV/ZeroEnumeration.lean, line 130.

Komorebi.PaperIV.SignedZeroEnumeration.orbit_norm
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 133.

Komorebi.PaperIV.SignedZeroEnumeration.orbit_injective
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 139.

Komorebi.PaperIV.SignedZeroEnumeration.orbit_zero_simple
theorem; KomorebiPaperIV/ZeroEnumeration.lean, line 153.

11.77 Row 77: 11.14 Full original-denominator counting

Exact row scope. For every N >= 2, the actual disk count, including all exceptional analytic multiplicities, differs from (N sin(pi/N)/pi) R by a uniform bounded amount for all R >= 0.

Manuscript labels. `thm:zero_counting`

`Komorebi.PaperIV.SignedZeroEnumeration.orbitDisk`
definition; `KomorebiPaperIV/FullZeroCounting.lean`, line 11.

`Komorebi.PaperIV.SignedZeroEnumeration.exceptions`
definition; `KomorebiPaperIV/FullZeroCounting.lean`, line 14.

`Komorebi.PaperIV.SignedZeroEnumeration.exceptions_finite`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 17.

`Komorebi.PaperIV.SignedZeroEnumeration.orbitDisk_finite_card`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 20.

`Komorebi.PaperIV.SignedZeroEnumeration.orbitDisk_subset`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 28.

`Komorebi.PaperIV.SignedZeroEnumeration.zeroDisk_subset_union`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 34.

`Komorebi.PaperIV.SignedZeroEnumeration.weighted_count_bounds`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 49.

`Komorebi.PaperIV.canonicalSignedDenominator_zero_counting`
theorem; `KomorebiPaperIV/FullZeroCounting.lean`, line 64.

11.78 Row 78: 11.14 Full reduced-denominator counting

Exact row scope. For every N >= 2, the actual reduced disk count differs from (sin(pi/N)/pi) R^(1/N) by a uniform bounded amount for all R >= 0, including exceptional multiplicities. Together with row 77, this fully encodes Theorem 11.14.

Manuscript labels. `thm:zero_counting`

`Komorebi.PaperIV.SignedZeroEnumeration.reducedRoot`
definition; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 11.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedDisk`
definition; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 14.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedExceptions`
definition; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 17.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedRoot_norm`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 20.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedRoot_injective`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 24.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedRoot_zero_simple`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 32.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedExceptions_finite`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 43.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedDisk_finite_card`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 47.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedDisk_subset`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 54.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedZeroDisk_subset_union`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 62.

`Komorebi.PaperIV.SignedZeroEnumeration.reduced_weighted_count_bounds`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 94.

`Komorebi.PaperIV.entireReducedDenominator_zero_counting_power`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 108.

`Komorebi.PaperIV.entireReducedDenominator_zero_counting`
theorem; `KomorebiPaperIV/ReducedZeroCounting.lean`, line 128.

11.79 Row 79: 11.15 Full zero convergence exponents

Exact row scope. The reciprocal-power series over all distinct zeros, weighted by actual analytic multiplicity, is summable exactly for s>1 for D_N and s>1/N for Delta_N, for every N >= 2. Both boundary divergences are included. Finite exceptional sets are removed only through a proved summability equivalence. This fully encodes Corollary 11.15.

Manuscript labels. `cor:exponent_of_convergence`

`Komorebi.PaperIV.ZeroPowerSummable`
definition; `KomorebiPaperIV/FullZeroSummability.lean`, line 14.

`Komorebi.PaperIV.summable_iff_comp_of_finite_complement`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 18.

`Komorebi.PaperIV.SignedZeroEnumeration.root_equivalent_lattice`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 28.

`Komorebi.PaperIV.SignedZeroEnumeration.root_reciprocal_rpow_summable_iff`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 46.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedRoot_reciprocal_rpow_summable_iff`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 64.

`Komorebi.PaperIV.SignedZeroEnumeration.orbitZero`
definition; `KomorebiPaperIV/FullZeroSummability.lean`, line 74.

`Komorebi.PaperIV.SignedZeroEnumeration.orbitZero_injective`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 78.

`Komorebi.PaperIV.SignedZeroEnumeration.orbitZero_finite_complement`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 83.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedZero`
definition; `KomorebiPaperIV/FullZeroSummability.lean`, line 95.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedZero_injective`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 99.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedZero_finite_complement`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 104.

`Komorebi.PaperIV.canonicalSignedDenominator_zero_power_summable_iff`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 122.

`Komorebi.PaperIV.entireReducedDenominator_zero_power_summable_iff`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 148.

`Komorebi.PaperIV.entireReducedDenominator_reciprocal_zeros_summable`
theorem; `KomorebiPaperIV/FullZeroSummability.lean`, line 164.

11.80 Row 80: 14.2 Absolute convergence and holomorphy

Exact row scope. Defines the exact complex summands m(z) exp(-s log(norm z)) over every zero. Proves both sharp absolute-convergence half-planes and failure of absolute convergence on and to their left, local uniform convergence by an explicit strip majorant, and holomorphy of both sums for every N >= 2. This fully encodes Theorem 14.2; meromorphic continuation in 14.3 is not asserted.

Manuscript labels. `thm:zeta_convergence`

`Komorebi.PaperIV.weightedDirichletTerm`
definition; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 13.

`Komorebi.PaperIV.weightedDirichletTerm_norm`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 16.

`Komorebi.PaperIV.weightedDirichletTerm_differentiable`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 26.

`Komorebi.PaperIV.weightedDirichletTerm_strip_bound`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 31.

`Komorebi.PaperIV.weightedDirichlet_locally_uniform`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 49.

`Komorebi.PaperIV.weightedDirichlet_differentiableOn`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 71.

`Komorebi.PaperIV.absoluteZeroZetaTerm`
definition; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 81.

`Komorebi.PaperIV.absoluteZeroZeta`
definition; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 85.

`Komorebi.PaperIV.zero_norm_pos_of_value_at_zero`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 87.

`Komorebi.PaperIV.absoluteZeroZetaTerm_norm`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 95.

`Komorebi.PaperIV.absoluteZeroZeta_summable_norm_iff`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 100.

`Komorebi.PaperIV.absoluteZeroZeta_locally_uniform`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 104.

`Komorebi.PaperIV.absoluteZeroZeta_differentiableOn`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 110.

`Komorebi.PaperIV.canonicalSignedDenominator_zeta_absolute_convergence`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 116.

`Komorebi.PaperIV.canonicalSignedDenominator_zeta_locally_uniform`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 121.

`Komorebi.PaperIV.canonicalSignedDenominator_zeta_holomorphic`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 128.

`Komorebi.PaperIV.entireReducedDenominator_zeta_absolute_convergence`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 134.

`Komorebi.PaperIV.entireReducedDenominator_zeta_locally_uniform`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 139.

`Komorebi.PaperIV.entireReducedDenominator_zeta_holomorphic`
theorem; `KomorebiPaperIV/ZeroZetaConvergence.lean`, line 146.

11.81 Row 81: 11.7 Complete large zero strings

Exact row scope. One theorem supplies a common tail cutoff, positive simple zeros, exponential displacement, unique global orbit/index representation, exact derivative covariance and the original derivative modulus asymptotic with exponential error measured at the actual zero. This fully encodes Theorem 11.7 for N >= 3. The alternating derivative sign in 11.8 is supplied by rows 82–83.

Manuscript labels. `thm:large_zero_strings`

`Komorebi.PaperIV.signedPositiveZero_derivative_asymptotic`
theorem; `KomorebiPaperIV/LargeZeroCompletion.lean`, line 14.

`Komorebi.PaperIV.signedPositiveZero_unique_global_strings`
theorem; `KomorebiPaperIV/LargeZeroCompletion.lean`, line 53.

`Komorebi.PaperIV.signedPositiveZero_derivative_rotation`
theorem; `KomorebiPaperIV/LargeZeroCompletion.lean`, line 97.

`Komorebi.PaperIV.canonicalSignedDenominator_large_zero_strings`
theorem; `KomorebiPaperIV/LargeZeroCompletion.lean`, line 114.

11.82 Row 82: 11.8 Reality and exact phase

Exact row scope. Canonical signed sections and their denominator derivative are real on the real axis. The lattice phase and the exact signed leading derivative factor are proved, including the alternating power of minus one.

Manuscript labels. `thm:first_zero_correction`

`Komorebi.PaperIV.entireSigned_canonical_conj`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 12.

`Komorebi.PaperIV.entireSigned_canonical_real`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 24.

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_real`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 30.

`Komorebi.PaperIV.signed_sine_mul_center`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 36.

`Komorebi.PaperIV.signedZeroCenter_positive_phase`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 41.

`Komorebi.PaperIV.signedZeroCenter_negative_phase`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 52.

`Komorebi.PaperIV.signedDerivativeScale`
definition; `KomorebiPaperIV/SignedRealPhase.lean`, line 60.

`Komorebi.PaperIV.signedDerivativeMainTerm`
definition; `KomorebiPaperIV/SignedRealPhase.lean`, line 63.

`Komorebi.PaperIV.signedDerivativeScale_pos`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 66.

`Komorebi.PaperIV.signedDerivativeMainTerm_ne_zero`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 70.

`Komorebi.PaperIV.signed_derivative_leading_phase`
theorem; `KomorebiPaperIV/SignedRealPhase.lean`, line 75.

11.83 Row 83: 11.8 Full leading correction and derivative sign

Exact row scope. At the actual named positive zeros, the leading correction is real and the displacement remainder is bounded by C exp(-2 gamma t). The signed derivative relative error is bounded by B exp(-gamma tau); the eventual alternating sign is also proved explicitly. Together with row 82 this fully encodes Theorem 11.8.

Manuscript labels. `thm:first_zero_correction`

`Komorebi.PaperIV.signedLeadingCorrection`
definition; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 12.

`Komorebi.PaperIV.signedLeadingCorrection_eq_real_quotient`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 16.

`Komorebi.PaperIV.signedLeadingCorrection_real`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 32.

`Komorebi.PaperIV.signedRemainderEnvelope_center_tendsto_zero`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 38.

`Komorebi.PaperIV.signedPositiveZero_first_correction`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 47.

`Komorebi.PaperIV.norm_real_phase_sub_one_le`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 70.

`Komorebi.PaperIV.relative_error_through_unit_phase`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 78.

`Komorebi.PaperIV.signedPositiveZero_signed_derivative_asymptotic`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 87.

`Komorebi.PaperIV.signedPositiveZero_eventual_derivative_sign`
theorem; `KomorebiPaperIV/SignedCorrectionCompletion.lean`, line 133.

11.84 Row 84: 11.9 Explicit cubic correction

Exact row scope. The actual cubic remainder, spectral gap 3/2, lattice centers, and correction (-1)^m exp(-3t/2)/sqrt(3) are identified exactly. The actual positive zeros have error bounded by C exp(-3t). This fully encodes Corollary 11.9.

Manuscript labels. `cor:N3_zero_correction`

`Komorebi.PaperIV.signedPolygonGap_three`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 10.

`Komorebi.PaperIV.signedZeroCenter_three`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 15.

`Komorebi.PaperIV.canonicalSignedVertex_three_one`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 21.

`Komorebi.PaperIV.canonicalSignedVertex_three_two`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 29.

`Komorebi.PaperIV.signedExponentialRemainder_three`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 41.

`Komorebi.PaperIV.signedLeadingCorrection_three`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 52.

`Komorebi.PaperIV.signedPositiveZero_three_correction`
theorem; `KomorebiPaperIV/CubicZeroCorrection.lean`, line 74.

11.85 Row 85: 14.8 Global derivative growth

Exact row scope. Derives the exponential derivative lower bound at every sufficiently large zero on every symmetry ray. Its inverse positive integer powers dominate every real polynomial weight. No growth certificate is assumed.

Manuscript labels. `thm:derivative_weighted_convergence`

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_lower_bound`
theorem; `KomorebiPaperIV/DerivativeGrowth.lean`, line 16.

`Komorebi.PaperIV.canonicalSignedDenominator_derivative_power_bound`
theorem; `KomorebiPaperIV/DerivativeGrowth.lean`, line 52.

11.86 Row 86: 14.8 Bounded logarithmic arguments

Exact row scope. A proved Weierstrass criterion for complex coefficients and bounded logarithmic arguments. Includes absolute norm summability, local uniform convergence, entire sums and half-plane holomorphy. The final denominator specialization in row 87 proves the required summability hypotheses.

Manuscript labels. `thm:derivative_weighted_convergence`

`Komorebi.PaperIV.branchedDirichletTerm`
definition; `KomorebiPaperIV/BranchedDirichlet.lean`, line 12.

`Komorebi.PaperIV.branchedDirichletTerm_differentiable`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 15.

`Komorebi.PaperIV.branchedDirichletTerm_norm`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 20.

`Komorebi.PaperIV.branchedDirichletTerm_strip_bound`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 30.

`Komorebi.PaperIV.branchedDirichlet_summable_norm`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 55.

`Komorebi.PaperIV.branchedDirichlet_locally_uniform`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 67.

`Komorebi.PaperIV.branchedDirichlet_entire`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 88.

`Komorebi.PaperIV.branchedDirichlet_halfplane`
theorem; `KomorebiPaperIV/BranchedDirichlet.lean`, line 101.

11.87 Row 87: 14.8 Full derivative-weighted convergence

Exact row scope. Defines the exact exp(-s Log z)/(D_N prime(z)^q) series over all actual simple zeros, retaining exceptional simple zeros and the complex derivative weight. For every N >= 3 and q >= 1, proves absolute and locally uniform convergence everywhere and an entire sum. For N=2, proves absolute and locally uniform convergence and holomorphy on Re(s)>1. Logarithms must exponentiate to the zero and have arguments in a fixed bounded interval. This fully encodes Theorem 14.8.

Manuscript labels. `thm:derivative_weighted_convergence`

`Komorebi.PaperIV.SimpleZeros`
definition; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 15.

`Komorebi.PaperIV.SimpleZeros.toZero`
definition; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 17.

`Komorebi.PaperIV.SimpleZeros.toZero_injective`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 20.

`Komorebi.PaperIV.SimpleZeros.order_one`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 25.

`Komorebi.PaperIV.canonicalSignedDenominator_simple_zero_power_summable`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 31.

`Komorebi.PaperIV.simple_logarithm_re`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 43.

`Komorebi.PaperIV.derivativeZeroWeight`
definition; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 50.

`Komorebi.PaperIV.derivativeZeroWeight_norm`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 53.

`Komorebi.PaperIV.derivativeZeroWeight_power_summable`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 59.

`Komorebi.PaperIV.canonicalSignedDenominator_two_derivative_norm`
theorem; `KomorebiPaperIV/DerivativeWeightedZeta.lean`, line 79.

`Komorebi.PaperIV.derivativeZeroWeight_two_power_summable`

theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 92.

Komorebi.PaperIV.derivativeWeightedZeroZetaTerm
definition; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 100.

Komorebi.PaperIV.derivativeWeightedZeroZeta
definition; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 105.

Komorebi.PaperIV.derivativeWeightedZeroZetaTerm_eq
theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 109.

Komorebi.PaperIV.derivativeWeightedZeroZeta_summable_norm
theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 117.

Komorebi.PaperIV.derivativeWeightedZeroZeta_two_summable_norm
theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 128.

Komorebi.PaperIV.derivativeWeightedZeroZeta_entire
theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 141.

Komorebi.PaperIV.derivativeWeightedZeroZeta_two_holomorphic
theorem; KomorebiPaperIV/DerivativeWeightedZeta.lean, line 156.

11.88 Row 88: 14.3 Entire exponential comparison

Exact row scope. A locally uniform summable majorant on every bounded parameter set proves that exponentially displaced positive sequences have an entire difference of zeta sums. The needed displacement and summability are derived in row 90.

Manuscript labels. thm:zeta_continuation

Komorebi.PaperIV.positiveZetaTerm
definition; KomorebiPaperIV/ExponentialZetaComparison.lean, line 15.

Komorebi.PaperIV.exponentialZetaDifference
definition; KomorebiPaperIV/ExponentialZetaComparison.lean, line 17.

Komorebi.PaperIV.positiveZetaTerm_entire
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 20.

Komorebi.PaperIV.exponentialZetaDifference_entire
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 24.

Komorebi.PaperIV.real_log_lipschitz_one
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 28.

Komorebi.PaperIV.positiveZetaTerm_norm_le_rpow
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 38.

Komorebi.PaperIV.exponentialZetaDifference_factor
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 49.

Komorebi.PaperIV.summable_polynomial_exponential
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 58.

Komorebi.PaperIV.exponentialZetaDifference_majorant
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 73.

Komorebi.PaperIV.exponentialZetaDifference_locally_uniform
theorem; KomorebiPaperIV/ExponentialZetaComparison.lean, line 108.

`Komorebi.PaperIV.exponentialZetaDifference_sum_entire`
theorem; `KomorebiPaperIV/ExponentialZetaComparison.lean`, line 127.

11.89 Row 89: 14.3 Hurwitz model and pole

Exact row scope. The scaled Hurwitz model is meromorphic on the whole plane, holomorphic off its sole possible pole, with the exact residue. A nonzero residue gives meromorphic order -1. Uses the pinned proved Mathlib Hurwitz theory.

Manuscript labels. `thm:zeta_continuation`

`Komorebi.PaperIV.hurwitzRegularPart`
definition; `KomorebiPaperIV/HurwitzLattice.lean`, line 15.

`Komorebi.PaperIV.hurwitzRegularPart_entire`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 18.

`Komorebi.PaperIV.hurwitzZeta_meromorphicAt`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 30.

`Komorebi.PaperIV.hurwitzLatticeModel`
definition; `KomorebiPaperIV/HurwitzLattice.lean`, line 40.

`Komorebi.PaperIV.hurwitzLatticeModel_meromorphicAt`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 44.

`Komorebi.PaperIV.hurwitzLatticeModel_differentiableAt`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 51.

`Komorebi.PaperIV.hurwitzLatticeModel_residue`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 63.

`Komorebi.PaperIV.meromorphicOrderAt_eq_neg_one_of_residue`
theorem; `KomorebiPaperIV/HurwitzLattice.lean`, line 95.

11.90 Row 90: 14.3 Actual positive-zero comparison

Exact row scope. The proved enumeration carries its actual exponential displacement and offset. Its entire correction and finite missing lattice prefix are constructed, and the exact root-zeta identity with the unshifted Hurwitz model is proved.

Manuscript labels. `thm:zeta_continuation`

`Komorebi.PaperIV.positiveZetaTerm_mul`
theorem; `KomorebiPaperIV/ZeroLatticeComparison.lean`, line 13.

`Komorebi.PaperIV.positiveZetaTerm_eq_cpow`
theorem; `KomorebiPaperIV/ZeroLatticeComparison.lean`, line 18.

`Komorebi.PaperIV.positiveZetaTerm_norm`
theorem; `KomorebiPaperIV/ZeroLatticeComparison.lean`, line 26.

`Komorebi.PaperIV.positiveZetaTerm_lattice_hasSum`
theorem; `KomorebiPaperIV/ZeroLatticeComparison.lean`, line 31.

`Komorebi.PaperIV.signedZeroCenter_positive`
theorem; `KomorebiPaperIV/ZeroLatticeComparison.lean`, line 43.

`Komorebi.PaperIV.signedZeroCenter_reciprocal_square_summable`

theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 48.

Komorebi.PaperIV.SignedZeroEnumeration.root_tendsto_atTop
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 55.

Komorebi.PaperIV.SignedZeroEnumeration.latticeCorrection
definition; KomorebiPaperIV/ZeroLatticeComparison.lean, line 61.

Komorebi.PaperIV.SignedZeroEnumeration.latticeCorrection_locally_uniform
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 64.

Komorebi.PaperIV.SignedZeroEnumeration.latticeCorrection_entire
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 77.

Komorebi.PaperIV.SignedZeroEnumeration.positiveZetaTerm_summable
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 83.

Komorebi.PaperIV.SignedZeroEnumeration.latticePrefix
definition; KomorebiPaperIV/ZeroLatticeComparison.lean, line 90.

Komorebi.PaperIV.SignedZeroEnumeration.latticePrefix_entire
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 93.

Komorebi.PaperIV.SignedZeroEnumeration.root_zeta_eq_model
theorem; KomorebiPaperIV/ZeroLatticeComparison.lean, line 99.

11.91 Row 91: 14.3 All-zero decomposition

Exact row scope. Both original and reduced zeta sums split into the actual simple tail and the finite exceptional zeros with their analytic multiplicities. The factor N and the scaled parameter Ns are proved.

Manuscript labels. `thm:zeta_continuation`

Komorebi.PaperIV.tsum_injection_add_complement
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 12.

Komorebi.PaperIV.finite_subtype_tsum_entire
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 18.

Komorebi.PaperIV.SignedZeroEnumeration.reducedExceptionZeta
definition; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 26.

Komorebi.PaperIV.SignedZeroEnumeration.absoluteExceptionZeta
definition; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 30.

Komorebi.PaperIV.absoluteZeroZetaTerm_entire
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 34.

Komorebi.PaperIV.SignedZeroEnumeration.reducedExceptionZeta_entire
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 39.

Komorebi.PaperIV.SignedZeroEnumeration.absoluteExceptionZeta_entire
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 44.

Komorebi.PaperIV.SignedZeroEnumeration.reducedZetaTerm
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 49.

Komorebi.PaperIV.SignedZeroEnumeration.orbitZetaTerm
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 60.

Komorebi.PaperIV.SignedZeroEnumeration.reducedZeta_decomposition

theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 67.

Komorebi.PaperIV.SignedZeroEnumeration.absoluteZeta_decomposition
theorem; KomorebiPaperIV/ZeroZetaDecomposition.lean, line 77.

11.92 Row 92: 14.3 Full unweighted continuation

Exact row scope. Constructs both entire corrections, proves equality on the original convergence domains, global meromorphic continuation, holomorphy off the sole pole, meromorphic order -1 at the pole, and both exact manuscript residues. Fully encodes Theorem 14.3.

Manuscript labels. `thm:zeta_continuation`

Komorebi.PaperIV.reducedZetaCorrection
definition; KomorebiPaperIV/ZeroZetaContinuation.lean, line 12.

Komorebi.PaperIV.absoluteZetaCorrection
definition; KomorebiPaperIV/ZeroZetaContinuation.lean, line 17.

Komorebi.PaperIV.reducedZetaCorrection_entire
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 22.

Komorebi.PaperIV.absoluteZetaCorrection_entire
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 29.

Komorebi.PaperIV.entireReducedDenominator_zeta_eq_hurwitz
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 38.

Komorebi.PaperIV.canonicalSignedDenominator_zeta_eq_hurwitz
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 51.

Komorebi.PaperIV.reducedZetaContinuation
definition; KomorebiPaperIV/ZeroZetaContinuation.lean, line 61.

Komorebi.PaperIV.absoluteZetaContinuation
definition; KomorebiPaperIV/ZeroZetaContinuation.lean, line 64.

Komorebi.PaperIV.reducedZetaContinuation_meromorphic
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 67.

Komorebi.PaperIV.absoluteZetaContinuation_meromorphic
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 72.

Komorebi.PaperIV.reducedZetaContinuation_holomorphic_off_pole
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 77.

Komorebi.PaperIV.absoluteZetaContinuation_holomorphic_off_pole
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 81.

Komorebi.PaperIV.entire_add_preserves_residue
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 87.

Komorebi.PaperIV.reducedZetaContinuation_residue
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 96.

Komorebi.PaperIV.absoluteZetaContinuation_residue
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 106.

Komorebi.PaperIV.reducedZetaContinuation_simple_pole
theorem; KomorebiPaperIV/ZeroZetaContinuation.lean, line 120.

`Komorebi.PaperIV.absoluteZetaContinuation_simple_pole`
theorem; `KomorebiPaperIV/ZeroZetaContinuation.lean`, line 127.

11.93 Row 93: 14.6 Full branch-dependent continuation

Exact row scope. Defines the actual multiplicity-weighted branched zero series. For logarithms real on the proved positive tail, it agrees with the absolute zeta plus an entire finite correction. Proves global continuation, the same single simple pole/residue, and an entire difference between choices. Intermediate finite-correction claims hold even for arbitrary exceptional complex values. Fully encodes Corollary 14.6.

Manuscript labels. `cor:branch_grouped_continuation`

`Komorebi.PaperIV.branchedZeroZetaTerm`
definition; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 14.

`Komorebi.PaperIV.branchedZeroZeta`
definition; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 18.

`Komorebi.PaperIV.branchedZeroZetaTerm_entire`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 21.

`Komorebi.PaperIV.RealLogOnReducedTail`
definition; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 29.

`Komorebi.PaperIV.reducedBranchCorrection`
definition; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 34.

`Komorebi.PaperIV.reducedBranchCorrection_entire`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 40.

`Komorebi.PaperIV.branchedZeroZetaTerm_eq_absolute_on_tail`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 49.

`Komorebi.PaperIV.branchedReducedZeta_summable`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 59.

`Komorebi.PaperIV.branchedReducedZeta_eq_absolute_add_entire`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 79.

`Komorebi.PaperIV.branchedReducedZetaContinuation`
definition; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 103.

`Komorebi.PaperIV.branchedReducedZetaContinuation_eq`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 107.

`Komorebi.PaperIV.branchedReducedZetaContinuation_meromorphic`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 114.

`Komorebi.PaperIV.branchedReducedZetaContinuation_holomorphic_off_pole`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 120.

`Komorebi.PaperIV.branchedReducedZetaContinuation_residue`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 125.

`Komorebi.PaperIV.branchedReducedZetaContinuation_simple_pole`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 131.

`Komorebi.PaperIV.branchedReducedZetaContinuation_difference_entire`
theorem; `KomorebiPaperIV/BranchZetaContinuation.lean`, line 141.

11.94 Row 94: 14.4 Exact quartic factorization and zeros

Exact row scope. Proves D4(z)=cos(z/sqrt(2))*cosh(z/sqrt(2)), its exhaustive real/imaginary zero strings, and simplicity of every zero.

Manuscript labels. `cor:exact_zeta_models`

`Komorebi.PaperIV.quartic_root_square`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 12.

`Komorebi.PaperIV.canonicalSignedDenominator_four_eq_cos_cosh`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 19.

`Komorebi.PaperIV.complex_cos_zero_iff_nat`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 43.

`Komorebi.PaperIV.quarticPositiveZero`
definition; `KomorebiPaperIV/QuarticExactZeros.lean`, line 49.

`Komorebi.PaperIV.quarticPositiveZero_pos`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 51.

`Komorebi.PaperIV.quarticPositiveZero_div_sqrt`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 55.

`Komorebi.PaperIV.canonicalSignedDenominator_four_zero_iff`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 63.

`Komorebi.PaperIV.quarticPositiveZero_simple`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 112.

`Komorebi.PaperIV.canonicalSignedDenominator_four_zero_order_one`
theorem; `KomorebiPaperIV/QuarticExactZeros.lean`, line 141.

11.95 Row 95: 14.4 Full exact order-two/four models

Exact row scope. Proves both exact zeta formulas in orders two and four with no omitted exceptional zeros. The identity principle proves that the entire corrections constructed in Theorem 14.3 vanish identically. Fully encodes Corollary 14.4.

Manuscript labels. `cor:exact_zeta_models`

`Komorebi.PaperIV.exactLatticeEnumeration`
definition; `KomorebiPaperIV/ExactZetaModels.lean`, line 13.

`Komorebi.PaperIV.SignedZeroEnumeration.reducedZero_surjective_of_orbitZero`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 41.

`Komorebi.PaperIV.SignedZeroEnumeration.exceptions_zero_of_surjective`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 57.

`Komorebi.PaperIV.zero_zeta_exact_of_lattice`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 66.

`Komorebi.PaperIV.signedZeroCenter_four`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 100.

`Komorebi.PaperIV.orderTwo_full_lattice`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 108.

`Komorebi.PaperIV.orderFour_full_lattice`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 129.

`Komorebi.PaperIV.orderTwo_exact_zero_zeta_models`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 157.

`Komorebi.PaperIV.orderFour_exact_zero_zeta_models`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 165.

`Komorebi.PaperIV.entire_zero_of_halfplane`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 177.

`Komorebi.PaperIV.zero_zeta_corrections_of_lattice`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 188.

`Komorebi.PaperIV.orderTwo_zeta_corrections_zero`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 207.

`Komorebi.PaperIV.orderFour_zeta_corrections_zero`
theorem; `KomorebiPaperIV/ExactZetaModels.lean`, line 213.

11.96 Row 96: 12.1 Complete multiplicity-indexed product

Exact row scope. Constructs the product over every actual reduced zero repeated according to analytic multiplicity, including all exceptional zeros and N=2. Proves countability, reciprocal summability, compact-uniform convergence, entire realization, normalization and exact zero set. Equality with Delta is supplied by row 102.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.ZeroMultiplicityIndex`
definition; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 14.

`Komorebi.PaperIV.indexedZero`
definition; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 17.

`Komorebi.PaperIV.fullCanonicalProduct`
definition; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 19.

`Komorebi.PaperIV.zeroMultiplicity_reciprocal_summable`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 22.

`Komorebi.PaperIV.fullCanonicalProduct_factors_summable`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 31.

`Komorebi.PaperIV.fullCanonicalProduct_multipliable`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 35.

`Komorebi.PaperIV.fullCanonicalProduct_uniform_on_compact`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 39.

`Komorebi.PaperIV.fullCanonicalProduct_locally_uniform`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 51.

`Komorebi.PaperIV.fullCanonicalProduct_entire`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 58.

`Komorebi.PaperIV.fullCanonicalProduct_zero`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 68.

`Komorebi.PaperIV.fullCanonicalProduct_ne_zero`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 71.

`Komorebi.PaperIV.fullCanonicalProduct_zero_iff_index`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 89.

`Komorebi.PaperIV.indexedZero_ne_zero`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 108.

`Komorebi.PaperIV.entire_zero_multiplicity_pos`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 116.

`Komorebi.PaperIV.fullCanonicalProduct_zero_iff`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 122.

`Komorebi.PaperIV.zeroMultiplicityIndex_countable`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 132.

`Komorebi.PaperIV.entireReducedDenominator_full_product_properties`
theorem; `KomorebiPaperIV/FullCanonicalProduct.lean`, line 143.

11.97 Row 97: 12.1 Exact product multiplicities

Exact row scope. Factors out every copy of a selected zero; the remaining product is entire and nonzero there. Proves that the complete product and the reduced denominator have exactly the same analytic order at every point.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.reciprocalProduct`
definition; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 13.

`Komorebi.PaperIV.reciprocalProduct_locally_uniform`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 15.

`Komorebi.PaperIV.reciprocalProduct_entire`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 27.

`Komorebi.PaperIV.reciprocalProduct_ne_zero`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 37.

`Komorebi.PaperIV.zeroIndexFiberEquiv`
definition; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 47.

`Komorebi.PaperIV.otherZeroProduct`
definition; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 60.

`Komorebi.PaperIV.otherZeroProduct_entire`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 63.

`Komorebi.PaperIV.otherZeroProduct_ne_zero_at_zero`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 67.

`Komorebi.PaperIV.fullCanonicalProduct_factor_at_zero`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 76.

`Komorebi.PaperIV.linear_zero_factor_order`
theorem; `KomorebiPaperIV/CanonicalProductMultiplicity.lean`, line 95.

`Komorebi.PaperIV.fullCanonicalProduct_order_at_zero`

theorem; KomorebiPaperIV/CanonicalProductMultiplicity.lean, line 103.

Komorebi.PaperIV.fullCanonicalProduct_same_orders
theorem; KomorebiPaperIV/CanonicalProductMultiplicity.lean, line 118.

Komorebi.PaperIV.entireReducedDenominator_full_product_same_orders
theorem; KomorebiPaperIV/CanonicalProductMultiplicity.lean, line 128.

11.98 Row 98: 12.1 Entire nonvanishing quotient

Exact row scope. Removes all quotient singularities using the proved order equality. Constructs a globally entire, nowhere-zero quotient with value 1 at zero, and proves Delta equals the full product times this quotient. Constancy is proved in row 102.

Manuscript labels. thm:genus_zero_product

Komorebi.PaperIV.canonicalProductQuotient
definition; KomorebiPaperIV/CanonicalProductQuotient.lean, line 12.

Komorebi.PaperIV.rawCanonicalProductQuotient_meromorphic
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 15.

Komorebi.PaperIV.rawCanonicalProductQuotient_order_zero
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 20.

Komorebi.PaperIV.canonicalProductQuotient_entire
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 32.

Komorebi.PaperIV.canonicalProductQuotient_ne_zero
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 40.

Komorebi.PaperIV.canonicalProductQuotient_eq_div
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 47.

Komorebi.PaperIV.fullCanonicalProduct_times_quotient
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 58.

Komorebi.PaperIV.canonicalProductQuotient_zero
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 66.

Komorebi.PaperIV.entireReducedDenominator_product_quotient
theorem; KomorebiPaperIV/CanonicalProductQuotient.lean, line 71.

11.99 Row 99: 12.1 Growth reduction for the quotient

Exact row scope. Constructs an entire normalized logarithm of the quotient. Proves a Borel-Caratheodory rigidity criterion from a uniform disk log-norm bound M(R) with M(R)/R tending to zero. The required bound for the actual quotient is derived in rows 100–102, completing Theorem 12.1.

Manuscript labels. thm:genus_zero_product

Komorebi.PaperIV.entire_normalized_exists_log
theorem; KomorebiPaperIV/SublinearExponentialRigidity.lean, line 16.

Komorebi.PaperIV.entire_realPart_sublinear_constant
theorem; KomorebiPaperIV/SublinearExponentialRigidity.lean, line 41.

Komorebi.PaperIV.entire_zeroFree_sublinear_logNorm_eq_one
theorem; KomorebiPaperIV/SublinearExponentialRigidity.lean, line 64.

`Komorebi.PaperIV.entireReducedDenominator_product_quotient_has_log`
theorem; `KomorebiPaperIV/SublinearExponentialRigidity.lean`, line 82.

11.100 Row 100: 12.1 Jensen-Poisson quotient estimate

Exact row scope. Proves a normalized entire reciprocal proximity bound by Jensen, quotient proximity by division, and a Poisson real-part estimate. These yield rigidity for a factorization with proved sublinear bounds on both entire factors.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.entire_logCounting_poles_zero`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 14.

`Komorebi.PaperIV.entire_normalized_proximity_inv_le`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 20.

`Komorebi.PaperIV.entire_eventually_codiscrete_ne_zero`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 32.

`Komorebi.PaperIV.entire_quotient_proximity_le`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 45.

`Komorebi.PaperIV.entire_re_le_three_proximity_exp`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 62.

`Komorebi.PaperIV.entire_proximity_le_of_growth`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 98.

`Komorebi.PaperIV.entire_sublinear_factor_rigidity`
theorem; `KomorebiPaperIV/EntireQuotientGrowth.lean`, line 113.

11.101 Row 101: 12.1 Actual denominator and product growth

Exact row scope. Derives the needed growth bounds from actual zero-power summability and the entire denominator estimate. The fixed exponent 3/4 is valid for every N>=2. No quotient-growth certificate is assumed.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.log_one_add_le_rpow_div`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 14.

`Komorebi.PaperIV.zeroMultiplicity_reciprocal_rpow_summable`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 23.

`Komorebi.PaperIV.reciprocalPowerMass`
definition; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 33.

`Komorebi.PaperIV.reciprocalPowerMass_nonneg`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 36.

`Komorebi.PaperIV.fullCanonicalProduct_growth`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 40.

`Komorebi.PaperIV.entireReducedDenominator_norm_le`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 82.

`Komorebi.PaperIV.entireReducedDenominator_and_product_growth`
theorem; `KomorebiPaperIV/CanonicalProductGrowth.lean`, line 91.

11.102 Row 102: 12.1 Full Hadamard identification

Exact row scope. The actual normalized quotient equals one. Proves Delta equals the complete multiplicity-indexed product, compact-uniform convergence to Delta, and the original-variable product for D_N. Fully encodes Theorem 12.1.

Manuscript labels. `thm:genus_zero_product`

`Komorebi.PaperIV.entireReducedDenominator_quotient_eq_one`
theorem; `KomorebiPaperIV/HadamardIdentification.lean`, line 15.

`Komorebi.PaperIV.entireReducedDenominator_eq_fullCanonicalProduct`
theorem; `KomorebiPaperIV/HadamardIdentification.lean`, line 32.

`Komorebi.PaperIV.entireReducedDenominator_hadamard_uniform`
theorem; `KomorebiPaperIV/HadamardIdentification.lean`, line 41.

`Komorebi.PaperIV.canonicalSignedDenominator_hadamard`
theorem; `KomorebiPaperIV/HadamardIdentification.lean`, line 50.

11.103 Row 103: 12.7 Convergent Mittag-Leffler expansion

Exact row scope. Proves the logarithmic derivative of the full product and the exact tangent partial fractions, with absolute and locally uniform convergence away from poles. Auxiliary tail-convergence lemmas use Lean total division at poles; the analytic identity is restricted to nonpoles. Fully encodes Corollary 12.7.

Manuscript labels. `cor:mittag_leffler_tangent`

`Komorebi.PaperIV.zeroPartialFraction`
definition; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 14.

`Komorebi.PaperIV.zeroPartialFraction_uniform_majorant`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 17.

`Komorebi.PaperIV.zeroPartialFraction_norm_summable`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 34.

`Komorebi.PaperIV.zeroPartialFraction_uniform_on_compact`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 43.

`Komorebi.PaperIV.logDeriv_linear_factor`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 53.

`Komorebi.PaperIV.fullCanonicalProduct_logDeriv`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 61.

`Komorebi.PaperIV.entireReducedDenominator_partialFractions`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 84.

`Komorebi.PaperIV.zeroPartialFraction_locally_uniform`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 97.

`Komorebi.PaperIV.analyticTransferTangent`
definition; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 105.

`Komorebi.PaperIV.analyticTransferTangent_eq_reduced_logDeriv`
theorem; `KomorebiPaperIV/CanonicalPartialFractions.lean`, line 108.

`Komorebi.PaperIV.analyticTransferTangent_mittagLeffler`

theorem; KomorebiPaperIV/CanonicalPartialFractions.lean, line 119.

`Komorebi.PaperIV.analyticTransferTangent_mittagLeffler_absolute`
theorem; KomorebiPaperIV/CanonicalPartialFractions.lean, line 127.

`Komorebi.PaperIV.analyticTransferTangent_mittagLeffler_locally_uniform`
theorem; KomorebiPaperIV/CanonicalPartialFractions.lean, line 135.

11.104 Row 104: 12.6 Actual zero moments and derivatives

Exact row scope. Defines moments over all actual zeros repeated by analytic multiplicity. Proves termwise derivatives of the locally uniform resolvent series and the exact factorial-normalized derivatives at zero.

Manuscript labels. `thm:newton_sums`

`Komorebi.PaperIV.locally_uniform_sum_deriv`
theorem; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 13.

`Komorebi.PaperIV.zeroResolventJet`
definition; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 22.

`Komorebi.PaperIV.zeroResolventJet_hasDerivAt`
theorem; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 25.

`Komorebi.PaperIV.zeroResolventJet_locally_uniform`
theorem; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 37.

`Komorebi.PaperIV.zeroPowerSum`
definition; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 65.

`Komorebi.PaperIV.zeroPowerSum_hasSum_from_resolvent`
theorem; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 68.

`Komorebi.PaperIV.zeroPowerSum_resolvent_derivative`
theorem; KomorebiPaperIV/ZeroMomentDerivatives.lean, line 75.

11.105 Row 105: 12.6 Kappa identification and Newton recurrence

Exact row scope. Identifies every positive-index formal kappa coefficient with the actual zero moment and proves the exact Newton recurrence. The logarithm-series clause is supplied by row 109.

Manuscript labels. `thm:newton_sums`

`Komorebi.PaperIV.entireReducedDenominator_iteratedDeriv_zero`
theorem; KomorebiPaperIV/ZeroMomentCoefficients.lean, line 14.

`Komorebi.PaperIV.zeroResolvent_analyticAt_zero`
theorem; KomorebiPaperIV/ZeroMomentCoefficients.lean, line 26.

`Komorebi.PaperIV.reducedZeroPowerSum_convolution`
theorem; KomorebiPaperIV/ZeroMomentCoefficients.lean, line 43.

`Komorebi.PaperIV.zeroMomentSeries`
definition; KomorebiPaperIV/ZeroMomentCoefficients.lean, line 90.

`Komorebi.PaperIV.reducedDenominator_mul_zeroMomentSeries`
theorem; KomorebiPaperIV/ZeroMomentCoefficients.lean, line 93.

`Komorebi.PaperIV.kappaCoeff_eq_zeroPowerSum`
theorem; `KomorebiPaperIV/ZeroMomentCoefficients.lean`, line 103.

`Komorebi.PaperIV.reducedZeroPowerSum_newton`
theorem; `KomorebiPaperIV/ZeroMomentCoefficients.lean`, line 114.

11.106 Row 106: 12.9 Tangent coefficients from zero sums

Exact row scope. Proves the precise factorial-normalized coefficient formula and its convergent Taylor realization as the actual analytic tangent on every zero-free disk in the reduced variable. Fully encodes Corollary 12.9.

Manuscript labels. `cor:tangent_zero_sums`

`Komorebi.PaperIV.transferTangentCoeff_zero_sum`
theorem; `KomorebiPaperIV/ZeroMomentCoefficients.lean`, line 135.

`Komorebi.PaperIV.analyticTransferTangent_coefficient_hasSum`
theorem; `KomorebiPaperIV/ZeroMomentLogarithm.lean`, line 110.

11.107 Row 107: 12.10 Reciprocal Bell formula from zero data

Exact row scope. The named reciprocal coefficient family equals the complete exponential Bell polynomial in the actual zero moments, for all n including zero and every complex parameter r. Fully encodes Corollary 12.10; earlier general-domain complex-power bridges retain their own boundaries.

Manuscript labels. `cor:reciprocal_zero_data`

`Komorebi.PaperIV.bellInput_zero_sums`
theorem; `KomorebiPaperIV/ZeroMomentCoefficients.lean`, line 140.

`Komorebi.PaperIV.reciprocalCoeff_zero_sum_bell`
theorem; `KomorebiPaperIV/ZeroMomentCoefficients.lean`, line 150.

11.108 Row 108: 12.8 Theta Bell formula from zero data

Exact row scope. Uses formal ODE uniqueness and the proved kappa/zero-moment identity to derive the exact theta coefficient formula, including the alternating sign and u^(Nm)+u factor. Fully encodes Proposition 12.8 for the named theta coefficient family.

Manuscript labels. `prop:theta_bell`

`Komorebi.PaperIV.powerSeries_linear_ode_unique`
theorem; `KomorebiPaperIV/ThetaZeroBell.lean`, line 10.

`Komorebi.PaperIV.thetaZeroBellInput`
definition; `KomorebiPaperIV/ThetaZeroBell.lean`, line 29.

`Komorebi.PaperIV.thetaZeroBellInput_derivative`
theorem; `KomorebiPaperIV/ThetaZeroBell.lean`, line 33.

`Komorebi.PaperIV.thetaSeries_eq_zero_bell_exp`
theorem; `KomorebiPaperIV/ThetaZeroBell.lean`, line 44.

`Komorebi.PaperIV.thetaCoeff_zero_sum_bell`
theorem; `KomorebiPaperIV/ThetaZeroBell.lean`, line 60.

11.109 Row 109: 12.6 Normalized analytic logarithm series

Exact row scope. Constructs the logarithm branch normalized at zero on every zero-free disk, proves exp(-L)=Delta, and proves its convergent zero-moment series throughout that disk using Cauchy estimates. Together with rows 104–105 this fully encodes Theorem 12.6. Also proves the convergent resolvent Taylor series.

Manuscript labels. `thm:newton_sums`

`Komorebi.PaperIV.differentiableOn_ball_taylor_hasSum`
theorem; `KomorebiPaperIV/ZeroMomentLogarithm.lean`, line 14.

`Komorebi.PaperIV.reducedZeroPowerSum_logarithm`
theorem; `KomorebiPaperIV/ZeroMomentLogarithm.lean`, line 34.

`Komorebi.PaperIV.reducedZeroPowerSum_resolvent_hasSum`
theorem; `KomorebiPaperIV/ZeroMomentLogarithm.lean`, line 87.

11.110 Row 110: 11.2 Dominant exponential on a vertex ray

Exact row scope. Proves finite exponential dominance with an explicit positive projection gap and specializes to the actual canonical signed denominator on a vertex-facing ray. No dominance or lower-growth certificate is assumed.

Manuscript labels. `prop:orders`

`Komorebi.PaperIV.finite_positive_gap`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 14.

`Komorebi.PaperIV.finite_exponential_dominant_bound`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 21.

`Komorebi.PaperIV.finite_exponential_dominant_tendsto`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 43.

`Komorebi.PaperIV.unit_complex_re_lt_one`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 59.

`Komorebi.PaperIV.canonicalSignedVertex_injective`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 71.

`Komorebi.PaperIV.canonicalSignedVertex_mul_star_self`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 78.

`Komorebi.PaperIV.canonicalSignedDenominator_dominant_ray`
theorem; `KomorebiPaperIV/DominantExponential.lean`, line 84.

11.111 Row 111: 11.2 Sharp growth order and exponential type

Exact row scope. Defines the closed-disk maximum modulus and the standard logarithmic order/type limsups. Proves two-sided maximum-modulus bounds, actual limits for both logarithmic growth ratios, order one and type one for D_N, and order 1/N for Delta_N. Fully encodes Proposition 11.2; imported standard fact 11.1 is not separately formalized or assumed by this proof.

Manuscript labels. `prop:orders`

`Komorebi.PaperIV.diskMaximumModulus`

definition; KomorebiPaperIV/SharpGrowth.lean, line 15.

Komorebi.PaperIV.entireGrowthOrder
definition; KomorebiPaperIV/SharpGrowth.lean, line 19.

Komorebi.PaperIV.entireExponentialType
definition; KomorebiPaperIV/SharpGrowth.lean, line 23.

Komorebi.PaperIV.norm_le_diskMaximumModulus
theorem; KomorebiPaperIV/SharpGrowth.lean, line 26.

Komorebi.PaperIV.diskMaximumModulus_le
theorem; KomorebiPaperIV/SharpGrowth.lean, line 31.

Komorebi.PaperIV.log_growth_ratio_of_exp_bounds
theorem; KomorebiPaperIV/SharpGrowth.lean, line 39.

Komorebi.PaperIV.order_ratio_of_log_growth
theorem; KomorebiPaperIV/SharpGrowth.lean, line 60.

Komorebi.PaperIV.canonicalSignedDenominator_ray_lower
theorem; KomorebiPaperIV/SharpGrowth.lean, line 77.

Komorebi.PaperIV.canonicalSignedDenominator_maximum_bounds
theorem; KomorebiPaperIV/SharpGrowth.lean, line 100.

Komorebi.PaperIV.canonicalSignedDenominator_type_limit
theorem; KomorebiPaperIV/SharpGrowth.lean, line 113.

Komorebi.PaperIV.canonicalSignedDenominator_order_limit
theorem; KomorebiPaperIV/SharpGrowth.lean, line 119.

Komorebi.PaperIV.canonicalSignedDenominator_order_one
theorem; KomorebiPaperIV/SharpGrowth.lean, line 124.

Komorebi.PaperIV.canonicalSignedDenominator_type_one
theorem; KomorebiPaperIV/SharpGrowth.lean, line 128.

Komorebi.PaperIV.entireReducedDenominator_maximum_bounds
theorem; KomorebiPaperIV/SharpGrowth.lean, line 133.

Komorebi.PaperIV.entireReducedDenominator_growth_limit
theorem; KomorebiPaperIV/SharpGrowth.lean, line 159.

Komorebi.PaperIV.entireReducedDenominator_order_limit
theorem; KomorebiPaperIV/SharpGrowth.lean, line 164.

Komorebi.PaperIV.entireReducedDenominator_order_exact
theorem; KomorebiPaperIV/SharpGrowth.lean, line 170.

11.112 Row 112: 11.4 Indicator polygon and tie geometry

Exact row scope. Identifies the actual support function with the manuscript maximum of cosines, proves continuity and nonnegativity for N>=2, and derives the exact periodic tie directions and strict maximizing vertex off those directions.

Manuscript labels. `thm:indicator_polygon`

Komorebi.PaperIV.rayDirection
definition; KomorebiPaperIV/IndicatorGeometry.lean, line 13.

`Komorebi.PaperIV.norm_rayDirection`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 15.

`Komorebi.PaperIV.signedProjection`
definition; `KomorebiPaperIV/IndicatorGeometry.lean`, line 18.

`Komorebi.PaperIV.signedIndicator`
definition; `KomorebiPaperIV/IndicatorGeometry.lean`, line 21.

`Komorebi.PaperIV.signedProjection_cos`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 24.

`Komorebi.PaperIV.signedIndicator_cos`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 32.

`Komorebi.PaperIV.continuous_signedProjection`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 40.

`Komorebi.PaperIV.continuous_signedIndicator`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 44.

`Komorebi.PaperIV.signedProjection_le_indicator`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 47.

`Komorebi.PaperIV.canonicalSignedVertex_sum_zero`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 51.

`Komorebi.PaperIV.signedIndicator_nonneg`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 56.

`Komorebi.PaperIV.canonicalSignedVertex_pow`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 65.

`Komorebi.PaperIV.signed_root_in_vertices`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 70.

`Komorebi.PaperIV.unit_complex_same_real`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 83.

`Komorebi.PaperIV.signed_maximizer_strict_of_not_tie`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 95.

`Komorebi.PaperIV.signedTieDirection`
definition; `KomorebiPaperIV/IndicatorGeometry.lean`, line 131.

`Komorebi.PaperIV.signedTieDirection_iff`
theorem; `KomorebiPaperIV/IndicatorGeometry.lean`, line 134.

11.113 Row 113: 11.4 Uniform dominance and zero-free sectors

Exact row scope. Derives a uniform positive projection gap on every compact set avoiding the actual tie directions. Proves the normalized dominant-exponential error bound (N-1) exp(-delta r) and eventual zero-freeness there, with one uniform delta and radius. Supplies every dominant-sector clause of Theorem 11.4.

Manuscript labels. `thm:indicator_polygon`

`Komorebi.PaperIV.range_erase_nonempty`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 12.

`Komorebi.PaperIV.vertexProjectionGap`
definition; `KomorebiPaperIV/UniformDominantSectors.lean`, line 18.

`Komorebi.PaperIV.directionSeparation`
definition; `KomorebiPaperIV/UniformDominantSectors.lean`, line 22.

`Komorebi.PaperIV.continuous_vertexProjectionGap`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 26.

`Komorebi.PaperIV.continuous_directionSeparation`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 30.

`Komorebi.PaperIV.directionSeparation_pos`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 34.

`Komorebi.PaperIV.compact_directions_uniform_gap`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 49.

`Komorebi.PaperIV.canonicalSignedDenominator_dominant_error`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 65.

`Komorebi.PaperIV.canonicalSignedDenominator_compact_dominant_sectors`
theorem; `KomorebiPaperIV/UniformDominantSectors.lean`, line 90.

11.114 Row 114: 11.4 Aligned radii in tie directions

Exact row scope. Constructs explicit radii tending to infinity on which the leading conjugate phases align; lower-projection terms decay and the normalized finite sum has a positive limit. This proves the indicator lower bound even at tie directions, without a phase-alignment certificate.

Manuscript labels. `thm:indicator_polygon`

`Komorebi.PaperIV.alignedRadii`
definition; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 14.

`Komorebi.PaperIV.alignedRadii_tendsto`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 17.

`Komorebi.PaperIV.alignedRadii_phase`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 26.

`Komorebi.PaperIV.alignedRadii_equal_projection`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 47.

`Komorebi.PaperIV.alignedRadii_lower_projection`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 65.

`Komorebi.PaperIV.finite_unit_exponential_aligned_limit`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 79.

`Komorebi.PaperIV.log_norm_slope_of_normalized_limit`
theorem; `KomorebiPaperIV/AlignedExponentialRadii.lean`, line 99.

11.115 Row 115: 11.4 Exact indicator in every direction

Exact row scope. Proves the exact indicator limsup for every angle and every N>=2. Includes the classical formulation along radii where the denominator is nonzero, so the theorem does not depend on Lean Real.log zero. Together with rows 112–114 this fully encodes Theorem 11.4, including uniform compact-sector estimates and eventual zero-freeness.

Manuscript labels. `thm:indicator_polygon`

`Komorebi.PaperIV.canonicalSignedDenominator_indicator_upper`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 14.

`Komorebi.PaperIV.canonicalSignedDenominator_log_indicator_upper`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 38.

`Komorebi.PaperIV.canonicalSignedDenominator_real_normalization`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 50.

`Komorebi.PaperIV.canonicalSignedDenominator_indicator_sequence`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 67.

`Komorebi.PaperIV.limsup_eq_of_upper_and_sequence`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 95.

`Komorebi.PaperIV.canonicalSignedDenominator_indicator`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 104.

`Komorebi.PaperIV.canonicalSignedDenominator_indicator_nonzero_radii`
theorem; `KomorebiPaperIV/IndicatorLimit.lean`, line 113.

11.116 Row 116: 12.2 DiagonalNuclear

Exact row scope. Actual diagonal operator on l2, norm-summable rank-one expansion, Hilbert nuclear/trace-class characterization, and exact basis and power actions.

Manuscript labels. `thm:trace_class_model`

`Komorebi.PaperIV.IsHilbertTraceClass`
definition; `KomorebiPaperIV/DiagonalNuclear.lean`, line 15.

`Komorebi.PaperIV.diagonalHilbertBasis`
definition; `KomorebiPaperIV/DiagonalNuclear.lean`, line 20.

`Komorebi.PaperIV.diagonalHilbertBasis_inner`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 23.

`Komorebi.PaperIV.diagonalHilbertBasis_norm`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 27.

`Komorebi.PaperIV.diagonalHilbertBasis_apply`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 30.

`Komorebi.PaperIV.diagonalRankOne`
definition; `KomorebiPaperIV/DiagonalNuclear.lean`, line 35.

`Komorebi.PaperIV.diagonalRankOne_norm`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 39.

`Komorebi.PaperIV.diagonalRankOne_summable`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 42.

`Komorebi.PaperIV.diagonalOperator`
definition; `KomorebiPaperIV/DiagonalNuclear.lean`, line 47.

`Komorebi.PaperIV.diagonalOperator_traceClass`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 50.

`Komorebi.PaperIV.diagonalOperator_apply`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 56.

`Komorebi.PaperIV.diagonalOperator_apply_basis`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 69.

`Komorebi.PaperIV.diagonalOperator_pow_apply`
theorem; `KomorebiPaperIV/DiagonalNuclear.lean`, line 76.

11.117 Row 117: 12.2 DiagonalTrace

Exact row scope. Parseval and absolutely convergent double sums prove the operator trace in every orthonormal Hilbert basis, including powers.

Manuscript labels. `thm:trace_class_model`

`Komorebi.PaperIV.hilbertBasis_hasSum_squared_coordinates`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 13.

`Komorebi.PaperIV.diagonal_trace_double_summable`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 19.

`Komorebi.PaperIV.hilbertBasisTrace`
definition; `KomorebiPaperIV/DiagonalTrace.lean`, line 36.

`Komorebi.PaperIV.diagonalOperator_trace_in_any_basis`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 39.

`Komorebi.PaperIV.diagonalOperator_trace_eq`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 73.

`Komorebi.PaperIV.diagonalOperator_pow_eq`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 78.

`Komorebi.PaperIV.diagonalOperator_pow_trace`
theorem; `KomorebiPaperIV/DiagonalTrace.lean`, line 84.

11.118 Row 118: 12.2 DiagonalFredholm

Exact row scope. Actual finite-compression matrix determinants converge locally uniformly to the reduced denominator; the zero-indexed operator is trace class and every positive power has trace kappa. Specialized diagonal Fredholm theory, with no general operator determinant assumed.

Manuscript labels. `thm:trace_class_model`

`Komorebi.PaperIV.diagonalCompression`
definition; `KomorebiPaperIV/DiagonalFredholm.lean`, line 14.

`Komorebi.PaperIV.diagonalCompression_eq_diagonal`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 17.

`Komorebi.PaperIV.diagonalCompression_determinant`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 28.

`Komorebi.PaperIV.diagonalFredholmDeterminant`
definition; `KomorebiPaperIV/DiagonalFredholm.lean`, line 40.

`Komorebi.PaperIV.diagonalFredholmDeterminant_compression_limit`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 42.

`Komorebi.PaperIV.reducedZeroEigenvalue`
definition; `KomorebiPaperIV/DiagonalFredholm.lean`, line 50.

`Komorebi.PaperIV.reducedZeroEigenvalue_norm_summable`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 53.

`Komorebi.PaperIV.reducedZeroOperator`
definition; `KomorebiPaperIV/DiagonalFredholm.lean`, line 58.

`Komorebi.PaperIV.reducedZeroOperator_traceClass`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 60.

`Komorebi.PaperIV.reducedZeroOperator_basis`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 64.

`Komorebi.PaperIV.reducedZeroOperator_fredholm`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 70.

`Komorebi.PaperIV.reducedZeroOperator_trace_power`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 77.

`Komorebi.PaperIV.reducedZeroOperator_finite_determinants_uniform`
theorem; `KomorebiPaperIV/DiagonalFredholm.lean`, line 87.

11.119 Row 119: 12.3 CompleteHomogeneous

Exact row scope. Finite complete homogeneous functions are actual monomial sums indexed by weak compositions; their generating function and Newton recurrence are proved independently of reciprocal coefficients.

Manuscript labels. `cor:complete_symmetric_form`

`Komorebi.PaperIV.powerSumLog`
definition; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 11.

`Komorebi.PaperIV.powerSumLog_constantCoeff`
theorem; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 14.

`Komorebi.PaperIV.powerSumLog_derivative_coeff`
theorem; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 17.

`Komorebi.PaperIV.homogeneousSeries`
definition; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 22.

`Komorebi.PaperIV.homogeneousCoeff`
definition; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 24.

`Komorebi.PaperIV.homogeneousSeries_constantCoeff`
theorem; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 26.

`Komorebi.PaperIV.homogeneousSeries_derivative`
theorem; `KomorebiPaperIV/CompleteHomogeneous.lean`, line 29.

`Komorebi.PaperIV.homogeneousCoeff_recurrence`

theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 34.

Komorebi.PaperIV.homogeneousGeometric
definition; KomorebiPaperIV/CompleteHomogeneous.lean, line 42.

Komorebi.PaperIV.homogeneousGeometric_coeff
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 44.

Komorebi.PaperIV.homogeneousGeometric_constantCoeff
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 47.

Komorebi.PaperIV.homogeneousGeometric_inverse
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 50.

Komorebi.PaperIV.homogeneousGeometric_derivative
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 55.

Komorebi.PaperIV.finiteHomogeneousSeries
definition; KomorebiPaperIV/CompleteHomogeneous.lean, line 72.

Komorebi.PaperIV.finiteHomogeneousSeries_constantCoeff
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 75.

Komorebi.PaperIV.finiteHomogeneousSeries_derivative
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 78.

Komorebi.PaperIV.finiteHomogeneousSeries_eq_homogeneous
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 88.

Komorebi.PaperIV.finiteHomogeneousSeries_monomials
theorem; KomorebiPaperIV/CompleteHomogeneous.lean, line 100.

11.120 Row 120: 12.3 InfiniteSymmetric

Exact row scope. Norm-summable eigenvalues give coefficientwise limits of the finite monomial sums. Repeating eigenvalues r times gives the exact positive-integer reciprocal coefficient; the complex-parameter Bell continuation is proved.

Manuscript labels. `cor:complete_symmetric_form`

Komorebi.PaperIV.summable_norm_positive_powers
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 13.

Komorebi.PaperIV.homogeneousCoeff_zero
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 21.

Komorebi.PaperIV.homogeneousCoeff_succ
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 24.

Komorebi.PaperIV.homogeneousCoeff_continuous
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 30.

Komorebi.PaperIV.homogeneousSeries_congr_positive
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 42.

Komorebi.PaperIV.completeHomogeneous
definition; KomorebiPaperIV/InfiniteSymmetric.lean, line 52.

Komorebi.PaperIV.finiteHomogeneousSeries_coefficient_limit
theorem; KomorebiPaperIV/InfiniteSymmetric.lean, line 55.

`Komorebi.PaperIV.completeHomogeneous_monomial_limit`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 77.

`Komorebi.PaperIV.repeatedEigenvalue`
definition; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 83.

`Komorebi.PaperIV.repeatedEigenvalue_norm_summable`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 85.

`Komorebi.PaperIV.repeatedEigenvalue_powerSum`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 91.

`Komorebi.PaperIV.reciprocalSeries_eq_homogeneous_zero_sums`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 98.

`Komorebi.PaperIV.reciprocalCoeff_completeHomogeneous`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 111.

`Komorebi.PaperIV.reciprocalCoeff_symmetric_polynomial_continuation`
theorem; `KomorebiPaperIV/InfiniteSymmetric.lean`, line 131.

11.121 Row 121: 12.4 NormalizedEntire

Exact row scope. The literal normalized factorial series is entire and equals the residue transfer section after multiplication by z^j. Actual Taylor derivatives and value one at zero are proved.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.entireNormalizedSection`
definition; `KomorebiPaperIV/NormalizedEntire.lean`, line 13.

`Komorebi.PaperIV.normalizedSection_coeff_norm`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 16.

`Komorebi.PaperIV.normalizedSection_entire_radius`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 22.

`Komorebi.PaperIV.analyticAt_entireNormalizedSection`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 38.

`Komorebi.PaperIV.hasSum_entireNormalizedSection`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 43.

`Komorebi.PaperIV.entireNormalizedSection_zero`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 51.

`Komorebi.PaperIV.entireNormalizedSection_iteratedDeriv_zero`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 54.

`Komorebi.PaperIV.entireNormalizedSection_transfer`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 67.

`Komorebi.PaperIV.entireTransfer_unit_norm_le`
theorem; `KomorebiPaperIV/NormalizedEntire.lean`, line 93.

11.122 Row 122: 12.4 JensenZeroGrowth

Exact row scope. Jensen bounds the actual multiplicity-weighted zero counts of a normalized entire function by its disk growth.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.finite_sum_le_finsum_nonneg`
theorem; `KomorebiPaperIV/JensenZeroGrowth.lean`, line 12.

`Komorebi.PaperIV.entire_divisor_eq_orderNat`
theorem; `KomorebiPaperIV/JensenZeroGrowth.lean`, line 22.

`Komorebi.PaperIV.entire_zero_finite_weight_growth`
theorem; `KomorebiPaperIV/JensenZeroGrowth.lean`, line 31.

`Komorebi.PaperIV.entire_zero_subtype_weight_growth`
theorem; `KomorebiPaperIV/JensenZeroGrowth.lean`, line 62.

`Komorebi.PaperIV.entire_zero_bounded_finite`
theorem; `KomorebiPaperIV/JensenZeroGrowth.lean`, line 74.

11.123 Row 123: 12.4 WeightedGrowthSummability

Exact row scope. Multiplicity-weighted counting growth implies the exact subcritical power-sum summability bounds, by finite shell estimates.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.weighted_growth_summable_large`
theorem; `KomorebiPaperIV/WeightedGrowthSummability.lean`, line 14.

`Komorebi.PaperIV.weighted_growth_summable`
theorem; `KomorebiPaperIV/WeightedGrowthSummability.lean`, line 61.

`Komorebi.PaperIV.entire_growth_zeroPowerSummable`
theorem; `KomorebiPaperIV/WeightedGrowthSummability.lean`, line 79.

11.124 Row 124: 12.4 SublinearHadamard

Exact row scope. General normalized entire functions with sublinear power growth equal the complete genus-zero product over their actual zeros with multiplicity. All summability and quotient-rigidity prerequisites are derived.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.entire_sublinear_growth_hadamard`
theorem; `KomorebiPaperIV/SublinearHadamard.lean`, line 15.

11.125 Row 125: 12.4 NormalizedGrowth

Exact row scope. Global factorial/exponential bounds for the actual normalized sections imply every zero-power summability exponent above 1/N.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.entireNormalizedSection_norm_le`
theorem; `KomorebiPaperIV/NormalizedGrowth.lean`, line 13.

`Komorebi.PaperIV.normalizedSectionGrowthConstant`
definition; `KomorebiPaperIV/NormalizedGrowth.lean`, line 39.

`Komorebi.PaperIV.normalizedSectionGrowthConstant_pos`
theorem; `KomorebiPaperIV/NormalizedGrowth.lean`, line 41.

`Komorebi.PaperIV.entireNormalizedSection_disk_growth`
theorem; `KomorebiPaperIV/NormalizedGrowth.lean`, line 47.

`Komorebi.PaperIV.entireNormalizedSection_zeroPowerSummable`
theorem; `KomorebiPaperIV/NormalizedGrowth.lean`, line 61.

11.126 Row 126: 12.4 NormalizedOrder

Exact row scope. The finite exponential representation proves the dominant positive-real-ray limit for each transfer residue section.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.entireTransfer_positive_ray`
theorem; `KomorebiPaperIV/NormalizedOrder.lean`, line 14.

11.127 Row 127: 12.4 NormalizedExactOrder

Exact row scope. Matching upper and polynomially corrected lower bounds prove the exact order 1/N, including N=1. The residue condition is j<N.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.log_growth_ratio_of_polynomial_exp_bounds`
theorem; `KomorebiPaperIV/NormalizedExactOrder.lean`, line 13.

`Komorebi.PaperIV.entireTransfer_positive_ray_lower`
theorem; `KomorebiPaperIV/NormalizedExactOrder.lean`, line 49.

`Komorebi.PaperIV.entireNormalizedSection_maximum_bounds`
theorem; `KomorebiPaperIV/NormalizedExactOrder.lean`, line 66.

`Komorebi.PaperIV.entireNormalizedSection_growth_limit`
theorem; `KomorebiPaperIV/NormalizedExactOrder.lean`, line 103.

`Komorebi.PaperIV.entireNormalizedSection_order_exact`
theorem; `KomorebiPaperIV/NormalizedExactOrder.lean`, line 110.

11.128 Row 128: 12.4 NormalizedProduct

Exact row scope. For N>=2, each normalized section equals its full multiplicity-indexed product, uniformly on compact sets. The actual analytic Taylor series is the named formal series.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.entireNormalizedSection_hadamard`
theorem; `KomorebiPaperIV/NormalizedProduct.lean`, line 16.

`Komorebi.PaperIV.entireNormalizedSection_hadamard_uniform`
theorem; `KomorebiPaperIV/NormalizedProduct.lean`, line 26.

`Komorebi.PaperIV.analyticTaylorSeries_entireNormalizedSection`
theorem; `KomorebiPaperIV/NormalizedProduct.lean`, line 35.

11.129 Row 129: 12.4 NormalizedZeroBell

Exact row scope. The logarithmic coefficients equal the actual zero power sums, giving the exact (Nn)!/n! Bell normalization for the named residue coefficients and every complex parameter.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.normalized_zeroMomentSeries`
theorem; `KomorebiPaperIV/NormalizedZeroBell.lean`, line 11.

`Komorebi.PaperIV.normalized_zeroPowerSum`
theorem; `KomorebiPaperIV/NormalizedZeroBell.lean`, line 29.

`Komorebi.PaperIV.bellInput_normalized_zeros`
theorem; `KomorebiPaperIV/NormalizedZeroBell.lean`, line 35.

`Komorebi.PaperIV.residuePowerCoeff_zero_sum_bell`
theorem; `KomorebiPaperIV/NormalizedZeroBell.lean`, line 52.

11.130 Row 130: 12.4 NormalizedAnalyticPowers

Exact row scope. Actual principal-logarithm powers near zero have the named residue formal Taylor series and exact zero-moment Bell coefficients for every complex parameter. This is a germ at zero; global branch domains remain an earlier mapping boundary.

Manuscript labels. `prop:normalized_section_product`

`Komorebi.PaperIV.powerSeries_eq_of_derivative_constant`
theorem; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 13.

`Komorebi.PaperIV.analyticTaylorSeries_log`
theorem; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 23.

`Komorebi.PaperIV.analyticTaylorSeries_exp`
theorem; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 46.

`Komorebi.PaperIV.normalizedAnalyticPower`
definition; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 63.

`Komorebi.PaperIV.normalizedAnalyticPower_taylor`
theorem; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 66.

`Komorebi.PaperIV.normalizedAnalyticPower_zero_sum_bell`
theorem; `KomorebiPaperIV/NormalizedAnalyticPowers.lean`, line 82.

11.131 Row 131: 13.1 AnalyticTaylorAlgebra

Exact row scope. Actual iterated-derivative Taylor coefficients preserve analytic germ equality, sums, products, natural powers and inverse products.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.analyticTaylorSeries`
definition; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 14.

`Komorebi.PaperIV.analyticTaylorSeries_coeff`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 16.

`Komorebi.PaperIV.analyticTaylorSeries_constantCoeff`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 20.

`Komorebi.PaperIV.analyticTaylorSeries_congr`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 23.

`Komorebi.PaperIV.analyticTaylorSeries_mul`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 28.

`Komorebi.PaperIV.analyticTaylorSeries_add`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 46.

`Komorebi.PaperIV.analyticTaylorSeries_const`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 51.

`Komorebi.PaperIV.analyticTaylorSeries_pow`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 58.

`Komorebi.PaperIV.analyticTaylorSeries_inverse_product`
theorem; `KomorebiPaperIV/AnalyticTaylorAlgebra.lean`, line 64.

11.132 Row 132: 12.4 and 13.1 AnalyticTaylorDifferential

Exact row scope. Actual Taylor coefficients respect differentiation and negation. Analytic Hadamard products yield the formal logarithmic zero-moment identity.

Manuscript labels. `prop:normalized_section_product`, `thm:nearest_circle_expansion`

`Komorebi.PaperIV.analyticTaylorSeries_neg`
theorem; `KomorebiPaperIV/AnalyticTaylorDifferential.lean`, line 13.

`Komorebi.PaperIV.analyticTaylorSeries_deriv`
theorem; `KomorebiPaperIV/AnalyticTaylorDifferential.lean`, line 18.

`Komorebi.PaperIV.analyticTaylorSeries_zeroResolvent`
theorem; `KomorebiPaperIV/AnalyticTaylorDifferential.lean`, line 29.

`Komorebi.PaperIV.hadamard_taylor_zeroMoment_identity`
theorem; `KomorebiPaperIV/AnalyticTaylorDifferential.lean`, line 38.

11.133 Row 133: 13.1 PoleCoefficientBounds

Exact row scope. Cauchy's estimate supplies explicit Taylor coefficient bounds, Big-O bounds and exact factorial/block rescaling.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.analyticTaylorCoefficient`
definition; `KomorebiPaperIV/PoleCoefficientBounds.lean`, line 13.

`Komorebi.PaperIV.analyticTaylorCoefficient_norm_le`
theorem; `KomorebiPaperIV/PoleCoefficientBounds.lean`, line 17.

`Komorebi.PaperIV.analyticTaylorCoefficient_isBigO`
theorem; `KomorebiPaperIV/PoleCoefficientBounds.lean`, line 28.

`Komorebi.PaperIV.analyticTaylorCoefficient_block_error`
theorem; `KomorebiPaperIV/PoleCoefficientBounds.lean`, line 45.

11.134 Row 134: 13.1 and 13.2 PrincipalPartCoefficients

Exact row scope. The normalized pole has the exact binomial Taylor coefficients. Lower pole orders satisfy a uniform polynomial coefficient bound.

Manuscript labels. `cor:leading_dominant_zero`, `thm:nearest_circle_expansion`

`Komorebi.PaperIV.principalPartSeries`
definition; `KomorebiPaperIV/PrincipalPartCoefficients.lean`, line 13.

`Komorebi.PaperIV.principalPartSeries_coefficient`
theorem; `KomorebiPaperIV/PrincipalPartCoefficients.lean`, line 16.

`Komorebi.PaperIV.principalPartSeries_inverse`
theorem; `KomorebiPaperIV/PrincipalPartCoefficients.lean`, line 25.

`Komorebi.PaperIV.lower_principal_part_choose_bound`
theorem; `KomorebiPaperIV/PrincipalPartCoefficients.lean`, line 33.

11.135 Row 135: 13.1 PoleRemoval

Exact row scope. The manuscript's local O(1) principal-part conditions imply nonnegative meromorphic order. The constructed normal-form remainder is holomorphic throughout the required zero-free disk after all finite principal parts are removed.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.poleTerm`
definition; `KomorebiPaperIV/PoleRemoval.lean`, line 14.

`Komorebi.PaperIV.polePrincipalPart`
definition; `KomorebiPaperIV/PoleRemoval.lean`, line 16.

`Komorebi.PaperIV.rawPoleRemainder`
definition; `KomorebiPaperIV/PoleRemoval.lean`, line 19.

`Komorebi.PaperIV.poleTerm_analyticAt`
theorem; `KomorebiPaperIV/PoleRemoval.lean`, line 22.

`Komorebi.PaperIV.poleTerm_meromorphicAt`

theorem; KomorebiPaperIV/PoleRemoval.lean, line 31.

Komorebi.PaperIV.polePrincipalPart_analyticAt
theorem; KomorebiPaperIV/PoleRemoval.lean, line 35.

Komorebi.PaperIV.rawPoleRemainder_meromorphic
theorem; KomorebiPaperIV/PoleRemoval.lean, line 39.

Komorebi.PaperIV.rawPoleRemainder_analyticAt
theorem; KomorebiPaperIV/PoleRemoval.lean, line 47.

Komorebi.PaperIV.meromorphicOrderAt_nonneg_of_bounded
theorem; KomorebiPaperIV/PoleRemoval.lean, line 56.

Komorebi.PaperIV.rawPoleRemainder_bounded_at_pole
theorem; KomorebiPaperIV/PoleRemoval.lean, line 66.

Komorebi.PaperIV.removedPoleRemainder
definition; KomorebiPaperIV/PoleRemoval.lean, line 87.

Komorebi.PaperIV.removedPoleRemainder_analyticOnNhd
theorem; KomorebiPaperIV/PoleRemoval.lean, line 90.

Komorebi.PaperIV.removedPoleRemainder_eq_raw
theorem; KomorebiPaperIV/PoleRemoval.lean, line 105.

11.136 Row 136: 13.1 PoleTaylor

Exact row scope. Actual analytic principal parts have the stated binomial coefficients. Integer reciprocal powers of the signed denominator have exactly the manuscript's factorial-scaled P coefficients.

Manuscript labels. `thm:nearest_circle_expansion`

Komorebi.PaperIV.analyticTaylorSeries_sub
theorem; KomorebiPaperIV/PoleTaylor.lean, line 13.

Komorebi.PaperIV.analyticTaylorSeries_finsetSum
theorem; KomorebiPaperIV/PoleTaylor.lean, line 18.

Komorebi.PaperIV.analyticTaylorSeries_id
theorem; KomorebiPaperIV/PoleTaylor.lean, line 29.

Komorebi.PaperIV.analyticTaylorSeries_poleTerm
theorem; KomorebiPaperIV/PoleTaylor.lean, line 37.

Komorebi.PaperIV.analyticTaylorCoefficient_principalPart
theorem; KomorebiPaperIV/PoleTaylor.lean, line 54.

Komorebi.PaperIV.analyticTaylorSeries_canonicalSignedDenominator
theorem; KomorebiPaperIV/PoleTaylor.lean, line 70.

Komorebi.PaperIV.reciprocalSeries_nat
theorem; KomorebiPaperIV/PoleTaylor.lean, line 79.

Komorebi.PaperIV.analyticTaylorSeries_reciprocal_signed
theorem; KomorebiPaperIV/PoleTaylor.lean, line 84.

Komorebi.PaperIV.reciprocalCoeff_analyticTaylor
theorem; KomorebiPaperIV/PoleTaylor.lean, line 102.

11.137 Row 137: 13.1 SimplePoleParts

Exact row scope. Analytic Taylor division constructs the full principal part at every simple zero and proves its leading Laurent coefficient and bounded remainder; no principal-part existence certificate is assumed.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.poleNumerator`
definition; `KomorebiPaperIV/SimplePoleParts.lean`, line 13.

`Komorebi.PaperIV.canonicalPoleCoefficient`
definition; `KomorebiPaperIV/SimplePoleParts.lean`, line 16.

`Komorebi.PaperIV.poleNumerator_analyticAt`
theorem; `KomorebiPaperIV/SimplePoleParts.lean`, line 19.

`Komorebi.PaperIV.canonicalPoleCoefficient_leading`
theorem; `KomorebiPaperIV/SimplePoleParts.lean`, line 31.

`Komorebi.PaperIV.poleNumerator_identity`
theorem; `KomorebiPaperIV/SimplePoleParts.lean`, line 35.

`Komorebi.PaperIV.polePrincipalPart_scaled`
theorem; `KomorebiPaperIV/SimplePoleParts.lean`, line 50.

`Komorebi.PaperIV.canonicalPoleCoefficient_local_bound`
theorem; `KomorebiPaperIV/SimplePoleParts.lean`, line 73.

11.138 Row 138: 13.1 PrincipalPartUniqueness

Exact row scope. Every principal part satisfying the local O(1) condition has leading coefficient (-rho D'(rho))^(-r), independently of the chosen construction.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.polePrincipalPart_scaled_general`
theorem; `KomorebiPaperIV/PrincipalPartUniqueness.lean`, line 12.

`Komorebi.PaperIV.polePrincipalPart_scaled_limit`
theorem; `KomorebiPaperIV/PrincipalPartUniqueness.lean`, line 25.

`Komorebi.PaperIV.principalPart_leading_coefficient`
theorem; `KomorebiPaperIV/PrincipalPartUniqueness.lean`, line 50.

11.139 Row 139: 13.1 NearestPoleExpansion

Exact row scope. The actual reciprocal coefficients satisfy the complete nearest-pole expansion with an explicit Cauchy constant for every coefficient. Its hypotheses state the prescribed local principal parts and the absence of other zeros inside the next modulus.

Manuscript labels. `thm:nearest_circle_expansion`

`Komorebi.PaperIV.removedPoleRemainder_taylor`
theorem; `KomorebiPaperIV/NearestPoleExpansion.lean`, line 12.

`Komorebi.PaperIV.nearest_pole_taylor_bound`
theorem; `KomorebiPaperIV/NearestPoleExpansion.lean`, line 41.

`Komorebi.PaperIV.reciprocalCoeff_nearest_pole_bound`
theorem; `KomorebiPaperIV/NearestPoleExpansion.lean`, line 62.

11.140 Row 140: 13.2 LeadingPoleBounds

Exact row scope. The leading term has separately controlled polynomial/radial and Cauchy errors, with actual simple-zero principal parts constructed automatically. For r=1 the first error constant is exactly zero.

Manuscript labels. `cor:leading_dominant_zero`

`Komorebi.PaperIV.principalCoefficientSum`
definition; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 13.

`Komorebi.PaperIV.leadingPoleCoefficient`
definition; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 16.

`Komorebi.PaperIV.lowerPrincipalConstant`
definition; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 19.

`Komorebi.PaperIV.lowerPrincipalConstant_nonneg`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 22.

`Komorebi.PaperIV.lowerPrincipalConstant_one`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 25.

`Komorebi.PaperIV.principalCoefficientSum_split`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 28.

`Komorebi.PaperIV.principalCoefficientSum_leading_error`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 41.

`Komorebi.PaperIV.principalCoefficientSum_one`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 67.

`Komorebi.PaperIV.reciprocalCoeff_leading_bound`
theorem; `KomorebiPaperIV/LeadingPoleBounds.lean`, line 72.

11.141 Row 141: 13.3 OrbitPoleReduction

Exact row scope. Derivative rotation and block exponent invariance give N identical contributions per orbit. An equivalence records the chosen representatives and all distinct rotations; the resulting main term retains both proved error bounds.

Manuscript labels. `cor:orbit_reduced_asymptotic`

`Komorebi.PaperIV.leadingZeroWeight`
definition; `KomorebiPaperIV/OrbitPoleReduction.lean`, line 12.

`Komorebi.PaperIV.leadingZeroWeight_rotation`
theorem; `KomorebiPaperIV/OrbitPoleReduction.lean`, line 15.

`Komorebi.PaperIV.leadingPoleCoefficient_eq_zero_weights`
theorem; `KomorebiPaperIV/OrbitPoleReduction.lean`, line 27.

`Komorebi.PaperIV.leading_zero_sum_orbit_reduction`
theorem; `KomorebiPaperIV/OrbitPoleReduction.lean`, line 34.

`Komorebi.PaperIV.reciprocalCoeff_orbit_leading_bound`
theorem; `KomorebiPaperIV/OrbitPoleReduction.lean`, line 55.

11.142 Row 142: 13.4 SecantAsymptotic

Exact row scope. The actual cosine zeros and derivatives give the quantitative coefficient error, exponentially vanishing relative error, ratio limit one, and asymptotic equivalence to 2(2n)!(2/pi)^(2n+1).

Manuscript labels. `cor:secant_asymptotic`

`Komorebi.PaperIV.cosineNearestZeros`
definition; `KomorebiPaperIV/SecantAsymptotic.lean`, line 13.

`Komorebi.PaperIV.canonicalSignedDenominator_two_cos`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 15.

`Komorebi.PaperIV.cosineNearestZeros_norm`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 19.

`Komorebi.PaperIV.cosineNearestZeros_root`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 23.

`Komorebi.PaperIV.cosineNearestZeros_simple`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 28.

`Komorebi.PaperIV.cosine_nearest_zero_exhaustion`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 33.

`Komorebi.PaperIV.secant_leading_coefficient`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 49.

`Komorebi.PaperIV.secant_coefficient_error`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 72.

`Komorebi.PaperIV.secantMainTerm`
definition; `KomorebiPaperIV/SecantAsymptotic.lean`, line 87.

`Komorebi.PaperIV.secantMainTerm_norm`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 90.

`Komorebi.PaperIV.secantMainTerm_ne_zero`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 95.

`Komorebi.PaperIV.secant_relative_error`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 100.

`Komorebi.PaperIV.secant_coefficient_ratio_limit`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 120.

`Komorebi.PaperIV.secant_coefficient_asymptotic`
theorem; `KomorebiPaperIV/SecantAsymptotic.lean`, line 131.

11.143 Row 143: 3.1, 3.2, 3.3 PointwiseFilters

Exact row scope. Actual pointwise finite averages, coefficient-series HasSum, orthogonal projector resolution, holomorphic range/covariance equivalence on invariant domains, and the complete cyclic product law.

Manuscript labels. `prop:covariance`, `thm:cyclic_product`, `thm:root_filter`

`Komorebi.PaperIV.pointwiseRootAverage`
definition; `KomorebiPaperIV/PointwiseFilters.lean`, line 14.

`Komorebi.PaperIV.root_weight_sum`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 17.

`Komorebi.PaperIV.mapsTo_root_pow`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 43.

`Komorebi.PaperIV.pointwiseRootAverage_covariant`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 52.

`Komorebi.PaperIV.pointwiseRootAverage_of_covariant`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 82.

`Komorebi.PaperIV.pointwiseRootAverage_projector`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 99.

`Komorebi.PaperIV.pointwiseRootAverage_resolution`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 106.

`Komorebi.PaperIV.differentiableOn_pointwiseRootAverage`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 128.

`Komorebi.PaperIV.pointwiseRootAverage_range_iff`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 138.

`Komorebi.PaperIV.pointwiseRootAverage_sum`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 152.

`Komorebi.PaperIV.pointwiseRootAverage_product`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 160.

`Komorebi.PaperIV.pointwiseRootAverage_hasSum`
theorem; `KomorebiPaperIV/PointwiseFilters.lean`, line 185.

11.144 Row 144: 4.4, 5.5 PointwiseAddition

Exact row scope. Actual entire transfer and signed addition formulas for arbitrary complex arguments; the signed carry factor is proved from rho^N=-1.

Manuscript labels. `cor:exponential_addition`, `prop:signed_addition`

`Komorebi.PaperIV.pointwiseRootAverage_scaled_exp`
theorem; `KomorebiPaperIV/PointwiseAddition.lean`, line 10.

`Komorebi.PaperIV.entireTransfer_addition`
theorem; `KomorebiPaperIV/PointwiseAddition.lean`, line 20.

`Komorebi.PaperIV.signed_addition_weight`
theorem; `KomorebiPaperIV/PointwiseAddition.lean`, line 38.

`Komorebi.PaperIV.entireSigned_addition`
theorem; `KomorebiPaperIV/PointwiseAddition.lean`, line 49.

11.145 Row 145: 7.4, 8.4, 8.5 AnalyticReciprocalLaws

Exact row scope. Parameter addition and reciprocal differentiation on every open domain with a single holomorphic logarithm; the actual differential Bell/Riccati identity at every point where D_N is nonzero. Formal coefficient clauses are covered by the earlier rows.

Manuscript labels. `cor:reciprocal_derivative`, `thm:higher_riccati`, `thm:reciprocal_parameter_addition`

`Komorebi.PaperIV.IsLogarithmOn`
definition; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 14.

`Komorebi.PaperIV.logarithmPower`
definition; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 18.

`Komorebi.PaperIV.logarithmPower_add`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 21.

`Komorebi.PaperIV.IsLogarithmOn.nonzero`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 28.

`Komorebi.PaperIV.IsLogarithmOn.deriv`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 33.

`Komorebi.PaperIV.hasDerivAt_logarithmPower`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 45.

`Komorebi.PaperIV.reciprocal_fixed_log_derivative`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 56.

`Komorebi.PaperIV.analyticDifferentialBell`
definition; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 65.

`Komorebi.PaperIV.analyticOnNhd_differentialBell`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 69.

`Komorebi.PaperIV.iteratedDeriv_eq_analyticDifferentialBell`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 75.

`Komorebi.PaperIV.analytic_higher_riccati`
theorem; `KomorebiPaperIV/AnalyticReciprocalLaws.lean`, line 93.

11.146 Row 146: 9.9 OperatorEncodingBridge

Exact row scope. The derivative-truncation and factorial/Hasse encodings are proved equal on every polynomial. The named residue Appell polynomial has the exact forward action and the Phi(partial_x^N)^r inverse, for arbitrary parameter and degree.

Manuscript labels. `prop:appell_operator_form`

`Komorebi.PaperIV.truncatedDifferentialOperator_apply`
theorem; `KomorebiPaperIV/OperatorEncodingBridge.lean`, line 12.

`Komorebi.PaperIV.truncatedDifferentialOperator_egfAction`
theorem; `KomorebiPaperIV/OperatorEncodingBridge.lean`, line 23.

`Komorebi.PaperIV.residueAppell_operator_forward`
theorem; `KomorebiPaperIV/OperatorEncodingBridge.lean`, line 37.

`Komorebi.PaperIV.residueAppell_operator_inverse`
theorem; `KomorebiPaperIV/OperatorEncodingBridge.lean`, line 48.

11.147 Row 147: 8.2, 8.3 TangentBoundary

Exact row scope. D_1=exp(-z), T_1=1, and all order-one tangent coefficients are identified. Together with the prior N>=2 theorem, the original-variable tangent series converges with the exact factorial normalization for every positive order.

Manuscript labels. `prop:tangent_kappa`, `thm:tangent_recurrence`

`Komorebi.PaperIV.canonicalSignedDenominator_one`
theorem; `KomorebiPaperIV/TangentBoundary.lean`, line 11.

`Komorebi.PaperIV.analyticTransferTangent_one`
theorem; `KomorebiPaperIV/TangentBoundary.lean`, line 19.

`Komorebi.PaperIV.transferTangentCoeff_one_order`
theorem; `KomorebiPaperIV/TangentBoundary.lean`, line 29.

`Komorebi.PaperIV.analyticTransferTangent_one_hasSum`
theorem; `KomorebiPaperIV/TangentBoundary.lean`, line 47.

`Komorebi.PaperIV.analyticTransferTangent_hasSum_all_orders`
theorem; `KomorebiPaperIV/TangentBoundary.lean`, line 58.

11.148 Row 148: 7.7 CompactReciprocal

Exact row scope. The second and third reciprocal polynomials equal the literal compact binomial/-factorial formulas in the manuscript, over every characteristic-zero field. The existing first-coefficient theorem completes all three displayed polynomials.

Manuscript labels. `cor:first_reciprocal_polynomials`

`Komorebi.PaperIV.reciprocalCoeff_two_compact`
theorem; `KomorebiPaperIV/CompactReciprocal.lean`, line 32.

`Komorebi.PaperIV.reciprocalCoeff_three_compact`
theorem; `KomorebiPaperIV/CompactReciprocal.lean`, line 44.

11.149 Row 149: 7.6 NewtonDeterminant

Exact row scope. The explicit lower Hessenberg matrix has the displayed p_k entries and negative integer superdiagonal. Its determinant equals n! b_n from Newton's recurrence, including n=0; specialization gives the literal reciprocal determinant formula.

Manuscript labels. `thm:logarithmic_newton_recurrence`

`Komorebi.PaperIV.newtonHessenberg`
definition; `KomorebiPaperIV/NewtonDeterminant.lean`, line 15.

`Komorebi.PaperIV.newtonHessenberg_det`
theorem; `KomorebiPaperIV/NewtonDeterminant.lean`, line 59.

`Komorebi.PaperIV.reciprocalOrdCoeff_newton_determinant`
theorem; `KomorebiPaperIV/NewtonDeterminant.lean`, line 112.

11.150 Row 150: 4.5 HypergeometricAnalytic

Exact row scope. The literal N-1 denominator parameters exclude the unique parameter 1 and have positive real parts. Gamma normalization identifies the ordinary hypergeometric coefficients with the exact factorial coefficients. The everywhere convergent Mathlib regularized series yields the pointwise manuscript formula, including N=1.

Manuscript labels. `thm:hypergeometric_form`

`Komorebi.PaperIV.residueHypergeometricParameters`
definition; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 14.

`Komorebi.PaperIV.residueHypergeometricParameters_card`
theorem; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 17.

`Komorebi.PaperIV.residueHypergeometricParameters_re_pos`
theorem; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 23.

`Komorebi.PaperIV.regularized_zeroF_normalization`
theorem; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 50.

`Komorebi.PaperIV.residueHypergeometric`
definition; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 71.

`Komorebi.PaperIV.residueHypergeometric_hasSum`
theorem; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 75.

`Komorebi.PaperIV.entireTransfer_hypergeometric`
theorem; `KomorebiPaperIV/HypergeometricAnalytic.lean`, line 99.

11.151 Row 151: 6.2 QuotientExponential

Exact row scope. The quotient algebras receive their finite-dimensional coefficient norm topology. The ordinary exponential power series converges at every complex argument to the displayed entire-section coordinates in both x^N=1 and x^N=-1 algebras. The already checked derivative systems and addition laws retain their exact normalizations.

Manuscript labels. `thm:finite_algebra_exponentials`

`Komorebi.PaperIV.finiteAlgebraExponential`
definition; `KomorebiPaperIV/QuotientExponential.lean`, line 13.

`Komorebi.PaperIV.hasSum_cyclicAlgebra_exponential`
theorem; `KomorebiPaperIV/QuotientExponential.lean`, line 22.

`Komorebi.PaperIV.hasSum_signedAlgebra_exponential`
theorem; `KomorebiPaperIV/QuotientExponential.lean`, line 45.

`Komorebi.PaperIV.cyclicAlgebraNormedAddCommGroup`
definition; `KomorebiPaperIV/QuotientExponential.lean`, line 70.

`Komorebi.PaperIV.cyclicAlgebraNormedSpace`
definition; `KomorebiPaperIV/QuotientExponential.lean`, line 76.

`Komorebi.PaperIV.signedAlgebraNormedAddCommGroup`
definition; `KomorebiPaperIV/QuotientExponential.lean`, line 80.

`Komorebi.PaperIV.signedAlgebraNormedSpace`
definition; `KomorebiPaperIV/QuotientExponential.lean`, line 86.

`Komorebi.PaperIV.cyclicAlgebra_pointwise_exponential`
theorem; `KomorebiPaperIV/QuotientExponential.lean`, line 91.

`Komorebi.PaperIV.signedAlgebra_pointwise_exponential`
theorem; `KomorebiPaperIV/QuotientExponential.lean`, line 99.

13 Navigation in the mathematical text

The canonical manuscript record numbers remain visible in the ledger and row catalog. The following labels link the formal contract to statement and section numbers in the Chapter IV mathematical text. These locators are editorial navigation, not additional theorem evidence.

Source label	Chapter IV locator
`thm:root_filter`	Theorem 3.1
`prop:covariance`	Proposition 3.2
`thm:cyclic_product`	Theorem 3.3
`def:transfer_exponential`	Definition 4.1
`thm:exponential_cycle`	Theorem 4.2
`cor:wronskians`	Corollary 4.3
`cor:exponential_addition`	Corollary 4.4
`thm:hypergeometric_form`	Theorem 4.5
`def:signed_sections`	Definition 5.1
`prop:signed_exponential_relation`	Proposition 5.2
`thm:signed_cycle`	Theorem 5.3
`cor:signed_wronskian`	Corollary 5.4
`prop:signed_addition`	Proposition 5.5
`thm:finite_exponential_representation`	Theorem 5.6
`def:reduced_denominator`	Definition 5.7
`cor:denominator_symmetries`	Corollary 5.8
`def:cyclic_algebras`	Definition 6.1
`thm:finite_algebra_exponentials`	Theorem 6.2
`def:signed_shift_matrix`	Definition 6.3
`thm:companion_matrix_realization`	Theorem 6.4
`cor:fundamental_matrix_entries`	Corollary 6.5
`cor:companion_addition_inversion`	Corollary 6.6
`def:reciprocal_family`	Definition 7.1
`thm:reciprocal_bell`	Theorem 7.2
`thm:parameter_recurrence`	Theorem 7.3
`thm:reciprocal_parameter_addition`	Theorem 7.4
`cor:binomial_type`	Corollary 7.5
`thm:logarithmic_newton_recurrence`	Theorem 7.6
`cor:first_reciprocal_polynomials`	Corollary 7.7
`cor:factor_r`	Corollary 7.8
`def:transfer_tangent`	Definition 8.1

Source label	Chapter IV locator
`prop:tangent_kappa`	Proposition 8.2
`thm:tangent_recurrence`	Theorem 8.3
`cor:reciprocal_derivative`	Corollary 8.4
`thm:higher_riccati`	Theorem 8.5
`def:normalized_section`	Definition 9.1
`thm:triangular_inverse`	Theorem 9.2
`def:residue_section_power`	Definition 9.3
`thm:residue_section_parameter_addition`	Theorem 9.4
`def:block_appell`	Definition 9.5
`thm:appell_laws`	Theorem 9.6
`prop:explicit_appell`	Proposition 9.7
`thm:appell_two_parameter_convolution`	Theorem 9.8
`prop:appell_operator_form`	Proposition 9.9
`cor:appell_symmetry`	Corollary 9.10
`def:unsigned_denominator`	Definition 10.1
`thm:theta_reconstruction`	Theorem 10.2
`prop:theta_factors`	Proposition 10.3
`prop:theta_newton_recurrence`	Proposition 10.4
`prop:orders`	Proposition 11.2
`thm:indicator_polygon`	Theorem 11.4
`lem:two_term_dominance`	Lemma 11.5
`lem:near_cancellation`	Lemma 11.6
`thm:large_zero_strings`	Theorem 11.7
`thm:first_zero_correction`	Theorem 11.8
`cor:N3_zero_correction`	Corollary 11.9
`cor:case_two_zeros`	Corollary 11.10
`cor:eventual_simplicity`	Corollary 11.11
`cor:reduced_zeros`	Corollary 11.13
`thm:zero_counting`	Theorem 11.14
`cor:exponent_of_convergence`	Corollary 11.15
`thm:genus_zero_product`	Theorem 12.1
`thm:trace_class_model`	Theorem 12.2
`cor:complete_symmetric_form`	Corollary 12.3
`prop:normalized_section_product`	Proposition 12.4
`thm:newton_sums`	Theorem 12.6

Source label	Chapter IV locator
`cor:mittag_leffler_tangent`	Corollary 12.7
`prop:theta_bell`	Proposition 12.8
`cor:tangent_zero_sums`	Corollary 12.9
`cor:reciprocal_zero_data`	Corollary 12.10
`thm:nearest_circle_expansion`	Theorem 13.1
`cor:leading_dominant_zero`	Corollary 13.2
`cor:orbit_reduced_asymptotic`	Corollary 13.3
`cor:secant_asymptotic`	Corollary 13.4
`thm:zeta_convergence`	Theorem 14.2
`thm:zeta_continuation`	Theorem 14.3
`cor:exact_zeta_models`	Corollary 14.4
`cor:branch_grouped_continuation`	Corollary 14.6
`thm:derivative_weighted_convergence`	Theorem 14.8

FORMAL COMPANION V

Multiresolution Signal Calculus

The formal companion to Chapter V explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 1937

Return to Chapter V

How to use this volume

A guide to the actual formal development. 20 scoped manuscript results. 225 theorem targets. 23 proof modules. The eight tutorial chapters explain the route from literal samples to Hilbert operators, finite filters, packet synthesis, adaptive trees, and projective actions. The complete ledgers then give exact declarations and manuscript locators. The mathematical text also contains 63 broader prose proof records. Their separate inventory is a reading aid and adds no formal coverage. The two edge recovery conjectures remain open. This guide preserves 52 exact Lean excerpts and reconciles source bytes, declaration locations, dependency pins, and the saved axiom audit. Kernel execution evidence comes from the recorded formal-core build. Its toolchain, source identities and replay instructions appear in the reproduction section. This reading guide does not claim a new kernel execution.

1 How to read the formal companion

Chapter V places several kinds of mathematics in one signal processing language: coordinate partitions, bounded Hilbert space operators, finite filter banks, finite tree optimization, formal series, and projective actions. The formal development keeps their types and hypotheses distinct. Following those distinctions is the most useful way to read its proofs.

A pointwise partition identity needs little structure. An energy theorem needs a normed space with the intended norm. A transform theorem needs convergence. An optimal finite pruning needs a finite family and an objective. A fractional linear expression needs a chart, while its projective action survives at the chart's poles. Each transition has a named construction or theorem in the source.

> **Follow the model before the theorem name**
>
> Start with the ambient type, then locate the bridge to the intended signal model. A coordinate identity, a bounded operator theorem, and an analytic transform law have different inputs. The exact excerpt shows where those inputs enter the declaration.

1.1 What the completed contract contains

The formal contract covers twenty scoped theorem-like manuscript results with explicit proofs and 225 supporting theorem targets. The saved namespace audit reports 827 declarations. These are different denominators: 827 includes definitions and compiler-generated declarations, while 225 includes supporting results rather than 225 independent new mathematical claims.

The formal core is the expanded manuscript preserved as Part I of the mathematical text. Part II preserves the broader Annals development. Its additional proof records are prose mathematics. They do not enter the twenty-result formalization contract. Section 10 gives a separate crosswalk.

The two edge recovery conjectures remain open. Known-edge cancellation is established, measurable-mask perturbation is established, and finite adaptive selection is established within its hypotheses. Those results do not provide an estimator with the full statistical and approximation guarantees requested by the conjectures.

1.2 Verification evidence

Evidence	Meaning
Source reconciliation	Current hashes, 225 declaration locations, 225 audit directives, and 225 inherited axiom-output records agree.
Recorded Lean execution	The saved compiler record reports the pinned build and namespace audit passing. The raw logs are preserved.
Manual correspondence	Manuscript statements have a reviewed, explicit relationship to formal declarations. Lean does not parse or certify the English PDF automatically.
Exact model tests	Finite independent models exercise indexing, masks, trees, lifting, filters, and boundary cases. They are computational evidence on their tested ranges.

Evidence	Meaning
Companion numerics	Signal experiments and figure data are empirical demonstrations. Their floating-point implementation is outside the Lean theorem claim.

The guide build performs no Lean compilation. Reproduction commands appear in Section 8, with the missing-tool case treated as a failure to establish a fresh kernel result. The final package's verification records distinguish current reruns from inherited evidence.

1.3 Read a theorem as a typed contract

The name after `theorem` is a reusable declaration. Parenthesized arguments are explicit inputs. Braced arguments are generally inferred. A hypothesis such as `(hN : 0 < N)` is an actual input to the theorem. A typeclass argument such as `[AddCommMonoid E]` specifies the algebraic laws available to its proof.

The expression after the colon is the proposition. The proof after `:= by` constructs evidence of that proposition. Common commands here have small, specific jobs: `ext` reduces equality to coordinates, `rw` rewrites using an equality, `simp` uses declared simplification rules, `ring` normalizes polynomial identities, `omega` handles integer arithmetic, and `linarith` combines linear inequalities. Their successful use belongs to the recorded compilation, not to a textual count of tactic names.

A definition marked `noncomputable` may use classical choice or real-number constructions that do not give executable numerical code. Its theorems can still be checked by the kernel. In particular, the pruning evaluator over exact real costs specifies a mathematical algorithm and a count of unit operations. Lean evaluation speed is outside that count.

1.4 Four kinds of signal space

Object	Lean representation	What it provides
Arbitrary sequence	`ℤ → K`	Every integer sample exists. No norm or convergence is assumed.
Finite filter	`ℤ →₀ K`	A finitely supported coefficient function. Convolution sums are finite.
Square-summable signal	`lp (fun _ : I => ℂ) 2`	The joint ℓ^2 norm, infinite norm-convergent expansions, bounded maps and adjoints.
Formal series	`R⟦X⟧`	Coefficients with formal addition and Cauchy multiplication. Analytic convergence is not part of the type.

The same informal word "packet" appears in all four settings. The guide always identifies the ambient type before using an identity across them.

1.5 Recommended reading routes

For the signal processing route, read integer addresses, the Hilbert bridge, FIR transfer, faithful filter banks, and Haar packets. For the adaptive route, read known edges, mask perturbations, complete prunings, the dynamic program, and its stability envelope. For the algebraic route, read formal cyclic products, homogeneous Smith actions, and CRT masks.

Every displayed code excerpt is copied from an immutable source declaration. The raw excerpt, its source line range, and its SHA256 hash accompany the PDF. Excerpts rely on the imports, namespaces, variables, and definitions in their source files. They are not advertised as independently compilable snippets. The complete project remains the authoritative executable object.

2 Partitions, signed addresses, and the Hilbert bridge

2.1 A packet first means a coordinate mask

Take an index set I, a label set R, and a phase map $\phi : I \to R$. The packet with label r keeps coordinate i exactly when $\phi(i) = r$:

$$(P_r x)(i) = \begin{cases} x(i), & \phi(i) = r, \\ 0, & \phi(i) \neq r. \end{cases}$$

The definition requires a zero in the value type and a decidable label equality. It does not require a Fourier transform, a measure, or an inner product.

Source: `KomorebiPaperV/Packets.lean`.

```
def packet (phase : I → R) (r : R) (x : I → V) : I → V :=
  fun i => if phase i = r then x i else 0
```

Applying the same mask twice changes nothing. Distinct labels cannot both be retained at the same coordinate. If the label type is finite and values can be added commutatively, the finite sum of packets returns the original value. At any one coordinate exactly one summand survives.

Source: `KomorebiPaperV/Packets.lean`.

```
theorem packet_complete (phase : I → R) (x : I → V) (i : I) :
    (∑ r : R, packet phase r x i) = x i := by
  classical
  calc
    (∑ r : R, packet phase r x i) = packet phase (phase i) x i := by
      apply Finset.sum_eq_single (phase i)
      · intro b _ hb
        simp [packet, Ne.symm hb]
      · simp
    _ = x i := by simp [packet]
```

The proof uses `Finset.sum_eq_single` to isolate the surviving label. This simple mechanism is reused after stronger ambient structures are introduced.

2.2 Signed polyphase coordinates are an equivalence

For $N > 0$, every integer has the unique decomposition

$$n = Nq + r, \qquad 0 \le r < N, \quad q \in \mathbb{Z}.$$

The integer quotient is Euclidean: for $N = 3$ and $n = -1$, the address is $(r, q) = (2, -1)$, not a negative residue. Lean represents the bounded residue by `Fin N`. Its stored value has a proof that it is smaller than N.

Source: `KomorebiPaperV/IntegerPolyphase.lean`.

```
def addressEquiv (N : ℕ) (hN : 0 < N) : ℤ ≃ Fin N × ℤ where
  toFun n := (phase N hN n, quotient N n)
  invFun p := address N p.1 p.2
  left_inv := address_phase_quotient N hN
  right_inv p := by
    apply Prod.ext
    · exact phase_address N hN p.1 p.2
    · exact quotient_address N hN p.1 p.2
```

The two inverse laws are mathematical content. They ensure both that analyzing then synthesizing recovers a signal and that synthesizing then analyzing recovers an arbitrary channel family. Merely checking the first direction would not establish that all channel coordinates are reachable.

Writing $x_r[q] = x[Nq + r]$, the pointwise maps are

$$Ax(r, q) = x[Nq + r], \qquad Sc(n) = c(n \bmod N, \lfloor n/N \rfloor).$$

These are meaningful for any value type. Energy is a later statement.

2.3 The two units of delay

A delay by d original samples is $(D_d x)[n] = x[n - d]$. A delay by d inside each channel is an original delay by Nd. For a general original delay, the residue changes and a signed carry enters the channel coordinate:

$$(AD_d x)_r[q] = x_{(r-d) \bmod N}\left[q + \left\lfloor \frac{r - d}{N} \right\rfloor\right].$$

Source: `KomorebiPaperV/IntegerPolyphase.lean`.

```
theorem sample_delay (N : ℕ) (hN : 0 < N) (d : ℤ) (x : ℤ → V)
    (r : Fin N) (q : ℤ) :
    analysis N (delay d x) r q =
      analysis N x (phase N hN ((r.val : ℤ) - d))
        (q + quotient N ((r.val : ℤ) - d)) := by
  dsimp [analysis, delay]
  congr 1
  have h := address_phase_quotient N hN ((r.val : ℤ) - d)
  dsimp [address] at h ⊢
  linear_combination -h
```

For $N = 2$ and a one-sample delay, the new even channel is the old odd channel delayed once, while the new odd channel is the old even channel at the same channel index:

$$(AD_1x)_0[q] = x_1[q-1], \qquad (AD_1x)_1[q] = x_0[q].$$

For a two-sample delay both channels are delayed once. This is why a filter-bank identity with channel delay d reconstructs $x[n - Nd]$.

2.4 Square summability is built into the type

Source: KomorebiPaperV/HilbertPackets.lean.

```
abbrev Signal (I : Type*) : Type _ := lp (fun _ : I => ℂ) 2
```

The bounded projector is obtained by first proving that masking cannot increase the norm. The proof works coordinate by coordinate and invokes monotonicity of the ℓ^2 norm.

Source: KomorebiPaperV/HilbertPackets.lean.

```
theorem projection_norm_le [DecidableEq R] (phase : I → R) (r : R)
    (x : Signal I) : ‖projection phase r x‖ ≤ ‖x‖ := by
  apply lp.norm_mono (by norm_num)
  intro i
  by_cases h : phase i = r <;> simp [projection_apply, h]
```

The same mask is made into a linear map and then promoted to a continuous linear map with bound one.

Source: KomorebiPaperV/HilbertPackets.lean.

```
noncomputable def projector [DecidableEq R] (phase : I → R) (r : R) :
    Signal I →L[ℂ] Signal I :=
  (projectionLinear phase r).mkContinuous 1 (by
    intro x
    change ‖projection phase r x‖ ≤ 1 * ‖x‖
    simpa using projection_norm_le phase r x)
```

The notation `→L[ℂ]` records continuous complex linearity. Idempotence alone would not establish boundedness, symmetry, or orthogonality. The file proves those separately. With finitely many labels, the sum is the identity as a continuous linear operator, not just a formula on finite test vectors.

2.5 The channel norm is the joint norm

The Hilbert space analysis target is $\ell^2(\mathrm{Fin}\, N \times \mathbb{Z})$:

$$\|Ax\|^2 = \sum_{r=0}^{N-1} \sum_{q \in \mathbb{Z}} |x[Nq + r]|^2 = \|x\|^2.$$

It is not an unqualified function space of channels carrying a supremum norm. The source packages the address equivalence into a complex linear isometric equivalence.

Source: KomorebiPaperV/HilbertPackets.lean.

```
noncomputable def polyphase (N : ℕ) (hN : 0 < N) :
    Signal ℤ ≃ₗᵢ[ℂ] Signal (Fin N × ℤ) :=
  reindexIsometry (KomorebiPaperV.IntegerPolyphase.addressEquiv N hN)
```

Its inverse laws are inherited from that equivalence, and its energy law follows from norm preservation. The proof of reindexing uses summability over the bijective address map. This is the actual bridge from arbitrary pointwise coordinates to the Hilbert statement.

2.6 Impulse expansion means norm convergence

The impulse at i with amplitude x_i is represented by `lp.single 2 i (x i)`. The full theorem is a `HasSum` statement in the ℓ^2 space.

Source: KomorebiPaperV/HilbertPackets.lean.

```
theorem impulse_expansion (x : Signal ℤ) :
    HasSum (fun i : ℤ => lp.single 2 i (x i)) x :=
  lp.hasSum_single (by norm_num) x
```

Thus the series of impulses converges to x in the ambient norm. The finite support version is an exact finite sum. The shift is an isometric equivalence with formula $S_d x[n] = x[n-d]$, and shifting the impulse at zero gives the impulse at d. Density of the impulse basis yields the dense shift-generated span.

2.7 Refinement uses divisibility

If $N \mid M$, reducing modulo M and then modulo N agrees with reducing directly modulo N. Consequently one coarse residue projector is the sum of the fine residue projectors lying above it.

Source: KomorebiPaperV/HilbertPackets.lean.

```
theorem residue_projector_refinement (N M : ℕ) (hN : 0 < N) (hM : 0 < M)
    (hNM : N ∣ M) (r : Fin N) :
    projector (IntegerPolyphase.phase N hN) r =
      ∑ s : Fin M, if IntegerPolyphase.phase N hN (s.val : ℤ) = r
        then projector (IntegerPolyphase.phase M hM) s else 0 := by
  have hc : (fun j : ℤ => IntegerPolyphase.phase N hN
      ((IntegerPolyphase.phase M hM j).val : ℤ)) = IntegerPolyphase.phase N hN := by
    funext j
    apply Fin.ext
    change (((IntegerPolyphase.phase M hM j).val : ℤ) % (N : ℤ)).toNat =
      (j % (N : ℤ)).toNat
    rw [IntegerPolyphase.phase_val, IntegerPolyphase.emod_coarsen N M hNM]
  have h := projector_refinement (IntegerPolyphase.phase M hM)
    (fun s : Fin M => IntegerPolyphase.phase N hN (s.val : ℤ)) r
  simpa only [Function.comp_def, hc] using h
```

For example, the even packet modulo 2 is the sum of packets 0, 2, and 4 modulo 6. The theorem is an equality of bounded operators on the full complex ℓ^2 space. The hypotheses $N > 0$, $M > 0$, and $N \mid M$ remain explicit. A relation between arbitrary unrelated moduli would require a different construction.

3 FIR transfer and faithful perfect reconstruction

3.1 Finite support makes time domain transfer unconditional

An FIR filter is a finitely supported function $h : \mathbb{Z} \to K$. Negative tap addresses are allowed, so the formal theorem includes advances as well as delays. For an arbitrary bilateral sequence,

$$(h * x)[n] = \sum_{t \in \operatorname{supp} h} h[t]x[n-t].$$

No convergence hypothesis is needed because this sum is finite.

Source: `KomorebiPaperV/FIRTransfer.lean`.

```
def fir (h : ℤ →₀ K) (x : ℤ → K) (n : ℤ) : K :=
  ∑ t ∈ h.support, h t * x (n - t)
```

Fix output phase k and input phase s. The tap carrying an input channel delay m is

$$t = Nm + k - s, \qquad h_{k,s}[m] = h[Nm + k - s].$$

The address identity is

$$(Nn + k) - (Nm + k - s) = N(n - m) + s.$$

It says exactly which original tap contributes to which channel and delay. The source's `tapEquiv` proves the relevant reindexing is bijective, then partitions the finite support by input phase. It does not enlarge the finite sum to an arbitrary infinite series.

Source: `KomorebiPaperV/FIRTransfer.lean`.

```
theorem fir_polyphase_transfer (N : ℕ) (hN : 0 < N) (h : ℤ →₀ K)
    (x : ℤ → K) (k : Fin N) (n : ℤ) :
    analysis N (fir h x) k n =
      ∑ s : Fin N, ∑ m ∈ delays N hN h k s,
        kernel N h k s m * analysis N x s (n - m) := by
  change fir h x (address N k n) = _
  rw [fir_reindex N hN h x k n]
  calc
    _ = ∑ s : Fin N, ∑ p ∈ (transferSupport N hN h k).filter (fun p => p.1 = s),
        kernel N h k p.1 p.2 * analysis N x p.1 (n - p.2) :=
      (Finset.sum_fiberwise (transferSupport N hN h k) Prod.fst
        (fun p => kernel N h k p.1 p.2 * analysis N x p.1 (n - p.2))).symm
    _ = _ := by
      apply Finset.sum_congr rfl
      intro s _
      exact sum_pair_fiber (transferSupport N hN h k) s
        (fun r m => kernel N h k r m * analysis N x r (n - m))
```

3.2 A two channel calculation

Three taps and their channel delays

Take $h[0] = a$, $h[1] = b$, $h[2] = c$, with all other taps zero. For $N = 2$ the four channel kernels are

$$y_0[n] = a x_0[n] + c x_0[n-1] + b x_1[n-1],$$
$$y_1[n] = b x_0[n] + a x_1[n] + c x_1[n-1].$$

The tap at one has different channel delays in the two off diagonal positions. In the transform convention used by the chapter, the transfer matrix is

$$H(z) = \begin{pmatrix} a + c z^{-1} & b z^{-1} \\ b & a + c z^{-1} \end{pmatrix}.$$

The time domain equations hold for every sequence. The transform equation $Y(z) = H(z)X(z)$ has the additional analytic hypotheses described next.

3.3 A transform is a convergent weighted sum

The source defines a totalized `tsum` expression, but the meaningful convergence predicate is `HasTransform`.

Source: `KomorebiPaperV/ZTransform.lean`.

```
def HasTransform (x : ℤ → ℂ) (z X : ℂ) : Prop := HasSum (fun n => x n * z ^ (-n)) X
```

It states that the bilateral weighted series $\sum_n x[n] z^{-n}$ has the specified sum. The delay theorem requires both a nonzero transform parameter and this convergence input.

Source: `KomorebiPaperV/ZTransform.lean`.

```
theorem delay_transform (x : ℤ → ℂ) (z X : ℂ) (hz : z ≠ 0)
    (h : HasTransform x z X) (d : ℤ) :
    HasTransform (fun n => x (n-d)) z (z ^ (-d) * X) := by
  have hh := h.mul_left (z ^ (-d))
  have he : (fun n : ℤ => z ^ (-d) * (x n * z ^ (-n))) =
      (fun n : ℤ => x (n-d) * z ^ (-n)) ∘ Equiv.addRight d := by
    funext n
    change z ^ (-d) * (x n * z ^ (-n)) = x (n+d-d) * z ^ (-(n+d))
    rw [add_sub_cancel_right, neg_add, zpow_add₀ hz]
    ring
  rw [he] at hh
  exact (Equiv.addRight d).hasSum_iff.mp hh
```

Finitely many delayed inputs can then be combined using finite sums of `HasSum` proofs. The matrix version assumes weighted summability separately for every input channel.

Source: KomorebiPaperV/ZTransform.lean.

```
theorem fir_matrix_transform (N : ℕ) (hN : 0 < N) (h : ℤ →₀ ℂ) (x : ℤ → ℂ)
    (z : ℂ) (hz : z ≠ 0)
    (hx : ∀ s, Summable (fun n => analysis N x s n * z ^ (-n))) :
    (fun k => transform (analysis N (fir h x) k) z) =
      (transferMatrix N hN h z).mulVec (fun s => transform (analysis N x s) z) := by
  funext k
  exact (fir_channel_transform N hN h x z hz _ (fun s => (hx s).hasSum) k).tsum_eq
```

An arbitrary ℓ^2 signal need not have an absolutely convergent bilateral transform at a chosen complex z. An L^2 Fourier boundary-value interpretation on the unit circle is a separate analytic interpretation. Off the unit circle, the displayed weighted summability conditions must be supplied. Neither the definition of `tsum` nor an FIR input filter removes them.

3.4 Laurent polynomials act as finite filters

The type `AddMonoidAlgebra ℂ ℤ` represents finite Laurent coefficients indexed by integer delays. Its multiplication is finite convolution. A monomial at index d acts by $x[n] \mapsto x[n-d]$, with transform symbol z^{-d}. The formal action is an algebra homomorphism into complex linear endomorphisms of all bilateral sequences.

Source: KomorebiPaperV/FilterBank.lean.

```
noncomputable def act : Laurent →ₐ[ℂ] Module.End ℂ Signal :=
  AddMonoidAlgebra.lift ℂ (Module.End ℂ Signal) ℤ shiftHom
```

The source proves `act_is_fir`, making the abstract algebra homomorphism coincide with the literal finite filter sum. It then defines a matrix bank by summing the action of each matrix entry over the input channels. The theorem `bank_mul` identifies matrix multiplication with composition of banks in the correct order.

3.5 Why the matrix criterion is an equivalence

The reverse direction of a perfect-reconstruction criterion needs faithfulness. If two matrix banks have the same action on every sequence, why must their Laurent coefficients agree? The source feeds a unit impulse into one channel at a time. Reading output channel k at sample n recovers exactly the nth Laurent coefficient of the (k, s) entry.

Source: KomorebiPaperV/FilterBank.lean.

```
theorem bank_faithful (A B : Matrix I I Laurent) :
    (∀ x, bank A x = bank B x) ↔ A = B := by
  constructor
  · intro h
    funext k s
    apply AddMonoidAlgebra.ext
    apply Finsupp.ext
    intro n
    have he := congrArg (fun y : I → Signal => y k n) (h (channelImpulse s))
```

```
    simpa only [bank_impulse] using he
  · rintro rfl
    intro x
    rfl
```

This eliminates a possible gap between an operator statement and a formal matrix identity. It relies on equality on every input, rather than agreement on a bounded collection of numerical examples.

3.6 Gain and delay

The target matrix is diagonal, with each diagonal entry the monomial of amplitude c at delay d. The theorem gives the equivalence between matrix equality and reconstruction of every channel sequence.

Source: KomorebiPaperV/FilterBank.lean.

```
theorem perfect_reconstruction_iff (E R : Matrix I I Laurent) (c : ℂ) (d : ℤ) :
    (∀ x : I → Signal, ∀ k n, bank R (bank E x) k n = c*x k (n-d)) ↔
      R*E = gainDelay c d := by
  rw [← bank_faithful]
  constructor
  · intro h x
    funext k n
    rw [bank_mul, gain_delay_action]
    exact h x k n
  · intro h x k n
    have he := congrArg (fun y : I → Signal => y k n) (h x)
    simpa only [bank_mul, gain_delay_action] using he
```

The formula allows any complex c. To interpret the result as invertible reconstruction up to gain, one additionally requires $c \neq 0$. At $c = 0$, the theorem remains a valid zero-output identity, but it is not an invertibility statement.

Returning from channel coordinates to original coordinates yields the literal sample formula.

Source: KomorebiPaperV/FilterBank.lean.

```
theorem original_sample_delay (N : ℕ) (hN : 0 < N)
    (E R : Matrix (Fin N) (Fin N) Laurent) (c : ℂ) (d : ℤ)
    (h : R*E = gainDelay c d) (x : Signal) (n : ℤ) :
    KomorebiPaperV.IntegerPolyphase.synthesis N hN
      (bank R (bank E (KomorebiPaperV.IntegerPolyphase.analysis N x))) n = c*x (n-(N:ℤ)*d) := by
  have hr := (perfect_reconstruction_iff E R c d).mpr h
  unfold KomorebiPaperV.IntegerPolyphase.synthesis
  rw [hr]
  congr 1
  unfold KomorebiPaperV.IntegerPolyphase.analysis
  congr 1
  have ha := KomorebiPaperV.IntegerPolyphase.address_phase_quotient N hN n
  dsimp [KomorebiPaperV.IntegerPolyphase.address] at ha ⊢
  linarith
```

This result is independent of numerical transform convergence. The transform representation of the bank is a further theorem with `HasTransform` hypotheses. The distinction is practical: an exact finite

reconstruction identity can be checked algebraically even when a particular signal's off-circle transform does not exist.

4 Haar, lifting, and finite depth packet synthesis

4.1 Interleaving is part of the theorem

For a bilateral signal, write $e[n] = x[2n]$ and $o[n] = x[2n+1]$. The source first proves that the squared norms of these two channels sum to the original squared norm. It then forms the normalized Haar channels

$$a[n] = \frac{e[n] + o[n]}{\sqrt{2}}, \qquad d[n] = \frac{e[n] - o[n]}{\sqrt{2}}.$$

The sample formula establishes that the abstract construction really acts on these literal interleaved samples.

Source: KomorebiPaperV/HaarChannels.lean.

```
theorem haar_low_samples (x : Signal ℤ) (n : ℤ) :
    analysisLow x n = ((Real.sqrt 2 : ℝ) : ℂ)⁻¹ * (x (2*n) + x (2*n+1)) := by
  rw [analysisLow, low_apply, even_apply, odd_apply]
```

The energy identity combines the Hilbert space parallelogram law with the joint channel norm.

Source: KomorebiPaperV/HaarChannels.lean.

```
theorem haar_energy (x : Signal ℤ) :
    ‖x‖ ^ 2 = ‖analysisLow x‖ ^ 2 + ‖analysisHigh x‖ ^ 2 := by
  rw [analysisLow, analysisHigh, energy, interleaved_energy]
```

This theorem holds for complex ℓ^2 signals, not only finite real arrays. A finite numerical check can expose normalization mistakes but cannot replace the infinite-space proof.

4.2 Lifting is a different normalization

The lifting analysis sends (e, o) to (s, d) with

$$d = o - e, \qquad s = e + \frac{d}{2} = \frac{e + o}{2}.$$

The inverse is $e = s - d/2$, $o = d + e$. Its determinant is one. The theorem is stated over a characteristic zero field, which makes division by two meaningful in the required way.

Source: KomorebiPaperV/Lifting.lean.

```
theorem lifting_synthesis_analysis (x : K × K) :
    liftingSynthesis (liftingAnalysis x) = x := by
  apply Prod.ext <;> dsimp [liftingAnalysis, liftingSynthesis] <;> ring
```

The lifting pair is invertible but is not the normalized Haar pair. For example, $(e, o) = (2, 4)$ gives lifting coordinates $(s, d) = (3, 2)$, whose squared Euclidean norm is 13, whereas the input norm squared is 20. Normalized Haar gives $(3\sqrt{2}, -\sqrt{2})$ with squared norm 20 under the source's high-channel sign convention. Invertibility and isometry are separate claims.

4.3 Flat finite signals have both inverse laws

The finite transform file builds a pairing equivalence between a flat vector of length $2N$ and N pairs. It transports lifting through that equivalence. The result is a theorem about the actual flattened signal, not an unconnected pairwise calculation.

Source: `KomorebiPaperV/FiniteTransforms.lean`.

```
theorem unlift_lift_flat (n : ℕ) (x : Fin (2 * n) → K) :
    unliftFlat n (liftFlat n x) = x := by
  simp only [unliftFlat, liftFlat, flat_analysis_synthesis,
    KomorebiPaperV.lifting_synthesis_analysis, flat_synthesis_analysis]
```

The companion direction, `lift_unlift_flat`, establishes reconstruction of arbitrary transformed coordinates as well. The finite Haar statements similarly include pair energy, finite energy, and a flat involution. These finite constructions are useful for matching array implementations, while the Hilbert files establish the infinite ℓ^2 statements.

4.4 A packet is an ordered composition

At depth zero there is one empty word and its operator is the identity. A depth-$n+1$ word consists of a first branch and a remaining word. The first branch acts on the signal before the remaining packet.

Source: `KomorebiPaperV/WaveletPackets.lean`.

```
noncomputable def packet : (n : ℕ) → Word n → H →L[ℂ] H
  | 0, _ => ContinuousLinearMap.id ℂ H
  | n+1, (b,w) => (packet n w).comp (childOp b)
```

Here `Word n` is a recursively finite type. Each child map is a bounded continuous linear operator. The low and high norm bounds come from the two channel energy identity, so their adjoints are legitimate Hilbert adjoints.

4.5 Induction preserves total energy

The proof at depth $n+1$ first sums over the remaining word for each child. The induction hypothesis reduces that sum to the child's norm squared. Summing over the two children returns the parent's norm squared.

Source: `KomorebiPaperV/WaveletPackets.lean`.

```
theorem packet_energy (n : ℕ) (x : H) : ∑ w : Word n, ‖packet n w x‖ ^ 2 = ‖x‖ ^ 2 := by
  induction n generalizing x with
  | zero =>
    change (∑ w : Unit, ‖x‖ ^ 2) = _
    simp
  | succ n ih =>
    change (∑ w : Fin 2 × Word n, ‖(packet n w.2) (childOp w.1 x)‖ ^ 2) = _
    rw [Fintype.sum_prod_type]
    simp_rw [ih]
    exact child_energy x
```

For depth two, this is the four-packet identity

$$\|W_{00}x\|^2 + \|W_{01}x\|^2 + \|W_{10}x\|^2 + \|W_{11}x\|^2 = \|x\|^2.$$

The theorem quantifies over every natural depth. Each individual depth has a finite packet family. It is not a theorem about convergence of an infinite depth packet expansion.

4.6 Energy gives an operator reconstruction identity

For finitely many bounded maps W_i on the complex Hilbert space, an energy identity valid for every x implies

$$\sum_i W_i^* W_i = I.$$

The formal proof uses inner product extensionality, the adjoint relation, and complex polarization through the library theorem it invokes.

Source: `KomorebiPaperV/WaveletPackets.lean`.

```
theorem energy_implies_adjoint_identity {I : Type*} [Fintype I]
    (W : I → H →L[ℂ] H) (h : ∀ x, ∑ i, ‖W i x‖ ^ 2 = ‖x‖ ^ 2) :
    (∑ i, (W i).adjoint.comp (W i)) = ContinuousLinearMap.id ℂ H := by
  have he : (∑ i, (W i).adjoint.comp (W i)).toLinearMap = LinearMap.id := by
    apply (ext_inner_map _ _).mp
    intro x
    change inner ℂ ((∑ i, (W i).adjoint.comp (W i)) x) x = inner ℂ x x
    simp only [sum_apply, ContinuousLinearMap.comp_apply, sum_inner,
      ContinuousLinearMap.adjoint_inner_left, inner_self_eq_norm_sq_to_K]
    exact_mod_cast h x
  apply ContinuousLinearMap.ext
  intro x
  exact LinearMap.congr_fun he x
```

This is a genuine adjoint identity. Writing a reconstruction map with a similar formula would not suffice without showing that it is the adjoint of the analysis map.

Applying the resulting operator equality to a signal gives the source level synthesis theorem.

Source: `KomorebiPaperV/WaveletPackets.lean`.

```
theorem packet_reconstruction (n : ℕ) (x : H) :
    (∑ w : Word n, (packet n w).adjoint (packet n w x)) = x := by
  have h := congrArg (fun T : H →L[ℂ] H => T x) (packet_operator_reconstruction n)
  simpa using h
```

The proof does not need every W_i to be invertible. It needs the collective energy identity. The family can divide information among its channels while the sum of adjoint synthesis terms recovers the original signal exactly.

5 Known edges and measurable perturbations

5.1 Exact retention separates the error algebraically

Write $f = s + j$, where s is the background and j is a known jump packet. If the reconstruction is $Qs + j$, the retained jump cancels:

$$f - (Qs + j) = s - Qs.$$

Source: `KomorebiPaperV/KnownEdge.lean`.

```
theorem known_edge_error (s j : E) (Q : E → E) :
    (s + j) - (Q s + j) = s - Q s := by
  abel
```

The cancellation theorem needs only an additive commutative group and an arbitrary function Q. It does not require Q to be linear. This is sufficient for cancellation because Q is applied to s before the same j is restored. A separate claim that commutes Q with addition, or estimates an operator commutator, would need the appropriate linearity hypotheses.

Taking norms yields exact equality of the full reconstruction error with the background error. A finite sum of known packets cancels in the same way. None of these statements estimates an unknown edge or proves that a data driven algorithm finds the correct mask.

5.2 Estimated packets have two errors

For $j = a\mathbf{1}_S$ and $\hat{j} = b\mathbf{1}_T$,

$$j - \hat{j} = (a - b)\mathbf{1}_S + b(\mathbf{1}_S - \mathbf{1}_T).$$

The first term changes the amplitude on the original mask. The second changes the mask while holding the estimated amplitude fixed. The symmetric difference $S \triangle T$ contains exactly the points where the two masks disagree.

For measurable sets and $1 \le p < \infty$, the resulting bound is

$$\|a\mathbf{1}_S - b\mathbf{1}_T\|_p \le |a - b|\mu(S)^{1/p} + |b|\mu(S \triangle T)^{1/p}.$$

The formal theorem lives first in the extended nonnegative reals, so infinite measures are handled without silently converting infinity to an ordinary real number.

Source: `KomorebiPaperV/MaskPerturbation.lean`.

```
theorem finite_p_perturbation (μ : Measure Ω) (S T : Set Ω)
    (hS : MeasurableSet S) (hT : MeasurableSet T) (p : ENNReal)
    (hp : 1 ≤ p) (hpTop : p ≠ ∞) (a b : ℂ) :
    eLpNorm (jump S a-jump T b) p μ ≤
      ‖a-b‖ₑ * μ S ^ (1 / p.toReal) + ‖b‖ₑ * μ (S Δ T) ^ (1 / p.toReal) := by
  have hp0 : p ≠ 0 := ne_of_gt (lt_of_lt_of_le (by norm_num) hp)
  rw [jump_error_decomposition]
  calc
    _ ≤ eLpNorm (jump S (a-b)) p μ + eLpNorm (jump S b-jump T b) p μ :=
      eLpNorm_add_le (jump_measurable S hS (a-b) μ)
        ((jump_measurable S hS b μ).sub (jump_measurable T hT b μ)) hp
```

```
    _ = _ := by
      rw [symmetric_difference_norm μ S T hS hT p hp0 hpTop,
        jump, eLpNorm_indicator_const hS hp0 hpTop]
```

The notation `ENNReal` permits ∞, and `eLpNorm` is an extended seminorm. The hypotheses `hp : 1 ≤ p` and `hpTop : p ≠ ∞` govern the triangle inequality and finite-exponent formula. A finite real valued corollary requires the displayed terms to be finite, for example through finite mask and symmetric-difference measures.

5.3 A concrete perturbation budget

An amplitude and mask budget

Suppose $p = 2$, $\mu(S) = 1$, $\mu(S \triangle T) = 0.01$, $a = 2$, and $b = 1.9$. The bound is

$$0.1\sqrt{1} + 1.9\sqrt{0.01} = 0.29.$$

If the background approximation error is at most 0.04, the full error is at most 0.33 by the additional triangle inequality in `full_signal_perturbation`. The numbers illustrate a deterministic budget conditional on the amplitude, mask, and background estimates. A statistical procedure achieving those estimates requires additional work.

5.4 Pointwise and essential supremum

The source first proves a bound at every point, then transfers it to the essential supremum seminorm.

Source: `KomorebiPaperV/MaskPerturbation.lean`.

```
theorem infinity_perturbation (μ : Measure Ω) (S T : Set Ω) (a b : ℂ) :
    eLpNorm (jump S a-jump T b) ∞ μ ≤
      ENNReal.ofReal (‖a-b‖ + if S Δ T = ∅ then 0 else ‖b‖) := by
  rw [eLpNorm_exponent_top]
  exact eLpNormEssSup_le_of_ae_bound (ae_of_all μ (pointwise_sup_bound S T a b))
```

The displayed bound uses whether $S \triangle T$ is empty as an actual set. An essential-supremum argument may ignore a null set, but a pointwise supremum cannot. This particular theorem remains a valid, sometimes coarse, bound when the sets differ only on a null set. Its statement should not be silently replaced by an assertion that pointwise and almost-everywhere equality coincide.

5.5 The open obligations remain identifiable

The stable edge-packet recovery conjecture asks for specified signal classes, contrast and separation assumptions, and a noise model under which a multiscale estimator controls both mask error and amplitude error at rates separated from smooth approximation. The formal perturbation bound explains how such estimates would enter reconstruction. It does not construct that estimator or prove those rates.

The edge-packet approximation-class conjecture asks for natural function spaces whose norms distinguish smooth approximation, boundary geometry, packet amplitude, and packet identification. The

current declarations do not define and close that proposed theory. Infinite-tree consistency, minimax claims, and optimal Smith sampling also remain outside the twenty-result formal scope.

These boundaries matter when reading Part II. Its broader deterministic and geometric theorems can develop the mathematics without turning every contextual statement into a Lean-certified target. The final traditional record and the separate Part II crosswalk identify that distinction explicitly.

6 Complete prunings, global selection, and stability

6.1 A finite tree carries positive finite branching

The tree type permits a leaf or a branch with $n + 1$ children. Thus every branch has at least one child, and unary branches are included.

Source: KomorebiPaperV/FiniteAdaptive.lean.

```
inductive Tree (A : Type*) where
  | leaf : A → Tree A
  | branch : (a : A) → (n : ℕ) → (Fin (n + 1) → Tree A) → Tree A
```

The tree itself is a finite inductive object. A cut is another typed inductive object: stop at the current root, or choose one cut in each child. The type prevents selecting an arbitrary collection that fails to cover a terminal path.

Source: KomorebiPaperV/FiniteAdaptive.lean.

```
inductive Cut : Tree A → Type _ where
  | stop (t : Tree A) : Cut t
  | split {a : A} {n : ℕ} {children : Fin (n + 1) → Tree A} :
      (∀ i, Cut (children i)) → Cut (.branch a n children)
```

A singleton leaf has only its stop cut. At a unary branch, stopping at the root and descending to its child are distinct cuts even if node labels happen to coincide. Formal nodes carry their positions in the tree, so equal labels do not collapse different nodes.

6.2 Conservation is an explicit hypothesis

For an additive quantity p at each node, conservation says that the parent's value is the sum of the child-root values and that this continues recursively.

Source: KomorebiPaperV/FiniteAdaptive.lean.

```
def conserves [AddCommMonoid E] (p : A → E) : Tree A → Prop
  | .leaf _ => True
  | .branch a _ children =>
      p a = ∑ i, p (label (children i)) ∧ ∀ i, conserves p (children i)
```

By induction on the cut, its sum equals the root value. The quantity may be a scalar, a vector, or an operator. The argument uses addition and the conservation identity. It does not derive conservation from an arbitrary tree topology.

For projector valued packets, one supplies both conservation and a root equal to the identity operator. Every complete pruning then reconstructs the signal. This makes selection compatible with exact synthesis: changing the cut changes the representation while preserving the total operator.

6.3 From recursive cuts to covering antichains

The manuscript describes a complete pruning as a node set that meets each root to terminal path exactly once. The source builds a positional node type, terminal paths, and a membership predicate for the nodes selected by a recursive cut. It proves the equivalence with the set based covering-antichain description.

Source: `KomorebiPaperV/CompletePrunings.lean`.

```
theorem complete_iff_covering_antichain {t : FiniteAdaptive.Tree A} (S : Set (Node t)) :
    Complete S ↔ CoveringAntichain S := by
  constructor
  · intro h
    constructor
    · intro p
      obtain ⟨v, hv, _⟩ := h p
      exact ⟨v, hv.2, hv.1⟩
    · intro u hu v hv huv
      obtain ⟨p, hp⟩ := node_on_terminal v
      obtain ⟨w, _, hw⟩ := h p
      exact (hw u ⟨ancestor_onPath huv hp, hu⟩).trans (hw v ⟨hp, hv⟩).symm
  · rintro ⟨hcover, hanti⟩ p
    obtain ⟨v, hv, hp⟩ := hcover p
    refine ⟨v, ⟨hp, hv⟩, ?_⟩
    intro u hu
    rcases path_nodes_comparable hu.1 hp with h | h
    · exact hanti u hu.2 v hv h
    · exact (hanti v hv u hu.2 h).symm
```

The file also proves that every complete set comes from a cut and that the cut is determined by its selected nodes. Together these results give `pruningEquiv`. This is why later theorems apply to every complete covering antichain, not just a preselected list of recursive examples.

6.4 Literal sums over selected nodes

`PruningSums` connects the recursive cut sum with the actual finite sum over the selected node set. Its operator theorem provides both the operator equality and the applied-signal equality.

Source: `KomorebiPaperV/PruningSums.lean`.

```
theorem packet_projector_synthesis {H : Type*} [NormedAddCommGroup H] [NormedSpace ℂ H]
    (p : A → H →L[ℂ] H) (t : FiniteAdaptive.Tree A)
    (h : conserves p t) (hr : p (label t) = ContinuousLinearMap.id ℂ H)
    (S : Pruning t) (x : H) :
    nodeSum p S = ContinuousLinearMap.id ℂ H ∧
      (∑ v ∈ S.val.toFinite.toFinset, p (nodeLabel v) x) = x := by
  have hs := (antichain_synthesis p t h S).trans hr
  refine ⟨hs, ?_⟩
  have hx := congrArg (fun T : H →L[ℂ] H => T x) hs
  simpa [nodeSum] using hx
```

The hypotheses `h : conserves p t` and `hr : p (label t) = ...id...` are inputs. There is no assertion that an arbitrary family of packet operators indexed by a tree automatically satisfies them.

6.5 Existence of a minimum allows nonadditive objectives

The manuscript's general criterion has the form

$$J(S) = R(S) + \eta L(S) + \tau K(S),$$

where R, L, and K can depend on the complete pruning as a whole. Finiteness alone ensures a minimum is attained.

Source: `KomorebiPaperV/PruningSums.lean`.

```
theorem objective_minimum (t : FiniteAdaptive.Tree A) (R L K : Pruning t → ℝ) (η τ : ℝ) :
    ∃ S : Pruning t, ∀ T, objective R L K η τ S ≤ objective R L K η τ T :=
  complete_pruning_minimum t _
```

No positivity or additivity of these functions is needed for the finite attainment theorem. Such assumptions can be useful in applications, but the logical reason for attainment here is a nonempty finite search space.

The dynamic program below solves an additive criterion $\sum_{v \in S} c(v)$. The general nonadditive existence result does not imply that the same recurrence efficiently minimizes an arbitrary coupled objective. A user of the recurrence must verify the required additive decomposition.

6.6 Strict gaps require competitors

If S_* is a strict minimizer and there is another pruning, the minimum positive objective difference is attained by a real competitor.

Source: `KomorebiPaperV/PruningSums.lean`.

```
theorem objective_gap_attained (t : FiniteAdaptive.Tree A) (J : Pruning t → ℝ) (S : Pruning t)
    (ho : ∃ T, T ≠ S) (hu : ∀ T, T ≠ S → J S < J T) :
    ∃ T : Pruning t, T ≠ S ∧ 0 < J T-J S ∧ ∀ U, U ≠ S → J T-J S ≤ J U-J S := by
  let : Fintype (Pruning t) := Fintype.ofEquiv (Cut t) (pruningEquiv t)
  classical
  obtain ⟨U,hU⟩ := ho
  obtain ⟨T,hT,hmin⟩ := KomorebiPaperV.finite_minimum (fun T => J T-J S)
    (Finset.univ.erase S) ⟨U, Finset.mem_erase.mpr ⟨hU, Finset.mem_univ _⟩⟩
  have hTS := (Finset.mem_erase.mp hT).1
  exact ⟨T,hTS,sub_pos.mpr (hu T hTS),fun V hV => hmin V
    (Finset.mem_erase.mpr ⟨hV, Finset.mem_univ _⟩)⟩
```

The explicit `ho` hypothesis excludes the empty competitor family. For a singleton pruning space, the selected pruning is unchanged for every objective, so a separate singleton theorem is the natural statement. A tie between distinct prunings gives zero separation and does not satisfy a positive strict-gap hypothesis.

6.7 The bottom up evaluator

For an additive node cost, the optimal value obeys

$$V(v) = \min\left(c(v), \sum_{u \text{ child of } v} V(u)\right).$$

The selector stops at the node when the two alternatives tie. The formal evaluator returns the cut, its cost, and three operation counters together.

Source: KomorebiPaperV/PruningAlgorithm.lean.

```
noncomputable def solve (c : A → ℝ) : (t : FiniteAdaptive.Tree A) → Result t
  | .leaf a => ⟨.stop _, c a, 0, 0, 1⟩
  | .branch a n ch =>
      let child := fun i => solve c (ch i)
      let total := ∑ i, (child i).cost
      let stop := c a ≤ total
      ⟨if stop then .stop _ else .split (fun i => (child i).cut),
       min (c a) total,
       1 + ∑ i, (child i).comparisons,
       (n+1) + ∑ i, (child i).additions,
       1 + ∑ i, (child i).visits⟩
```

The child results are introduced once as a family, and their stored costs are summed. The correctness theorem identifies both outputs with the specification: cost equals `value`, and cut equals `best`.

Source: KomorebiPaperV/PruningAlgorithm.lean.

```
theorem solve_correct (c : A → ℝ) (t : FiniteAdaptive.Tree A) :
    (solve c t).cost = value c t ∧ (solve c t).cut = best c t := by
  induction t with
  | leaf a => exact ⟨rfl,rfl⟩
  | branch a n ch ih =>
    have hc : (∑ i, (solve c (ch i)).cost) = ∑ i, value c (ch i) :=
      Finset.sum_congr rfl (fun i _ => (ih i).1)
    constructor
    · simp only [solve, value, hc]
    · simp only [solve, best, hc]
      split_ifs
      · rfl
      · congr 1
        funext i
        exact (ih i).2
```

The proof uses tree induction. In a branch, the induction hypotheses identify all cached child values with their specification values. The same comparison then controls both the evaluator and the specification, including equality.

6.8 An explicit small tree

Consider a root r with two leaf children a and b. Its complete prunings are $\{r\}$ and $\{a, b\}$. Set $c(r) = 7$, $c(a) = 2$, and $c(b) = 3$. The split costs 5 and is selected. The gap to the other pruning is 2.

Changing $c(r)$ to 5 creates a tie. The selector now stops at the root by convention. This deterministic choice is defined, but it is not protected by a positive separation margin. An arbitrarily small perturbation can change which pruning has smaller cost. Tie handling and strict stability address different questions.

If $p(r) = p(a) + p(b) = I$, both prunings synthesize exactly. At a tie, two representations can exchange preference while both preserve the full signal through the same synthesis identity.

6.9 What the linear-work count charges

The source proves one visit per node, one comparison per internal node, and one addition per edge. The last identity is expressed as additions plus one equals the node count. Combining them yields

Source: KomorebiPaperV/PruningAlgorithm.lean.

```
theorem solve_linear_work (c : A → ℝ) (t : FiniteAdaptive.Tree A) :
    (solve c t).visits + (solve c t).comparisons + (solve c t).additions ≤ 3 * nodes t := by
  have hv := solve_visits c t
  have hc := solve_comparisons c t
  have ha := solve_additions c t
  have hi := internal_le_nodes t
  omega
```

Node costs are supplied inputs. Their estimation from data, the cost of exact real arithmetic, output serialization, and the runtime behavior of a particular compiler are not charged by this theorem. The claimed bound is precise for its stated unit operation model. It is not a wall clock performance certificate or a cost bound for a nonadditive objective.

6.10 The exact worst pruning error envelope

Suppose local perturbations obey $|d(v) - c(v)| \le e(v)$. The maximum possible sum of the error weights along a complete pruning has the recurrence

$$E(v) = \max\left(e(v), \sum_{u \text{ child of } v} E(u)\right).$$

Source: KomorebiPaperV/FiniteAdaptive.lean.

```
noncomputable def envelope (e : A → ℝ) : Tree A → ℝ
  | .leaf a => e a
  | .branch a _ children => max (e a) (∑ i, envelope e (children i))
```

The source proves both that every cut weight is at most this envelope and that some cut attains it. It is therefore the exact maximum of the additive weights over complete prunings, rather than a loose sum over all nodes.

For the two-leaf example, if $e(r) = 0.3$ and $e(a) = e(b) = 0.2$, then $E(r) = \max(0.3, 0.4) = 0.4$. Both candidate objective values are perturbed by at most 0.4. The original gap is 2, and $2E(r) = 0.8 < 2$, so the selected split is protected.

6.11 Why the margin contains a factor of two

When comparing a competitor with the minimizer, the competitor can move down by the full error budget while the minimizer moves up by the same budget. Their separation can shrink by twice the budget. The antichain theorem preserves the selected pruning under the strict inequality.

Source: `KomorebiPaperV/PruningSums.lean`.

```
theorem antichain_selector_stable (c d e : A → ℝ) (h : ∀ a, |d a-c a| ≤ e a)
    (t : FiniteAdaptive.Tree A) (γ : ℝ)
    (hg : ∀ S : Pruning t, S ≠ selected c t → γ ≤ nodeSum c S-nodeSum c (selected c t))
    (hm : 2*envelope e t < γ) : selected d t = selected c t := by
  apply congrArg (pruningEquiv t)
  apply best_unchanged_of_gap c d e h t γ _ hm
  intro q hq
  have hne : pruningEquiv t q ≠ selected c t := fun he => hq ((pruningEquiv t).injective he)
  simpa only [selected, pruning_nodeSum] using hg (pruningEquiv t q) hne
```

Equality at the boundary is insufficient in general. With an original gap $\gamma = 2$ and budget $E = 1$, the winner can move up by one and the competitor down by one, producing a tie. A deterministic tie rule can then change the selected object. The theorem's strict inequality is part of its content.

6.12 Probability is transferred from a joint event

The probabilistic statement assumes one event on which all required local inequalities hold simultaneously. Deterministic stability then makes that event a subset of the event that the selector is unchanged.

Source: `KomorebiPaperV/PruningAlgorithm.lean`.

```
theorem antichain_probability {Ω : Type*} [MeasurableSpace Ω]
    (μ : MeasureTheory.Measure Ω) (good : Set Ω) (δ : ENNReal)
    (c e : A → ℝ) (d : Ω → A → ℝ) (t : FiniteAdaptive.Tree A) (γ : ℝ)
    (hgood : ∀ ω ∈ good, ∀ a, |d ω a-c a| ≤ e a)
    (hg : ∀ S : Pruning t, S ≠ selected c t → γ ≤ nodeSum c S-nodeSum c (selected c t))
    (hm : 2*envelope e t < γ) (hprob : 1-δ ≤ μ good) :
    1-δ ≤ μ {ω | selected (d ω) t = selected c t} := by
  apply hprob.trans
  apply MeasureTheory.measure_mono
  intro ω hω
  exact antichain_selector_stable c (d ω) e (hgood ω hω) t γ hg hm
```

Separate marginal statements for each node do not automatically give the same probability for their intersection. One needs a supplied joint event or an additional argument, such as a correctly budgeted union bound. No independence assumption or tail concentration theorem is inferred by the source.

The theorem is written for a measure with an explicit lower bound `1-δ ≤ μ good`. Lean's measure notation also evaluates arbitrary sets through the associated outer measure, so the set-inclusion proof

does not itself establish event measurability. For ordinary success probabilities, use a probability measure, measurable random node costs, and a measurable joint good event. The selector on a finite tree is then determined by finitely many measurable comparisons, making its success event measurable. Finite tree structure alone cannot make arbitrary random cost functions measurable. The mathematical text makes these probability hypotheses explicit. In Part II, consistency statements based only on probabilities tending to one are interpreted as convergence in probability. Almost-sure conclusions need additional hypotheses and an argument such as summable failure probabilities.

7 Formal cyclic products and projective Smith actions

7.1 A formal residue projector filters coefficients

For a formal power series $f = \sum_{n \geq 0} f_n X^n$, the jth cyclic projector retains the degrees congruent to j modulo N.

Source: KomorebiPaperV/CyclicSeries.lean.

```
def cyclicProjector (N j : ℕ) (f : R⟦X⟧) : R⟦X⟧ :=
  PowerSeries.mk fun n => if n % N = j % N then PowerSeries.coeff n f else 0
```

The ambient coefficient ring is a commutative semiring. There is no evaluation point and no convergence assumption in this definition. Multiplication is the formal Cauchy product, whose individual coefficients are finite sums.

The projector laws express idempotence, disjointness of different residue classes, and resolution of the identity for $N > 0$. A pure packet in residue i multiplied by one in residue j lies in residue $i + j$ modulo N.

7.2 The complete cyclic product law

Decompose both factors into packets, distribute multiplication, and retain only pairs whose residue labels add to the requested output label.

Source: KomorebiPaperV/CyclicSeries.lean.

```
theorem cyclic_product_law {N : ℕ} (hN : 0 < N) (k : ℕ) (f g : R⟦X⟧) :
    cyclicProjector N k (f * g) =
      ∑ i ∈ Finset.range N, ∑ j ∈ Finset.range N,
        if (i + j) % N = k % N then
          cyclicProjector N i f * cyclicProjector N j g
        else 0 := by
  calc
    cyclicProjector N k (f * g) =
        cyclicProjector N k
          ((∑ i ∈ Finset.range N, cyclicProjector N i f) *
            (∑ j ∈ Finset.range N, cyclicProjector N j g)) := by
      rw [sum_cyclicProjector hN, sum_cyclicProjector hN]
    _ = ∑ i ∈ Finset.range N, ∑ j ∈ Finset.range N,
          if (i + j) % N = k % N then
            cyclicProjector N i f * cyclicProjector N j g
```

```
          else 0 := by
      rw [Finset.sum_mul]
      simp_rw [Finset.mul_sum]
      rw [cyclicProjector_sum]
      apply Finset.sum_congr rfl
      intro i hi
      rw [cyclicProjector_sum]
      apply Finset.sum_congr rfl
      intro j hj
      by_cases hlabel : (i + j) % N = k % N
      · simp only [hlabel, ↓reduceIte]
        rw [cyclicProjector_eq_of_mod_eq (i := k) (j := (i + j) % N)
          (by simpa using hlabel.symm)]
        exact cyclicProjector_product_packet N i j f g
      · simp only [hlabel, ↓reduceIte]
        exact cyclicProjector_product_other (R := R) (fun h => hlabel h.symm) f g
```

For $N = 2$, write $f = f_0 + f_1$ and $g = g_0 + g_1$. Then

$$(fg)_0 = f_0 g_0 + f_1 g_1, \qquad (fg)_1 = f_0 g_1 + f_1 g_0.$$

The subscripts label degree parity. They are not independent scalar coefficients. This formula explains nonlinear coupling between packet labels without leaving the formal-series setting.

7.3 Powers include the empty product

The power theorem sums over all q-tuples of residue labels, retaining those whose sum has the requested residue.

Source: KomorebiPaperV/CyclicSeries.lean.

```
theorem cyclic_power_law {N : ℕ} (hN : 0 < N) (k q : ℕ) (f : R⟦X⟧) :
    cyclicProjector N k (f ^ q) =
      ∑ i : Fin q → Fin N, if (∑ a, (i a).val) % N = k % N then
        ∏ a, cyclicProjector N (i a).val f else 0 := by
  classical
  have hex : f ^ q = ∑ i : Fin q → Fin N, ∏ a, cyclicProjector N (i a).val f := by
    have h := Fintype.sum_pow (fun i : Fin N => cyclicProjector N i.val f) q
    rw [sum_fin_projector hN] at h
    exact h
  rw [hex, cyclicProjector_sum]
  apply Finset.sum_congr rfl
  intro i _
  exact project_finite_product N k Finset.univ (fun a => (i a).val)
    (fun a => cyclicProjector N (i a).val f)
    (fun a _ => cyclicProjector_idempotent N (i a).val f)
```

At $q = 0$, the type `Fin 0 → Fin N` has one function, the empty assignment. Its sum of labels is zero and its product is one. Thus the formula returns the constant series one for the zero residue and zero otherwise. This boundary case is part of the statement, not an unstated convention added afterward.

The foundational coefficient-projector proofs identify their adaptation from Chapter IV in source comments and in the cumulative census. Reusing them in a new application does not make them independent additional discoveries. Analytic root-of-unity formulas require their own convergence domain when formal series are evaluated.

7.4 Use homogeneous coordinates at poles

For a nonzero complex parameter a, the normalized homogeneous Smith action is

$$A_a \begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1+a & 1-a \\ 1-a & 1+a \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Source: `KomorebiPaperV/SmithAction.lean`.

```
noncomputable def action (a : ℂ) (p : ℂ × ℂ) : ℂ × ℂ :=
  (((1 + a) * p.1 + (1 - a) * p.2) / 2,
   ((1 - a) * p.1 + (1 + a) * p.2) / 2)
```

The source proves $A_a A_b = A_{ab}$, $A_1 = I$, and preservation of nonzero pairs when $a \neq 0$. Multiplying both input coordinates by the same nonzero scalar multiplies both output coordinates by that scalar.

A projective point is a nonzero pair modulo this common scaling. The source constructs the equivalence relation explicitly.

Source: `KomorebiPaperV/SmithCyclic.lean`.

```
noncomputable def pointSetoid : Setoid Point where
  r := same
  iseqv := {
    refl := by intro p; exact ⟨1, one_ne_zero, by simp⟩
    symm := by
      rintro p q ⟨c, hc, hq⟩
      refine ⟨c⁻¹, inv_ne_zero hc, ?_⟩
      rw [hq]
      simp [hc]
    trans := by
      rintro p q r ⟨c, hc, hq⟩ ⟨d, hd, hr⟩
      exact ⟨d*c, mul_ne_zero hd hc, by rw [hr, hq]; simp [mul_assoc]⟩ }
```

The quotient is the full projective line in this coordinate model. It includes $[0 : 1]$, representing zero impedance, and $[1 : 0]$, representing infinity. The action respects scaling, so it descends to the quotient.

7.5 Cyclic composition

The parameter $\omega_N = \exp(2\pi i/N)$ is nonzero and satisfies $\omega_N^N = 1$ for $N > 0$. The resulting projective actions compose by addition of exponents modulo N.

Source: KomorebiPaperV/SmithCyclic.lean.

```
theorem cyclic_composition (N : ℕ) (hN : 0 < N) (k l : ℕ) (p : Projective) :
    cyclicAction N k (cyclicAction N l p) = cyclicAction N ((k+l)%N) p := by
  unfold cyclicAction
  rw [projective_mul]
  have he : omega N ^ k * omega N ^ l = omega N ^ ((k+l)%N) := by
    rw [← pow_add, pow_eq_pow_mod (k+l) (omega_pow N hN)]
  congr 1
```

This source target establishes the modulo-N composition law. It should not be described as a separate formal proof that the action has exact order N unless the required primitivity and nontriviality converse are established as additional formal declarations. The broader traditional treatment contains further action analysis. Its prose scope is separately recorded.

7.6 Cayley coordinates explain the action

In the pair coordinates $(u-v, u+v)$, the action multiplies the first coordinate by a and leaves the second fixed.

Source: KomorebiPaperV/SmithCyclic.lean.

```
theorem cayley_conjugacy (a : ℂ) (p : ℂ × ℂ) :
    ((action a p).1 - (action a p).2, (action a p).1 + (action a p).2) =
      (a * (p.1-p.2), p.1+p.2) := by
  apply Prod.ext <;> dsimp [action] <;> ring
```

On a finite chart where the denominators are nonzero, this corresponds to rotation or multiplication of the reflection coefficient. The rational impedance formula is

$$z \longmapsto \frac{(1+a)z + (1-a)}{(1-a)z + (1+a)}.$$

A vanishing denominator means the image is the projective point at infinity. Lean's field division is totalized, so a bare identity of division expressions does not by itself assert that a displayed chart exists. The homogeneous action carries the geometry through such poles, while chart composition theorems carry explicit denominator conditions.

7.7 The half-turn swaps zero and infinity

For $N = 2$, $\omega_2 = -1$ and the homogeneous action swaps the two coordinates.

Source: KomorebiPaperV/SmithCyclic.lean.

```
theorem two_phase_swap (p : ℂ × ℂ) : action (omega 2 ^ 1) p = (p.2,p.1) := by
  rw [pow_one, omega_two, half_turn]
```

On the finite nonzero chart this gives $z \mapsto z^{-1}$. At $z = 0$ it maps $[0 : 1]$ to $[1 : 0]$. At infinity it does the reverse. The reciprocal formula is therefore a chart description of a globally defined projective transformation, not a rule that discards poles.

7.8 Dyadic and odd masks factor by CRT

If $N = 2^a M$ with M odd, the two factors are coprime. An integer satisfies a residue condition modulo their product precisely when it satisfies both factor conditions. The source proves the corresponding equality of integer-residue masks.

Source: `KomorebiPaperV/CRT.lean`.

```
theorem dyadic_odd_packet_factorization (a d : ℕ) (hd : Odd d)
    (r : ZMod (2 ^ a * d)) (x : ℤ → V) :
    KomorebiPaperV.packet (fun i : ℤ => (i : ZMod (2 ^ a * d))) r x =
      KomorebiPaperV.packet (fun i : ℤ => (i : ZMod (2 ^ a)))
        (ZMod.chineseRemainder (dyadic_odd_coprime a d hd) r).1
        (KomorebiPaperV.packet (fun i : ℤ => (i : ZMod d))
          (ZMod.chineseRemainder (dyadic_odd_coprime a d hd) r).2 x) :=
  residue_mask_factorization _ _ (dyadic_odd_coprime a d hd) r x
```

This is an exact mask factorization. It does not prove that a particular nonuniform sample allocation is optimal, nor does it provide a minimax sampling theorem. It supplies the arithmetic structure on which more specialized sampling constructions can be built.

8 Trust boundaries and reproduction

8.1 What the inherited audit actually checks

The preserved compiler log records 225 explicit theorem targets and 827 namespace declarations. The audit driver first requires each target to be present as a theorem. It then traverses the `KomorebiPaperV` namespace, rejects unsafe declarations, rejects local axiom and opaque declarations, and collects transitive axiom dependencies.

Only `propext`, `Classical.choice`, and `Quot.sound` are permitted. The audit checks the dependencies of the compiled declarations. It does not prove that English prose is a correct paraphrase of those declarations, that a Python implementation matches a Lean definition, or that an application has supplied a theorem's hypotheses.

The current guide build reconciles each target name with its immutable source location and inherited axiom-output record. It verifies pinned source hashes and performs a lexical scan for proof placeholders and prohibited source constructs. These checks protect provenance and locate obvious source-policy violations. They do not replace elaboration, type checking, or a new kernel traversal.

8.2 Recorded verification

The saved compiler transcript identifies the checked sources and their pinned dependencies. The accompanying reports distinguish that kernel execution from the source checks and finite computations described here.

The recorded computational review ran the twenty exact model tests, comprising 305,933 counted assertions, and all sixteen verifier control methods. It ran the 33 companion pytest tests and the 27 overlapping retained tests with no skips. These finite tests of implementations and models have separate contracts for their tested scope.

8.7 A checklist for using the theorems

Before using a result, identify its ambient type and explicit hypotheses, locate the exact declaration, and check how the application supplies each hypothesis. For an FIR bank, confirm finite Laurent support and delay units. For a transform equality, supply weighted summability. For pruning synthesis, supply conservation. For selector stability, establish the strict gap and joint local-error bound. For a Smith chart, check denominator nonvanishing or use the projective action.

This is the recurring pattern of the formal development: construct the object in the correct space, prove the bridge to the intended model, then use the identity whose hypotheses have actually been supplied.

9 The complete twenty-result correspondence

Each entry gives the scoped formal statement, its principal declarations, and the traditional manuscript locator. The 225-target catalog includes the supporting theorems as well. Principal targets may recur because one synthesis fact supports more than one manuscript statement. These are navigation relationships, not additive theorem counts. Traditional locators use printed Arabic pages. The unnumbered cover and Roman opening pages have separate physical PDF indices, preserved in the companion crosswalk.

9.1 1.1 Impulse seed theorem

Traditional locator. Theorem 1.1, final paper p. 2; `result:01`. **Exact reviewed scope.** Complex l2 impulse expansion, finite-support identity, unitary shifts and dense shift-generated span. `KomorebiPaperV.HilbertPackets.impulse_expansion`

`KomorebiPaperV/HilbertPackets.lean`, line 135.

`KomorebiPaperV.HilbertPackets.finite_impulse_expansion`

`KomorebiPaperV/HilbertPackets.lean`, line 207.

`KomorebiPaperV.HilbertPackets.shift_generated_dense_span`

`KomorebiPaperV/HilbertPackets.lean`, line 203.

`KomorebiPaperV.HilbertPackets.shift_impulse`

`KomorebiPaperV/HilbertPackets.lean`, line 183.

9.2 2.1 Polyphase projector theorem

Traditional locator. Theorem 2.1, final paper p. 3; `result:02`. **Exact reviewed scope.** Bounded orthogonal residue projectors on complex l2. `KomorebiPaperV.HilbertPackets.projector_idempotent`

`KomorebiPaperV/HilbertPackets.lean`, line 56.

`KomorebiPaperV.HilbertPackets.projector_annihilate`

`KomorebiPaperV/HilbertPackets.lean`, line 62.

`KomorebiPaperV.HilbertPackets.projector_complete`

`KomorebiPaperV/HilbertPackets.lean`, line 69.

`KomorebiPaperV.HilbertPackets.projector_symmetric`

`KomorebiPaperV/HilbertPackets.lean`, line 139.

`KomorebiPaperV.HilbertPackets.projector_orthogonal`

`KomorebiPaperV/HilbertPackets.lean`, line 147.

9.3 2.2 Analysis–synthesis identity

Traditional locator. Corollary 2.2, final paper p. 3; `result:03`. **Exact reviewed scope.** Signed integer polyphase linear isometry; channels represented by joint l2 coordinates. `KomorebiPaperV.HilbertPackets.polyphase_synthesis_analysis`

`KomorebiPaperV/HilbertPackets.lean`, line 123.

`KomorebiPaperV.HilbertPackets.polyphase_analysis_synthesis`

`KomorebiPaperV/HilbertPackets.lean`, line 127.

`KomorebiPaperV.HilbertPackets.polyphase_energy`

`KomorebiPaperV/HilbertPackets.lean`, line 131.

9.4 2.3 Exact nested refinement

Traditional locator. Theorem 2.3, final paper p. 5; `result:04`. **Exact reviewed scope.** Literal divisible-modulus bounded projector refinement. `KomorebiPaperV.HilbertPackets.residue_projector_refinement`

`KomorebiPaperV/HilbertPackets.lean`, line 187.

9.5 3.1 Exact FIR polyphase transfer law

Traditional locator. Theorem 3.1, final paper p. 5; `result:05`. **Exact reviewed scope.** Time-domain transfer for every bilateral sequence; channel z-transform identity for z nonzero wherever each weighted input series is summable. `KomorebiPaperV.ZTransform.fir_channel_transform`

`KomorebiPaperV/ZTransform.lean`, line 49.

`KomorebiPaperV.ZTransform.fir_matrix_transform`

`KomorebiPaperV/ZTransform.lean`, line 64.

`KomorebiPaperV.FIRTransfer.fir_polyphase_transfer`

`KomorebiPaperV/FIRTransfer.lean`, line 96.

9.6 4.1 Perfect-reconstruction criterion

Traditional locator. Theorem 4.1, final paper p. 7; `result:06`. **Exact reviewed scope.** Concrete finite Laurent/FIR banks on every bilateral sequence, faithful matrix/operator correspondence, gain and channel delay; original-sample delay is N*d. Arbitrary scalar-matrix criterion is also retained in Finite-Transforms. `KomorebiPaperV.FilterBank.act_is_fir`

`KomorebiPaperV/FilterBank.lean`, line 36.

`KomorebiPaperV.FilterBank.bank_mul`

`KomorebiPaperV/FilterBank.lean`, line 84.

`KomorebiPaperV.FilterBank.bank_faithful`

`KomorebiPaperV/FilterBank.lean`, line 104.

`KomorebiPaperV.FilterBank.perfect_reconstruction_iff`

`KomorebiPaperV/FilterBank.lean`, line 133.

`KomorebiPaperV.FilterBank.bank_transform`

`KomorebiPaperV/FilterBank.lean`, line 147.

`KomorebiPaperV.FilterBank.original_sample_delay`

`KomorebiPaperV/FilterBank.lean`, line 155.

9.7 4.2 Haar energy identity

Traditional locator. Corollary 4.2, final paper p. 8; `result:07`. **Exact reviewed scope.** Literal interleaved input and normalized low/detail channels in complex l2. `KomorebiPaperV.HaarChannels.haar_energy`

`KomorebiPaperV/HaarChannels.lean`, line 40.

`KomorebiPaperV.HaarChannels.haar_low_samples`

`KomorebiPaperV/HaarChannels.lean`, line 44.

`KomorebiPaperV.HaarChannels.haar_high_samples`

`KomorebiPaperV/HaarChannels.lean`, line 48.

9.8 4.3 Exact lifting reconstruction

Traditional locator. Theorem 4.3, final paper p. 10; `result:08`. **Exact reviewed scope.** Pairwise predict/update inverse and determinant one, with both flat finite-signal inverse laws. `KomorebiPaperV.FiniteTransforms.unlift_lift_flat`

`KomorebiPaperV/FiniteTransforms.lean`, line 78.

`KomorebiPaperV.FiniteTransforms.lift_unlift_flat`

`KomorebiPaperV/FiniteTransforms.lean`, line 83.

`KomorebiPaperV.lifting_synthesis_analysis`

`KomorebiPaperV/Lifting.lean`, line 21.

`KomorebiPaperV.lifting_analysis_synthesis`

`KomorebiPaperV/Lifting.lean`, line 25.

`KomorebiPaperV.lifting_determinant`

`KomorebiPaperV/Lifting.lean`, line 29.

9.9 5.1 Wavelet-packet reconstruction and energy

Traditional locator. Theorem 5.1, final paper p. 10; `result:09`. **Exact reviewed scope.** All finite depths of literal Haar packet operators; sums of true Hilbert adjoints reconstruct, and packet norms preserve energy. `KomorebiPaperV.WaveletPackets.packet_energy`

`KomorebiPaperV/WaveletPackets.lean`, line 68.

`KomorebiPaperV.WaveletPackets.packet_operator_reconstruction`

`KomorebiPaperV/WaveletPackets.lean`, line 94.

`KomorebiPaperV.WaveletPackets.packet_reconstruction`

`KomorebiPaperV/WaveletPackets.lean`, line 98.

9.10 6.1 Exact known-edge theorem

Traditional locator. Theorem 6.1, final paper p. 12; `result:10`. **Exact reviewed scope.** Exact known-edge cancellation in arbitrary seminormed additive groups. `KomorebiPaperV.known_edge_error`

`KomorebiPaperV/KnownEdge.lean`, line 17.

`KomorebiPaperV.known_edge_norm`

`KomorebiPaperV/KnownEdge.lean`, line 34.

9.11 6.2 Several exact edge packets

Traditional locator. Corollary 6.2, final paper p. 12; `result:11`. **Exact reviewed scope.** Finite families of exactly retained jump packets. `KomorebiPaperV.several_known_edges_error`

`KomorebiPaperV/KnownEdge.lean`, line 25.

`KomorebiPaperV.several_known_edges_norm`

`KomorebiPaperV/KnownEdge.lean`, line 38.

9.12 7.1 Packet perturbation bound

Traditional locator. Theorem 7.1, final paper p. 13; `result:12`. **Exact reviewed scope.** Measurable scalar masks in arbitrary measure spaces; extended finite-p norms, and pointwise plus essential supremum bounds. Finite mask and symmetric-difference measures yield the finite real-valued statement. `KomorebiPaperV.MaskPerturbation.indicator_difference_norm`

`KomorebiPaperV/MaskPerturbation.lean`, line 25.

`KomorebiPaperV.MaskPerturbation.finite_p_perturbation`

`KomorebiPaperV/MaskPerturbation.lean`, line 43.

`KomorebiPaperV.MaskPerturbation.pointwise_sup_bound`

`KomorebiPaperV/MaskPerturbation.lean`, line 59.

`KomorebiPaperV.MaskPerturbation.infinity_perturbation`

`KomorebiPaperV/MaskPerturbation.lean`, line 73.

`KomorebiPaperV.MaskPerturbation.full_signal_perturbation`

`KomorebiPaperV/MaskPerturbation.lean`, line 79.

9.13 11.2 Exact synthesis from every complete pruning

Traditional locator. Theorem 11.2, final paper p. 20; `result:13`. **Exact reviewed scope.** Literal finite node sums over covering antichains, with exact operator and applied-signal synthesis under the manuscript conservation hypotheses. `KomorebiPaperV.PruningSums.cutSum_eq_nodeSum`

`KomorebiPaperV/PruningSums.lean`, line 54.

`KomorebiPaperV.PruningSums.antichain_synthesis`

`KomorebiPaperV/PruningSums.lean`, line 81.

`KomorebiPaperV.PruningSums.packet_projector_synthesis`

`KomorebiPaperV/PruningSums.lean`, line 88.

9.14 11.3 Finite globally optimal stable selector

Traditional locator. Theorem 11.3, final paper p. 20; `result:14`. **Exact reviewed scope.** Named non-additive representation/leakage/complexity criterion; attained positive gap when competitors exist, uniform perturbation stability, and singleton case. Every pruning preserves synthesis. `KomorebiPaperV.PruningSums.objective_minimum`

`KomorebiPaperV/PruningSums.lean`, line 134.

`KomorebiPaperV.PruningSums.objective_gap_attained`

`KomorebiPaperV/PruningSums.lean`, line 165.

`KomorebiPaperV.PruningSums.objective_stability`

`KomorebiPaperV/PruningSums.lean`, line 148.

`KomorebiPaperV.PruningSums.singleton_selector`

`KomorebiPaperV/PruningSums.lean`, line 158.

`KomorebiPaperV.PruningSums.packet_projector_synthesis`

`KomorebiPaperV/PruningSums.lean`, line 88.

9.15 11.4 Dynamic-programming best packet basis

Traditional locator. Theorem 11.4, final paper p. 21; `result:15`. **Exact reviewed scope.** Cached bottom-up evaluator equals the global selector; stop on ties. One comparison per internal node; at most 3*node-count unit operations for visits/comparisons/value additions with node costs supplied. Not a Lean runtime or scalar-cost evaluation bound. `KomorebiPaperV.PruningAlgorithm.solve_correct`

`KomorebiPaperV/PruningAlgorithm.lean`, line 36.

`KomorebiPaperV.PruningAlgorithm.solve_comparisons`

`KomorebiPaperV/PruningAlgorithm.lean`, line 52.

`KomorebiPaperV.PruningAlgorithm.solve_visits`

`KomorebiPaperV/PruningAlgorithm.lean`, line 61.

`KomorebiPaperV.PruningAlgorithm.solve_additions`

`KomorebiPaperV/PruningAlgorithm.lean`, line 69.

`KomorebiPaperV.PruningAlgorithm.solve_linear_work`

`KomorebiPaperV/PruningAlgorithm.lean`, line 89.

`KomorebiPaperV.PruningAlgorithm.computed_antichain_optimal`

`KomorebiPaperV/PruningAlgorithm.lean`, line 97.

9.16 11.5 Local perturbation certificate

Traditional locator. Corollary 11.5, final paper p. 21; `result:16`. **Exact reviewed scope.** Exact maximum antichain error envelope and strict-gap stability; probability transfer from a joint local-error event is proved in PruningAlgorithm. `KomorebiPaperV.PruningSums.antichain_perturbation`

`KomorebiPaperV/PruningSums.lean`, line 108.

`KomorebiPaperV.PruningSums.antichain_envelope_attained`

`KomorebiPaperV/PruningSums.lean`, line 115.

`KomorebiPaperV.PruningSums.antichain_selector_stable`

`KomorebiPaperV/PruningSums.lean`, line 120.

`KomorebiPaperV.PruningAlgorithm.antichain_probability`

`KomorebiPaperV/PruningAlgorithm.lean`, line 102.

9.17 12.1 Cyclic product law

Traditional locator. Theorem 12.1, final paper p. 24; `result:17`. **Exact reviewed scope.** Formal power-series cyclic product and literal q-tuple power formula over commutative semirings, including q=0. Analytic interpretations apply on their convergence domains; no arbitrary divergent evaluation is asserted. `KomorebiPaperV.CyclicSeries.cyclic_product_law`

`KomorebiPaperV/CyclicSeries.lean`, line 126.

`KomorebiPaperV.CyclicSeries.cyclic_power_law`

`KomorebiPaperV/CyclicSeries.lean`, line 195.

9.18 13.2 Polyphase Möbius action

Traditional locator. Theorem 13.2, final paper p. 26; `result:18`. **Exact reviewed scope.** Explicit exponential root of unity and modulo-N action on the projective quotient of nonzero homogeneous pairs; Cayley conjugacy and the finite-chart formula. `KomorebiPaperV.SmithCyclic.omega_pow`

`KomorebiPaperV/SmithCyclic.lean`, line 12.

`KomorebiPaperV.SmithCyclic.cyclic_composition`

`KomorebiPaperV/SmithCyclic.lean`, line 78.

`KomorebiPaperV.SmithCyclic.cayley_conjugacy`

`KomorebiPaperV/SmithCyclic.lean`, line 87.

`KomorebiPaperV.SmithCyclic.cyclic_chart`

`KomorebiPaperV/SmithCyclic.lean`, line 92.

`KomorebiPaperV.SmithCyclic.projective_mul`

`KomorebiPaperV/SmithCyclic.lean`, line 56.

9.19 13.3 Impedance–admittance is the two-phase split

Traditional locator. Corollary 13.3, final paper p. 26; `result:19`. **Exact reviewed scope.** The N=2 phase is -1; its action swaps homogeneous coordinates, exchanging zero/infinity and giving inverse impedance on the nonzero chart. `KomorebiPaperV.SmithCyclic.omega_two`

`KomorebiPaperV/SmithCyclic.lean`, line 19.

`KomorebiPaperV.SmithCyclic.two_phase_swap`

`KomorebiPaperV/SmithCyclic.lean`, line 97.

`KomorebiPaperV.SmithCyclic.two_phase_impedance_admittance`

`KomorebiPaperV/SmithCyclic.lean`, line 100.

9.20 14.1 Dyadic–odd packet factorization

Traditional locator. Proposition 14.1, final paper p. 29; `result:20`. **Exact reviewed scope.** Signed integer CRT residue masks and dyadic-odd factorization. `KomorebiPaperV.CRT.residue_mask_factorization`

`KomorebiPaperV/CRT.lean`, line 20.

`KomorebiPaperV.CRT.dyadic_odd_packet_factorization`

`KomorebiPaperV/CRT.lean`, line 34.

10 The broader traditional developments

The mathematical text preserves two substantial source strands. Part I contains the twenty-result formal core. Part II contains sixty-three proof records from the broader Annals development. Together they produce 83 proof records, with overlap in mathematical content. The number 83 counts presented proofs. It does not replace the twenty-result Lean denominator or establish 83 independent new results. The table below connects groups of Part II developments to nearby formal-core concepts. "Nearby" is a reading aid. It does not assert a formal reduction or that the stated Part II result has been compiled in Lean.

Part II topic	Nearest formal core	Additional boundary
Operator partitions and refinement	Results 1.1–2.3, `HilbertPackets`	Broader operator settings or frames require their own assumptions and proofs.
Polyphase symbols and banks	Results 3.1–4.3, `FilterBank`, `ZTransform`	Unit-circle and L^2 boundary interpretations must be distinguished from off-circle weighted summability.
Frame trees and adaptive objectives	Results 11.2–11.5, `PruningSums`	Local frame bounds are nonnegative. The dynamic-programming result applies to additive costs.
Known edges, masks, and commutators	Results 6.1–7.1, `KnownEdge`, `MaskPerturbation`	A commutator using Q as an operator requires bounded linearity. Cancellation alone does not.
Boundary tubes and edge geometry	Mask perturbation theorem 7.1	Tube estimates use their local range $\delta < \delta_0$. Geometric constants and regularity are additional data.
Continuous Haar edge models	Discrete Haar results 4.2 and 5.1	Half-line step functions need care: their wavelet coefficients can be used even when the steps themselves are not in $L^2(\mathbb{R})$.
Random or asymptotic selection	Finite joint-event transfer 11.5	Probability bounds tending to one establish convergence in probability. Almost sure claims need stronger evidence.
Nonlinear cyclic products	Result 12.1, `CyclicSeries`	Analytic evaluation and stronger functional-analytic claims are additional to formal coefficient identities.
Smith geometry and sampling	Results 13.2–14.1, `SmithCyclic`, `CRT`	The formal composition law alone is not a target proving exact-order converse or optimal sampling.

The final traditional audit records the precise scope repairs applied to Part II. This guide preserves the resulting hypotheses in its discussion. It does not incorporate altered claims into unchanged Lean targets. The following inventory provides a locator for every Part II proof record and explicitly keeps its formal status separate.

10.1 All sixty-three Part II proof records

Every row below has status: prose proof; no additional Lean claim. The final E-prefixed result number and page refer to the combined traditional paper. The nearest-core column identifies related reading only.

Final locator	Part II statement	Nearby core reading
E2.1, p. 42	Theorem: Impulse generation	1.1
E2.3, p. 43	Proposition: Finite periodic signals	1.1
E3.2, p. 43	Theorem: Exact lattice polyphase decomposition	2.1–2.3
E3.3, p. 44	Corollary: One dimensional residue coordinates	2.1–2.3
E3.4, p. 44	Proposition: Translation covariance	2.1–2.3
E3.5, p. 44	Theorem: Exact lattice refinement	2.1–2.3
E4.2, p. 46	Theorem: Exact convolution transfer law	3.1
E4.3, p. 46	Proposition: Transfer norm control	3.1
E4.4, p. 47	Proposition: Composition of transfer matrices	3.1
E4.5, p. 47	Corollary: Finite impulse response closure	3.1
E5.1, p. 48	Theorem: Perfect reconstruction criterion	4.1
E5.2, p. 49	Theorem: Quantitative imperfect reconstruction	4.1
E5.4, p. 49	Theorem: Stability under filter perturbation	4.1
E5.5, p. 50	Theorem: Frame bounds from the polyphase symbol	4.1
E5.6, p. 50	Corollary: Coefficient noise under canonical synthesis	4.1
E5.7, p. 50	Theorem: Perturbation of frame bounds	4.1
E5.9, p. 51	Corollary: Paraunitary bank	4.1
E5.10, p. 51	Corollary: Haar reconstruction and energy	4.2
E5.11, p. 52	Theorem: Exact lifting update	4.3
E6.2, p. 53	Theorem: Reconstruction on a pruned packet tree	5.1, 11.2
E6.3, p. 53	Corollary: Coefficient perturbations in a unitary tree	5.1, 11.2
E6.4, p. 53	Proposition: Bounds for nonunitary local splits	5.1, 11.2
E6.5, p. 54	Theorem: Exact additive best basis recursion	11.3–11.5
E6.6, p. 55	Theorem: Stability of the optimal cost	11.3–11.5
E6.7, p. 55	Corollary: Stability of the selected basis	11.3–11.5
E7.2, p. 56	Theorem: Higher order singular packet decomposition	6.1–6.2
E7.3, p. 57	Corollary: Uniqueness of the trace packet	6.1–6.2
E7.5, p. 57	Theorem: Exact retained packet identity	6.1–6.2
E7.7, p. 58	Corollary: Sobolev rate after packet separation	6.1–6.2
E7.8, p. 58	Theorem: Reconstruction with an estimated packet	7.1

Final locator	Part II statement	Nearby core reading
E8.1, p. 59	Theorem: Exact scalar region formula	7.1
E8.2, p. 60	Corollary: Location error for one step	7.1
E8.3, p. 60	Theorem: Location stability for higher order packets	7.1
E8.4, p. 61	Theorem: Overlap matrix formula	7.1
E8.5, p. 61	Corollary: Background and packet error	7.1
E8.7, p. 62	Theorem: Geometric packet bound	7.1, 11.5
E9.1, p. 63	Lemma: Contrast gap	7.1, 11.5
E9.2, p. 64	Theorem: Deterministic jump cell recovery	7.1, 11.5
E9.3, p. 64	Corollary: Jump count and sign recovery	7.1, 11.5
E9.4, p. 64	Corollary: Packet reconstruction after detection	7.1, 11.5
E9.6, p. 65	Corollary: Deterministic consistency	7.1, 11.5
E9.8, p. 67	Theorem: Probability recovery bound	7.1, 11.5
E9.9, p. 67	Corollary: Repeated measurements	7.1, 11.5
E9.11, p. 67	Corollary: Probability consistency	7.1, 11.5
E9.12, p. 68	Proposition: Vector jump recovery	7.1, 11.5
E10.1, p. 68	Theorem: Smooth Haar coefficient bound	4.2; edge context
E10.2, p. 69	Theorem: Exact Haar coefficient of a step	4.2; edge context
E10.3, p. 69	Corollary: Scale separation	4.2; edge context
E10.4, p. 69	Theorem: Half cell shift guarantee	4.2; edge context
E10.5, p. 70	Corollary: Alignment free two grid Haar detection	4.2; edge context
E10.6, p. 71	Theorem: Polynomial annihilation and boundary support	6.1–7.1
E10.7, p. 71	Proposition: Noise amplification by differences	6.1–7.1
E10.8, p. 71	Corollary: Robust breakpoint bands	6.1–7.1
E11.2, p. 72	Theorem: Uniform convergence with exact jump packets	6.1–7.1
E11.3, p. 73	Theorem: Estimated packet Gibbs bound	6.1–7.1
E11.4, p. 73	Corollary: Boundary band error	6.1–7.1
E12.1, p. 74	Theorem: Residue law for Cauchy products	12.1
E12.2, p. 75	Theorem: Shifted pointwise monomial law	12.1
E12.3, p. 75	Corollary: Polynomial shift filters	12.1
E14.1, p. 77	Theorem: Cyclic Mobius action	13.2–13.3
E14.2, p. 78	Corollary: Impedance and admittance	13.2–13.3
E14.3, p. 78	Proposition: Radial and phase coordinates	13.2–13.3
E14.4, p. 78	Proposition: Coprime phase factorization	14.1

10.2 Open conjecture locators

Stable edge-packet recovery. Conjecture 7.2, final traditional p. 13; `core:record:17`. Open; excluded from the twenty-result formal contract.

Edge-packet approximation class. Conjecture 15.1, final traditional p. 29; `core:record:35`. Open; excluded from the twenty-result formal contract.

11 All 225 theorem targets

This catalog reproduces every qualified target from the saved candidate_contract.json, grouped by its actual source module. Line numbers refer to the immutable bundled Lean sources. Every target was reconciled with its declaration line, audit directive, and inherited axiom-output record. This reconciliation does not perform Lean elaboration.

11.1 BinaryAdaptive: 6 targets

Source: `formalization/lean/KomorebiPaperV/BinaryAdaptive.lean`.

`KomorebiPaperV.Binary.cutSum_eq_root`
Source line 36; theorem.

`KomorebiPaperV.Binary.value_le_cost`
Source line 59; theorem.

`KomorebiPaperV.Binary.best_cost`
Source line 69; theorem.

`KomorebiPaperV.Binary.best_is_global_minimum`
Source line 79; theorem.

`KomorebiPaperV.Binary.root_le_envelope`
Source line 88; theorem.

`KomorebiPaperV.Binary.cut_perturbation`
Source line 94; theorem.

11.2 CRT: 4 targets

Source: `formalization/lean/KomorebiPaperV/CRT.lean`.

`KomorebiPaperV.CRT.residue_pair_iff`
Source line 13; theorem.

`KomorebiPaperV.CRT.residue_mask_factorization`
Source line 20; theorem.

`KomorebiPaperV.CRT.dyadic_odd_coprime`
Source line 31; theorem.

`KomorebiPaperV.CRT.dyadic_odd_packet_factorization`
Source line 34; theorem.

11.3 CompletePrunings: 15 targets

Source: `formalization/lean/KomorebiPaperV/CompletePrunings.lean`.

`KomorebiPaperV.CompletePrunings.terminal_exists`
Source line 36; theorem.

`KomorebiPaperV.CompletePrunings.node_on_terminal`
Source line 43; theorem.

`KomorebiPaperV.CompletePrunings.selected_stop_iff`

Source line 52; theorem.

`KomorebiPaperV.CompletePrunings.selected_split_root_false`
Source line 58; theorem.

`KomorebiPaperV.CompletePrunings.selected_split_down_iff`
Source line 62; theorem.

`KomorebiPaperV.CompletePrunings.cut_complete`
Source line 69; theorem.

`KomorebiPaperV.CompletePrunings.ancestor_onPath`
Source line 97; theorem.

`KomorebiPaperV.CompletePrunings.path_nodes_comparable`
Source line 108; theorem.

`KomorebiPaperV.CompletePrunings.complete_iff_covering_antichain`
Source line 120; theorem.

`KomorebiPaperV.CompletePrunings.complete_root_unique`
Source line 140; theorem.

`KomorebiPaperV.CompletePrunings.complete_child`
Source line 149; theorem.

`KomorebiPaperV.CompletePrunings.complete_has_cut`
Source line 164; theorem.

`KomorebiPaperV.CompletePrunings.cut_ext`
Source line 192; theorem.

`KomorebiPaperV.CompletePrunings.complete_pruning_finite`
Source line 226; theorem.

`KomorebiPaperV.CompletePrunings.complete_pruning_minimum`
Source line 230; theorem.

11.4 CyclicSeries: 15 targets

Source: `formalization/lean/KomorebiPaperV/CyclicSeries.lean`.

`KomorebiPaperV.CyclicSeries.coeff_cyclicProjector`
Source line 33; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_idempotent`
Source line 38; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_orthogonal`
Source line 43; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_eq_zero_of_residue_ne`
Source line 51; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_eq_of_mod_eq`
Source line 56; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_add`
Source line 62; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_sum`

Source line 67; theorem.

`KomorebiPaperV.CyclicSeries.sum_cyclicProjector`
Source line 78; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_product_packet`
Source line 92; theorem.

`KomorebiPaperV.CyclicSeries.cyclicProjector_product_other`
Source line 118; theorem.

`KomorebiPaperV.CyclicSeries.cyclic_product_law`
Source line 126; theorem.

`KomorebiPaperV.CyclicSeries.pure_finite_product`
Source line 159; theorem.

`KomorebiPaperV.CyclicSeries.project_finite_product`
Source line 177; theorem.

`KomorebiPaperV.CyclicSeries.sum_fin_projector`
Source line 189; theorem.

`KomorebiPaperV.CyclicSeries.cyclic_power_law`
Source line 195; theorem.

11.5 FIRTransfer: 6 targets

Source: `formalization/lean/KomorebiPaperV/FIRTransfer.lean`.

`KomorebiPaperV.FIRTransfer.tapEquiv_symm`
Source line 34; theorem.

`KomorebiPaperV.FIRTransfer.tapIndex_equiv`
Source line 41; theorem.

`KomorebiPaperV.FIRTransfer.transfer_address`
Source line 48; theorem.

`KomorebiPaperV.FIRTransfer.fir_reindex`
Source line 65; theorem.

`KomorebiPaperV.FIRTransfer.sum_pair_fiber`
Source line 82; theorem.

`KomorebiPaperV.FIRTransfer.fir_polyphase_transfer`
Source line 96; theorem.

11.6 FilterBank: 13 targets

Source: `formalization/lean/KomorebiPaperV/FilterBank.lean`.

`KomorebiPaperV.FilterBank.act_apply`
Source line 31; theorem.

`KomorebiPaperV.FilterBank.act_is_fir`
Source line 36; theorem.

`KomorebiPaperV.FilterBank.act_impulse`

Source line 43; theorem.

`KomorebiPaperV.FilterBank.act_single`
Source line 50; theorem.

`KomorebiPaperV.FilterBank.evalZ_apply`
Source line 65; theorem.

`KomorebiPaperV.FilterBank.filter_transform`
Source line 70; theorem.

`KomorebiPaperV.FilterBank.bank_mul`
Source line 84; theorem.

`KomorebiPaperV.FilterBank.bank_impulse`
Source line 93; theorem.

`KomorebiPaperV.FilterBank.bank_faithful`
Source line 104; theorem.

`KomorebiPaperV.FilterBank.gain_delay_action`
Source line 121; theorem.

`KomorebiPaperV.FilterBank.perfect_reconstruction_iff`
Source line 133; theorem.

`KomorebiPaperV.FilterBank.bank_transform`
Source line 147; theorem.

`KomorebiPaperV.FilterBank.original_sample_delay`
Source line 155; theorem.

11.7 FiniteAdaptive: 18 targets

Source: `formalization/lean/KomorebiPaperV/FiniteAdaptive.lean`.

`KomorebiPaperV.FiniteAdaptive.cutSum_eq_root`
Source line 39; theorem.

`KomorebiPaperV.FiniteAdaptive.cutSum_eq_identity`
Source line 53; theorem.

`KomorebiPaperV.FiniteAdaptive.pathHits_one`
Source line 67; theorem.

`KomorebiPaperV.FiniteAdaptive.arbitrary_objective_minimum`
Source line 107; theorem.

`KomorebiPaperV.FiniteAdaptive.strict_minimum_positive_gap`
Source line 114; theorem.

`KomorebiPaperV.FiniteAdaptive.value_le_cost`
Source line 136; theorem.

`KomorebiPaperV.FiniteAdaptive.best_cost`
Source line 146; theorem.

`KomorebiPaperV.FiniteAdaptive.best_is_global_minimum`
Source line 156; theorem.

`KomorebiPaperV.FiniteAdaptive.best_preserves_synthesis`

Source line 161; theorem.

`KomorebiPaperV.FiniteAdaptive.stop_on_tie`
Source line 165; theorem.

`KomorebiPaperV.FiniteAdaptive.root_le_envelope`
Source line 174; theorem.

`KomorebiPaperV.FiniteAdaptive.cost_le_envelope`
Source line 179; theorem.

`KomorebiPaperV.FiniteAdaptive.envelope_attained`
Source line 187; theorem.

`KomorebiPaperV.FiniteAdaptive.cut_perturbation`
Source line 201; theorem.

`KomorebiPaperV.FiniteAdaptive.value_perturbation`
Source line 213; theorem.

`KomorebiPaperV.FiniteAdaptive.best_unchanged_of_gap`
Source line 223; theorem.

`KomorebiPaperV.FiniteAdaptive.one_comparison_per_internal_node`
Source line 243; theorem.

`KomorebiPaperV.FiniteAdaptive.selector_probability`
Source line 252; theorem.

11.8 FiniteSelection: 5 targets

Source: `formalization/lean/KomorebiPaperV/FiniteSelection.lean`.

`KomorebiPaperV.finite_minimum`
Source line 15; theorem.

`KomorebiPaperV.positive_gap`
Source line 48; theorem.

`KomorebiPaperV.gap_stability`
Source line 58; theorem.

`KomorebiPaperV.singleton_minimum`
Source line 69; theorem.

`KomorebiPaperV.singleton_selection_unchanged`
Source line 76; theorem.

11.9 FiniteTransforms: 12 targets

Source: `formalization/lean/KomorebiPaperV/FiniteTransforms.lean`.

`KomorebiPaperV.FiniteTransforms.unpair_pair`
Source line 43; theorem.

`KomorebiPaperV.FiniteTransforms.pair_unpair`
Source line 48; theorem.

`KomorebiPaperV.FiniteTransforms.flat_synthesis_analysis`

Source line 59; theorem.

`KomorebiPaperV.FiniteTransforms.flat_analysis_synthesis`
Source line 64; theorem.

`KomorebiPaperV.FiniteTransforms.unlift_lift_flat`
Source line 78; theorem.

`KomorebiPaperV.FiniteTransforms.lift_unlift_flat`
Source line 83; theorem.

`KomorebiPaperV.FiniteTransforms.haar_involution`
Source line 92; theorem.

`KomorebiPaperV.FiniteTransforms.haar_pair_energy`
Source line 103; theorem.

`KomorebiPaperV.FiniteTransforms.haar_finite_energy`
Source line 110; theorem.

`KomorebiPaperV.FiniteTransforms.haar_flat_involution`
Source line 118; theorem.

`KomorebiPaperV.FiniteTransforms.matrix_eq_of_all_vectors`
Source line 125; theorem.

`KomorebiPaperV.FiniteTransforms.perfect_reconstruction_matrix_iff`
Source line 134; theorem.

11.10 HaarChannels: 8 targets

Source: `formalization/lean/KomorebiPaperV/HaarChannels.lean`.

`KomorebiPaperV.HaarChannels.norm_sq_tsum`
Source line 12; theorem.

`KomorebiPaperV.HaarChannels.channel_energy`
Source line 15; theorem.

`KomorebiPaperV.HaarChannels.even_apply`
Source line 26; theorem.

`KomorebiPaperV.HaarChannels.odd_apply`
Source line 30; theorem.

`KomorebiPaperV.HaarChannels.interleaved_energy`
Source line 32; theorem.

`KomorebiPaperV.HaarChannels.haar_energy`
Source line 40; theorem.

`KomorebiPaperV.HaarChannels.haar_low_samples`
Source line 44; theorem.

`KomorebiPaperV.HaarChannels.haar_high_samples`
Source line 48; theorem.

11.11 HaarHilbert: 5 targets

Source: `formalization/lean/KomorebiPaperV/HaarHilbert.lean`.

`KomorebiPaperV.HaarHilbert.inverse_low`
Source line 15; theorem.

`KomorebiPaperV.HaarHilbert.inverse_high`
Source line 24; theorem.

`KomorebiPaperV.HaarHilbert.energy`
Source line 33; theorem.

`KomorebiPaperV.HaarHilbert.low_apply`
Source line 42; theorem.

`KomorebiPaperV.HaarHilbert.high_apply`
Source line 47; theorem.

11.12 HilbertPackets: 23 targets

Source: `formalization/lean/KomorebiPaperV/HilbertPackets.lean`.

`KomorebiPaperV.HilbertPackets.projection_apply`
Source line 26; theorem.

`KomorebiPaperV.HilbertPackets.projection_norm_le`
Source line 30; theorem.

`KomorebiPaperV.HilbertPackets.projector_apply`
Source line 53; theorem.

`KomorebiPaperV.HilbertPackets.projector_idempotent`
Source line 56; theorem.

`KomorebiPaperV.HilbertPackets.projector_annihilate`
Source line 62; theorem.

`KomorebiPaperV.HilbertPackets.projector_complete`
Source line 69; theorem.

`KomorebiPaperV.HilbertPackets.projector_refinement`
Source line 79; theorem.

`KomorebiPaperV.HilbertPackets.reindex_norm`
Source line 102; theorem.

`KomorebiPaperV.HilbertPackets.polyphase_apply`
Source line 120; theorem.

`KomorebiPaperV.HilbertPackets.polyphase_synthesis_analysis`
Source line 123; theorem.

`KomorebiPaperV.HilbertPackets.polyphase_analysis_synthesis`
Source line 127; theorem.

`KomorebiPaperV.HilbertPackets.polyphase_energy`
Source line 131; theorem.

`KomorebiPaperV.HilbertPackets.impulse_expansion`

Source line 135; theorem.

`KomorebiPaperV.HilbertPackets.projector_symmetric`
Source line 139; theorem.

`KomorebiPaperV.HilbertPackets.projector_orthogonal`
Source line 147; theorem.

`KomorebiPaperV.HilbertPackets.impulse_basis_apply`
Source line 161; theorem.

`KomorebiPaperV.HilbertPackets.impulse_dense_span`
Source line 165; theorem.

`KomorebiPaperV.HilbertPackets.shift_apply`
Source line 175; theorem.

`KomorebiPaperV.HilbertPackets.reindex_single`
Source line 177; theorem.

`KomorebiPaperV.HilbertPackets.shift_impulse`
Source line 183; theorem.

`KomorebiPaperV.HilbertPackets.residue_projector_refinement`
Source line 187; theorem.

`KomorebiPaperV.HilbertPackets.shift_generated_dense_span`
Source line 203; theorem.

`KomorebiPaperV.HilbertPackets.finite_impulse_expansion`
Source line 207; theorem.

11.13 IntegerPolyphase: 11 targets

Source: `formalization/lean/KomorebiPaperV/IntegerPolyphase.lean`.

`KomorebiPaperV.IntegerPolyphase.phase_val`
Source line 26; theorem.

`KomorebiPaperV.IntegerPolyphase.address_phase_quotient`
Source line 31; theorem.

`KomorebiPaperV.IntegerPolyphase.address_divmod`
Source line 37; theorem.

`KomorebiPaperV.IntegerPolyphase.phase_address`
Source line 47; theorem.

`KomorebiPaperV.IntegerPolyphase.quotient_address`
Source line 54; theorem.

`KomorebiPaperV.IntegerPolyphase.synthesis_analysis`
Source line 73; theorem.

`KomorebiPaperV.IntegerPolyphase.analysis_synthesis`
Source line 78; theorem.

`KomorebiPaperV.IntegerPolyphase.channel_delay`
Source line 87; theorem.

`KomorebiPaperV.IntegerPolyphase.sample_delay`

Source line 94; theorem.

`KomorebiPaperV.IntegerPolyphase.emod_coarsen`
Source line 106; theorem.

`KomorebiPaperV.IntegerPolyphase.residue_refinement`
Source line 111; theorem.

11.14 KnownEdge: 6 targets

Source: `formalization/lean/KomorebiPaperV/KnownEdge.lean`.

`KomorebiPaperV.known_edge_error`
Source line 17; theorem.

`KomorebiPaperV.estimated_edge_error`
Source line 21; theorem.

`KomorebiPaperV.several_known_edges_error`
Source line 25; theorem.

`KomorebiPaperV.known_edge_norm`
Source line 34; theorem.

`KomorebiPaperV.several_known_edges_norm`
Source line 38; theorem.

`KomorebiPaperV.estimated_edge_norm_bound`
Source line 43; theorem.

11.15 Lifting: 4 targets

Source: `formalization/lean/KomorebiPaperV/Lifting.lean`.

`KomorebiPaperV.lifting_synthesis_analysis`
Source line 21; theorem.

`KomorebiPaperV.lifting_analysis_synthesis`
Source line 25; theorem.

`KomorebiPaperV.lifting_determinant`
Source line 29; theorem.

`KomorebiPaperV.lifting_coordinates_roundtrip`
Source line 39; theorem.

11.16 MaskPerturbation: 9 targets

Source: `formalization/lean/KomorebiPaperV/MaskPerturbation.lean`.

`KomorebiPaperV.MaskPerturbation.jump_measurable`
Source line 14; theorem.

`KomorebiPaperV.MaskPerturbation.symmetric_difference_norm`
Source line 18; theorem.

`KomorebiPaperV.MaskPerturbation.indicator_difference_norm`

Source line 25; theorem.

`KomorebiPaperV.MaskPerturbation.jump_sub_same_set`
Source line 32; theorem.

`KomorebiPaperV.MaskPerturbation.jump_error_decomposition`
Source line 38; theorem.

`KomorebiPaperV.MaskPerturbation.finite_p_perturbation`
Source line 43; theorem.

`KomorebiPaperV.MaskPerturbation.pointwise_sup_bound`
Source line 59; theorem.

`KomorebiPaperV.MaskPerturbation.infinity_perturbation`
Source line 73; theorem.

`KomorebiPaperV.MaskPerturbation.full_signal_perturbation`
Source line 79; theorem.

11.17 Packets: 8 targets

Source: `formalization/lean/KomorebiPaperV/Packets.lean`.

`KomorebiPaperV.packet_idempotent`
Source line 20; theorem.

`KomorebiPaperV.packet_annihilate`
Source line 25; theorem.

`KomorebiPaperV.packet_complete`
Source line 36; theorem.

`KomorebiPaperV.packet_refinement`
Source line 48; theorem.

`KomorebiPaperV.address_synthesize_analyze`
Source line 74; theorem.

`KomorebiPaperV.address_analyze_synthesize`
Source line 79; theorem.

`KomorebiPaperV.residuePacket_idempotent`
Source line 91; theorem.

`KomorebiPaperV.residuePacket_complete`
Source line 95; theorem.

11.18 PruningAlgorithm: 8 targets

Source: `formalization/lean/KomorebiPaperV/PruningAlgorithm.lean`.

`KomorebiPaperV.PruningAlgorithm.solve_correct`
Source line 36; theorem.

`KomorebiPaperV.PruningAlgorithm.solve_comparisons`
Source line 52; theorem.

`KomorebiPaperV.PruningAlgorithm.solve_visits`

Source line 61; theorem.

`KomorebiPaperV.PruningAlgorithm.solve_additions`
Source line 69; theorem.

`KomorebiPaperV.PruningAlgorithm.internal_le_nodes`
Source line 81; theorem.

`KomorebiPaperV.PruningAlgorithm.solve_linear_work`
Source line 89; theorem.

`KomorebiPaperV.PruningAlgorithm.computed_antichain_optimal`
Source line 97; theorem.

`KomorebiPaperV.PruningAlgorithm.antichain_probability`
Source line 102; theorem.

11.19 PruningSums: 15 targets

Source: `formalization/lean/KomorebiPaperV/PruningSums.lean`.

`KomorebiPaperV.PruningSums.node_sum_branch`
Source line 41; theorem.

`KomorebiPaperV.PruningSums.mem_selectedNodes`
Source line 51; theorem.

`KomorebiPaperV.PruningSums.cutSum_eq_nodeSum`
Source line 54; theorem.

`KomorebiPaperV.PruningSums.pruning_nodeSum`
Source line 78; theorem.

`KomorebiPaperV.PruningSums.antichain_synthesis`
Source line 81; theorem.

`KomorebiPaperV.PruningSums.packet_projector_synthesis`
Source line 88; theorem.

`KomorebiPaperV.PruningSums.selected_global_minimum`
Source line 102; theorem.

`KomorebiPaperV.PruningSums.antichain_perturbation`
Source line 108; theorem.

`KomorebiPaperV.PruningSums.antichain_envelope_attained`
Source line 115; theorem.

`KomorebiPaperV.PruningSums.antichain_selector_stable`
Source line 120; theorem.

`KomorebiPaperV.PruningSums.objective_minimum`
Source line 134; theorem.

`KomorebiPaperV.PruningSums.objective_positive_gap`
Source line 138; theorem.

`KomorebiPaperV.PruningSums.objective_stability`
Source line 148; theorem.

`KomorebiPaperV.PruningSums.singleton_selector`

Source line 158; theorem.

`KomorebiPaperV.PruningSums.objective_gap_attained`
Source line 165; theorem.

11.20 SmithAction: 10 targets

Source: `formalization/lean/KomorebiPaperV/SmithAction.lean`.

`KomorebiPaperV.SmithAction.action_mul`
Source line 16; theorem.

`KomorebiPaperV.SmithAction.action_one`
Source line 20; theorem.

`KomorebiPaperV.SmithAction.action_scale`
Source line 23; theorem.

`KomorebiPaperV.SmithAction.action_nonzero`
Source line 28; theorem.

`KomorebiPaperV.SmithAction.half_turn`
Source line 36; theorem.

`KomorebiPaperV.SmithAction.half_turn_zero`
Source line 39; theorem.

`KomorebiPaperV.SmithAction.half_turn_infinity`
Source line 42; theorem.

`KomorebiPaperV.SmithAction.homogeneous_chart`
Source line 49; theorem.

`KomorebiPaperV.SmithAction.chart_composition`
Source line 55; theorem.

`KomorebiPaperV.SmithAction.finite_half_turn`
Source line 66; theorem.

11.21 SmithCyclic: 12 targets

Source: `formalization/lean/KomorebiPaperV/SmithCyclic.lean`.

`KomorebiPaperV.SmithCyclic.omega_ne_zero`
Source line 10; theorem.

`KomorebiPaperV.SmithCyclic.omega_pow`
Source line 12; theorem.

`KomorebiPaperV.SmithCyclic.omega_two`
Source line 19; theorem.

`KomorebiPaperV.SmithCyclic.action_respects_scaling`
Source line 45; theorem.

`KomorebiPaperV.SmithCyclic.projective_mul`
Source line 56; theorem.

`KomorebiPaperV.SmithCyclic.projective_one`

Source line 65; theorem.

`KomorebiPaperV.SmithCyclic.cyclic_zero`
Source line 75; theorem.

`KomorebiPaperV.SmithCyclic.cyclic_composition`
Source line 78; theorem.

`KomorebiPaperV.SmithCyclic.cayley_conjugacy`
Source line 87; theorem.

`KomorebiPaperV.SmithCyclic.cyclic_chart`
Source line 92; theorem.

`KomorebiPaperV.SmithCyclic.two_phase_swap`
Source line 97; theorem.

`KomorebiPaperV.SmithCyclic.two_phase_impedance_admittance`
Source line 100; theorem.

11.22 WaveletPackets: 7 targets

Source: `formalization/lean/KomorebiPaperV/WaveletPackets.lean`.

`KomorebiPaperV.WaveletPackets.low_norm_bound`
Source line 35; theorem.

`KomorebiPaperV.WaveletPackets.high_norm_bound`
Source line 39; theorem.

`KomorebiPaperV.WaveletPackets.child_energy`
Source line 53; theorem.

`KomorebiPaperV.WaveletPackets.packet_energy`
Source line 68; theorem.

`KomorebiPaperV.WaveletPackets.energy_implies_adjoint_identity`
Source line 80; theorem.

`KomorebiPaperV.WaveletPackets.packet_operator_reconstruction`
Source line 94; theorem.

`KomorebiPaperV.WaveletPackets.packet_reconstruction`
Source line 98; theorem.

11.23 ZTransform: 5 targets

Source: `formalization/lean/KomorebiPaperV/ZTransform.lean`.

`KomorebiPaperV.ZTransform.transform_unique`
Source line 18; theorem.

`KomorebiPaperV.ZTransform.delay_transform`
Source line 21; theorem.

`KomorebiPaperV.ZTransform.finite_transfer_transform`
Source line 34; theorem.

`KomorebiPaperV.ZTransform.fir_channel_transform`

Source line 49; theorem.

`KomorebiPaperV.ZTransform.fir_matrix_transform`
Source line 64; theorem.

12 Source architecture and exact dependency pins

The import list below is extracted from the 25 Lean files: the 23 proof modules, the aggregate import file, and the audit driver. Imports beginning with `Mathlib` are external mathematical dependencies. Imports beginning with `KomorebiPaperV` are local. The import graph describes compilation dependencies, which can be broader than the logical dependency of an individual theorem.

12.1 KernelAudit.lean

`KomorebiPaperV`

12.2 KomorebiPaperV.lean

`KomorebiPaperV.Packets`, `KomorebiPaperV.KnownEdge`, `KomorebiPaperV.Lifting`, `KomorebiPaperV.FiniteSelection`, `KomorebiPaperV.BinaryAdaptive`, `KomorebiPaperV.IntegerPolyphase`, `KomorebiPaperV.FIRTransfer`, `KomorebiPaperV.FiniteAdaptive`, `KomorebiPaperV.FiniteTransforms`, `KomorebiPaperV.HilbertPackets`, `KomorebiPaperV.CompletePrunings`, `KomorebiPaperV.CRT`, `KomorebiPaperV.SmithAction`, `KomorebiPaperV.HaarHilbert`, `KomorebiPaperV.SmithCyclic`, `KomorebiPaperV.CyclicSeries`, `KomorebiPaperV.HaarChannels`, `KomorebiPaperV.PruningSums`, `KomorebiPaperV.MaskPerturbation`, `KomorebiPaperV.WaveletPackets`, `KomorebiPaperV.PruningAlgorithm`, `KomorebiPaperV.ZTransform`, `KomorebiPaperV.FilterBank`

12.3 KomorebiPaperV/BinaryAdaptive.lean

`Mathlib.Basic.Real.Basic`, `Mathlib.Tactic`

12.4 KomorebiPaperV/CRT.lean

`Mathlib.Data.Nat.Prime.Basic`, `KomorebiPaperV.Packets`

12.5 KomorebiPaperV/CompletePrunings.lean

`KomorebiPaperV.FiniteAdaptive`

12.6 KomorebiPaperV/CyclicSeries.lean

`Mathlib.RingTheory.PowerSeries.Basic`, `Mathlib.Algebra.BigOperators.Ring.Finset`, `Mathlib.Algebra.BigOperators.Fin`, `Mathlib.Tactic`

12.7 KomorebiPaperV/FIRTransfer.lean

`KomorebiPaperV.IntegerPolyphase`

12.8 KomorebiPaperV/FilterBank.lean

`KomorebiPaperV.ZTransform`, `Mathlib.Algebra.MonoidAlgebra.Basic`, `Mathlib.LinearAlgebra.Matrix.ToLin`

12.9 KomorebiPaperV/FiniteAdaptive.lean

Mathlib.MeasureTheory.Measure.Basic, KomorebiPaperV.FiniteSelection

12.10 KomorebiPaperV/FiniteSelection.lean

Mathlib.Basic.Real.Basic, Mathlib.Tactic

12.11 KomorebiPaperV/FiniteTransforms.lean

Mathlib.Analysis.Real.Sqrt, KomorebiPaperV.Lifting, KomorebiPaperV.Packets

12.12 KomorebiPaperV/HaarChannels.lean

KomorebiPaperV.HaarHilbert

12.13 KomorebiPaperV/HaarHilbert.lean

KomorebiPaperV.HilbertPackets, Mathlib.Analysis.Real.Sqrt

12.14 KomorebiPaperV/HilbertPackets.lean

Mathlib.Analysis.InnerProductSpace.l2Space, Mathlib.Analysis.Complex.Basic, Mathlib.Analysis.Normed.Lp.lpSpace, KomorebiPaperV.IntegerPolyphase

12.15 KomorebiPaperV/IntegerPolyphase.lean

KomorebiPaperV.Packets

12.16 KomorebiPaperV/KnownEdge.lean

Mathlib.Analysis.Normed.Group.Basic, Mathlib.Tactic

12.17 KomorebiPaperV/Lifting.lean

Mathlib.LinearAlgebra.Matrix.Determinant.Basic, Mathlib.LinearAlgebra.Matrix.Notation, Mathlib.Tactic

12.18 KomorebiPaperV/MaskPerturbation.lean

Mathlib.MeasureTheory.Function.LpSeminorm.Indicator, Mathlib.MeasureTheory.Function.LpSeminorm.TriangleInequality, Mathlib.Basic.Complex.Basic, Mathlib.Tactic

12.19 KomorebiPaperV/Packets.lean

Mathlib.Data.ZMod.Basic, Mathlib.Tactic

12.20 KomorebiPaperV/PruningAlgorithm.lean

KomorebiPaperV.PruningSums

12.21 KomorebiPaperV/PruningSums.lean

KomorebiPaperV.CompletePrunings, KomorebiPaperV.HilbertPackets

12.22 KomorebiPaperV/SmithAction.lean

Mathlib.Basic.Complex.Basic, Mathlib.Tactic

12.23 KomorebiPaperV/SmithCyclic.lean

KomorebiPaperV.SmithAction, Mathlib.Analysis.SpecialFunctions.Trigonometric.Basic

12.24 KomorebiPaperV/WaveletPackets.lean

KomorebiPaperV.HaarChannels, Mathlib.Analysis.InnerProductSpace.Adjoint

12.25 KomorebiPaperV/ZTransform.lean

KomorebiPaperV.FIRTransfer, Mathlib.Analysis.SpecialFunctions.Pow.Complex, Mathlib.LinearAlgebra.Matrix.ToLin

Transfer Semigroups and Memory

The formal companion to Chapter VI explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 1993

Return to Chapter VI

How to use this volume

A guide to the actual formal development. 33 scoped manuscript results. 341 supporting theorem targets. 47 proof modules. The eleven chapters explain the mathematical interfaces and how to reproduce their recorded evidence. The complete ledgers then give exact declarations, source locations, imports, and nine dependency pins. This guide preserves 48 exact Lean excerpts. The broader traditional development has 142 theorem-like result records and 140 written proof bodies across overlapping branches. Its extra written results do not extend the 33-result formal contract. Three continuum conjectures remain open. Kernel execution evidence comes from the recorded thirty-three-result formal-core build. Source identities, declaration locations, axiom records and finite computations are tracked separately. This reading guide does not claim a new kernel execution.

1 How to read the formal companion

The Lean project formalizes the explicitly scoped mathematical core of Chapter VI. It turns an equation such as $Ce^{tA} = e^{tB}C$ into a statement with named spaces, scalar fields, operator types, and hypotheses. This guide follows those interfaces. It explains how the formal pieces fit together and where the mathematical text goes beyond them.

The formal ledger records all 33 scoped results as complete. The source contract names 341 explicit theorem declarations across 47 proof modules. The saved compiler and namespace audit reports record that 782 declarations were inspected in the project namespace. These counts measure manuscript endpoints, named theorem declarations, files and the wider namespace. None counts independent new mathematical discoveries.

The traditional monograph preserves the larger written development in Parts I through III, followed by the worked atlas in Part IV. Part I gives the formal core. Part II retains the strengthened exposition. Part III retains earlier results not subsumed by that exposition and the bounded Banach operator material migrated from Chapter IV. The theorem-like result records across these parts include conceptual overlap. Formal completion remains attached to the 33 specific Part I ledger entries.

Read one endpoint through its whole interface

Start with the statement and proof in the traditional volume. Locate its numbered row in the 33-result ledger. Read the surrounding variables in the original Lean module, then compare the declaration's hypotheses and conclusion with the mathematical statement. Finally, identify whether the evidence is an inherited kernel record, a current source reconciliation, or a bounded computation. These steps answer different questions.

1.1 Three forms of evidence

Evidence	What it establishes here
Inherited kernel evidence	The saved build and axiom audit for the pinned source and de- pendencies. This guide does not claim a new Lean compilation.
Recorded source and computational checks	Source hashes, target inventories, verifier controls, and available finite examples were rerun in the recorded monograph review. These are not kernel checking or universal proofs.
Mathematical correspondence	The relationship between prose, hypotheses, definitions, ordinary real/complex specializations, and theorem interfaces is a manual review responsibility.

The source report explicitly records `full_paper_formalized: false`. That remains correct even though `all_scoped_results_complete: true` is recorded for the recorded kernel run. Figures, explanatory prose, broader extensions, and three open conjectures are not swallowed by the scoped denominator.

1.2 A practical reading route

Read the matching traditional argument first, use the visual guide to inspect a finite example, and then find its ledger entry in Section 12. Follow the named endpoint to a source excerpt or the complete declaration catalog. Check the ambient variables at the top of the original module before attempting to reuse an excerpt. A snippet does not repeat every `variable`, `open`, instance, or imported notation in its file.

All displayed Lean blocks are exact contiguous source excerpts from the preserved production modules. Raw UTF-8 copies and source hashes accompany the editable guide. Font fallback in the PDF changes rendering only. The snippets are reading excerpts, not independent programs that can be compiled without their original context.

1.3 The notation that matters most

Lean notation	Mathematical reading
`Matrix I J ℂ`	A complex matrix with rows indexed by I and columns by J. Different index types permit rectangular maps.
`E →L[ℂ] F`	A bounded complex linear map from E to F. Completeness and inner-product assumptions come from ambient instances.
`C.comp A`	Composition $C \circ A$: apply A first, then C.
`Module.End ℂ V`	A complex linear endomorphism of V. Finite dimension is a separate hypothesis.
`IntegrableOn f (Ioi 0)`	Bochner integrability of f over $(0, \infty)$ with the stated measure, here ordinarily Lebesgue measure.
`Tendsto f atTop (𝓝 0)`	The limit of $f(t)$ is zero as t tends to $+\infty$.
`0 ≤ A`	Positivity in the operator or C^*-algebra order in the fractional modules. This is not entrywise matrix positivity.
`∀ i j, 0 ≤ K i j`	Entrywise nonnegativity of a finite real matrix.
`Ring.inverse`	A total inverse operation. The surrounding proof must establish the required invertibility before an inverse formula has its intended analytic meaning.

1.4 Common mistakes that the interfaces prevent

Column-stochastic transition matrices act on probability columns. Observable propagation uses the transpose. Probability aggregation is a rectangular mass-summing map. A weighted conditional expectation is a square projector on observables. Their identities are related but their types differ. Similarly, e^{tA} describes a general generator, while a positive fractional operator produces $e^{-tA^{\alpha}}$. The minus sign changes the Duhamel defect in the fractional semigroup theorem.

Zero evidence, one coarse packet, a zero-dimensional space, a zero spectral value, and a nonnormal hidden generator are not interchangeable edge cases. Each belongs to a particular interface. This guide identifies the relevant hypothesis at the point where it is used.

2 Semigroups and rectangular defects

The algebraic layer begins with finite packet blocks P_bAP_a in a ring. It proves reconstruction, multiplication through intermediate packets, and telescoping formulas without assuming stochasticity or self-adjointness. The analytic layer then uses the actual algebra exponential. The definition is short because convergence and differentiation are supplied by the imported analysis library and the ambient Banach-algebra structure.

Source: KomorebiPaperVI/Semigroups.lean.

```
noncomputable def transfer (A : R) (t : ℝ) : R := exp (t • A)
```

Source: KomorebiPaperVI/Semigroups.lean.

```
theorem transfer_add (A : R) (s t : ℝ) : transfer A (s+t) = transfer A s*transfer A t := by
  let : NormedAlgebra ℚ R := NormedAlgebra.restrictScalars ℚ ℝ R
  dsimp [transfer]
  rw [add_smul]
  exact exp_add_of_commute (((Commute.refl A).smul_left s).smul_right t)
```

2.1 Reconstruction is stronger than a diagram

If $\sum_a P_a = 1$, every input/output block participates in reconstruction. Idempotence supplies the block product identity. Pairwise reduction requires additional hypotheses on the off-diagonal generator blocks. It does not follow merely from writing A in blocks. The endpoint for core 2.1 uses the actual exponential law:

Source: KomorebiPaperVI/Semigroups.lean.

```
theorem exponential_packet_convolution {I : Type*} [Fintype I]
    (P : I → R) (hP : ∑ i, P i=1) (hid : ∀ i, P i*P i=P i)
    (A : R) (s t : ℝ) (b a : I) :
    Algebra.block P (transfer A (s+t)) b a =
      ∑ c, Algebra.block P (transfer A s) b c*Algebra.block P (transfer A t) c a :=
  Algebra.packet_semigroup_convolution P hP hid (transfer A) (transfer_add A) s t b a
```

For core 2.2, `ReducingPackets.generator_commutes` derives commutation from vanishing off-diagonal generator blocks. The subsequent theorems prove vanishing exponential cross blocks and equality with the compressed evolution. A projection that is not reducing can produce a memory term when its complement is eliminated. The later memory chapter explains why.

2.2 The domain of the refinement map is explicit

Let $A : E \to E$, $B : F \to F$, and $C : E \to F$. Define $D = CA - BC$. The real Banach-space theorem differentiates $e^{(t-r)B}Ce^{rA}$ and integrates that derivative. Rectangular types enforce the order of every composition.

Source: `KomorebiPaperVI/Semigroups.lean`.

```
theorem duhamel_rectangular (A : E →L[ℝ] E) (B : F →L[ℝ] F)
    (C : E →L[ℝ] F) (t : ℝ) :
    C.comp (transfer A t)-(transfer B t).comp C =
      ∫ r in 0..t, (transfer B (t-r)).comp ((C.comp A-B.comp C).comp (transfer A r)) := by
  have hc : Continuous (fun r => (transfer B (t-r)).comp
      ((C.comp A-B.comp C).comp (transfer A r))) := by
    fun_prop
  have h := intervalIntegral.integral_eq_sub_of_hasDerivAt
    (fun r _ => duhamel_integrand_derivative A B C t r) (hc.intervalIntegrable 0 t)
  simpa [transfer_zero, ContinuousLinearMap.one_def] using h.symm
```

The identity itself is valid for every real t as an oriented interval integral. Bounds such as $\|Ce^{tA} - e^{tB}C\| \le t\|D\|$ additionally assume $t \ge 0$ and contraction estimates for both semigroups. The general growth estimate lives in `OperatorBounds.growth_duhamel_bound`. The exact case specializes the same identity to $D = 0$:

Source: `KomorebiPaperVI/Semigroups.lean`.

```
theorem exact_rectangular_intertwining (A : E →L[ℝ] E) (B : F →L[ℝ] F)
    (C : E →L[ℝ] F) (h : C.comp A=B.comp C) (t : ℝ) :
    C.comp (transfer A t)=(transfer B t).comp C := by
  have hd := duhamel_rectangular A B C t
  simp only [h, sub_self, ContinuousLinearMap.zero_comp, ContinuousLinearMap.comp_zero,
    intervalIntegral.integral_zero] at hd
  exact sub_eq_zero.mp hd
```

2.3 Discrete time and resolvents

Core 5.2 is an exact finite sum before a norm estimate is applied. Horizon zero is included by the empty sum. Its source uses ring-valued literal matrices with independently typed row and column spaces:

Source: `KomorebiPaperVI/Algebra.lean`.

```
theorem finite_horizon_defect (C : Matrix m n R) (A : Matrix n n R) (B : Matrix m m R)
    (k : ℕ) :
    C*A^k-B^k*C = ∑ j ∈ Finset.range k, B^(k-1-j)*(C*A-B*C)*A^j := by
  induction k with
  | zero => simp
  | succ k ih =>
    rw [Finset.sum_range_succ]
    have hs : (∑ j ∈ Finset.range k, B^(k+1-1-j)*(C*A-B*C)*A^j) =
        B*(∑ j ∈ Finset.range k, B^(k-1-j)*(C*A-B*C)*A^j) := by
      rw [Matrix.mul_sum]
```

```
      apply Finset.sum_congr rfl
      intro j hj
      have hjk := Finset.mem_range.mp hj
      have he : k+1-1-j = (k-1-j)+1 := by omega
      rw [he, pow_succ']
      simp only [Matrix.mul_assoc]
    rw [hs, ← ih]
    simp only [Nat.add_sub_cancel, Nat.sub_self, pow_zero, Matrix.one_mul]
    simp only [pow_succ', Matrix.mul_sub, Matrix.sub_mul, Matrix.mul_assoc]
    abel
```

The resolvent identity requires one inverse equation on each side. This is deliberate: the proof does not invoke a vague invertibility claim when only the relevant cancellations are needed. The scalar must commute with the entries of C in the general ring statement.

Source: `KomorebiPaperVI/Algebra.lean`.

```
theorem rectangular_resolvent_identity (C : Matrix m n R)
    (A : Matrix n n R) (B : Matrix m m R)
    (Rf : Matrix n n R) (Rc : Matrix m m R) (z : R)
    (hz : ∀ i j, z*C i j = C i j*z)
    (hf : (Matrix.scalar n z-A)*Rf=1)
    (hc : Rc*(Matrix.scalar m z-B)=1) :
    C*Rf-Rc*C=Rc*(C*A-B*C)*Rf := by
  have hzC : Matrix.scalar m z*C=C*Matrix.scalar n z := by
    ext i j
    simpa using hz i j
  have hd : C*A-B*C=(Matrix.scalar m z-B)*C-C*(Matrix.scalar n z-A) := by
    rw [Matrix.sub_mul, Matrix.mul_sub, hzC]
    abel
  rw [hd, Matrix.mul_sub, Matrix.sub_mul]
  simp only [← Matrix.mul_assoc, hc, Matrix.one_mul]
  rw [Matrix.mul_assoc (Rc*C), hf, Matrix.mul_one]
```

For ordinary complex matrices the scalar commutation is automatic. For core 3.3 the specified inverse equations are supplied by the actual resolvents. The broad Banach-space and contour extensions in the traditional chapter have their own domains and hypotheses. The literal matrix interface is the formal endpoint recorded in this ledger entry.

3 Markov mass and conditional expectation

3.1 Column conservation

`MarkovSemigroup` proves nonnegativity of the actual matrix exponential by shifting a Metzler generator by a scalar identity, expanding a nonnegative matrix exponential, and then undoing the scalar shift. Conservation follows from zero column sums of the generator. The final probability statement combines these two independent facts.

Source: `KomorebiPaperVI/MarkovSemigroup.lean`.

```
theorem markov_probability (Q : Matrix I I ℝ) (hpos : ∀ i j, i≠j → 0 ≤ Q i j)
    (hcol : ∀ j, ∑ i, Q i j=0) (p : I → ℝ) (hp : ∀ i, 0 ≤ p i)
    (hpsum : ∑ i, p i=1) (t : ℝ) (ht : 0 ≤ t) :
    (∀ i, 0 ≤ (exp (t • Q)).mulVec p i) ∧ ∑ i, (exp (t • Q)).mulVec p i=1 := by
  constructor
  · intro i
    exact Finset.sum_nonneg (fun j _ => mul_nonneg (markov_nonnegative Q hpos t ht i j) (hp j))
  · simp only [Matrix.mulVec,dotProduct,Finset.sum_comm,← Finset.sum_mul,markov_column_sum Q hcol,
      one_mul,hpsum]
```

Here Q_{ij} is the rate from state j to state i. Off-diagonal nonnegativity and zero column sums are the generator hypotheses. The condition $t \geq 0$ is essential for the positivity argument. Exact discrete lumpability is core 5.1's rectangular identity $CK^n = \bar{K}^n C$. The complete quotient obstruction and detailed coordinate characterization in Part II are related written results, not additional completed Lean ledger entries.

3.2 Weighted projection on observables

For a partition map $f : I \to A$, `Coarsening.coarsen` sums probability mass over fibers. In contrast, `ConditionalExpectation.expect` returns a function on the original fine space. It averages an observable within each fiber with respect to finite weights p_i. Strict positivity guarantees that the occupied fiber denominator at each fine point is nonzero.

Source: `KomorebiPaperVI/ConditionalExpectation.lean`.

```
theorem expect_idempotent (f : I → A) (p : I → ℝ) (hp : ∀ i, 0 < p i) (v : I → ℂ) :
    expect f p (expect f p v)=expect f p v :=
  expect_constant f p hp (average f p v)
```

The source also proves that the range is exactly the block-constant functions and that the expectation equals a literal kernel matrix applied to the observable. Self-adjointness is with respect to the weighted inner product, not ordinarily the unweighted Euclidean inner product:

Source: `KomorebiPaperVI/ConditionalExpectation.lean`.

```
theorem expect_self_adjoint (f : I → A) (p : I → ℝ) (v w : I → ℂ) :
    weightedInner p (expect f p v) w=weightedInner p v (expect f p w) := by
  rw [expect_matrix,expect_matrix]
  simp only [weightedInner,Matrix.mulVec,dotProduct,Finset.mul_sum,Finset.sum_mul,
```

```
    map_sum,map_mul]
  rw [Finset.sum_comm]
  apply Finset.sum_congr rfl
  intro i _
  apply Finset.sum_congr rfl
  intro j _
  have h := kernel_weighted_symmetry f p j i
  calc
    _ = ((p j:ℂ)*kernel f p j i)*v i*conj (w j) := by ring
    _ = _ := by rw [h]; ring
```

The displayed self-adjoint identity is algebraic and its signature has no positivity premise. Calling the pairing a positive-definite inner product, or deriving range separation from it, uses the strict positivity hypotheses elsewhere in the module. Reuse the endpoint matching the intended claim rather than adding an interpretation that its type does not supply.

3.3 Towers and martingale packets

Composing $f : I \to A$ with $g : A \to B$ makes a coarser partition. Both products of the corresponding conditional kernels equal the coarse kernel:

Source: `KomorebiPaperVI/ConditionalTowers.lean`.

```
theorem conditional_kernel_product (f : I → A) (g : A → B) (p : I → ℝ)
    (hp : ∀ i, 0 < p i) :
    kernel (g ∘ f) p*kernel f p=kernel (g ∘ f) p ∧
      kernel f p*kernel (g ∘ f) p=kernel (g ∘ f) p := by
  constructor
  · apply Matrix.mulVec_injective
    funext v
    rw [← Matrix.mulVec_mulVec,← expect_matrix,← expect_matrix,coarse_after_fine f g p
     ↪ hp,expect_matrix]
  · apply Matrix.mulVec_injective
    funext v
    rw [← Matrix.mulVec_mulVec,← expect_matrix,← expect_matrix,fine_after_coarse f g p
     ↪ hp,expect_matrix]
```

The module extends this to arbitrary pairs in a nested family: the product is the kernel at the coarser index. `MartingalePackets` then supplies the abstract difference-projector algebra, and `ConditionalTowers` applies it to concrete kernels. The differences are pairwise annihilating idempotents and sum to the terminal projector. The formal scope is finite weighted conditional expectation. A general measure-theoretic conditional-expectation tower is a separate classical extension in the traditional exposition.

4 Bayesian normalization, sufficiency, and flow

4.1 An equation is not a probability claim

Lean defines the posterior by a total real division. This makes the expression meaningful as a term even when evidence is zero. Theorems asserting unit mass or probability preservation require nonzero or positive evidence explicitly. The basic definition is:

Source: KomorebiPaperVI/FiniteBayes.lean.

```
noncomputable def update (p l : I → ℝ) (i : I) : ℝ := l i*p i / evidence p l
def l1 (p : I → ℝ) : ℝ := ∑ i, |p i|
```

Core 7.1 proves composition of two likelihood updates with both the first evidence and the combined evidence nonzero. The proof rewrites sequential evidence and then performs field algebra.

Source: KomorebiPaperVI/FiniteBayes.lean.

```
theorem posterior_composition (p l k : I → ℝ) (hz : evidence p l ≠ 0)
    (hzk : evidence p (fun i => l i*k i) ≠ 0) :
    update (update p l) k = update p (fun i => l i*k i) := by
  funext i
  rw [update, sequential_evidence]
  dsimp [update]
  field_simp
```

An application with nonnegative priors and likelihoods must establish positive evidence before using probability interpretations. Normalization does not manufacture an observation with positive probability.

4.2 Perturbation has a denominator

The weighted ℓ^1 likelihood error is divided by the minimum of the two positive evidences. This gives a usable stability estimate while exposing its sensitivity to nearly impossible observations.

Source: KomorebiPaperVI/FiniteBayes.lean.

```
theorem likelihood_perturbation_bound (p l k : I → ℝ)
    (hp : ∀ i, 0 ≤ p i) (_hl : ∀ i, 0 ≤ l i) (hk : ∀ i, 0 ≤ k i)
    (hZl : 0 < evidence p l) (hZk : 0 < evidence p k) :
    l1 (fun i => update p l i-update p k i) ≤
      2*(∑ i, |l i-k i| *p i)/min (evidence p l) (evidence p k) := by
  have he : l1 (fun i => l i*p i-k i*p i) = ∑ i, |l i-k i| *p i := by
    simp only [l1, ← sub_mul, abs_mul, abs_of_nonneg (hp _)]
  have h := normalized_l1_bound_left (fun i => l i*p i) (fun i => k i*p i)
    (fun i => mul_nonneg (hk i) (hp i)) hZl hZk
  change l1 (fun i => update p l i-update p k i) ≤
    2*l1 (fun i => l i*p i-k i*p i)/evidence p l at h
  rw [he] at h
  exact h.trans (div_le_div_of_nonneg_left
    (mul_nonneg (by norm_num) (Finset.sum_nonneg (fun i _ => mul_nonneg (abs_nonneg _) (hp i))))
    (lt_min hZl hZk) (min_le_left _ _))
```

The finite numerical examples can check this inequality for chosen data. They do not replace the universally quantified source theorem or supply an evidence lower bound for a different model.

4.3 Effective likelihood and fixed coarse likelihood

Given a prior p, the effective coarse likelihood averages the fine likelihood within each fiber using that prior. The factorization theorem includes zero-prior-mass fibers. The proof separates these cases rather than silently dividing by their mass.

Source: `KomorebiPaperVI/Coarsening.lean`.

```
theorem effective_factorization (f : I → A) (p l : I → ℝ) (hp : ∀ i, 0 ≤ p i) :
    coarsen f (update p l) = update (coarsen f p) (effective f p l) := by
  rw [coarsen_posterior]
  funext a
  simp only [update, effective_weight f p l hp, effective_evidence f p l hp]
```

The effective likelihood depends on the prior. A single prescribed coarse likelihood that works for every prior is a stronger requirement. The necessary-and-sufficient theorem derives that requirement using actual two-point priors. With at least two coarse states, surjectivity, and strict positivity, fine likelihoods must agree with the pulled-back coarse likelihood up to one positive global scale:

Source: `KomorebiPaperVI/BayesianNecessity.lean`.

```
theorem refinement_iff (f : I → A) (hf : Function.Surjective f) [Nontrivial A]
    (l : I → ℝ) (k : A → ℝ) (hl : ∀ i, 0 < l i) (hk : ∀ a, 0 < k a) :
    AllPriorCommutes f l k ↔ ∃ c : ℝ, 0 < c ∧ ∀ i, l i=c*k (f i) := by
  constructor
  · intro h
    obtain ⟨a⟩ := (inferInstance : Nonempty A)
    obtain ⟨i,hi⟩ := hf a
    refine ⟨l i/k (f i), div_pos (hl i) (hk (f i)), ?_⟩
    intro j
    have he := all_packet_ratio f hf l k hl hk h j i
    apply (eq_div_iff (ne_of_gt (hk (f i)))).mpr at he
    calc
      l j = (l i*k (f j))/k (f i) := he
      _ = _ := by ring
  · rintro ⟨c,hc,he⟩ p hp
    have heq : l=fun i => c*k (f i) := funext he
    rw [heq]
    exact scaled_blockwise_sufficiency f p k c hc.ne'
      (positive_evidence p (k ∘ f) hp (fun i => hk (f i))).ne'
```

The nontriviality assumption is part of the mathematics. If all mass is sent to one packet, every valid posterior aggregates to the unique unit probability, so commutation no longer forces likelihood constancy:

Source: KomorebiPaperVI/BayesianTowers.lean.

```
theorem singleton_commutes (f : I → A) (l : I → ℝ) (k : A → ℝ)
    (hl : ∀ i, 0 < l i) (hk : ∀ a, 0 < k a) [Subsingleton A] :
    AllPriorCommutes f l k := by
  intro p hp
  funext a
  have hs (v : A → ℝ) : mass v=v a :=
    Finset.sum_eq_single a (fun b _ hb => False.elim (hb (Subsingleton.elim b a))) (by simp)
  have h1 : mass (coarsen f (update p l))=1 :=
    (coarsen_mass f _).trans (posterior_mass p l (positive_evidence p l hp hl).ne')
  have h2 := posterior_mass (coarsen f p) k
    (positive_evidence _ k (coarsen_probability f p hp) hk).ne'
  exact (hs _).symm.trans (h1.trans (h2.symm.trans (hs _)))
```

The adjacent tower criterion includes this singleton alternative. Composition then gives exact commutation over a finite span of varying state spaces.

4.4 The coarsest meaningful partition

`BayesianPartitions` forms a quotient by equality of the whole likelihood profile. Its canonical coarse likelihood recovers each fine likelihood. Under strict positivity, every nontrivial all-prior-closed partition refines the profile partition: the canonical profile map factors uniquely through the given partition map:

Source: KomorebiPaperVI/BayesianPartitions.lean.

```
theorem canonical_coarsest_nontrivial (l : Y → I → ℝ) (hl : ∀ y i, 0 < l y i)
    (f : I → A) (hf : Function.Surjective f) [Nontrivial A] (h : FamilyCloses l f) :
    ∃! u : A → Packet l, ∀ i, u (f i)=packetMap l i :=
  canonical_unique_factor l f hf ((all_prior_family_iff_resolves l hl f hf).mp h)
```

The qualifier “nontrivial” excludes the automatic one-packet case. The theorem does not claim that every family has a nontrivial canonical quotient: a separate declaration supplies nontriviality when two profiles differ. A broader statistical sufficiency statement requires its own statistical model and hypotheses.

4.5 Information loss is a specific matrix

The formal information-loss matrix is $\mathrm{diag}(P) - bP^{\top}$, where P is coarse prior mass and b is the coarse posterior. It acts on the mismatch between the effective and prescribed coarse likelihoods. The exact factorization is:

Source: KomorebiPaperVI/InformationLoss.lean.

```
theorem information_loss_matrix (f : I → A) (p l : I → ℝ) (k : A → ℝ)
    (hp : ∀ i, 0 ≤ p i) (hz : evidence p l ≠ 0)
    (hk : evidence (coarsen f p) k ≠ 0) (a : A) :
    coarsen f (update p l) a-update (coarsen f p) k a =
      (lossMatrix (coarsen f p) (update (coarsen f p) k)).mulVec
```

```
      (fun j => effective f p l j-k j) a/evidence p l := by
  rw [effective_factorization f p l hp, lossMatrix_apply]
  have he := posterior_difference (coarsen f p) (effective f p l) k
    (by rwa [effective_evidence f p l hp]) hk a
  simpa only [effective_evidence f p l hp] using he
```

The scale direction is in its nullspace because multiplying a likelihood by a nonzero scalar does not change its normalized posterior. The norm estimate then combines this factorization with the likelihood perturbation bound. This is the matrix-valued defect used in core 8.4. Relative-entropy chain rules in Part II are different claims and are not implied by the shared word "information."

4.6 A normalized exponential flow

For a score function g, $p_i(t)$ is proportional to $p_i(0)e^{tg_i}$. The source establishes positive evidence, simplex preservation, the additive-time law, and the centered multiplicative derivative:

Source: `KomorebiPaperVI/BayesianFlow.lean`.

```
theorem flow_derivative (p g : I → ℝ) (hp : IsProbability p) (t : ℝ) (i : I) :
    HasDerivAt (fun u => flow p g u i)
      ((g i-(∑ j, g j*flow p g t j))*flow p g t i) t := by
  have hz := (positive_evidence p g hp t).ne'
  have hd := ((likelihood_derivative g t i).mul_const (p i)).div
    (evidence_derivative p g t) hz
  convert! hd using 1
  simp only [flow, update]
  simp only [← mul_div_assoc, ← Finset.sum_div]
  field_simp
```

The derivative contains the posterior mean score. It is therefore nonlinear in the probability vector even though unnormalized multiplication is linear. The additional covariance, cumulant, concentration, and filter stability results retained in the mathematical text have separate scope rows in Section 10.

5 Fractional calculus and refinement towers

5.1 Two constructions with different starting data

`SpectralPackets` starts with a supplied finite spectral resolution. Its finite sums give multiplication, packet reduction, and the actual exponential. It does not silently prove a spectral decomposition for every operator. The additive-exponent law explicitly allows zero eigenvalues while requiring positive exponents:

Source: `KomorebiPaperVI/SpectralPackets.lean`.

```
theorem fractional_add (P : I → R) (h : ∀ i j, P i*P j=if i=j then P i else 0)
    (lam : I → ℝ) (hl : ∀ i, 0 ≤ lam i) (a b : ℝ) (ha : 0 < a) (hb : 0 < b) :
    fractional P lam a*fractional P lam b=fractional P lam (a+b) := by
  unfold fractional
  rw [resolve_mul P h]
  congr 1
  funext i
  exact (Real.rpow_add' (hl i) (add_pos ha hb).ne').symm
```

The bounded Hilbert-space refinement result uses continuous functional calculus and a normalized Bochner resolvent integral. This is a different route and has a broader operator domain than a finite spectral list. The ledger records the exact setting for each result.

5.2 Poisson subordination includes the integral

The square-root transfer is recovered from the heat transfer using the normalized Poisson kernel. `ReciprocalGaussian` proves the scalar integral, `PoissonKernel` supplies integrability, and `PoissonSubordination` transports the scalar identities across a finite spectral sum. Its endpoint asserts both integrability and equality with the actual exponential:

Source: `KomorebiPaperVI/PoissonSubordination.lean`.

```
theorem operator_subordination (P : I → R) (hP : ∑ i,P i=1)
    (h : ∀ i j,P i*P j=if i=j then P i else 0) (lam : I → ℝ)
    (hl : ∀ i,0 ≤ lam i) (t : ℝ) (ht : 0 < t) :
    IntegrableOn (fun s => kernel t s • NormedSpace.exp ((-s) • resolve P lam)) (Ioi 0) ∧
    NormedSpace.exp ((-t) • fractional P lam (1/2))=
      ∫ s in Ioi 0,kernel t s • NormedSpace.exp ((-s) • resolve P lam) := by
  simp only [operator_integrand P hP h lam]
  refine ⟨finite_spectral_integrable P lam hl t ht,?_⟩
  rw [finite_spectral_subordination P lam hl t ht,spectral_transfer_exponential P hP h]
  simp only [spectralTransfer,resolve,Real.sqrt_eq_rpow]
```

The integral formula is stated for $t > 0$. The semigroup value at $t = 0$ is obtained separately from the exponential definition. Nonnegative spectral values include zero. General stable-subordinator statements for arbitrary fractional exponents in Part II are not new endpoints of this square-root formal theorem.

5.3 The normalization constant is proved

For $0 < \alpha < 1$, the resolvent representation uses $\sin(\pi\alpha)/\pi$. `BetaNormalization` derives the scalar integrals and constants. In particular, the common-gap integral evaluates exactly to $\alpha m^{\alpha-1}$ when $m > 0$:

Source: `KomorebiPaperVI/BetaNormalization.lean`.

```
theorem normalized_gap_integral (p m : ℝ) (hp : p∈Ioo 0 1) (hm : 0 < m) :
    (Real.sin (Real.pi*p)/Real.pi)*(∫ t in Ioi 0,t^p/(t+m)^2)=p*m^(p-1) := by
  rw [gap_integral p m hp hm]
  have hs : Real.sin (Real.pi*p) ≠ 0 := by
    intro hs
    have h := sine_factor_positive p hp
    rw [hs,zero_div] at h
    exact (lt_irrefl 0) h
  field_simp
```

This identity supplies the sharp constant in the displayed common-gap certificate. It is not left as an unnamed positive normalization factor.

5.4 The rectangular resolvent defect

The spaces E and F in `RectangularFractional` are complete complex Hilbert spaces. A and B are bounded nonnegative self-adjoint operators in the operator order, and $C : E \to F$ is bounded. Put

$$D = CA - BC, \qquad D_\alpha = CA^\alpha - B^\alpha C.$$

The theorem records the actual inverse-valued integrand and its Bochner integrability:

Source: `KomorebiPaperVI/RectangularFractional.lean`.

```
theorem literal_resolvent_integral (p : ℝ≥0) (hp : (p:ℝ)∈Ioo 0 1)
    (A : E →L[ℂ] E) (B : F →L[ℂ] F) (C : E →L[ℂ] F) (hA : 0 ≤ A) (hB : 0 ≤ B) :
    let f := fun t : ℝ => t^(p:ℝ) • (Ring.inverse (algebraMap ℝ (F →L[ℂ] F) t+B)).comp
      ((defect A B C).comp (Ring.inverse (algebraMap ℝ (E →L[ℂ] E) t+A)))
    IntegrableOn f (Ioi 0) ∧ fractionalDefect A B C p=coefficient p • ∫ t in Ioi 0,f t := by
  dsimp only
  have he : (fun t : ℝ => t^(p:ℝ) • (Ring.inverse (algebraMap ℝ (F →L[ℂ] F) t+B)).comp
      ((defect A B C).comp (Ring.inverse (algebraMap ℝ (E →L[ℂ] E) t+A))))
      =ᵐ[volume.restrict (Ioi 0)] integrand A B C p := by
    filter_upwards [ae_restrict_mem measurableSet_Ioi] with t ht
    rw [literal_integrand p t hp ht A B C hA hB,
      PositiveResolvent.resolvent_inverse A hA t ht,PositiveResolvent.resolvent_inverse B hB t ht]
  have h := normalized_integral p hp A B C hA hB
  exact (h.1.congr he.symm,h.2.trans (congrArg (coefficient p • ·) (integral_congr_ae he.symm)))
```

The sign matches the convention $D = CA - BC$. In this formula the resolvents are $(tI + B)^{-1}$ and $(tI + A)^{-1}$, and the order of the sandwich is fixed by the rectangular domain and codomain.

5.5 A gap-free estimate and a common-gap estimate

The gap-free estimate combines low- and high-parameter bounds on the same integrand. A split at the appropriate scale gives a Hölder estimate with the explicit constant defined in `FractionalNormBounds.holderConstant`. The source handles $C = 0$ and $D = 0$ before dividing by their norms.

Source: `KomorebiPaperVI/RectangularFractional.lean`.

```
theorem holder_estimate (p : ℝ≥0) (hp : (p:ℝ)∈Ioo 0 1)
    (A : E →L[ℂ] E) (B : F →L[ℂ] F) (C : E →L[ℂ] F) (hA : 0 ≤ A) (hB : 0 ≤ B) :
    ‖fractionalDefect A B C p‖ ≤
      FractionalNormBounds.holderConstant p*‖C‖^(1-(p:ℝ))*‖defect A B C‖^(p:ℝ) := by
  by_cases hc : C=0
  · subst C
    simp [fractionalDefect,defect,Real.zero_rpow hp.1.ne']
  by_cases hd : defect A B C=0
  · have he := exact_fractional_intertwining p hp A B C hA hB hd
    rw [he,norm_zero,hd]
    simp [Real.zero_rpow hp.1.ne']
  have hcn : 0 < ‖C‖ := norm_pos_iff.mpr hc
  have hdn : 0 < ‖defect A B C‖ := norm_pos_iff.mpr hd
  have hf := normalized_integral p hp A B C hA hB
  have hk : 0 < coefficient p := BetaNormalization.sine_factor_positive p hp
  have hbound := FractionalScalarBounds.split_norm_bound (integrand A B C p) p (2*‖C‖) ‖defect A B
   ↪  C‖
    (‖defect A B C‖/(2*‖C‖)) hp (by positivity) hf.1
    (fun t ht => low_bound p t hp ht A B C hA hB)
    (fun t ht => high_bound p t ht hp A B C hA hB)
  rw [FractionalScalarBounds.optimize_split p (2*‖C‖) ‖defect A B C‖ hp (by positivity) hdn] at
   ↪  hbound
  rw [hf.2,norm_smul,Real.norm_of_nonneg hk.le]
  calc
    _ ≤ coefficient p*((2*‖C‖)^(1-(p:ℝ))*‖defect A B C‖^(p:ℝ)/((p:ℝ)*(1-(p:ℝ)))) :=
      mul_le_mul_of_nonneg_left hbound hk.le
    _ = _ := by
      rw [Real.mul_rpow (by norm_num : (0:ℝ) ≤ 2) (norm_nonneg C)]
      unfold FractionalNormBounds.holderConstant
      ring
```

With a common lower spectral bound $mI \le A, B$ and $m > 0$, the endpoint is linear in the original defect:

Source: `KomorebiPaperVI/RectangularFractional.lean`.

```
theorem common_gap_estimate (p : ℝ≥0) (hp : (p:ℝ)∈Ioo 0 1) (m : ℝ) (hm : 0 < m)
    (A : E →L[ℂ] E) (B : F →L[ℂ] F) (C : E →L[ℂ] F) (hA : 0 ≤ A) (hB : 0 ≤ B)
    (hgA : algebraMap ℝ (E →L[ℂ] E) m ≤ A) (hgB : algebraMap ℝ (F →L[ℂ] F) m ≤ B) :
    ‖fractionalDefect A B C p‖ ≤ (p:ℝ)*m^((p:ℝ)-1)*‖defect A B C‖ := by
  have hd := normalized_integral p hp A B C hA hB
  have hk : 0 < coefficient p := BetaNormalization.sine_factor_positive p hp
  have hs : 0 < Real.sin (Real.pi*(p:ℝ)) := by
    apply Real.sin_pos_of_pos_of_lt_pi
```

```
    · exact mul_pos Real.pi_pos hp.1
    · nlinarith [Real.pi_pos,hp.2]
  have hg : IntegrableOn (fun t : ℝ => t^(p:ℝ)/(t+m)^2) (Ioi 0) := by
    by_contra hn
    have hz := integral_undef hn
    rw [BetaNormalization.gap_integral p m hp hm] at hz
    exact (mul_pos (Real.rpow_pos_of_pos hm _) (div_pos (mul_pos hp.1 Real.pi_pos) hs)).ne' hz
  rw [hd.2,norm_smul,Real.norm_of_nonneg hk.le]
  calc
    _ ≤ coefficient p*(∫ t in Ioi 0,‖integrand A B C p t‖) := by
      apply mul_le_mul_of_nonneg_left (norm_integral_le_integral_norm _) hk.le
    _ ≤ coefficient p*(∫ t in Ioi 0,‖defect A B C‖*(t^(p:ℝ)/(t+m)^2)) := by
      apply mul_le_mul_of_nonneg_left _ hk.le
      apply setIntegral_mono_on hd.1.norm (hg.const_mul _) measurableSet_Ioi
      intro t ht
      exact gap_bound p t m ht hm.le hp A B C hA hB hgA hgB
    _ = _ := by
      rw [integral_const_mul,mul_left_comm,coefficient,BetaNormalization.normalized_gap_integral p
       ↪  m hp hm]
      ring
```

Real finite-matrix applications use the ordinary complexification specialization disclosed in the inherited review. The project does not count that manual specialization as a separately formalized construction.

5.6 The sign in fractional evolution

Fractional evolution is $e^{-tA^{\alpha}}$. Its Duhamel formula has a minus sign relative to the e^{tA} convention. Combining decay with the common-gap estimate yields the final generator-level bound:

Source: `KomorebiPaperVI/RectangularSemigroup.lean`.

```
theorem generator_gap_bound (A : E →L[ℂ] E) (B : F →L[ℂ] F) (C : E →L[ℂ] F)
    (hA : 0 ≤ A) (hB : 0 ≤ B) (p : ℝ≥0) (hp : (p:ℝ)∈Ioo 0 1) (t m : ℝ) (ht : 0 ≤ t) (hm : 0 < m)
    (hgA : algebraMap ℝ (E →L[ℂ] E) m ≤ A) (hgB : algebraMap ℝ (F →L[ℂ] F) m ≤ B) :
    ‖timeDefect A B C p t‖ ≤ t*Real.exp (-t*m^(p:ℝ))*(p:ℝ)*m^((p:ℝ)-1)*‖defect A B C‖ := by
  calc
    _ ≤ t*Real.exp (-t*m^(p:ℝ))*‖fractionalDefect A B C p‖ :=
      fractional_gap_bound A B C hA hB p hp.1 t m ht hm.le hgA hgB
    _ ≤ t*Real.exp (-t*m^(p:ℝ))*((p:ℝ)*m^((p:ℝ)-1)*‖defect A B C‖) :=
      mul_le_mul_of_nonneg_left (common_gap_estimate p hp m hm A B C hA hB hgA hgB) (by
       ↪  positivity)
    _ = _ := by ring
```

The bound vanishes when the generator defect vanishes. It assumes nonnegative time and the same positive common gap as the fractional power estimate. It does not establish norm convergence for arbitrary unbounded generators.

5.7 What is summed across varying spaces

In `RectangularTowers`, H_n may vary with n, and $C_n : H_{n+1} \to H_n$. The individual defects therefore lie in different operator spaces. The formal series is a series of *real norms*. The endpoint says its partial sums converge uniformly on bounded nonnegative time intervals:

Source: `KomorebiPaperVI/RectangularTowers.lean`.

```
theorem bounded_time_uniform_convergence (A : ∀ n,H n →L[ℂ] H n) (C : ∀ n,H (n+1) →L[ℂ] H n)
    (hA : ∀ n,0 ≤ A n) (p : ℝ≥0) (hp : 0 < p) (T : ℝ)
    (hs : Summable (fractionalNorm A C p)) :
    TendstoUniformlyOn (fun N t => ∑ n ∈ Finset.range N,timeNorm A C p t n)
      (fun t => ∑' n,timeNorm A C p t n) Filter.atTop (Icc 0 T) := by
  apply tendstoUniformlyOn_tsum_nat (hs.mul_left T)
  intro n t ht
  rw [Real.norm_of_nonneg (show 0 ≤ timeNorm A C p t n from norm_nonneg _)]
  exact bounded_time_majorant A C hA p hp T t ht n
```

Two sufficient generator criteria precede this theorem: summability of the α-powers of generator-defect norms with uniformly bounded C_n, or ordinary summability with a common positive spectral gap. An inherited fixed-space formulation also proves operator-series convergence. It must not be relabeled as an operator sum across unrelated Hom spaces. A coherent limiting evolution additionally needs the ambient completion and closure hypotheses discussed in the retained written extensions.

6 Green reconstruction and finite paths

6.1 The inverse is the literal inverse

`DiscreteGreen` builds the Dirichlet tridiagonal matrix on `Fin n`. It proves a second-difference identity for the kernel, treats both boundary terms, multiplies the matrix by that kernel, and then identifies the actual nonsingular inverse. The endpoint is short because the preceding multiplication identity carries the main work:

Source: `KomorebiPaperVI/DiscreteGreen.lean`.

```
theorem inverse_eq_kernel (n : ℕ) : (tridiagonal n)⁻¹=inverseKernel n := by
  exact Matrix.inv_eq_right_inv (tridiagonal_inverse_kernel n)
```

For $h = 1/(n+1)$ and $L_h = h^{-2}\,\mathrm{tridiag}(-1, 2, -1)$, the precise scaling is

$$h^{-1}(L_h^{-1})_{ij} = G(x_i, x_j), \qquad G(x, y) = \min(x, y) - xy.$$

The formal statement uses zero-based `Fin n` indices and adds one to obtain the interior nodes:

Source: `KomorebiPaperVI/DiscreteGreen.lean`.

```
theorem grid_reconstruction (n : ℕ) (i j : Fin n) :
    ((n:ℝ)+1)*(laplacian n)⁻¹ i j=
      min (((i.val:ℝ)+1)/((n:ℝ)+1)) (((j.val:ℝ)+1)/((n:ℝ)+1))-
        (((i.val:ℝ)+1)/((n:ℝ)+1))*(((j.val:ℝ)+1)/((n:ℝ)+1)) := by
  rw [laplacian_inverse]
  simp only [Matrix.smul_apply,smul_eq_mul,inverseKernel,green,Nat.cast_add,Nat.cast_one]
  rw [min_div_div_right (by positivity)]
  field_simp
```

This is an exact node identity. It is not by itself a uniform convergence theorem for an arbitrary continuous forcing function or a higher-dimensional mesh. Those quadrature and PDE estimates need the separate regularity assumptions stated in the traditional chapter.

6.2 Low modes have an explicit error bound

`LowModeConvergence` proves a scalar sine-square estimate and substitutes it into the discrete eigenvalue formula. It gives

$$0 \le \omega^2 - \frac{4}{h^2}\sin^2\!\left(\frac{\omega h}{2}\right) \le \frac{\omega^4 h^2}{12}, \qquad h \neq 0.$$

The Lean endpoint retains the nonzero-spacing hypothesis:

Source: `KomorebiPaperVI/LowModeConvergence.lean`.

```
theorem low_mode_error (w h : ℝ) (hh : h≠0) :
    0 ≤ w^2-eigenvalue w h ∧ w^2-eigenvalue w h ≤ w^4/12*h^2 := by
  have hs := sine_square_error (w*h/2)
  have hpos : 0 ≤ 4/h^2 := by positivity
  have he : w^2-eigenvalue w h=4/h^2*((w*h/2)^2-Real.sin (w*h/2)^2) := by
```

```
    unfold eigenvalue
    field_simp
    ring
  rw [he]
  constructor
  · exact mul_nonneg hpos hs.1
  · calc
      _ ≤ 4/h^2*((w*h/2)^4/3) := mul_le_mul_of_nonneg_left hs.2 hpos
      _ = w^4/12*h^2 := by field_simp; ring
```

For a fixed low mode, set $\omega = k\pi$ and let positive h tend to zero. The constant depends on that fixed ω. This statement does not make high-frequency error uniform when k grows with mesh resolution.

6.3 Probability columns and observable rows

For the finite path formula, K_{ji} is the transition probability from i to j. An observable f propagates by $K^\top$, and the potential is evaluated at the departure state. Therefore the source transfer is

$$F = \operatorname{diag}(e^{-V})K^\top.$$

Source: KomorebiPaperVI/FeynmanKac.lean.

```
noncomputable def transfer (K : Matrix I I ℝ) (V : I → ℝ) : Matrix I I ℝ :=
  Matrix.diagonal (fun i => Real.exp (-V i))*K.transpose
```

The path law is defined recursively for exactly n steps. At $n = 0$ there is an empty path, path probability one, zero accumulated potential, and the starting state as terminal state. Induction on the path length gives the matrix-power formula:

Source: KomorebiPaperVI/FeynmanKac.lean.

```
theorem feynman_kac_path_formula (K : Matrix I I ℝ) (V f : I → ℝ) (n : ℕ) :
    (transfer K V ^ n).mulVec f=pathExpectation K V f n := by
  induction n with
  | zero => funext i; simp [path_expectation_zero]
  | succ n ih =>
    rw [pow_succ',← Matrix.mulVec_mulVec,ih]
    funext i
    rw [path_expectation_step]
    simp only [Matrix.mulVec,dotProduct,transfer_entry,Finset.mul_sum,mul_assoc]
```

The algebraic equality itself does not require stochastic hypotheses. To call the path weights a probability law, use the separate declarations proving nonnegativity and total mass one from entrywise nonnegative K and unit column sums. Separate product, potential-sum, and terminal-index theorems ensure that the recursive law is the intended finite path expression. The continuous-time Feynman–Kac and product-slicing extensions retained in Parts II and III remain outside this discrete endpoint.

7 Schur elimination and nonnormal memory

7.1 A projected resolvent needs two invertibility facts

For retained and hidden coordinates $P \oplus Q$, the Schur complement is

$$S(z) = zI - A_{PP} - A_{PQ}(zI - A_{QQ})^{-1}A_{QP}.$$

`SchurFeshbach` uses literal complex block matrices. The projected inverse formula assumes invertibility of both the full resolvent matrix and the hidden block:

Source: `KomorebiPaperVI/SchurFeshbach.lean`.

```
theorem projected_resolvent (App : Matrix P P ℂ) (Apq : Matrix P Q ℂ)
    (Aqp : Matrix Q P ℂ) (Aqq : Matrix Q Q ℂ) (z : ℂ)
    [Invertible (Matrix.scalar Q z-Aqq)]
    [Invertible (Matrix.scalar (P ⊕ Q) z-Matrix.fromBlocks App Apq Aqp Aqq)] :
    ((Matrix.scalar (P ⊕ Q) z-Matrix.fromBlocks App Apq Aqp Aqq)⁻¹).submatrix
      Sum.inl Sum.inl = (schur App Apq Aqp Aqq z)⁻¹ := by
  let D := Matrix.scalar Q z-Aqq
  let A := Matrix.scalar P z-App
  have he := resolvent_blocks App Apq Aqp Aqq z
  let : Invertible (Matrix.fromBlocks A (-Apq) (-Aqp) D) := Invertible.copy ‹_› _ he.symm
  let : Invertible (A-(-Apq)*⅟D*(-Aqp)) := Matrix.invertibleOfFromBlocks₂₂Invertible A (-Apq)
   ↪  (-Aqp) D
  have hinv := Matrix.invOf_fromBlocks₂₂_eq A (-Apq) (-Aqp) D
  have hs : A-(-Apq)*⅟D*(-Aqp)=schur App Apq Aqp Aqq z := by
    simp only [Matrix.neg_mul,Matrix.mul_neg,neg_neg,Matrix.invOf_eq_nonsing_inv,schur,D,A]
  rw [he,← Matrix.invOf_eq_nonsing_inv,hinv]
  change ⅟(A-(-Apq)*⅟D*(-Aqp))=(schur App Apq Aqp Aqq z)⁻¹
  rw [Matrix.invOf_eq_nonsing_inv]
  congr 1
```

The module also proves the invertibility equivalence between the full block matrix and its Schur complement when the hidden block is invertible. The retained block is an inverse Schur complement, not simply the resolvent of A_{PP}. The difference is the self-energy $A_{PQ}(zI - A_{QQ})^{-1}A_{QP}$.

7.2 The fifth checked block removes a normality shortcut

The earlier memory layer establishes Laplace inversion from a supplied exponential growth estimate and also treats normal elements. The latest core 13.2 goes further for arbitrary finite complex matrices: a strict spectral half-plane alone yields integrability and decay. Nonnormality can create polynomial factors in time, so the proof must account for nilpotent generalized eigenspace contributions.

The scalar input is integrability of a complex exponential times a polynomial whenever the real part of the exponent is negative:

Source: `KomorebiPaperVI/ScalarDecay.lean`.

```
theorem exp_pow_integrable (a : ℂ) (ha : a.re < 0) (k : ℕ) :
    IntegrableOn (fun t : ℝ => Complex.exp (a*t)*(t:ℂ)^k) (Ioi 0) := by
  have hi := integrableOn_rpow_mul_exp_neg_mul_rpow
```

```
    (s := (k:ℝ)) (p := (1:ℝ)) (b := -a.re) (by linarith [Nat.cast_nonneg (α := ℝ) k]) zero_lt_one
    ↪ (by linarith)
  simp only [Real.rpow_natCast,Real.rpow_one,neg_neg] at hi
  apply hi.mono' (by fun_prop)
  filter_upwards [ae_restrict_mem measurableSet_Ioi] with t ht
  rw [norm_exp_pow a k t ht.le]
```

There is also a separate zero-limit theorem. Neither a numerical quadrature nor an assertion about eigenvalue locations substitutes for these analytic steps.

7.3 From generalized eigenspaces to a finite sum

`NonnormalMemory` studies left multiplication by A on a finite-dimensional complex C^*-algebra. A complex matrix algebra is a specialization. The source decomposes this vector space into generalized eigenspaces. On one component the exponential is a finite polynomial times $e^{\mu t}$:

Source: `KomorebiPaperVI/NonnormalMemory.lean`.

```
theorem exp_mul_component (A : R) (μ : ℂ) (x : R)
    (hx : x ∈ (Module.End.maxGenEigenspace (LinearMap.mulLeft ℂ A)) μ) (t : ℝ) :
    transfer A t*x=Complex.exp (μ*t) •
      ∑ k ∈ Finset.range (finrank ℂ R),((t:ℂ)^k/(k.factorial:ℂ)) • ((A-μ • 1)^k*x) := by
  classical
  let : NormedAlgebra ℚ R := NormedAlgebra.restrictScalars ℚ ℝ R
  let D := A-μ • 1
  have hkill (k : ℕ) (hk : finrank ℂ R ≤ k) : D^k*x=0 := by
    rw [← left_shift_power]
    exact CanonicalJets.power_kills _ μ x hx k hk
  have hs := (exp_series_hasSum_exp' (𝕂 := ℂ) ((t:ℂ) • D)).mul_right x
  have he : exp ((t:ℂ) • D)*x=
      ∑ k ∈ Finset.range (finrank ℂ R),((t:ℂ)^k/(k.factorial:ℂ)) • (D^k*x) := by
    have hs' : HasSum (fun k : ℕ => ((t:ℂ)^k/(k.factorial:ℂ)) • (D^k*x))
        (exp ((t:ℂ) • D)*x) := by
      simpa only [smul_pow,Algebra.smul_mul_assoc,smul_smul,div_eq_mul_inv,mul_comm] using hs
    exact hs'.unique (hasSum_sum_of_ne_finset_zero (fun k hk => by
      rw [hkill k (by simpa using hk),smul_zero]))
  have ht : t • A=algebraMap ℂ R (μ*t)+(t:ℂ) • D := by
    simp only [D,smul_sub,Algebra.algebraMap_eq_smul_one,smul_smul]
    rw [RCLike.real_smul_eq_coe_smul (K := ℂ),mul_comm μ (t:ℂ)]
    abel
  rw [transfer,ht,exp_add_of_commute (Algebra.commutes _ _),
    ← algebraMap_exp_comm,← Complex.exp_eq_exp_ℂ,Algebra.algebraMap_eq_smul_one,
    Algebra.smul_mul_assoc,one_mul,Algebra.smul_mul_assoc,he]
```

The cutoff is the complex dimension of the ambient algebra. For an $n \times n$ matrix algebra that can be n^2. It need not be a minimal Jordan-block length. The proof only needs a valid nilpotence cutoff. Terms beyond that cutoff vanish on the component.

After damping by e^{-zt}, each coefficient has exponent $\mu - z$ with negative real part. The source combines the scalar integrability and limit results over finite sums and then over the generalized-eigenspace decomposition. Applying the resulting identity to the algebra unit gives the operator-valued decay:

Source: KomorebiPaperVI/NonnormalMemory.lean.

```
theorem spectral_decay (A : R) (z : ℂ) (hz : ∀ μ∈spectrum ℂ A,μ.re < z.re) :
    IntegrableOn (damped A z) (Ioi 0) ∧ Tendsto (damped A z) atTop (𝓝 0) := by
  have h (x : R) : IntegrableOn (fun t => damped A z t*x) (Ioi 0) ∧
      Tendsto (fun t => damped A z t*x) atTop (𝓝 0) := by
    apply Submodule.iSup_induction (Module.End.maxGenEigenspace (LinearMap.mulLeft ℂ A))
      (x := x) (motive := fun x => IntegrableOn (fun t : ℝ => damped A z t*x) (Ioi 0) volume ∧
        Tendsto (fun t => damped A z t*x) atTop (𝓝 0))
      (by rw [Module.End.iSup_maxGenEigenspace_eq_top]; trivial)
    · exact component_decay A z hz
    · simp
    · intro x y hx hy
      simpa only [Pi.add_def,mul_add,add_zero] using And.intro (hx.1.add hy.1) (hx.2.add hy.2)
  simpa using h 1
```

The condition is strict: $\operatorname{Re}\mu < \operatorname{Re} z$ for every spectral value. The conclusion includes both Bochner integrability and the limit zero. It does not rely on a claimed universal bound $\|e^{tA}\| \leq e^{t \max \operatorname{Re}\sigma(A)}$, which would fail for general nonnormal matrices.

7.4 Why both derivative identities appear

The improper-integral fundamental theorem is applied once to left multiplication and once to right multiplication. These establish both inverse equations. This is what connects the integral to the actual inverse, rather than just presenting a formal solution to one equation.

Source: KomorebiPaperVI/NonnormalMemory.lean.

```
theorem spectral_laplace_inverse (A : R) (z : ℂ)
    (hz : ∀ μ∈spectrum ℂ A,μ.re < z.re) :
    IntegrableOn (damped A z) (Ioi 0) ∧ IsUnit (algebraMap ℂ R z-A) ∧
      (∫ t in Ioi 0,damped A z t)=Ring.inverse (algebraMap ℂ R z-A) := by
  have hs := spectral_decay A z hz
  let B := A-algebraMap ℂ R z
  have he : damped A z=transfer B := funext (damped_eq_shifted A z)
  rw [he] at hs ⊢
  have hl := integral_Ioi_of_hasDerivAt_of_tendsto'
    (fun t (_ : t∈Ici (0:ℝ)) => transfer_derivative_left B t)
    ((ContinuousLinearMap.mul ℝ R B).integrable_comp hs.1) hs.2
  have hr := integral_Ioi_of_hasDerivAt_of_tendsto'
    (fun t (_ : t∈Ici (0:ℝ)) => transfer_derivative_right B t)
    (((ContinuousLinearMap.mul ℝ R).flip B).integrable_comp hs.1) hs.2
  rw [integral_const_mul_of_integrable hs.1,transfer_zero,zero_sub] at hl
  rw [integral_mul_const_of_integrable hs.1,transfer_zero,zero_sub] at hr
  have hn : algebraMap ℂ R z-A= -B := by dsimp [B]; abel
  have hl' : (algebraMap ℂ R z-A)*(∫ t in Ioi 0,transfer B t)=1 := by
    rw [hn,neg_mul,hl,neg_neg]
  have hr' : (∫ t in Ioi 0,transfer B t)*(algebraMap ℂ R z-A)=1 := by
    rw [hn,mul_neg,hr,neg_neg]
  have hu : IsUnit (algebraMap ℂ R z-A) := ⟨⟨_,_,hl',hr'⟩,rfl⟩
  refine ⟨hs.1,hu,?_⟩
```

```
  calc
    _ = Ring.inverse (algebraMap ℂ R z-A)*((algebraMap ℂ R z-A)*(∫ t in Ioi 0,transfer B t)) :=
      (Ring.inverse_mul_cancel_left _ _ hu).symm
    _ = _ := by rw [hl',mul_one]
```

The result supplies invertibility as part of the conclusion. The total operation `Ring.inverse` is therefore being used at an established unit. This is the formal endpoint connecting the strict spectral half-plane with a genuine resolvent formula.

7.5 Retained and hidden dimensions can differ

The final matrix theorem uses A on hidden coordinates J, $U : J \to I$, and $V : I \to J$. It wraps $B \mapsto UBV$ as a continuous linear map and commutes that map with the Bochner integral. The result is the literal rectangular memory/self-energy identity:

Source: `KomorebiPaperVI/MatrixMemory.lean`.

```
theorem rectangular_memory_self_energy (A : Matrix J J ℂ)
    (U : Matrix I J ℂ) (V : Matrix J I ℂ) (z : ℂ)
    (hz : ∀ μ∈spectrum ℂ A,μ.re < z.re) :
    IntegrableOn (fun t : ℝ => Complex.exp (-z*t) • (U*transfer A t*V)) (Ioi 0) ∧
    (∫ t : ℝ in Ioi 0,Complex.exp (-z*t) • (U*transfer A t*V))=
      U*(Matrix.scalar J z-A)⁻¹*V := by
  have h := NonnormalMemory.spectral_laplace_inverse A z hz
  have he : (fun t : ℝ => Complex.exp (-z*t) • (U*transfer A t*V))=
      (fun t => sandwich U V (damped A z t)) := by
    ext t i j
    simp [sandwich,damped]
  rw [he]
  refine ⟨(sandwich U V).integrable_comp h.1,?_⟩
  rw [(sandwich U V).integral_comp_comm h.1,h.2.2]
  change U*Ring.inverse (algebraMap ℂ (Matrix J J ℂ) z-A)*V=_
  rw [Matrix.nonsing_inv_eq_ringInverse]
  rfl
```

The source imposes no equality of retained and hidden dimensions and no nonemptiness assumption on the finite index types. The theorem includes nilpotent Jordan parts and zero-dimensional cases. It proves a finite-dimensional Laplace identity. General infinite hidden systems, globally optimal model reduction, and arbitrary memory closure remain separate questions.

8 Canonical jets and holomorphic naturality

Core 15.1 closes a common source of ambiguity: what exactly is meant by $f(A)$ when A is a nonnormal finite matrix and f is holomorphic on a possibly disconnected neighborhood of the spectrum? The final layer defines a canonical finite-jet operator, proves uniqueness and intertwining, and identifies its coefficients with genuine local Cauchy integrals under analytic hypotheses.

8.1 The finite-jet definition

For coefficient data $a(\mu, k)$, the jet at μ is a polynomial in $A - \mu I$:

Source: KomorebiPaperVI/CanonicalJets.lean.

```
noncomputable def jet (A : End ℂ V) (a : ℂ → ℕ → ℂ) (μ : ℂ) (n : ℕ) : End ℂ V :=
  ∑ k ∈ Finset.range n, a μ k • (A-μ • 1)^k
```

The canonical operator applies this polynomial on each generalized eigenspace and reassembles the direct sum:

Source: KomorebiPaperVI/CanonicalJets.lean.

```
noncomputable def calculus (A : End ℂ V) (a : ℂ → ℕ → ℂ) : End ℂ V := by
  classical
  exact (DirectSum.toModule ℂ ℂ V (fun μ =>
    (jet A a μ (finrank ℂ V)).comp (A.maxGenEigenspace μ).subtype)).comp
    (DirectSum.decomposeLinearEquiv A.maxGenEigenspace).toLinearMap
```

The decomposition is selected noncomputably, but the resulting operator is characterized uniquely by its action on each component. `HolomorphicFinite.canonical_unique` proves that characterization. "Noncomputable" means the definition uses classical constructions without an executable algorithm. It does not mean the proof is admitted.

8.2 Unequal dimensions require a common cutoff

If $CA = BC$, then C takes a generalized eigenspace of A into the corresponding one of B. Source and target may have different dimensions, so the naturality proof first extends both finite jets to a common maximum cutoff. Truncation stability makes that extension harmless:

Source: KomorebiPaperVI/CanonicalJets.lean.

```
theorem jet_stable (A : End ℂ V) (a : ℂ → ℕ → ℂ) (μ : ℂ) (x : V)
    (hx : x ∈ A.maxGenEigenspace μ) (n : ℕ) (hn : finrank ℂ V ≤ n) :
    jet A a μ n x=jet A a μ (finrank ℂ V) x := by
  classical
  unfold jet
  simp only [LinearMap.sum_apply,LinearMap.smul_apply]
  symm
  apply Finset.sum_subset (Finset.range_mono hn)
  intro k _ hk
  rw [power_kills A μ x hx k (by simpa using hk),smul_zero]
```

The source then applies polynomial intertwining componentwise and uses the direct-sum extensionality principle:

Source: KomorebiPaperVI/CanonicalJets.lean.

```
theorem canonical_naturality (A : End ℂ V) (B : End ℂ W) (C : V →ₗ[ℂ] W)
    (h : B.comp C=C.comp A) (a : ℂ → ℕ → ℂ) :
    C.comp (calculus A a)=(calculus B a).comp C := by
  apply DirectSum.decompose_lhom_ext A.maxGenEigenspace
  intro μ
  ext x
  have hx := Module.End.mapsTo_genEigenspace_of_comp C h μ ⊤ x.property
  change C (calculus A a x)=calculus B a (C x)
  rw [calculus_on_component A a μ x x.property,
    calculus_on_component B a μ (C x) hx,
    ← jet_stable A a μ x x.property (max (finrank ℂ V) (finrank ℂ W)) (le_max_left _ _),
    ← jet_stable B a μ (C x) hx (max (finrank ℂ V) (finrank ℂ W)) (le_max_right _ _)]
  exact jet_intertwines A B C h a μ _ x
```

This is an algebraic theorem about arbitrary coefficient data a. It is already enough to explain why the same local jets must be used on both sides of an intertwining equation.

8.3 Analyticity enters the interpretation

`CanonicalJets.holomorphic` substitutes the derivative coefficients $f^{(k)}(\mu)/k!$. The term exists for any function because Lean's derivative operations are totalized. The raw jet-naturality theorem therefore has no analyticity assumption. That fact must not be mistaken for a theorem that an arbitrary function has an analytic holomorphic functional calculus.

For an analytic f, Cauchy's formula connects the derivative coefficient to an actual positive-radius circle integral:

Source: KomorebiPaperVI/HolomorphicFinite.lean.

```
theorem cauchyCoefficient_eq (f : ℂ → ℂ) (μ : ℂ) (r : ℝ) (hr : 0 < r)
    (hf : DifferentiableOn ℂ f (Metric.closedBall μ r)) (k : ℕ) :
    cauchyCoefficient f μ r k=iteratedDeriv k f μ/(k.factorial:ℂ) := by
  have hh := hf.circleIntegral_one_div_sub_center_pow_smul hr k
  simp only [smul_eq_mul,one_div] at hh
  unfold cauchyCoefficient
  have he : (fun z => f z/(z-μ)^(k+1))=(fun z => ((z-μ)^(k+1))⁻¹*f z) := by
    ext z
    simp [div_eq_mul_inv,mul_comm]
  rw [he,hh]
  field_simp
```

The radius is positive and f is differentiable on the whole closed disk in this statement. A later theorem chooses suitable local disks inside any declared open neighborhood where f is analytic:

Source: KomorebiPaperVI/HolomorphicFinite.lean.

```
theorem canonical_contours_exist (A : End ℂ V) (f : ℂ → ℂ) (U : Set ℂ)
    (hU : IsOpen U) (hf : AnalyticOnNhd ℂ f U) (hA : spectrum ℂ A ⊆ U) :
    ∃ r : ℂ → ℝ,(∀ μ∈spectrum ℂ A,0 < r μ ∧ Metric.closedBall μ (r μ) ⊆ U) ∧
      calculus A (fun μ k => cauchyCoefficient f μ (r μ) k)=holomorphic A f := by
```

```
  classical
  have he (μ : ℂ) : ∃ r : ℝ,0 < r ∧
      (μ∈spectrum ℂ A → Metric.closedBall μ r ⊆ U) := by
    by_cases hm : μ∈spectrum ℂ A
    · obtain ⟨r,hr,hsub,_⟩ := admissible_circle f U hU hf μ (hA hm)
      exact ⟨r,hr,fun _ => hsub⟩
    · exact ⟨1,zero_lt_one,fun h => (hm h).elim⟩
  choose r hr hsub using he
  refine ⟨r,fun μ hμ => ⟨hr μ,hsub μ hμ⟩,?_⟩
  apply canonical_cauchy_representation A f r (fun μ _ => hr μ)
  intro μ hμ z hz
  exact (hf z (hsub μ hμ hz)).differentiableAt.differentiableWithinAt
```

The local radius can depend on the spectral value. This supports disconnected analytic neighborhoods and avoids assuming a single globally convergent Taylor series. Equality of the Cauchy coefficients proves independence of the admissible local circles. General contour deformation in arbitrary Banach algebras is not the formal scope of this finite-dimensional construction.

8.4 The final literal matrix equation

`matrixFunction` transfers the finite-jet operator from linear maps back to matrices. The resulting equation is rectangular and uses the original matrix intertwining hypothesis:

Source: `KomorebiPaperVI/HolomorphicFinite.lean`.

```
theorem matrix_holomorphic_naturality (A : Matrix I I ℂ) (B : Matrix J J ℂ)
    (C : Matrix J I ℂ) (h : C*A=B*C) (f : ℂ → ℂ) :
    C*matrixFunction A f=matrixFunction B f*C := by
  have hh : B.toLin'.comp C.toLin'=C.toLin'.comp A.toLin' := by
    rw [← Matrix.toLin'_mul,← Matrix.toLin'_mul,h]
  have hc := holomorphic_naturality A.toLin' B.toLin' C.toLin' hh f
  have hm := congrArg LinearMap.toMatrix' hc
  simpa only [LinearMap.toMatrix'_comp,LinearMap.toMatrix'_toLin',matrixFunction] using hm
```

Read this as naturality of the defined jet operator for arbitrary f, and as ordinary finite-dimensional holomorphic naturality when the preceding analytic hypotheses identify those jets with the desired holomorphic calculus. The bounded Banach reciprocal-family substitution moved from Chapter IV has a broader setting and retains its own written proof in Part III. It has not acquired a new Lean endpoint by being moved into Chapter VI.

9 Survival, hazard, and incomplete absorption

`Survival` begins with a finite real matrix whose off-diagonal entries are nonnegative and whose column sums are nonpositive. Its exponential evolves the surviving transient-state mass. With a nonnegative initial probability vector, survival is positive at every finite nonnegative time, the density is nonnegative, and survival is decreasing.

9.1 Finite-time positivity makes the hazard meaningful

The source proves strict positivity of $S(t)$, rather than treating $h(t) = f(t)/S(t)$ as legitimate by notation alone. Integrating the logarithmic derivative then yields the hazard identity:

Source: `KomorebiPaperVI/Survival.lean`.

```
theorem survival_hazard (A : Matrix I I ℝ) (hA : ∀ i j, i≠j → 0 ≤ A i j)
    (p : I → ℝ) (hp : ∀ i,0 ≤ p i) (hs : ∑ i,p i=1) (t : ℝ) (ht : 0 ≤ t) :
    survival A p t=Real.exp (-∫ r in 0..t,hazard A p r) := by
  have hd : ∀ r∈Icc 0 t, HasDerivAt (fun u => Real.log (survival A p u)) (-hazard A p r) r := by
    intro r hr
    simpa [hazard,neg_div] using (survival_derivative A p r).log
      (survival_positive A hA p hp hs r hr.1).ne'
  have hscont : Continuous (survival A p) :=
    (show Differentiable ℝ (survival A p) from fun r => (survival_derivative A p
    ↪  r).differentiableAt).continuous
  have hhcont : ContinuousOn (hazard A p) (Icc 0 t) :=
    (density_continuous A p).continuousOn.div hscont.continuousOn
      (fun r hr => (survival_positive A hA p hp hs r hr.1).ne')
  have hi := intervalIntegral.integral_eq_sub_of_hasDerivAt
    (by simpa [uIcc_of_le ht] using hd)
    (hhcont.neg.intervalIntegrable_of_Icc ht)
  simp only [Pi.neg_apply,intervalIntegral.integral_neg,survival_zero,hs,Real.log_one,sub_zero] at
  ↪  hi
  rw [hi,Real.exp_log (survival_positive A hA p hp hs t ht)]
```

The displayed theorem's hypotheses establish the relevant finite-time positivity and normalization. The nonpositive-column-sum condition is used in the density and monotonicity results. Keep these neighboring statements together when interpreting a model as a killed Markov evolution.

9.2 A density need not integrate to one

A finite subgenerator can leave positive surviving mass as $t \to \infty$. The formal result therefore proves existence of a limit and integrates the density to $1 - \ell$, not automatically to one:

Source: `KomorebiPaperVI/Survival.lean`.

```
theorem defective_density_mass (A : Matrix I I ℝ) (hA : ∀ i j, i≠j → 0 ≤ A i j)
    (hc : ∀ j,∑ i,A i j ≤ 0) (p : I → ℝ) (hp : ∀ i,0 ≤ p i) (hs : ∑ i,p i=1) :
    ∃ l,0 ≤ l ∧ Tendsto (survival A p) atTop (𝓝 l) ∧
      IntegrableOn (density A p) (Ioi 0) ∧ (∫ t in Ioi 0,density A p t)=1-l := by
  obtain ⟨l,hl,ht⟩ := survival_has_limit A hA hc p hp
  have hd := fun t (_ : t∈Ici (0:ℝ)) => survival_derivative A p t
```

```
have hn := fun t (ht : t∈Ioi (0:ℝ)) =>
  neg_nonpos.mpr (density_nonnegative A hA hc p hp t ht.le)
have hi := integrableOn_Ioi_deriv_of_nonpos' hd hn ht
have he := integral_Ioi_of_hasDerivAt_of_nonpos' hd hn ht
refine ⟨l,hl,ht,integrable_neg_iff.mp hi,?_⟩
simp only [integral_neg,survival_zero,hs] at he
linarith
```

The separate `eventual_absorption_density` endpoint assumes $S(t) \to 0$ and then obtains a unit-mass density. This distinction is essential in a phase-type application: eventual absorption is a hypothesis to establish, not an effect of calling a matrix transient in prose. Transform, moment, residual-life, and competing-risk formulas in the retained extensions have additional assumptions and are mapped separately.

10 The retained extensions and the formal boundary

The mathematical text contains more mathematics than the 33-result kernel ledger. A nearby formal identity can explain an extension's architecture without proving that extension. This section maps the regions by mathematical content. "Related core" is a navigation aid, not a certificate for the entire row.

10.1 Part II: the strengthened branch

The labels in this table use Part II's E prefix. The 84 theorem-like result records in that branch retain their written proofs. The precise hypotheses are stated in the mathematical text.
The table does not enlarge the formal denominator.

Written region	Related core	Formal boundary
E2: packet semigroups	1.2, 2.1, 2.2	Formal actual exponential and finite packet convolution. General C_0 semigroups, Dyson expansions, and packet-path resolvent series require their retained written assumptions.
E3: generator defects	3.1–3.3, 15.1	Rectangular Duhamel and matrix resolvent endpoints are formal. General Banach holomorphic defect formulas exceed finite canonical-jet naturality.
E4: limiting refinement	10.5	Formal varying-space norm summability does not construct every inverse-limit semigroup. Weighted completion, range leakage, generator closure, and limiting state-space choices are separate written results.
E5: Markov lumping	4.1, 5.1, 5.2	Exponential probability preservation and exact/approximate iterates are formal. Quotient obstructions, complete descent characterizations, and stationary descent have separate written proofs.
E6: conditional towers	6.1, 6.2	Finite weighted kernels and difference packets are formal. General conditional expectation and every expanded variance identity are not additional ledger endpoints.
E7: Gaussian/Hermite	None directly	The Ornstein–Uhlenbeck semigroup, infinite Hermite reconstruction, generator, Dirichlet form, spectral gap, and fractional Hermite transfer are outside this formal core.
E8: Bayesian transfer	7.1–7.3	Composition, evidence packets, and likelihood perturbation are formal. Prior-stability statements require their own exact interface and are not counted by analogy.
E9: finite sufficiency	8.1, 8.2, 8.4–8.6	Effective likelihoods, all-prior criteria, coarsest nontrivial partitions, and the loss matrix are formal within the ledger scope. Relative-entropy decomposition and data processing are separate written results.
E10: Bayesian flows/HMM	9.1, 9.3	The normalized exponential flow and one-step simplex preservation are formal. Cumulants, concentration, and quantitative filtering bounds are retained written extensions.
E11: fractional operators	10.1–10.5	Actual normalized integrals, rectangular bounded Hilbert-space defects, and square-root subordination are formal. General stable-subordinator exposition is not a new formal endpoint.

Written region	Related core	Formal boundary
E12: Green/PDE	11.1, 11.2	Exact finite Dirichlet inverse, grid-node identity, and scalar low-mode error are formal. Quadrature, forcing-error, and broader mode-transfer estimates retain their stated regularity assumptions.
E13: Feynman–Kac	12.1	The exact discrete finite path law is formal. Continuous-time probabilistic representation and product slicing have separate written proofs.
E14: Schur elimination	13.1, 13.2	Actual block inverse and finite-dimensional Laplace memory identity are formal. The full time-domain closure criterion is not separately included in the ledger.
E15: memory compression	13.2	The finite nonnormal exponential-polynomial mechanism is formal. Certified block selection, realizations, compression optimization, and Volterra trajectory bounds are written extensions.
E16: survival/risks	14.1	Finite survival, hazard, and defective density mass are formal. Transform moments, residual-life descriptions, and competing-risk decompositions have separate written scope.

10.2 Part III: earlier results retained in their own right

The earlier Annals branch contains results whose scope or constants are not fully subsumed by the strengthened text. Their retained statements and proofs appear in Part III, with R-prefixed numbering. The accompanying navigation crosswalk lists every retained theorem-like record and its final traditional location. Retention preserves the mathematics. It does not create new Lean declarations.

These results include multilevel telescoping and coordinate-generator arguments, reversible descent and Dirichlet forms, prior-relative Bayesian sufficiency, quantitative normalization and information-loss estimates, covariance and hidden-Markov formulas, potential perturbation and coarse Feynman–Kac transfer, and finite rational memory or phase-type descriptions. Each has to be compared against a formal theorem by its exact hypotheses and conclusion. Shared vocabulary is insufficient.

10.3 The operator substitution migrated from Chapter IV

The migrated reciprocal-family proposition concerns bounded Banach operators and holomorphic functional calculus, together with an explicit classical functional-calculus input. It belongs with Chapter VI's operator language and is retained with its written proof. The core 15.1 result instead constructs and proves naturality of canonical finite-dimensional jets. Their relationship is conceptual. The migrated proposition is outside the 33-result formal contract.

10.4 The three original continuum conjectures

The formal core retains three open conjectures: transfer-semigroup realization on completed state spaces; uniform refinement control for meaningful discretization families; and latent-packet closure for stable infinite packet systems. Finite or bounded special cases proved elsewhere in this book do not settle these universal formulations. In particular, summable local defects alone do not guarantee a semigroup limit without the appropriate completion, compatibility, and closure conditions.

10.5 Traditional chapter navigation

The Printed and PDF columns locate results in the attached Chapter VI source document. Printed labels and physical PDF pages differ because of the opening material. These locations refer to that source document rather than the continuous folios of this book.

Core	Result	Printed	PDF
1.2	Finite semigroup law	3	10
2.1	Packet semigroup convolution	4	11
2.2	Reducing packets	4	11
3.1	Exact Duhamel refinement identity	5	12
3.2	Norm bound	6	13
3.3	Resolvent refinement identity	6	13
4.1	Finite Markov semigroup	7	14
5.1	Exact lumpability	8	15
5.2	Finite-horizon defect formula	8	15
6.1	Finite conditional-expectation projector	9	16
6.2	Tower law and martingale packets	10	17
7.1	Independent-likelihood composition	11	18
7.2	Bayesian packet reconstruction	11	18
7.3	Likelihood perturbation bound	12	19
8.1	Exact effective-likelihood factorization	13	20
8.2	Bayesian refinement criterion	14	21
8.4	Bayesian information-loss packet	14	21
8.5	Nested Bayesian packet tower	15	22
8.6	Canonical coarsest likelihood-resolving partition	15	22
9.1	Bayesian nonlinear semigroup	18	25
9.3	Prediction-update reconstruction	19	26
10.1	Fractional packet calculus	20	27
10.2	Poisson subordination	21	28
10.3	Fractional intertwining theorem	22	29
10.4	Fractional semigroup refinement	23	30
10.5	Fractional refinement towers	23	30
11.1	Exact discrete Green reconstruction at grid nodes	25	32
11.2	Second-order low-mode convergence	25	32
12.1	Discrete Feynman–Kac path formula	27	34
13.1	Schur–Feshbach projected resolvent	28	35
13.2	Memory/self-energy correspondence	29	36
14.1	Semigroup survival identities	30	37
15.1	Naturality under exact intertwining	32	39

The 24 earlier results retained in Part III. These have written proofs. This list creates no additional Lean completion claim.

Final	Result	Printed	PDF
R1.1	Packet convolution	90	97
R1.2	Packet resolvent convolution	90	97
R4.1	Fractional low mode dispersion	96	103

Final	Result	Printed	PDF
R5.1	Stability of memory compression	97	104
R1.3	Minimum transfer order	91	98
R1.4	Several level telescoping identity	91	98
R1.5	Summable generator defects	91	98
R1.7	Exact projective limit semigroup	92	99
R1.8	Coordinate generators	92	99
R2.1	Reversibility descends	92	99
R2.2	Coarse Dirichlet form	93	100
R3.1	Sufficiency relative to a prior disintegration	93	100
R3.2	Normalization bound	93	100
R3.3	Exact information loss formula	94	101
R3.4	Oscillation bound	94	101
R3.5	Covariance identity	95	102
R3.6	Exact coarse hidden Markov filter	95	102
R3.7	One step filter defect	95	102
R4.2	Potential perturbation bound	96	103
R4.3	Coarse Feynman Kac transfer	96	103
R5.2	Rational memory realization	98	105
R5.3	Exponential sum memory	98	105
R5.4	Phase type rationality	98	105
R5.5	One phase model	98	105

The migrated bounded Banach reciprocal proposition is R7.2, printed page 101 (PDF page 108). Its scope remains outside the 33-result contract.

11.5 Building this reading guide

From the directory for the editable Lean guide, run `python3 build.py`. It compiles the supplied TeX and excerpt files with XeLaTeX and does not invoke Lean or regenerate mathematical evidence. The build uses installed URW P052 and Nimbus Sans, Latin Modern mono and math, and DejaVu Sans Mono fonts. Font files are resolved locally with `kpsewhich` and standard system locations, then copied only into the private build stage. No font binaries are bundled or downloaded. The monograph style is supplied beside the TeX source. The script copies the full authored guide tree to a temporary directory and compiles it in four passes. No installation is performed. `verify_pdf.py` performs the documented PDF checks and renders all pages into contact sheets.

`assemble.py` is a separate step for generating documentation. It reads the preserved source snapshot, exact source contract, selected excerpt list, and a current verification report. Its default inputs are relative to the supplied package. It also accepts `--source` and `--verification` paths when those inputs are stored elsewhere. Its generated tables are already supplied for ordinary PDF reproduction. Run `python3 reconcile_sources.py` to check those tables and excerpts against the preserved sources without changing their mathematical content.

11.6 When a reuse needs a new proof

Before adapting an endpoint, check the scalar field, finite versus bounded setting, positivity notion, time domain, evidence condition, and domain/codomain of every operator. Changing a hypothesis is not a formatting edit. A new theorem about unbounded generators, general statistical sufficiency, infinite hidden-state elimination, or unrestricted Banach holomorphic calculus requires its own proof and correspondence review. The preserved project supplies a precise starting point for that work.

12 Complete ledger of the 33 scoped results

This ledger reproduces the recorded checked scope from `THEOREM_LEDGER.json`. Numbering refers to the latest core in Part I of the traditional chapter. These are 33 manuscript result records, not 33 source declarations and not a certification of every retained extension. The declaration catalog in Section 13 resolves every named target to its module and original line.

12.1 1.2 / Finite semigroup law

Recorded status. Complete scoped statement in the formal-core correspondence.

Real Banach-algebra exponential law and both derivatives; finite operators are a specialization.

Declared targets.

- `KomorebiPaperVI.Semigroups.transfer_add`. Source line 20.
- `KomorebiPaperVI.Semigroups.transfer_zero`. Source line 18.
- `KomorebiPaperVI.Semigroups.transfer_derivative_left`. Source line 29.
- `KomorebiPaperVI.Semigroups.transfer_derivative_right`. Source line 26.

12.2 2.1 / Packet semigroup convolution

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual exponential packet convolution for a finite idempotent resolution of identity.

Declared targets.

- `KomorebiPaperVI.Algebra.block_reconstruction`. Source line 12.
- `KomorebiPaperVI.Semigroups.exponential_packet_convolution`. Source line 44.

12.3 2.2 / Reducing packets

Recorded status. Complete scoped statement in the formal-core correspondence.

A finite orthogonal idempotent resolution with zero off-diagonal generator blocks commutes with the generator; actual exponential cross blocks vanish and compressed dynamics agree with the exponential of the compressed generator.

Declared targets.

- `KomorebiPaperVI.ReducingPackets.generator_commutes`. Source line 12.
- `KomorebiPaperVI.ReducingPackets.reducing_exponential`. Source line 32.
- `KomorebiPaperVI.ReducingPackets.compressed_evolution`. Source line 39.

12.4 3.1 / Exact Duhamel refinement identity

Recorded status. Complete scoped statement in the formal-core correspondence.

Bochner interval-integral identity for rectangular bounded real operators on Banach spaces, and exact intertwining.

Declared targets.

- `KomorebiPaperVI.Semigroups.duhamel_integrand_derivative`. Source line 57.

- `KomorebiPaperVI.Semigroups.duhamel_rectangular`. Source line 70.
- `KomorebiPaperVI.Semigroups.exact_rectangular_intertwining`. Source line 81.

12.5 3.2 / Norm bound

Recorded status. Complete scoped statement in the formal-core correspondence.

The displayed exponential-growth integral estimate for rectangular bounded real operators, with the contraction specialization.

Declared targets.

- `KomorebiPaperVI.OperatorBounds.growth_duhamel_bound`. Source line 10.
- `KomorebiPaperVI.Semigroups.contraction_duhamel_bound`. Source line 89.

12.6 3.3 / Resolvent refinement identity

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal rectangular matrices with left/right inverse equations and central scalar; applies to the stated resolvents.

Declared targets.

- `KomorebiPaperVI.Algebra.rectangular_resolvent_identity`. Source line 85.

12.7 4.1 / Finite Markov semigroup

Recorded status. Complete scoped statement in the formal-core correspondence.

The actual real matrix exponential of any finite generator has nonnegative entries and columns summing to one for nonnegative time; consequently it preserves probability vectors.

Declared targets.

- `KomorebiPaperVI.MarkovSemigroup.markov_nonnegative`. Source line 48.
- `KomorebiPaperVI.MarkovSemigroup.markov_column_sum`. Source line 71.
- `KomorebiPaperVI.MarkovSemigroup.markov_probability`. Source line 84.

12.8 5.1 / Exact lumpability

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal rectangular matrix intertwining for every natural-number iterate.

Declared targets.

- `KomorebiPaperVI.Algebra.intertwining_powers`. Source line 53.

12.9 5.2 / Finite-horizon defect formula

Recorded status. Complete scoped statement in the formal-core correspondence.

The literal rectangular finite-sum defect identity and the norm bound for arbitrary rectangular contractions, including horizon zero and zero spaces.

Declared targets.

- `KomorebiPaperVI.Algebra.finite_horizon_defect`. Source line 64.
- `KomorebiPaperVI.OperatorBounds.finite_horizon_contraction`. Source line 50.

12.10 6.1 / Finite conditional-expectation projector

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal finite complex weighted conditional expectation: idempotence, self-adjointness for the weighted inner product, and range exactly the block-constant functions. Strictly positive finite weights are explicit.

Declared targets.

- `KomorebiPaperVI.ConditionalExpectation.expect_idempotent`. Source line 49.
- `KomorebiPaperVI.ConditionalExpectation.expect_self_adjoint`. Source line 105.
- `KomorebiPaperVI.ConditionalExpectation.expect_range`. Source line 74.
- `KomorebiPaperVI.ConditionalExpectation.expect_matrix`. Source line 84.

12.11 6.2 / Tower law and martingale packets

Recorded status. Complete scoped statement in the formal-core correspondence.

Both concrete conditional-expectation tower identities; arbitrary pairs of nested projector kernels multiply to the coarser kernel; their differences are pairwise orthogonal idempotents and sum to the terminal projector.

Declared targets.

- `KomorebiPaperVI.ConditionalTowers.conditional_kernel_product`. Source line 37.
- `KomorebiPaperVI.ConditionalTowers.nested_kernel_min`. Source line 53.
- `KomorebiPaperVI.ConditionalTowers.martingale_packet_product`. Source line 65.
- `KomorebiPaperVI.ConditionalTowers.martingale_packet_reconstruction`. Source line 72.

12.12 7.1 / Independent-likelihood composition

Recorded status. Complete scoped statement in the formal-core correspondence.

Composition with explicit nonzero first and combined evidences; positive evidences imply these hypotheses.

Declared targets.

- `KomorebiPaperVI.FiniteBayes.sequential_evidence`. Source line 37.
- `KomorebiPaperVI.FiniteBayes.posterior_composition`. Source line 45.

12.13 7.2 / Bayesian packet reconstruction

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal finite fibers and evidence-weighted reconstruction, including zero-evidence packets; specialize nonnegative weights to likelihood times prior.

Declared targets.

- `KomorebiPaperVI.Coarsening.packet_reconstruction_point`. Source line 90.
- `KomorebiPaperVI.Coarsening.packet_reconstruction`. Source line 99.
- `KomorebiPaperVI.Coarsening.zero_packet_component`. Source line 48.

12.14 7.3 / Likelihood perturbation bound

Recorded status. Complete scoped statement in the formal-core correspondence.

The printed finite weighted L1 perturbation bound under nonnegativity and positive evidence.

Declared targets.

- `KomorebiPaperVI.FiniteBayes.normalized_l1_bound_left`. Source line 73.
- `KomorebiPaperVI.FiniteBayes.likelihood_perturbation_bound`. Source line 102.

12.15 8.1 / Exact effective-likelihood factorization

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal finite coarsening and prior-dependent effective likelihood, including zero prior-mass fibers.

Declared targets.

- `KomorebiPaperVI.Coarsening.effective_weight`. Source line 66.
- `KomorebiPaperVI.Coarsening.effective_evidence`. Source line 78.
- `KomorebiPaperVI.Coarsening.effective_factorization`. Source line 83.

12.16 8.2 / Bayesian refinement criterion

Recorded status. Complete scoped statement in the formal-core correspondence.

The all-prior equivalence for strictly positive fine and coarse likelihoods and a surjective partition with at least two packets; necessity is derived from actual two-state priors. Singleton coarsening is automatic.

Declared targets.

- `KomorebiPaperVI.BayesianNecessity.refinement_iff`. Source line 112.
- `KomorebiPaperVI.BayesianNecessity.cross_packet_ratio`. Source line 67.
- `KomorebiPaperVI.BayesianTowers.singleton_commutes`. Source line 30.

12.17 8.4 / Bayesian information-loss packet

Recorded status. Complete scoped statement in the formal-core correspondence.

The literal information-loss matrix factorization, null scale direction, minimum-evidence L1 estimate, and the final fine-to-coarse weighted bound, including zero prior-mass fibers.

Declared targets.

- `KomorebiPaperVI.InformationLoss.information_loss_matrix`. Source line 17.
- `KomorebiPaperVI.InformationLoss.information_loss_null`. Source line 28.
- `KomorebiPaperVI.InformationLoss.information_loss_norm`. Source line 68.
- `KomorebiPaperVI.InformationLoss.total_loss_bound`. Source line 60.
- `KomorebiPaperVI.FiniteBayes.likelihood_perturbation_bound`. Source line 102.

12.18 8.5 / Nested Bayesian packet tower

Recorded status. Complete scoped statement in the formal-core correspondence.

The necessary-and-sufficient adjacent likelihood criterion with singleton exception, and exact Bayesian commutation over every finite span of a varying-state-space tower.

Declared targets.

- `KomorebiPaperVI.BayesianTowers.adjacent_criterion`. Source line 43.
- `KomorebiPaperVI.BayesianTowers.tower_criterion`. Source line 80.
- `KomorebiPaperVI.BayesianTowers.tower_commutes`. Source line 73.
- `KomorebiPaperVI.BayesianTowers.coarsen_comp`. Source line 12.

12.19 8.6 / Canonical coarsest likelihood-resolving partition

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual quotient by equality of likelihood profiles, canonical coarse likelihoods and exact closure, unique factorization from every likelihood-resolving partition, and the corrected coarsest nontrivial all-prior-closed property under strict positivity and distinct profiles.

Declared targets.

- `KomorebiPaperVI.BayesianPartitions.packet_eq_iff`. Source line 28.
- `KomorebiPaperVI.BayesianPartitions.canonical_coarsening`. Source line 38.
- `KomorebiPaperVI.BayesianPartitions.canonical_unique_factor`. Source line 46.
- `KomorebiPaperVI.BayesianPartitions.all_prior_family_iff_resolves`. Source line 67.
- `KomorebiPaperVI.BayesianPartitions.canonical_coarsest_nontrivial`. Source line 85.
- `KomorebiPaperVI.BayesianPartitions.canonical_nontrivial`. Source line 91.

12.20 9.1 / Bayesian nonlinear semigroup

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual normalized real exponential flow on the finite probability simplex, its semigroup law, and centered multiplicative derivative.

Declared targets.

- `KomorebiPaperVI.BayesianFlow.positive_evidence`. Source line 13.
- `KomorebiPaperVI.BayesianFlow.flow_probability`. Source line 25.
- `KomorebiPaperVI.BayesianFlow.flow_add`. Source line 36.
- `KomorebiPaperVI.BayesianFlow.flow_zero`. Source line 45.
- `KomorebiPaperVI.BayesianFlow.flow_derivative`. Source line 64.

12.21 9.3 / Prediction-update reconstruction

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal column-stochastic matrix prediction followed by a nonnegative positive-evidence Bayesian update.

Declared targets.

- `KomorebiPaperVI.FiniteBayes.prediction_probability`. Source line 124.
- `KomorebiPaperVI.FiniteBayes.prediction_update_probability`. Source line 135.

12.22 10.1 / Fractional packet calculus

Recorded status. Complete scoped statement in the formal-core correspondence.

Fractional powers and transfer for the given finite spectral resolution, including zero eigenvalues; multiplication/addition laws, packet reduction, and identification with the actual algebra exponential. Finite spectral resolution is the stated starting data.

Declared targets.

- `KomorebiPaperVI.SpectralPackets.fractional_add`. Source line 58.
- `KomorebiPaperVI.SpectralPackets.spectral_transfer_exponential`. Source line 68.
- `KomorebiPaperVI.SpectralPackets.spectral_transfer_add`. Source line 77.
- `KomorebiPaperVI.SpectralPackets.fractional_reduces`. Source line 89.
- `KomorebiPaperVI.SpectralPackets.spectral_transfer_reduces`. Source line 96.

12.23 10.2 / Poisson subordination

Recorded status. Complete scoped statement in the formal-core correspondence.

The actual normalized improper Poisson integral for every nonnegative eigenvalue, including zero; Bochner integrability and equality with the actual exponential of the finite spectral square root.

Declared targets.

- `KomorebiPaperVI.PoissonKernel.kernel_mass`. Source line 35.
- `KomorebiPaperVI.PoissonKernel.heat_integrand_integrable`. Source line 54.
- `KomorebiPaperVI.ReciprocalGaussian.reciprocal_square_gaussian`. Source line 127.

- `KomorebiPaperVI.PoissonSubordination.scalar_subordination`. Source line 42.
- `KomorebiPaperVI.PoissonSubordination.finite_spectral_subordination`. Source line 52.
- `KomorebiPaperVI.PoissonSubordination.operator_subordination`. Source line 74.

12.24 10.3 / Fractional intertwining theorem

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual normalized Lebesgue/Bochner resolvent-defect integral for bounded nonnegative self-adjoint operators between distinct complex Hilbert spaces, the exact Holder constant, the alpha*m^(alpha-1) common-gap estimate, and exact fractional intertwining. Real finite matrices use their standard complexification; this ordinary specialization is manually reviewed.

Declared targets.

- `KomorebiPaperVI.BetaNormalization.normalizer_exact`. Source line 80.
- `KomorebiPaperVI.BetaNormalization.normalized_gap_integral`. Source line 164.
- `KomorebiPaperVI.NormalizedFractional.cfc_lebesgue_integral`. Source line 75.
- `KomorebiPaperVI.PositiveResolvent.resolvent_inverse`. Source line 18.
- `KomorebiPaperVI.PositiveResolvent.resolvent_gap_norm`. Source line 45.
- `KomorebiPaperVI.FractionalNormBounds.holder_estimate`. Source line 97.
- `KomorebiPaperVI.FractionalNormBounds.common_gap_estimate`. Source line 67.
- `KomorebiPaperVI.RectangularFractional.literal_resolvent_integral`. Source line 95.
- `KomorebiPaperVI.RectangularFractional.holder_estimate`. Source line 173.
- `KomorebiPaperVI.RectangularFractional.common_gap_estimate`. Source line 145.
- `KomorebiPaperVI.RectangularFractional.exact_fractional_intertwining`. Source line 110.

12.25 10.4 / Fractional semigroup refinement

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual fractional exponentials between distinct complex Hilbert spaces, rectangular Duhamel identity, contraction estimate, and the final common-gap bound in terms of the original generator defect.

Declared targets.

- `KomorebiPaperVI.RectangularSemigroup.evolution_exponential`. Source line 19.
- `KomorebiPaperVI.RectangularSemigroup.contraction`. Source line 30.
- `KomorebiPaperVI.RectangularSemigroup.gap_decay`. Source line 35.
- `KomorebiPaperVI.RectangularSemigroup.fractional_duhamel`. Source line 41.
- `KomorebiPaperVI.RectangularSemigroup.fractional_contraction_bound`. Source line 53.
- `KomorebiPaperVI.RectangularSemigroup.generator_gap_bound`. Source line 94.

12.26 10.5 / Fractional refinement towers

Recorded status. Complete scoped statement in the formal-core correspondence.

Varying complex Hilbert-space towers: both printed generator criteria imply summability of fractional-defect norms and time-defect norms, with uniform convergence of the real norm series on bounded nonnegative time intervals. Operator-series convergence is additionally proved in the inherited fixed-space formulation; no sum across unrelated operator spaces is asserted.

Declared targets.

- `KomorebiPaperVI.RectangularTowers.generator_holder_summable`. Source line 40.
- `KomorebiPaperVI.RectangularTowers.generator_gap_summable`. Source line 49.
- `KomorebiPaperVI.RectangularTowers.bounded_time_majorant`. Source line 58.
- `KomorebiPaperVI.RectangularTowers.time_norms_summable`. Source line 64.
- `KomorebiPaperVI.RectangularTowers.bounded_time_uniform_convergence`. Source line 72.
- `KomorebiPaperVI.RectangularTowers.holder_tower`. Source line 82.
- `KomorebiPaperVI.RectangularTowers.gap_tower`. Source line 93.
- `KomorebiPaperVI.FractionalTowers.bounded_time_uniform_convergence`. Source line 35.

12.27 11.1 / Exact discrete Green reconstruction at grid nodes

Recorded status. Complete scoped statement in the formal-core correspondence.

The literal finite Dirichlet tridiagonal matrix, its actual inverse, the scaled discrete Laplacian inverse, and the exact grid-node continuum Green identity for h=1/(n+1).

Declared targets.

- `KomorebiPaperVI.DiscreteGreen.green_second_difference`. Source line 22.
- `KomorebiPaperVI.DiscreteGreen.tridiagonal_inverse_kernel`. Source line 85.
- `KomorebiPaperVI.DiscreteGreen.inverse_eq_kernel`. Source line 109.
- `KomorebiPaperVI.DiscreteGreen.laplacian_inverse`. Source line 115.
- `KomorebiPaperVI.DiscreteGreen.grid_reconstruction`. Source line 122.

12.28 11.2 / Second-order low-mode convergence

Recorded status. Complete scoped statement in the formal-core correspondence.

The exact bound 0 <= w^2 - 4/h^2 sin(w*h/2)^2 <= w^4*h^2/12 for every nonzero spacing, with w=k*pi. This uniform bound implies the printed O(h^2) assertion as positive h tends to zero.

Declared targets.

- `KomorebiPaperVI.LowModeConvergence.sine_square_error`. Source line 8.
- `KomorebiPaperVI.LowModeConvergence.low_mode_error`. Source line 25.

12.29 12.1 / Discrete Feynman–Kac path formula

Recorded status. Complete scoped statement in the formal-core correspondence.

The actual potential-weighted matrix power equals the expectation over all finite paths; the path law is nonnegative and has total mass one for column-stochastic K. Product transition probabilities, the sum over departure-state potentials, and the terminal-state index are proved explicitly.

Declared targets.

- `KomorebiPaperVI.FeynmanKac.path_probability_nonnegative`. Source line 41.
- `KomorebiPaperVI.FeynmanKac.path_probability_mass`. Source line 48.
- `KomorebiPaperVI.FeynmanKac.path_probability_product`. Source line 85.
- `KomorebiPaperVI.FeynmanKac.path_potential_sum`. Source line 95.
- `KomorebiPaperVI.FeynmanKac.terminal_last`. Source line 104.

- `KomorebiPaperVI.FeynmanKac.feynman_kac_path_formula`. Source line 74.

12.30 13.1 / Schur–Feshbach projected resolvent

Recorded status. Complete scoped statement in the formal-core correspondence.

Literal finite complex block matrices: under invertibility of the full resolvent matrix and the hidden resolvent block, the retained block of the actual inverse is the inverse Schur complement; invertibility equivalence is included.

Declared targets.

- `KomorebiPaperVI.SchurFeshbach.resolvent_blocks`. Source line 14.
- `KomorebiPaperVI.SchurFeshbach.projected_resolvent`. Source line 21.
- `KomorebiPaperVI.SchurFeshbach.schur_invertible`. Source line 40.

12.31 13.2 / Memory/self-energy correspondence

Recorded status. Complete scoped statement in the formal-core correspondence.

For arbitrary finite complex matrices, including nonnormal matrices and nilpotent Jordan parts, the strict spectral half-plane implies actual Bochner integrability and decay of the complex-damped exponential. Both inverse identities are derived by the improper-integral fundamental theorem. The literal rectangular retained/hidden memory integral equals U*(zI-A)^(-1)*V, with the actual matrix inverse and no supplied growth bound.

Declared targets.

- `KomorebiPaperVI.ScalarDecay.exp_pow_integrable`. Source line 16.
- `KomorebiPaperVI.ScalarDecay.exp_pow_tendsto_zero`. Source line 25.
- `KomorebiPaperVI.NonnormalMemory.exp_mul_component`. Source line 38.
- `KomorebiPaperVI.NonnormalMemory.spectral_decay`. Source line 99.
- `KomorebiPaperVI.NonnormalMemory.spectral_laplace_inverse`. Source line 114.
- `KomorebiPaperVI.MatrixMemory.rectangular_memory_self_energy`. Source line 19.

12.32 14.1 / Semigroup survival identities

Recorded status. Complete scoped statement in the formal-core correspondence.

Actual finite Metzler-subgenerator exponential: survival derivative and strict positivity at finite nonnegative time, monotonicity, integrated hazard, existence of the survival limit, total defective density mass 1 minus that limit, and the eventual-absorption specialization.

Declared targets.

- `KomorebiPaperVI.Survival.survival_derivative`. Source line 54.
- `KomorebiPaperVI.Survival.survival_positive`. Source line 30.
- `KomorebiPaperVI.Survival.survival_antitone`. Source line 69.
- `KomorebiPaperVI.Survival.survival_hazard`. Source line 82.
- `KomorebiPaperVI.Survival.survival_has_limit`. Source line 100.
- `KomorebiPaperVI.Survival.defective_density_mass`. Source line 115.
- `KomorebiPaperVI.Survival.eventual_absorption_density`. Source line 129.

12.33 15.1 / Naturality under exact intertwining

Recorded status. Complete scoped statement in the formal-core correspondence.

Canonical basis-free finite Taylor-jet functional calculus on generalized eigenspaces, with uniqueness, truncation stability, dependence only on spectral jets, admissible local Cauchy circles in every declared open holomorphic neighborhood, and independence of those circles. Exact naturality holds for literal rectangular complex matrices without assuming a contour representation or one globally convergent Taylor series. The paper now states this standard finite-dimensional definition explicitly; real matrices are treated by the previously disclosed ordinary complexification specialization.

Declared targets.

- `KomorebiPaperVI.CanonicalJets.calculus_on_component`. Source line 28.
- `KomorebiPaperVI.CanonicalJets.jet_stable`. Source line 46.
- `KomorebiPaperVI.CanonicalJets.canonical_naturality`. Source line 81.
- `KomorebiPaperVI.HolomorphicFinite.canonical_unique`. Source line 17.
- `KomorebiPaperVI.HolomorphicFinite.calculus_spectrum_congr`. Source line 38.
- `KomorebiPaperVI.HolomorphicFinite.cauchyCoefficient_eq`. Source line 58.
- `KomorebiPaperVI.HolomorphicFinite.canonical_contours_exist`. Source line 96.
- `KomorebiPaperVI.HolomorphicFinite.canonical_contours_independent`. Source line 113.
- `KomorebiPaperVI.HolomorphicFinite.matrix_holomorphic_naturality`. Source line 129.

13 All 341 supporting theorem declarations

The inventory is generated from the saved source contract. “Support” means the declaration is included in the audited target inventory but is not named directly as an endpoint in the 33-row manuscript ledger. It can still support a ledger endpoint. This is not a proof-dependency computation. Source lines refer to the immutable production files, not the display excerpts in this guide.

13.1 Algebra: 9 theorem declarations

Source: `KomorebiPaperVI/Algebra.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.Algebra.block_reconstruction`	12; 2.1
`KomorebiPaperVI.Algebra.block_multiplication`	16; support
`KomorebiPaperVI.Algebra.packet_semigroup_convolution`	28; support
`KomorebiPaperVI.Algebra.difference_idempotent`	35; support
`KomorebiPaperVI.Algebra.difference_annihilates_coarse`	39; support
`KomorebiPaperVI.Algebra.tower_telescope`	44; support
`KomorebiPaperVI.Algebra.intertwining_powers`	53; 5.1
`KomorebiPaperVI.Algebra.finite_horizon_defect`	64; 5.2
`KomorebiPaperVI.Algebra.rectangular_resolvent_identity`	85; 3.3

13.2 BanachDefects: 3 theorem declarations

Source: `KomorebiPaperVI/BanachDefects.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BanachDefects.duhamel_derivative`	8; support
`KomorebiPaperVI.BanachDefects.duhamel`	21; support
`KomorebiPaperVI.BanachDefects.contraction_bound`	29; support

13.3 BayesianFlow: 8 theorem declarations

Source: `KomorebiPaperVI/BayesianFlow.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BayesianFlow.positive_evidence`	13; 9.1
`KomorebiPaperVI.BayesianFlow.flow_probability`	25; 9.1
`KomorebiPaperVI.BayesianFlow.likelihood_add`	30; support
`KomorebiPaperVI.BayesianFlow.flow_add`	36; 9.1
`KomorebiPaperVI.BayesianFlow.flow_zero`	45; 9.1
`KomorebiPaperVI.BayesianFlow.likelihood_derivative`	52; support
`KomorebiPaperVI.BayesianFlow.evidence_derivative`	56; support
`KomorebiPaperVI.BayesianFlow.flow_derivative`	64; 9.1

13.4 BayesianNecessity: 9 theorem declarations

Source: `KomorebiPaperVI/BayesianNecessity.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BayesianNecessity.pair_weighted_sum`	13; support
`KomorebiPaperVI.BayesianNecessity.pair_probability`	19; support
`KomorebiPaperVI.BayesianNecessity.pair_fiber_weight`	25; support
`KomorebiPaperVI.BayesianNecessity.pair_coarsen`	35; support
`KomorebiPaperVI.BayesianNecessity.pair_coarse_posterior`	54; support
`KomorebiPaperVI.BayesianNecessity.cross_packet_ratio`	67; 8.2
`KomorebiPaperVI.BayesianNecessity.all_packet_ratio`	85; support
`KomorebiPaperVI.BayesianNecessity.positive_evidence`	101; support
`KomorebiPaperVI.BayesianNecessity.refinement_iff`	112; 8.2

13.5 BayesianPartitions: 9 theorem declarations

Source: `KomorebiPaperVI/BayesianPartitions.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BayesianPartitions.packetMap_surjective`	14; support
`KomorebiPaperVI.BayesianPartitions.packetLikelihood_apply`	24; support
`KomorebiPaperVI.BayesianPartitions.packet_eq_iff`	28; 8.6
`KomorebiPaperVI.BayesianPartitions.packetLikelihood_positive`	33; support
`KomorebiPaperVI.BayesianPartitions.canonical_coarsening`	38; 8.6
`KomorebiPaperVI.BayesianPartitions.canonical_unique_factor`	46; 8.6
`KomorebiPaperVI.BayesianPartitions.all_prior_family_iff_resolves`	67; 8.6
`KomorebiPaperVI.BayesianPartitions.canonical_coarsest_nontrivial`	85; 8.6
`KomorebiPaperVI.BayesianPartitions.canonical_nontrivial`	91; 8.6

13.6 BayesianTowers: 7 theorem declarations

Source: `KomorebiPaperVI/BayesianTowers.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BayesianTowers.coarsen_comp`	12; 8.5
`KomorebiPaperVI.BayesianTowers.singleton_commutes`	30; 8.2
`KomorebiPaperVI.BayesianTowers.adjacent_criterion`	43; 8.5
`KomorebiPaperVI.BayesianTowers.compose_commutes`	57; support
`KomorebiPaperVI.BayesianTowers.coarsen_id`	70; support
`KomorebiPaperVI.BayesianTowers.tower_commutes`	73; 8.5
`KomorebiPaperVI.BayesianTowers.tower_criterion`	80; 8.5

13.7 BetaNormalization: 15 theorem declarations

Source: `KomorebiPaperVI/BetaNormalization.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.BetaNormalization.beta_value`	12; support
`KomorebiPaperVI.BetaNormalization.change_derivative`	33; support
`KomorebiPaperVI.BetaNormalization.change_injective`	38; support
`KomorebiPaperVI.BetaNormalization.change_image`	46; support
`KomorebiPaperVI.BetaNormalization.change_integrand`	60; support
`KomorebiPaperVI.BetaNormalization.rational_integral`	70; support
`KomorebiPaperVI.BetaNormalization.normalizer_exact`	80; 10.3
`KomorebiPaperVI.BetaNormalization.sine_factor_positive`	88; support
`KomorebiPaperVI.BetaNormalization.scalar_fractional_integral`	95; support
`KomorebiPaperVI.BetaNormalization.beta_real_integral`	99; support
`KomorebiPaperVI.BetaNormalization.weighted_beta_value`	116; support
`KomorebiPaperVI.BetaNormalization.gap_change_integrand`	123; support
`KomorebiPaperVI.BetaNormalization.unit_gap_integral`	132; support
`KomorebiPaperVI.BetaNormalization.gap_integral`	143; support
`KomorebiPaperVI.BetaNormalization.normalized_gap_integral`	164; 10.3

13.8 CanonicalJets: 8 theorem declarations

Source: `KomorebiPaperVI/CanonicalJets.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.CanonicalJets.calculus_on_component`	28; 15.1
`KomorebiPaperVI.CanonicalJets.power_kills`	38; support
`KomorebiPaperVI.CanonicalJets.jet_stable`	46; 15.1
`KomorebiPaperVI.CanonicalJets.shift_intertwines`	58; support
`KomorebiPaperVI.CanonicalJets.power_intertwines`	65; support
`KomorebiPaperVI.CanonicalJets.jet_intertwines`	75; support
`KomorebiPaperVI.CanonicalJets.canonical_naturality`	81; 15.1
`KomorebiPaperVI.CanonicalJets.holomorphic_naturality`	98; support

13.9 Coarsening: 21 theorem declarations

Source: `KomorebiPaperVI/Coarsening.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.Coarsening.coarsen_mass`	15; support
`KomorebiPaperVI.Coarsening.coarsen_nonnegative`	21; support
`KomorebiPaperVI.Coarsening.coarsen_probability`	29; support
`KomorebiPaperVI.Coarsening.coarsen_div`	34; support
`KomorebiPaperVI.Coarsening.coarsen_posterior`	43; support
`KomorebiPaperVI.Coarsening.zero_packet_component`	48; 7.2

Declaration	Line; ledger
KomorebiPaperVI.Coarsening.zero_packet_weighted	57; support
KomorebiPaperVI.Coarsening.effective_weight	66; 8.1
KomorebiPaperVI.Coarsening.effective_nonnegative	73; support
KomorebiPaperVI.Coarsening.effective_evidence	78; 8.1
KomorebiPaperVI.Coarsening.effective_factorization	83; 8.1
KomorebiPaperVI.Coarsening.packet_reconstruction_point	90; 7.2
KomorebiPaperVI.Coarsening.packet_reconstruction	99; 7.2
KomorebiPaperVI.Coarsening.blockwise_weight	106; support
KomorebiPaperVI.Coarsening.blockwise_evidence	115; support
KomorebiPaperVI.Coarsening.blockwise_sufficiency	120; support
KomorebiPaperVI.Coarsening.scaled_blockwise_sufficiency	126; support
KomorebiPaperVI.Coarsening.one_packet_posterior	135; support
KomorebiPaperVI.Profiles.profile_equivalence	147; support
KomorebiPaperVI.Profiles.likelihood_resolving_iff_refines	153; support
KomorebiPaperVI.Profiles.exact_profile_factorization	159; support

13.10 ComplexMemory: 7 theorem declarations

Source: KomorebiPaperVI/ComplexMemory.lean.

Declaration	Line; ledger
KomorebiPaperVI.ComplexMemory.damped_eq_shifted	18; support
KomorebiPaperVI.ComplexMemory.shifted_decay	29; support
KomorebiPaperVI.ComplexMemory.integral_transfer_right	43; support
KomorebiPaperVI.ComplexMemory.laplace_inverse	55; support
KomorebiPaperVI.ComplexMemory.memory_self_energy	77; support
KomorebiPaperVI.ComplexMemory.normal_growth	91; support
KomorebiPaperVI.ComplexMemory.normal_memory_self_energy	101; support

13.11 ComplexSemigroups: 4 theorem declarations

Source: KomorebiPaperVI/ComplexSemigroups.lean.

Declaration	Line; ledger
KomorebiPaperVI.ComplexSemigroups.duhamel_integrand_derivative	13; support
KomorebiPaperVI.ComplexSemigroups.duhamel_rectangular	29; support
KomorebiPaperVI.ComplexSemigroups.exact_rectangular_intertwining	40; support
KomorebiPaperVI.ComplexSemigroups.contraction_duhamel_bound	48; support

13.12 ConditionalExpectation: 11 theorem declarations

Source: `KomorebiPaperVI/ConditionalExpectation.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.ConditionalExpectation.fiber_mass_cast`	17; support
`KomorebiPaperVI.ConditionalExpectation.fiber_pull`	23; support
`KomorebiPaperVI.ConditionalExpectation.fiber_mass_ne`	32; support
`KomorebiPaperVI.ConditionalExpectation.expect_constant`	39; support
`KomorebiPaperVI.ConditionalExpectation.expect_block_constant`	46; support
`KomorebiPaperVI.ConditionalExpectation.expect_idempotent`	49; 6.1
`KomorebiPaperVI.ConditionalExpectation.expect_fixes_iff`	53; support
`KomorebiPaperVI.ConditionalExpectation.expect_range`	74; 6.1
`KomorebiPaperVI.ConditionalExpectation.expect_matrix`	84; 6.1
`KomorebiPaperVI.ConditionalExpectation.kernel_weighted_symmetry`	94; support
`KomorebiPaperVI.ConditionalExpectation.expect_self_adjoint`	105; 6.1

13.13 ConditionalTowers: 7 theorem declarations

Source: `KomorebiPaperVI/ConditionalTowers.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.ConditionalTowers.weighted_inner_separates`	11; support
`KomorebiPaperVI.ConditionalTowers.fine_after_coarse`	19; support
`KomorebiPaperVI.ConditionalTowers.coarse_after_fine`	26; support
`KomorebiPaperVI.ConditionalTowers.conditional_kernel_product`	37; 6.2
`KomorebiPaperVI.ConditionalTowers.nested_kernel_min`	53; 6.2
`KomorebiPaperVI.ConditionalTowers.martingale_packet_product`	65; 6.2
`KomorebiPaperVI.ConditionalTowers.martingale_packet_reconstruction`	72; 6.2

13.14 ContourNaturality: 4 theorem declarations

Source: `KomorebiPaperVI/ContourNaturality.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.ContourNaturality.circle_mul_left`	20; support
`KomorebiPaperVI.ContourNaturality.circle_mul_right`	31; support
`KomorebiPaperVI.ContourNaturality.resolvent_intertwines`	44; support
`KomorebiPaperVI.ContourNaturality.common_contour_naturality`	57; support

13.15 DiscreteGreen: 9 theorem declarations

Source: `KomorebiPaperVI/DiscreteGreen.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.DiscreteGreen.green_zero`	13; support
`KomorebiPaperVI.DiscreteGreen.green_end`	15; support
`KomorebiPaperVI.DiscreteGreen.green_second_difference`	22; 11.1
`KomorebiPaperVI.DiscreteGreen.sparse_sum`	52; support
`KomorebiPaperVI.DiscreteGreen.tridiagonal_mul`	65; support
`KomorebiPaperVI.DiscreteGreen.tridiagonal_inverse_kernel`	85; 11.1
`KomorebiPaperVI.DiscreteGreen.inverse_eq_kernel`	109; 11.1
`KomorebiPaperVI.DiscreteGreen.laplacian_inverse`	115; 11.1
`KomorebiPaperVI.DiscreteGreen.grid_reconstruction`	122; 11.1

13.16 FeynmanKac: 10 theorem declarations

Source: `KomorebiPaperVI/FeynmanKac.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.FeynmanKac.sum_paths_succ`	30; support
`KomorebiPaperVI.FeynmanKac.transfer_entry`	37; support
`KomorebiPaperVI.FeynmanKac.path_probability_nonnegative`	41; 12.1
`KomorebiPaperVI.FeynmanKac.path_probability_mass`	48; 12.1
`KomorebiPaperVI.FeynmanKac.path_expectation_zero`	57; support
`KomorebiPaperVI.FeynmanKac.path_expectation_step`	61; support
`KomorebiPaperVI.FeynmanKac.feynman_kac_path_formula`	74; 12.1
`KomorebiPaperVI.FeynmanKac.path_probability_product`	85; 12.1
`KomorebiPaperVI.FeynmanKac.path_potential_sum`	95; 12.1
`KomorebiPaperVI.FeynmanKac.terminal_last`	104; 12.1

13.17 FiniteBayes: 14 theorem declarations

Source: `KomorebiPaperVI/FiniteBayes.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.FiniteBayes.normalized_mass`	15; support
`KomorebiPaperVI.FiniteBayes.posterior_mass`	19; support
`KomorebiPaperVI.FiniteBayes.posterior_nonnegative`	23; support
`KomorebiPaperVI.FiniteBayes.evidence_scale`	27; support
`KomorebiPaperVI.FiniteBayes.posterior_scale`	31; support
`KomorebiPaperVI.FiniteBayes.sequential_evidence`	37; 7.1
`KomorebiPaperVI.FiniteBayes.posterior_composition`	45; 7.1
`KomorebiPaperVI.FiniteBayes.posterior_difference`	53; support
`KomorebiPaperVI.FiniteBayes.scale_null_direction`	65; support
`KomorebiPaperVI.FiniteBayes.normalized_l1_bound_left`	73; 7.3

Declaration	Line; ledger
KomorebiPaperVI.FiniteBayes.likelihood_perturbation_bound	102; 7.3, 8.4
KomorebiPaperVI.FiniteBayes.posterior_probability	120; support
KomorebiPaperVI.FiniteBayes.prediction_probability	124; 9.3
KomorebiPaperVI.FiniteBayes.prediction_update_probability	135; 9.3

13.18 FractionalIntegral: 3 theorem declarations

Source: KomorebiPaperVI/FractionalIntegral.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalIntegral.common_measure_defect	14; support
KomorebiPaperVI.FractionalIntegral.scalar_normalizer_positive	32; support
KomorebiPaperVI.FractionalIntegral.scalar_normalized_representation	36; support

13.19 FractionalNormBounds: 6 theorem declarations

Source: KomorebiPaperVI/FractionalNormBounds.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalNormBounds.integrand_saturation	16; support
KomorebiPaperVI.FractionalNormBounds.integrand_low_bound	29; support
KomorebiPaperVI.FractionalNormBounds.integrand_gap_bound	44; support
KomorebiPaperVI.FractionalNormBounds.integrand_high_bound	61; support
KomorebiPaperVI.FractionalNormBounds.common_gap_estimate	67; 10.3
KomorebiPaperVI.FractionalNormBounds.holder_estimate	97; 10.3

13.20 FractionalResolvent: 4 theorem declarations

Source: KomorebiPaperVI/FractionalResolvent.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalResolvent.cfc_integrand	13; support
KomorebiPaperVI.FractionalResolvent.integrand_defect	26; support
KomorebiPaperVI.FractionalResolvent.normalized_defect	38; support
KomorebiPaperVI.FractionalResolvent.exact_fractional_intertwining	59; support

13.21 FractionalScalarBounds: 5 theorem declarations

Source: KomorebiPaperVI/FractionalScalarBounds.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalScalarBounds.low_integral	8; support
KomorebiPaperVI.FractionalScalarBounds.high_integral	14; support
KomorebiPaperVI.FractionalScalarBounds.split_domain	20; support
KomorebiPaperVI.FractionalScalarBounds.split_norm_bound	33; support
KomorebiPaperVI.FractionalScalarBounds.optimize_split	56; support

13.22 FractionalSemigroup: 8 theorem declarations

Source: KomorebiPaperVI/FractionalSemigroup.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalSemigroup.power_nonnegative	19; support
KomorebiPaperVI.FractionalSemigroup.exponential_cfc	21; support
KomorebiPaperVI.FractionalSemigroup.exponential_contraction	33; support
KomorebiPaperVI.FractionalSemigroup.exponential_gap	42; support
KomorebiPaperVI.FractionalSemigroup.evolution_transfer	55; support
KomorebiPaperVI.FractionalSemigroup.fractional_duhamel	58; support
KomorebiPaperVI.FractionalSemigroup.fractional_contraction_bound	69; support
KomorebiPaperVI.FractionalSemigroup.fractional_gap_bound	81; support

13.23 FractionalTowers: 4 theorem declarations

Source: KomorebiPaperVI/FractionalTowers.lean.

Declaration	Line; ledger
KomorebiPaperVI.FractionalTowers.bounded_time_majorant	15; support
KomorebiPaperVI.FractionalTowers.time_defects_norm_summable	21; support
KomorebiPaperVI.FractionalTowers.time_defects_summable	30; support
KomorebiPaperVI.FractionalTowers.bounded_time_uniform_convergence	35; 10.5

13.24 HolomorphicContours: 3 theorem declarations

Source: KomorebiPaperVI/HolomorphicContours.lean.

Declaration	Line; ledger
KomorebiPaperVI.HolomorphicContours.resolvent_continuousAt	15; support
KomorebiPaperVI.HolomorphicContours.holomorphic_circle_integrable	31; support
KomorebiPaperVI.HolomorphicContours.holomorphic_common_contour_naturality	42; support

13.25 HolomorphicFinite: 11 theorem declarations

Source: KomorebiPaperVI/HolomorphicFinite.lean.

Declaration	Line; ledger
KomorebiPaperVI.HolomorphicFinite.canonical_unique	17; 15.1
KomorebiPaperVI.HolomorphicFinite.holomorphic_on_component	25; support
KomorebiPaperVI.HolomorphicFinite.component_mem_spectrum	33; support
KomorebiPaperVI.HolomorphicFinite.calculus_spectrum_congr	38; 15.1
KomorebiPaperVI.HolomorphicFinite.cauchyCoefficient_eq	58; 15.1
KomorebiPaperVI.HolomorphicFinite.cauchyCoefficient_independent	70; support
KomorebiPaperVI.HolomorphicFinite.admissible_circle	77; support
KomorebiPaperVI.HolomorphicFinite.canonical_cauchy_representation	88; support
KomorebiPaperVI.HolomorphicFinite.canonical_contours_exist	96; 15.1
KomorebiPaperVI.HolomorphicFinite.canonical_contours_independent	113; 15.1
KomorebiPaperVI.HolomorphicFinite.matrix_holomorphic_naturality	129; 15.1

13.26 InformationLoss: 6 theorem declarations

Source: KomorebiPaperVI/InformationLoss.lean.

Declaration	Line; ledger
KomorebiPaperVI.InformationLoss.lossMatrix_apply	12; support
KomorebiPaperVI.InformationLoss.information_loss_matrix	17; 8.4
KomorebiPaperVI.InformationLoss.information_loss_null	28; 8.4
KomorebiPaperVI.InformationLoss.fiber_loss_bound	35; support
KomorebiPaperVI.InformationLoss.total_loss_bound	60; 8.4
KomorebiPaperVI.InformationLoss.information_loss_norm	68; 8.4

13.27 LowModeConvergence: 2 theorem declarations

Source: KomorebiPaperVI/LowModeConvergence.lean.

Declaration	Line; ledger
KomorebiPaperVI.LowModeConvergence.sine_square_error	8; 11.2
KomorebiPaperVI.LowModeConvergence.low_mode_error	25; 11.2

13.28 MarkovSemigroup: 8 theorem declarations

Source: KomorebiPaperVI/MarkovSemigroup.lean.

Declaration	Line; ledger
KomorebiPaperVI.MarkovSemigroup.matrix_powers_nonnegative	12; support

Declaration	Line; ledger
KomorebiPaperVI.MarkovSemigroup.exponential_nonnegative	20; support
KomorebiPaperVI.MarkovSemigroup.shifted_generator_nonnegative	29; support
KomorebiPaperVI.MarkovSemigroup.scalar_exponential	40; support
KomorebiPaperVI.MarkovSemigroup.markov_nonnegative	48; 4.1
KomorebiPaperVI.MarkovSemigroup.generator_power_columns	63; support
KomorebiPaperVI.MarkovSemigroup.markov_column_sum	71; 4.1
KomorebiPaperVI.MarkovSemigroup.markov_probability	84; 4.1

13.29 MartingalePackets: 2 theorem declarations

Source: KomorebiPaperVI/MartingalePackets.lean.

Declaration	Line; ledger
KomorebiPaperVI.MartingalePackets.packet_product	12; support
KomorebiPaperVI.MartingalePackets.packet_sum	40; support

13.30 MatrixMemory: 1 theorem declarations

Source: KomorebiPaperVI/MatrixMemory.lean.

Declaration	Line; ledger
KomorebiPaperVI.MatrixMemory.rectangular_memory_self_energy	19; 13.2

13.31 MemoryLaplace: 5 theorem declarations

Source: KomorebiPaperVI/MemoryLaplace.lean.

Declaration	Line; ledger
KomorebiPaperVI.MemoryLaplace.decay_integrable	13; support
KomorebiPaperVI.MemoryLaplace.decay_tendsto_zero	22; support
KomorebiPaperVI.MemoryLaplace.integral_generator_transfer	33; support
KomorebiPaperVI.MemoryLaplace.shifted_laplace_right_inverse	45; support
KomorebiPaperVI.MemoryLaplace.memory_integral	51; support

13.32 NonnormalMemory: 8 theorem declarations

Source: `KomorebiPaperVI/NonnormalMemory.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.NonnormalMemory.left_shift_power`	16; support
`KomorebiPaperVI.NonnormalMemory.component_spectrum`	24; support
`KomorebiPaperVI.NonnormalMemory.exp_mul_component`	38; 13.2
`KomorebiPaperVI.NonnormalMemory.damped_mul_component`	64; support
`KomorebiPaperVI.NonnormalMemory.component_decay`	76; support
`KomorebiPaperVI.NonnormalMemory.spectral_decay`	99; 13.2
`KomorebiPaperVI.NonnormalMemory.spectral_laplace_inverse`	114; 13.2
`KomorebiPaperVI.NonnormalMemory.nonnormal_self_energy`	142; support

13.33 NormalizedFractional: 3 theorem declarations

Source: `KomorebiPaperVI/NormalizedFractional.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.NormalizedFractional.scalar_measure`	19; support
`KomorebiPaperVI.NormalizedFractional.cfc_explicit_measure`	32; support
`KomorebiPaperVI.NormalizedFractional.cfc_lebesgue_integral`	75; 10.3

13.34 OperatorBounds: 3 theorem declarations

Source: `KomorebiPaperVI/OperatorBounds.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.OperatorBounds.growth_duhamel_bound`	10; 3.2
`KomorebiPaperVI.OperatorBounds.contraction_powers`	40; support
`KomorebiPaperVI.OperatorBounds.finite_horizon_contraction`	50; 5.2

13.35 PoissonKernel: 9 theorem declarations

Source: `KomorebiPaperVI/PoissonKernel.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.PoissonKernel.inverse_half_square`	14; support
`KomorebiPaperVI.PoissonKernel.kernel_substitution`	18; support
`KomorebiPaperVI.PoissonKernel.kernel_integrable`	28; support
`KomorebiPaperVI.PoissonKernel.kernel_mass`	35; 10.2
`KomorebiPaperVI.PoissonKernel.kernel_nonnegative`	43; support
`KomorebiPaperVI.PoissonKernel.heat_integrand_bounds`	47; support

Declaration	Line; ledger
KomorebiPaperVI.PoissonKernel.heat_integrand_integrable	54; 10.2
KomorebiPaperVI.PoissonKernel.zero_mode_subordination	67; support
KomorebiPaperVI.PoissonKernel.finite_spectral_integrable	72; support

13.36 PoissonSubordination: 6 theorem declarations

Source: KomorebiPaperVI/PoissonSubordination.lean.

Declaration	Line; ledger
KomorebiPaperVI.PoissonSubordination.scaled_reciprocal_gaussian	9; support
KomorebiPaperVI.PoissonSubordination.heat_substitution	30; support
KomorebiPaperVI.PoissonSubordination.scalar_subordination	42; 10.2
KomorebiPaperVI.PoissonSubordination.finite_spectral_subordination	52; 10.2
KomorebiPaperVI.PoissonSubordination.operator_integrand	65; support
KomorebiPaperVI.PoissonSubordination.operator_subordination	74; 10.2

13.37 PositiveResolvent: 9 theorem declarations

Source: KomorebiPaperVI/PositiveResolvent.lean.

Declaration	Line; ledger
KomorebiPaperVI.PositiveResolvent.spectral_positive	13; support
KomorebiPaperVI.PositiveResolvent.resolvent_inverse	18; 10.3
KomorebiPaperVI.PositiveResolvent.resolvent_left_inverse	24; support
KomorebiPaperVI.PositiveResolvent.resolvent_right_inverse	37; support
KomorebiPaperVI.PositiveResolvent.resolvent_gap_norm	45; 10.3
KomorebiPaperVI.PositiveResolvent.resolvent_norm	53; support
KomorebiPaperVI.PositiveResolvent.resolvent_defect	57; support
KomorebiPaperVI.PositiveResolvent.saturation_cfc	71; support
KomorebiPaperVI.PositiveResolvent.saturation_norm	79; support

13.38 ReciprocalGaussian: 13 theorem declarations

Source: KomorebiPaperVI/ReciprocalGaussian.lean.

Declaration	Line; ledger
KomorebiPaperVI.ReciprocalGaussian.phi_derivative	8; support
KomorebiPaperVI.ReciprocalGaussian.phi_injective	14; support
KomorebiPaperVI.ReciprocalGaussian.phi_surjective	21; support
KomorebiPaperVI.ReciprocalGaussian.gaussian_jacobian_integrable	35; support
KomorebiPaperVI.ReciprocalGaussian.gaussian_jacobian_integral	45; support
KomorebiPaperVI.ReciprocalGaussian.reciprocal_derivative	59; support
KomorebiPaperVI.ReciprocalGaussian.reciprocal_image	65; support

Declaration	Line; ledger
KomorebiPaperVI.ReciprocalGaussian.reciprocal_injective	74; support
KomorebiPaperVI.ReciprocalGaussian.reciprocal_phi	81; support
KomorebiPaperVI.ReciprocalGaussian.reciprocal_gaussian_symmetry	86; support
KomorebiPaperVI.ReciprocalGaussian.phi_gaussian_integrable	100; support
KomorebiPaperVI.ReciprocalGaussian.phi_gaussian_integral	113; support
KomorebiPaperVI.ReciprocalGaussian.reciprocal_square_gaussian	127; 10.2

13.39 RectangularFractional: 10 theorem declarations

Source: KomorebiPaperVI/RectangularFractional.lean.

Declaration	Line; ledger
KomorebiPaperVI.RectangularFractional.normalized_integral	23; support
KomorebiPaperVI.RectangularFractional.low_bound	42; support
KomorebiPaperVI.RectangularFractional.resolvent_defect	59; support
KomorebiPaperVI.RectangularFractional.literal_integrand	80; support
KomorebiPaperVI.RectangularFractional.literal_resolvent_integral	95; 10.3
KomorebiPaperVI.RectangularFractional.exact_fractional_intertwining	110; 10.3
KomorebiPaperVI.RectangularFractional.gap_bound	121; support
KomorebiPaperVI.RectangularFractional.high_bound	139; support
KomorebiPaperVI.RectangularFractional.common_gap_estimate	145; 10.3
KomorebiPaperVI.RectangularFractional.holder_estimate	173; 10.3

13.40 RectangularSemigroup: 8 theorem declarations

Source: KomorebiPaperVI/RectangularSemigroup.lean.

Declaration	Line; ledger
KomorebiPaperVI.RectangularSemigroup.evolution_exponential	19; 10.4
KomorebiPaperVI.RectangularSemigroup.evolution_cfc	23; support
KomorebiPaperVI.RectangularSemigroup.contraction	30; 10.4
KomorebiPaperVI.RectangularSemigroup.gap_decay	35; 10.4
KomorebiPaperVI.RectangularSemigroup.fractional_duhamel	41; 10.4
KomorebiPaperVI.RectangularSemigroup.fractional_contraction_bound	53; 10.4
KomorebiPaperVI.RectangularSemigroup.fractional_gap_bound	63; support
KomorebiPaperVI.RectangularSemigroup.generator_gap_bound	94; 10.4

13.41 RectangularTowers: 9 theorem declarations

Source: `KomorebiPaperVI/RectangularTowers.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.RectangularTowers.holder_constant_positive`	21; support
`KomorebiPaperVI.RectangularTowers.holder_majorant`	27; support
`KomorebiPaperVI.RectangularTowers.generator_holder_summable`	40; 10.5
`KomorebiPaperVI.RectangularTowers.generator_gap_summable`	49; 10.5
`KomorebiPaperVI.RectangularTowers.bounded_time_majorant`	58; 10.5
`KomorebiPaperVI.RectangularTowers.time_norms_summable`	64; 10.5
`KomorebiPaperVI.RectangularTowers.bounded_time_uniform_convergence`	72; 10.5
`KomorebiPaperVI.RectangularTowers.holder_tower`	82; 10.5
`KomorebiPaperVI.RectangularTowers.gap_tower`	93; 10.5

13.42 ReducingPackets: 3 theorem declarations

Source: `KomorebiPaperVI/ReducingPackets.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.ReducingPackets.generator_commutes`	12; 2.2
`KomorebiPaperVI.ReducingPackets.reducing_exponential`	32; 2.2
`KomorebiPaperVI.ReducingPackets.compressed_evolution`	39; 2.2

13.43 ScalarDecay: 3 theorem declarations

Source: `KomorebiPaperVI/ScalarDecay.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.ScalarDecay.norm_exp_pow`	11; support
`KomorebiPaperVI.ScalarDecay.exp_pow_integrable`	16; 13.2
`KomorebiPaperVI.ScalarDecay.exp_pow_tendsto_zero`	25; 13.2

13.44 SchurFeshbach: 3 theorem declarations

Source: `KomorebiPaperVI/SchurFeshbach.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.SchurFeshbach.resolvent_blocks`	14; 13.1
`KomorebiPaperVI.SchurFeshbach.projected_resolvent`	21; 13.1
`KomorebiPaperVI.SchurFeshbach.schur_invertible`	40; 13.1

13.45 Semigroups: 11 theorem declarations

Source: `KomorebiPaperVI/Semigroups.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.Semigroups.transfer_zero`	18; 1.2
`KomorebiPaperVI.Semigroups.transfer_add`	20; 1.2
`KomorebiPaperVI.Semigroups.transfer_derivative_right`	26; 1.2
`KomorebiPaperVI.Semigroups.transfer_derivative_left`	29; 1.2
`KomorebiPaperVI.Semigroups.transfer_continuous`	33; support
`KomorebiPaperVI.Semigroups.transfer_intertwining`	36; support
`KomorebiPaperVI.Semigroups.exponential_packet_convolution`	44; 2.1
`KomorebiPaperVI.Semigroups.duhamel_integrand_derivative`	57; 3.1
`KomorebiPaperVI.Semigroups.duhamel_rectangular`	70; 3.1
`KomorebiPaperVI.Semigroups.exact_rectangular_intertwining`	81; 3.1
`KomorebiPaperVI.Semigroups.contraction_duhamel_bound`	89; 3.2

13.46 SpectralPackets: 9 theorem declarations

Source: `KomorebiPaperVI/SpectralPackets.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.SpectralPackets.resolve_mul`	12; support
`KomorebiPaperVI.SpectralPackets.resolve_left_packet`	19; support
`KomorebiPaperVI.SpectralPackets.resolve_right_packet`	25; support
`KomorebiPaperVI.SpectralPackets.exp_resolution`	39; support
`KomorebiPaperVI.SpectralPackets.fractional_add`	58; 10.1
`KomorebiPaperVI.SpectralPackets.spectral_transfer_exponential`	68; 10.1
`KomorebiPaperVI.SpectralPackets.spectral_transfer_add`	77; 10.1
`KomorebiPaperVI.SpectralPackets.fractional_reduces`	89; 10.1
`KomorebiPaperVI.SpectralPackets.spectral_transfer_reduces`	96; 10.1

13.47 Survival: 11 theorem declarations

Source: `KomorebiPaperVI/Survival.lean`.

Declaration	Line; ledger
`KomorebiPaperVI.Survival.survival_zero`	23; support
`KomorebiPaperVI.Survival.state_nonnegative`	26; support
`KomorebiPaperVI.Survival.survival_positive`	30; 14.1
`KomorebiPaperVI.Survival.density_nonnegative`	48; support
`KomorebiPaperVI.Survival.survival_derivative`	54; 14.1
`KomorebiPaperVI.Survival.survival_antitone`	69; 14.1
`KomorebiPaperVI.Survival.density_continuous`	78; support
`KomorebiPaperVI.Survival.survival_hazard`	82; 14.1
`KomorebiPaperVI.Survival.survival_has_limit`	100; 14.1

Declaration	Line; ledger
`KomorebiPaperVI.Survival.defective_density_mass`	115; 14.1
`KomorebiPaperVI.Survival.eventual_absorption_density`	129; 14.1

14 Local module imports and locked dependencies

All 47 production modules are imported by `KomorebiPaperVI.lean`. The table gives direct local imports only. Mathlib dependencies remain recorded in the source. Import edges describe module organization, not the complete dependency graph of individual proofs.

Module (targets)	Local imports
`Algebra` (9)	Mathlib only
`BanachDefects` (3)	`Semigroups`
`BayesianFlow` (8)	`FiniteBayes`
`BayesianNecessity` (9)	`Coarsening`
`BayesianPartitions` (9)	`BayesianNecessity`
`BayesianTowers` (7)	`BayesianNecessity`
`BetaNormalization` (15)	Mathlib only
`CanonicalJets` (8)	Mathlib only
`Coarsening` (21)	`FiniteBayes`
`ComplexMemory` (7)	`MemoryLaplace`
`ComplexSemigroups` (4)	`Semigroups`; `FractionalSemigroup`; `RectangularFractional`
`ConditionalExpectation` (11)	`Coarsening`
`ConditionalTowers` (7)	`ConditionalExpectation`; `Algebra`; `MartingalePackets`
`ContourNaturality` (4)	`Algebra`
`DiscreteGreen` (9)	Mathlib only
`FeynmanKac` (10)	Mathlib only
`FiniteBayes` (14)	Mathlib only
`FractionalIntegral` (3)	Mathlib only
`FractionalNormBounds` (6)	`FractionalResolvent`; `FractionalScalarBounds`
`FractionalResolvent` (4)	`NormalizedFractional`; `PositiveResolvent`
`FractionalScalarBounds` (5)	`BetaNormalization`
`FractionalSemigroup` (8)	`Semigroups`; `BanachDefects`
`FractionalTowers` (4)	`FractionalSemigroup`
`HolomorphicContours` (3)	`ContourNaturality`
`HolomorphicFinite` (11)	`CanonicalJets`
`InformationLoss` (6)	`Coarsening`
`LowModeConvergence` (2)	Mathlib only
`MarkovSemigroup` (8)	Mathlib only
`MartingalePackets` (2)	`Algebra`
`MatrixMemory` (1)	`NonnormalMemory`
`MemoryLaplace` (5)	`Semigroups`
`NonnormalMemory` (8)	`ComplexMemory`; `CanonicalJets`; `ScalarDecay`
`NormalizedFractional` (3)	`BetaNormalization`; `FractionalIntegral`
`OperatorBounds` (3)	`Semigroups`
`PoissonKernel` (9)	`SpectralPackets`
`PoissonSubordination` (6)	`PoissonKernel`; `ReciprocalGaussian`
`PositiveResolvent` (9)	`FractionalSemigroup`
`ReciprocalGaussian` (13)	Mathlib only
`RectangularFractional` (10)	`FractionalNormBounds`
`RectangularSemigroup` (8)	`ComplexSemigroups`
`RectangularTowers` (9)	`RectangularSemigroup`
`ReducingPackets` (3)	`Semigroups`
`ScalarDecay` (3)	Mathlib only
`SchurFeshbach` (3)	Mathlib only

Module (targets)	Local imports
`Semigroups` (11)	`Algebra`
`SpectralPackets` (9)	`Semigroups`
`Survival` (11)	`MarkovSemigroup`; `Semigroups`

Quantum Seeds and Effective Dynamics

The formal companion to Chapter VII explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2055

Return to Chapter VII

How to use this volume

A guide to the actual formal development. The formal development covers 19 finite items through 323 named theorem declarations in 25 proof modules. The teaching chapters explain the mathematical interfaces and their hypotheses. The complete registers then give exact declarations, source locations, imports, and nine dependency pins. This guide preserves 67 exact Lean excerpts. Six constructed objects have a separate inherited audit of their proof fields. Those six objects are recorded separately from the catalog of 323 theorems. Five continuum physics conjectures remain open. Kernel execution evidence comes from the recorded formal core build for nineteen results. The guide identifies the source files, declaration locations, saved axiom records and printed excerpts. Finite computations are documented in the reproduction section. This reading guide does not claim a new kernel execution.

1 Reading a Lean theorem from Chapter VII

1.1 Three readings of one construction

The traditional reading begins with a finite state space, its operator algebra and a packet resolution. It derives matrix identities and explains what they mean for dynamics and recovery. The visual reading displays the same objects as corners, address sets, syndrome subspaces and local coordinates. The Lean reading asks a further question: what precise data must be supplied before each conclusion can be constructed?

An endpoint is a theorem whose conclusion expresses the desired result. A supporting lemma may isolate one algebraic or analytic step. A definition constructs an object but is not itself a proof that the object has every advertised property. The formalization deliberately uses several layers: general ring identities, complex matrix specializations, norm estimates, real calculus, spectral constructions and manifold calculus. Reading only a final theorem name loses the distinction between these layers.

The formal source distinguishes three constructions that should not be conflated. Orthogonal syndrome data permit direct recovery. General positive Gram data require spectral reduction and a necessity proof. The range-bundle construction supplies local charts and global manifold curvature. This guide follows the resulting declarations and makes their differing assumptions explicit.

1.2 Finite types, matrix types and type classes

Lean's `Matrix d m ℂ` means a function with a row index in `d`, a column index in `m`, and values in the complex numbers. The indices are types, not necessarily integers. A `Fintype d` instance gives the finite enumeration needed for a sum or dimension. A `DecidableEq d` instance lets a definition decide whether two indices agree. `Fin n` is the type of natural numbers smaller than `n`. It supplies a concrete index type without replacing an arbitrary finite type throughout the library.

The notation `Type*` leaves a universe level implicit. Braces in a binder, such as `{n : Type*}`, denote an implicit argument inferred from later data. Round parentheses denote an explicit argument. Square brackets denote an argument supplied through a type class. This is a structure or property that instance search should obtain. These brackets do not make an assumption disappear. A theorem with `[Nonempty n]` is only available when the state index has an inhabitant. A theorem with `[Invertible D]` has an actual invertibility requirement even though no ordinary parameter named `hD` is printed.

The declarations set `autoImplicit false`. Consequently an undeclared symbol is not silently promoted to a fresh theorem variable. A local `variable` command can still supply variables and type classes to subsequent declarations. An excerpt must therefore be read together with its surrounding section, not as an independent file. The chapter contracts explain those surrounding assumptions. The source file and line register allow the reader to check them directly.

Source notation	Mathematical reading
`Matrix d m ℂ`	This is a rectangular complex matrix. It can also represent the coefficient array of a bipartite state.
`n → ℂ`	This is a coordinate vector on a finite index type.
`→ₗ[ℂ]`	This map is linear over the complex numbers, with linearity stored in its structure.
`→L[ℝ]`	This continuous map is linear over the real numbers. It is used for derivatives and operators that vary with a parameter.
`≃ₗ[ℂ]`	This is a linear equivalence with an inverse and linearity laws.

Source notation	Mathematical reading
`≃ₜ`	This is a homeomorphism, including continuity in both directions.
`≃L[ℝ]`	This is a continuous equivalence that is linear over the real numbers.
`star`, `conjTranspose`	These denote the involution in an abstract star ring and the conjugate transpose for complex matrices.
`PosSemidef`, `PosDef`	These predicates describe positive semidefinite and positive definite matrices, respectively.
`IsStarProjection P`	This predicate requires both idempotence and self-adjointness of the projector.
`HasDerivAt f D t`	The actual real derivative of `f` at `t` is `D`.
`fderiv ℝ`, `mvfderiv I`	These are the Fréchet derivative and the derivative of a map from a manifold to a vector space.
`IsLeast S r`	This asserts that `r` belongs to `S` and is a lower bound for it.

1.3 Quantifiers and the information a structure stores

A family `P : I → R` is one element of the ring for each packet index. A dependent family `P : (i : F) → J i → Matrix (D i) (D i) ℂ` allows both the local dimension and the type of local packet addresses to vary with the tensor factor. This is more general than a tensor power of one fixed matrix.

The packet structure stores pairwise multiplication and completeness. Its multiplication law includes the diagonal case, so idempotence is a consequence. Self-adjointness is supplied separately when the construction is specialized to a star ring. This distinction lets circuit reconstruction work over any ring while the unitary identities require an adjoint.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 22 to 25. Excerpt `packet_structure`. Hash in Section 12.

```
structure PacketFamily (R I : Type*) [Ring R] [Fintype I] [DecidableEq I] where
  proj : I → R
  mul_proj : ∀ i j, proj i * proj j = if i = j then proj i else 0
  sum_proj : ∑ i, proj i = 1
```

Read `proj : I → R` as the object, `mul_proj` as its orthogonality law and `sum_proj` as its complete resolution of the identity. In a matrix realization the first law is $P_i P_j = \delta_{ij} P_i$. The second is $\sum_i P_i = 1$. No commutative multiplication assumption is present: matrix multiplication remains operator composition.

Structures used later have a different role. `OrthogonalSyndromes` stores an already normalized orthogonal error family, and `KnillLaflamme.Data` stores a general positive Gram matrix. `Data.diagonal` constructs diagonal error data. Its active positive directions are then normalized to construct `OrthogonalSyndromes`. Sufficiency assumes the existence of the general Gram data. Necessity constructs such data from a recovery. The final equivalence joins both directions. The correct reading keeps these roles separate rather than interpreting every structure field as a fact proved from scratch.

1.4 How to read a proof script

The scripts repeatedly use a small set of proof moves. `rw` replaces an expression using a stated equality. `simp only` performs a controlled reduction with an explicit list of lemmas. The broader `simp` tactic also uses the available simplification database. `ext i j` reduces a matrix equality to equality of each entry. `funext` proves equality of functions by comparing values. `constructor` starts the two obligations of a conjunction or equivalence. `rintro` decomposes quantified and existential data. `obtain` names an existential witness already proved to exist. `refine` fixes the overall shape of a term and leaves specified holes to prove.

The tactic `noncomm_ring` normalizes ring expressions without commuting factors. It is appropriate for identities involving $P_b U P_a$. Ordinary `ring` is used for scalar arithmetic or commutative coefficient expressions. A normalization step must therefore be read in the ring where it is used. It does not permit arbitrary matrix factors to be reordered.

A line beginning `have` proves or names a local intermediate assertion. The goal remains the same until that assertion is used. A `calc` block presents a chain of equalities or inequalities. Each link is a separate obligation. Induction on a list proves all finite circuit depths, including the empty list. A finite example verified by `norm_num` or `fin_cases` proves that concrete example, not a theorem in arbitrary dimension.

For a derivative identity, `HasDerivAt.unique` often supplies the final step: two proven derivatives of the same function at a point must coincide. The script first differentiates a product, then rewrites the product as a constant or as a known function, then invokes uniqueness. Thus the differentiated projector and unitary identities are derived from actual functions and their derivatives, rather than postulated as unrelated data of first order.

1.5 Scope is a part of the theorem

The finite results have broad mathematical scope, but their hypotheses also place clear limits on their use. A statement about the span of an algebra orbit permits every matrix in that algebra. Reachability under a physically specified set of unitary gates requires a separate argument. A theorem establishing complete positivity for finite channels quantifies over every finite reference index. Arbitrary infinite reference Hilbert spaces lie outside those declarations. A derivative computation under supplied smoothness does not prove the existence of a Schrödinger solution for every initial condition.

The modules for the projected connection do use arbitrary smooth real parameter manifolds and complete normed model spaces, which is broader than a single finite-dimensional chart. This generality of the mathematical parameter space is distinct from the paper's unproved continuum physics conjectures. Neither the word "manifold" nor an estimate in an abstract C-star algebra supplies a continuum tower, a physical constraint algebra, a renormalized limit or a Lorentzian emergence theorem.

Read each conclusion under its actual hypotheses and use its source locator to recover the surrounding assumptions. Source identity and excerpt fidelity are current documentation checks. Saved compiler and axiom transcripts are inherited proof evidence. This distinction continues through the replay chapter and full result map.

2 Navigation through the traditional paper

The Printed and PDF columns locate results in the attached Chapter VII source document. Physical PDF pages include its cover and opening material. These locations refer to that source document rather than the continuous folios of this book. A shared subject does not imply identical formal coverage.

2.1 All 19 finite items in the traditional core

Item	Number	Core statement	Printed	PDF
2.1	I.2.1	Orbit dimension of one quantum seed	5	13
2.2	I.2.2	Quantum seed-rank formula	5	13
2.3	I.2.3	Local orbit and Schmidt rank	6	14
2.4	I.2.4	One seed for the computational basis	6	14
3.1	I.3.1	Cyclic quantum projectors	7	15
3.2	I.3.2	Tensor/polyphase refinement	8	16
4.1	I.4.1	Exact circuit path expansion	9	17
4.2	I.4.2	Packetwise unitarity conservation	9	17
5.1	I.5.1	Exact symmetry packets	10	18
5.2	I.5.2	Commutator leakage bound	11	19
6.1	I.6.1	Moving-frame Hamiltonian	12	20
6.3	I.6.3	Canonical packet connection	13	21
6.4	I.6.4	Projected connection	14	22
7.1	I.7.1	Block-diagonal measurement invariance	17	25
7.2	I.7.2	Finite Kraus channel	17	25
8.1	I.8.1	Finite Knill–Laflamme criterion	19	27
9.1	I.9.1	Exact Feshbach resolvent	21	29
10.1	I.10.1	Finite Euclidean transfer	24	32
11.1	I.11.1	Low-mode second-order bound	26	34

2.2 Strengthened statements and their formal counterparts

This index includes all 79 named statements with proofs in strengthened Part II. A relation identifies shared ingredients or a shared finite result. It does not claim that the whole stronger statement has a dedicated Lean theorem. Entries without a corresponding named endpoint remain written mathematical statements. The formal ledger still contains 19 items, and the declaration inventory still contains 323 entries.

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.2.2 (36)	Orbit dimension of one seed	Item 2.1 gives the same orbit dimension result.	44
II.2.3 (36)	Quantum seed rank	Item 2.2 gives the same sharp minimum number of seeds.	44
II.2.4 (36)	Local orbit and Schmidt rank	Item 2.3 relates the local orbit to reduced support.	44
II.2.5 (37)	Computational address orbit	Item 2.4 constructs the basis by addressed local X gates.	45

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.3.1 (38)	Finite group isotypic projectors	The traditional statement has no corresponding named endpoint among the current 25 modules.	46
II.3.2 (38)	Representation matrix units	The traditional statement has no corresponding named endpoint among the current 25 modules.	46
II.3.3 (39)	Cyclic charge projectors	Item 3.1 formalizes the cyclic specialization.	47
II.3.5 (39)	Charge selection rule	The traditional statement has no corresponding named endpoint among the current 25 modules.	47
II.3.6 (39)	Charge addition	The traditional statement has no corresponding named endpoint among the current 25 modules.	47
II.3.7 (40)	Tensor charge refinement	Item 3.2 formalizes finite grouping of tensor packets.	48
II.3.8 (40)	Tensor selection rule	The traditional statement has no corresponding named endpoint among the current 25 modules.	48
II.4.1 (41)	Exact circuit path expansion	Item 4.1 covers every finite circuit depth.	49
II.4.2 (41)	Packet unitarity identities	Item 4.2 gives both operator unitarity identities.	49
II.4.3 (41)	Probability conservation from one packet	Item 4.2 gives the operator identity. The probability corollary is proved in the traditional text.	49
II.4.4 (42)	Path interference formula	The traditional statement has no corresponding named endpoint among the current 25 modules.	50
II.4.5 (42)	Charge pruning of circuit paths	The traditional statement has no corresponding named endpoint among the current 25 modules.	50
II.5.1 (43)	Exact invariant packets	Item 5.1 proves packet invariance for the exponential of a fixed generator.	51
II.5.2 (43)	Off diagonal norm	The traditional statement has no corresponding named endpoint among the current 25 modules.	51
II.5.3 (44)	Fixed packet leakage	Item 5.2 gives the commutator leakage bound. The norm equality for the off-diagonal corner is proved in the traditional text.	52
II.5.4 (44)	Leakage probability	The traditional statement has no corresponding named endpoint among the current 25 modules.	52
II.6.1 (45)	Canonical packet connection	Item 6.3 gives the canonical connection identities along a curve.	53
II.6.2 (45)	Exact Kato transport	The traditional statement has no corresponding named endpoint among the current 25 modules.	53
II.6.3 (46)	Moving frame Hamiltonian	Item 6.1 proves the differentiated equation in moving coordinates.	54
II.6.4 (46)	Moving packet leakage	The traditional statement has no corresponding named endpoint among the current 25 modules.	54

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.6.5 (47)	Direct packet rotation	The traditional statement has no corresponding named endpoint among the current 25 modules.	55
II.6.6 (47)	Convergence of short packet rotations	The traditional statement has no corresponding named endpoint among the current 25 modules.	55
II.6.7 (49)	Nonabelian packet connection	Item 6.4 gives projected manifold curvature and gauge gluing. The local formulas in orthonormal frames are proved in the traditional text.	57
II.6.8 (49)	Bloch sphere curvature	The traditional statement has no corresponding named endpoint among the current 25 modules.	57
II.7.2 (50)	Complete operator packet system	The traditional statement has no corresponding named endpoint among the current 25 modules.	58
II.7.4 (50)	Dephasing as conditional expectation	The traditional statement has no corresponding named endpoint among the current 25 modules.	58
II.7.5 (51)	Block diagonal measurement invariance	Item 7.1 proves trace invariance for block diagonal observables.	59
II.7.6 (52)	Exact channel block calculus	The traditional statement has no corresponding named endpoint among the current 25 modules.	60
II.7.8 (53)	Coherence order projectors	The traditional statement has no corresponding named endpoint among the current 25 modules.	61
II.7.9 (53)	Covariant channels preserve coherence order	The traditional statement has no corresponding named endpoint among the current 25 modules.	61
II.7.10 (53)	Choi symmetry criterion	The traditional statement has no corresponding named endpoint among the current 25 modules.	61
II.7.11 (54)	Choi packet decomposition	The traditional statement has no corresponding named endpoint among the current 25 modules.	62
II.7.14 (55)	Charge preservation by a Lindblad generator	The traditional statement has no corresponding named endpoint among the current 25 modules.	63
II.7.15 (55)	Independent coherence evolution	The traditional statement has no corresponding named endpoint among the current 25 modules.	63
II.8.2 (56)	Knill Laflamme as obstruction vanishing	Item 8.1 gives the criterion for unnormalized error families and recovery on their span. The traditional text also uses a quotient formulation.	64
II.8.3 (57)	Orthogonal syndrome construction	Item 8.1 reduces general positive Gram data by spectral diagonalization.	65
II.8.4 (57)	Exact recovery on the error span	Item 8.1 uses one total recovery for normalized channels in the error span.	65

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.8.5 (58)	Syndrome measurement preserves logical information	The identities for syndrome cross errors support item 8.1. The traditional text supplies the measurement interpretation.	66
II.8.7 (58)	Meaning of the approximate defect	The traditional statement has no corresponding named endpoint among the current 25 modules.	66
II.9.2 (60)	Exact projected resolvent	Item 9.1 gives the projected resolvent with both required invertibility hypotheses.	68
II.9.3 (60)	Spectral equivalence	Item 9.1 relates invertibility of the full matrix and its Schur complement. The traditional text adds spectral and eigenvector formulations.	68
II.9.4 (61)	Upper half plane sign	The traditional statement has no corresponding named endpoint among the current 25 modules.	69
II.9.5 (61)	Uniform self energy bounds	The traditional statement has no corresponding named endpoint among the current 25 modules.	69
II.9.6 (62)	Low energy expansion	The traditional statement has no corresponding named endpoint among the current 25 modules.	70
II.9.7 (62)	Bare truncation error identity	The traditional statement has no corresponding named endpoint among the current 25 modules.	70
II.9.8 (63)	Exact retained memory equation	The traditional statement has no corresponding named endpoint among the current 25 modules.	71
II.9.9 (63)	Positive definite quantum memory	The traditional statement has no corresponding named endpoint among the current 25 modules.	71
II.9.10 (63)	Finite oscillatory memory expansion	The traditional statement has no corresponding named endpoint among the current 25 modules.	71
II.9.11 (64)	Exact local closure criterion	The traditional statement has no corresponding named endpoint among the current 25 modules.	72
II.9.12 (65)	Stability of the self energy	The traditional statement has no corresponding named endpoint among the current 25 modules.	73
II.9.13 (65)	Uniform Feshbach control	The traditional statement has no corresponding named endpoint among the current 25 modules.	73
II.10.1 (66)	Finite Euclidean transfer	Item 10.1 constructs the shifted exponential and proves its positivity and spectral bounds under the stated hypotheses.	74
II.10.2 (66)	Hamiltonian reconstruction from transfer	The traditional statement has no corresponding named endpoint among the current 25 modules.	74
II.10.3 (66)	Finite reflection positivity	The traditional statement has no corresponding named endpoint among the current 25 modules.	74

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.10.4 (67)	Reflection positive correlators	The traditional statement has no corresponding named endpoint among the current 25 modules.	75
II.10.5 (67)	Ground packet convergence	The traditional statement has no corresponding named endpoint among the current 25 modules.	75
II.10.6 (67)	Symmetry resolved transfer	The traditional statement has no corresponding named endpoint among the current 25 modules.	75
II.11.2 (68)	Finite scale limits	The traditional statement has no corresponding named endpoint among the current 25 modules.	76
II.11.3 (68)	Heat packet decomposition	The traditional statement has no corresponding named endpoint among the current 25 modules.	76
II.11.4 (69)	Low mode dispersion in dimension d	Item 11.1 gives the scalar bound in one dimension. The traditional text proves the summed result in d dimensions.	77
II.11.5 (69)	Fixed time heat trace convergence	The traditional statement has no corresponding named endpoint among the current 25 modules.	77
II.11.7 (70)	Continuum torus small time dimension	The traditional statement has no corresponding named endpoint among the current 25 modules.	78
II.11.8 (70)	Rigorous intermediate dimension window	The traditional statement has no corresponding named endpoint among the current 25 modules.	78
II.12.2 (71)	Finite constraint closure criterion	The traditional statement has no corresponding named endpoint among the current 25 modules.	79
II.12.3 (72)	Physical compression is not a closure test	ConstraintAudit formalizes finite compression to the common kernel.	80
II.12.4 (72)	Invariance of the physical space	The traditional statement has no corresponding named endpoint among the current 25 modules.	80
II.12.5 (73)	Exact constraint transport	The traditional statement has no corresponding named endpoint among the current 25 modules.	81
II.12.6 (73)	Approximate constraint transport	The traditional statement has no corresponding named endpoint among the current 25 modules.	81
II.13.2 (75)	Bounded Hamiltonian direct limit	The traditional statement has no corresponding named endpoint among the current 25 modules.	83
II.13.3 (75)	Exact packet direct limit	The traditional statement has no corresponding named endpoint among the current 25 modules.	83
II.13.4 (76)	Norm summable Hamiltonian tower	The traditional statement has no corresponding named endpoint among the current 25 modules.	84

Number (page)	Strengthened statement	Relation to the current formal source	PDF
II.13.5 (76)	Limit of packet leakage	The traditional statement has no corresponding named endpoint among the current 25 modules.	84
II.13.6 (77)	Reflection positivity is closed under limits	The traditional statement has no corresponding named endpoint among the current 25 modules.	85
II.13.7 (77)	Constraint closure under norm convergence	The traditional statement has no corresponding named endpoint among the current 25 modules.	85
II.13.8 (77)	Heat trace convergence from trace norm	The traditional statement has no corresponding named endpoint among the current 25 modules.	85

2.3 Retained supplements explain the constructed proofs

Part III presents the mathematical content of the six formal supplements. The endpoint map identifies the precise formal coverage of each result and its source. A broader construction is read with its own hypotheses rather than inferred from a restricted special case.

Sections	Retained proof material	Printed	PDF
III.1–III.5	This supplement develops packet algebra, symmetry and measurements. It states the corrected resolvent and spacing hypotheses and audits finite constraints.	81–85	89–93
III.6–III.10	This supplement constructs an addressed computational seed, tensor refinements, moving coordinates, the canonical connection and Kraus amplification.	86–90	94–98
III.11–III.14	This supplement derives orbit spaces and sharp seed counts. It defines Schmidt rank independently and develops cyclic characters and the Euclidean exponential.	91–94	99–102
III.15–III.17	This supplement proves leakage for the exponential, projected curvature in a chart and total recovery from orthogonal syndromes.	95–97	103–105
III.18–III.23	This supplement proves spectral reduction for general Gram data, necessity from recovery, the bridge from states to operators, and correction with finite references and error spans.	98–104	106–112
III.24–III.27	This supplement constructs local products for the range, smooth transitions and the global connection. It proves the gauge law and manifold curvature formula.	104–106	112–114

3 Seeds, local orbits, and the packet resolution

3.1 One seed: identify the actual image before counting it

The starting coefficient array is $C \in \mathrm{Mat}_{d\times m}(\mathbb{C})$. Left multiplication by an arbitrary $d \times d$ matrix changes each row independently within the row space of C. This observation gives the orbit dimension formula. Lean first defines the row space as the range of the linear map given by right multiplication. It then defines the orbit as the range of that map acting on all d rows. The dimension is derived from this construction.

Contract. The row and multiplicity index types `d` and `m` have finite enumerations. The orbit under the algebra action is $\{AC : A \in \mathrm{Mat}_{d\times d}(\mathbb{C})\}$, which is already a linear subspace. The dimension formula permits an empty row type. The equivalence characterizing a full orbit adds `[Nonempty d]` because it cancels the positive row cardinality.

The lemma `mem_orbit_iff_rows` supplies both directions. If $X = AC$, each row of X is a linear combination of rows of C. Conversely, choose one coefficient row for each row of X, collect them into a matrix A, and obtain $AC = X$. The latter choice is an existence argument using classical choice. It is not a numerical algorithm for choosing coefficients.

The explicit linear equivalence below expresses this correspondence for each row. Its domain is the subspace of reachable coefficient arrays. Its codomain is the function space whose value at each row is a member of the row space. Both inverse laws reduce to the underlying functions.

Source: `formalization/SeedOrbit.lean`.

Exact lines 45 to 52. Excerpt `row_equiv`. Hash in Section 12.

```
noncomputable def orbitRowEquiv (C : Matrix d m ℂ) :
    orbitSpace C ≃ₗ[ℂ] (d → rowSpace C) where
  toFun X i := ⟨X.val i, (mem_orbit_iff_rows C X.val).mp X.property i⟩
  invFun f := ⟨fun i => (f i).val, (mem_orbit_iff_rows C _).mpr (fun i => (f i).property)⟩
  left_inv _ := rfl
  right_inv _ := rfl
  map_add' _ _ := rfl
  map_smul' _ _ := rfl
```

Proof reading. The `toFun` field extracts the rows and attaches the proof that each row belongs to the row space. The `invFun` field assembles those rows, using the reverse membership implication. The equalities of functions are definitional, so the inverse and linearity proofs are `rfl`. Taking dimensions of this equivalence gives $|d|$ rank C. The extra `orbit_span` lemma identifies the subspace with the span of the literal action set.

Source: `formalization/SeedOrbit.lean`.

Exact lines 68 to 71. Excerpt `orbit_dimension`. Hash in Section 12.

```
theorem span_orbit_dimension (C : Matrix d m ℂ) :
    finrank ℂ (Submodule.span ℂ (Set.range fun A : Matrix d d ℂ => A * C)) =
      Fintype.card d * C.rank := by
  rw [orbit_span, orbit_dimension]
```

Source: formalization/SeedOrbit.lean.

Exact lines 73 to 80. Excerpt cyclic_rank. Hash in Section 12.

```
theorem cyclic_iff_full_rank [Nonempty d] (C : Matrix d m ℂ) :
    orbitSpace C = ⊤ ↔ C.rank = Fintype.card m := by
  rw [Submodule.eq_top_iff_finrank_eq, orbit_dimension]
  have hdim : finrank ℂ (Matrix d m ℂ) = Fintype.card d * Fintype.card m := by
    change finrank ℂ (d → m → ℂ) = _
    simp [Module.finrank_pi_fintype]
  rw [hdim]
  exact Nat.mul_left_cancel_iff (Fintype.card_pos)
```

The nonempty condition deserves attention. If there are no row indices, the ambient coefficient space is zero and every orbit is already full, regardless of the multiplicity index. Cancelling $|d|$ would be invalid. Lean's final cancellation explicitly uses `Fintype.card_pos`. This records why the test for full rank needs an inhabitant. Multiplicity zero is permitted and gives a target of dimension zero.

3.2 Several seeds: stack them and prove a minimum

Continue with a nonempty row index type. For k seeds C_a, the allowed generation is

$$X = \sum_{a \in k} A_a C_a,$$

where the algebra element A_a may differ for each seed. The question is whether this family generates the whole space linearly. The implementation records its universal quantifier over X and its existential family of coefficient matrices.

Source: formalization/SeedOrbit.lean.

Exact lines 127 to 128. Excerpt seed_generate. Hash in Section 12.

```
def Generates (C : k → Matrix d m ℂ) : Prop :=
  ∀ X : Matrix d m ℂ, ∃ A : k → Matrix d d ℂ, ∑ a, A a * C a = X
```

Source: formalization/SeedOrbit.lean.

Exact lines 148 to 150. Excerpt stack_rank. Hash in Section 12.

```
theorem generates_iff_rank [Nonempty d] (C : k → Matrix d m ℂ) :
    Generates C ↔ (stackSeeds C).rank = Fintype.card m := by
  rw [generates_iff_rowOrbit, rowOrbit_full]
```

Stacking the seeds gives one matrix with row index $k \times d$. Its row space is the span of all the seed rows. The lemma `stacked_action` proves entrywise that multiplication by the large coefficient matrix is exactly the displayed sum. The rank test therefore says that the stacked seed matrix has full column rank. A rank upper bound gives the necessary inequality $|m| \leq |k||d|$.

The reverse direction is constructive at the matrix level. When the number of columns does not exceed the available row slots, choose an injection of column indices into row indices and place a 1 at each corresponding coordinate, with zeros elsewhere. The associated multiplication map is injective because its selected output coordinates reproduce the original input coordinates. The rank theorem follows from that injectivity.

Source: formalization/SeedOrbit.lean.

Exact lines 158 to 171. Excerpt full_column. Hash in Section 12.

```
theorem exists_full_column_rank (h : Fintype.card m ≤ Fintype.card d) :
    ∃ C : Matrix d m ℂ, C.rank = Fintype.card m := by
  classical
  obtain ⟨f⟩ := Function.Embedding.nonempty_of_card_le h
  let C : Matrix d m ℂ := fun i j => if i=f j then 1 else 0
  have hC (x : m → ℂ) (j : m) : (C *ᵥ x) (f j) = x j := by
    simp [C, Matrix.mulVec, dotProduct, f.injective.eq_iff]
  have hi : Function.Injective C.mulVecLin := by
    intro x y hxy
    funext j
    have h := congrFun hxy (f j)
    simpa only [Matrix.mulVecLin_apply, hC] using h
  refine ⟨C, ?_⟩
  rw [Matrix.rank, LinearMap.finrank_range_of_inj hi, Module.finrank_pi]
```

For a finite family of direct summands, dd mm : I → ℕ gives the matrix row dimension and multiplicity of each block. A seed contains one coefficient array in every block. BlockGenerates quantifies over simultaneous block targets and permits simultaneous block algebra elements. The equivalence block_generates_iff proves that this is exactly the collection of individual generation tests. It does not replace the simultaneous problem by an unproved heuristic.

Contract. Every row dimension dd i is positive. Multiplicities mm i may vanish. The block index I is any finite type, including the empty type. The formula is the finite supremum of ceiling divisions in the natural numbers, so an empty block family gives zero.

Define

$$r = \max_{i \in I} \left\lceil \frac{m_i}{d_i} \right\rceil .$$

The source writes this as Finset.univ.sup with natural ceiling division. The lemma seed_formula_le_iff translates $r \leq s$ into all inequalities $m_i \leq s d_i$. The existence equivalence constructs seeds exactly when those inequalities hold. The final result is IsLeast: it proves both that r seeds exist and that every successful seed count is at least r.

Source: formalization/SeedOrbit.lean.

Exact lines 239 to 246. Excerpt seed_minimum. Hash in Section 12.

```
theorem quantum_seed_rank (dd mm : I → ℕ) (hd : ∀ i, 0 < dd i) :
    IsLeast {s : ℕ | ∃ C : Fin s → (i : I) → Matrix (Fin (dd i)) (Fin (mm i)) ℂ,
      BlockGenerates dd mm C} (seedRankFormula dd mm) := by
  constructor
  · exact (exists_block_seeds_iff dd mm hd _).mpr
      ((seed_formula_le_iff dd mm hd _).mp le_rfl)
  · intro s hs
    exact (seed_formula_le_iff dd mm hd s).mpr ((exists_block_seeds_iff dd mm hd s).mp hs)
```

Proof reading. The first half of the pair uses le_rfl to show the formula satisfies its own inequalities, then applies the proved existence criterion. The second half takes any witness at count s, converts its existence to the inequalities, then converts those inequalities back to $r \leq s$. Thus minimality and attainability are both present. The conclusion is stronger than merely displaying a candidate formula.

3.3 Schmidt rank is independently defined

For a bipartite vector $\psi : d \times m \to \mathbb{C}$, `coefficientEquiv` identifies $\psi(i,j)$ with the coefficient matrix C_{ij}. The local action $A \otimes 1$ becomes AC in those standard product coordinates. This identity is proved by expanding the Kronecker product and summing over the product index.

Source: `formalization/SeedOrbit.lean`.

Exact lines 261 to 265. Excerpt `local_action`. Hash in Section 12.

```
theorem local_tensor_action [DecidableEq m] (A : Matrix d d ℂ) (psi : d × m → ℂ) :
    coefficientEquiv ((A ⊗ₖ (1 : Matrix m m ℂ)) *ᵥ psi) = A * coefficientEquiv psi := by
  ext i j
  simp [coefficientEquiv, Matrix.mulVec, dotProduct,
    Fintype.sum_prod_type, Matrix.one_apply, Matrix.mul_apply]
```

The source defines Schmidt rank through product decompositions before proving that it agrees with matrix rank. `HasProductDecomposition C s` means an actual representation

$$C_{ij} = \sum_{a<s} u_a(i) v_a(j).$$

Such a representation factors C through an s-dimensional coordinate space, so rank $C \le s$. Conversely, choose a basis of the row space and express each row in that basis. The basis vectors of the row space and their coordinate functions give a product decomposition of length exactly rank C. These two directions prove that matrix rank is the least possible product length.

Source: `formalization/SeedOrbit.lean`.

Exact lines 298 to 300. Excerpt `schmidt_minimum`. Hash in Section 12.

```
theorem product_length_minimum (C : Matrix d m ℂ) :
    IsLeast {s : ℕ | HasProductDecomposition C s} C.rank := by
  exact ⟨product_decomposition_at_rank C, fun _ h => product_decomposition_rank_bound C h⟩
```

Source: `formalization/SeedOrbit.lean`.

Exact lines 303 to 306. Excerpt `schmidt_definition`. Hash in Section 12.

```
noncomputable def schmidtRank (psi : d × m → ℂ) : ℕ := by
  classical
  exact Nat.find (show ∃ s, HasProductDecomposition (coefficientEquiv psi) s from
    ⟨_, product_decomposition_at_rank _⟩)
```

`Nat.find` takes the least natural number for which a product decomposition exists. The existence needed by `Nat.find` has already been proved at the rank. The equality `schmidt_rank_eq_matrix_rank` then uses the upper witness and the rank lower bound in opposite directions. It is an equality between independently introduced objects.

The reduced B matrix is also constructed explicitly:

$$(\rho_B)_{j\ell} = \sum_i \psi(i,j)\overline{\psi(i,\ell)}.$$

In the chosen convention it is $C^T(C^T)^*$. Its positivity follows from this Gram representation. Its rank agrees with the matrix rank of C, hence with Schmidt rank. Full support is expressed as surjectivity of its multiplication linear map. Combining this with the local orbit dimension gives the endpoint below.

Source: `formalization/SeedOrbit.lean`.

Exact lines 364 to 366. Excerpt `reduced_support`. Hash in Section 12.

```
theorem local_orbit_reduced_support [Nonempty d] (psi : d × m → ℂ) :
    localOrbit psi = ⊤ ↔ LinearMap.range (reducedB psi).mulVecLin = ⊤ := by
  rw [local_orbit_full_iff, reduced_full_support_iff]
```

Contract. The equivalence between a full local orbit and full reduced support assumes `[Nonempty d]`. The dimension equality itself includes zero vectors and empty index types. No normalization of ψ is required for these algebraic rank statements. The action uses every matrix A, so the result concerns the linear span under the local operator algebra. It does not assert reachability under an arbitrary restricted family of unitary controls.

3.4 Addressed bit flips give a computational basis

`Address n` is defined as `Fin n → ZMod 2`, the finite set of bit strings. `State n` is the complex vector space of functions on that address set. The seed is the coordinate vector at the zero address. The linear operator `translation a` evaluates a vector at $x - a$. On a ket it gives $|b\rangle \mapsto |a + b\rangle$.

Each `localX j` translates by the bit vector supported at coordinate j. The operator named `controlledX a j` in this module is selected by the classical address bit a_j: it is identity when that bit is zero, and the local X otherwise. This definition selects a gate according to the address. Despite its name, it does not separately formalize the controlled-NOT gate on two qubits.

The address circuit is the product of this finite gate list. Translation products add their addresses, and the sum of the local address contributions equals a. Therefore the circuit translates by a and sends the zero seed to the coordinate basis vector at a.

Source: `formalization/ComputationalSeed.lean`.

Exact lines 72 to 77. Excerpt `seed_translation`. Hash in Section 12.

```
theorem circuit_eq_translation {n : ℕ} (a : Address n) :
    addressCircuit a = translation a := by
  unfold addressCircuit
  simp_rw [controlledX_eq_translation]
  simpa only [List.map_ofFn, Function.comp_def, List.sum_ofFn, sum_local_addresses] using
    translation_list_product (List.ofFn (fun j : Fin n => Pi.single j (a j)))
```

Source: `formalization/ComputationalSeed.lean`.

Exact lines 80 to 82. Excerpt `seed_basis`. Hash in Section 12.

```
theorem circuit_seed_basis {n : ℕ} (a : Address n) :
    addressCircuit a (seed n) = Pi.basisFun ℂ (Address n) a := by
  rw [circuit_seed, Pi.basisFun_apply, ket]
```

The basis theorem then supplies linear independence and full span. The cardinality theorem gives 2^n. When $n = 0$, the type of bit strings has exactly one empty address, the gate list is empty, and its product is identity. The source covers this case without assuming that the number of qubits is positive.

3.5 Cyclic projectors from powers of an element of finite order

Let $N > 0$ and $S^N = 1$. The source uses `ZMod N` as the residue type and the standard primitive additive character χ. It defines

$$P_j = \frac{1}{N} \sum_{r \in \mathbb{Z}/N} \chi(-rj) S^{r.\mathrm{val}}.$$

The representation `cyclicPowers S` is constructed from the actual powers of S. The proof of its addition law uses $S^N = 1$ to reduce an exponent modulo N. The root of unity is identified with $\exp(2\pi i/N)$, and the coefficient lemma recovers the familiar inverse power ω^{-rj}.

Source: `formalization/CyclicProjectors.lean`.

Exact lines 22 to 26. Excerpt `cyclic_character`. Hash in Section 12.

```
theorem character_sum (t : ZMod N) :
    ∑ r : ZMod N, stdAddChar (t*r) = if t=0 then (N : ℂ) else 0 := by
  split_ifs with h
  · simp [h]
  · exact sum_eq_zero_of_ne_one (isPrimitive_stdAddChar N h)
```

The character sum provides the cancellation needed in the proof. When its argument is zero, the sum is N. When the argument is nonzero, primitivity makes the sum vanish. To prove multiplication of packets, first show that left multiplication by the representation acts on a packet by its character. Substitute that eigenaction into the packet product and collapse the remaining character sum.

Source: `formalization/CyclicProjectors.lean`.

Exact lines 44 to 58. Excerpt `cyclic_mul`. Hash in Section 12.

```
theorem packet_mul (rho : ZMod N → A)
    (hadd : ∀ a b, rho (a+b) = rho a * rho b) (j k : ZMod N) :
    packet rho j * packet rho k = if j=k then packet rho j else 0 := by
  conv_lhs => lhs; unfold packet
  rw [Algebra.smul_mul_assoc, Finset.sum_mul]
  simp only [Algebra.smul_mul_assoc, packet_eigen rho hadd, smul_smul,
    ← map_add_eq_mul]
  have hf (r : ZMod N) : -(r*j) + r*k = (k-j)*r := by ring
  simp only [hf]
  rw [← Finset.sum_smul, character_sum]
  by_cases h : j=k
  · subst k
    simp [NeZero.ne N]
  · have hk : k-j ≠ 0 := sub_ne_zero.mpr (Ne.symm h)
    simp [hk, h]
```

Completeness follows from exchanging the two finite sums and applying the same orthogonality in the other index. Adjointness requires a separate argument in a star ring: conjugate the character, replace r by $-r$, and use the adjoint law for unitary powers. An element of finite order gives the idempotent decomposition. Unitarity supplies the Hermitian condition in the theorem for complex matrices.

Source: formalization/CyclicProjectors.lean.

Exact lines 157 to 170. Excerpt cyclic_matrix. Hash in Section 12.

```
theorem finite_cyclic_projectors {n : Type*} [Fintype n] [DecidableEq n]
    (S : Matrix n n ℂ) (hS : S^N=1) (hU : S ∈ unitary (Matrix n n ℂ)) :
    (∀ j k : ZMod N, packet (cyclicPowers S) j * packet (cyclicPowers S) k =
      if j=k then packet (cyclicPowers S) j else 0) ∧
    (∑ j : ZMod N, packet (cyclicPowers S) j) = 1 ∧
    (∀ j : ZMod N, S * packet (cyclicPowers S) j =
      omega (N := N)^j.val • packet (cyclicPowers S) j) ∧
    ∀ j : ZMod N, (packet (cyclicPowers S) j).IsHermitian := by
  obtain ⟨hp, hs, he⟩ := cyclic_quantum_projectors S hS
  refine ⟨hp, hs, ?_, ?_⟩
  · intro j
    simpa only [character_as_power] using he j
  · intro j
    exact cyclic_packet_selfAdjoint S hS hU j
```

Contract. The ambient general construction is a complex algebra over a ring. The concrete endpoint has a finite matrix state index and decidable equality. `[NeZero N]` excludes zero modulus, includes $N = 1$, and justifies division by N. The theorem does not require every packet to have nonzero rank.

3.6 Tensor factors and regrouping by residue

The tensor implementation uses a finite factor index `F`, with a potentially different state index `D i` in every factor. A tensor matrix is defined by the product of its local entries. Multiplication expands into a sum over dependent tuples of intermediate indices. The finite identity expressing a product of sums converts that sum back into the product of local matrix multiplications.

Source: formalization/TensorRefinement.lean.

Exact lines 20 to 27. Excerpt tensor_mul. Hash in Section 12.

```
theorem tensor_mul (A B : (i : F) → Matrix (D i) (D i) ℂ) :
    tensor (fun i => A i * B i) = tensor A * tensor B := by
  ext x y
  simp only [tensor, Matrix.mul_apply]
  rw [Fintype.prod_sum]
  apply Finset.sum_congr rfl
  intro z _
  exact Finset.prod_mul_distrib
```

The product formula, identity formula, adjoint formula and sum formula then prove that the fine packets indexed by tuples are mutually orthogonal, complete and Hermitian. The coarse construction sums fine packets with the same address. The general proof over a ring shows that grouping an orthogonal resolution by any finite address map remains an orthogonal resolution. For polyphase grouping, the address is the sum of the factor residues in `ZMod N`.

Source: `formalization/TensorRefinement.lean`.

Exact lines 139 to 154. Excerpt `tensor_refinement`. Hash in Section 12.

```
theorem tensor_polyphase_refinement {F : Type*} [Fintype F] [DecidableEq F] {D : F → Type*}
    [∀ i, Fintype (D i)] [∀ i, DecidableEq (D i)]
    (N : ℕ) [NeZero N] (P : (i : F) → ZMod N → Matrix (D i) (D i) ℂ)
    (hmul : ∀ i j k, P i j * P i k = if j = k then P i j else 0)
    (hsum : ∀ i, ∑ j, P i j = 1)
    (hstar : ∀ i j, (P i j).conjTranspose = P i j) :
    let Q := fun a : F → ZMod N => tensor (fun i => P i (a i))
    let C := coarse Q (fun a => ∑ i, a i)
    (∀ r s, C r * C s = if r = s then C r else 0) ∧
      (∑ r, C r) = 1 ∧ (∀ r, (C r).conjTranspose = C r) := by
  classical
  dsimp only
  refine ⟨?_, ?_, ?_⟩
  · exact coarse_mul _ _ (fine_mul P hmul)
  · exact coarse_complete _ _ (fine_complete P hsum)
  · exact coarse_selfAdjoint _ _ (fine_selfAdjoint P hstar)
```

Proof reading. The two `let` bindings expose the fine family Q_a and the coarse family C_r inside the statement. After `dsimp only`, each conclusion is a direct application of the generic coarse theorem to a theorem already proved for the fine family. The code therefore separates tensor construction from finite grouping, which makes the proof reusable.

The input here is a packet resolution in each tensor factor. The preceding theorem can construct these local packets from each factor's cyclic symmetry. The tensor theorem takes the resulting local resolutions as input. Together the two constructions implement the traditional passage from cyclic local packets to a tensor refinement and packets grouped by total residue.

4 Circuit corners, conserved packets, and leakage

4.1 The order of the indices is part of the definition

For a packet resolution P, the corner of an operator is

$$U_{ba} = P_b U P_a.$$

The target index b comes first, and the source index a comes second. Matrix factors act from the right on a vector. The gate list in `QuantumPacketCore` is written latest to earliest, $[U_m, \dots, U_1]$, so its product is $U_m \cdots U_1$. The source comments and definitions make both conventions explicit. They determine the orientation of every path.

Contract. The basic corner identities require an arbitrary ring `R` and a finite packet index with decidable equality. They require the packet multiplication and completeness laws, but neither an adjoint nor a norm. The coefficients are generally noncommutative.

Inserting $\sum_c P_c = 1$ between two operators gives

$$(UV)_{ba} = \sum_c U_{bc} V_{ca}.$$

The code first rewrites an individual product of corners. The adjacent $P_c P_c$ collapses by idempotence, while all other factors keep their order. Only then does it combine the finite sum and use completeness.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 41 to 50. Excerpt `compose_blocks`. Hash in Section 12.

```
theorem compose_blocks (U V : R) (b a : I) :
    P.block (U * V) b a = ∑ c, P.block U b c * P.block V c a := by
  have term (c : I) : P.block U b c * P.block V c a =
      P.proj b * U * P.proj c * V * P.proj a := by
    unfold block
    calc
      _ = P.proj b * U * (P.proj c * P.proj c) * V * P.proj a := by noncomm_ring
      _ = _ := by rw [P.idempotent]
  simp_rw [term]
  simp only [← Finset.sum_mul, ← Finset.mul_sum, P.sum_proj, block, mul_assoc, one_mul]
```

This is the local mechanism for the whole path expansion. `pathSum` is an explicit recursively nested finite sum. For no gates, its value is $P_b P_a$, preserving the boundary packet. For a gate U followed by a shorter circuit, it sums U_{bc} times the shorter path from a to c.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 63 to 68. Excerpt `circuit_path`. Hash in Section 12.

```
theorem circuit_path_expansion (Us : List R) (b a : I) :
    P.block (circuit Us) b a = P.pathSum Us b a := by
  induction Us generalizing b a with
  | nil => simp [circuit, pathSum, block]
  | cons U Us ih =>
      simp only [circuit, pathSum, P.compose_blocks, ih]
```

Proof reading. The induction generalizes both boundary indices, which makes the induction hypothesis available at every intermediate c. The empty case reduces the identity corner. In the nonempty case, the lemma for composing corners creates the finite sum required by the recursive definition. The induction hypothesis then replaces each shorter corner. The resulting proof covers every finite circuit depth.

Summing the resulting path values over source and target reconstructs the complete circuit operator. It retains coherent complex amplitudes and operator products. No step converts the path sum into a stochastic transition law or assumes that different paths are mutually incoherent.

4.2 The two unitarity identities use different hypotheses

To take adjoints, supply the structure of a star ring and self-adjoint packet elements. The adjoint swaps source and target:

$$U^*_{ba} = (U^*)_{ab}.$$

For the column identity, the source needs only $U^*U = 1$:

$$\sum_b U^*_{ba} U_{bc} = \delta_{ac} P_a.$$

For the row identity, it needs $UU^* = 1$:

$$\sum_a U_{ba} U^*_{da} = \delta_{bd} P_b.$$

The source states these separately rather than hiding both inside a single unitary predicate. The distinction matters in abstract star rings where a one-sided inverse need not be two-sided.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 119 to 131. Excerpt `unitarity_columns`. Hash in Section 12.

```
theorem unitarity_columns (U : R) (hU : star U * U = 1) (a c : I) :
    (∑ b, star (P.block U b a) * P.block U b c) =
      if a = c then P.proj a else 0 := by
  have term (b : I) : star (P.block U b a) * P.block U b c =
      P.proj a * star U * P.proj b * U * P.proj c := by
    rw [P.adjoint_block hstar]
    unfold block
    calc
      _ = P.proj a * star U * (P.proj b * P.proj b) * U * P.proj c := by noncomm_ring
      _ = _ := by rw [P.idempotent]
  simp_rw [term]
  rw [← Finset.sum_mul, ← Finset.sum_mul, ← Finset.mul_sum, P.sum_proj, mul_one]
  rw [mul_assoc (P.proj a), hU, mul_one, P.mul_proj]
```

Proof reading. The local term is reorganized to $P_a U^* P_b U P_c$, using self-adjointness and $P_b^2 = P_b$. Summing over b replaces the middle resolution by the identity. The given equation $U^*U = 1$ removes the operator factors. The final packet multiplication gives the conditional expression in the conclusion. The concrete matrix wrappers merely specialize this already proved identity.

For an actual state, this identity can be combined with the inner product on the Hilbert space to interpret conservation of total packet probability. The formal statement displayed here is the operator

identity. A chosen vector, state normalization and the corresponding interpretation through the inner product are additional data in such an application.

4.3 Exact commutation propagates to the actual exponential

If H commutes with each P_a, its corners between distinct packets vanish and the diagonal corners reconstruct H. The dynamics layer uses a normed complex algebra and the exponential defined by its power series. A `Commute` proof for H and P_a propagates through complex scalar multiplication and the exponential. Thus e^{zH} is diagonal in the same packet resolution for every complex scalar z.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 168 to 171. Excerpt `symmetry_exp`. Hash in Section 12.

```
theorem symmetry_exponential_no_transfer (H : R)
    (hH : ∀ i, H * P.proj i = P.proj i * H) (z : ℂ)
    (b a : I) (hba : b ≠ a) : P.block (NormedSpace.exp (z • H)) b a = 0 :=
  P.symmetry_no_transfer _ (P.symmetry_exponential_commutes H hH z) b a hba
```

For evolution in real time, set $z = -it$. The conclusion is exact zero transfer between distinct packets. The commutation hypothesis is still required. It can be proved from a chosen symmetry or supplied as data. A packet resolution alone does not imply it.

4.4 Leakage begins with a differentiated conjugation

The quantitative result works first in an abstract complex C-star algebra. Its matrix specialization activates the operator norm on the Hilbert space through `Matrix.Norms.L2Operator`. This scope is essential: the displayed bound is not an arbitrary entrywise matrix norm estimate.

Let $G^* = -G$ and set $V(t) = e^{tG}Pe^{-tG}$. The source constructs this function and differentiates the actual exponential product:

$$V'(t) = e^{tG}(GP - PG)e^{-tG}.$$

It proves that the exponentials are unitary. Left and right multiplication by unitary elements preserve the C-star norm, so $\|V'(t)\| = \|GP - PG\|$.

Source: `formalization/LeakageBound.lean`.

Exact lines 31 to 38. Excerpt `transport_derivative`. Hash in Section 12.

```
theorem transported_derivative (G P : A) (t : ℝ) :
    HasDerivAt (transported G P)
      (exp (t • G) * (G * P - P * G) * exp (t • (-G))) t := by
  have hd := ((hasDerivAt_exp_smul_const G t).mul_const P).mul
    (hasDerivAt_exp_smul_const' (-G) t)
  convert hd using 1
  · rfl
  · noncomm_ring
```

Source: formalization/LeakageBound.lean.

Exact lines 46 to 54. Excerpt transport_lipschitz. Hash in Section 12.

```
theorem transported_lipschitz (G P : A) (hG : star G = -G) (t s : ℝ) :
    ‖transported G P t - transported G P s‖ ≤
      |t-s| * ‖G*P-P*G‖ := by
  have hh := Convex.norm_image_sub_le_of_norm_deriv_le
    (f := transported G P) (s := Set.univ)
    (fun x _ => (transported_derivative G P x).differentiableAt)
    (fun x _ => (transported_derivative_norm G P hG x).le)
    convex_univ (Set.mem_univ s) (Set.mem_univ t)
  simpa [Real.norm_eq_abs, mul_comm] using hh
```

Proof reading. The Lipschitz proof applies a mean-value norm inequality on the convex set of all real times. The existence of the derivative follows from transported_derivative, and its uniform derivative bound comes from transported_derivative_norm. Both are proved for the actual function. Taking one time to be zero gives a bound for $V(t)-P$. Multiplying by e^{tG} converts it into a bound for $e^{tG}P-Pe^{tG}$.

For an orthogonal projection,

$$(1-P)UP=(1-P)(UP-PU)P.$$

The two projection norms are at most one. Hence the leakage corner is bounded by the exponential commutator. Substitute $G=-iH$ for self-adjoint H. Its commutator norm equals the norm of $HP-PH$. This gives the concrete endpoint.

Source: formalization/LeakageBound.lean.

Exact lines 167 to 171. Excerpt leakage_matrix. Hash in Section 12.

```
theorem matrix_commutator_leakage_bound {n : Type*} [Fintype n] [DecidableEq n]
    (H P : Matrix n n ℂ) (hH : H.IsHermitian) (hP : P.IsHermitian)
    (hPP : P*P=P) (t : ℝ) :
    ‖(1-P) * exp ((-(t : ℂ)*Complex.I) • H) * P‖ ≤ |t| * ‖H*P-P*H‖ := by
  exact commutator_leakage_bound H P hH.isSelfAdjoint ⟨hPP, hP.isSelfAdjoint⟩ t
```

Contract. The matrix index may be empty. H must be Hermitian and P must be both Hermitian and idempotent. Time is every real t, including zero and negative times. The proof does not divide by projector rank, dimension or time, so it includes $P=0$ and $P=1$. The estimate concerns a fixed generator H and its exponential, rather than a general time-dependent propagator.

The same file proves a cap at one, giving $\min(1,|t|\|[H,P]\|)$, and exact zero leakage when the commutator is zero. If $H=H_0+V$ with $[H_0,P]=0$ and both terms self-adjoint, only V contributes to the commutator. The elementary estimate $\|[V,P]\|\le 2\|V\|$ gives the perturbation corollary.

Source: formalization/LeakageBound.lean.

Exact lines 151 to 164. Excerpt leakage_perturbation. Hash in Section 12.

```
theorem perturbation_leakage_bound (H₀ V P : A)
    (hH₀ : IsSelfAdjoint H₀) (hV : IsSelfAdjoint V)
    (hP : IsStarProjection P) (hcomm : H₀*P=P*H₀) (t : ℝ) :
```

```
    ‖(1-P)*exp ((-(t : ℂ)*Complex.I) • (H₀+V))*P‖ ≤ 2*|t| *‖V‖ := by
  have hc : (H₀+V)*P-P*(H₀+V) = V*P-P*V := by
    rw [add_mul, mul_add, hcomm]
    noncomm_ring
  have hh := commutator_leakage_bound (H₀+V) P (hH₀.add hV) hP t
  rw [hc] at hh
  calc
    _ ≤ |t| *‖V*P-P*V‖ := hh
    _ ≤ |t| *(2*‖V‖) := mul_le_mul_of_nonneg_left
      (projection_commutator_norm_le V P hP) (abs_nonneg t)
    _ = _ := by ring
```

4.5 Dephasing and a precisely restricted measurement invariance

The dephasing map is $\Delta(\rho) = \sum_a P_a \rho P_a$. It removes off-diagonal corners and is idempotent. For finite complex matrices, trace duality gives

$$\mathrm{tr}(\Delta(E)\rho) = \mathrm{tr}(E\Delta(\rho)).$$

It follows by expanding the finite sum and cyclically rotating matrix factors inside each trace. The subsequent measurement theorem supplies the condition $\Delta(E) = E$, which says precisely that the observable is block diagonal.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 208 to 211. Excerpt `measurement`. Hash in Section 12.

```
theorem measurement_invariance (E rho : Matrix n n ℂ) (hE : P.dephase E = E) :
    Matrix.trace (E * rho) = Matrix.trace (E * P.dephase rho) := by
  rw [← hE] at ⊢
  simpa only [P.dephase_idempotent] using dephase_trace_duality P (P.dephase E) rho
```

Neither positivity nor a trace of one for ρ is needed for this algebraic identity. Calling it a probability equality requires an appropriate state and effect in the application. Taking $E = 1$ proves trace preservation, since completeness and idempotence give $\Delta(1) = 1$. General observables with off-diagonal packet components need not satisfy the invariance premise. Dephasing may change their expectation values.

This chapter explains why the finite core begins with a resolution in a ring and only adds hypotheses when a later operation needs them. The exact circuit formula, conservation identities, exponential dynamics and leakage bound are related constructions. Each theorem states the hypotheses needed for its own conclusion.

5 Moving frames and an actual projected connection

5.1 Differentiate the coordinates before rewriting the equation

Let $U(t)$ be a unitary frame of finite complex matrices and $\psi(t)$ an actual differentiable vector. With $y(t) = U(t)^*\psi(t)$, the effective Hamiltonian is

$$H_{\text{eff}} = U^*HU - iU^*\dot{U}.$$

The source uses real time derivatives of complex-valued entries. The variables `dU` and `dpsi` are candidate derivative values, and the hypotheses `HasDerivAt` assert that they are the actual derivatives at the specified time.

Source: `formalization/MovingFrames.lean`.

Exact lines 23 to 29. Excerpt `frame_product`. Hash in Section 12.

```
theorem matrix_vector_derivative (A : ℝ → Matrix n n ℂ) (y : ℝ → n → ℂ)
    (dA : Matrix n n ℂ) (dy : n → ℂ) (t : ℝ)
    (ha : ∀ i j, HasDerivAt (fun s => A s i j) (dA i j) t)
    (hy : ∀ i, HasDerivAt (fun s => y s i) (dy i) t) (i : n) :
    HasDerivAt (fun s => (A s *ᵥ y s) i) ((dA *ᵥ y t + A t *ᵥ dy) i) t := by
  simpa only [Matrix.mulVec, dotProduct, Pi.add_apply, Pi.mul_apply, Finset.sum_add_distrib] using
    HasDerivAt.fun_sum (u := Finset.univ) (fun j _ => (ha i j).mul (hy j))
```

Proof reading. The proof applies the scalar product rule to each summand of the product of a matrix and a vector, then differentiates the finite sum. It produces the two expected terms without assuming differentiability of the transformed coordinates in advance. The lemma for differentiating an adjoint conjugates the derivative of the transposed entry. It uses the fact that conjugation is linear over the real numbers.

Differentiating $U^*U = 1$ gives $\dot{U}^*U + U^*\dot{U} = 0$, so $U^*\dot{U}$ is skew-adjoint and $-iU^*\dot{U}$ is Hermitian. The conjugated term U^*HU is Hermitian when H is Hermitian. That last premise is explicitly present in the final source. Unitarity of U by itself does not establish Hermiticity of an arbitrary H.

Source: `formalization/MovingFrames.lean`.

Exact lines 110 to 132. Excerpt `moving_coordinates`. Hash in Section 12.

```
theorem moving_frame_coordinates [DecidableEq n]
    (U : ℝ → Matrix n n ℂ) (psi : ℝ → n → ℂ)
    (H dU : Matrix n n ℂ) (dpsi : n → ℂ) (t : ℝ)
    (hu : ∀ i j, HasDerivAt (fun s => U s i j) (dU i j) t)
    (hp : ∀ i, HasDerivAt (fun s => psi s i) (dpsi i) t)
    (hleft : ∀ s, (U s).conjTranspose * U s = 1)
    (hright : ∀ s, U s * (U s).conjTranspose = 1)
    (hsch : Complex.I • dpsi = H *ᵥ psi t)
    (hH : H.conjTranspose = H) :
    let y := fun s => (U s).conjTranspose *ᵥ psi s
    let dy := dU.conjTranspose *ᵥ psi t + (U t).conjTranspose *ᵥ dpsi
    (∀ i, HasDerivAt (fun s => y s i) (dy i) t) ∧
      Complex.I • dy = effectiveHamiltonian (U t) H dU *ᵥ y t ∧
```

```
    (effectiveHamiltonian (U t) H dU).conjTranspose =
      effectiveHamiltonian (U t) H dU := by
  dsimp only
  have hy := matrix_vector_derivative (fun s => (U s).conjTranspose) psi
    dU.conjTranspose dpsi t (adjoint_derivative U dU t hu) hp
  refine ⟨hy, ?_, effective_hamiltonian_hermitian U H dU t hu hleft hH⟩
  apply moving_frame_equation U psi (fun s => (U s).conjTranspose *ᵥ psi s)
    H dU dpsi _ t hu hp hy (hleft t) _ hsch
  intro s
  rw [Matrix.mulVec_mulVec, hright, Matrix.one_mulVec]
```

Contract. The matrix index is finite with decidable equality. Entrywise derivative hypotheses are at one real time t. Both unitary identities hold for every time, permitting construction of the inverse coordinates. The original Schrödinger equation $i\dot{\psi} = H\psi(t)$ is a premise, and $H^* = H$ is the premise for the Hermiticity conclusion. The theorem constructs y, constructs its derivative and proves its equation. Existence of a solution for arbitrary initial data is a separate question.

Proof reading. After unfolding the two `let` expressions, the proof obtains the actual derivative of $U^*\psi$ from the product rule. It gives that derivative proof to `moving_frame_equation`. Unitarity on the right reconstructs $\psi = Uy$, while unitarity on the left removes U^*U when the differentiated equation is multiplied by U^*. The separate Hermiticity lemma uses the differentiated unitary identity and the supplied Hermiticity of H.

5.2 The canonical connection derives its identities for first derivatives

For a differentiable complete orthogonal family $P_a(t)$, write $D_a = \dot{P}_a(t)$. Differentiating multiplication gives

$$D_a P_b + P_a D_b = \delta_{ab} D_a.$$

Differentiating completeness gives $\sum_a D_a = 0$. Differentiating self-adjointness gives $D_a^* = D_a$. These are consequences of actual derivatives of the original functions.

Source: `formalization/CanonicalConnection.lean`.

Exact lines 78 to 95. Excerpt `connection_jet`. Hash in Section 12.

```
theorem differentiated_orthogonality
    (hd : ∀ a, HasDerivAt (P a) (D a) t)
    (hmul : ∀ s a b, P a s * P b s = if a = b then P a s else 0)
    (a b : I) : D a * P b t + P a t * D b = if a = b then D a else 0 := by
  have h := (hd a).mul (hd b)
  change HasDerivAt (fun s => P a s * P b s) (D a * P b t + P a t * D b) t at h
  by_cases hab : a = b
  · subst b
    have heq : (fun s => P a s * P a s) = P a := by
      funext s
      simpa only [ite_true] using hmul s a a
    rw [heq] at h
    simpa only [ite_true] using h.unique (hd a)
  · have heq : (fun s => P a s * P b s) = fun _ => (0 : R) := by
      funext s
      simpa only [hab, ite_false] using hmul s a b
```

```
    rw [heq] at h
    simpa only [hab, ite_false] using h.unique (hasDerivAt_const t (0 : R))
```

The general algebraic part of the file can take these jet identities as hypotheses. The endpoint below proves those hypotheses from the derivative data before applying that algebraic part. It constructs

$$C = \sum_a D_a P_a = \frac{1}{2} \sum_a [D_a, P_a]$$

and establishes $C^* = -C$ and $[C, P_b] = D_b$.

Source: `formalization/CanonicalConnection.lean`.

Exact lines 113 to 131. Excerpt `canonical_connection`. Hash in Section 12.

```
theorem canonical_connection_of_hasDerivAt
    (hd : ∀ a, HasDerivAt (P a) (D a) t)
    (hmul : ∀ s a b, P a s * P b s = if a = b then P a s else 0)
    (hsum : ∀ s, ∑ a, P a s = 1)
    (hstar : ∀ s a, star (P a s) = P a s) :
    let C := canonical (fun a => P a t) D
    star C = -C ∧
      (∀ b, C * P b t - P b t * C = D b) ∧
      (1 / 2 : ℝ) • (∑ a, (D a * P a t - P a t * D a)) = C := by
  dsimp only
  have hj := differentiated_orthogonality P D t hd hmul
  have hdiag (a : I) : D a * P a t + P a t * D a = D a := by
    simpa only [ite_true] using hj a a
  have hz := differentiated_completeness P D t hd hsum
  refine ⟨?_, ?_, ?_⟩
  · exact canonical_skewAdjoint _ _ (hstar t)
      (differentiated_selfAdjoint P D t hd hstar) hz hdiag
  · exact canonical_commutator _ _ (hmul t) (hsum t) hj
  · exact canonical_half_commutator _ _ hz hdiag
```

Contract. The theorem for actual derivatives works in a normed real algebra with a star ring structure, compatible real scalar action and continuous involution. The finite packet index has decidable equality. The projector multiplication, completeness and adjoint identities hold for all real times. This theorem treats a curve of projectors. Manifold curvature is developed in the later modules.

The skew-adjointness calculation sums $D_a P_a + P_a D_a = D_a$. Their sum vanishes, giving $C + C^* = 0$. The commutator calculation first isolates $C P_b = D_b P_b$, then obtains $D_b + P_b C = D_b P_b$ from the differentiated multiplication identities and completeness. Subtracting gives the desired commutator. The factor of one half uses the real scalar structure and the same zero sum.

5.3 The off-diagonal derivative block

For an idempotent P that varies with a parameter, differentiating $P^2 = P$ in direction v gives

$$D_v P\, P + P\, D_v P = D_v P.$$

Multiplying this equation by P and cancelling an identical summand gives $P(D_v P)P = 0$. The curvature calculation therefore uses a consequence of idempotence and differentiation.

Source: `formalization/ProjectedCurvature.lean`.

Exact lines 22 to 26. Excerpt `sandwich_zero`. Hash in Section 12.

```
theorem jet_sandwich_zero (P D : R) (hP : P*P=P)
    (hD : D*P+P*D=D) : P*D*P=0 := by
  have hh := congrArg (fun z => P*z) hD
  simp only [mul_add, ← mul_assoc, hP] at hh
  exact add_eq_right.mp hh
```

For a range section s with $Ps = s$, differentiating the range condition gives

$$D_v s = (D_v P)s + P(D_v s).$$

The chart connection is $\nabla_X s = P(D_X s)$. Its curvature is the actual commutator

$$R(X,Y)s = \nabla_X \nabla_Y s - \nabla_Y \nabla_X s - \nabla_{[X,Y]} s.$$

The bracket term is essential: it cancels the change in the vector fields in the second directional derivatives. The proof uses Mathlib's actual Fréchet derivative and actual Lie bracket, rather than independently supplied coefficients of second order.

Source: `formalization/LocalProjectedCurvature.lean`.

Exact lines 79 to 94. Excerpt `local_curvature`. Hash in Section 12.

```
theorem projected_connection_curvature (P : E → F →L[ℝ] F)
    (X Y : E → E) (s : E → F) (x : E)
    (hP : DifferentiableAt ℝ P x) (hPP : ∀ x, P x * P x = P x)
    (hX : DifferentiableAt ℝ X x) (hY : DifferentiableAt ℝ Y x)
    (hs : ContDiffAt ℝ 2 s x) (hPs : ∀ x, P x (s x)=s x) :
    curvature P X Y s x = curvatureForm P x (X x) (Y x) (s x) := by
  have hs1 : DifferentiableAt ℝ s x := hs.differentiableAt (by norm_num)
  have hz (v : E) (z : F) : P x (fderiv ℝ P x v (P x z)) = 0 := by
    have hh := congrArg (fun L : F →L[ℝ] F => L z)
      (projector_differential_sandwich P x hP hPP v)
    simpa [mul_apply_eq_comp] using hh
  rw [curvature_expansion P X Y s x hP hPP hX hY hs]
  conv_lhs => rw [section_jet P s x hP hs1 hPs (Y x),
    section_jet P s x hP hs1 hPs (X x)]
  simp only [map_add, hz, add_zero]
  simp [curvatureForm, hPs, map_sub]
```

Contract. The local theorem has real normed model and fiber spaces. P, X and Y are differentiable at the specified point. The section s is twice continuously differentiable there. The idempotent equation

and the equation expressing that the section lies in the range hold as functions. An earlier theorem in global chart coordinates assumes global differentiability. The local variant supports use inside arbitrary chart neighborhoods.

Proof reading. Expand the derivative of the projected directional derivative by the product rule. The derivative identity for the Lie bracket cancels the second derivatives of s and the derivatives of X, Y. Substitute the differentiated identity for a section in the range into the two remaining terms involving first derivatives. The components $P(D_v P)P(D_w s)$ vanish by the sandwich lemma. The surviving term is $P[(D_X P), (D_Y P)]s$, and $Ps = s$ puts the final right-hand P into the curvature form.

5.4 Build the range atlas from the projector

The global development begins with an operator family $P : M \to \mathcal{L}_{\mathbb{R}}(F, F)$. Its fiber is the fixed subspace $\{v : P(x)v = v\}$ with the subspace topology. The total space is the subtype of $M \times F$ consisting of such fixed pairs. The atlas is constructed from P rather than supplied as a hypothesis.

For idempotents p, q, define

$$T(p, q) = pq + (1 - p)(1 - q).$$

This is identity at $p = q$, and it intertwines the two projectors. Near a base point a, the map $T(P(x), P(a))$ is invertible because invertibility is open in the complete operator algebra and it is identity at $x = a$.

Source: `formalization/GlobalRangeBundle.lean`.

Exact lines 43 to 43. Excerpt `range_transport`. Hash in Section 12.

```
def transport (p q : F →L[ℝ] F) : F →L[ℝ] F := p*q+(1-p)*(1-q)
```

Source: `formalization/GlobalRangeBundle.lean`.

Exact lines 49 to 54. Excerpt `range_intertwines`. Hash in Section 12.

```
theorem transport_intertwines (p q : F →L[ℝ] F) (hp : p*p=p) (hq : q*q=q) :
    p*transport p q=transport p q*q := by
  unfold transport
  have ha : p*(p*q)=p*q := by rw [← mul_assoc,hp]
  have hb : p*(q*q)=p*q := by rw [hq]
  noncomm_ring [hp,hq,ha,hb]
```

Let $U_a = \{x : T(P(x), P(a)) \text{ is invertible}\}$. Reconstruction transports the fixed fiber at a into the ambient space at x. Coordinates apply the inverse transport and project to the fixed fiber at a. The intertwiners show that both maps land in the correct fibers. On U_a, the two inverse identities follow from genuine inverse cancellation.

Source: `formalization/GlobalRangeBundle.lean`.

Exact lines 180 to 209. Excerpt `range_product`. Hash in Section 12.

```
def localProduct (P : M → F →L[ℝ] F) (hPP : ∀ x, P x*P x=P x)
    (hP : ContMDiff I 𝓘(ℝ, F →L[ℝ] F) ∞ P) (a : M) :
    LocalTotal P a ≃ₜ (domain P a × fiber (P a)) where
```

```
  toFun z := ((z.val.val.1,z.property),toCoord P hPP a z.val.val.1 z.val.val.2)
  invFun y := (((y.1.val,fromCoord P a y.1.val y.2),
    fromCoord_range P hPP a y.1.val y.2),y.1.property)
  left_inv := by
    intro z
    apply Subtype.ext
    apply Subtype.ext
    apply Prod.ext
    · rfl
    · exact fromCoord_toCoord P hPP a _ z.property _ z.val.property
  right_inv := by
    intro y
    apply Prod.ext
    · rfl
    · exact toCoord_fromCoord P hPP a _ y.1.property y.2
  continuous_toFun := by
    have hx : Continuous (fun z : LocalTotal P a => z.val.val.1) := by fun_prop
    have hv : Continuous (fun z : LocalTotal P a => z.val.val.2) := by fun_prop
    have hc : Continuous (fun z : LocalTotal P a => toCoord P hPP a z.val.val.1) :=
      (smooth_toCoord P hPP hP a).continuousOn.comp_continuous hx (fun z => z.property)
    exact (hx.subtype_mk _).prodMk (hc.clm_apply hv)
  continuous_invFun := by
    have hx : Continuous (fun y : domain P a × fiber (P a) => y.1.val) := by fun_prop
    have hv : Continuous (fun y : domain P a × fiber (P a) => y.2) := by fun_prop
    have hc : Continuous (fun y : domain P a × fiber (P a) => fromCoord P a y.1.val) :=
      (smooth_fromCoord P hP a).continuous.comp hx
    exact ((hx.prodMk (hc.clm_apply hv)).subtype_mk _).subtype_mk _
```

This declaration is a homeomorphism from the restricted actual total space to $U_a \times \mathrm{fiber}(P(a))$. In addition to inverse laws, it proves continuity in both directions using the smooth coordinate and reconstruction maps. Transitions are composed from these actual maps. They are continuous linear maps in each fiber, smooth in the base, and satisfy the cocycle law by cancelling the reconstruction and coordinate maps in the middle.

Source: `formalization/GlobalRangeBundle.lean`.

Exact lines 241 to 252. Excerpt `range_atlas`. Hash in Section 12.

```
theorem smooth_range_bundle_atlas (P : M → F →L[ℝ] F) (hPP : ∀ x, P x*P x=P x)
    (hP : ContMDiff I 𝓘(ℝ, F →L[ℝ] F) ∞ P) :
    (∀ a, IsOpen (domain P a) ∧ a ∈ domain P a) ∧
    (∀ a, Nonempty (LocalTotal P a ≃ₜ (domain P a × fiber (P a)))) ∧
    (∀ a b, ContMDiffOn I 𝓘(ℝ, fiber (P a) →L[ℝ] fiber (P b)) ∞
      (transition P hPP a b) (domain P a ∩ domain P b)) ∧
    (∀ a x, x ∈ domain P a → transition P hPP a a x=1) ∧
    (∀ a b c x, x ∈ domain P a ∩ domain P b ∩ domain P c →
      (transition P hPP b c x).comp (transition P hPP a b x)=transition P hPP a c x) := by
  exact ⟨fun a => ⟨isOpen_domain P hP.continuous a,mem_domain P hPP a⟩,
    fun a => ⟨localProduct P hPP hP a),smooth_transition P hPP hP,
    transition_self P hPP,fun a b c x hx => transition_cocycle P hPP a b c x hx.1.2⟩
```

Contract. The construction of the range atlas uses a smooth real projector map on a charted parameter space and a complete real normed fiber space. This bundle construction requires idempotence and

does not require self-adjointness. The explicit atlas theorem does not demand a global chart. Local linear fiber equivalences prove local constancy of finite rank whenever finite rank is being interpreted. The underlying fiber equivalence is the stronger structural statement.

The implementation uses this explicit smooth atlas directly. It does not package it as a Mathlib `ContMDiffVectorBundle` instance. The total space, open domains, local product homeomorphisms, transitions and their identities are all constructed and proved. The distinction concerns how these results are packaged in Lean.

5.5 The global derivative and gauge gluing

On the manifold, the global connection is defined using `mvfderiv`, the derivative of a map from a manifold to a vector space. A tangent vector at x has type `TangentSpace I x`, so a vector field is a dependent function giving one such vector at every point.

Source: `formalization/GlobalProjectedConnection.lean`.

Exact lines 29 to 30. Excerpt `global_nabla`. Hash in Section 12.

```
def nabla (P : M → F →L[ℝ] F) (X : ∀ x : M, TangentSpace I x) (s : M → F) (x : M) : F :=
  P x (mvfderiv I s x (X x))
```

Source: `formalization/GlobalProjectedConnection.lean`.

Exact lines 59 to 63. Excerpt `global_leibniz`. Hash in Section 12.

```
theorem nabla_leibniz (P : M → F →L[ℝ] F) (X : ∀ x : M, TangentSpace I x)
    (s : M → F) (c : M → ℝ) (x : M) (hs : MDifferentiableAt I 𝓘(ℝ,F) s x)
    (hc : MDifferentiableAt I 𝓘(ℝ,ℝ) c x) (hPs : P x (s x)=s x) :
    nabla I P X (c • s) x = (mvfderiv I c x (X x)) • s x + c x • nabla I P X s x := by
  simp [nabla,mvfderiv_smul hc hs,hPs,add_comm]
```

The global range property follows immediately from idempotence. Additivity in sections uses actual differentiability premises. The Leibniz rule needs the range condition for $s(x)$ because the scalar derivative term initially contains $P(x)s(x)$. Linearity in the vector field uses linearity of the derivative. A germ lemma says that locally equal functions have the same covariant derivative at the point.

Local coordinate connections reconstruct a range section, differentiate it globally and then take coordinates. On an open domain, local reconstruction equals the original section in a neighborhood. The germ lemma therefore identifies the local and global connection. On an overlap, coordinate transitions carry one representation into the other.

Source: `formalization/GlobalProjectedConnection.lean`.

Exact lines 115 to 124. Excerpt `global_glue`. Hash in Section 12.

```
theorem local_connections_glue (P : M → F →L[ℝ] F) (hPP : ∀ x, P x*P x=P x)
    (hP : ContMDiff I 𝓘(ℝ,F →L[ℝ] F) ∞ P) (a b : M)
    (X : ∀ x : M, TangentSpace I x) (s : M → F) (hPs : ∀ x, P x (s x)=s x)
    (x : M) (hx : x ∈ domain P a ∩ domain P b) :
    transition P hPP a b x (localNabla I P hPP a X (fun y => toCoord P hPP a y (s y)) x) =
      localNabla I P hPP b X (fun y => toCoord P hPP b y (s y)) x := by
```

```
  rw [localNabla_represents_global I P hPP hP a X s hPs x hx.1,
    localNabla_represents_global I P hPP hP b X s hPs x hx.2]
  change toCoord P hPP b x (fromCoord P a x (toCoord P hPP a x (nabla I P X s x))) = _
  rw [fromCoord_toCoord P hPP a x hx.1 _ (nabla_in_range I P hPP X s x)]
```

Source: `formalization/ProjectedBundleCompletion.lean`.

Exact lines 40 to 53. Excerpt `smooth_section`. Hash in Section 12.

```
theorem smooth_section_iff (P : M → F →L[ℝ] F) (hPP : ∀ x, P x*P x=P x)
    (hP : ContMDiff I 𝓘(ℝ,F →L[ℝ] F) ∞ P) (a x : M) (hx : x ∈ domain P a)
    (s : M → F) (hPs : ∀ᶠ y in 𝒩 x, P y (s y)=s y) :
    ContMDiffAt I 𝓘(ℝ,F) ∞ s x ↔
      ContMDiffAt I 𝓘(ℝ,fiber (P a)) ∞ (fun y => toCoord P hPP a y (s y)) x := by
  constructor
  · intro hs
    exact ((smooth_toCoord P hPP hP a x hx).contMDiffAt
      ((isOpen_domain P hP.continuous a).mem_nhds hx)).clm_apply hs
  · intro hs
    have hr := (smooth_fromCoord P hP a x).clm_apply hs
    apply hr.congr_of_eventuallyEq
    filter_upwards [hPs,(isOpen_domain P hP.continuous a).mem_nhds hx] with y hy hya
    exact (fromCoord_toCoord P hPP a y hya (s y) hy).symm
```

Source: `formalization/ProjectedBundleCompletion.lean`.

Exact lines 57 to 70. Excerpt `gauge_law`. Hash in Section 12.

```
theorem local_connections_gauge (P : M → F →L[ℝ] F) (hPP : ∀ x, P x*P x=P x)
    (hP : ContMDiff I 𝓘(ℝ,F →L[ℝ] F) ∞ P) (a b x : M)
    (hx : x ∈ domain P a ∩ domain P b) (X : ∀ y : M, TangentSpace I y)
    (u : M → fiber (P a)) :
    localNabla I P hPP b X (fun y => transition P hPP a b y (u y)) x =
      transition P hPP a b x (localNabla I P hPP a X u x) := by
  have hs : ∀ y, P y (fromCoord P a y (u y))=fromCoord P a y (u y) :=
    fun y => fromCoord_range P hPP a y (u y)
  change localNabla I P hPP b X (fun y => toCoord P hPP b y (fromCoord P a y (u y))) x = _
  rw [localNabla_represents_global I P hPP hP b X (fun y => fromCoord P a y (u y)) hs x hx.2]
  change toCoord P hPP b x (nabla I P X (fun y => fromCoord P a y (u y)) x) =
    toCoord P hPP b x (fromCoord P a x
      (toCoord P hPP a x (nabla I P X (fun y => fromCoord P a y (u y)) x)))
  rw [fromCoord_toCoord P hPP a x hx.1 _ (nabla_in_range I P hPP X _ x)]
```

The equivalence for smooth sections needs only an eventual range condition near the point, not an unproved globally extended section. The gauge law in the completion file applies to arbitrary local coordinate functions and compares their reconstructed connection values on an overlap. It is not restricted to a prechosen global frame.

5.6 Manifold curvature and its endpoint

The final curvature theorem uses a smooth real boundaryless manifold modeled on a complete real normed space E. The model is $\mathcal{J}(\mathbb{R}, E)$. The manifold has `ChartedSpace E M` and a smooth `IsManifold` instance for that model. The source pulls vector fields into a chart centered at the point, identifies the manifold derivative and bracket with their chart versions, and then proves the cancellation using actual manifold operations.

Source: `formalization/ManifoldProjectedCurvature.lean`.

Exact lines 214 to 232. Excerpt `manifold_curvature`. Hash in Section 12.

```
theorem projected_connection_curvature (P : M → F →L[ℝ] F)
    (X Y : ∀ x : M, TangentSpace 𝓙(ℝ,E) x) (s : M → F) (x : M)
    (hP : ContMDiff 𝓙(ℝ,E) 𝓙(ℝ,F →L[ℝ] F) ∞ P) (hPP : ∀ x, P x*P x=P x)
    (hX : ContMDiff 𝓙(ℝ,E) 𝓙(ℝ,E).tangent ∞
      (fun x => (X x : TangentBundle 𝓙(ℝ,E) M)))
    (hY : ContMDiff 𝓙(ℝ,E) 𝓙(ℝ,E).tangent ∞
      (fun x => (Y x : TangentBundle 𝓙(ℝ,E) M)))
    (hs : ContMDiff 𝓙(ℝ,E) 𝓙(ℝ,F) ∞ s) (hPs : ∀ x, P x (s x)=s x) :
    curvature 𝓙(ℝ,E) P X Y s x = curvatureForm 𝓙(ℝ,E) P x (X x) (Y x) (s x) := by
  have hP1 := hP.mdifferentiableAt (x := x) (by simp)
  have hs1 := hs.mdifferentiableAt (x := x) (by simp)
  have hz (v : TangentSpace 𝓙(ℝ,E) x) (z : F) : P x (mvfderiv 𝓙(ℝ,E) P x v (P x z))=0 := by
    have hh := congrArg (fun L : F →L[ℝ] F => L z)
      (projector_differential_sandwich P x hP1 hPP v)
    simpa [mul_apply_eq_comp] using hh
  rw [curvature_expansion P X Y s x hP hPP hX hY hs]
  conv_lhs => rw [section_jet P s x hP1 hs1 hPs (Y x),section_jet P s x hP1 hs1 hPs (X x)]
  simp only [map_add,hz,add_zero]
  simp [curvatureForm,hPs,map_sub]
```

Contract. Here P, the two vector fields and the ambient range section are smooth. The equations $P(x)^2 = P(x)$ and $P(x)s(x) = s(x)$ are given globally. The conclusion is $R(X, Y)s = P(dP \wedge dP)P\,s$ at an arbitrary point. The final manifold curvature endpoint assumes no global chart, no supplied curvature identity and no supplied range atlas. The atlas portion separately requires completeness of the fiber space. Finite complex quantum spaces fit by restriction to their underlying real spaces.

The next theorem proves tensoriality: replacing the two fields and the section by smooth extensions with the same values at the point leaves curvature unchanged. The completion file makes the curvature form alternating and linear over the real numbers, proves it maps into the actual fiber, and defines the resulting continuous linear fiber endomorphism. Constant projector maps, including zero and identity, have zero curvature form.

The six files on projected geometry follow this mathematical development. `ProjectedCurvature` establishes the algebra and calculus on the parameter space. `LocalProjectedCurvature` makes the hypotheses local, and `GlobalRangeBundle` constructs the atlas. `GlobalProjectedConnection` defines the connection and proves that its local descriptions agree. `ManifoldProjectedCurvature` proves the global curvature formula. Finally, `ProjectedBundleCompletion` proves compatibility with intrinsic sections, gauge transformations and the curvature form on each fiber. The later global constructions extend the results of the earliest chart file.

6 Finite channels and the full Knill–Laflamme criterion

6.1 Complete positivity has an actual amplification map

For finite complex matrices, the Kraus channel is the map

$$\Phi(\rho) = \sum_a K_a \rho K_a^*.$$

Its additivity and complex homogeneity construct an actual linear map. Each summand preserves positive semidefiniteness by a matrix congruence, so the finite sum is positive. Trace preservation is a separate theorem under $\sum_a K_a^* K_a = 1$, proved by cyclic trace rotation and completeness of that Kraus sum.

To prove complete positivity the source does not merely restate positivity. It defines blockwise amplification on matrices indexed by $J \times n$ and identifies it entrywise with the actual Kraus family $1_J \otimes K_a$.

Source: `formalization/KrausChannels.lean`.

Exact lines 48 to 56. Excerpt `kraus_amplify`. Hash in Section 12.

```
theorem amplification_eq_kraus {J : Type*} [Fintype J] [DecidableEq J]
    (K : A → Matrix n n ℂ) (rho : Matrix (J × n) (J × n) ℂ) :
    amplify (channel K) rho =
      channel (fun a => (1 : Matrix J J ℂ) ⊗ₖ K a) rho := by
  ext ⟨i, u⟩ ⟨j, v⟩
  simp_rw [amplify, channel, Matrix.sum_apply, Matrix.mul_apply,
    Matrix.conjTranspose_apply, Matrix.kroneckerMap_apply]
  simp [Fintype.sum_prod_type, Matrix.one_apply, apply_ite]
  rfl
```

Source: `formalization/KrausChannels.lean`.

Exact lines 58 to 62. Excerpt `kraus_cp`. Hash in Section 12.

```
theorem channel_completely_positive {J : Type*} [Fintype J] [DecidableEq J]
    (K : A → Matrix n n ℂ) {rho : Matrix (J × n) (J × n) ℂ}
    (hrho : rho.PosSemidef) : (amplify (channel K) rho).PosSemidef := by
  rw [amplification_eq_kraus]
  exact channel_positive _ hrho
```

Contract. The system and Kraus index types are finite. The reference type `J` is an arbitrary finite type with decidable equality. Complete positivity itself does not require Kraus normalization. Normalization is needed for trace preservation and preservation of density matrices. Every finite reference dimension is quantified in the theorem, rather than sampled in an example. Infinite reference systems are not encoded by these finite matrix declarations.

Proof reading. The amplification identity expands entries of a Kronecker product, uses the identity matrix to force equality of the reference indices, and collapses the corresponding sums. Once amplification is literally a Kraus map, the earlier positivity theorem applies without a new positivity hypothesis about the amplified channel.

6.2 Orthogonal syndromes are the recovery stage

The recovery construction begins with a code projection P and an orthogonal normalized family E_a satisfying

$$E_a P = E_a, \qquad E_a^* E_b = \delta_{ab} P.$$

Their output packets $Q_a = E_a E_a^*$ are orthogonal star projections. Their sum Q is a star projection and $Q^\perp = 1 - Q$ is its complement.

Source: `formalization/SyndromeRecovery.lean`.

Exact lines 21 to 27. Excerpt `syndrome_structure`. Hash in Section 12.

```
structure OrthogonalSyndromes (R I : Type*) [Ring R] [StarRing R]
    [Fintype I] [DecidableEq I] where
  code : R
  code_projection : IsStarProjection code
  error : I → R
  restricted : ∀ a, error a * code = error a
  gram : ∀ a b, star (error a) * error b = if a=b then code else 0
```

The decoder is

$$\mathcal{R}(\sigma) = \sum_a E_a^* \sigma E_a + Q^\perp \sigma Q^\perp.$$

The complement term is important. Without it the decoder would generally lose trace on input outside the sum of syndrome output spaces. The implementation uses `Option I` to index the complete Kraus family for recovery: `some a` denotes E_a^*, and `none` denotes $Q^\perp$. Thus the extra branch is encoded in the actual finite family.

Source: `formalization/SyndromeRecovery.lean`.

Exact lines 84 to 90. Excerpt `syndrome_complete`. Hash in Section 12.

```
theorem recoveryKraus_complete :
    (∑ a, star (S.recoveryKraus a)*S.recoveryKraus a) = 1 := by
  rw [Fintype.sum_option]
  simp only [recoveryKraus, star_star, S.complement_projection.isSelfAdjoint.star_eq,
    S.complement_projection.isIdempotentElem.eq]
  change S.complement + S.total = 1
  simp [complement]
```

Proof reading. The sum over `Option I` splits into the complement branch and the syndrome branches. Self-adjointness and idempotence turn the complement contribution into $Q^\perp$. The syndrome contribution is Q. Their sum is the identity. This proves trace preservation on the entire ambient matrix space, not merely on noisy code states.

For an arbitrary operator supported on the code, $P\rho P = \rho$, the decoder of a cross error is

$$\mathcal{R}(E_a \rho E_b^*) = \delta_{ab} \rho.$$

The orthogonal Gram law kills every nonmatching syndrome contribution, and the complement annihilates each error output.

Source: `formalization/SyndromeRecovery.lean`.

Exact lines 103 to 124. Excerpt `syndrome_cross`. Hash in Section 12.

```
theorem decoder_cross_error (rho : R) (hrho : S.code*rho*S.code=rho) (a b : I) :
    S.decoder (S.error a*rho*star (S.error b)) = if a=b then rho else 0 := by
  have hc (c : I) :
      star (S.error c)*(S.error a*rho*star (S.error b))*S.error c =
        if c=a then (if a=b then rho else 0) else 0 := by
    have heq : star (S.error c)*(S.error a*rho*star (S.error b))*S.error c =
        (star (S.error c)*S.error a)*rho*(star (S.error b)*S.error c) := by
      noncomm_ring
    rw [heq, S.gram, S.gram]
    by_cases hca : c=a
    · subst c
      by_cases hab : a=b
      · subst b; simp [hrho]
      · simp [hab, Ne.symm hab]
    · simp [hca]
  have hq : S.complement*(S.error a*rho*star (S.error b))*S.complement=0 := by
    calc
      _ = (S.complement*S.error a)*rho*(star (S.error b)*S.complement) := by
        noncomm_ring
      _ = 0 := by rw [S.complement_error, zero_mul, zero_mul]
  simp only [decoder, hc, hq, add_zero]
  simp
```

Over an abstract star ring, this gives an algebraic correction identity. The construction for complex matrices additionally proves positivity, trace preservation, complete positivity at every finite reference dimension and preservation of density matrices. A weighted normalized mixture and a coherent error superposition are then handled by linearity and the formula for cross errors.

This file alone would prove recovery for supplied orthogonal syndrome data. It would not prove that arbitrary Knill–Laflamme Gram data can be put into that form, and it would not prove necessity of the Gram condition. The current source has separate modules for those obligations.

6.3 General Gram data are not assumed diagonal

The finite Knill–Laflamme data consist of a star projection P, errors E_a, and a positive semidefinite coefficient matrix C with

$$PE_a^*E_bP = C_{ab}P.$$

No invertibility of C is required. The matrix can be singular and can have off-diagonal coefficients.

Source: `formalization/KnillLaflammeSufficiency.lean`.

Exact lines 187 to 193. Excerpt `kl_data`. Hash in Section 12.

```
structure Data (n I : Type*) [Fintype n] [DecidableEq n] [Fintype I] [DecidableEq I] where
  code : Matrix n n ℂ
  code_projection : IsStarProjection code
  error : I → Matrix n n ℂ
  gramMatrix : Matrix I I ℂ
```

```
  gram_positive : gramMatrix.PosSemidef
  gram : ∀ a b, code * star (error a) * error b * code = gramMatrix a b • code
```

First restrict each error to the code by replacing E_a with $E_a P$. This changes neither the noisy output of a supported operator nor its compressed Gram data. The source's `mix E U r` uses the columns of U as the new error coordinates. It proves that mixing transforms the Gram matrix to $U^* C U$ and that unitary mixing leaves the Kraus noise map unchanged.

The actual Hermitian spectral theorem supplies `eigenvectorUnitary` and real eigenvalues. Positive semidefiniteness proves those eigenvalues are nonnegative. The declaration below constructs the diagonal error data directly from that spectral information.

Source: `formalization/KnillLaflammeSufficiency.lean`.

Exact lines 204 to 214. Excerpt `kl_diagonal`. Hash in Section 12.

```
noncomputable def diagonal : DiagonalErrors n I where
  code := K.code
  code_projection := K.code_projection
  error := mix (fun a => K.error a*K.code)
    (K.gram_positive.isHermitian.eigenvectorUnitary : Matrix I I ℂ)
  restricted := mix_restricted K.code (fun a => K.error a*K.code)
    (fun a => by rw [mul_assoc, K.code_projection.isIdempotentElem.eq]) _
  weight := K.gram_positive.isHermitian.eigenvalues
  nonnegative := K.gram_positive.eigenvalues_nonneg
  gram := diagonalized_gram K.code (fun a => K.error a*K.code) K.gramMatrix
    K.gram_positive K.restricted_error_gram
```

Proof reading. The `error` field mixes the errors restricted to the code by the spectral unitary. The `weight` field uses the actual eigenvalues. The `gram` field applies the proved diagonalization identity. The `restricted` field uses idempotence of the code projector, and the `nonnegative` field comes from positivity of the input Gram matrix. These fields construct the required diagonal data.

6.4 Singular directions are eliminated without division by zero

If a diagonal weight is zero, the corresponding restricted error satisfies $E^* E = 0$. For finite complex matrices this forces $E = 0$. Therefore the construction does not normalize a zero spectral direction.

Source: `formalization/KnillLaflammeSufficiency.lean`.

Exact lines 107 to 110. Excerpt `kl_null`. Hash in Section 12.

```
theorem zero_weight_error (a : I) (ha : D.weight a = 0) : D.error a = 0 := by
  have hh : star (D.error a) * D.error a = 0 := by
    simpa only [ite_true, ha, Complex.ofReal_zero, zero_smul] using D.gram a a
  exact Matrix.conjTranspose_mul_self_eq_zero.mp hh
```

The active index is the subtype `{a : I // 0 < D.weight a}`. For an active error define $F_a = \lambda_a^{-1/2} E_a$. The source proves the square root is nonzero and its square equals the weight before using its inverse. The normalized errors satisfy $F_a^* F_b = \delta_{ab} P$, so the preceding syndrome construction applies.

For an unnormalized original error family, the decoder corrects the noise to the exact scale

$$\mathcal{R}\left(\sum_a E_a \rho E_a^*\right) = \operatorname{tr}(C)\rho.$$

Unitary mixing preserves the noise, the restricted noise agrees on the code, and the spectral eigenvalue sum is the trace of C. If the original errors form a normalized channel and $P \neq 0$, compressing the Kraus completeness relation gives $\operatorname{tr}(C)P = P$, so the scale is one. If $P = 0$, every supported operator is zero and the correction assertion follows directly.

Source: `formalization/KnillLaflammeSufficiency.lean`.

Exact lines 252 to 259. Excerpt `kl_corrects`. Hash in Section 12.

```
theorem recovery_corrects (hcomplete : (∑ a, star (K.error a)*K.error a) = 1)
    (rho : Matrix n n ℂ) (hrho : K.code*rho*K.code=rho) :
    K.diagonal.syndromes.decoder (noise K.error rho) = rho := by
  rw [K.recovery_noise_scaled rho hrho]
  by_cases hp : K.code = 0
  · have hr : rho=0 := by simpa [hp] using hrho.symm
    simp [hr]
  · rw [K.gram_trace_one hp hcomplete, one_smul]
```

Contract. This sufficiency statement assumes finite complex matrices, a star code projection, a positive Gram matrix and its compressed Gram relation, together with global Kraus normalization for exact channel correction with scale one. The correction conclusion covers all operators supported on the code, including those that are not positive. The decoder is an actual finite CPTP extension on the whole space. Its spectral construction is mathematical and noncomputable. The theorem supplies neither a numerical eigenvalue routine nor a performance guarantee for an executable decoder.

6.5 Necessity starts from actual recovery

The reverse implication takes a finite recovery family R_r with $\sum_r R_r^* R_r = 1$, and assumes its actual composite Kraus map fixes every operator supported on the code. It must derive the Gram condition rather than take it as a premise.

The key rigidity lemma says that if A^*A is the rank-one Gram matrix of a vector p, then every row of A is a scalar multiple of p. If $p = 0$, positivity gives $A = 0$. Otherwise choose a nonzero coordinate p_t. A vector supported in two coordinates and orthogonal to p is in the kernel of A^*A, hence of A. The resulting equation in two coordinates yields the row proportionality.

Source: `formalization/KnillLaflammeNecessity.lean`.

Exact lines 19 to 45. Excerpt `kl_rigidity`. Hash in Section 12.

```
theorem rank_one_gram_rigidity {q J : Type*} [Fintype q] [DecidableEq q]
    [Fintype J] [DecidableEq J] (A : Matrix J q ℂ) (p : q → ℂ)
    (hgram : A.conjTranspose*A = Matrix.vecMulVec (star p) p) :
    ∃ z : J → ℂ, ∀ j t, A j t = z j*p t := by
  by_cases hp : p=0
  · have hA : A=0 := Matrix.conjTranspose_mul_self_eq_zero.mp (by simpa [hp] using hgram)
    exact ⟨fun _ => 0, by simp [hA]⟩
```

```
  · obtain ⟨t,ht⟩ : ∃ t, p t ≠ 0 := Function.ne_iff.mp hp
    refine ⟨fun j => A j t / p t, ?_⟩
    intro j s
    let v : q → ℂ := Pi.single s (p t) - Pi.single t (p s)
    have hpv : p ⬝ᵥ v = 0 := by
      simp [v, dotProduct_sub, dotProduct_single, mul_comm]
    have hav : A *ᵥ v=0 := by
      apply (Matrix.conjTranspose_mul_self_mulVec_eq_zero A v).mp
      rw [hgram]
      ext u
      change (∑ x, star (p u) * p x * v x) = 0
      simp_rw [mul_assoc]
      rw [← Finset.mul_sum]
      change star (p u) * (p ⬝ᵥ v) = 0
      rw [hpv, mul_zero]
    have heq : A j s*p t - A j t*p s=0 := by
      simpa [v, Matrix.mulVec_sub, Matrix.mulVec_single, mul_comm] using congrFun hav j
    change A j s = (A j t / p t) * p s
    field_simp [ht]
    linear_combination heq
```

To apply this to Kraus operators, flatten each operator's entries into a row indexed by a pair of matrix coordinates. Evaluate the asserted map equality on matrix units. This yields the rank-one Gram equality for the flattened matrix. Each composite recovery operator $R_r E_a P$ must therefore be $z_{ra} P$.

Now insert the recovery completeness relation into $P E_a^* E_b P$. It becomes the sum of adjoint products of those composite operators. Their scalar form gives a positive Gram matrix $C = Z^* Z$ with the required coefficients.

Source: `formalization/KnillLaflammeNecessity.lean`.

Exact lines 86 to 105. Excerpt `kl_necessity`. Hash in Section 12.

```
theorem gram_of_recovery (P : Matrix n n ℂ) (hP : IsStarProjection P)
    (E : I → Matrix n n ℂ) (R : J → Matrix n n ℂ)
    (hR : (∑ r, star (R r)*R r)=1)
    (hcorrect : ∀ rho, P*rho*P=rho → noise R (noise E rho)=rho) :
    ∃ C : Matrix I I ℂ, C.PosSemidef ∧
      ∀ a b, P*star (E a)*E b*P=C a b • P := by
  obtain ⟨z,hz⟩ := corrected_kraus_family P hP E R hcorrect
  let Z : Matrix J I ℂ := fun r a => z (r,a)
  refine ⟨Z.conjTranspose*Z, Matrix.posSemidef_conjTranspose_mul_self Z, ?_⟩
  intro a b
  have heq : P*star (E a)*E b*P = ∑ r, star (R r*E a*P)*(R r*E b*P) := by
    calc
      _ = P*star (E a)*(∑ r, star (R r)*R r)*E b*P := by rw [hR, mul_one]
      _ = _ := by
        simp only [Finset.mul_sum, Finset.sum_mul, star_mul,
          hP.isSelfAdjoint.star_eq, mul_assoc]
  rw [heq]
  simp_rw [hz, star_smul, hP.isSelfAdjoint.star_eq, smul_mul_assoc,
    mul_smul_comm, smul_smul, hP.isIdempotentElem.eq]
  simp only [Matrix.mul_apply, Matrix.conjTranspose_apply, Finset.sum_smul, Z]
```

Proof reading. The witness Z^*Z is positive by an existing matrix theorem. The intermediate `heq` is obtained by inserting $\sum_r R_r^* R_r = 1$ without reordering the error factors. After rewriting every $R_r E_a P$ as a scalar times P, self-adjointness and idempotence reduce the sum to the entries of Z^*Z times P. Thus positivity is proved from the recovery, not imposed on an unknown matrix.

6.6 The state formulation is bridged to all code operators

A physical criterion is usually stated for density matrices: positive semidefinite, trace one and supported on the code. The necessity argument above uses all operators supported on the code. The source supplies the missing bridge.

First, a nonzero positive matrix has positive nonzero trace. Scaling by the inverse real trace produces a density matrix. Linearity then removes that normalization. Zero positive matrices are handled separately. Next, the complex polarization identity expresses an arbitrary outer product xy^* as a linear combination of pure positive matrices. Finally, matrix units span all matrices. Two maps that are linear over the complex numbers therefore agree on every operator if they agree on all pure states.

Source: `formalization/QuantumStateBridge.lean`.

Exact lines 119 to 129. Excerpt `density_bridge`. Hash in Section 12.

```
theorem density_corrects_iff_operators {I J : Type*} [Fintype I] [Fintype J]
    (P : Matrix n n ℂ) (hP : IsStarProjection P)
    (E : I → Matrix n n ℂ) (R : J → Matrix n n ℂ) :
    (∀ rho : Matrix n n ℂ, rho.PosSemidef → Matrix.trace rho=1 → P*rho*P=rho →
      noise R (noise E rho)=rho) ↔
    (∀ rho : Matrix n n ℂ, P*rho*P=rho → noise R (noise E rho)=rho) := by
  constructor
  · intro hh rho hs
    exact fixes_operators_of_density P hP ((noiseLinear R).comp (noiseLinear E)) hh rho hs
  · intro hh rho _ _ hs
    exact hh rho hs
```

The predicate `StateCorrectable P E` existentially quantifies a natural number and a recovery family on `Fin m`. Its recovery is normalized and fixes all normalized code density matrices. The final endpoint combines the density bridge with the equivalence already proved on the operator space.

Source: `formalization/QuantumStateBridge.lean`.

Exact lines 139 to 151. Excerpt `kl_criterion`. Hash in Section 12.

```
theorem state_correctable_iff_gram {I : Type*} [Fintype I] [DecidableEq I]
    (P : Matrix n n ℂ) (hP : IsStarProjection P) (E : I → Matrix n n ℂ)
    (hE : (∑ a, star (E a)*E a)=1) :
    StateCorrectable P E ↔ ∃ C : Matrix I I ℂ, C.PosSemidef ∧
      ∀ a b, P*star (E a)*E b*P=C a b • P := by
  rw [← correctable_iff_gram P hP E hE]
  constructor
  · rintro ⟨m,R,hR,hcorrect⟩
    exact ⟨m,R,hR,(density_corrects_iff_operators P hP E R).mp hcorrect⟩
  · rintro ⟨m,R,hR,hcorrect⟩
    exact ⟨m,R,hR,(density_corrects_iff_operators P hP E R).mpr hcorrect⟩

open scoped Kronecker
```

Contract. The code is a star projection. The error family is finite and normalized by $\sum_a E_a^* E_a = 1$. The existential Gram matrix must be positive semidefinite and satisfy every compressed cross relation. The theorem permits the zero code. In that case, no nonzero normalized code density matrix exists, and the operator bridge remains valid because the supported operator space is zero.

6.7 Reference systems and error spans

The theorem for finite reference systems identifies each reference block of $1 \otimes \mathcal{R}$ and $1 \otimes \mathcal{E}$ with the corresponding system noise map. The supported reference matrix condition implies each system block is supported on the code, so the operator correction theorem applies block by block.

Source: `formalization/QuantumStateBridge.lean`.

Exact lines 164 to 182. Excerpt `kl_reference`. Hash in Section 12.

```
theorem recovery_with_reference {I J A : Type*} [Fintype I] [Fintype J] [Fintype A] [DecidableEq
 ↪  A]
    (P : Matrix n n ℂ) (E : I → Matrix n n ℂ) (R : J → Matrix n n ℂ)
    (hcorrect : ∀ rho, P*rho*P=rho → noise R (noise E rho)=rho)
    (rho : Matrix (A × n) (A × n) ℂ)
    (hs : ((1 : Matrix A A ℂ) ⊗ₖ P)*rho*((1 : Matrix A A ℂ) ⊗ₖ P)=rho) :
    amplifiedNoise R (amplifiedNoise E rho)=rho := by
  have hblock (i j : A) :
      P*(Matrix.of fun a b => rho (i,a) (j,b))*P=(Matrix.of fun a b => rho (i,a) (j,b)) := by
    ext u v
    have hh := congrArg (fun M : Matrix (A × n) (A × n) ℂ => M (i,u) (j,v)) hs
    simpa [Matrix.mul_apply, Matrix.kroneckerMap_apply, Fintype.sum_prod_type, Matrix.one_apply]
     ↪  using hh
  ext ⟨i,u⟩ ⟨j,v⟩
  rw [amplifiedNoise_block]
  have hin : (fun a b => amplifiedNoise E rho (i,a) (j,b)) =
      noise E (fun a b => rho (i,a) (j,b)) := by
    ext a b
    exact amplifiedNoise_block E rho i j a b
  rw [hin]
  exact congrArg (fun M : Matrix n n ℂ => M u v) (hcorrect _ (hblock i j))
```

It follows that the same correction identity holds for entangled states with any finite reference system. The displayed algebraic identity requires neither positivity nor a trace of one. These conditions can be supplied to interpret the matrix as a density state. This is stronger than checking only product states and does not rely on assuming separability.

The module for recovery on the error span keeps track of cross coefficients. With the source's Gram orientation,

$$\mathcal{R}(E_a \rho E_b^*) = C_{ba} \rho.$$

The reversed indices are deliberate. Expanding a coherent combination produces the scalar $\sum_{a,b} c_a \overline{c_b} C_{ba}$. Summing such expressions over a new Kraus family F_j in the old error span gives a common scalar multiplier. If the new family is normalized, trace preservation forces that multiplier to one on a nonzero code. For the zero code, the only supported operator is again zero.

Source: formalization/ErrorSpanRecovery.lean.

Exact lines 138 to 165. Excerpt kl_span. Hash in Section 12.

```
theorem recovery_span_channel (F : J → Matrix n n ℂ) (c : J → I → ℂ)
    (hc : ∀ j, F j=∑ a, c j a • K.error a)
    (hF : (∑ j, star (F j)*F j)=1)
    (rho : Matrix n n ℂ) (hrho : K.code*rho*K.code=rho) :
    K.diagonal.syndromes.decoder (noise F rho)=rho := by
  let q : ℂ := ∑ j, ∑ a, ∑ b, c j a*star (c j b)*K.gramMatrix b a
  have hscaled (rho : Matrix n n ℂ) (hs : K.code*rho*K.code=rho) :
      K.diagonal.syndromes.decoder (noise F rho)=q • rho := by
    change K.diagonal.syndromes.decoderLinear (∑ j, F j*rho*star (F j)) = _
    rw [map_sum]
    change (∑ j, K.diagonal.syndromes.decoder (F j*rho*star (F j))) = _
    simp_rw [hc, K.recovery_error_superposition rho hs]
    rw [← Finset.sum_smul]
  rw [hscaled rho hrho]
  by_cases hp : K.code=0
  · have hz : rho=0 := by simpa [hp] using hrho.symm
    simp [hz]
  · have hpos : K.code.PosSemidef := by
      have hx := Matrix.posSemidef_conjTranspose_mul_self K.code
      change (star K.code*K.code).PosSemidef at hx
      rw [K.code_projection.isSelfAdjoint.star_eq, K.code_projection.isIdempotentElem.eq] at hx
      exact hx
    have ht : Matrix.trace K.code ≠ 0 := (hpos.trace_eq_zero_iff).not.mpr hp
    have hs : K.code*K.code*K.code=K.code := by simp only [K.code_projection.isIdempotentElem.eq]
    have hh := congrArg Matrix.trace (hscaled K.code hs)
    rw [K.recovery_preserves_trace, noise_preserves_trace F hF, Matrix.trace_smul] at hh
    have hq : q=1 := mul_right_cancel₀ ht (by simpa using hh.symm)
    rw [hq, one_smul]
```

This is one fixed recovery correcting every normalized channel in the error span, with the coefficients explicitly witnessing span membership. It does not claim correction with scale one for an arbitrary unnormalized linear combination. For unnormalized error families, the separate criterion requires a nonnegative scale t independent of the supported input operator and proves equivalence with the Gram relation.

Source: formalization/ErrorSpanRecovery.lean.

Exact lines 188 to 209. Excerpt kl_family. Hash in Section 12.

```
theorem error_family_correctable_iff_gram (P : Matrix n n ℂ) (hP : IsStarProjection P)
    (E : I → Matrix n n ℂ) :
    ErrorFamilyCorrectable P E ↔ ∃ C : Matrix I I ℂ, C.PosSemidef ∧
      ∀ a b, P*star (E a)*E b*P=C a b • P := by
  constructor
  · rintro ⟨m,R,hR,t,ht,hcorrect⟩
    exact gram_of_scaled_recovery P hP E R hR t ht hcorrect
  · rintro ⟨C,hC,hgram⟩
    let K : Data n I := ⟨P,hP,E,C,hC,hgram⟩
    let e := Fintype.equivFin (Option K.diagonal.Active)
```

```
    let R := K.diagonal.syndromes.recoveryKraus
    refine ⟨Fintype.card (Option K.diagonal.Active), fun r => R (e.symm r), ?_,
      (Matrix.trace C).re, (RCLike.nonneg_iff.mp hC.trace_nonneg).1, ?_⟩
    · exact (e.symm.sum_comp (fun a => star (R a)*R a)).trans K.recovery_complete
    · intro rho hrho
      change (∑ r, R (e.symm r)*noise E rho*star (R (e.symm r))) = _
      rw [e.symm.sum_comp (fun a => R a*noise E rho*star (R a))]
      have hreal : ((Matrix.trace C).re : ℂ)=Matrix.trace C := by
        exact Complex.ext rfl (by simpa using (RCLike.nonneg_iff.mp hC.trace_nonneg).2.symm)
      rw [hreal]
      exact (K.diagonal.syndromes.decoder_eq_kraus (noise E rho)).symm.trans
        (K.recovery_noise_scaled rho hrho)
```

The example module checks illustrative finite cases, including singular Gram matrices and Gram matrices that are not diagonal. Those examples help expose conventions, but the general theorem rests on spectral reduction, normalized syndrome recovery, necessity through the rank-one argument, and the bridge from density matrices to operators. The resulting theorem is the full criterion for finite complex matrices under its stated hypotheses. The earlier construction for orthogonal syndromes was one stage of this proof.

7 Effective resolvents, Euclidean transfer, and finite bounds

7.1 A resolvent identity needs two invertibility hypotheses

Split the state index into a retained type P and an eliminated type Q. Let the full operator have blocks $A_{pp}, A_{pq}, A_{qp}, A_{qq}$. The eliminated resolvent block is $D = z1_Q - A_{qq}$, and the Schur complement is

$$S(z) = z1_P - A_{pp} - A_{pq}D^{-1}A_{qp}.$$

The theorem identifies the retained corner of the full inverse with $S(z)^{-1}$. The two assumptions supplied through bracketed type classes assert invertibility of D and invertibility of the full resolvent matrix.

Source: `formalization/SchurFeshbach.lean`.

Exact lines 27 to 44. Excerpt `feshbach`. Hash in Section 12.

```
theorem projected_resolvent (App : Matrix P P ℂ) (Apq : Matrix P Q ℂ)
    (Aqp : Matrix Q P ℂ) (Aqq : Matrix Q Q ℂ) (z : ℂ)
    [Invertible (Matrix.scalar Q z-Aqq)]
    [Invertible (Matrix.scalar (P ⊕ Q) z-Matrix.fromBlocks App Apq Aqp Aqq)] :
    ((Matrix.scalar (P ⊕ Q) z-Matrix.fromBlocks App Apq Aqp Aqq)⁻¹).submatrix
      Sum.inl Sum.inl = (schur App Apq Aqp Aqq z)⁻¹ := by
  let D := Matrix.scalar Q z-Aqq
  let A := Matrix.scalar P z-App
  have he := resolvent_blocks App Apq Aqp Aqq z
  let : Invertible (Matrix.fromBlocks A (-Apq) (-Aqp) D) := Invertible.copy ‹_› _ he.symm
  let : Invertible (A-(-Apq)*⅟D*(-Aqp)) := Matrix.invertibleOfFromBlocks₂₂Invertible A (-Apq)
   ↪  (-Aqp) D
  have hinv := Matrix.invOf_fromBlocks₂₂_eq A (-Apq) (-Aqp) D
  have hs : A-(-Apq)*⅟D*(-Aqp)=schur App Apq Aqp Aqq z := by
    simp only [Matrix.neg_mul,Matrix.mul_neg,neg_neg,Matrix.invOf_eq_nonsing_inv,schur,D,A]
  rw [he,← Matrix.invOf_eq_nonsing_inv,hinv]
  change ⅟(A-(-Apq)*⅟D*(-Aqp))=(schur App Apq Aqp Aqq z)⁻¹
  rw [Matrix.invOf_eq_nonsing_inv]
  congr 1
```

Contract. Both finite types of state indices have decidable equality. The eliminated block and the full resolvent matrix are invertible. Lean's matrix inverse notation is a total operation, but that notation alone does not supply genuine invertibility at a singular matrix. The theorem uses the invertibility assumptions to justify the actual inverse identities.

Proof reading. First rewrite $z1 - A$ as the displayed block matrix, retaining the negative signs on the two off-diagonal blocks. Copy the given full invertibility instance to that rewritten matrix. A Mathlib theorem for block inverses supplies invertibility of its Schur complement and the full inverse formula. Rewrite the inverse supplied by the invertibility instance as the nonsingular matrix inverse, then select the retained submatrix. The two negative off-diagonal factors multiply to the sign already encoded in the definition of `schur`.

The source retains a concrete counterexample to dropping full invertibility: at $z = 0$ with $H = \mathrm{diag}(0, -1)$, the eliminated resolvent block is 1, while the full matrix $z1 - H = \mathrm{diag}(0, 1)$ is singular.

Source: formalization/SchurFeshbach.lean.

Exact lines 59 to 64. Excerpt feshbach_counterexample. Hash in Section 12.

```
theorem eliminated_invertible_full_singular :
    IsUnit (1 : ℂ) ∧ ¬ IsUnit (!![0, 0; 0, 1] : Matrix (Fin 2) (Fin 2) ℂ) := by
  constructor
  · exact isUnit_one
  · rw [Matrix.isUnit_iff_isUnit_det]
    norm_num [Matrix.det_fin_two]
```

The companion schur_invertible theorem says that, when the eliminated block is invertible, full invertibility is equivalent to invertibility of the Schur complement. This locates the remaining pole rather than assuming it absent. The corrected assumption is part of the checked interpretation of the traditional result. This source reuses the retained Chapter VI proof. The source correspondence is recorded separately.

7.2 Euclidean transfer is defined by the exponential series

The transfer is

$$T_e(t) = \exp(-t(H - e1)).$$

The definition uses the actual matrix exponential. For Hermitian H, a theorem from continuous functional calculus identifies it with the scalar function $x \mapsto e^{-t(x-e)}$ applied to H.

Source: formalization/EuclideanTransfer.lean.

Exact lines 39 to 47. Excerpt euclidean_cfc. Hash in Section 12.

```
theorem transfer_cfc (H : Matrix n n ℂ) (hH : H.IsHermitian) (e t : ℝ) :
    transfer H e t = cfc (fun x : ℝ => Real.exp ((-t)*(x-e))) H := by
  rw [cfc_comp' Real.exp (fun x : ℝ => (-t)*(x-e)) H]
  have hlin : cfc (fun x : ℝ => (-t)*(x-e)) H = (-t) • (H - e • (1 : Matrix n n ℂ)) := by
    rw [cfc_const_mul (-t) (fun x : ℝ => x-e) H,
      cfc_sub (fun x : ℝ => x) (fun _ : ℝ => e) H, cfc_id' ℝ H hH.isSelfAdjoint, cfc_const e H
      ↪ hH.isSelfAdjoint]
    simp only [Algebra.algebraMap_eq_smul_one]
  rw [hlin, CFC.real_exp_eq_normedSpace_exp (shifted_hermitian H hH e t).isSelfAdjoint]
  rfl
```

Proof reading. The proof composes the functional calculus for the exponential with the affine scalar function $-t(x - e)$. It evaluates that affine function through the calculus as $-t(H - e1)$. A theorem relating real continuous functional calculus to the exponential in a normed space identifies the final object with the original power series. The spectral conclusions therefore apply to the actual transfer, not an independently postulated spectral surrogate.

The shift e_0 is the minimum of the actual Hermitian eigenvalues. It is defined only when the finite state index is nonempty, because an empty spectrum has no minimum element. The source proves both membership of e_0 in the spectrum and the assertion that it is a lower bound. Positive definiteness of $T_e(t)$ follows from strict positivity of the scalar exponential for every real t. Semigroup composition follows because the two scaled generators commute.

Source: formalization/EuclideanTransfer.lean.

Exact lines 89 to 96. Excerpt euclidean_endpoint. Hash in Section 12.

```
theorem finite_euclidean_transfer [Nonempty n] (H : Matrix n n ℂ)
    (hH : H.IsHermitian) (t s : ℝ) (ht : 0 ≤ t) :
    (transfer H (groundEnergy hH) t).PosDef ∧
    transfer H (groundEnergy hH) (t+s) =
      transfer H (groundEnergy hH) t * transfer H (groundEnergy hH) s ∧
    spectrum ℝ (transfer H (groundEnergy hH) t) ⊆ Set.Ioc 0 1 := by
  exact ⟨transfer_posDef H hH _ _, transfer_add H _ _ _,
    transfer_spectrum_interval H hH _ _ ht (ground_energy_le hH)⟩
```

Contract. The endpoint assumes a nonempty finite state index, Hermitian H and nonnegative t. It includes $t = 0$. The other time s is arbitrary real in the semigroup identity. Spectral containment in $(0, 1]$ uses $t \geq 0$ and the shift by the ground energy. A separate endpoint states containment for the complex spectrum as the real interval's complex image. No infinite-dimensional transfer or continuum limit is concluded.

7.3 The low-mode bound keeps its exact constant

The dispersion expression for the finite circle is $\lambda_h(w) = 4h^{-2}\sin^2(wh/2)$. The source first proves

$$0 \leq x^2 - \sin^2 x \leq \frac{x^4}{3}$$

for every real x, using the quadratic upper bound for sine squared, the cubic error bound for $x - \sin x$ and the estimate $|x + \sin x| \leq 2|x|$. Substituting $x = wh/2$ yields

$$0 \leq w^2 - \lambda_h(w) \leq \frac{w^4}{12}h^2.$$

Source: formalization/LowModeConvergence.lean.

Exact lines 32 to 45. Excerpt low_mode. Hash in Section 12.

```
theorem low_mode_error (w h : ℝ) (hh : h≠0) :
    0 ≤ w^2-eigenvalue w h ∧ w^2-eigenvalue w h ≤ w^4/12*h^2 := by
  have hs := sine_square_error (w*h/2)
  have hpos : 0 ≤ 4/h^2 := by positivity
  have he : w^2-eigenvalue w h=4/h^2*((w*h/2)^2-Real.sin (w*h/2)^2) := by
    unfold eigenvalue
    field_simp
    ring
  rw [he]
  constructor
  · exact mul_nonneg hpos hs.1
  · calc
      _ ≤ 4/h^2*((w*h/2)^4/3) := mul_le_mul_of_nonneg_left hs.2 hpos
      _ = w^4/12*h^2 := by field_simp; ring
```

Proof reading. The proof scales the sine inequality by the nonnegative quantity $4/h^2$. A field calculation identifies the scaled error with $w^2 - \lambda_h(w)$. The final algebra simplifies its upper bound to the coefficient $1/12$. The hypothesis that the spacing is nonzero justifies division. The conclusion holds for every real w, without a restriction to small frequencies.

For a fixed frequency, the right-hand side tends to zero quadratically as h tends to zero through nonzero values. A growing frequency range needs control of w^4h^2. The theorem by itself does not prove convergence of an entire spectrum, a plateau in spectral dimension, a dynamics limit or a spacetime reconstruction. Those applications require their own quantified range and analytic arguments.

7.4 A compressed anomaly can vanish for purely algebraic reasons

The constraint audit sits beside the 19 finite ledger items. If every constraint annihilates P on the right, then every product term in the proposed anomaly also annihilates P on the right, so $P\mathcal{A}P = 0$ for arbitrary structure coefficients. The source needs no projector axioms for this observation.

Source: `formalization/ConstraintAudit.lean`.

Exact lines 31 to 34. Excerpt `constraint_kernel`. Hash in Section 12.

```
theorem compressed_anomaly_zero (C : I → R) (f : I → R)
    (P : R) (hP : ∀ c, C c * P = 0) (a b : I) :
    P * anomaly C f a b * P = 0 := by
  rw [mul_assoc, anomaly_annihilates_common_kernel C f P hP, mul_zero]
```

It also constructs a nonzero physical projector and two rational 3×3 constraints that act on its complement. Their commutator is nonzero, yet its physical compression is zero.

Source: `formalization/ConstraintAudit.lean`.

Exact lines 56 to 60. Excerpt `constraint_nonclosure`. Hash in Section 12.

```
theorem commutator_nonzero : constraintX * constraintZ - constraintZ * constraintX ≠ 0 := by
  intro h
  have h12 := congrArg (fun A : Matrix (Fin 3) (Fin 3) ℚ => A 1 2) h
  norm_num [constraintX, constraintZ, Matrix.mul_apply, Fin.sum_univ_succ] at h12
  norm_num [Matrix.cons_val_two] at h12
```

This supplies a concrete scope warning: exact compression to a common kernel cannot by itself certify operator closure on the whole space. It does not prove that an intended continuum constraint algebra exists or that every refinement has the same behavior. The hypotheses of the finite audit must be checked before applying it to a proposed physical construction.

7.5 The broader manuscript and the five open targets

Parts I and II discuss memory, geometry, diagnostics and physical interpretation beyond the named finite endpoints. A traditional proof can be more explanatory than a formal theorem statement, and a strengthened mathematical statement can go beyond the selected formal endpoint. Coverage is assigned by the full result map, not by similarity of wording or proximity on a page.

The original manuscript names five continuum conjectures: refinement without anomalies, uniform Feshbach control, Lorentzian reconstruction, an Einstein infrared limit and a massless sector with spin

two. None is a theorem target among the 19 ledger items. Finite path reconstruction, finite recovery, a projected connection, a resolvent identity and a dispersion error bound do not jointly prove those existence and limiting claims. The formal source provides rigorous finite ingredients and, in the geometric case, a broad calculus on parameter manifolds under its stated assumptions.

The complete theorem catalog includes seven declarations from the constraint audit, four Schur–Feshbach declarations and the other supporting lemmas. The count 323 therefore means named declarations in the current finite development, not 323 independently stated claims in the traditional paper, and not 323 continuum results.

9 Complete map of finite results

The following map reproduces all 19 ledger scope statements verbatim and resolves their current formal endpoints. Their recorded completion is supported by the saved compiler record under the stated hypotheses. It does not assert that every sentence in the traditional paper is separately formalized. Source pointers added where the ledger gives only a module are identified explicitly.

9.1 2.1 / Orbit dimension of one quantum seed

Preserved ledger scope. Actual matrix multiplication orbit, explicit equivalence with independently chosen rows, d times rank(C), and full-rank cyclicity when d is nonempty. No orbit-dimension hypothesis is assumed.

Formal endpoints.

- `KomorebiPaperVII.SeedOrbit.span_orbit_dimension`, `SeedOrbit.lean`, line 68.

9.2 2.2 / Quantum seed-rank formula

Preserved ledger scope. Actual sums of k matrix actions, full-column-rank seed construction, simultaneous finite direct-sum generation, and an IsLeast proof for max_i ceil(m_i/d_i). Every d_i is positive. Empty block families use supremum 0.

Formal endpoints.

- `KomorebiPaperVII.SeedOrbit.quantum_seed_rank`, `SeedOrbit.lean`, line 239.

9.3 2.3 / Local orbit and Schmidt rank

Preserved ledger scope. Actual Kronecker local action in standard product coordinates. Schmidt rank is independently defined as minimum product-vector decomposition length, with its equality to matrix rank proved. Derives local orbit dimension and equivalence with full support of the actual B-reduced matrix; the full-orbit equivalence assumes a nonempty A index.

Formal endpoints.

- `KomorebiPaperVII.SeedOrbit.local_orbit_reduced_support`, `SeedOrbit.lean`, line 364.

9.4 2.4 / One seed for the computational basis

Preserved ledger scope. Explicit local Pauli X addressing for every n, including n=0. The constructed gate word sends the zero seed to the standard coordinate basis, with linear independence, span and cardinality 2^n.

Formal endpoints.

- `KomorebiPaperVII.ComputationalSeed.circuit_seed_basis`, `ComputationalSeed.lean`, line 80.

9.5 3.1 / Cyclic quantum projectors

Preserved ledger scope. Constructs the primitive complex character and rho(r)=S^val(r) from S^N=1 for every N>0, including N=1. Proves exact exponential-root coefficient formula, character orthogonality, multiplication, completeness, eigenaction, and self-adjointness for unitary S. Includes a concrete complex-matrix theorem.

Formal endpoints.

- `KomorebiPaperVII.CyclicProjectors.finite_cyclic_projectors`, `CyclicProjectors.lean`, line 157.

9.6 3.2 / Tensor/polyphase refinement

Preserved ledger scope. Arbitrary finite tensor factors and dimensions. Proves tensor product multiplication, identity, adjoint and sum laws, then orthogonality, completeness and self-adjointness after grouping by total residue for N>0.

Formal endpoints.

- `KomorebiPaperVII.TensorRefinement.tensor_polyphase_refinement`, `TensorRefinement.lean`, line 139.

9.7 4.1 / Exact circuit path expansion

Preserved ledger scope. Arbitrary finite gate list, corner expansion and complete reconstruction.

Source pointers added by the guide.

- `KomorebiPaperVII.PacketFamily.circuit_path_expansion`, `QuantumPacketCore.lean`, line 63.
- `KomorebiPaperVII.PacketFamily.circuit_reconstruction`, `QuantumPacketCore.lean`, line 70.

9.8 4.2 / Packetwise unitarity conservation

Preserved ledger scope. Both identities in a star ring, plus concrete complex-matrix specializations.

Source pointers added by the guide.

- `KomorebiPaperVII.PacketFamily.unitarity_columns`, `QuantumPacketCore.lean`, line 119.
- `KomorebiPaperVII.PacketFamily.unitarity_rows`, `QuantumPacketCore.lean`, line 134.
- `KomorebiPaperVII.matrix_unitarity_columns`, `QuantumPacketCore.lean`, line 181.
- `KomorebiPaperVII.matrix_unitarity_rows`, `QuantumPacketCore.lean`, line 188.

9.9 5.1 / Exact symmetry packets

Preserved ledger scope. Generator blocks and actual exponential; zero cross-packet transfer.

Source pointers added by the guide.

- `KomorebiPaperVII.PacketFamily.symmetry_exponential_no_transfer`, `QuantumPacketCore.lean`, line 168.
- `KomorebiPaperVII.PacketFamily.symmetry_exponential_diagonal`, `QuantumPacketCore.lean`, line 163.

9.10 5.2 / Commutator leakage bound

Preserved ledger scope. Actual exponential with self-adjoint H and orthogonal P. Proves the Hilbert operator-norm bound for all real times, without nonzero-rank or nonempty-matrix assumptions. Also proves observable-conjugation Lipschitz continuity, the cap at one, and a symmetry-breaking perturbation estimate.

Formal endpoints.

- `KomorebiPaperVII.Leakage.matrix_commutator_leakage_bound`, `LeakageBound.lean`, line 167.

9.11 6.1 / Moving-frame Hamiltonian

Preserved ledger scope. Entrywise actual real derivatives of finite complex matrices and vectors. Constructs y=U*psi and its derivative, verifies the transformed Schrodinger equation and Hermiticity. The effective-Hamiltonian Hermiticity conclusion explicitly assumes H*=H.

Formal endpoints.

- `KomorebiPaperVII.MovingFrames.moving_frame_coordinates`, `MovingFrames.lean`, line 110.

9.12 6.3 / Canonical packet connection

Preserved ledger scope. Actual derivatives in a normed real star algebra with continuous star. Differentiates projector multiplication, completeness and self-adjointness to prove both formulas for C, skew-adjointness and [C,P_b]=dot(P_b). Does not include manifold curvature.

Formal endpoints.

- `KomorebiPaperVII.Connection.canonical_connection_of_hasDerivAt`, `CanonicalConnection.lean`, line 113.

9.13 6.4 / Projected connection

Preserved ledger scope. Smooth real boundaryless parameter manifolds modeled on a complete normed space; complete real normed fibers, including finite quantum spaces through underlying real calculus. Constructs the actual range total space, open local product homeomorphisms, smooth fiber-linear transitions, inverse and cocycle laws, and locally constant rank. Proves local smooth-section equivalence, the actual global covariant derivative, connection laws and gauge gluing. Derives curvature using actual manifold derivatives and the actual Lie bracket, and proves tensoriality. Uses an explicit smooth atlas rather than a packaged Mathlib ContMDiffVectorBundle instance. No global chart, pre-existing range-bundle atlas or curvature identity is assumed.

Formal endpoints.

- `KomorebiPaperVII.ManifoldCurvature.projected_connection_curvature`, `ManifoldProjectedCurvature.lean`, line 214.
- `KomorebiPaperVII.RangeBundle.smooth_range_bundle_atlas`, `GlobalRangeBundle.lean`, line 241.
- `KomorebiPaperVII.BundleCompletion.localProduct_overlap`, `ProjectedBundleCompletion.lean`, line 31.
- `KomorebiPaperVII.BundleCompletion.smooth_section_iff`, `ProjectedBundleCompletion.lean`, line 40.

- `KomorebiPaperVII.GlobalConnection.covariant`, `GlobalProjectedConnection.lean`, line 44.
- `KomorebiPaperVII.GlobalConnection.local_connections_glue`, `GlobalProjectedConnection.lean`, line 115.
- `KomorebiPaperVII.BundleCompletion.local_connections_gauge`, `ProjectedBundleCompletion.lean`, line 57.

9.14 7.1 / Block-diagonal measurement invariance

Preserved ledger scope. Trace duality, measurement invariance and trace preservation for finite complex matrices.

Source pointers added by the guide.

- `KomorebiPaperVII.measurement_invariance`, `QuantumPacketCore.lean`, line 208.
- `KomorebiPaperVII.dephase_trace_preserving`, `QuantumPacketCore.lean`, line 213.

9.15 7.2 / Finite Kraus channel

Preserved ledger scope. Finite complex matrix Kraus map: linearity, positivity, trace preservation under sum(K* K)=I, identity-tensor amplification at every finite auxiliary dimension, complete positivity and density-matrix preservation.

Formal endpoints.

- `KomorebiPaperVII.Kraus.channel_completely_positive`, `KrausChannels.lean`, line 58.

9.16 8.1 / Finite Knill-Laflamme criterion

Preserved ledger scope. Full finite complex-matrix Knill-Laflamme criterion. Constructs a total completely positive trace-preserving recovery from general positive semidefinite Gram data using actual spectral diagonalization, positive-eigenvalue support and proof that null errors vanish. Derives necessity from actual recovery via rank-one Kraus rigidity. Proves equivalence for normalized code density matrices, preservation with arbitrary finite references, the unnormalized error-family criterion with an explicit nonnegative scale, and correction of every normalized channel in the same error span by the same recovery. Zero code is handled separately. This is a mathematical, noncomputable spectral construction, not a numerical decoder implementation.

Formal endpoints.

- `KomorebiPaperVII.KnillLaflamme.state_correctable_iff_gram`, `QuantumStateBridge.lean`, line 139.
- `KomorebiPaperVII.KnillLaflamme.error_family_correctable_iff_gram`, `ErrorSpanRecovery.lean`, line 188.
- `KomorebiPaperVII.KnillLaflamme.Data.recovery_span_channel`, `ErrorSpanRecovery.lean`, line 138.
- `KomorebiPaperVII.KnillLaflamme.recovery_with_reference`, `QuantumStateBridge.lean`, line 164.

9.17 9.1 / Exact Feshbach resolvent

Preserved ledger scope. Corrected: eliminated block AND full resolvent matrix invertible. Paper VI proof reused.

Source pointers added by the guide.

- `KomorebiPaperVII.SchurFeshbach.projected_resolvent`, `SchurFeshbach.lean`, line 27.
- `KomorebiPaperVII.SchurFeshbach.schur_invertible`, `SchurFeshbach.lean`, line 46.

9.18 10.1 / Finite Euclidean transfer

Preserved ledger scope. Actual power-series matrix exponential, identified with continuous functional calculus. Constructs the minimum eigenvalue on a nonempty finite state space, proves positive definiteness, semigroup composition, and real and complex spectrum containment in (0,1] for nonnegative time.

Formal endpoints.

- `KomorebiPaperVII.EuclideanTransfer.finite_euclidean_transfer`, `EuclideanTransfer.lean`, line 89.

9.19 11.1 / Low-mode second-order bound

Preserved ledger scope. Explicit h != 0; stronger all-real-frequency bound with constant 1/12. Paper VI proof reused.

Source pointers added by the guide.

- `KomorebiPaperVII.LowModeConvergence.low_mode_error`, `LowModeConvergence.lean`, line 32.

10 All named theorem declarations

The following catalog is derived from active declarations in the 25 current source modules, with nested namespaces resolved. It is reconciled against the saved theorem inventory and all 323 saved axiom reports. It includes supporting lemmas, the finite constraint audit, and strengthened endpoints. The count therefore describes formal declarations rather than 323 separate manuscript theorems. The source line identifies the declaration. "Core" marks an endpoint in the map of finite results, while "support" marks a declaration used in the formal development or its scope audit. All saved axiom sets lie within `propext`, `Classical.choice`, and `Quot.sound`. The complete sets accompany the guide with their source identities.

10.1 CanonicalConnection: 9 theorem declarations

Source: `formalization/CanonicalConnection.lean`.

Named declaration	Source line
`KomorebiPaperVII.Connection.canonical_right`	18, support
`KomorebiPaperVII.Connection.canonical_left_relation`	23, support
`KomorebiPaperVII.Connection.canonical_commutator`	39, support
`KomorebiPaperVII.Connection.canonical_skewAdjoint`	48, support
`KomorebiPaperVII.Connection.canonical_half_commutator`	59, support
`KomorebiPaperVII.Connection.differentiated_orthogonality`	78, support
`KomorebiPaperVII.Connection.differentiated_completeness`	98, support
`KomorebiPaperVII.Connection.differentiated_selfAdjoint`	106, support
`KomorebiPaperVII.Connection.canonical_connection_of_hasDerivAt`	113, Core 6.3

10.2 ComputationalSeed: 14 theorem declarations

Source: `formalization/ComputationalSeed.lean`.

Named declaration	Source line
`KomorebiPaperVII.ComputationalSeed.translation_zero`	24, support
`KomorebiPaperVII.ComputationalSeed.translation_add`	27, support
`KomorebiPaperVII.ComputationalSeed.translation_ket`	31, support
`KomorebiPaperVII.ComputationalSeed.translation_involution`	35, support
`KomorebiPaperVII.ComputationalSeed.localX_on_ket`	40, support
`KomorebiPaperVII.ComputationalSeed.controlledX_eq_translation`	51, support
`KomorebiPaperVII.ComputationalSeed.translation_list_product`	63, support
`KomorebiPaperVII.ComputationalSeed.sum_local_addresses`	68, support
`KomorebiPaperVII.ComputationalSeed.circuit_eq_translation`	72, support
`KomorebiPaperVII.ComputationalSeed.circuit_seed`	78, support
`KomorebiPaperVII.ComputationalSeed.circuit_seed_basis`	80, Core 2.4
`KomorebiPaperVII.ComputationalSeed.addressed_seed_linearIndependent`	83, support
`KomorebiPaperVII.ComputationalSeed.addressed_seed_spans`	87, support
`KomorebiPaperVII.ComputationalSeed.number_of_addresses`	91, support

10.3 ConstraintAudit: 7 theorem declarations

Source: `formalization/ConstraintAudit.lean`.

Named declaration	Source line
`KomorebiPaperVII.ConstraintAudit.anomaly_annihilates_common_kernel`	26, support
`KomorebiPaperVII.ConstraintAudit.compressed_anomaly_zero`	31, support
`KomorebiPaperVII.ConstraintAudit.physical_idempotent`	43, support
`KomorebiPaperVII.ConstraintAudit.physical_nonzero`	46, support
`KomorebiPaperVII.ConstraintAudit.constraints_annihilate_physical`	51, support
`KomorebiPaperVII.ConstraintAudit.commutator_nonzero`	56, support
`KomorebiPaperVII.ConstraintAudit.physical_commutator_zero`	62, support

10.4 CyclicProjectors: 18 theorem declarations

Source: `formalization/CyclicProjectors.lean`.

Named declaration	Source line
`KomorebiPaperVII.CyclicProjectors.character_sum`	22, support
`KomorebiPaperVII.CyclicProjectors.packet_eigen`	28, support
`KomorebiPaperVII.CyclicProjectors.packet_mul`	44, support
`KomorebiPaperVII.CyclicProjectors.packet_idempotent`	60, support
`KomorebiPaperVII.CyclicProjectors.packet_complete`	65, support
`KomorebiPaperVII.CyclicProjectors.powers_zero`	76, support
`KomorebiPaperVII.CyclicProjectors.powers_add`	79, support
`KomorebiPaperVII.CyclicProjectors.powers_one`	84, support
`KomorebiPaperVII.CyclicProjectors.cyclic_quantum_projectors`	87, support
`KomorebiPaperVII.CyclicProjectors.star_character`	97, support
`KomorebiPaperVII.CyclicProjectors.packet_selfAdjoint`	104, support
`KomorebiPaperVII.CyclicProjectors.powers_star`	116, support
`KomorebiPaperVII.CyclicProjectors.cyclic_packet_selfAdjoint`	127, support
`KomorebiPaperVII.CyclicProjectors.omega_exponential`	136, support
`KomorebiPaperVII.CyclicProjectors.character_as_power`	140, support
`KomorebiPaperVII.CyclicProjectors.packet_coefficient`	143, support
`KomorebiPaperVII.CyclicProjectors.cyclic_projector_formula`	151, support
`KomorebiPaperVII.CyclicProjectors.finite_cyclic_projectors`	157, Core 3.1

10.5 ErrorSpanRecovery: 9 theorem declarations

Source: `formalization/ErrorSpanRecovery.lean`.

Named declaration	Source line
`KomorebiPaperVII.KnillLaflamme.scaled_corrected_kraus_family`	23, support
`KomorebiPaperVII.KnillLaflamme.compressed_gram_from_scalar`	48, support
`KomorebiPaperVII.KnillLaflamme.noise_preserves_trace`	65, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_scalar_composites`	79, support

Named declaration	Source line
KomorebiPaperVII.KnillLaflamme.Data.recovery_cross_error	94, support
KomorebiPaperVII.KnillLaflamme.Data.recovery_error_superposition	112, support
KomorebiPaperVII.KnillLaflamme.Data.recovery_span_channel	138, Core 8.1
KomorebiPaperVII.KnillLaflamme.gram_of_scaled_recovery	175, support
KomorebiPaperVII.KnillLaflamme.error_family_correctable_iff_gram	188, Core 8.1

10.6 EuclideanTransfer: 12 theorem declarations

Source: formalization/EuclideanTransfer.lean.

Named declaration	Source line
KomorebiPaperVII.EuclideanTransfer.shifted_hermitian	23, support
KomorebiPaperVII.EuclideanTransfer.transfer_hermitian	27, support
KomorebiPaperVII.EuclideanTransfer.transfer_zero	30, support
KomorebiPaperVII.EuclideanTransfer.transfer_add	33, support
KomorebiPaperVII.EuclideanTransfer.transfer_cfc	39, support
KomorebiPaperVII.EuclideanTransfer.transfer_spectrum	49, support
KomorebiPaperVII.EuclideanTransfer.transfer_posDef	54, support
KomorebiPaperVII.EuclideanTransfer.transfer_spectrum_interval	65, support
KomorebiPaperVII.EuclideanTransfer.ground_energy_le	77, support
KomorebiPaperVII.EuclideanTransfer.ground_energy_mem	83, support
KomorebiPaperVII.EuclideanTransfer.finite_euclidean_transfer	89, Core 10.1
KomorebiPaperVII.EuclideanTransfer.transfer_complex_spectrum	99, support

10.7 GlobalProjectedConnection: 10 theorem declarations

Source: formalization/GlobalProjectedConnection.lean.

Named declaration	Source line
KomorebiPaperVII.GlobalConnection.nabla_in_range	48, support
KomorebiPaperVII.GlobalConnection.nabla_add_sections	53, support
KomorebiPaperVII.GlobalConnection.nabla_leibniz	59, support
KomorebiPaperVII.GlobalConnection.nabla_add_vectors	65, support
KomorebiPaperVII.GlobalConnection.nabla_smul_vector	69, support
KomorebiPaperVII.GlobalConnection.nabla_germ	73, support
KomorebiPaperVII.GlobalConnection.smooth_directional	81, support
KomorebiPaperVII.GlobalConnection.smooth_nabla	88, support
KomorebiPaperVII.GlobalConnection.localNabla_represents_global	102, support
KomorebiPaperVII.GlobalConnection.local_connections_glue	115, Core 6.4

10.8 GlobalRangeBundle: 22 theorem declarations

Source: `formalization/GlobalRangeBundle.lean`.

Named declaration	Source line
`KomorebiPaperVII.RangeBundle.mem_fiber`	30, support
`KomorebiPaperVII.RangeBundle.project_coe`	36, support
`KomorebiPaperVII.RangeBundle.project_fiber`	39, support
`KomorebiPaperVII.RangeBundle.transport_self`	45, support
`KomorebiPaperVII.RangeBundle.transport_intertwines`	49, support
`KomorebiPaperVII.RangeBundle.transport_inverse_intertwines`	56, support
`KomorebiPaperVII.RangeBundle.mem_domain`	73, support
`KomorebiPaperVII.RangeBundle.isOpen_domain`	76, support
`KomorebiPaperVII.RangeBundle.fromCoord_range`	90, support
`KomorebiPaperVII.RangeBundle.toCoord_fromCoord`	97, support
`KomorebiPaperVII.RangeBundle.fromCoord_toCoord`	105, support
`KomorebiPaperVII.RangeBundle.transition_self`	121, support
`KomorebiPaperVII.RangeBundle.transition_cocycle`	127, support
`KomorebiPaperVII.RangeBundle.smooth_transport`	139, support
`KomorebiPaperVII.RangeBundle.smooth_inverse_transport`	144, support
`KomorebiPaperVII.RangeBundle.smooth_toCoord`	154, support
`KomorebiPaperVII.RangeBundle.smooth_fromCoord`	159, support
`KomorebiPaperVII.RangeBundle.smooth_transition`	164, support
`KomorebiPaperVII.RangeBundle.localProduct_base`	212, support
`KomorebiPaperVII.RangeBundle.localProduct_transition`	217, support
`KomorebiPaperVII.RangeBundle.fiber_rank_locally_constant`	234, support
`KomorebiPaperVII.RangeBundle.smooth_range_bundle_atlas`	241, Core 6.4

10.9 KnillLaflammeExamples: 8 theorem declarations

Source: `formalization/KnillLaflammeExamples.lean`.

Named declaration	Source line
`KomorebiPaperVII.KnillLaflamme.Examples.redundant_complete`	25, support
`KomorebiPaperVII.KnillLaflamme.Examples.redundant_gram_positive`	32, support
`KomorebiPaperVII.KnillLaflamme.Examples.redundant_gram_singular`	36, support
`KomorebiPaperVII.KnillLaflamme.Examples.redundant_gram_off_diagonal`	41, support
`KomorebiPaperVII.KnillLaflamme.Examples.redundant_correctable`	45, support
`KomorebiPaperVII.KnillLaflamme.Examples.dephasing_complete`	57, support
`KomorebiPaperVII.KnillLaflamme.Examples.dephasing_not_correctable`	64, support
`KomorebiPaperVII.KnillLaflamme.Examples.zero_code_correctable`	74, support

10.10 KnillLaflammeNecessity: 7 theorem declarations

Source: `formalization/KnillLaflammeNecessity.lean`.

Named declaration	Source line
`KomorebiPaperVII.KnillLaflamme.rank_one_gram_rigidity`	19, support
`KomorebiPaperVII.KnillLaflamme.kraus_rank_one_rigidity`	52, support
`KomorebiPaperVII.KnillLaflamme.corrected_kraus_family`	68, support
`KomorebiPaperVII.KnillLaflamme.gram_of_recovery`	86, support
`KomorebiPaperVII.KnillLaflamme.correctable_iff_gram`	115, support
`KomorebiPaperVII.KnillLaflamme.not_correctable_of_nonscalar`	133, support
`KomorebiPaperVII.KnillLaflamme.recovery_cross_from_scalar`	142, support

10.11 KnillLaflammeSufficiency: 22 theorem declarations

Source: `formalization/KnillLaflammeSufficiency.lean`.

Named declaration	Source line
`KomorebiPaperVII.KnillLaflamme.mix_restricted`	28, support
`KomorebiPaperVII.KnillLaflamme.mix_gram`	33, support
`KomorebiPaperVII.KnillLaflamme.mix_noise`	45, support
`KomorebiPaperVII.KnillLaflamme.diagonalized_gram`	74, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.zero_weight_error`	107, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.sqrt_weight_ne_zero`	118, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.sqrt_weight_sq`	121, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.normalized_restricted`	125, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.normalized_gram`	128, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.error_eq_sqrt_normalized`	152, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.decode_diagonal_error`	156, support
`KomorebiPaperVII.KnillLaflamme.DiagonalErrors.decode_noise`	175, support
`KomorebiPaperVII.KnillLaflamme.Data.restricted_error_gram`	198, support
`KomorebiPaperVII.KnillLaflamme.Data.restricted_noise`	216, support
`KomorebiPaperVII.KnillLaflamme.Data.diagonal_noise`	227, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_noise_scaled`	232, support
`KomorebiPaperVII.KnillLaflamme.Data.gram_trace_one`	238, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_corrects`	252, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_complete`	262, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_preserves_trace`	267, support
`KomorebiPaperVII.KnillLaflamme.Data.recovery_completely_positive`	271, support
`KomorebiPaperVII.KnillLaflamme.Data.exists_recovery`	278, support

10.12 KrausChannels: 7 theorem declarations

Source: `formalization/KrausChannels.lean`.

Named declaration	Source line
`KomorebiPaperVII.Kraus.channel_add`	16, support
`KomorebiPaperVII.Kraus.channel_smul`	20, support
`KomorebiPaperVII.Kraus.channel_positive`	29, support
`KomorebiPaperVII.Kraus.channel_trace_preserving`	33, support
`KomorebiPaperVII.Kraus.amplification_eq_kraus`	48, support
`KomorebiPaperVII.Kraus.channel_completely_positive`	58, Core 7.2
`KomorebiPaperVII.Kraus.channel_density_matrix`	64, support

10.13 LeakageBound: 17 theorem declarations

Source: `formalization/LeakageBound.lean`.

Named declaration	Source line
`KomorebiPaperVII.Leakage.exp_unitary`	25, support
`KomorebiPaperVII.Leakage.transported_derivative`	31, support
`KomorebiPaperVII.Leakage.transported_derivative_norm`	40, support
`KomorebiPaperVII.Leakage.transported_lipschitz`	46, support
`KomorebiPaperVII.Leakage.exponential_commutator_bound`	56, support
`KomorebiPaperVII.Leakage.leakage_corner`	67, support
`KomorebiPaperVII.Leakage.leakage_corner_norm`	75, support
`KomorebiPaperVII.Leakage.skew_generator_leakage_bound`	88, support
`KomorebiPaperVII.Leakage.generator_skew`	96, support
`KomorebiPaperVII.Leakage.generator_commutator_norm`	100, support
`KomorebiPaperVII.Leakage.commutator_leakage_bound`	106, support
`KomorebiPaperVII.Leakage.exact_symmetry_zero_leakage`	114, support
`KomorebiPaperVII.Leakage.unitary_corner_norm`	122, support
`KomorebiPaperVII.Leakage.capped_commutator_leakage_bound`	133, support
`KomorebiPaperVII.Leakage.projection_commutator_norm_le`	141, support
`KomorebiPaperVII.Leakage.perturbation_leakage_bound`	151, support
`KomorebiPaperVII.Leakage.matrix_commutator_leakage_bound`	167, Core 5.2

10.14 LocalProjectedCurvature: 6 theorem declarations

Source: `formalization/LocalProjectedCurvature.lean`.

Named declaration	Source line
`KomorebiPaperVII.LocalCurvature.projector_jet`	25, support
`KomorebiPaperVII.LocalCurvature.projector_differential_sandwich`	33, support
`KomorebiPaperVII.LocalCurvature.section_jet`	38, support
`KomorebiPaperVII.LocalCurvature.nabla_derivative`	47, support
`KomorebiPaperVII.LocalCurvature.curvature_expansion`	60, support

Named declaration	Source line
KomorebiPaperVII.LocalCurvature.projected_connection_curvature	79, support

10.15 LowModeConvergence: 2 theorem declarations

Source: formalization/LowModeConvergence.lean.

Named declaration	Source line
KomorebiPaperVII.LowModeConvergence.sine_square_error	15, support
KomorebiPaperVII.LowModeConvergence.low_mode_error	32, Core 11.1

10.16 ManifoldProjectedCurvature: 17 theorem declarations

Source: formalization/ManifoldProjectedCurvature.lean.

Named declaration	Source line
KomorebiPaperVII.ManifoldCurvature.derivative_in_chart	28, support
KomorebiPaperVII.ManifoldCurvature.differentiable_in_chart	34, support
KomorebiPaperVII.ManifoldCurvature.smooth_in_chart	39, support
KomorebiPaperVII.ManifoldCurvature.derivative_clm_apply	49, support
KomorebiPaperVII.ManifoldCurvature.chartField_self	65, support
KomorebiPaperVII.ManifoldCurvature.chartField_differentiable	74, support
KomorebiPaperVII.ManifoldCurvature.bracket_in_chart	80, support
KomorebiPaperVII.ManifoldCurvature.directional_chart_pullback	91, support
KomorebiPaperVII.ManifoldCurvature.second_directional_in_chart	115, support
KomorebiPaperVII.ManifoldCurvature.derivative_lieBracket	132, support
KomorebiPaperVII.ManifoldCurvature.projector_jet	150, support
KomorebiPaperVII.ManifoldCurvature.projector_differential_sandwich	161, support
KomorebiPaperVII.ManifoldCurvature.section_jet	167, support
KomorebiPaperVII.ManifoldCurvature.nabla_derivative	178, support
KomorebiPaperVII.ManifoldCurvature.curvature_expansion	191, support
KomorebiPaperVII.ManifoldCurvature.projected_connection_curvature	214, Core 6.4
KomorebiPaperVII.ManifoldCurvature.curvature_is_tensorial	236, support

10.17 MovingFrames: 9 theorem declarations

Source: formalization/MovingFrames.lean.

Named declaration	Source line
KomorebiPaperVII.MovingFrames.matrix_product_derivative	15, support
KomorebiPaperVII.MovingFrames.matrix_vector_derivative	23, support
KomorebiPaperVII.MovingFrames.adjoint_derivative	32, support
KomorebiPaperVII.MovingFrames.differentiated_unitarity	37, support
KomorebiPaperVII.MovingFrames.frame_velocity_skew	52, support

Named declaration	Source line
KomorebiPaperVII.MovingFrames.connection_hermitian	60, support
KomorebiPaperVII.MovingFrames.effective_hamiltonian_hermitian	72, support
KomorebiPaperVII.MovingFrames.moving_frame_equation	83, support
KomorebiPaperVII.MovingFrames.moving_frame_coordinates	110, Core 6.1

10.18 ProjectedBundleCompletion: 12 theorem declarations

Source: formalization/ProjectedBundleCompletion.lean.

Named declaration	Source line
KomorebiPaperVII.BundleCompletion.total_domain_open	25, support
KomorebiPaperVII.BundleCompletion.localProduct_overlap	31, Core 6.4
KomorebiPaperVII.BundleCompletion.smooth_section_iff	40, Core 6.4
KomorebiPaperVII.BundleCompletion.local_connections_gauge	57, Core 6.4
KomorebiPaperVII.BundleCompletion.curvatureForm_self	73, support
KomorebiPaperVII.BundleCompletion.curvatureForm_swap	76, support
KomorebiPaperVII.BundleCompletion.curvatureForm_add_left	81, support
KomorebiPaperVII.BundleCompletion.curvatureForm_smul_left	86, support
KomorebiPaperVII.BundleCompletion.curvatureForm_in_range	90, support
KomorebiPaperVII.BundleCompletion.fiberCurvature_coe	103, support
KomorebiPaperVII.BundleCompletion.constant_projector_flat	108, support
KomorebiPaperVII.BundleCompletion.smooth_section_chart_independent	112, support

10.19 ProjectedCurvature: 20 theorem declarations

Source: formalization/ProjectedCurvature.lean.

Named declaration	Source line
KomorebiPaperVII.Curvature.jet_sandwich_zero	22, support
KomorebiPaperVII.Curvature.nabla_in_range	47, support
KomorebiPaperVII.Curvature.projector_jet	52, support
KomorebiPaperVII.Curvature.projector_differential_sandwich	60, support
KomorebiPaperVII.Curvature.section_jet	65, support
KomorebiPaperVII.Curvature.nabla_derivative	74, support
KomorebiPaperVII.Curvature.curvature_expansion	87, support
KomorebiPaperVII.Curvature.projected_connection_curvature	106, support
KomorebiPaperVII.Curvature.curvatureForm_alternating	123, support
KomorebiPaperVII.Curvature.curvatureForm_antisymmetric	126, support
KomorebiPaperVII.Curvature.curvatureForm_left_range	131, support
KomorebiPaperVII.Curvature.curvatureForm_right_range	136, support
KomorebiPaperVII.Curvature.curvatureForm_pullback	142, support
KomorebiPaperVII.Curvature.nabla_scalar_leibniz	149, support
KomorebiPaperVII.Curvature.nabla_add_vectorfields	157, support
KomorebiPaperVII.Curvature.nabla_smul_vectorfield	162, support
KomorebiPaperVII.Curvature.curvatureForm_add_left	167, support

Named declaration	Source line
KomorebiPaperVII.Curvature.curvatureForm_add_right	172, support
KomorebiPaperVII.Curvature.curvatureForm_smul_left	177, support
KomorebiPaperVII.Curvature.curvatureForm_smul_right	181, support

10.20 QuantumPacketCore: 19 theorem declarations

Source: formalization/QuantumPacketCore.lean.

Named declaration	Source line
KomorebiPaperVII.PacketFamily.idempotent	32, support
KomorebiPaperVII.PacketFamily.reconstruct	38, support
KomorebiPaperVII.PacketFamily.compose_blocks	41, support
KomorebiPaperVII.PacketFamily.circuit_path_expansion	63, Core 4.1
KomorebiPaperVII.PacketFamily.circuit_reconstruction	70, Core 4.1
KomorebiPaperVII.PacketFamily.symmetry_no_transfer	76, support
KomorebiPaperVII.PacketFamily.symmetry_diagonal	80, support
KomorebiPaperVII.PacketFamily.dephase_idempotent	90, support
KomorebiPaperVII.PacketFamily.adjoint_block	114, support
KomorebiPaperVII.PacketFamily.unitarity_columns	119, Core 4.2
KomorebiPaperVII.PacketFamily.unitarity_rows	134, Core 4.2
KomorebiPaperVII.PacketFamily.symmetry_exponential_commutes	156, support
KomorebiPaperVII.PacketFamily.symmetry_exponential_diagonal	163, Core 5.1
KomorebiPaperVII.PacketFamily.symmetry_exponential_no_transfer	168, Core 5.1
KomorebiPaperVII.matrix_unitarity_columns	181, Core 4.2
KomorebiPaperVII.matrix_unitarity_rows	188, Core 4.2
KomorebiPaperVII.dephase_trace_duality	195, support
KomorebiPaperVII.measurement_invariance	208, Core 7.1
KomorebiPaperVII.dephase_trace_preserving	213, Core 7.1

10.21 QuantumStateBridge: 9 theorem declarations

Source: formalization/QuantumStateBridge.lean.

Named declaration	Source line
KomorebiPaperVII.KnillLaflamme.outer_polarization	21, support
KomorebiPaperVII.KnillLaflamme.linearMap_eq_of_pure	30, support
KomorebiPaperVII.KnillLaflamme.sandwich_pure	64, support
KomorebiPaperVII.KnillLaflamme.fixes_positive_of_density	70, support
KomorebiPaperVII.KnillLaflamme.fixes_operators_of_density	100, support
KomorebiPaperVII.KnillLaflamme.density_corrects_iff_operators	119, support
KomorebiPaperVII.KnillLaflamme.state_correctable_iff_gram	139, Core 8.1
KomorebiPaperVII.KnillLaflamme.amplifiedNoise_block	157, support
KomorebiPaperVII.KnillLaflamme.recovery_with_reference	164, Core 8.1

10.22 SchurFeshbach: 4 theorem declarations

Source: `formalization/SchurFeshbach.lean`.

Named declaration	Source line
`KomorebiPaperVII.SchurFeshbach.resolvent_blocks`	20, support
`KomorebiPaperVII.SchurFeshbach.projected_resolvent`	27, Core 9.1
`KomorebiPaperVII.SchurFeshbach.schur_invertible`	46, Core 9.1
`KomorebiPaperVII.SchurFeshbach.eliminated_invertible_full_singular`	59, support

10.23 SeedOrbit: 34 theorem declarations

Source: `formalization/SeedOrbit.lean`.

Named declaration	Source line
`KomorebiPaperVII.SeedOrbit.mem_orbit_iff`	31, support
`KomorebiPaperVII.SeedOrbit.mem_orbit_iff_rows`	35, support
`KomorebiPaperVII.SeedOrbit.row_dimension`	54, support
`KomorebiPaperVII.SeedOrbit.orbit_dimension`	58, support
`KomorebiPaperVII.SeedOrbit.orbit_span`	63, support
`KomorebiPaperVII.SeedOrbit.span_orbit_dimension`	68, Core 2.1
`KomorebiPaperVII.SeedOrbit.cyclic_iff_full_rank`	73, support
`KomorebiPaperVII.SeedOrbit.cyclic_multiplicity_bound`	82, support
`KomorebiPaperVII.SeedOrbit.rowOrbit_mem`	93, support
`KomorebiPaperVII.SeedOrbit.rowOrbit_dimension`	112, support
`KomorebiPaperVII.SeedOrbit.rowOrbit_full`	117, support
`KomorebiPaperVII.SeedOrbit.stacked_action`	130, support
`KomorebiPaperVII.SeedOrbit.generates_iff_rowOrbit`	135, support
`KomorebiPaperVII.SeedOrbit.generates_iff_rank`	148, support
`KomorebiPaperVII.SeedOrbit.seed_count_lower`	152, support
`KomorebiPaperVII.SeedOrbit.exists_full_column_rank`	158, support
`KomorebiPaperVII.SeedOrbit.exists_generating_seeds_iff`	173, support
`KomorebiPaperVII.SeedOrbit.block_generates_iff`	195, support
`KomorebiPaperVII.SeedOrbit.exists_block_seeds_iff`	210, support
`KomorebiPaperVII.SeedOrbit.seed_formula_le_iff`	229, support
`KomorebiPaperVII.SeedOrbit.quantum_seed_rank`	239, Core 2.2
`KomorebiPaperVII.SeedOrbit.local_tensor_action`	261, support
`KomorebiPaperVII.SeedOrbit.product_decomposition_rank_bound`	272, support
`KomorebiPaperVII.SeedOrbit.product_decomposition_at_rank`	289, support
`KomorebiPaperVII.SeedOrbit.product_length_minimum`	298, support
`KomorebiPaperVII.SeedOrbit.schmidt_rank_eq_matrix_rank`	308, support
`KomorebiPaperVII.SeedOrbit.local_orbit_is_tensor_span`	319, support
`KomorebiPaperVII.SeedOrbit.local_orbit_schmidt_dimension`	328, support
`KomorebiPaperVII.SeedOrbit.partial_trace_gram`	336, support
`KomorebiPaperVII.SeedOrbit.reduced_rank_eq_schmidt`	343, support
`KomorebiPaperVII.SeedOrbit.reduced_positive`	348, support
`KomorebiPaperVII.SeedOrbit.reduced_full_support_iff`	352, support
`KomorebiPaperVII.SeedOrbit.local_orbit_full_iff`	357, support

Named declaration	Source line
KomorebiPaperVII.SeedOrbit.local_orbit_reduced_support	364, Core 2.3

10.24 SyndromeRecovery: 18 theorem declarations

Source: formalization/SyndromeRecovery.lean.

Named declaration	Source line
KomorebiPaperVII.Recovery.OrthogonalSyndromes.packet_selfAdjoint	37, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.packet_mul	41, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.total_projection	53, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.complement_projection	61, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.total_error	64, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.complement_error	68, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.error_adjoint_complement	71, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.recoveryKraus_complete	84, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_eq_kraus	92, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_add	97, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_cross_error	103, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_smul	129, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_weighted_noise	138, support
KomorebiPaperVII.Recovery.OrthogonalSyndromes.decoder_error_superposition	149, support
KomorebiPaperVII.Recovery.recovery_positive	180, support
KomorebiPaperVII.Recovery.recovery_trace_preserving	186, support
KomorebiPaperVII.Recovery.recovery_complete_positivity	197, support
KomorebiPaperVII.Recovery.recovery_density_matrix	212, support

10.25 TensorRefinement: 11 theorem declarations

Source: formalization/TensorRefinement.lean.

Named declaration	Source line
KomorebiPaperVII.TensorRefinement.tensor_mul	20, support
KomorebiPaperVII.TensorRefinement.tensor_one	30, support
KomorebiPaperVII.TensorRefinement.tensor_sum	42, support
KomorebiPaperVII.TensorRefinement.tensor_adjoint	54, support
KomorebiPaperVII.TensorRefinement.fine_mul	67, support
KomorebiPaperVII.TensorRefinement.fine_complete	86, support
KomorebiPaperVII.TensorRefinement.fine_selfAdjoint	94, support
KomorebiPaperVII.TensorRefinement.coarse_mul	108, support
KomorebiPaperVII.TensorRefinement.coarse_complete	126, support
KomorebiPaperVII.TensorRefinement.coarse_selfAdjoint	133, support
KomorebiPaperVII.TensorRefinement.tensor_polyphase_refinement	139, Core 3.2

11 Module dependencies, source identities, and pins

11.1 Current local module graph

The dependencies below are read from public imports and commands of the form import all. Duplicate imports count once. Importing all declarations matters in this Lean source because downstream proofs can require private implementation details. A module with no local import still imports the pinned Mathlib files named in its source.

Module	Direct current local imports
`CanonicalConnection`	Mathlib only
`ComputationalSeed`	Mathlib only
`ConstraintAudit`	Mathlib only
`CyclicProjectors`	Mathlib only
`ErrorSpanRecovery`	`KnillLaflammeNecessity`, `KnillLaflammeSufficiency`, `QuantumStateBridge`, `SyndromeRecovery`
`EuclideanTransfer`	Mathlib only
`GlobalProjectedConnection`	`GlobalRangeBundle`, `LocalProjectedCurvature`, `ProjectedCurvature`
`GlobalRangeBundle`	Mathlib only
`KnillLaflammeExamples`	`ErrorSpanRecovery`, `KnillLaflammeNecessity`, `KnillLaflammeSufficiency`, `QuantumStateBridge`
`KnillLaflammeNecessity`	`KnillLaflammeSufficiency`
`KnillLaflammeSufficiency`	`SyndromeRecovery`
`KrausChannels`	Mathlib only
`LeakageBound`	Mathlib only
`LocalProjectedCurvature`	`ProjectedCurvature`
`LowModeConvergence`	Mathlib only
`ManifoldProjectedCurvature`	`GlobalProjectedConnection`, `ProjectedCurvature`
`MovingFrames`	Mathlib only
`ProjectedBundleCompletion`	`ManifoldProjectedCurvature`
`ProjectedCurvature`	Mathlib only
`QuantumPacketCore`	Mathlib only
`QuantumStateBridge`	`KnillLaflammeNecessity`, `KnillLaflammeSufficiency`
`SchurFeshbach`	Mathlib only
`SeedOrbit`	Mathlib only
`SyndromeRecovery`	Mathlib only
`TensorRefinement`	Mathlib only

11.2 Current source SHA-256 identities

CanonicalConnection

`f2b433bdd432e18c9cb7af45403a1adcdd0426f5e9115d098317744e4b0fc3ed`

ComputationalSeed

`6b92535db98ff199d4b5fd3307c2c3935d5fe6c769761eae9fe4af71bb64c6c4`

ConstraintAudit

`e00fe71eb73e8ed8a652839d4f927e282784001d6c454f8cb2fe66f866de0371`

CyclicProjectors

f0f88121acd4fc2f270de7e0ad7ca8d9e273945987a869935df670248e5d206b

ErrorSpanRecovery

f8a02ecda26d3a7d55378c06599597ada186b801750af09be0bbb2c3ae9191ed

EuclideanTransfer

5a259e9245571bf84048a3ff2f7d332cff29c8c2c9b3fcac515dceb43f594bad

GlobalProjectedConnection

ec618cd64d17d1bca4dee9ea0fae3c7f943c3347f4b1c8be063f3dd8517e0beb

GlobalRangeBundle

cbd9b611d6160a91a76332cad47d246f89cd82f02d38854089bf7f4eaa988b33

KnillLaflammeExamples

52f6ea5ccfc664ba41150c026015bec8a99d9eeb82880206b26456dee4abae05

KnillLaflammeNecessity

ff7b077fc8d55474c3a3b8ad6ae1017f4082e16dd9282ca07ecc5240f759c65d

KnillLaflammeSufficiency

3b71fbe77e55d8f603580678ada7ce4fd6303e8e1f1f5e028afb68c6bfc2e017

KrausChannels

915ebc9806b5f47ec8f958a72bfafe02723c5dcc47fceccdade1bcd3579e078f

LeakageBound

859f23f52f1274d485c2496828938cf48a8b5bf5a574264a7f04107b5b089887

LocalProjectedCurvature

6f59e61c3b60fca127a7cf9fdce369e26b421e2f20843c263201019cdf2af1c0

LowModeConvergence

7e077b24ccdfb970b349171d55551f39127a4556b2e2d247c47f6a4bc0774595

ManifoldProjectedCurvature

9803d99d975303179ccecbbfae1133bcab47496cee452cf3e130f3c4870a1e12

MovingFrames

8961ac05b66911ad82c84bc49ade3cac6eac2a84728898cc86a8286599f0ddef

ProjectedBundleCompletion

29afe46959d4538fc2833de8e014faf8be2203f811654796847d1b7597868d77

ProjectedCurvature

ad7b8cd20af860c3f455874c0009b3d18f40a5fe937531cd6798007406316686

QuantumPacketCore

92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595

QuantumStateBridge

b3a76f530fb11075d5494f45a94d102c9084125e291c2bcd9ffa5210f6e8bdb2

SchurFeshbach

2ce60c9300b85ff06064d296cdf773f7cf9d2e0eafb7acbda15973fe851bb479

12 Exact excerpt register

Each excerpt is a contiguous block of the unchanged source, retaining its full declaration and proof. The printed code uses explicit font fallback only where the monospaced font lacks a glyph. The corresponding `.lean.txt` file preserves raw UTF-8, without editorial substitutions. The source and excerpt hashes below bind the printed selection to its original context. The assumptions declared in a module remain in force when an excerpt is read. The teaching chapters state the relevant type classes and hypotheses.

row_equiv

Source: `formalization/SeedOrbit.lean`.

Lines 45 to 52. Declaration `orbitRowEquiv`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `a4d1d840bdcebbbffc42ae57380b479bc57d2afe07de79b1fe9e34bdda086c28`.

orbit_dimension

Source: `formalization/SeedOrbit.lean`.

Lines 68 to 71. Declaration `span_orbit_dimension`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `365ce586b2f58c4a3574bdc7af540baa1094dd028e25cd712a1334bb6782eb24`.

cyclic_rank

Source: `formalization/SeedOrbit.lean`.

Lines 73 to 80. Declaration `cyclic_iff_full_rank`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `f82dfbcc44084af989f282f53c4380f8b99bd21b2ced9722c556a65551ab7fdc`.

seed_generate

Source: `formalization/SeedOrbit.lean`.

Lines 127 to 128. Declaration `Generates`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `f5f2e4255c8ba1ace030b5315a2b6be0db3fc24fe4beda857449e94a5b41b6de`.

stack_rank

Source: `formalization/SeedOrbit.lean`.

Lines 148 to 150. Declaration `generates_iff_rank`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `f490fd86582b597c2a514ab1effe39cab73d59a96ca7a6b9923d52eb8c02702c`.

full_column

Source: `formalization/SeedOrbit.lean`.

Lines 158 to 171. Declaration `exists_full_column_rank`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `63d1eecf433f1ddc3dbfc779101bca490c43fcc6709bfa193f2c7e5ee31ee615`.

seed_minimum

Source: `formalization/SeedOrbit.lean`.

Lines 239 to 246. Declaration `quantum_seed_rank`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `89ee9cb7e0961e27954fe763e202dd0c3f44282f2b5fa3cc2abbf7bc71b316c5`.

schmidt_definition

Source: `formalization/SeedOrbit.lean`.

Lines 303 to 306. Declaration `schmidtRank`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `1279ef8ea57e64959c0d7d67193ccbb143617830b9007b93e5aae3a6de863f42`.

schmidt_minimum

Source: `formalization/SeedOrbit.lean`.

Lines 298 to 300. Declaration `product_length_minimum`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `435e38531c7e19906f0ce32c7a69346a9990726ce7d9e697e8b6443d9f95942c`.

local_action

Source: `formalization/SeedOrbit.lean`.

Lines 261 to 265. Declaration `local_tensor_action`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `8130668bea5e0e3ab38489f1a45a4a7b7d0593996ab788615ef7bbc36ed52ab8`.

reduced_support

Source: `formalization/SeedOrbit.lean`.

Lines 364 to 366. Declaration `local_orbit_reduced_support`.

Source SHA-256: `0b1da8c7e74ccc7bbca8e2b7c0c26bce01bc95d4089616ba95c1e4fe2f147773`.

Excerpt SHA-256: `147a14439c7983ad43f62238f014efeebb80b403a15160f3c8a64c23af1bdc6d`.

seed_basis

Source: formalization/ComputationalSeed.lean.

Lines 80 to 82. Declaration circuit_seed_basis.

Source SHA-256: 6b92535db98ff199d4b5fd3307c2c3935d5fe6c769761eae9fe4af71bb64c6c4.

Excerpt SHA-256: e228b9b252604336965a67975ccbebbe2c014dadc8da35b9e38f9457fad20162.

seed_translation

Source: formalization/ComputationalSeed.lean.

Lines 72 to 77. Declaration circuit_eq_translation.

Source SHA-256: 6b92535db98ff199d4b5fd3307c2c3935d5fe6c769761eae9fe4af71bb64c6c4.

Excerpt SHA-256: 42c2842952870bb25e00b99c1d316542ccb39edcc35b9bf5282899c130919b8c.

cyclic_character

Source: formalization/CyclicProjectors.lean.

Lines 22 to 26. Declaration character_sum.

Source SHA-256: f0f88121acd4fc2f270de7e0ad7ca8d9e273945987a869935df670248e5d206b.

Excerpt SHA-256: 6182ca8cbfb31865af273e0554bd4027e66efedb0195c6e5548ff6041df9fbad.

cyclic_mul

Source: formalization/CyclicProjectors.lean.

Lines 44 to 58. Declaration packet_mul.

Source SHA-256: f0f88121acd4fc2f270de7e0ad7ca8d9e273945987a869935df670248e5d206b.

Excerpt SHA-256: 5d5eb96a4f93624594bcc443e22a1fd3b50078f6219ca12787f9492f83e149f5.

cyclic_matrix

Source: formalization/CyclicProjectors.lean.

Lines 157 to 170. Declaration finite_cyclic_projectors.

Source SHA-256: f0f88121acd4fc2f270de7e0ad7ca8d9e273945987a869935df670248e5d206b.

Excerpt SHA-256: 314f68975d510c4ac2dbf381e55d004877797f7e011199f51b0743c4231fe886.

tensor_mul

Source: formalization/TensorRefinement.lean.

Lines 20 to 27. Declaration tensor_mul.

Source SHA-256: b0587b32f55d50125790345695b777786d23e3f46141e7056049d3ef5a11e0f2.

Excerpt SHA-256: 840d1c4953c639cdfb51054f3e6524dd7e5e58c9f56328e253b43677ef4950d9.

tensor_refinement

Source: formalization/TensorRefinement.lean.

Lines 139 to 154. Declaration tensor_polyphase_refinement.
Source SHA-256: b0587b32f55d50125790345695b777786d23e3f46141e7056049d3ef5a11e0f2.
Excerpt SHA-256: 2f58f8936bffee8ed36834f4618bcb5de3ef8454cffcd992cf9bd226fe21bb07.

packet_structure

Source: formalization/QuantumPacketCore.lean.

Lines 22 to 25. Declaration PacketFamily.
Source SHA-256: 92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595.
Excerpt SHA-256: 94b93d351a97d973bf990a4a8d1fcd4a8d04e12ae60192ab2eaf58fd05722bde.

compose_blocks

Source: formalization/QuantumPacketCore.lean.

Lines 41 to 50. Declaration compose_blocks.
Source SHA-256: 92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595.
Excerpt SHA-256: d62cbe9447e98700c0387ade81a7695945d771ac4aad6dc2e445fb925825dec1.

circuit_path

Source: formalization/QuantumPacketCore.lean.

Lines 63 to 68. Declaration circuit_path_expansion.
Source SHA-256: 92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595.
Excerpt SHA-256: 8a7447d90ccb095429646dbfcb867b148a3807779708cc2e65b183c3691fc338.

unitarity_columns

Source: formalization/QuantumPacketCore.lean.

Lines 119 to 131. Declaration unitarity_columns.
Source SHA-256: 92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595.
Excerpt SHA-256: 54be9086567756fa2f7a0f2136a49ed2b27eb5289098ad7171c6e4db4ca7b8ba.

symmetry_exp

Source: formalization/QuantumPacketCore.lean.

Lines 168 to 171. Declaration symmetry_exponential_no_transfer.
Source SHA-256: 92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595.
Excerpt SHA-256: 39905e80eae8288643f906f8bce86131420f979f90a22972af30af14907e2fdb.

measurement

Source: `formalization/QuantumPacketCore.lean`.

Lines 208 to 211. Declaration `measurement_invariance`.

Source SHA-256: `92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595`.

Excerpt SHA-256: `d98f2c05feb20e8ec9edc53e090c36e404cc45977db5bf30b82abb62d4b2cad2`.

transport_derivative

Source: `formalization/LeakageBound.lean`.

Lines 31 to 38. Declaration `transported_derivative`.

Source SHA-256: `859f23f52f1274d485c2496828938cf48a8b5bf5a574264a7f04107b5b089887`.

Excerpt SHA-256: `4f6cf06917d31eef4a0cb83ee091863c72390050b998c3ec82f81ed4715b0536`.

transport_lipschitz

Source: `formalization/LeakageBound.lean`.

Lines 46 to 54. Declaration `transported_lipschitz`.

Source SHA-256: `859f23f52f1274d485c2496828938cf48a8b5bf5a574264a7f04107b5b089887`.

Excerpt SHA-256: `f7a3612f65196115dc1c6dc0b6721a1a057db6f39fedd361e43acbe872a31cf7`.

leakage_matrix

Source: `formalization/LeakageBound.lean`.

Lines 167 to 171. Declaration `matrix_commutator_leakage_bound`.

Source SHA-256: `859f23f52f1274d485c2496828938cf48a8b5bf5a574264a7f04107b5b089887`.

Excerpt SHA-256: `2da717dabf6f741c4b0d5d435b27898cd4360e6a75bd401e9b9ca08bbea787b1`.

leakage_perturbation

Source: `formalization/LeakageBound.lean`.

Lines 151 to 164. Declaration `perturbation_leakage_bound`.

Source SHA-256: `859f23f52f1274d485c2496828938cf48a8b5bf5a574264a7f04107b5b089887`.

Excerpt SHA-256: `6c29b54826e1a17a70cc88a993747995a66d98bc4de068407fe606ab14f39e84`.

frame_product

Source: `formalization/MovingFrames.lean`.

Lines 23 to 29. Declaration `matrix_vector_derivative`.

Source SHA-256: `8961ac05b66911ad82c84bc49ade3cac6eac2a84728898cc86a8286599f0ddef`.

Excerpt SHA-256: `b541e7998198257db7256977da20097d88c1093f7a302793f133b501727fb886`.

moving_coordinates

Source: formalization/MovingFrames.lean.

Lines 110 to 132. Declaration moving_frame_coordinates.

Source SHA-256: 8961ac05b66911ad82c84bc49ade3cac6eac2a84728898cc86a8286599f0ddef.

Excerpt SHA-256: a54d0ad271dcb2a4a32e0800cb13e680e61dc3f6fa699184ba31ae9c5d81b095.

connection_jet

Source: formalization/CanonicalConnection.lean.

Lines 78 to 95. Declaration differentiated_orthogonality.

Source SHA-256: f2b433bdd432e18c9cb7af45403a1adcdd0426f5e9115d098317744e4b0fc3ed.

Excerpt SHA-256: eb57f9d5f332db8c25c84c3e5a6bb54c1cd10454626ef4f3d200a3c3878d1c2e.

canonical_connection

Source: formalization/CanonicalConnection.lean.

Lines 113 to 131. Declaration canonical_connection_of_hasDerivAt.

Source SHA-256: f2b433bdd432e18c9cb7af45403a1adcdd0426f5e9115d098317744e4b0fc3ed.

Excerpt SHA-256: ee12fd06d9eaa15681457221c5d4aca3d1aaff2b2487d0b6389cedebd82bc40e.

sandwich_zero

Source: formalization/ProjectedCurvature.lean.

Lines 22 to 26. Declaration jet_sandwich_zero.

Source SHA-256: ad7b8cd20af860c3f455874c0009b3d18f40a5fe937531cd6798007406316686.

Excerpt SHA-256: 7dc40b766cb348d7f859e22f3264fe7832a0243c2573b3c72b2e0670f962ae77.

local_curvature

Source: formalization/LocalProjectedCurvature.lean.

Lines 79 to 94. Declaration projected_connection_curvature.

Source SHA-256: 6f59e61c3b60fca127a7cf9fdce369e26b421e2f20843c263201019cdf2af1c0.

Excerpt SHA-256: 536eabe9e44ef8fea3518c58157a34c7e2485e4bd4cd031aecced8c419426013.

range_transport

Source: formalization/GlobalRangeBundle.lean.

Lines 43 to 43. Declaration transport.

Source SHA-256: cbd9b611d6160a91a76332cad47d246f89cd82f02d38854089bf7f4eaa988b33.

Excerpt SHA-256: dfd3703c4861cefd45738415feb43fd6d139ae903e22fcfa54a52f3ae3166b59.

range_intertwines

Source: `formalization/GlobalRangeBundle.lean`.

Lines 49 to 54. Declaration `transport_intertwines`.

Source SHA-256: `cbd9b611d6160a91a76332cad47d246f89cd82f02d38854089bf7f4eaa988b33`.

Excerpt SHA-256: `270a82bb94c15f6815e19aacca566c805b4c02366ca5aa7f0af3a21edeff6ea7`.

range_product

Source: `formalization/GlobalRangeBundle.lean`.

Lines 180 to 209. Declaration `localProduct`.

Source SHA-256: `cbd9b611d6160a91a76332cad47d246f89cd82f02d38854089bf7f4eaa988b33`.

Excerpt SHA-256: `1f6345cfbb46c8c9cfd05c107d4c065ff4203c2f6579ab9ad3f77f677446c0a7`.

range_atlas

Source: `formalization/GlobalRangeBundle.lean`.

Lines 241 to 252. Declaration `smooth_range_bundle_atlas`.

Source SHA-256: `cbd9b611d6160a91a76332cad47d246f89cd82f02d38854089bf7f4eaa988b33`.

Excerpt SHA-256: `8e70b5873ddcf4e12a7c8f010839e666e3eed7810e9e45ac18fe8694cf61d9fd`.

global_nabla

Source: `formalization/GlobalProjectedConnection.lean`.

Lines 29 to 30. Declaration `nabla`.

Source SHA-256: `ec618cd64d17d1bca4dee9ea0fae3c7f943c3347f4b1c8be063f3dd8517e0beb`.

Excerpt SHA-256: `21fd6220b421762dd36a043194975b4cb173eb8ecba9904f1e82558148f56361`.

global_leibniz

Source: `formalization/GlobalProjectedConnection.lean`.

Lines 59 to 63. Declaration `nabla_leibniz`.

Source SHA-256: `ec618cd64d17d1bca4dee9ea0fae3c7f943c3347f4b1c8be063f3dd8517e0beb`.

Excerpt SHA-256: `7c4fed39fa233c060d4c3cb834aec746aa20118fac168fe896e2b97d1b71e20e`.

global_glue

Source: `formalization/GlobalProjectedConnection.lean`.

Lines 115 to 124. Declaration `local_connections_glue`.

Source SHA-256: `ec618cd64d17d1bca4dee9ea0fae3c7f943c3347f4b1c8be063f3dd8517e0beb`.

Excerpt SHA-256: `44e2c6deb0f447d93e739a761bebd05a7202663add66936ed4ca8b04f8140f56`.

manifold_curvature

Source: formalization/ManifoldProjectedCurvature.lean.

Lines 214 to 232. Declaration projected_connection_curvature.
Source SHA-256: 9803d99d975303179ccecbbfae1133bcab47496cee452cf3e130f3c4870a1e12.
Excerpt SHA-256: b16a27faaca85a392b119d8a58b11243610f490f85808d3f503f924be9d4b759.

smooth_section

Source: formalization/ProjectedBundleCompletion.lean.

Lines 40 to 53. Declaration smooth_section_iff.
Source SHA-256: 29afe46959d4538fc2833de8e014faf8be2203f811654796847d1b7597868d77.
Excerpt SHA-256: 4d880ef0afe4f2904521de95d6987431163c41a619192455f0d4963381a109ab.

gauge_law

Source: formalization/ProjectedBundleCompletion.lean.

Lines 57 to 70. Declaration local_connections_gauge.
Source SHA-256: 29afe46959d4538fc2833de8e014faf8be2203f811654796847d1b7597868d77.
Excerpt SHA-256: 3515991be601fc80774ba245cf2fd97b415f112cf14f1303201a2f488392eca3.

kraus_amplify

Source: formalization/KrausChannels.lean.

Lines 48 to 56. Declaration amplification_eq_kraus.
Source SHA-256: 915ebc9806b5f47ec8f958a72bfafe02723c5dcc47fceccdade1bcd3579e078f.
Excerpt SHA-256: 1323a35ef1a6247fba9aa829edaebc571ab8045b4e0362a24ca12abd66386fa6.

kraus_cp

Source: formalization/KrausChannels.lean.

Lines 58 to 62. Declaration channel_completely_positive.
Source SHA-256: 915ebc9806b5f47ec8f958a72bfafe02723c5dcc47fceccdade1bcd3579e078f.
Excerpt SHA-256: aecdfd6e1b918aac49f394badbf080146717c04ebf3d94a03f06528db544435e.

syndrome_structure

Source: formalization/SyndromeRecovery.lean.

Lines 21 to 27. Declaration OrthogonalSyndromes.
Source SHA-256: 23a176970c7adb9e995005d50722eebf953889147802f35ea9024b0081d6af84.
Excerpt SHA-256: 73f2e29a5c2b47648bf6729fa98b330f3532a0f288038c89b1388338a3042e0d.

syndrome_complete

Source: formalization/SyndromeRecovery.lean.

Lines 84 to 90. Declaration recoveryKraus_complete.

Source SHA-256: 23a176970c7adb9e995005d50722eebf953889147802f35ea9024b0081d6af84.

Excerpt SHA-256: ee0a642c16544ee89b34db86766f682bfc4789777d83f2b7d51b749b359521e7.

syndrome_cross

Source: formalization/SyndromeRecovery.lean.

Lines 103 to 124. Declaration decoder_cross_error.

Source SHA-256: 23a176970c7adb9e995005d50722eebf953889147802f35ea9024b0081d6af84.

Excerpt SHA-256: c5524fa7c2b23fb404c2cc2b7c66c4d1be519b0fc2c8189e0b6cd5cbb0aacb15.

kl_data

Source: formalization/KnillLaflammeSufficiency.lean.

Lines 187 to 193. Declaration Data.

Source SHA-256: 3b71fbe77e55d8f603580678ada7ce4fd6303e8e1f1f5e028afb68c6bfc2e017.

Excerpt SHA-256: ee0fe883d0c774bf87c6bb168c1419d60e29142fc7711f195c500c9fa12187db.

kl_diagonal

Source: formalization/KnillLaflammeSufficiency.lean.

Lines 204 to 214. Declaration diagonal.

Source SHA-256: 3b71fbe77e55d8f603580678ada7ce4fd6303e8e1f1f5e028afb68c6bfc2e017.

Excerpt SHA-256: db1af099e2586af393a0b1574fb59cc0f8464f63a93001e6a52e70d3e62db4a8.

kl_null

Source: formalization/KnillLaflammeSufficiency.lean.

Lines 107 to 110. Declaration zero_weight_error.

Source SHA-256: 3b71fbe77e55d8f603580678ada7ce4fd6303e8e1f1f5e028afb68c6bfc2e017.

Excerpt SHA-256: 062b1f1c10c1db14ed3d3fa301585f6746b2a816af2a875256e671740ef79ddc.

kl_corrects

Source: formalization/KnillLaflammeSufficiency.lean.

Lines 252 to 259. Declaration recovery_corrects.

Source SHA-256: 3b71fbe77e55d8f603580678ada7ce4fd6303e8e1f1f5e028afb68c6bfc2e017.

Excerpt SHA-256: 0edf5306e8febc4c8d0dd4bf808c41f3498f9c568687a547926941d315f33bf5.

kl_rigidity

Source: formalization/KnillLaflammeNecessity.lean.

Lines 19 to 45. Declaration rank_one_gram_rigidity.

Source SHA-256: ff7b077fc8d55474c3a3b8ad6ae1017f4082e16dd9282ca07ecc5240f759c65d.

Excerpt SHA-256: c7b1e822236c6453ea334c28938823c2af3318444fef3cf82ee5311d7310c0da.

kl_necessity

Source: formalization/KnillLaflammeNecessity.lean.

Lines 86 to 105. Declaration gram_of_recovery.

Source SHA-256: ff7b077fc8d55474c3a3b8ad6ae1017f4082e16dd9282ca07ecc5240f759c65d.

Excerpt SHA-256: b3955cdc9224b5c2af85de359794d01818749364c809e7ce3d991b9c109ad40f.

density_bridge

Source: formalization/QuantumStateBridge.lean.

Lines 119 to 129. Declaration density_corrects_iff_operators.

Source SHA-256: b3a76f530fb11075d5494f45a94d102c9084125e291c2bcd9ffa5210f6e8bdb2.

Excerpt SHA-256: ae811fac94c396d9ee2052588ea8668a2e69f5d24c3bcdd26560c4567b6fe95f.

kl_criterion

Source: formalization/QuantumStateBridge.lean.

Lines 139 to 151. Declaration state_correctable_iff_gram.

Source SHA-256: b3a76f530fb11075d5494f45a94d102c9084125e291c2bcd9ffa5210f6e8bdb2.

Excerpt SHA-256: 3e7dd0d4356f6ce599db17a3d9a4049a86fb612e0bb8254114f6f5e15df70dcb.

kl_reference

Source: formalization/QuantumStateBridge.lean.

Lines 164 to 182. Declaration recovery_with_reference.

Source SHA-256: b3a76f530fb11075d5494f45a94d102c9084125e291c2bcd9ffa5210f6e8bdb2.

Excerpt SHA-256: e6301010a1248fffde60cbd86f25278f5615775f2ef4fc02bb5a57276b510366.

kl_span

Source: formalization/ErrorSpanRecovery.lean.

Lines 138 to 165. Declaration recovery_span_channel.

Source SHA-256: f8a02ecda26d3a7d55378c06599597ada186b801750af09be0bbb2c3ae9191ed.

Excerpt SHA-256: 108c56b74be5789d672f77a69cf5d11fac6376476eda291ca96358ad2f9c5aba.

kl_family

Source: formalization/ErrorSpanRecovery.lean.

Lines 188 to 209. Declaration error_family_correctable_iff_gram.

Source SHA-256: f8a02ecda26d3a7d55378c06599597ada186b801750af09be0bbb2c3ae9191ed.

Excerpt SHA-256: 5d027f08e3b3e93ee21eda17960f631aadd79cae3e6d0e47df52bdced92e4df7.

feshbach

Source: formalization/SchurFeshbach.lean.

Lines 27 to 44. Declaration projected_resolvent.

Source SHA-256: 2ce60c9300b85ff06064d296cdf773f7cf9d2e0eafb7acbda15973fe851bb479.

Excerpt SHA-256: d84ccc733fea567da198271006256848b4afa3112df09490598de9968c00ca70.

feshbach_counterexample

Source: formalization/SchurFeshbach.lean.

Lines 59 to 64. Declaration eliminated_invertible_full_singular.

Source SHA-256: 2ce60c9300b85ff06064d296cdf773f7cf9d2e0eafb7acbda15973fe851bb479.

Excerpt SHA-256: 377aa16415eeb74cce76d96a5bc1e33ac627d32dbccdcb9465f52bb456552283.

euclidean_cfc

Source: formalization/EuclideanTransfer.lean.

Lines 39 to 47. Declaration transfer_cfc.

Source SHA-256: 5a259e9245571bf84048a3ff2f7d332cff29c8c2c9b3fcac515dceb43f594bad.

Excerpt SHA-256: 9eb0a3b8ac13e09838ab8074571b5149b2040c810140f3518039b8edb8915e61.

euclidean_endpoint

Source: formalization/EuclideanTransfer.lean.

Lines 89 to 96. Declaration finite_euclidean_transfer.

Source SHA-256: 5a259e9245571bf84048a3ff2f7d332cff29c8c2c9b3fcac515dceb43f594bad.

Excerpt SHA-256: ee9c869877cc94941a4308d727f4d422bb8d5c13864aa4c33755d0f5a1ef3d8a.

low_mode

Source: formalization/LowModeConvergence.lean.

Lines 32 to 45. Declaration low_mode_error.

Source SHA-256: 7e077b24ccdfb970b349171d55551f39127a4556b2e2d247c47f6a4bc0774595.

Excerpt SHA-256: 7de00df6c8e15233a57e9432f25072c725e03efb58d3b6e0cfa6a72d61a0b97c.

constraint_kernel

Source: formalization/ConstraintAudit.lean.

Lines 31 to 34. Declaration compressed_anomaly_zero.

Source SHA-256: e00fe71eb73e8ed8a652839d4f927e282784001d6c454f8cb2fe66f866de0371.

Excerpt SHA-256: 121148fa69ae880879db5578d139bcff070684810952dc73d45f9fe7106ce78d.

constraint_nonclosure

Source: formalization/ConstraintAudit.lean.

Lines 56 to 60. Declaration commutator_nonzero.

Source SHA-256: e00fe71eb73e8ed8a652839d4f927e282784001d6c454f8cb2fe66f866de0371.

Excerpt SHA-256: 2a18c6988e67ff03d9308f041e90dc1e3713f983dc36f07f9b1f28f91454d3bd.

Packet Operator Algorithms

The formal companion to Chapter VIII explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2133

Return to Chapter VIII

How to use this volume

This guide teaches the actual finite Lean development. It follows the represented objects, states their hypotheses, and reads representative proofs step by step. It includes 61 exact source selections, the complete 56-result correspondence, four separately tracked classical inputs, and all 458 named audit entries.

The source has 40 modules, 392 theorem declarations (373 for Chapter VIII and 19 in the reused foundation), and 66 construction audits. These counts include supporting results and specializations. They do not count independent mathematical discoveries.

Three broader research targets remain open: unrestricted discovery of profitable packet structure, compilation with nonlocal costs and overlapping decompositions, and a packet memory communication theorem for a specified architecture.

The recorded formal run contains successful cumulative compilation and axiom audits. Source identities, saved logs, declaration names and exact excerpts identify the scope of that evidence. The reproduction section supplies the toolchain and replay procedure. This reading guide does not claim a new kernel execution.

The traditional paper develops the mathematical arguments. The visual guide works through their constructions. This companion explains their formal counterparts and the exact boundary between proved conclusions and supplied data. The accompanying `README.md` gives the complete monograph regeneration sequence, including static evidence refresh and final page navigation before rebuilding.

Source base: `original_snapshot/formalization/`.

Primary inherited evidence: `original_snapshot/verification/sixth_final/`.

Raw UTF-8 excerpts and their SHA-256 register accompany the printed selections.

1 Reading a finite formal development

1.1 The objects precede the theorem names

A packet representation starts with a finite projector family and an operator. A structured representation adds explicit factor matrices or tensor coordinates. A compiler starts with a finite candidate tree and an independently defined set of legal cuts. An error theorem starts with actual errors in a normed space and budgets that bound them. In each case, the theorem follows the object being modeled. Reading only the endpoint name misses the domain, norm, convention, and supplied data on which its conclusion depends.

Use the traditional paper's original item numbers to find the corresponding subsection in Section 12. Then locate its declaration in Section 13 and its module in Section 14. For a first reading, work through the chapters in order. The small-core chapter is longer because the final block proves optimality from several independent ingredients, rather than accepting an optimality certificate as input.

1.2 A short Lean vocabulary

Source form	Meaning in this guide
`Matrix m n K`	A function with row type `m`, column type `n`, and coefficients in `K`. Finite index instances provide finite sums. Rectangular shapes are visible in the type.
`[Fintype I]`	A finite enumeration of the index type. Empty index types are allowed unless a nonempty hypothesis is separately stated.
`[DecidableEq I]`	Equality can be decided where the construction needs it. It is not a numerical tolerance test.
`Fin k`	Natural indices smaller than k. The type `Fin 0` is empty.
`[Ring R]`	Addition and multiplication are available, but multiplication need not commute.
`IsUnit D`	The matrix or algebra element has a two-sided multiplicative inverse. A displayed inverse expression alone does not establish invertibility.
`X →L[K] X`	A continuous linear map over `K`, with its operator norm.
`Option R`	Either `some v`, a finite candidate value, or `none`, an unreachable / infeasible state.
`let A := ...`	A local definition in the statement. It abbreviates the constructed object. It adds no hypothesis.
`noncomputable def`	A mathematically defined object using classical choice or other noncomputable ingredients. It does not advertise an executable numerical routine.

A theorem statement may inherit `variable` declarations from its section. An `omit ... in` command removes unused instances for that declaration. `autoImplicit false` means an undeclared symbol is rejected rather than silently turned into an implicit parameter. Names such as `SmallCore.spectralNorm` explicitly choose the intended norm. A bare matrix norm also depends on the open norm scope surrounding its declaration.

1.3 How to read a proof

`rw` rewrites by an equality. `simp` applies named simplification facts, while `simp only` restricts that collection. `funext` reduces equality of functions to equality at an arbitrary argument, while `ext` exposes entries of a structured object. `refine` supplies a proof expression with remaining holes. `have` names an intermediate proposition and its proof. `exact` closes a goal with a term of the required type.

Induction is visible in the recursive structured algorithms and dynamic programs. A `calc` chain shows the successive equalities or inequalities used in a norm estimate. Arithmetic tactics have limited roles: `ring` normalizes polynomial identities, `omega` handles integer and natural-number constraints, and `linarith`/`nlinarith` combine supplied real inequalities. They do not supply unrecorded analytic or hardware assumptions.

1.4 Which evidence is current

The inherited sixth-block report records the compiler, source hashes, local dependency hashes, saved log hashes, axiom sets, and two rejected invalid-proof controls. The present guide replays the static identity checks on those retained bytes. It does not reconstruct compiled interfaces or simulate a Lean invocation. The exact distinction is preserved in `generated/documentation_reconciliation.json`.

The allowed transitive axiom policy is `propext`, `Classical.choice`, and `Quot.sound`, or subsets of them. This is the policy accepted by the recorded run. A passing audit excludes an admitted-proof axiom under that policy. It does not independently rebuild all imported Mathlib proofs. Manuscript correspondence remains mathematical review.

2 Navigation through the traditional volume

The Printed and PDF columns locate results in the attached Chapter VIII source document. Physical PDF pages include its cover and opening material. These locations refer to that source document rather than the continuous folios of this book. Original result numbers remain the correspondence keys.

2.1 All 56 registered finite results

The page map locates each manuscript statement. The formal hypotheses in Section 12 describe its actual finite counterpart. These page locations do not expand the formal contract.

Item	Number	Statement	Printed	PDF
2.3	2.3	Exact evaluation	7	13
2.5	2.5	Block composition	8	14
2.6	2.6	Adjoint and tensor semantics	9	15
2.9	2.9	Contract conversion	9	15
3.1	3.1	Dense and packet application counts	10	16
3.2	3.2	Equal diagonal packets	10	16
3.4	3.4	BDLR application and storage	11	17
3.6	3.6	Streaming byte count	11	17
3.8	3.8	Roofline lower bound	12	18
3.9	3.9	Capacity threshold	12	18
3.10	3.10	Materialization cost	13	19
4.1	4.1	Equal block banded count	13	19
4.2	4.2	Sparse plus low rank application	14	20
4.3	4.3	Circulant spectral representation	14	20
4.4	4.4	Toeplitz circulant embedding	15	21
4.5	4.5	Kronecker modewise application	15	21
4.6	4.6	Equal tensor factors	15	21
4.7	4.7	Walsh Hadamard recursion	16	22
5.1	5.1	Exact BDLR product	16	22
5.2	5.2	Rank growth in a product chain	17	23
5.3	5.3	Woodbury inversion	18	24
5.4	5.4	Inverse norm bound	18	24
5.6	5.6	Small core recompression	18	24
5.8	5.8	Product chain perturbation	19	25
5.9	5.9	Certified repeated recompression	20	26
6.1	6.1	Path support of powers	21	27
6.2	6.2	Polynomial support	21	27
6.3	6.3	Acyclic transfer nilpotence	21	27
6.5	6.5	Level schedule for a triangular solve	22	28
6.6	6.6	Sequential depth	22	28
6.7	6.7	Inverse support lies in the transitive closure	23	29
6.8	6.8	Schur support inclusion	23	29
7.2	7.2	Invariant packet Krylov decomposition	24	30
7.3	7.3	Approximate packet Krylov bound	24	30
7.4	7.4	Block Jacobi condition bound	25	31
7.6	7.6	Certified packet preconditioning	25	31
8.1	8.1	Packet perturbation certificate	26	32
8.3	8.3	Mixed precision product chain	27	33

Item	Number	Statement	Printed	PDF
8.4	8.4	Continuous precision allocation	27	33
8.5	8.5	Clipped precision allocation	28	34
8.7	8.7	Exact discrete precision program	29	35
9.2	9.2	Exact block discovery certificate	30	36
9.4	9.4	Exact Lagrangian packet compiler	31	37
9.5	9.5	Shared compilation for an operator family	31	37
9.6	9.6	Monotone Lagrangian frontier	32	38
9.7	9.7	Supported Pareto points	32	38
9.8	9.8	Quantized budget dynamic program	33	39
9.9	9.9	Pseudo polynomial complexity	33	39
9.11	9.11	Compiler stability	33	39
9.12	9.12	Local recompilation	34	40
10.2	10.2	Launch reduction by shape	35	41
10.3	10.3	Acyclic schedule depth	35	41
10.4	10.4	Amortization threshold	35	41
10.5	10.5	Graph replay threshold	35	41
10.6	10.6	Materialization threshold	36	42
11.3	11.3	Fair matching is an equivalence relation	36	42

2.2 The four separately supplied classical inputs

These inputs are tracked separately from the 56 finite results. Their exact boundaries are discussed in the replay chapter.

Item	Number	Classical input	Printed	PDF
5.5	5.5	Eckart Young Mirsky theorem	18	24
7.5	7.5	Preconditioned conjugate gradient convergence	25	31
8.2	8.2	Floating point matrix product model	27	33
9.1	9.1	Simultaneous spectral decomposition	30	36

2.3 The open research boundary

The traditional boundary discussion is on printed page 42 (PDF page 48). It retains the three broader targets: discovery of profitable structure for arbitrary operators, controlled compilation for overlapping decompositions and nonlocal costs, and a communication theorem for a specified packet memory architecture. Their finite special cases do not settle those broader targets.

3 Exact packet semantics

3.1 The reused projector family

The unchanged `QuantumPacketCore` foundation uses a ring `R` and finite packet type `I`. The projectors multiply orthogonally and sum to one. Their block $P_b A P_a$ uses the target index first. This algebraic model already covers noncommutative operator multiplication.

Source: `formalization/QuantumPacketCore.lean`.

Exact lines 22–25. Excerpt `packet_family`. SHA-256 in Section 15.

```
structure PacketFamily (R I : Type*) [Ring R] [Fintype I] [DecidableEq I] where
  proj : I → R
  mul_proj : ∀ i j, proj i * proj j = if i = j then proj i else 0
  sum_proj : ∑ i, proj i = 1
```

Proof reading. The structure fields are assumptions defining a packet family. Idempotence is derived by setting the two indices equal, and reconstruction follows by expanding the two sums of projectors. A finite projector family is supplied to later theorems, so discovery of an appropriate family is a separate mathematical or application task.

3.2 Deleting exact zero blocks

`activeBlocks P A` filters the finite set of ordered packet pairs by the predicate that the actual block is unequal to zero. `evalActive` sums only those surviving blocks.

Source: `formalization/PacketIR.lean`.

Exact lines 41–49. Excerpt `active_evaluation`. SHA-256 in Section 15.

```
theorem exact_active_evaluation (A : R) : evalActive P A = A := by
  classical
  have term (e : I × I) :
      (if P.block A e.1 e.2 ≠ 0 then P.block A e.1 e.2 else 0) =
      P.block A e.1 e.2 := by
    by_cases h : P.block A e.1 e.2 = 0 <;> simp [h]
  simp only [evalActive, activeBlocks, Finset.sum_filter]
  simp_rw [term]
  simpa only [Fintype.sum_prod_type] using P.reconstruct A
```

Proof reading. The local lemma `term` checks two cases: a block equals zero or it does not. In either case, replacing it by an if-expression preserves the value. The proof expands the filtered sum, rewrites every term by this lemma, and applies the foundation's reconstruction theorem. No thresholded support is used.

Source: `formalization/PacketIR.lean`.

Exact lines 145–149. Excerpt `batch`. SHA-256 in Section 15.

```
theorem exact_matrix_batch {C : Type*} (P : PacketFamily (Matrix X X K) I)
    (A : Matrix X X K) (Z : Matrix X C K) :
```

```
    A * Z = ∑ e ∈ activeBlocks P A, (P.block A e.1 e.2) * Z := by
  rw [← Matrix.sum_mul]
  exact congrArg (fun a : Matrix X X K => a * Z) (exact_active_evaluation P A).symm
```

Proof reading. A rectangular right-hand-side matrix `Z` may have any column type `C`. Distribution of matrix multiplication over the finite sum turns exact operator reconstruction into equality of the whole output batch. The result concerns actual matrix multiplication, rather than a Boolean sparsity specification.

3.3 Support composition and cancellation

Source: `formalization/PacketIR.lean`.

Exact lines 63–77. Excerpt `support_inclusion`. SHA-256 in Section 15.

```
theorem product_support_inclusion (A B : R) (b a : I)
    (h : P.block (A * B) b a ≠ 0) :
    ∃ c, P.block A b c ≠ 0 ∧ P.block B c a ≠ 0 := by
  classical
  by_contra no
  apply h
  rw [P.compose_blocks]
  apply Finset.sum_eq_zero
  intro c _
  by_cases hc : P.block A b c = 0
  · simp [hc]
  · have hd : P.block B c a = 0 := by
      by_contra hd
      exact no ⟨c, hc, hd⟩
    simp [hd]
```

Proof reading. The premise says the product block is nonzero. Assume no intermediate packet has both nonzero constituent blocks. For each intermediate index, either the first block is zero or the assumption forces the second to be zero. Every summand in the composition formula vanishes, contradicting the premise. This proves an inclusion of support relations. It supplies no converse: nonzero paths can cancel. The separate `BoundaryChecks.paths_can_cancel` declaration records a concrete control.

3.4 An adjoint needs the right projector hypothesis

Source: `formalization/PacketIR.lean`.

Exact lines 95–97. Excerpt `adjoint`. SHA-256 in Section 15.

```
theorem exact_adjoint_block (A : R) (b a : I) :
    star (P.block A b a) = P.block (star A) a b :=
  P.adjoint_block hstar A b a
```

Contract. The surrounding section has a star ring and an explicit `hstar : ∀ a, star (P.proj a) = P.proj a`.

Proof reading. The source calls the foundation's adjoint identity with that self-adjointness proof. Adjoint reversal swaps target and source. Orthogonal idempotence in an arbitrary ring alone does not establish self-adjointness. Tensor semantics are separately proved with actual Kronecker products. Their address types are pairs of packet indices.

3.5 The Python prototype is a different object

The supplied Python prototype's support accessor returns stored block keys, which may include explicitly stored zeros. The formal `activeBlocks` set instead tests exact nonzero matrix blocks. The two objects agree only when the stored representation has already removed all zero blocks and realizes the same matrices. No automatic equivalence between the prototype and the Lean representation is asserted here.

4 Representation algebra and declared cost models

4.1 Why a count theorem can have a short proof

`PacketCostModels` defines finite address and event types before counting them. The block widths are natural numbers. A diagonal block has a pair of within-block indices. A thin factor has one ambient and one rank index. An application event additionally has a right-hand-side index and a Boolean distinguishing multiplication from accumulation. The Boolean gives the declared two-scalar-operation convention.

Source: `formalization/PacketCostModels.lean`.

Exact lines 68–73. Excerpt `bdlr_count`. SHA-256 in Section 15.

```
theorem bdlr_application_count {I : Type*} [Fintype I]
    (width : I → ℕ) (n r q : ℕ) :
    Fintype.card (BDLROperation I width n r q) =
      2 * q * ((∑ i, width i ^ 2) + 2 * n * r) := by
  simp [BDLROperation, pow_two]
  ring
```

Proof reading. The type `BDLROperation` is a sum of the diagonal-action, right-factor, and left-factor event types. `simp` evaluates cardinalities of finite products and sums, and `ring` rearranges the resulting polynomial. The count includes the accumulation into the diagonal result through the declared third kernel. It does not establish how many GPU instructions run.

The storage count concerns stored scalar values. Sparse index metadata and workspace are outside the corresponding models. The materialization theorem counts an explicit dense low-rank multiplication followed by one addition per output entry. Its quadratic remainder is exactly n^2. This is a count for that declared construction, rather than a lower bound for every conceivable implementation.

Source: `formalization/BandedCount.lean`.

Exact lines 73–77. Excerpt `band_count`. SHA-256 in Section 15.

```
theorem banded_block_count (p w : ℕ) (hw : w < p) :
    Fintype.card (BandedAddress p w) = p*(2*w+1)-w*(w+1) := by
  rw [bandCount_card]
  have he := bandCount_balance p w hw
  omega
```

Proof reading. The banded-count development counts the actual subtype of pairs in `Fin p × Fin p` whose distance is at most `w`. The premise `w < p` controls the two triangular boundary losses. The preceding recurrence and finite-sum lemmas establish the cardinality. This endpoint presents the closed formula. Replacing p by n/b additionally requires positive block width and exact divisibility.

4.2 Capacity inequalities are hypotheses

Source: `formalization/PacketCostModels.lean`.

Exact lines 118–124. Excerpt `roofline`. SHA-256 in Section 15.

```
theorem roofline_time_bound (F B P W T : ℝ) (hP : 0 < P) (hW : 0 < W)
    (hF : F ≤ P * T) (hB : B ≤ W * T) : max (F / P) (B / W) ≤ T := by
  apply max_le
  · apply (div_le_iff₀ hP).2
    nlinarith
  · apply (div_le_iff₀ hW).2
    nlinarith
```

Proof reading. Positive peak arithmetic and bandwidth rates make division preserve order. The assumed inequalities `F ≤ P * T` and `B ≤ W * T` yield the two lower bounds on time, then `max_le` combines them. Lean does not measure a device rate or validate that an experimental execution satisfies these capacity assumptions.

4.3 An exact low-rank product

Contract. `LowRankAlgebra` works with actual finite matrices over a field with an involution. The left and right factors have explicit possibly different width types.

Source: `formalization/LowRankAlgebra.lean`.

Exact lines 57–65. Excerpt `bdlr_product`. SHA-256 in Section 15.

```
theorem bdlr_product (DA DB : Matrix X X K) (UA VA : Matrix X RA K)
    (UB VB : Matrix X RB K) :
    (DA + UA * VA.conjTranspose) * (DB + UB * VB.conjTranspose) =
      DA * DB + bdlrProductLeft DA UA VA UB * (bdlrProductRight DB VA VB).conjTranspose := by
  simp only [bdlrProductLeft, bdlrProductRight,
    Matrix.conjTranspose_fromCols_eq_fromRows_conjTranspose,
    Matrix.fromCols_mul_fromRows, Matrix.conjTranspose_mul, Matrix.conjTranspose_conjTranspose,
    Matrix.add_mul, Matrix.mul_add, Matrix.mul_assoc]
  abel
```

Proof reading. The constructors `bdlrProductLeft` and `bdlrProductRight` concatenate columns, so the new correction width is the sum of the two input widths. The proof expands the concatenated product, reverses conjugate transposes of products, distributes multiplication, and normalizes additive terms. No rank identity is assumed in this algebraic equality. A separate theorem bounds the correction rank by its column count. Packet diagonality of the base product is also proved separately.

Source: `formalization/LowRankChains.lean`.

Exact lines 56–66. Excerpt `chain_factors`. SHA-256 in Section 15.

```
theorem chainFactors_correct (n : ℕ) :
    chainProduct (fun j => D j + U j * (V j).conjTranspose) n =
      chainProduct D n + (chainFactors r D U V n).1 * (chainFactors r D U V n).2.conjTranspose :=
      ↪ by
```

```
  induction n with
  | zero =>
    ext i j
    simp [chainProduct, chainFactors, ChainColumns]
  | succ n ih =>
    rw [chainProduct, ih]
    exact bdlr_product (D n) (chainProduct D n) (U n) (V n)
      (chainFactors r D U V n).1 (chainFactors r D U V n).2
```

Proof reading. The empty chain is the identity and has an empty correction-column type. For the successor step, rewrite the current chain using the induction hypothesis and invoke the actual two-factor identity. The source thereby constructs factors for every length. The subsequent rank theorem bounds their width by the sum of input widths, including zero widths.

4.4 Woodbury exposes both invertibility assumptions

Source: `formalization/LowRankAlgebra.lean`.

Exact lines 89–94. Excerpt `woodbury`. SHA-256 in Section 15.

```
theorem bdlr_woodbury (D : Matrix X X K) (U V : Matrix X Rk K)
    (hD : IsUnit D) (hS : IsUnit (1 + V.conjTranspose * D⁻¹ * U)) :
    (D + U * V.conjTranspose)⁻¹ =
      D⁻¹ - D⁻¹ * U * (1 + V.conjTranspose * D⁻¹ * U)⁻¹ * V.conjTranspose * D⁻¹ := by
  simpa using Matrix.add_mul_mul_inv_eq_sub D U (1 : Matrix Rk Rk K)
    V.conjTranspose hD isUnit_one (by simpa using hS)
```

Proof reading. `hD` proves the base matrix is a unit. `hS` proves the small update matrix is a unit. The endpoint specializes Mathlib's Woodbury identity, with the middle rank-space matrix equal to one. The update's invertibility, batch solve formula, and L2 operator norm bound are independent declarations in the same module. A singular base or singular update matrix cannot be passed through the displayed theorem simply because an inverse notation exists.

5 Structured algorithms carry their coordinate conventions

5.1 Vectorization and recursive tensor contraction

Source: `formalization/StructuredTensor.lean`.

Exact lines 19–22. Excerpt `kronecker`. SHA-256 in Section 15.

```
theorem kronecker_modewise {M N : Type*} [Fintype M] [Fintype N]
    (A : Matrix M M ℂ) (B : Matrix N N ℂ) (X : Matrix N M ℂ) :
    (A ⊗ₖ B) *ᵥ Matrix.vec X = Matrix.vec (B * X * Aᵀ) := by
  exact Matrix.kronecker_mulVec_vec B X A
```

Proof reading. The right factor in the matrix expression is $\mathtt{A}^{\mathrm{T}}$, the ordinary transpose. It is not the conjugate transpose. The argument order reflects Mathlib's column-major `Matrix.vec` convention: $(A \otimes B)\,\mathrm{vec}(X) = \mathrm{vec}(BXA^{T})$. The one-line proof invokes that exact library identity.

`TensorIndex d 0` is `Unit`, and the successor index is `Fin d × TensorIndex d k`. This makes the tensor dimension d^{k} visible without an implicit flattening convention. `tensorMatrix` builds iterated actual Kronecker matrices. `tensorApply` contracts trailing slices before the first mode.

Source: `formalization/StructuredTensor.lean`.

Exact lines 73–96. Excerpt `tensor_apply`. SHA-256 in Section 15.

```
theorem tensor_apply_correct (d k : ℕ) (A : Fin k → Matrix (Fin d) (Fin d) ℂ)
    (x : TensorIndex d k → ℂ) :
    tensorApply d k A x = tensorMatrix d k A *ᵥ x := by
  induction k with
  | zero =>
    funext i
    cases i
    change x PUnit.unit = ∑ i : Unit, (1 : ℂ) * x i
    simp only [one_mul]
    symm
    apply Finset.sum_eq_single PUnit.unit
    · intro b _ h
      cases b
      exact (h rfl).elim
    · intro h
      exact (h (Finset.mem_univ _)).elim
  | succ k ih =>
    funext p
    change (∑ j, A 0 p.1 j * tensorApply d k (fun i => A i.succ) (fun u => x (j,u)) p.2) = _
    simp_rw [ih]
    change (∑ j, A 0 p.1 j * (∑ u, tensorMatrix d k (fun i => A i.succ) p.2 u * x (j,u))) =
      ∑ i : Fin d × TensorIndex d k,
        (A 0 p.1 i.1 * tensorMatrix d k (fun j => A j.succ) p.2 i.2) * x i
    simp only [Fintype.sum_prod_type, Finset.mul_sum, mul_assoc]
```

Proof reading. The zero-mode case reduces to the one-coordinate identity. In the successor case, expand one contraction, rewrite trailing contractions using the induction hypothesis, and express the

full matrix sum over a product type as two nested sums. Associativity and distribution align the two formulas. The event type counted later follows this recursive contraction, yielding $2kd^{k+1}$ operations in the declared model and kd^2 factor values. Cases $k = 0$ and $d = 0$ are represented by their actual index types.

5.2 The Walsh butterfly proves an implemented recursion

Source: `formalization/WalshTransform.lean`.

Exact lines 60–84. Excerpt `walsh_apply`. SHA-256 in Section 15.

```
theorem walsh_apply_correct (k : ℕ) (x : TensorIndex 2 k → ℂ) :
    walshApply k x = walshMatrix k *ᵥ x := by
  induction k with
  | zero =>
    funext i
    cases i
    change x PUnit.unit = ∑ i : Unit, (1:ℂ)*x i
    simp only [one_mul]
    symm
    apply Finset.sum_eq_single PUnit.unit
    · intro b _ h
      cases b
      exact (h rfl).elim
    · intro h
      exact (h (Finset.mem_univ _)).elim
  | succ k ih =>
    funext p
    rcases p with ⟨i,u⟩
    change (if i = 0 then walshApply k (fun v => x (0,v)+x (1,v)) u
      else walshApply k (fun v => x (0,v)-x (1,v)) u) =
        ∑ v : Fin 2 × TensorIndex 2 k, (walshTwo i v.1 * walshMatrix k u v.2) * x v
    simp_rw [ih]
    fin_cases i <;>
      simp [walshTwo, Matrix.mulVec, dotProduct, Fintype.sum_prod_type,
        Fin.sum_univ_two, mul_add, Finset.sum_add_distrib, sub_eq_add_neg]
```

Proof reading. The recursively constructed `walshMatrix` matrix is paired with the actual sum/difference routine `walshApply`. The proof follows the same split of tensor coordinates and expands the two butterfly branches. Its equality states algorithm correctness. The separate operation theorem counts additions and subtractions, giving $N \log_2 N$ for $N = 2^k$, including $N = 1$. Orthogonality uses the ordinary transpose of these real-valued complex matrices.

5.3 Toeplitz padding is concrete

Contract. The matrix and generator definitions assume `[NeZero n]`, so the length is positive. The zero-padding helper itself is meaningful without that instance.

Source: `formalization/ToeplitzEmbedding.lean`.

Exact lines 54–58. Excerpt `toeplitz_apply`. SHA-256 in Section 15.

```
theorem toeplitz_embedding_action (c r x : Fin n → ℂ) (i : Fin n) :
    (Matrix.circulant (toeplitzGenerator c r) *ᵥ zeroPad x) (Fin.castAdd n i) =
      (toeplitzMatrix c r *ᵥ x) i := by
  simp only [Matrix.mulVec, dotProduct, Fin.sum_univ_add, zero_pad_left, zero_pad_right,
    mul_zero, Finset.sum_const_zero, add_zero, toeplitz_circulant_block]
```

Proof reading. The preceding theorem identifies the actual top-left Toeplitz block of the length-$n+n$ circulant. This application theorem expands the zero-padded vector with `Fin.addCases`. Terms in the right half vanish, leaving precisely the Toeplitz action. The generator reverses the row tail in its second half. Its first entry uses the column datum `c 0`, and `r 0` is irrelevant. No symmetry or Hermiticity is assumed.

5.4 Fourier diagonalization is exact, with no FFT timing claim

Source: `formalization/CirculantFourier.lean`.

Exact lines 150–153. Excerpt `fourier`. SHA-256 in Section 15.

```
theorem circulant_spectral_representation (c : ZMod n → ℂ) :
    Matrix.circulant c = (fourierMatrix n)ᴴ *
      Matrix.diagonal ((Real.sqrt (n:ℝ) : ℂ) • (fourierMatrix n *ᵥ c)) * fourierMatrix n := by
  rw [normalized_fourier_eigenvalues, circulant_spectral_dft]
```

Proof reading. For a positive length n, the module defines the actual circulant and the unitary-normalized Fourier matrix on `ZMod n`. The proof rewrites the DFT diagonalization using the normalization identities and actual Fourier unitarity. Supporting declarations establish the complex exponential entry formula, eigenvalues, and both unitarity products. The final application formula uses two transforms. This establishes the represented algebraic action. An FFT algorithm, its execution cost, and hardware timing remain distinct tasks.

6 Transfer paths, inverses, and acyclic solves

6.1 A path expansion covers length zero

Contract. `TransferSupport` uses a finite packet family in a ring. `TransferPath P A n b a` records a path of exactly n active edges, from source a to target b.

Source: `formalization/TransferSupport.lean`.

Exact lines 37–52. Excerpt `power_paths`. SHA-256 in Section 15.

```
theorem power_path_expansion (A : R) (n : ℕ) (b a : I) :
    P.block (A ^ n) b a = activePathSum P A n b a := by
  classical
  induction n generalizing b a with
  | zero =>
      by_cases h : b = a
      · subst b
        simp [activePathSum, PacketFamily.block, P.mul_proj]
      · simp [activePathSum, PacketFamily.block, P.mul_proj, h]
  | succ n ih =>
      rw [pow_succ', P.compose_blocks]
      simp only [activePathSum]
      apply Finset.sum_congr rfl
      intro c _
      rw [ih]
      by_cases h : P.block A b c = 0 <;> simp [h]
```

Proof reading. Induct on the exponent. At length zero, the identity block is a projector on the diagonal and zero off the diagonal. At a successor, use block composition and the induction hypothesis for each intermediate packet. A zero first edge contributes nothing, while a nonzero first edge extends a path. The result is an exact sum of path products, so cancellation still allows its support inclusion to be strict.

Source: `formalization/TransferSupport.lean`.

Exact lines 86–90. Excerpt `nilpotent`. SHA-256 in Section 15.

```
theorem nilpotent_of_path_bound (A : R) (h : ℕ)
    (hpaths : ∀ b a, ¬ TransferPath P A (h + 1) b a) : A ^ (h + 1) = 0 := by
  apply all_blocks_zero P
  intro b a
  exact power_block_zero_of_no_path P A (h + 1) b a (hpaths b a)
```

Proof reading. If no pair has a path of the stated length, every block of that power is zero. Reconstruction then forces the whole power to be zero. The level-based version derives a path bound from a supplied strict level function. It does not assume that nonzero paths always survive in the operator.

6.2 Inverse support uses finite dimensionality

Contract. The inverse-support module has a field `K`, a `K`-algebra `R`, and `FiniteDimensional K R`. It defines the subspace of elements with zero components at every packet unreachable from the starting packet. Reachability includes zero-length paths.

Source: `formalization/InverseSupport.lean`.

Exact lines 94–100. Excerpt `inverse_support`. SHA-256 in Section 15.

```
theorem inverse_block_zero_of_no_path (A B : R) (hBA : B*A = 1) (b a : I)
    (h : ¬ Reachable P A b a) : P.block B b a = 0 := by
  have hm := invariant_space_inverse (K := K) A B hBA
    (reachableSpace (K := K) P A a) (reachableSpace_invariant P A a)
    (P.proj a) (seed_mem_reachableSpace P A a)
  have hz := hm b h
  simpa only [PacketFamily.block,mul_assoc] using hz
```

Proof reading. The source first proves that the forward multiplication by `A` preserves the reachable subspace. A left inverse makes that restricted map injective, and finite dimensionality makes it surjective. The helper `invariant_space_inverse` therefore shows that multiplication by `B` preserves the same subspace. Apply it to the seed projector and read its unreachable component. The matrix specialization supplies the relevant inverse hypothesis from matrix invertibility. No inverse sparsity pattern is assumed.

6.3 Schur fill follows routes through eliminated packets

Source: `formalization/SchurSupport.lean`.

Exact lines 57–73. Excerpt `schur_support`. SHA-256 in Section 15.

```
theorem schur_support_inclusion (A : Matrix X X K) (B : Matrix X Y K)
    (H : Matrix Y Y K) (C : Matrix Y X K) (hH : IsUnit H.det) (b a : I)
    (h : P.block (A-B*H⁻¹*C) b a ≠ 0) :
    P.block A b a ≠ 0 ∨
      ∃ j k, P.proj b*B*Q.proj j ≠ 0 ∧ Reachable Q H j k ∧
        Q.proj k*C*P.proj a ≠ 0 := by
  by_cases ha : P.block A b a ≠ 0
  · exact Or.inl ha
  · right
    apply schur_correction_support P Q B H C hH b a
    intro hc
    apply h
    have hz : P.block A b a = 0 := not_ne_iff.mp ha
    change P.proj b*(A-B*H⁻¹*C)*P.proj a = 0
    rw [Matrix.mul_sub,Matrix.sub_mul]
    change P.block A b a - P.block (B*H⁻¹*C) b a = 0
    rw [hz,hc,sub_self]
```

Proof reading. The premise is a nonzero block of the actual Schur complement $A - BH^{-1}C$. If the original A-block is nonzero, the first alternative applies. Otherwise, the correction block must be

nonzero. Expand it through the eliminated packet family and use inverse support: a surviving term has an incoming block, a route through H, and an outgoing block. The determinant-unit premise establishes invertibility of H. The conclusion allows fill. It does not say every route creates a nonzero fill block.

6.4 The graph level is constructed

`DAG.Graph` contains a finite edge relation and the hypothesis that no vertex lies on a nonempty directed cycle. The source proves that the ancestor set strictly grows along an edge. Its finite cardinality gives a well-founded recursion. The resulting level is one plus the maximum predecessor level. Sources start at level one. The empty graph's maximum and depth are zero.

Source: `formalization/GraphSchedules.lean`.

Exact lines 150–154. Excerpt `schedule_depth`. SHA-256 in Section 15.

```
theorem optimal_depth (s : V → ℕ) (hs : G.ValidSchedule s) :
    G.depth ≤ Finset.univ.sup s := by
  classical
  exact Finset.sup_le (fun v _ => (G.level_minimal s hs v).trans
    (Finset.le_sup (Finset.mem_univ v)))
```

Proof reading. The earlier `level_minimal` theorem proves by well-founded induction that the constructed level is no later than any positive valid schedule at each vertex. Taking the finite maximum yields the displayed lower bound. Separate declarations prove the level schedule valid and its maximum attained by a path, so the construction is optimal in the unit-duration, unrestricted-parallelism model. Device resources, variable task durations, and launch overhead are outside that model.

6.5 Actual forward substitution in dependent block spaces

For each vertex v, `X v` is its block space. Unequal block dimensions are allowed. `D v : X v ≃ₗ[K] X v` is an invertible linear map, and `T b a` is a linear transfer from the source space to the target space. The action sums only graph edges.

Source: `formalization/TriangularSolve.lean`.

Exact lines 38–43. Excerpt `solve_correct`. SHA-256 in Section 15.

```
theorem solve_correct (rhs : ∀ v, X v) :
    G.triangularAction X D T (G.solve X D T rhs) = rhs := by
  funext v
  unfold triangularAction
  rw [G.solve_eq X D T rhs v, LinearEquiv.apply_symm_apply]
  exact sub_add_cancel _ _
```

Proof reading. The constructed `solve` uses well-founded forward substitution, applying `(D v).symm` to the right-hand side minus previously solved transfers. Function extensionality chooses one vertex. Rewrite the solve equation, cancel the inverse linear equivalence, then add back the transfer sum. Uniqueness is a separate well-founded induction over the same graph. The staged solver is separately proved correct through each level. The theorem begins with no assumed solution or solve equation.

7 Error propagation and exact precision allocation

7.1 Local block errors are actual operators

Contract. `PacketPerturbation` works over a nontrivially normed field and a normed vector space. `A e` and `Δ e` are continuous linear maps. The source-support hypothesis is `Δ e = (Δ e).comp (P (src e))`.

Source: `formalization/PacketPerturbation.lean`.

Exact lines 35–39. Excerpt `packet_error`. SHA-256 in Section 15.

```
theorem packet_perturbation_bound (P : I → X →L[K] X) (src : E → I)
    (A Δ : E → X →L[K] X) (hΔ : ∀ e, Δ e = (Δ e).comp (P (src e))) (x : X) :
    ‖((∑ e, (A e + Δ e)) - (∑ e, A e)) x‖ ≤ ∑ e, ‖Δ e‖ * ‖P (src e) x‖ := by
  rw [packet_error_action_identity P src A Δ hΔ]
  exact (norm_sum_le _ _).trans (Finset.sum_le_sum (fun e _ => (Δ e).le_opNorm _))
```

Proof reading. Rewrite the total action error as a sum of source-supported error actions. The triangle inequality bounds the norm of that sum. Each map's operator-norm estimate supplies $\|\Delta_e\|\|P_{\mathrm{src}(e)}x\|$. The exact support of the A_e terms is not required for this inequality. A later theorem assumes local relative budgets. Another adds contractive source projectors for a uniform certificate. Neither conclusion certifies IEEE arithmetic or a specific CUDA kernel.

7.2 Products preserve multiplication order

Contract. `ProductChainError` uses a normed ring. The product-chain endpoint additionally assumes `NormOneClass R`, so the empty product has norm one. Products are built by left multiplication at each successor step.

Source: `formalization/ProductChainError.lean`.

Exact lines 129–145. Excerpt `chain_error`. SHA-256 in Section 15.

```
theorem product_chain_error_bound [NormOneClass R] (A B : ℕ → R) (δ : ℕ → ℝ)
    (hδ : ∀ j, ‖B j - A j‖ ≤ δ j) (n : ℕ) :
    ‖chainProduct B n - chainProduct A n‖ ≤
      ∑ j ∈ Finset.range n,
        (∏ i ∈ Finset.Ico (j + 1) n, ‖B i‖) *
          (δ j * ∏ i ∈ Finset.range j, ‖A i‖) := by
  rw [← propagatedError_eq_sum]
  apply norm_recurrence_bound (fun k => chainProduct B k - chainProduct A k)
    (fun k => ‖B k‖) (fun k => δ k * ∏ i ∈ Finset.range k, ‖A i‖)
    (by simp [chainProduct]) (fun k => norm_nonneg _)
  intro k
  rw [product_discrepancy_step]
  apply (norm_add_le _ _).trans
  apply add_le_add (norm_mul_le _ _)
  apply (norm_mul_le _ _).trans
  exact mul_le_mul (hδ k) (chainProduct_norm_bound A k) (norm_nonneg _)
    ((norm_nonneg _).trans (hδ k))
```

Proof reading. The discrepancy at a successor splits into the perturbed factor times the previous discrepancy, plus the new local error times the exact prefix. The generic norm recurrence propagates these two contributions. Its unrolled finite sum uses perturbed suffix norms and exact prefix norms in the displayed order. Submultiplicativity and the local error bound close each successor estimate. The empty chain has zero discrepancy. No commutativity is used.

Source: `formalization/ProductChainError.lean`.

Exact lines 100–111. Excerpt `recompression_error`. SHA-256 in Section 15.

```
theorem repeated_recompression_bound (A E : ℕ → R) (δ : ℕ → ℝ)
    (hδ : ∀ j, ‖E j‖ ≤ δ j) (n : ℕ) :
    ‖recompressedProduct A E n - chainProduct A n‖ ≤
      ∑ j ∈ Finset.range n, (∏ i ∈ Finset.Ico (j + 1) n, ‖A i‖) * δ j := by
  rw [← propagatedError_eq_sum]
  apply norm_recurrence_bound
      (fun k => recompressedProduct A E k - chainProduct A k)
      (fun k => ‖A k‖) δ (by simp [chainProduct, recompressedProduct])
      (fun k => norm_nonneg _)
  intro k
  rw [recompression_discrepancy_step]
  exact (norm_add_le _ _).trans (add_le_add (norm_mul_le _ _) (hδ k))
```

Proof reading. Here each iterate has an additive residual `E j` after multiplication by `A j`. The successor error is A_j times the old error plus E_j. The same recurrence yields the product of subsequent operator norms times each certified residual budget. This theorem uses actual residuals with verified bounds. It does not itself choose the recompression algorithm or infer its floating-point error.

7.3 The continuous minimizer is global

The finite edge type is explicitly nonempty. Every sensitivity s_e, cost c_e, and budget is positive. `AllocationFeasible` requires positive tolerances and exact weighted budget equality. The objective is $\sum_e c_e \log(1/\varepsilon_e)$, and the candidate is

$$\varepsilon_e^* = \frac{\mathrm{budget}\, c_e}{s_e \sum_f c_f}.$$

Source: `formalization/ContinuousPrecision.lean`.

Exact lines 80–97. Excerpt `continuous_precision`. SHA-256 in Section 15.

```
theorem precision_allocation_optimal [Nonempty E] (s c : E → ℝ) (budget : ℝ)
    (hs : ∀ e, 0 < s e) (hc : ∀ e, 0 < c e) (hb : 0 < budget)
    (eps : E → ℝ) (heps : AllocationFeasible s budget eps) :
    precisionObjective c (optimalTolerance s c budget) ≤ precisionObjective c eps := by
  have ho := optimal_tolerance_positive s c budget hs hc hb
  have hbalance := relative_budget_identity s c eps budget hs hc hb heps.2
  have hlog : (∑ e, c e * Real.log (eps e / optimalTolerance s c budget e)) ≤ 0 := by
    calc
      _ ≤ ∑ e, c e * (eps e / optimalTolerance s c budget e - 1) := by
        apply Finset.sum_le_sum
        intro e _
```

```
        exact mul_le_mul_of_nonneg_left
          (Real.log_le_sub_one_of_pos (div_pos (heps.1 e) (ho e))) (le_of_lt (hc e))
      _ = 0 := by
        simp_rw [mul_sub, mul_one]
        rw [Finset.sum_sub_distrib, hbalance, sub_self]
  have hd := precision_objective_difference c eps (optimalTolerance s c budget) heps.1 ho
  linarith
```

Proof reading. First prove positivity of the candidate and an exact weighted-ratio balance. Compare each competitor ratio $\varepsilon_e/\varepsilon_e^*$ with the inequality $\log x \le x - 1$. Multiply by nonnegative costs and sum. The balance makes the upper sum zero, and the objective-difference identity changes its sign to obtain the global lower bound. The strict logarithmic inequality at a differing edge gives strictness and then uniqueness in subsequent declarations. The proof uses no stationary-point optimality assumption.

7.4 Caps separate the allocation from its multiplier

`clippedTolerance s c m t e` is $\min(m_e, tc_e/s_e)$. The cap vector is positive. The budget is positive and no larger than the total weighted cap. A continuous monotone total-allocation function yields an attaining positive scale.

Source: `formalization/ClippedPrecision.lean`.

Exact lines 211–234. Excerpt `clipped_precision`. SHA-256 in Section 15.

```
theorem clipped_unique_minimizer [Nonempty E] (s c m : E → ℝ) (budget : ℝ)
    (hs : ∀ e, 0 < s e) (hc : ∀ e, 0 < c e) (hm : ∀ e, 0 < m e)
    (hb : 0 < budget) (hcap : budget ≤ ∑ e, s e * m e) :
    ∃ t, 0 < t ∧ ClippedFeasible s m budget (clippedTolerance s c m t) ∧
      ∀ eps, ClippedFeasible s m budget eps →
        ((∀ other, ClippedFeasible s m budget other →
            precisionObjective c eps ≤ precisionObjective c other) ↔
          eps = clippedTolerance s c m t) := by
  obtain ⟨t, ht, heq⟩ := clipped_scale_exists s c m budget hs hc hm hb hcap
  have hf : ClippedFeasible s m budget (clippedTolerance s c m t) :=
    ⟨⟨clipped_positive s c m t hs hc hm ht, heq⟩, fun _ => min_le_left _ _⟩
  refine ⟨t, ht, hf, ?_⟩
  intro eps heps
  constructor
  · intro hmin
    by_contra hne
    have hl := clipped_allocation_strict s c m budget t hs hc hm ht heq eps heps hne
    have hu := hmin _ hf
    linarith
  · intro h
    subst eps
    exact fun other hother => clipped_allocation_optimal s c m budget t hs hc hm ht heq other
      ↪ hother

/- The parameter is the reciprocal of the manuscript's multiplier. -/
```

Proof reading. Obtain an attaining scale from `clipped_scale_exists`. Its positivity and budget identity establish feasibility. The minimum supplies the cap inequalities. A distinct feasible vector

has strictly larger objective by the supporting logarithmic inequality. This rules out every competing minimizer and gives the equivalence with the constructed tolerance vector. At total-cap budget, the tolerance vector remains unique but the scale, and hence the reciprocal multiplier, need not be unique. The endpoint-nonuniqueness declaration records that distinction.

7.5 Discrete choices guard natural subtraction

Each edge has a finite list of pairs: quantized error weight in $\mathbb{N}$ and exact ordered-field cost. `Assignment edges` independently chooses one listed option per edge. An empty option list is infeasible, while an empty edge list has the empty assignment.

Source: `formalization/DiscretePrecision.lean`.

Exact lines 38–42. Excerpt `discrete_recurrence`. SHA-256 in Section 15.

```
def discreteTable : List (List (ℕ × R)) → ℕ → Option R
  | [], b => singletonTable 0 0 b
  | opts :: edges, b =>
    bestOver (fun o => if o.1 ≤ b then
      add (discreteTable edges (b - o.1)) (some o.2) else none) opts
```

Proof reading. The recursive table records exact total weight. Its guard checks that the current option fits before using `b - o.1`. Natural subtraction truncates at zero, so an unguarded recurrence could accept an overweight option through a zero residual. The development proves exact-weight semantics by induction on the edge list, then scans weights up to the budget. Zero-error options are allowed.

Source: `formalization/DiscretePrecision.lean`.

Exact lines 110–121. Excerpt `discrete_endpoint`. SHA-256 in Section 15.

```
theorem discreteBudget_some_iff (edges : List (List (ℕ × R))) (B : ℕ) (v : R) :
    discreteBudget edges B = some v ↔
      (∃ a : Assignment edges, a.weight ≤ B ∧ a.cost = v) ∧
        ∀ a : Assignment edges, a.weight ≤ B → v ≤ a.cost := by
  rw [minimum_some_iff _ _ (discrete_precision_program edges B)]
  constructor
  · intro h
    exact ⟨h.1, fun a ha => h.2 _ ⟨a, ha, rfl⟩⟩
  · intro h
    refine ⟨h.1, ?_⟩
    rintro z ⟨a, ha, rfl⟩
    exact h.2 a ha
```

Proof reading. The returned `some v` is equivalent to an attained feasible assignment cost together with a lower bound for every feasible assignment. The helper's `IsMinimum` semantics guarantees both parts. The proof translates its value predicate back to an actual assignment. The companion `discreteBudget_none_iff` equates `none` with absence of a complete feasible assignment. Finite sentinel costs are never substituted for infeasibility.

8 Exact packet compilation and cached refinement

8.1 An independently defined feasible set

The tree label type α contains application-specific packet choices. Internal nodes have `arity + 1` children, so this constructor has at least one child. Costs live in a linearly ordered field with its compatible strict ring order. This permits exact rational execution as well as real-valued models.

Source: `formalization/ExactTreeCompiler.lean`.

Exact lines 21–23. Excerpt `candidate_tree`. SHA-256 in Section 15.

```
inductive CandidateTree (α : Type*) where
  | leaf (label : α)
  | node (label : α) (arity : ℕ) (children : Fin (arity + 1) → CandidateTree α)
```

Source: `formalization/ExactTreeCompiler.lean`.

Exact lines 33–36. Excerpt `cut`. SHA-256 in Section 15.

```
inductive Cut : CandidateTree α → Type _ where
  | keep (t : CandidateTree α) : Cut t
  | split {a : α} {n : ℕ} {ts : Fin (n + 1) → CandidateTree α}
      (cs : ∀ i, Cut (ts i)) : Cut (.node a n ts)
```

Proof reading. `Cut t` is indexed by the tree it prunes. Keeping a tree gives a legal cut, while splitting an internal node requires one legal cut of every child. This type is defined before the optimization recurrence. Therefore correctness has an independent feasible set against which to compare the recurrence.

Source: `formalization/ExactTreeCompiler.lean`.

Exact lines 45–47. Excerpt `tree_recurrence`. SHA-256 in Section 15.

```
def optimalValue (keep overhead : α → R) : CandidateTree α → R
  | .leaf a => keep a
  | .node a _ ts => min (keep a) (overhead a + ∑ i, optimalValue keep overhead (ts i))
```

Proof reading. A leaf must be kept. At an internal node, compare the node's keep cost against the split overhead plus child optimal values. The proof `optimalValue_le_cut` proceeds by induction on the tree and cases on a legal cut. Keeping selects the left minimum bound, while splitting combines the right minimum bound with all child bounds.

Source: `formalization/ExactTreeCompiler.lean`.

Exact lines 69–76. Excerpt `compile_attains`. SHA-256 in Section 15.

```
theorem compile_attains (keep overhead : α → R) (t : CandidateTree α) :
    cutValue keep overhead (compile keep overhead t) = optimalValue keep overhead t := by
  induction t with
  | leaf a => rfl
  | node a n ts ih =>
    by_cases h : keep a ≤ overhead a + ∑ i, optimalValue keep overhead (ts i)
    · simp [compile, h, cutValue, root, optimalValue]
    · simp [compile, h, cutValue, ih, optimalValue, min_eq_right (le_of_not_ge h)]
```

Proof reading. The actual `compile` returns a cut and chooses keep at ties. Induction proves its objective equals the recurrence. If the keep comparison holds, simplifying the conditional proves the keep case. Otherwise the induction hypotheses rewrite every compiled child cost and the right branch realizes the minimum.

Source: `formalization/ExactTreeCompiler.lean`.

Exact lines 79–83. Excerpt `tree_exact`. SHA-256 in Section 15.

```
theorem exact_tree_compiler (keep overhead : α → R) (t : CandidateTree α) :
    (∀ c : Cut t, optimalValue keep overhead t ≤ cutValue keep overhead c) ∧
    ∃ c : Cut t, cutValue keep overhead c = optimalValue keep overhead t := by
  exact ⟨optimalValue_le_cut keep overhead t, compile keep overhead t,
    compile_attains keep overhead t⟩
```

Proof reading. The final conjunction pairs the universal lower bound with the constructed attaining cut. Optimality is thus stronger than a recurrence-value identity. Lagrangian compilation later specializes keep scores to cost plus multiplier times error. Weighted families combine their actual cut objectives before applying this same theorem.

8.2 The Pareto statement has a boundary

The monotone-frontier theorem compares two genuinely optimal cuts at distinct nonnegative multipliers. Adding the two optimality inequalities bounds their errors in opposite order to their costs. A positive-multiplier optimum is Pareto efficient. The source also includes `unsupported_pareto_example`: weighted sums need not select every efficient point of a discrete feasible set. The exact budget compiler therefore has a separate role.

8.3 An exact quantized table

Source: `formalization/QuantizedTreeCompiler.lean`.

Exact lines 27–33. Excerpt `budget_recurrence`. SHA-256 in Section 15.

```
def budgetTable (κ : α → R) (w : α → ℕ) (σ : α → R) :
    CandidateTree α → ℕ → Option R
  | .leaf a, b => BudgetDP.singletonTable (w a) (κ a) b
  | .node a _ ts, b =>
    BudgetDP.choose (BudgetDP.singletonTable (w a) (κ a) b)
      (BudgetDP.add (some (σ a))
        (BudgetDP.combine (fun i => budgetTable κ w σ (ts i)) b))
```

Proof reading. Each keep option occupies its exact weight, while splitting convolves the child exact-weight tables and adds overhead. `BudgetDP.choose` compares two attainable options. The min-plus infrastructure proves that sums and finite choices preserve the independent `IsMinimum` predicate. Induction on the actual candidate tree therefore connects every table entry to legal cuts of exactly that weight.

Source: formalization/QuantizedTreeCompiler.lean.

Exact lines 107–119. Excerpt budget_endpoint. SHA-256 in Section 15.

```
theorem compileBudget_some_iff (κ : α → R) (w : α → ℕ) (σ : α → R)
    (t : CandidateTree α) (B : ℕ) (v : R) :
    compileBudget κ w σ t B = some v ↔
      (∃ c : Cut t, cutWeight w c ≤ B ∧ cutValue κ σ c = v) ∧
      ∀ c : Cut t, cutWeight w c ≤ B → v ≤ cutValue κ σ c := by
  rw [BudgetDP.minimum_some_iff _ _ (quantized_budget_compiler κ w σ t B)]
  constructor
  · intro h
    exact ⟨h.1, fun c hc => h.2 _ ⟨c, hc, rfl⟩⟩
  · intro h
    refine ⟨h.1, ?_⟩
    rintro z ⟨c, hc, rfl⟩
    exact h.2 c hc
```

Proof reading. The budget scan ranges over weights from zero through B. The endpoint states that a finite answer is attained by a legal cut of weight at most B, and no feasible cut costs less. This is an optimum over the original tree's cuts, rather than a continuous relaxation or a weighted-sum surrogate.

Source: formalization/QuantizedTreeCompiler.lean.

Exact lines 122–126. Excerpt budget_none. SHA-256 in Section 15.

```
theorem compileBudget_none_iff (κ : α → R) (w : α → ℕ) (σ : α → R)
    (t : CandidateTree α) (B : ℕ) :
    compileBudget κ w σ t B = none ↔ ¬ ∃ c : Cut t, cutWeight w c ≤ B := by
  rw [BudgetDP.minimum_none_iff _ _ (quantized_budget_compiler κ w σ t B)]
  simp
```

Proof reading. The infeasibility branch is exactly the negation of the existence of a budget-feasible legal cut. Both directions follow from the same proved minimum semantics. This is why the option-valued model distinguishes none from an expensive but attainable value.

8.4 The specification and stored implementation are separate

The functional recurrence above is a convenient specification. Evaluating it independently at many weights can recompute child tables. StoredBudgetCompiler adds vectors of $B+1$ option-valued entries, bounded lookup, tabulation, cached child arrays, and stored min-plus merges.

Source: formalization/StoredBudgetCompiler.lean.

Exact lines 109–118. Excerpt stored_tree. SHA-256 in Section 15.

```
def storedTree (B : ℕ) (κ : α → R) (w : α → ℕ) (σ : α → R) :
    CandidateTree α → BudgetDP.StoredTable R B
  | .leaf a => BudgetDP.tabulate B (BudgetDP.singletonTable (w a) (κ a))
  | .node a n ts =>
    let cached := Vector.ofFn (fun i => storedTree B κ w σ (ts i))
    let merged := BudgetDP.combineStored (fun i : Fin (n + 1) => cached.get i)
```

```
    BudgetDP.tabulate B (fun b => BudgetDP.choose (BudgetDP.singletonTable (w a) (κ a) b)
      (BudgetDP.add (some (σ a)) (BudgetDP.lookup merged b)))

/- The implemented cached array agrees entrywise with the proved exact recurrence. -/
```

Proof reading. At a node, `Vector.ofFn` computes the child arrays before combining them. The final tabulation reads the merged array at each requested weight. Every array is finite and each child result is reused. The loop structure justifies the stored-table work model. It is not inferred from the earlier functional recurrence.

Source: `formalization/StoredBudgetCompiler.lean`.

Exact lines 120–133. Excerpt `stored_correct`. SHA-256 in Section 15.

```
theorem storedTree_correct (B : ℕ) (κ : α → R) (w : α → ℕ) (σ : α → R)
    (t : CandidateTree α) (b : ℕ) (hb : b ≤ B) :
    BudgetDP.lookup (storedTree B κ w σ t) b = budgetTable κ w σ t b := by
  induction t generalizing b with
  | leaf a => exact BudgetDP.lookup_tabulate B _ b hb
  | node a n ts ih =>
    simp only [storedTree, budgetTable, BudgetDP.lookup_tabulate _ _ _ hb]
    congr 2
    rw [BudgetDP.combineStored_correct _ b hb]
    apply BudgetDP.combine_congr_local
    intro i k hk
    simpa only [BudgetDP.vector_get_ofFn] using ih i k (hk.trans hb)

/- The executable tabulated compiler scans its stored root table. -/
```

Proof reading. Induct on the tree. The leaf is bounded lookup of a tabulated singleton. At a node, unfold the two constructions, use correctness of stored child combination, and restrict convolution dependencies to indices no larger than the queried weight. Apply each child induction hypothesis within that bound. This yields entrywise equality with the specification. The stored optimizer's attained minimum is then derived from the already proved specification semantics.

Source: `formalization/StoredBudgetCompiler.lean`.

Exact lines 256–260. Excerpt `pseudo_polynomial`. SHA-256 in Section 15.

```
theorem tabulation_pseudopolynomial (B : ℕ) (t : CandidateTree α) :
    tabulationWork B t ≤ nodeCount t * (3 * (B + 1) + (B + 1)^2) := by
  have h := tabulation_work_bound B t
  have hm := mergeCount_add_one t
  nlinarith
```

Proof reading. The source separately defines the tabulation work and number of merges. The bound `tabulation_work_bound` supplies per-node table and merge costs, and `mergeCount_add_one` relates merges to the tree size. Algebra combines them into the displayed bound. The complexity is polynomial in the numeric budget B, so pseudo-polynomial in a binary encoding. Costs are unit-cost table cells and candidate trials. Arbitrary-precision bit complexity, compiler runtime, and device timing are excluded.

8.5 Stability requires controlled branch gaps

Source: `formalization/TreeCompilerUpdates.lean`.

Exact lines 45–59. Excerpt `compiler_stability`. SHA-256 in Section 15.

```
theorem compiler_stability (keep overhead keep' overhead' : α → R)
    (t : CandidateTree α) (hg : GapControlled keep overhead keep' overhead' t) :
    compile keep' overhead' t = compile keep overhead t := by
  induction t with
  | leaf a => rfl
  | node a n ts ih =>
    rcases hg with ⟨hk, hs, hc⟩
    have ho := half_gap_order _ _ _ _ hk hs
    simp only [compile]
    by_cases h : keep a ≤ overhead a + ∑ i, optimalValue keep overhead (ts i)
    · rw [ite_eq_left h, ite_eq_left (ho.mpr h)]
    · rw [ite_eq_right h, ite_eq_right (fun hp => h (ho.mp hp))]
      congr 1
      funext i
      exact ih i (hc i)
```

Proof reading. `GapControlled` recursively supplies bounds on perturbations of keep and split scores, including all child problems. `half_gap_order` proves the original and perturbed comparisons have the same truth value. The proof follows the two branches of `compile`. When both split, induction identifies every child cut. A near tie without this control is not covered by an unchanged-decision guarantee.

Source: `formalization/TreeCompilerUpdates.lean`.

Exact lines 142–150. Excerpt `local_update`. SHA-256 in Section 15.

```
theorem local_recompilation (keep overhead : α → R) (ctx : CachedContext α R)
    (hv : Valid keep overhead ctx) (newTree : CandidateTree α) :
    (∀ c : Cut (fill ctx newTree),
      updateValue keep overhead ctx (optimalValue keep overhead newTree) ≤
        cutValue keep overhead c) ∧
    ∃ c : Cut (fill ctx newTree), cutValue keep overhead c =
      updateValue keep overhead ctx (optimalValue keep overhead newTree) := by
  rw [updateValue_correct keep overhead ctx hv newTree]
  exact exact_tree_compiler keep overhead (fill ctx newTree)
```

Proof reading. A `CachedContext` stores the ancestor context around a replaced subtree, including sibling data. `Valid` asserts those cached values match their subtree optima. `updateValue_correct` identifies the refreshed root value with the optimum of the filled tree. The exact compiler theorem then supplies its lower bound and attaining witness. Separate declarations prove sibling trees and sibling values unchanged. The displayed theorem is correctness of local recomputation. A hardware time-saving claim requires its own execution evidence.

9 Output contracts, batching, and Krylov certificates

9.1 The decoding map is part of the contract

The signature for a fair timing comparison contains a case key, output contract, reference, and timing scope. Equality of those four fields defines `FairMatch`. The source proves that it is an equivalence relation. This theorem does not establish that recorded benchmark metadata contain every field needed to instantiate the relation.

Source: `formalization/SchedulingContracts.lean`.

Exact lines 54–60. Excerpt `contract`. SHA-256 in Section 15.

```
theorem contract_conversion {X C Y : Type*} (compressed : X → C) (explicit : X → Y)
    (decode : C → Y) (runTime : X → ℝ) (decodeTime : C → ℝ)
    (hdecode : ∀ x, decode (compressed x) = explicit x) (x : X) :
    (decodedRun compressed decode runTime decodeTime x).1 = explicit x ∧
    (decodedRun compressed decode runTime decodeTime x).2 =
      runTime x + decodeTime (compressed x) := by
  exact ⟨hdecode x, rfl⟩
```

Proof reading. The hypothesis `hdecode` says the actual decoding function converts the compressed result to the requested explicit object. The modeled decoded run returns that object and adds the declared conversion time. The two conclusion components follow from the decoding identity and the definition. The times are input models, so the theorem does not independently verify their measurement. A separate declaration shows that omitting a positive decoding time understates the modeled total.

9.2 Grouping preserves all contributions

Source: `formalization/SchedulingContracts.lean`.

Exact lines 89–93. Excerpt `grouping`. SHA-256 in Section 15.

```
theorem grouped_execution_output {E Shape V : Type*} [Fintype E] [DecidableEq Shape]
    [AddCommMonoid V] (key : E → Shape) (value : E → V) :
    (∑ k ∈ activeShapeClasses key, ∑ e ∈ Finset.univ.filter (fun e => key e = k), value e) =
      ∑ e, value e := by
  exact Finset.sum_fiberwise_of_maps_to (fun e _ => each_block_has_shape key e) value
```

Proof reading. The complete kernel key maps every active event to one shape class. Summing over the image classes and their fibers reindexes the original finite sum, and the library's fiberwise sum theorem proves equality. The companion work theorem repeats this identity for natural-valued work. The launch-count theorem bounds the number of classes by the number of events. The model requires that the key really include the fields needed for a common kernel. It does not verify an arbitrary runtime grouping implementation.

9.3 Strict profitability uses floor plus one

Source: `formalization/SchedulingContracts.lean`.

Exact lines 117–123. Excerpt `amortization`. SHA-256 in Section 15.

```
theorem integer_amortization_threshold (setup baseline candidate : ℝ)
    (hs : 0 ≤ setup) (h : candidate < baseline) (reuse : ℕ) :
    setup + (reuse : ℝ) * candidate < (reuse : ℝ) * baseline ↔
      minimumReuse setup baseline candidate ≤ reuse := by
  rw [amortization_threshold setup baseline candidate reuse h, minimumReuse]
  rw [← Nat.floor_lt (div_nonneg hs (le_of_lt (sub_pos.mpr h)))]
  omega
```

Proof reading. With nonnegative setup and a strictly smaller per-use candidate time, solve the strict total-cost inequality by division through the positive saving. The natural floor lemma converts a strict real threshold into the minimum integer reuse. Equality at break-even is not a profitable use, which accounts for the added one. The minimum-attainment theorem checks that threshold and all smaller reuse counts. Capture/replay and materialization have separate specializations with their own premises.

9.4 Invariant packet Krylov spaces

Contract. The Krylov development uses bounded linear maps on a normed space over a nontrivially normed field, with a finite packet family. The exact theorem assumes the operator commutes with every packet projector.

Source: `formalization/KrylovPackets.lean`.

Exact lines 89–99. Excerpt `krylov`. SHA-256 in Section 15.

```
theorem invariant_packet_krylov (P : PacketFamily (X →L[K] X) I)
    (A : X →L[K] X) (hA : ∀ i, Commute A (P.proj i)) (v : X) (m : ℕ) :
    krylovSpace A v m ≤ ⨆ i, krylovSpace (P.block A i i) (P.proj i v) m := by
  apply Submodule.span_le.mpr
  rintro x ⟨j, rfl⟩
  change (A ^ j.val) v ∈ (⨆ i, krylovSpace (P.block A i i) (P.proj i v) m : Submodule K X)
  rw [invariant_packet_power_expansion P A hA v j.val]
  apply Submodule.sum_mem
  intro i _
  exact Submodule.mem_iSup_of_mem i
    (krylov_generator_mem (P.block A i i) (P.proj i v) m j.val j.isLt)
```

Proof reading. To prove inclusion of a span, it suffices to prove membership for each generator $A^j v$. The preceding exact power-expansion theorem writes that generator as the sum of local packet powers acting on projected seeds. Each term belongs to its local Krylov subspace, hence to the supremum of those spaces. The finite sum remains inside. This endpoint is an inclusion. `local_krylov_direct` separately proves directness using the mutually annihilating packet projectors. It does not assert equality of the global seed Krylov space with the full independent sum of all local spaces.

Source: formalization/KrylovPackets.lean.

Exact lines 183–195. Excerpt approx_krylov. SHA-256 in Section 15.

```
theorem approximate_packet_krylov_distance (P : PacketFamily (X →L[K] X) I)
    (D E : X →L[K] X) (hD : PacketDiagonal P D) (v : X) (j : ℕ) (hj : 0 < j) :
    Metric.infDist (((D + E) ^ j) v)
      ((⨆ i, krylovSpace (P.block D i i) (P.proj i v) (j + 1) : Submodule K X) : Set X) ≤
      (j : ℝ) * (max ‖D + E‖ ‖D‖) ^ (j - 1) * ‖E‖ * ‖v‖ := by
  have hm : (D ^ j) v ∈ (⨆ i, krylovSpace (P.block D i i) (P.proj i v) (j + 1) : Submodule K X)
   ↪  :=
    invariant_packet_krylov P D (packet_diagonal_commutes P D hD) v (j + 1)
      (krylov_generator_mem D v (j + 1) j (Nat.lt_succ_self j))
  calc
    _ ≤ dist (((D + E) ^ j) v) ((D ^ j) v) := Metric.infDist_le_dist_of_mem hm
    _ = ‖((D + E) ^ j - D ^ j) v‖ := by rw [dist_eq_norm]; rfl
    _ ≤ ‖(D + E) ^ j - D ^ j‖ * ‖v‖ := ((D + E) ^ j - D ^ j).le_opNorm v
    _ ≤ _ := mul_le_mul_of_nonneg_right (approximate_packet_power_bound D E j hj) (norm_nonneg _)
```

Proof reading. Packet diagonality of D gives commutation, so $D^j v$ lies in the local Krylov sum of length $j+1$. The distance to that subspace is no larger than the distance to this particular vector. Bound the difference action by its operator norm, then apply the power-perturbation estimate. The displayed premise 0 < j permits the exponent $j-1$. A zero-power identity is handled separately in the power development.

9.5 The Jacobi normalization uses the L2 operator norm

invSqrt D is the actual inverse of the positive matrix square root CFC.sqrt D. matrixCondition A is the product of the Euclidean operator norms of A and its inverse. The matrix index type is finite and nonempty at the condition-bound endpoint. Positivity and invertibility assumptions are explicit.

Source: formalization/BlockJacobi.lean.

Exact lines 87–95. Excerpt jacobi. SHA-256 in Section 15.

```
theorem block_jacobi_condition (D K : Matrix X X ℂ)
    (hD : D.PosDef) (hA : (D+K).PosDef) (η : ℝ)
    (hE : ‖invSqrt D*K*invSqrt D‖ ≤ η) (hη : η < 1) :
    matrixCondition (invSqrt D*(D+K)*invSqrt D) ≤ (1+η)/(1-η) := by
  have hunit := ((invSqrt_isUnit D hD).mul hA.isUnit).mul (invSqrt_isUnit D hD)
  have hdet := (Matrix.isUnit_iff_isUnit_det _).mp hunit
  have hinv := Matrix.mul_nonsing_inv (invSqrt D*(D+K)*invSqrt D) hdet
  rw [normalized_decomposition D K hD] at hinv ⊢
  exact near_one_condition_bound _ _ hinv η hE hη
```

Proof reading. Positive definiteness of D proves its inverse square root is a unit, and positive definiteness of $D+K$ proves the normalized matrix is a unit. Its inverse identity is therefore justified. The normalization theorem writes the matrix as $I+E$, and the near-one condition estimate uses $\|E\| \le \eta < 1$. The theorem requires that supplied norm bound. It does not compute an empirical estimator of η.

Source: formalization/BlockJacobi.lean.

Exact lines 134–141. Excerpt cg_bound. SHA-256 in Section 15.

```
theorem certified_energy_error {κ κmax e0 em : ℝ} (m : ℕ)
    (hκ : 1 ≤ κ) (hbound : κ ≤ κmax) (he0 : 0 ≤ e0)
    (hCG : em ≤ 2*(cgFactor κ)^m*e0) :
    em ≤ 2*(cgFactor κmax)^m*e0 := by
  have hp : (cgFactor κ)^m ≤ (cgFactor κmax)^m :=
    pow_le_pow_left₀ (cgFactor_nonneg hκ) (cgFactor_mono hbound) m
  exact hCG.trans (mul_le_mul_of_nonneg_right
    (mul_le_mul_of_nonneg_left hp (by norm_num)) he0)
```

Proof reading. The classical CG energy-error estimate is the hypothesis hCG. The proved monotonicity of its convergence factor replaces the actual condition number by a certified upper bound. Nonnegativity permits raising the factor and multiplying the initial error. This is a formal comparison of factors and bounds. It does not independently prove CG iteration correctness or the classical convergence theorem. Item 7.5 stays a separate classical input.

9.6 Discovery residuals are orthogonal in Frobenius geometry

Source: formalization/DiscoveryCertificate.lean.

Exact lines 126–137. Excerpt discovery. SHA-256 in Section 15.

```
theorem discovery_frobenius_certificate (hP : ∀ a, (P.proj a)ᴴ = P.proj a)
    (A : Matrix X X ℂ) :
    ‖A-diagonalCompression P A‖^2 =
      ∑ b, ∑ a, if b=a then 0 else ‖P.block A b a‖^2 := by
  rw [compression_residual P A,
    frobenius_sum_sq (offDiagonalTerm P A) (offDiagonal_trace_orthogonal P hP A)]
  simp only [Fintype.sum_prod_type,offDiagonalTerm]
  apply Finset.sum_congr rfl
  intro b _
  apply Finset.sum_congr rfl
  intro a _
  by_cases h : b=a <;> simp [h]
```

Proof reading. Self-adjoint packet projectors make distinct block terms trace-orthogonal. The source first expresses the compression residual as its off-diagonal block sum. The finite Pythagorean identity for those terms gives the squared Frobenius norm as the sum of squared block norms, with diagonal terms zeroed. An operator-norm certificate is then derived from the Frobenius bound. This is exact geometry of a supplied packet family. Simultaneous spectral decomposition is a separately tracked classical input rather than an algorithm proved here.

10 Small-core optimal recompression from constructed ranges

10.1 The shape of the proof

The sixth block completes item 5.6 with ten modules and 79 new theorem declarations. Start with arbitrary finite complex matrices L and R of equal column type, including rank-deficient and zero factors. Construct compact orthonormal bases of their actual column spaces. Express LR^* as an isometric lift of its small core. Supply an ordered thin SVD of that core by its exact equations. Prove diagonal low-rank lower bounds, then transfer them to every ambient rank-bounded competitor. Finally construct factors with exactly $\min(d, k)$ columns.

Every stage has its own declarations. A general transfer lemma may accept core optimality as a premise. The final theorem discharges that premise with independently proved diagonal lower bounds. The supplied SVD data contain no optimality conclusion or singular-tail formula.

10.2 Compact dimensions are the actual matrix ranks

`columnSpace A` is the range of the actual Euclidean linear map associated with A. `compactDim A` is its finite dimension. `columnBasis A` chooses an orthonormal basis of that subspace. `compactQ A` represents its inclusion into the ambient space, and `compactR A` represents the range-restricted map in that basis.

Source: `formalization/CompactRangeFactors.lean`.

Exact lines 51–57. Excerpt `compact_factor`. SHA-256 in Section 15.

```
theorem compact_factorization (A : Matrix m n ℂ) : compactQ A * compactR A = A := by
  unfold compactQ compactR
  rw [←LinearMap.toMatrix_comp]
  have he : (columnSpace A).subtype ∘ₗ (Matrix.toEuclideanLin A).rangeRestrict =
      Matrix.toEuclideanLin A := by ext x i; rfl
  rw [he,Matrix.toEuclideanLin_eq_toLin_orthonormal]
  exact LinearMap.toMatrix_toLin _ _ _
```

Proof reading. Convert the product of the two coordinate matrices to the matrix of a composition. The inclusion composed with the range restriction is the original linear map, by entrywise reflexivity. Converting that map back to the original matrix proves the reconstruction. Separate theorems prove `compactDim A = A.rank`, orthonormality of `compactQ`, and full row rank of `compactR`. These are derived facts about actual matrices. The construction uses classical choice of an orthonormal basis. It is not a numerical triangular QR procedure.

The actual factor core is compactR(L) compactR$(R)^*$. Its row and column dimensions equal the two original input ranks. Its rank equals the rank of LR^*. A numerical QR prototype can keep zero directions, so its reported intermediate size need not equal these compact dimensions. No numerical-to-formal correspondence is inferred from matching array shapes alone.

10.3 A best-rank conclusion quantifies over all competitors

Source: `formalization/SmallCoreTransfer.lean`.

Exact lines 16–17. Excerpt `best_rank`. SHA-256 in Section 15.

```
def BestRank (N : Matrix m n ℂ → ℝ) (A B : Matrix m n ℂ) (k : ℕ) : Prop :=
  B.rank ≤ k ∧ ∀ C, C.rank ≤ k → N (A-B) ≤ N (A-C)
```

Proof reading. `BestRank N A B k` combines a rank bound on the constructed approximation with the inequality for every matrix `C` of rank at most `k`. Its norm functional `N` is explicit. It neither restricts competitors to a chosen factorization nor assumes they live in the input column spaces.

For isometric rectangular factors Q, V, the source defines `lift` and `compress` by

$$\mathrm{lift}(Q, V, G) = QGV^*, \qquad \mathrm{compress}(Q, V, B) = Q^*BV.$$

The norm module proves that lifting preserves spectral and Frobenius norms, compression does not increase them, and compression does not increase rank. The spectral norm is the L2 operator norm, while the Frobenius norm is selected in its separate norm scope.

Source: `formalization/SmallCoreTransfer.lean`.

Exact lines 25–35. Excerpt `ambient_transfer`. SHA-256 in Section 15.

```
theorem spectral_best_lift (Q : Matrix m r ℂ) (V : Matrix n s ℂ)
    (hQ : Qᴴ*Q=1) (hV : Vᴴ*V=1) (G H : Matrix r s ℂ) (k : ℕ)
    (hbest : BestRank spectralNorm G H k) :
    BestRank spectralNorm (lift Q V G) (lift Q V H) k := by
  refine ⟨(rank_lift_le Q V H).trans hbest.1,fun B hB => ?_⟩
  calc
    spectralNorm (lift Q V G-lift Q V H) = spectralNorm (G-H) := by
      rw [←lift_sub,spectral_lift Q V hQ hV]
    _ ≤ spectralNorm (G-compress Q V B) := hbest.2 _ ((rank_compress_le Q V B).trans hB)
    _ = spectralNorm (compress Q V (lift Q V G-B)) := by rw [compressed_error Q V hQ hV]
    _ ≤ spectralNorm (lift Q V G-B) := spectral_compress_le Q V hQ hV _
```

Proof reading. Fix an arbitrary ambient competitor `B`. Its compressed matrix still has rank at most k. Compare the core approximation with this compressed competitor using the core best-rank hypothesis. Re-express that error as compression of the ambient error and use contraction. Isometric lifting preserves the approximation error exactly. The lifted approximation also has the required rank. This proves global ambient optimality, once the core premise is independently discharged. The parallel Frobenius lemma follows the same chain with its own norm facts.

10.4 The SVD input states equations, rather than desired conclusions

Source: `formalization/SmallCoreRecompression.lean`.

Exact lines 29–37. Excerpt `ordered_svd`. SHA-256 in Section 15.

```
structure OrderedThinSVD (G : Matrix r s ℂ) (d : ℕ) where
  left : Matrix r (Fin d) ℂ
  right : Matrix s (Fin d) ℂ
  values : Fin d → ℝ
  left_isometry : leftᴴ*left=1
  right_isometry : rightᴴ*right=1
  nonnegative : ∀ i, 0 ≤ values i
  sorted : Antitone values
  reconstruction : G=lift left right (diag values)
```

Proof reading. The two column matrices are isometries. Values are real, nonnegative, and antitone in the finite index order. Reconstruction says the core equals the isometric lift of their diagonal. These fields supply the exact algebraic thin SVD data. They assume no rank-bound optimality, tail error, norm preservation, or lower bound. They are also not a theorem that such data exist for every input or that a floating-point routine returns exact equations.

The formal dimension d is the number of supplied thin SVD columns. Zero values are permitted. The output column type `KeptColumn d k` retains indices below k. Its cardinality is proved to be $\min(d, k)$. Square roots are taken only of nonnegative retained values. No zero singular value is inverted.

10.5 The spectral lower bound has an actual kernel witness

For a competitor of rank at most $k < d$, restrict it to the first $k + 1$ coordinate directions. The restricted matrix has rank smaller than its domain dimension. Rank-nullity yields a nonzero kernel vector. The prefix embedding makes it an ambient vector supported in those coordinates.

Source: `formalization/SpectralOptimality.lean`.

Exact lines 55–82. Excerpt `kernel_witness`. SHA-256 in Section 15.

```
theorem prefix_kernel_witness (k : ℕ) (hk : k<d)
    (B : Matrix (Fin d) (Fin d) ℂ) (hB : B.rank≤k) :
    ∃ y : EuclideanSpace ℂ (Fin d), y≠0 ∧ B*ᵥ y=0 ∧
      ∀ i : Fin d, k+1 ≤ i.val → y i=0 := by
  let J := prefixEmbedding k (by omega : k+1≤d)
  have hj : Jᴴ*J=1 := prefix_isometry k _
  have hr : (B*J).rank < Fintype.card (Fin (k+1)) := by
    have := Matrix.rank_mul_le_left B J
    simp only [Fintype.card_fin]
    omega
  obtain ⟨z,hz,hbz⟩ := rank_lt_kernel_witness (B*J) hr
  let y : EuclideanSpace ℂ (Fin d) := WithLp.toLp 2 (J*ᵥ z)
  have hy : y≠0 := by
    intro he
    have he' : J*ᵥ z=0 := congrArg WithLp.ofLp he
    have hz' : (z : Fin (k+1) → ℂ)=0 := by
```

```
      calc
        (z : Fin (k+1) → ℂ) = (J^H*J)*ᵥ z := by rw [hj,Matrix.one_mulVec]
        _ = J^H*ᵥ(J*ᵥ z) := (Matrix.mulVec_mulVec _ _ _).symm
        _ = 0 := by rw [he',Matrix.mulVec_zero]
    apply hz
    exact (EuclideanSpace.equiv (Fin (k+1)) ℂ).injective hz'
  refine ⟨y,hy,?_,?_⟩
  · change B*ᵥ(J*ᵥ z)=0
    rw [Matrix.mulVec_mulVec]
    exact hbz
  · intro i hi
    exact prefix_support k _ z i hi
```

Proof reading. `rank_lt_kernel_witness` constructs a nonzero vector in the restricted kernel. The prefix embedding's isometry proves its image remains nonzero. Associativity of matrix-vector multiplication proves the competitor kills that image. The prefix-support lemma proves all later coordinates vanish. Thus the lower-bound argument has a concrete witness. It does not posit the existence of a favorable test vector as an independent assumption.

The diagonal action on that vector has norm at least the first discarded value times its vector norm. The competitor contributes zero, so the operator-norm inequality and positivity of the witness norm give the lower bound. If $k \geq d$, the first discarded value is defined to be zero and norm nonnegativity suffices.

Source: `formalization/SpectralOptimality.lean`.

Exact lines 120–125. Excerpt `spectral_best`. SHA-256 in Section 15.

```
theorem diagonal_spectral_best (σ : Fin d → ℝ) (hσ : ∀ i, 0≤σ i)
    (hsort : Antitone σ) (k : ℕ) :
    BestRank spectralNorm (diag σ) (diag (kept σ k)) k := by
  refine ⟨rank_kept_le σ k,fun B hB => ?_⟩
  rw [diagonal_residual,spectral_diagonal_tail σ hσ hsort]
  exact diagonal_spectral_lower σ hσ hsort k B hB
```

Proof reading. The retained diagonal has rank at most k. Its residual norm is the proved first-discarded-value formula. The separate `diagonal_spectral_lower` applies to every competitor and supplies the final comparison. Both the attained error and the universal lower bound are proved within the diagonal development.

10.6 The Frobenius lower bound uses kernel mass

Choose an orthonormal basis Q of the competitor's actual kernel. The row weights w_i of that basis are nonnegative, at most one, and sum to its dimension. Rank-nullity makes that total at least $d - k$. Decreasing squared singular values therefore allocate their least weighted mass at the tail.

Source: `formalization/FrobeniusOptimality.lean`.

Exact lines 35–60. Excerpt `weighted_tail`. SHA-256 in Section 15.

```
theorem weighted_tail_lower (a w : Fin d → ℝ) (ha : ∀ i, 0 ≤ a i)
    (hsort : Antitone a) (hw0 : ∀ i, 0 ≤ w i) (hw1 : ∀ i, w i ≤ 1)
```

```
    (k : ℕ) (hw : (d-k : ℕ) ≤ ∑ i,w i) :
    (∑ i, if i.val<k then (0:ℝ) else a i) ≤ ∑ i, a i*w i := by
  by_cases hk : k<d
  · let μ := a ⟨k,hk⟩
    have hμ : 0 ≤ μ := ha _
    have hp : ∀ i : Fin d,
        (if i.val<k then 0 else a i) + μ*(w i-(if i.val<k then 0 else 1)) ≤ a i*w i := by
      intro i
      by_cases hi : i.val<k
      · have hle : μ ≤ a i := hsort (by show i.val ≤ k; omega)
        simp only [hi,ite_true,zero_add,sub_zero]
        exact mul_le_mul_of_nonneg_right hle (hw0 i)
      · have hle : a i ≤ μ := hsort (by show k ≤ i.val; omega)
        simp only [hi,ite_false]
        nlinarith [mul_nonneg (sub_nonneg.mpr hle) (sub_nonneg.mpr (hw1 i))]
    have hs := Finset.sum_le_sum (fun i (_ : i ∈ (Finset.univ : Finset (Fin d))) => hp i)
    simp only [Finset.sum_add_distrib] at hs
    have ht := tail_indicator_sum (d:=d) k hk.le
    have hz : 0 ≤ μ*((∑ i,w i)-(d-k : ℕ)) := mul_nonneg hμ (sub_nonneg.mpr hw)
    rw [←Finset.mul_sum,Finset.sum_sub_distrib,ht] at hs
    linarith
  · have he : ∀ i : Fin d, i.val<k := fun i => by omega
    simp only [he,ite_true,Finset.sum_const_zero]
    exact Finset.sum_nonneg (fun i _ => mul_nonneg (ha i) (hw0 i))
```

Proof reading. For $k < d$, use the threshold value a_k. In the prefix, ordering bounds a_k above by a_i and nonnegative weights preserve the inequality. In the tail, $w_i \leq 1$ supplies the corresponding bound. Summing adds a nonnegative threshold times the excess total mass. The tail indicator sum is exactly $d - k$. For $k \geq d$, the tail sum is zero. This finite inequality is proved directly and invokes no Eckart–Young optimality theorem.

Source: `formalization/FrobeniusOptimality.lean`.

Exact lines 76–104. Excerpt `frobenius_lower`. SHA-256 in Section 15.

```
theorem diagonal_frobenius_lower (σ : Fin d → ℝ) (hσ : ∀ i, 0 ≤ σ i)
    (hsort : Antitone σ) (k : ℕ) (B : Matrix (Fin d) (Fin d) ℂ) (hB : B.rank≤k) :
    tailEnergy σ k ≤ frobeniusNorm (diag σ-B)^2 := by
  let Q := kernelFrame B
  have hQ : Qᴴ*Q=1 := kernelFrame_isometry B
  have hw : (d-k : ℕ) ≤ ∑ i,rowWeight Q i := by
    rw [rowWeight_sum Q hQ]
    simp only [Fintype.card_fin]
    have hn := kernel_rank_nullity B
    simp only [Fintype.card_fin] at hn
    exact_mod_cast (by omega : d-k ≤ kernelDim B)
  have hsort2 : Antitone (fun i => σ i^2) := by
    intro i j hij
    have hle := hsort hij
    nlinarith [hσ i,hσ j]
  have hl := weighted_tail_lower (fun i => σ i^2) (rowWeight Q)
    (fun i => sq_nonneg _) hsort2 (rowWeight_nonneg Q) (rowWeight_le_one Q hQ) k hw
```

```
  have he : tailEnergy σ k = ∑ i, if i.val<k then (0:ℝ) else σ i^2 := by
    unfold tailEnergy discarded
    apply Finset.sum_congr rfl
    intro i _
    split_ifs <;> simp
  rw [←he,←diagonal_frame_energy σ Q] at hl
  have hc := frobenius_right_compress_le Q hQ (diag σ-B)
  have hbq : (diag σ-B)*Q=diag σ*Q := by
    rw [Matrix.sub_mul,mul_kernelFrame,sub_zero]
  rw [hbq] at hc
  unfold frobeniusNorm at hl ⊢
  nlinarith [norm_nonneg (diag σ*Q),norm_nonneg (diag σ-B)]
```

Proof reading. Construct the actual kernel frame and establish its row-weight mass from rank-nullity. Apply the finite weighted-tail inequality to σ_i^2. The frame-energy identity converts its right side into the squared Frobenius norm of the diagonal action on the kernel frame. The competitor kills the frame, so the same action is the residual times that frame. Frobenius compression contraction bounds it by the full residual. Norm nonnegativity permits the final squared comparison.

10.7 Combining both optimalities and the exact errors

Source: `formalization/SmallCoreRecompression.lean`.

Exact lines 91–120. Excerpt `recompression_optimal`. SHA-256 in Section 15.

```
theorem recompression_optimality (Q : Matrix m r ℂ) (V : Matrix n s ℂ)
    (hQ : Qᴴ*Q=1) (hV : Vᴴ*V=1) {G : Matrix r s ℂ}
    (svd : OrderedThinSVD G d) (k : ℕ) :
    let A := lift Q V G
    let B := recompressLeft Q svd k * (recompressRight V svd k)ᴴ
    BestRank spectralNorm A B k ∧ BestRank frobeniusNorm A B k ∧
    spectralNorm (A-B)=nextSingular svd.values k ∧
    frobeniusNorm (A-B)^2=tailEnergy svd.values k ∧
    frobeniusNorm (A-B)=Real.sqrt (tailEnergy svd.values k) := by
  dsimp only
  have hG : lift Q V G=lift (Q*svd.left) (V*svd.right) (diag svd.values) :=
    (congrArg (lift Q V) svd.reconstruction).trans (lift_comp Q V _ _ _)
  rw [recompressed_product,hG]
  have hU := isometry_comp Q svd.left hQ svd.left_isometry
  have hZ := isometry_comp V svd.right hV svd.right_isometry
  have hs := spectral_best_lift (Q*svd.left) (V*svd.right) hU hZ _ _ k
    (diagonal_spectral_best svd.values svd.nonnegative svd.sorted k)
  have hf := frobenius_best_lift (Q*svd.left) (V*svd.right) hU hZ _ _ k
    (diagonal_frobenius_best svd.values svd.nonnegative svd.sorted k)
  refine ⟨hs,hf,?_,?_,?_⟩
  · rw [←lift_sub,spectral_lift _ _ hU hZ,diagonal_residual]
    exact spectral_diagonal_tail _ svd.nonnegative svd.sorted k
  · rw [←lift_sub,frobenius_lift _ _ hU hZ,diagonal_residual]
    exact frobenius_diagonal_tail _ k
  · rw [←lift_sub,frobenius_lift _ _ hU hZ,diagonal_residual]
    rw [←frobenius_diagonal_tail svd.values k,Real.sqrt_sq_eq_abs]
```

```
    have hn : 0 ≤ frobeniusNorm (diag (discarded svd.values k)) := by
      unfold frobeniusNorm
      exact norm_nonneg (diag (discarded svd.values k))
    exact (abs_of_nonneg hn).symm
```

Proof reading. Rewrite the core reconstruction as a lift through the composed isometries. Rewrite the factor product as the retained diagonal lift. Derive isometry of each composed factor. Apply the spectral and Frobenius transfer lemmas to their independently proved diagonal best-rank conclusions. For the three error formulas, use linearity of lift, norm preservation, and the diagonal tail identities. The square-root formula uses nonnegativity of the Frobenius norm. These are conclusions of the theorem, not fields of its SVD input.

Source: `formalization/SmallCoreRecompression.lean`.

Exact lines 124–141. Excerpt `small_core_endpoint`. SHA-256 in Section 15.

```
theorem small_core_optimal_recompression (L : Matrix m t ℂ) (R : Matrix n t ℂ)
    (svd : OrderedThinSVD (factorCore L R) d) (k : ℕ) :
    let Lk := recompressLeft (compactQ L) svd k
    let Rk := recompressRight (compactQ R) svd k
    BestRank spectralNorm (L*Rᴴ) (Lk*Rkᴴ) k ∧
    BestRank frobeniusNorm (L*Rᴴ) (Lk*Rkᴴ) k ∧
    spectralNorm (L*Rᴴ-Lk*Rkᴴ)=nextSingular svd.values k ∧
    frobeniusNorm (L*Rᴴ-Lk*Rkᴴ)^2=tailEnergy svd.values k ∧
    frobeniusNorm (L*Rᴴ-Lk*Rkᴴ)=Real.sqrt (tailEnergy svd.values k) ∧
    compactDim L=L.rank ∧ compactDim R=R.rank ∧
    Fintype.card (KeptColumn d k)=min d k := by
  dsimp only
  have h := recompression_optimality (compactQ L) (compactQ R)
    (compactQ_isometry L) (compactQ_isometry R) svd k
  dsimp only at h
  rw [←compact_product_reconstruction L R] at h
  exact ⟨h.1,h.2.1,h.2.2.1,h.2.2.2.1,h.2.2.2.2,
    compactDim_eq_rank L,compactDim_eq_rank R,keptColumn_card k⟩
```

Proof reading. Instantiate the preceding theorem with the compact factors constructed from the original L and R. Rewrite their lifted core as LR^*. Pair its two global optimality conclusions and exact tail errors with the proved equalities between compact dimensions and original ranks, and with the retained-column cardinality. Thus the endpoint concerns the original rectangular factor product and every ambient rank-bounded competitor.

10.8 Zero rank and full retention remain defined

Source: `formalization/RecompressionEndpoints.lean`.

Exact lines 70–74. Excerpt `full_retention`. SHA-256 in Section 15.

```
theorem recompression_full (Q : Matrix m r ℂ) (V : Matrix n s ℂ)
    {G : Matrix r s ℂ} (svd : OrderedThinSVD G d) (k : ℕ) (hk : d ≤ k) :
    recompressLeft Q svd k*(recompressRight V svd k)ᴴ = lift Q V G := by
  rw [recompressed_product,kept_all _ k hk,←lift_comp]
  exact congrArg (lift Q V) svd.reconstruction.symm
```

Proof reading. When $d \leq k$, retaining the entire value list gives the original diagonal. Rewrite the constructed product by that full-retention identity, then use the supplied SVD reconstruction under the outer lift. The separate rank-zero endpoint proves that keeping no columns gives the zero product. Empty index types avoid dummy inverse operations and support zero original factors. Repeated or zero singular values are permitted by the non-strict ordering.

10.9 The formal boundary

The ordered-SVD-data theorem proves conditional exact optimality and exact tail errors. General SVD existence, identification with the canonical Mathlib `singularValues` API, a floating-point SVD or QR algorithm, and its numerical stability are outside this formal block. Classical item 5.5 remains separately tracked, with these checked supporting results. The formal construction does not establish GPU performance, a generic dense matrix exponent, or equivalence to the supplied NumPy prototype.

11.4 Classical inputs are not folded into the 56 count

Item	Classical input	Precise present boundary
5.5	Eckart–Young–Mirsky	Ordered diagonal spectral and Frobenius best-rank bounds and isometric transfer are checked as sup- port for 5.6. General SVD existence and the canonical singular-values API theorem are not formalized here.
7.5	CG convergence	The classical energy-error estimate is supplied. The formal block proves monotonicity and replacement by a certified condition bound, rather than proving the CG algorithm or its classical convergence theorem.
8.2	Floating-point product model	Local error inequalities are supplied to exact normed ring propagation. IEEE rounding semantics and a device implementation are not independently proved.
9.1	Simultaneous spectral decomposition	Exact residual and off-diagonal certificates are proved for a supplied finite self-adjoint projector family. A general simultaneous-decomposition algorithm is not independently formalized.

11.5 Finite numerical examples

The recorded numerical mirror covers 182 factor pairs, 638 truncation cases, 3,190 arbitrary rank-bounded competitors, nine invalid-input controls and two assumption-removal controls. It uses NumPy complex128 and numerical rank thresholds. It is neither extracted nor compiled Lean and does not prove universal matrix optimality. The CPU/GPU measurements remain separate and do not automatically instantiate the complete formal fair-match signature.

11.6 Reproducible limits

The three companions share the original result keys and preserve the complete raw mathematical and formal sources. The paper-to-Lean map explains recorded hypotheses and scope. It is not an automatically checked semantic-equivalence theorem between prose, Lean, and prototype Python. No fresh CUDA correctness, device stability, timing advantage, full imported-library rebuild, or claims beyond the stated finite mathematics follow from the documentation checks.

12 Complete manuscript-result correspondence

The original manuscript numbering below is retained as a stable key across the companions. The preserved ledger has 85 numbered records, including definitions and expository entries. This map prints its 56 manuscript-proved results and four classical inputs. Source locations identify declarations. They do not claim that Lean certifies prose-to-code equivalence automatically.

12.1 2.3 — Exact evaluation

Traditional original item 2.3, current statement 2.3. Printed page 7, PDF page 13.

Recorded status. checked.

Recorded hypotheses and scope. Finite orthogonal idempotent family summing to one. Exact nonzero support, not thresholded support. Actual rectangular matrix batches are covered.

- `KomorebiPaperVIII.exact_active_evaluation`; `PacketIR.lean`, line 41.
- `KomorebiPaperVIII.exact_active_action`; `PacketIR.lean`, line 52.
- `KomorebiPaperVIII.exact_matrix_batch`; `PacketIR.lean`, line 145.

12.2 2.5 — Block composition

Traditional original item 2.5, current statement 2.5. Printed page 8, PDF page 14.

Recorded status. checked.

Recorded hypotheses and scope. Finite projector family over a ring. Relational composition gives an inclusion because nonzero path terms can cancel.

- `KomorebiPaperVIII.exact_block_composition`; `PacketIR.lean`, line 58.
- `KomorebiPaperVIII.product_support_inclusion`; `PacketIR.lean`, line 63.

12.3 2.6 — Adjoint and tensor semantics

Traditional original item 2.6, current statement 2.6. Printed page 9, PDF page 15.

Recorded status. checked.

Recorded hypotheses and scope. Adjoint identity explicitly assumes self-adjoint projectors in a star ring. Tensor identity uses actual Kronecker products of finite matrices.

- `KomorebiPaperVIII.exact_adjoint_block`; `PacketIR.lean`, line 95.
- `KomorebiPaperVIII.exact_tensor_block`; `PacketIR.lean`, line 135.

12.4 2.9 — Contract conversion

Traditional original item 2.9, current statement 2.9. Printed page 9, PDF page 15.

Recorded status. checked.

Recorded hypotheses and scope. Formal output conversion includes the actual decoding map and its declared time. Direct downstream composition is proved. This is a contract model, not independent validation of timings.

- `KomorebiPaperVIII.contract_conversion`; `SchedulingContracts.lean`, line 54.

- `KomorebiPaperVIII.direct_compressed_consumption`; `SchedulingContracts.lean`, line 62.
- `KomorebiPaperVIII.omitted_decode_understates_time`; `SchedulingContracts.lean`, line 69.

12.5 3.1 — Dense and packet application counts

Traditional original item 3.1, current statement 3.1. Printed page 10, PDF page 16.

Recorded status. checked.

Recorded hypotheses and scope. Finite scalar address and modeled multiply-accumulate event cardinalities. Two scalar operations per coefficient/right-hand side. No device instruction claim.

- `KomorebiPaperVIII.packet_storage_count`; `PacketCostModels.lean`, line 27.
- `KomorebiPaperVIII.packet_application_count`; `PacketCostModels.lean`, line 31.
- `KomorebiPaperVIII.dense_storage_count`; `PacketCostModels.lean`, line 36.
- `KomorebiPaperVIII.dense_application_count`; `PacketCostModels.lean`, line 39.

12.6 3.2 — Equal diagonal packets

Traditional original item 3.2, current statement 3.2. Printed page 10, PDF page 16.

Recorded status. checked.

Recorded hypotheses and scope. Equal widths, n = card(I)*b. Positive/nonzero denominators for real ratios. Work-ratio specialization follows from equal_bdlr_reduction at r=0.

- `KomorebiPaperVIII.equal_diagonal_storage`; `PacketCostModels.lean`, line 44.
- `KomorebiPaperVIII.equal_diagonal_work`; `PacketCostModels.lean`, line 48.
- `KomorebiPaperVIII.equal_block_reduction`; `PacketCostModels.lean`, line 87.

12.7 3.4 — BDLR application and storage

Traditional original item 3.4, current statement 3.4. Printed page 11, PDF page 17.

Recorded status. checked.

Recorded hypotheses and scope. Explicit diagonal and two thin factor address/event types; third kernel accumulates into the diagonal result. The actual algebraic action is the unchanged LowRankAlgebra.bdlr_apply theorem.

- `KomorebiPaperVIII.bdlr_storage_count`; `PacketCostModels.lean`, line 58.
- `KomorebiPaperVIII.bdlr_application_count`; `PacketCostModels.lean`, line 68.
- `KomorebiPaperVIII.equal_bdlr_storage`; `PacketCostModels.lean`, line 75.
- `KomorebiPaperVIII.equal_bdlr_work`; `PacketCostModels.lean`, line 81.
- `KomorebiPaperVIII.equal_bdlr_reduction`; `PacketCostModels.lean`, line 91.

12.8 3.6 — Streaming byte count

Traditional original item 3.6, current statement 3.6. Printed page 11, PDF page 17.

Recorded status. checked.

Recorded hypotheses and scope. One block read, one source read, target read and write per active edge, no interedge reuse. Count is of declared events.

- `KomorebiPaperVIII.streaming_value_count`; `PacketCostModels.lean`, line 102.
- `KomorebiPaperVIII.streaming_byte_count`; `PacketCostModels.lean`, line 111.

12.9 3.8 — Roofline lower bound

Traditional original item 3.8, current statement 3.8. Printed page 12, PDF page 18.

Recorded status. checked.

Recorded hypotheses and scope. Time bound assumes positive peak rates and actual capacity inequalities. Work-rate bound assumes nonnegative work, positive bytes and elapsed time. These hypotheses are explicit.

- `KomorebiPaperVIII.roofline_time_bound`; `PacketCostModels.lean`, line 118.
- `KomorebiPaperVIII.roofline_work_rate`; `PacketCostModels.lean`, line 126.

12.10 3.9 — Capacity threshold

Traditional original item 3.9, current statement 3.9. Printed page 12, PDF page 18.

Recorded status. checked.

Recorded hypotheses and scope. Dense scalar capacity equivalence uses n,M nonnegative and scalar byte size positive. BDLR natural-number storage inequality is exact; workspace memory is excluded.

- `KomorebiPaperVIII.dense_capacity_threshold`; `PacketCostModels.lean`, line 139.
- `KomorebiPaperVIII.bdlr_capacity_threshold`; `PacketCostModels.lean`, line 143.

12.11 3.10 — Materialization cost

Traditional original item 3.10, current statement 3.10. Printed page 13, PDF page 19.

Recorded status. checked.

Recorded hypotheses and scope. Explicit dense low-rank product followed by one scalar addition per output has 2*n^2*r+n^2 modeled operations. The remainder has the exact quadratic bound n^2, stronger than the displayed big-O term; no generic implementation lower bound.

- `KomorebiPaperVIII.materialization_work_count`; `PacketCostModels.lean`, line 151.
- `KomorebiPaperVIII.materialization_quadratic_remainder`; `PacketCostModels.lean`, line 156.
- `KomorebiPaperVIII.dense_storage_count`; `PacketCostModels.lean`, line 36.

12.12 4.1 — Equal block banded count

Traditional original item 4.1, current statement 4.1. Printed page 13, PDF page 19.

Recorded status. checked.

Recorded hypotheses and scope. Actual band address subtype in Fin p x Fin p. Count p*(2*w+1)-w*(w+1) proved from finite sums under w<p. Explicit stored-scalar and multiply/accumulate event types give b^2*N and 2*q*b^2*N. The manuscript substitution p=n/b additionally needs b>0 and exact divisibility. No hardware operation timing claim.

- `KomorebiPaperVIII.banded_block_count`; `BandedCount.lean`, line 73.
- `KomorebiPaperVIII.banded_storage_count`; `BandedCount.lean`, line 85.
- `KomorebiPaperVIII.banded_operation_count`; `BandedCount.lean`, line 90.

12.13 4.2 — Sparse plus low rank application

Traditional original item 4.2, current statement 4.2. Printed page 14, PDF page 20.

Recorded status. checked.

Recorded hypotheses and scope. Counts actual sparse-value and thin-factor address types, excluding sparse index metadata. Algebraic action follows from the upstream bdlr_apply identity, which does not require its D argument to be diagonal.

- `KomorebiPaperVIII.sparse_low_rank_storage_count`; `PacketCostModels.lean`, line 164.
- `KomorebiPaperVIII.sparse_low_rank_work_count`; `PacketCostModels.lean`, line 169.

12.14 4.3 — Circulant spectral representation

Traditional original item 4.3, current statement 4.3. Printed page 14, PDF page 20.

Recorded status. checked.

Recorded hypotheses and scope. Actual circulant and unitary-normalized Fourier matrix on ZMod n for every n>0. The exponential formula, both unitarity identities, exact diagonalization and two-transform application are checked. No FFT complexity or hardware execution is claimed.

- `KomorebiPaperVIII.fourier_matrix_exponential`; `CirculantFourier.lean`, line 39.
- `KomorebiPaperVIII.fourier_unitary`; `CirculantFourier.lean`, line 83.
- `KomorebiPaperVIII.circulant_spectral_representation`; `CirculantFourier.lean`, line 150.
- `KomorebiPaperVIII.circulant_fourier_application`; `CirculantFourier.lean`, line 155.

12.15 4.4 — Toeplitz circulant embedding

Traditional original item 4.4, current statement 4.4. Printed page 15, PDF page 21.

Recorded status. checked.

Recorded hypotheses and scope. Actual complex Toeplitz matrix, explicitly reversed generator of length n+n and Fin.addCases zero padding for n>0. The diagonal uses c[0]; r[0] is irrelevant. Neither symmetry nor Hermiticity is assumed.

- `KomorebiPaperVIII.toeplitz_circulant_block`; `ToeplitzEmbedding.lean`, line 28.
- `KomorebiPaperVIII.toeplitz_embedding_action`; `ToeplitzEmbedding.lean`, line 54.

- `KomorebiPaperVIII.toeplitz_generator_first_half`; `ToeplitzEmbedding.lean`, line 60.
- `KomorebiPaperVIII.toeplitz_generator_middle`; `ToeplitzEmbedding.lean`, line 64.

12.16 4.5 — Kronecker modewise application

Traditional original item 4.5, current statement 4.5. Printed page 15, PDF page 21.

Recorded status. checked.

Recorded hypotheses and scope. Actual column-major Matrix.vec and actual Kronecker products; the right factor uses transpose, not adjoint. Arithmetic and factor-storage counts are cardinalities of finite address/event types.

- `KomorebiPaperVIII.kronecker_modewise`; `StructuredTensor.lean`, line 19.
- `KomorebiPaperVIII.kronecker_modewise_work`; `StructuredTensor.lean`, line 27.
- `KomorebiPaperVIII.kronecker_factor_storage`; `StructuredTensor.lean`, line 32.

12.17 4.6 — Equal tensor factors

Traditional original item 4.6, current statement 4.6. Printed page 15, PDF page 21.

Recorded status. checked.

Recorded hypotheses and scope. Actual arbitrary-length recursive tensor matrix and slice-by-slice contraction. Exact event count 2*k*d^(k+1) and storage k*d^2; includes k=0 and d=0. Model charges two operations per multiply-accumulate.

- `KomorebiPaperVIII.tensor_dimension`; `StructuredTensor.lean`, line 51.
- `KomorebiPaperVIII.tensor_apply_correct`; `StructuredTensor.lean`, line 73.
- `KomorebiPaperVIII.tensor_modewise_work`; `StructuredTensor.lean`, line 112.
- `KomorebiPaperVIII.tensor_factor_storage`; `StructuredTensor.lean`, line 126.

12.18 4.7 — Walsh Hadamard recursion

Traditional original item 4.7, current statement 4.7. Printed page 16, PDF page 22.

Recorded status. checked.

Recorded hypotheses and scope. Recursively constructed complex Walsh matrices and actual sum/difference butterfly. Proves ordinary-transpose orthogonality, algorithm correctness and exactly N*log2(N) additions/subtractions for N=2^k, including N=1.

- `KomorebiPaperVIII.walsh_orthogonal`; `WalshTransform.lean`, line 33.
- `KomorebiPaperVIII.walsh_apply_correct`; `WalshTransform.lean`, line 60.
- `KomorebiPaperVIII.walsh_operation_count`; `WalshTransform.lean`, line 99.
- `KomorebiPaperVIII.walsh_log_operation_count`; `WalshTransform.lean`, line 115.

12.19 5.1 — Exact BDLR product

Traditional original item 5.1, current statement 5.1. Printed page 16, PDF page 22.

Recorded status. checked.

Recorded hypotheses and scope. Finite rectangular factors over a field with involution. Diagonal closure and actual correction rank are proved. No materialization or timing claim.

- `KomorebiPaperVIII.packet_diagonal_product`; `LowRankAlgebra.lean`, line 33.
- `KomorebiPaperVIII.bdlr_product`; `LowRankAlgebra.lean`, line 57.
- `KomorebiPaperVIII.bdlr_product_rank_bound`; `LowRankAlgebra.lean`, line 68.

12.20 5.2 — Rank growth in a product chain

Traditional original item 5.2, current statement 5.2. Printed page 17, PDF page 23.

Recorded status. checked.

Recorded hypotheses and scope. Actual variable-rank thin matrix factors are constructed for every length, including zero rank and the empty product. The correction matrix rank is bounded by the sum of factor widths. Packet diagonality of the base product is derived.

- `KomorebiPaperVIII.chainFactors_correct`; `LowRankChains.lean`, line 56.
- `KomorebiPaperVIII.bdlr_chain_rank_bound`; `LowRankChains.lean`, line 69.
- `KomorebiPaperVIII.packet_diagonal_chain`; `LowRankChains.lean`, line 87.

12.21 5.3 — Woodbury inversion

Traditional original item 5.3, current statement 5.3. Printed page 18, PDF page 24.

Recorded status. checked.

Recorded hypotheses and scope. Both D and I+V* D^{-1} U are units. The update is proved invertible. Ordinary matrix inverse and arbitrary right-hand-side batches. Uses Mathlib classical Woodbury identity.

- `KomorebiPaperVIII.bdlr_woodbury`; `LowRankAlgebra.lean`, line 89.
- `KomorebiPaperVIII.bdlr_update_isUnit`; `LowRankAlgebra.lean`, line 97.
- `KomorebiPaperVIII.bdlr_woodbury_solve`; `LowRankAlgebra.lean`, line 110.

12.22 5.4 — Inverse norm bound

Traditional original item 5.4, current statement 5.4. Printed page 18, PDF page 24.

Recorded status. checked.

Recorded hypotheses and scope. Real or complex matrices with the Hilbert-space L2 operator norm, including rectangular factors. Same two invertibility assumptions as 5.3.

- `KomorebiPaperVIII.bdlr_inverse_norm_bound`; `LowRankAlgebra.lean`, line 126.

12.23 5.5 — Eckart Young Mirsky theorem

Traditional original item 5.5, current statement 5.5. Printed page 18, PDF page 24.

Recorded status. external_input_with_checked_svd_data_specialization.

Recorded hypotheses and scope. Ordered diagonal spectral/Frobenius best-rank inequalities and rectangular isometric transfer are now independently checked as supporting results for 5.6. General SVD existence and a theorem phrased in the canonical Mathlib singularValues API are not formalized here. This classical entry stays separately tracked, not added to the 56 manuscript-proved count.

12.24 5.6 — Small core recompression

Traditional original item 5.6, current statement 5.6. Printed page 18, PDF page 24.

Recorded status. checked.

Recorded hypotheses and scope. Actual compact orthonormal column-space factorizations with dimensions equal to the input matrix ranks. For any supplied ordered thin SVD of the actual small core, constructs thin square-root factors with min(d,k) columns, proves rank <= k, global spectral and Frobenius optimality against every ambient competitor, and exact tail errors. No assumed optimality, tail formulas, norm invariance or rank identities. Includes zero and rank-deficient factors, k=0, k>=d and repeated singular values. SVD existence/computation, floating-point QR/SVD, and canonical Mathlib singularValues identification are not independently formalized. The supplied SVD equations are explicit hypotheses, as in the manuscript.

- `KomorebiPaperVIII.SmallCore.small_core_optimal_recompression`; `SmallCoreRecompression.lean`, line 124.
- `KomorebiPaperVIII.SmallCore.recompression_optimality`; `SmallCoreRecompression.lean`, line 91.
- `KomorebiPaperVIII.SmallCore.compact_product_reconstruction`; `SmallCoreTransfer.lean`, line 61.
- `KomorebiPaperVIII.SmallCore.diagonal_spectral_best`; `SpectralOptimality.lean`, line 120.
- `KomorebiPaperVIII.SmallCore.diagonal_frobenius_best`; `FrobeniusOptimality.lean`, line 106.

12.25 5.8 — Product chain perturbation

Traditional original item 5.8, current statement 5.8. Printed page 19, PDF page 25.

Recorded status. checked.

Recorded hypotheses and scope. Actual ordered products in a normed ring with norm-one identity. Includes the empty chain. The displayed finite sum uses perturbed suffix norms and exact prefix norms. This covers operators on nonzero normed spaces, without a commutativity assumption.

- `KomorebiPaperVIII.product_chain_perturbation`; `ProductChainError.lean`, line 148.
- `KomorebiPaperVIII.product_chain_error_bound`; `ProductChainError.lean`, line 129.

12.26 5.9 — Certified repeated recompression

Traditional original item 5.9, current statement 5.9. Printed page 20, PDF page 26.

Recorded status. checked.

Recorded hypotheses and scope. Actual left-multiplied iterates with additive post-product residuals in a normed ring. Both actual residual norms and arbitrary certified error budgets are supported, including the empty iteration.

- `KomorebiPaperVIII.repeated_recompression_bound`; `ProductChainError.lean`, line 100.
- `KomorebiPaperVIII.repeated_recompression_actual_error`; `ProductChainError.lean`, line 114.

12.27 6.1 — Path support of powers

Traditional original item 6.1, current statement 6.1. Printed page 21, PDF page 27.

Recorded status. checked.

Recorded hypotheses and scope. Recursive finite sum over active paths equals the actual block of A^n. Includes length zero explicitly.

- `KomorebiPaperVIII.power_path_expansion`; `TransferSupport.lean`, line 37.
- `KomorebiPaperVIII.power_block_zero_of_no_path`; `TransferSupport.lean`, line 74.
- `KomorebiPaperVIII.power_support_inclusion`; `TransferSupport.lean`, line 79.

12.28 6.2 — Polynomial support

Traditional original item 6.2, current statement 6.2. Printed page 21, PDF page 27.

Recorded status. checked.

Recorded hypotheses and scope. Actual finite polynomial sum in an algebra. Support contained in the union over powers, including the constant term.

- `KomorebiPaperVIII.polynomial_support_inclusion`; `TransferSupport.lean`, line 127.
- `KomorebiPaperVIII.polynomial_zero_of_no_paths`; `TransferSupport.lean`, line 142.

12.29 6.3 — Acyclic transfer nilpotence

Traditional original item 6.3, current statement 6.3. Printed page 21, PDF page 27.

Recorded status. checked.

Recorded hypotheses and scope. Absence of paths of length h+1 forces A^(h+1)=0. Also derived from a supplied strictly increasing topological level certificate. No graph-level construction algorithm is claimed.

- `KomorebiPaperVIII.nilpotent_of_path_bound`; `TransferSupport.lean`, line 86.
- `KomorebiPaperVIII.nilpotent_of_topological_levels`; `TransferSupport.lean`, line 109.

12.30 6.5 — Level schedule for a triangular solve

Traditional original item 6.5, current statement 6.5. Printed page 22, PDF page 28.

Recorded status. checked.

Recorded hypotheses and scope. Actual well-founded forward substitution and stage evaluator in dependent, possibly unequal block spaces. Finite acyclic graph, invertible local diagonal maps, and arbitrary linear edge transfers. No precomputed solution or topological schedule is assumed. Stages read only predecessors and solve the original block equations at canonical depth.

- `KomorebiPaperVIII.DAG.Graph.solve_correct`; `TriangularSolve.lean`, line 38.
- `KomorebiPaperVIII.DAG.Graph.solve_unique`; `TriangularSolve.lean`, line 46.
- `KomorebiPaperVIII.DAG.Graph.triangular_bijective`; `TriangularSolve.lean`, line 65.
- `KomorebiPaperVIII.DAG.Graph.stagedSolve_correct_through`; `TriangularSolve.lean`, line 91.
- `KomorebiPaperVIII.DAG.Graph.level_schedule_solves`; `TriangularSolve.lean`, line 112.

12.31 6.6 — Sequential depth

Traditional original item 6.6, current statement 6.6. Printed page 22, PDF page 28.

Recorded status. checked.

Recorded hypotheses and scope. Canonical predecessor-max levels are constructed from acyclicity through strict ancestor-cardinality decrease. For nonempty finite graphs, attained maximum equals one plus longest edge-path length. Empty graph has depth zero. This is an ideal unit-task schedule, not a hardware timing theorem.

- `KomorebiPaperVIII.DAG.Graph.level_attained`; `GraphSchedules.lean`, line 101.
- `KomorebiPaperVIII.DAG.Graph.path_length_le_depth`; `GraphSchedules.lean`, line 156.
- `KomorebiPaperVIII.DAG.Graph.depth_attained`; `GraphSchedules.lean`, line 164.
- `KomorebiPaperVIII.DAG.Graph.empty_depth`; `GraphSchedules.lean`, line 177.

12.32 6.7 — Inverse support lies in the transitive closure

Traditional original item 6.7, current statement 6.7. Printed page 23, PDF page 29.

Recorded status. checked.

Recorded hypotheses and scope. Actual reachable packet subspace in a finite-dimensional algebra is invariant. A left inverse makes the restricted forward map injective, hence surjective, yielding inverse invariance. Actual nonsingular matrix inverse specialization. Reachability is reflexive: zero-length paths are included. No inverse-support hypothesis is assumed.

- `KomorebiPaperVIII.inverse_block_zero_of_no_path`; `InverseSupport.lean`, line 94.
- `KomorebiPaperVIII.inverse_support_inclusion`; `InverseSupport.lean`, line 102.
- `KomorebiPaperVIII.matrix_inverse_support`; `InverseSupport.lean`, line 116.

12.33 6.8 — Schur support inclusion

Traditional original item 6.8, current statement 6.8. Printed page 23, PDF page 29.

Recorded status. checked.

Recorded hypotheses and scope. Actual rectangular retained-hidden factors and nonsingular hidden matrix inverse. Nonzero Schur block implies a direct retained edge or nonzero entrance, reachable hidden path (possibly zero length), and nonzero exit. Only support inclusion is asserted; cancellation may remove a permitted block.

- `KomorebiPaperVIII.schur_correction_expansion`; `SchurSupport.lean`, line 25.
- `KomorebiPaperVIII.schur_correction_support`; `SchurSupport.lean`, line 33.
- `KomorebiPaperVIII.schur_support_inclusion`; `SchurSupport.lean`, line 57.
- `KomorebiPaperVIII.schur_zero_of_no_route`; `SchurSupport.lean`, line 76.

12.34 7.2 — Invariant packet Krylov decomposition

Traditional original item 7.2, current statement 7.2. Printed page 24, PDF page 30.

Recorded status. checked.

Recorded hypotheses and scope. Actual bounded linear maps and spans of finite power orbits. Commutation with the complete finite projector family gives the power expansion and Krylov inclusion. Local spaces lie in the actual projector ranges and their sum is direct. No equality of the global and local Krylov spaces is asserted.

- `KomorebiPaperVIII.invariant_packet_power_expansion`; `KrylovPackets.lean`, line 76.
- `KomorebiPaperVIII.invariant_packet_krylov`; `KrylovPackets.lean`, line 89.
- `KomorebiPaperVIII.local_krylov_direct`; `KrylovPackets.lean`, line 131.

12.35 7.3 — Approximate packet Krylov bound

Traditional original item 7.3, current statement 7.3. Printed page 24, PDF page 30.

Recorded status. checked.

Recorded hypotheses and scope. Positive integer power, actual operator norm, and actual Metric.infDist to the sum of local Krylov spaces. Membership of the comparison vector D^j v is derived from packet diagonality. The power estimate also allows a trivial normed ring.

- `KomorebiPaperVIII.approximate_packet_power_bound`; `KrylovPackets.lean`, line 170.
- `KomorebiPaperVIII.approximate_packet_krylov_distance`; `KrylovPackets.lean`, line 183.

12.36 7.4 — Block Jacobi condition bound

Traditional original item 7.4, current statement 7.4. Printed page 25, PDF page 31.

Recorded status. checked.

Recorded hypotheses and scope. Nonempty finite complex matrices, D positive definite and D+K positive definite. D^(-1/2) is the actual inverse of CFC.sqrt D. With ||D^(-1/2) K D^(-1/2)||_2 <= eta <1, proves the condition product ||B||_2*||B^(-1)||_2 <=(1+eta)/(1-eta). Norms are the Euclidean operator norm. The inverse estimate does not require a separately assumed Hermitian perturbation.

- `KomorebiPaperVIII.invSqrt_isUnit`; `BlockJacobi.lean`, line 59.

- `KomorebiPaperVIII.invSqrt_normalizes`; `BlockJacobi.lean`, line 65.
- `KomorebiPaperVIII.normalized_decomposition`; `BlockJacobi.lean`, line 80.
- `KomorebiPaperVIII.block_jacobi_condition`; `BlockJacobi.lean`, line 87.

12.37 7.5 — Preconditioned conjugate gradient convergence

Traditional original item 7.5, current statement 7.5. Printed page 25, PDF page 31.

Recorded status. external_input_not_independently_formalized.

Recorded hypotheses and scope. Classical input tracked separately; see the boundary discussion.

12.38 7.6 — Certified packet preconditioning

Traditional original item 7.6, current statement 7.6. Printed page 25, PDF page 31.

Recorded status. checked.

Recorded hypotheses and scope. Monotonicity of the actual sqrt-based CG factor and its value at the normalized matrix condition number. Propagates an explicit classical energy-error estimate to the certified eta bound. Classical input 7.5 (CG convergence algorithm theorem) is not reproved or silently assumed as a new axiom.

- `KomorebiPaperVIII.matrix_condition_ge_one`; `BlockJacobi.lean`, line 97.
- `KomorebiPaperVIII.cgFactor_mono`; `BlockJacobi.lean`, line 106.
- `KomorebiPaperVIII.certified_packet_factor`; `BlockJacobi.lean`, line 126.
- `KomorebiPaperVIII.certified_energy_error`; `BlockJacobi.lean`, line 134.

12.39 8.1 — Packet perturbation certificate

Traditional original item 8.1, current statement 8.1. Printed page 26, PDF page 32.

Recorded status. checked.

Recorded hypotheses and scope. Actual continuous linear maps and their operator norm. Source support is assumed only for errors, a weaker sufficient hypothesis. Vector and uniform operator bounds. Not an IEEE/CUDA implementation verification.

- `KomorebiPaperVIII.packet_perturbation_bound`; `PacketPerturbation.lean`, line 35.
- `KomorebiPaperVIII.packet_relative_perturbation_bound`; `PacketPerturbation.lean`, line 42.
- `KomorebiPaperVIII.packet_operator_perturbation_bound`; `PacketPerturbation.lean`, line 51.
- `KomorebiPaperVIII.contractive_packet_operator_bound`; `PacketPerturbation.lean`, line 59.

12.40 8.2 — Floating point matrix product model

Traditional original item 8.2, current statement 8.2. Printed page 27, PDF page 33.

Recorded status. external_input_not_independently_formalized.

Recorded hypotheses and scope. Classical input tracked separately; see the boundary discussion.

12.41 8.3 — Mixed precision product chain

Traditional original item 8.3, current statement 8.3. Printed page 27, PDF page 33.

Recorded status. checked.

Recorded hypotheses and scope. Stagewise packet bounds are uniform operator-norm certificates on a nonzero normed space, not vector-specific bounds reused as operator norms. The exact-product perturbation theorem propagates those certificates. Vector error follows by multiplying the operator bound by the input norm.

- `KomorebiPaperVIII.mixed_precision_product_chain`; `ProductChainError.lean`, line 164.
- `KomorebiPaperVIII.chain_action_error`; `ProductChainError.lean`, line 180.

12.42 8.4 — Continuous precision allocation

Traditional original item 8.4, current statement 8.4. Printed page 27, PDF page 33.

Recorded status. checked.

Recorded hypotheses and scope. Finite nonempty edge set, strictly positive sensitivities, costs, and budget. Actual sum c*log(1/eps). Feasibility, global optimality, strict inequality away from formula, and unique minimizer. No KKT or convexity result is assumed.

- `KomorebiPaperVIII.optimal_tolerance_feasible`; `ContinuousPrecision.lean`, line 48.
- `KomorebiPaperVIII.precision_allocation_optimal`; `ContinuousPrecision.lean`, line 80.
- `KomorebiPaperVIII.precision_allocation_strict`; `ContinuousPrecision.lean`, line 100.
- `KomorebiPaperVIII.continuous_precision_unique_minimizer`; `ContinuousPrecision.lean`, line 131.

12.43 8.5 — Clipped precision allocation

Traditional original item 8.5, current statement 8.5. Printed page 28, PDF page 34.

Recorded status. checked.

Recorded hypotheses and scope. Finite nonempty set; strictly positive sensitivities, costs and caps; 0<budget<=total capped tolerance. Existence by IVT; global strict supporting-log inequality and unique minimizing vector. Unique positive multiplier only strictly below the maximum; endpoint nonuniqueness checked.

- `KomorebiPaperVIII.clipped_scale_exists`; `ClippedPrecision.lean`, line 74.
- `KomorebiPaperVIII.clipped_unique_minimizer`; `ClippedPrecision.lean`, line 211.
- `KomorebiPaperVIII.clipped_multiplier_exists_unique`; `ClippedPrecision.lean`, line 250.
- `KomorebiPaperVIII.clipped_endpoint_nonunique_multiplier`; `ClippedPrecision.lean`, line 244.

12.44 8.7 — Exact discrete precision program

Traditional original item 8.7, current statement 8.7. Printed page 29, PDF page 35.

Recorded status. checked.

Recorded hypotheses and scope. Independent legal assignments choose one listed option per edge. Nonnegative integer weights and exact ordered-field costs. Exact-weight states plus the final budget scan give the optimum with attainment or exact infeasibility. The residual-budget subtraction is

guarded. StoredBudgetCompiler proves the actual array-backed implementation equal to this specification.

- `KomorebiPaperVIII.BudgetDP.discrete_precision_program`; `DiscretePrecision.lean`, line 96.
- `KomorebiPaperVIII.BudgetDP.discreteBudget_some_iff`; `DiscretePrecision.lean`, line 110.
- `KomorebiPaperVIII.BudgetDP.discreteBudget_none_iff`; `DiscretePrecision.lean`, line 124.
- `KomorebiPaperVIII.BudgetDP.storedDiscreteBudget_correct`; `StoredBudgetCompiler.lean`, line 186.

12.45 9.1 — Simultaneous spectral decomposition

Traditional original item 9.1, current statement 9.1. Printed page 30, PDF page 36.

Recorded status. external_input_not_independently_formalized.

Recorded hypotheses and scope. Classical input tracked separately; see the boundary discussion.

12.46 9.2 — Exact block discovery certificate

Traditional original item 9.2, current statement 9.2. Printed page 30, PDF page 36.

Recorded status. checked.

Recorded hypotheses and scope. Actual finite complex matrices and arbitrary complete orthogonal projector family, not only standard coordinate projectors. Hilbert-Schmidt orthogonality via adjoint products and cyclic trace proves the squared Frobenius residual identity. Separate Euclidean operator-norm triangle inequality certificate. This certifies a supplied packet system, not autonomous discovery.

- `KomorebiPaperVIII.compression_residual`; `DiscoveryCertificate.lean`, line 49.
- `KomorebiPaperVIII.block_trace_orthogonal`; `DiscoveryCertificate.lean`, line 54.
- `KomorebiPaperVIII.discovery_frobenius_certificate`; `DiscoveryCertificate.lean`, line 126.
- `KomorebiPaperVIII.discovery_operator_certificate`; `DiscoveryCertificate.lean`, line 144.

12.47 9.4 — Exact Lagrangian packet compiler

Traditional original item 9.4, current statement 9.4. Printed page 31, PDF page 37.

Recorded status. checked.

Recorded hypotheses and scope. Arbitrary finite branching rooted trees, independent inductively defined cuts. Returned cut attains the recurrence and beats every legal cut. Generic ordered field covers rationals and reals. No universal discovery/approximation claim.

- `KomorebiPaperVIII.CandidateTree.exact_tree_compiler`; `ExactTreeCompiler.lean`, line 79.
- `KomorebiPaperVIII.CandidateTree.compile_attains`; `ExactTreeCompiler.lean`, line 69.
- `KomorebiPaperVIII.CandidateTree.exact_lagrangian_compiler`; `ExactTreeCompiler.lean`, line 102.

12.48 9.5 — Shared compilation for an operator family

Traditional original item 9.5, current statement 9.5. Printed page 31, PDF page 37.

Recorded status. checked.

Recorded hypotheses and scope. Exact weighted family aggregation at pruning level including setup and split overhead. Proved even without sign restrictions; intended application uses nonnegative weights.

- `KomorebiPaperVIII.CandidateTree.cutValue_weighted_family`; `ExactTreeCompiler.lean`, line 113.
- `KomorebiPaperVIII.CandidateTree.shared_family_compiler`; `ExactTreeCompiler.lean`, line 125.

12.49 9.6 — Monotone Lagrangian frontier

Traditional original item 9.6, current statement 9.6. Printed page 32, PDF page 38.

Recorded status. checked.

Recorded hypotheses and scope. Any two optimal cuts, 0 <= lambda1 < lambda2; error nonincreasing and cost nondecreasing. Optimality is discharged by the constructive compiler theorem for its returned cuts.

- `KomorebiPaperVIII.CandidateTree.monotone_lagrangian_frontier`; `ExactTreeCompiler.lean`, line 142.

12.50 9.7 — Supported Pareto points

Traditional original item 9.7, current statement 9.7. Printed page 32, PDF page 38.

Recorded status. checked.

Recorded hypotheses and scope. Positive scalarization parameter. Every true optimum is undominated. TreeCompilerUpdates.unsupported_pareto_example supplies the rational three-point unsupported-frontier inequality. Quantized-budget compiler remains unformalized.

- `KomorebiPaperVIII.CandidateTree.positive_lagrangian_pareto`; `ExactTreeCompiler.lean`, line 157.

12.51 9.8 — Quantized budget dynamic program

Traditional original item 9.8, current statement 9.8. Printed page 33, PDF page 39.

Recorded status. checked.

Recorded hypotheses and scope. The actual candidate tree and independent Cut type from the previous block, with arbitrary nonempty finite branching. Min-plus convolution over integer weights proves lower bounds, attainment, and infeasibility. StoredBudgetCompiler certifies the actual cached array-backed output. The implementation returns the minimum cost, not a backtracked cut.

- `KomorebiPaperVIII.CandidateTree.budgetTable_correct`; `QuantizedTreeCompiler.lean`, line 36.
- `KomorebiPaperVIII.CandidateTree.quantized_budget_compiler`; `QuantizedTreeCompiler.lean`, line 91.
- `KomorebiPaperVIII.CandidateTree.compileBudget_some_iff`; `QuantizedTreeCompiler.lean`, line 107.
- `KomorebiPaperVIII.CandidateTree.compileBudget_none_iff`; `QuantizedTreeCompiler.lean`, line 122.
- `KomorebiPaperVIII.CandidateTree.storedBudget_minimum`; `StoredBudgetCompiler.lean`, line 147.

12.52 9.9 — Pseudo polynomial complexity

Traditional original item 9.9, current statement 9.9. Printed page 33, PDF page 39.

Recorded status. checked.

Recorded hypotheses and scope. Array-backed tables cache child results. Concrete convolution candidate indices have at most (B+1)^2 elements. The declared unit-cost tabulation model is bounded by nodeCount*(3*(B+1)+(B+1)^2). Root minimum scan is an additional linear pass. This is pseudo-polynomial in numeric B and does not count exact-arithmetic bit complexity, compiler internals, or hardware instructions.

- `KomorebiPaperVIII.CandidateTree.merge_trial_count`; `StoredBudgetCompiler.lean`, line 222.
- `KomorebiPaperVIII.CandidateTree.merge_trial_bound`; `StoredBudgetCompiler.lean`, line 226.
- `KomorebiPaperVIII.CandidateTree.tabulation_pseudopolynomial`; `StoredBudgetCompiler.lean`, line 256.

12.53 9.11 — Compiler stability

Traditional original item 9.11, current statement 9.11. Printed page 33, PDF page 39.

Recorded status. checked.

Recorded hypotheses and scope. Actual solved keep/split scores satisfy strict half-gap inequalities at every internal node. Entire returned cut is unchanged. Ties cannot satisfy these strict hypotheses; no robustness at ties is claimed.

- `KomorebiPaperVIII.CandidateTree.half_gap_order`; `TreeCompilerUpdates.lean`, line 18.
- `KomorebiPaperVIII.CandidateTree.compiler_stability`; `TreeCompilerUpdates.lean`, line 45.

12.54 9.12 — Local recompilation

Traditional original item 9.12, current statement 9.12. Printed page 34, PDF page 40.

Recorded status. checked.

Recorded hypotheses and scope. Explicit one-hole context with cached sibling values. Replacing a subtree models localized data changes. Cache validity is constructed by refresh, not the new optimum. updateValue reads only new subtree score and caches on ancestor frames. Exact new global optimum and attaining cut proved.

- `KomorebiPaperVIII.CandidateTree.CachedContext.refresh_valid`; `TreeCompilerUpdates.lean`, line 95.
- `KomorebiPaperVIII.CandidateTree.CachedContext.updateValue_correct`; `TreeCompilerUpdates.lean`, line 109.
- `KomorebiPaperVIII.CandidateTree.CachedContext.sibling_tree_unchanged`; `TreeCompilerUpdates.lean`, line 128.
- `KomorebiPaperVIII.CandidateTree.CachedContext.sibling_value_unchanged`; `TreeCompilerUpdates.lean`, line 134.
- `KomorebiPaperVIII.CandidateTree.CachedContext.local_recompilation`; `TreeCompilerUpdates.lean`, line 142.

12.55 10.2 — Launch reduction by shape

Traditional original item 10.2, current statement 10.2. Printed page 35, PDF page 41.

Recorded status. checked.

Recorded hypotheses and scope. Finite grouping by complete shape signature, partition/fiberwise sum equals original output and work. One launch per active class is a backend capability assumption, not verified CUDA code.

- `KomorebiPaperVIII.shape_launch_count`; `SchedulingContracts.lean`, line 80.
- `KomorebiPaperVIII.grouped_execution_output`; `SchedulingContracts.lean`, line 89.
- `KomorebiPaperVIII.grouped_execution_work`; `SchedulingContracts.lean`, line 95.

12.56 10.3 — Acyclic schedule depth

Traditional original item 10.3, current statement 10.3. Printed page 35, PDF page 41.

Recorded status. checked.

Recorded hypotheses and scope. Finite acyclic task graph, unit positive-integer stages, dependency precedence, and unlimited concurrency. Constructed canonical level schedule is pointwise earliest and has minimal makespan; its depth is attained by an actual path when nonempty. Empty graph depth is zero. No weighted-task/resource-limited scheduling claim.

- `KomorebiPaperVIII.DAG.Graph.level_valid`; `GraphSchedules.lean`, line 126.
- `KomorebiPaperVIII.DAG.Graph.level_minimal`; `GraphSchedules.lean`, line 129.
- `KomorebiPaperVIII.DAG.Graph.optimal_depth`; `GraphSchedules.lean`, line 150.
- `KomorebiPaperVIII.DAG.Graph.depth_attained`; `GraphSchedules.lean`, line 164.
- `KomorebiPaperVIII.DAG.Graph.same_level_independent`; `GraphSchedules.lean`, line 181.
- `KomorebiPaperVIII.DAG.Graph.empty_depth`; `GraphSchedules.lean`, line 177.

12.57 10.4 — Amortization threshold

Traditional original item 10.4, current statement 10.4. Printed page 35, PDF page 41.

Recorded status. checked.

Recorded hypotheses and scope. Actual declared setup+R*candidate vs R*baseline inequality. Positive per-use advantage; setup nonnegative for smallest natural reuse floor+1. Equality is break-even, not profitability.

- `KomorebiPaperVIII.amortization_threshold`; `SchedulingContracts.lean`, line 101.
- `KomorebiPaperVIII.no_amortization`; `SchedulingContracts.lean`, line 107.
- `KomorebiPaperVIII.integer_amortization_threshold`; `SchedulingContracts.lean`, line 117.
- `KomorebiPaperVIII.minimum_reuse_attains`; `SchedulingContracts.lean`, line 125.

12.58 10.5 — Graph replay threshold

Traditional original item 10.5, current statement 10.5. Printed page 35, PDF page 41.

Recorded status. checked.

Recorded hypotheses and scope. Capture plus repeated replay vs repeated ordinary execution under replay<ordinary. Pure timing-model identity, not new GPU data.

- `KomorebiPaperVIII.graph_replay_threshold`; `SchedulingContracts.lean`, line 137.

12.59 10.6 — Materialization threshold

Traditional original item 10.6, current statement 10.6. Printed page 36, PDF page 42.

Recorded status. checked.

Recorded hypotheses and scope. Nonnegative materialization cost and reuse count. Strict profitability iff explicit per-use time is lower and reuse exceeds the ratio.

- `KomorebiPaperVIII.materialization_threshold`; `SchedulingContracts.lean`, line 142.

12.60 11.3 — Fair matching is an equivalence relation

Traditional original item 11.3, current statement 11.3. Printed page 36, PDF page 42.

Recorded status. checked.

Recorded hypotheses and scope. Equality of complete case, output, correctness reference, and timing-scope metadata. No inference of actual timing accuracy from matching metadata.

- `KomorebiPaperVIII.fair_match_iff_eq`; `SchedulingContracts.lean`, line 32.
- `KomorebiPaperVIII.fair_matching_equivalence`; `SchedulingContracts.lean`, line 40.

13 All named theorem and construction audits

The saved inventory contains 458 names: 392 theorems and 66 additional construction audits. The following catalog resolves namespaces from active source and records every audited name, its declaration kind, and its line. Theorem names include supporting lemmas and scope controls. Construction audits inspect transitive axioms of definitions, abbreviations, or structures. They are counted separately. The full source and the individual saved axiom sets are preserved in the accompanying files.

13.1 BandedCount: 9 theorems; 4 construction audits

Source: `formalization/BandedCount.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.bandCount_card`	theorem; 24
`KomorebiPaperVIII.finite_prefix_count`	theorem; 27
`KomorebiPaperVIII.bandCount_succ`	theorem; 38
`KomorebiPaperVIII.bandCount_full`	theorem; 50
`KomorebiPaperVIII.bandCount_balance`	theorem; 58
`KomorebiPaperVIII.banded_block_count`	theorem; 73
`KomorebiPaperVIII.banded_storage_count`	theorem; 85
`KomorebiPaperVIII.banded_operation_count`	theorem; 90
`KomorebiPaperVIII.banded_diagonal_case`	theorem; 96
`KomorebiPaperVIII.BandedAddress`	abbrev; 17
`KomorebiPaperVIII.BandedStorage`	abbrev; 80
`KomorebiPaperVIII.BandedOperation`	abbrev; 83
`KomorebiPaperVIII.bandCount`	def; 20

13.2 BlockJacobi: 12 theorems; 3 construction audits

Source: `formalization/BlockJacobi.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.inverse_near_one_bound`	theorem; 20
`KomorebiPaperVIII.near_one_condition_bound`	theorem; 37
`KomorebiPaperVIII.invSqrt_isUnit`	theorem; 59
`KomorebiPaperVIII.invSqrt_normalizes`	theorem; 65
`KomorebiPaperVIII.normalized_decomposition`	theorem; 80
`KomorebiPaperVIII.block_jacobi_condition`	theorem; 87
`KomorebiPaperVIII.matrix_condition_ge_one`	theorem; 97
`KomorebiPaperVIII.cgFactor_mono`	theorem; 106
`KomorebiPaperVIII.cgFactor_nonneg`	theorem; 114
`KomorebiPaperVIII.cgFactor_lt_one`	theorem; 119
`KomorebiPaperVIII.certified_packet_factor`	theorem; 126
`KomorebiPaperVIII.certified_energy_error`	theorem; 134
`KomorebiPaperVIII.invSqrt`	def; 55
`KomorebiPaperVIII.matrixCondition`	def; 57
`KomorebiPaperVIII.cgFactor`	def; 104

13.3 BoundaryChecks: 6 theorems; 0 construction audits

Source: `formalization/BoundaryChecks.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.cancellation_square_zero`	theorem; 29
`KomorebiPaperVIII.cancellation_block_nonzero`	theorem; 34
`KomorebiPaperVIII.paths_can_cancel`	theorem; 42
`KomorebiPaperVIII.positive_allocation_budget`	theorem; 52
`KomorebiPaperVIII.no_positive_allocation_nonpositive_budget`	theorem; 58
`KomorebiPaperVIII.one_edge_zero_budget_infeasible`	theorem; 66

13.4 BudgetConvolution: 10 theorems; 0 construction audits

Source: `formalization/BudgetConvolution.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.BudgetDP.minimum_congr`	theorem; 50
`KomorebiPaperVIII.BudgetDP.choose_correct`	theorem; 57
`KomorebiPaperVIII.BudgetDP.add_correct`	theorem; 78
`KomorebiPaperVIII.BudgetDP.bestOver_correct`	theorem; 98
`KomorebiPaperVIII.BudgetDP.convolution_correct`	theorem; 112
`KomorebiPaperVIII.BudgetDP.singletonTable_correct`	theorem; 136
`KomorebiPaperVIII.BudgetDP.combine_correct`	theorem; 148
`KomorebiPaperVIII.BudgetDP.withinBudget_correct`	theorem; 188
`KomorebiPaperVIII.BudgetDP.minimum_some_iff`	theorem; 197
`KomorebiPaperVIII.BudgetDP.minimum_none_iff`	theorem; 208

13.5 CirculantFourier: 13 theorems; 1 construction audits

Source: `formalization/CirculantFourier.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.fourier_scale_normalization`	theorem; 22
`KomorebiPaperVIII.fourier_scale_sqrt`	theorem; 33
`KomorebiPaperVIII.fourier_matrix_exponential`	theorem; 39
`KomorebiPaperVIII.fourier_action`	theorem; 49
`KomorebiPaperVIII.fourier_adjoint_entry`	theorem; 59
`KomorebiPaperVIII.fourier_adjoint_action`	theorem; 66
`KomorebiPaperVIII.fourier_unitary`	theorem; 83
`KomorebiPaperVIII.dft_circulant`	theorem; 105
`KomorebiPaperVIII.fourier_circulant_intertwine`	theorem; 128
`KomorebiPaperVIII.circulant_spectral_dft`	theorem; 138
`KomorebiPaperVIII.normalized_fourier_eigenvalues`	theorem; 145
`KomorebiPaperVIII.circulant_spectral_representation`	theorem; 150
`KomorebiPaperVIII.circulant_fourier_application`	theorem; 155

Audited declaration	Kind; line
KomorebiPaperVIII.fourierMatrix	def; 19

13.6 ClippedPrecision: 17 theorems; 2 construction audits

Source: formalization/ClippedPrecision.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.clipped_positive	theorem; 29
KomorebiPaperVIII.clipped_total_continuous	theorem; 34
KomorebiPaperVIII.clipped_total_mono	theorem; 38
KomorebiPaperVIII.clipped_total_zero	theorem; 47
KomorebiPaperVIII.saturation_scale_positive	theorem; 54
KomorebiPaperVIII.saturation_scale_attains	theorem; 60
KomorebiPaperVIII.clipped_scale_exists	theorem; 74
KomorebiPaperVIII.clipped_total_strict_below_cap	theorem; 95
KomorebiPaperVIII.clipped_scale_unique	theorem; 116
KomorebiPaperVIII.clipped_supporting_inequality	theorem; 131
KomorebiPaperVIII.clipped_relative_sum_le_zero	theorem; 149
KomorebiPaperVIII.clipped_allocation_optimal	theorem; 165
KomorebiPaperVIII.clipped_allocation_strict	theorem; 182
KomorebiPaperVIII.clipped_unique_minimizer	theorem; 211
KomorebiPaperVIII.clipped_multiplier_formula	theorem; 236
KomorebiPaperVIII.clipped_endpoint_nonunique_multiplier	theorem; 244
KomorebiPaperVIII.clipped_multiplier_exists_unique	theorem; 250
KomorebiPaperVIII.clippedTolerance	def; 19
KomorebiPaperVIII.saturationScale	def; 52

13.7 CompactRangeFactors: 7 theorems; 5 construction audits

Source: formalization/CompactRangeFactors.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.compactDim_eq_rank	theorem; 30
KomorebiPaperVIII.SmallCore.compactQ_apply	theorem; 37
KomorebiPaperVIII.SmallCore.compactQ_isometry	theorem; 42
KomorebiPaperVIII.SmallCore.compact_factorization	theorem; 51
KomorebiPaperVIII.SmallCore.compactR_is_compression	theorem; 60
KomorebiPaperVIII.SmallCore.compactR_full_row_rank	theorem; 67
KomorebiPaperVIII.SmallCore.compact_rank_zero	theorem; 77
KomorebiPaperVIII.SmallCore.columnBasis	def; 19
KomorebiPaperVIII.SmallCore.columnSpace	def; 16
KomorebiPaperVIII.SmallCore.compactDim	def; 18
KomorebiPaperVIII.SmallCore.compactQ	def; 22
KomorebiPaperVIII.SmallCore.compactR	def; 25

13.8 CompactTruncation: 5 theorems; 2 construction audits

Source: `formalization/CompactTruncation.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.SmallCore.keptColumn_card`	theorem; 16
`KomorebiPaperVIII.SmallCore.rootKept_tail_zero`	theorem; 23
`KomorebiPaperVIII.SmallCore.drop_zero_columns`	theorem; 31
`KomorebiPaperVIII.SmallCore.rootMask_square`	theorem; 46
`KomorebiPaperVIII.SmallCore.thin_factor_product`	theorem; 53
`KomorebiPaperVIII.SmallCore.KeptColumn`	abbrev; 14
`KomorebiPaperVIII.SmallCore.rootMask`	def; 19

13.9 ContinuousPrecision: 9 theorems; 1 construction audits

Source: `formalization/ContinuousPrecision.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.total_cost_positive`	theorem; 27
`KomorebiPaperVIII.optimal_tolerance_positive`	theorem; 31
`KomorebiPaperVIII.optimal_tolerance_budget`	theorem; 36
`KomorebiPaperVIII.optimal_tolerance_feasible`	theorem; 48
`KomorebiPaperVIII.relative_budget_identity`	theorem; 54
`KomorebiPaperVIII.precision_objective_difference`	theorem; 68
`KomorebiPaperVIII.precision_allocation_optimal`	theorem; 80
`KomorebiPaperVIII.precision_allocation_strict`	theorem; 100
`KomorebiPaperVIII.continuous_precision_unique_minimizer`	theorem; 131
`KomorebiPaperVIII.optimalTolerance`	def; 21

13.10 DiagonalTruncation: 9 theorems; 6 construction audits

Source: `formalization/DiagonalTruncation.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.SmallCore.kept_nonneg`	theorem; 24
`KomorebiPaperVIII.SmallCore.discarded_nonneg`	theorem; 27
`KomorebiPaperVIII.SmallCore.diagonal_residual`	theorem; 30
`KomorebiPaperVIII.SmallCore.rank_kept_le`	theorem; 38
`KomorebiPaperVIII.SmallCore.rootKept_selfAdjoint`	theorem; 53
`KomorebiPaperVIII.SmallCore.rootKept_square`	theorem; 57
`KomorebiPaperVIII.SmallCore.discarded_le_next`	theorem; 66
`KomorebiPaperVIII.SmallCore.spectral_diagonal_tail`	theorem; 80
`KomorebiPaperVIII.SmallCore.frobenius_diagonal_tail`	theorem; 104
`KomorebiPaperVIII.SmallCore.diag`	def; 15
`KomorebiPaperVIII.SmallCore.discarded`	def; 14
`KomorebiPaperVIII.SmallCore.kept`	def; 13

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.nextSingular	def; 17
KomorebiPaperVIII.SmallCore.rootKept	def; 21
KomorebiPaperVIII.SmallCore.tailEnergy	def; 19

13.11 DiscoveryCertificate: 8 theorems; 1 construction audits

Source: formalization/DiscoveryCertificate.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.compression_decomposition	theorem; 30
KomorebiPaperVIII.compression_residual	theorem; 49
KomorebiPaperVIII.block_trace_orthogonal	theorem; 54
KomorebiPaperVIII.offDiagonal_trace_orthogonal	theorem; 83
KomorebiPaperVIII.frobenius_sq_trace	theorem; 97
KomorebiPaperVIII.frobenius_sum_sq	theorem; 108
KomorebiPaperVIII.discovery_frobenius_certificate	theorem; 126
KomorebiPaperVIII.discovery_operator_certificate	theorem; 144
KomorebiPaperVIII.diagonalCompression	def; 24

13.12 DiscretePrecision: 8 theorems; 2 construction audits

Source: formalization/DiscretePrecision.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.BudgetDP.discreteTable_correct	theorem; 45
KomorebiPaperVIII.BudgetDP.discrete_precision_program	theorem; 96
KomorebiPaperVIII.BudgetDP.discreteBudget_some_iff	theorem; 110
KomorebiPaperVIII.BudgetDP.discreteBudget_none_iff	theorem; 124
KomorebiPaperVIII.BudgetDP.over_budget_option_rejected	theorem; 130
KomorebiPaperVIII.BudgetDP.zero_budget_option_accepted	theorem; 134
KomorebiPaperVIII.BudgetDP.empty_precision_options_rejected	theorem; 139
KomorebiPaperVIII.BudgetDP.empty_precision_problem	theorem; 143
KomorebiPaperVIII.BudgetDP.discreteTable	def; 38
KomorebiPaperVIII.BudgetDP.discreteBudget	def; 92

13.13 ExactTreeCompiler: 12 theorems; 2 construction audits

Source: formalization/ExactTreeCompiler.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.CandidateTree.optimalValue_le_cut	theorem; 57
KomorebiPaperVIII.CandidateTree.compile_attains	theorem; 69
KomorebiPaperVIII.CandidateTree.exact_tree_compiler	theorem; 79

Audited declaration	Kind; line
KomorebiPaperVIII.CandidateTree.compile_minimizes	theorem; 85
KomorebiPaperVIII.CandidateTree.lagrangian_decomposition	theorem; 92
KomorebiPaperVIII.CandidateTree.exact_lagrangian_compiler	theorem; 102
KomorebiPaperVIII.CandidateTree.cutValue_weighted_family	theorem; 113
KomorebiPaperVIII.CandidateTree.shared_family_compiler	theorem; 125
KomorebiPaperVIII.CandidateTree.monotone_lagrangian_frontier	theorem; 142
KomorebiPaperVIII.CandidateTree.positive_lagrangian_pareto	theorem; 157
KomorebiPaperVIII.CandidateTree.keep_decision_stable	theorem; 170
KomorebiPaperVIII.CandidateTree.split_decision_stable	theorem; 177
KomorebiPaperVIII.CandidateTree.compile	def; 50
KomorebiPaperVIII.CandidateTree.optimalValue	def; 45

13.14 FrobeniusOptimality: 5 theorems; 0 construction audits

Source: formalization/FrobeniusOptimality.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.tail_indicator_sum	theorem; 17
KomorebiPaperVIII.SmallCore.weighted_tail_lower	theorem; 35
KomorebiPaperVIII.SmallCore.diagonal_frame_energy	theorem; 65
KomorebiPaperVIII.SmallCore.diagonal_frobenius_lower	theorem; 76
KomorebiPaperVIII.SmallCore.diagonal_frobenius_best	theorem; 106

13.15 GraphSchedules: 17 theorems; 2 construction audits

Source: formalization/GraphSchedules.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.DAG.Graph.mem_ancestors	theorem; 32
KomorebiPaperVIII.DAG.Graph.ancestors_strict	theorem; 37
KomorebiPaperVIII.DAG.Graph.ancestors_card_lt	theorem; 51
KomorebiPaperVIII.DAG.Graph.wellFounded	theorem; 55
KomorebiPaperVIII.DAG.Graph.level_eq	theorem; 64
KomorebiPaperVIII.DAG.Graph.level_positive	theorem; 69
KomorebiPaperVIII.DAG.Graph.edge_level	theorem; 74
KomorebiPaperVIII.DAG.Graph.path_level_bound	theorem; 87
KomorebiPaperVIII.DAG.Graph.level_attained	theorem; 101
KomorebiPaperVIII.DAG.Graph.level_valid	theorem; 126
KomorebiPaperVIII.DAG.Graph.level_minimal	theorem; 129
KomorebiPaperVIII.DAG.Graph.level_le_depth	theorem; 146
KomorebiPaperVIII.DAG.Graph.optimal_depth	theorem; 150
KomorebiPaperVIII.DAG.Graph.path_length_le_depth	theorem; 156
KomorebiPaperVIII.DAG.Graph.depth_attained	theorem; 164
KomorebiPaperVIII.DAG.Graph.empty_depth	theorem; 177
KomorebiPaperVIII.DAG.Graph.same_level_independent	theorem; 181

Audited declaration	Kind; line
KomorebiPaperVIII.DAG.Graph.level	def; 59
KomorebiPaperVIII.DAG.Graph.depth	def; 144

13.16 InverseSupport: 9 theorems; 1 construction audits

Source: formalization/InverseSupport.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.reachable_refl	theorem; 27
KomorebiPaperVIII.reachable_step	theorem; 29
KomorebiPaperVIII.seed_mem_reachableSpace	theorem; 45
KomorebiPaperVIII.reachableSpace_invariant	theorem; 54
KomorebiPaperVIII.invariant_space_inverse	theorem; 73
KomorebiPaperVIII.inverse_block_zero_of_no_path	theorem; 94
KomorebiPaperVIII.inverse_support_inclusion	theorem; 102
KomorebiPaperVIII.unit_inverse_support	theorem; 108
KomorebiPaperVIII.matrix_inverse_support	theorem; 116
KomorebiPaperVIII.reachableSpace	def; 35

13.17 KernelFrame: 8 theorems; 5 construction audits

Source: formalization/KernelFrame.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.subspaceFrame_apply	theorem; 20
KomorebiPaperVIII.SmallCore.subspaceFrame_isometry	theorem; 26
KomorebiPaperVIII.SmallCore.kernelFrame_isometry	theorem; 40
KomorebiPaperVIII.SmallCore.mul_kernelFrame	theorem; 44
KomorebiPaperVIII.SmallCore.kernel_rank_nullity	theorem; 56
KomorebiPaperVIII.SmallCore.rowWeight_nonneg	theorem; 66
KomorebiPaperVIII.SmallCore.rowWeight_sum	theorem; 70
KomorebiPaperVIII.SmallCore.rowWeight_le_one	theorem; 83
KomorebiPaperVIII.SmallCore.kernelDim	def; 36
KomorebiPaperVIII.SmallCore.kernelFrame	def; 37
KomorebiPaperVIII.SmallCore.kernelSpace	def; 34
KomorebiPaperVIII.SmallCore.rowWeight	def; 63
KomorebiPaperVIII.SmallCore.subspaceFrame	def; 14

13.18 KrylovPackets: 11 theorems; 0 construction audits

Source: `formalization/KrylovPackets.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.packet_diagonal_commutes`	theorem; 31
`KomorebiPaperVIII.power_on_packet`	theorem; 50
`KomorebiPaperVIII.krylov_generator_mem`	theorem; 71
`KomorebiPaperVIII.invariant_packet_power_expansion`	theorem; 76
`KomorebiPaperVIII.invariant_packet_krylov`	theorem; 89
`KomorebiPaperVIII.local_krylov_le_range`	theorem; 102
`KomorebiPaperVIII.project_packet_sum`	theorem; 115
`KomorebiPaperVIII.local_krylov_direct`	theorem; 131
`KomorebiPaperVIII.power_difference_bound_succ`	theorem; 144
`KomorebiPaperVIII.approximate_packet_power_bound`	theorem; 170
`KomorebiPaperVIII.approximate_packet_krylov_distance`	theorem; 183

13.19 LowRankAlgebra: 8 theorems; 0 construction audits

Source: `formalization/LowRankAlgebra.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.packet_diagonal_product`	theorem; 33
`KomorebiPaperVIII.bdlr_product`	theorem; 57
`KomorebiPaperVIII.bdlr_product_rank_bound`	theorem; 68
`KomorebiPaperVIII.bdlr_apply`	theorem; 79
`KomorebiPaperVIII.bdlr_woodbury`	theorem; 89
`KomorebiPaperVIII.bdlr_update_isUnit`	theorem; 97
`KomorebiPaperVIII.bdlr_woodbury_solve`	theorem; 110
`KomorebiPaperVIII.bdlr_inverse_norm_bound`	theorem; 126

13.20 LowRankChains: 5 theorems; 1 construction audits

Source: `formalization/LowRankChains.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.chainColumns_card`	theorem; 34
`KomorebiPaperVIII.chainFactors_correct`	theorem; 56
`KomorebiPaperVIII.bdlr_chain_rank_bound`	theorem; 69
`KomorebiPaperVIII.packet_diagonal_one`	theorem; 82
`KomorebiPaperVIII.packet_diagonal_chain`	theorem; 87
`KomorebiPaperVIII.chainFactors`	def; 48

13.21 PacketCostModels: 22 theorems; 0 construction audits

Source: `formalization/PacketCostModels.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.packet_storage_count`	theorem; 27
`KomorebiPaperVIII.packet_application_count`	theorem; 31
`KomorebiPaperVIII.dense_storage_count`	theorem; 36
`KomorebiPaperVIII.dense_application_count`	theorem; 39
`KomorebiPaperVIII.equal_diagonal_storage`	theorem; 44
`KomorebiPaperVIII.equal_diagonal_work`	theorem; 48
`KomorebiPaperVIII.bdlr_storage_count`	theorem; 58
`KomorebiPaperVIII.bdlr_application_count`	theorem; 68
`KomorebiPaperVIII.equal_bdlr_storage`	theorem; 75
`KomorebiPaperVIII.equal_bdlr_work`	theorem; 81
`KomorebiPaperVIII.equal_block_reduction`	theorem; 87
`KomorebiPaperVIII.equal_bdlr_reduction`	theorem; 91
`KomorebiPaperVIII.streaming_value_count`	theorem; 102
`KomorebiPaperVIII.streaming_byte_count`	theorem; 111
`KomorebiPaperVIII.roofline_time_bound`	theorem; 118
`KomorebiPaperVIII.roofline_work_rate`	theorem; 126
`KomorebiPaperVIII.dense_capacity_threshold`	theorem; 139
`KomorebiPaperVIII.bdlr_capacity_threshold`	theorem; 143
`KomorebiPaperVIII.materialization_work_count`	theorem; 151
`KomorebiPaperVIII.materialization_quadratic_remainder`	theorem; 156
`KomorebiPaperVIII.sparse_low_rank_storage_count`	theorem; 164
`KomorebiPaperVIII.sparse_low_rank_work_count`	theorem; 169

13.22 PacketIR: 11 theorems; 1 construction audits

Source: `formalization/PacketIR.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.active_iff`	theorem; 35
`KomorebiPaperVIII.exact_active_evaluation`	theorem; 41
`KomorebiPaperVIII.exact_active_action`	theorem; 52
`KomorebiPaperVIII.exact_block_composition`	theorem; 58
`KomorebiPaperVIII.product_support_inclusion`	theorem; 63
`KomorebiPaperVIII.block_sum`	theorem; 79
`KomorebiPaperVIII.all_blocks_zero`	theorem; 83
`KomorebiPaperVIII.exact_adjoint_block`	theorem; 95
`KomorebiPaperVIII.coordinate_projector_apply`	theorem; 129
`KomorebiPaperVIII.exact_tensor_block`	theorem; 135
`KomorebiPaperVIII.exact_matrix_batch`	theorem; 145
`KomorebiPaperVIII.coordinatePackets`	def; 106

13.23 PacketPerturbation: 6 theorems; 0 construction audits

Source: `formalization/PacketPerturbation.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.packet_error_operator_identity`	theorem; 21
`KomorebiPaperVIII.packet_error_action_identity`	theorem; 28
`KomorebiPaperVIII.packet_perturbation_bound`	theorem; 35
`KomorebiPaperVIII.packet_relative_perturbation_bound`	theorem; 42
`KomorebiPaperVIII.packet_operator_perturbation_bound`	theorem; 51
`KomorebiPaperVIII.contractive_packet_operator_bound`	theorem; 59

13.24 ProductChainError: 13 theorems; 0 construction audits

Source: `formalization/ProductChainError.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.propagatedError_eq_sum`	theorem; 37
`KomorebiPaperVIII.propagatedError_mono`	theorem; 53
`KomorebiPaperVIII.norm_recurrence_bound`	theorem; 62
`KomorebiPaperVIII.product_discrepancy_step`	theorem; 75
`KomorebiPaperVIII.recompression_discrepancy_step`	theorem; 83
`KomorebiPaperVIII.chainProduct_const`	theorem; 90
`KomorebiPaperVIII.repeated_recompression_bound`	theorem; 100
`KomorebiPaperVIII.repeated_recompression_actual_error`	theorem; 114
`KomorebiPaperVIII.chainProduct_norm_bound`	theorem; 120
`KomorebiPaperVIII.product_chain_error_bound`	theorem; 129
`KomorebiPaperVIII.product_chain_perturbation`	theorem; 148
`KomorebiPaperVIII.mixed_precision_product_chain`	theorem; 164
`KomorebiPaperVIII.chain_action_error`	theorem; 180

13.25 QuantizedTreeCompiler: 5 theorems; 2 construction audits

Source: `formalization/QuantizedTreeCompiler.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.CandidateTree.budgetTable_correct`	theorem; 36
`KomorebiPaperVIII.CandidateTree.quantized_budget_compiler`	theorem; 91
`KomorebiPaperVIII.CandidateTree.compileBudget_some_iff`	theorem; 107
`KomorebiPaperVIII.CandidateTree.compileBudget_none_iff`	theorem; 122
`KomorebiPaperVIII.CandidateTree.compileBudget_monotone`	theorem; 129
`KomorebiPaperVIII.CandidateTree.budgetTable`	def; 27
`KomorebiPaperVIII.CandidateTree.compileBudget`	def; 86

13.26 QuantumPacketCore: 19 theorems; 0 construction audits

Source: `formalization/QuantumPacketCore.lean`.

Audited declaration	Kind; line
`KomorebiPaperVII.PacketFamily.idempotent`	theorem; 32
`KomorebiPaperVII.PacketFamily.reconstruct`	theorem; 38
`KomorebiPaperVII.PacketFamily.compose_blocks`	theorem; 41
`KomorebiPaperVII.PacketFamily.circuit_path_expansion`	theorem; 63
`KomorebiPaperVII.PacketFamily.circuit_reconstruction`	theorem; 70
`KomorebiPaperVII.PacketFamily.symmetry_no_transfer`	theorem; 76
`KomorebiPaperVII.PacketFamily.symmetry_diagonal`	theorem; 80
`KomorebiPaperVII.PacketFamily.dephase_idempotent`	theorem; 90
`KomorebiPaperVII.PacketFamily.adjoint_block`	theorem; 114
`KomorebiPaperVII.PacketFamily.unitarity_columns`	theorem; 119
`KomorebiPaperVII.PacketFamily.unitarity_rows`	theorem; 134
`KomorebiPaperVII.PacketFamily.symmetry_exponential_commutes`	theorem; 156
`KomorebiPaperVII.PacketFamily.symmetry_exponential_diagonal`	theorem; 163
`KomorebiPaperVII.PacketFamily.symmetry_exponential_no_transfer`	theorem; 168
`KomorebiPaperVII.matrix_unitarity_columns`	theorem; 181
`KomorebiPaperVII.matrix_unitarity_rows`	theorem; 188
`KomorebiPaperVII.dephase_trace_duality`	theorem; 195
`KomorebiPaperVII.measurement_invariance`	theorem; 208
`KomorebiPaperVII.dephase_trace_preserving`	theorem; 213

13.27 RecompressionEndpoints: 6 theorems; 2 construction audits

Source: `formalization/RecompressionEndpoints.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.SmallCore.kept_zero`	theorem; 40
`KomorebiPaperVIII.SmallCore.kept_all`	theorem; 44
`KomorebiPaperVIII.SmallCore.tailEnergy_zero_of_full`	theorem; 49
`KomorebiPaperVIII.SmallCore.nextSingular_zero_of_full`	theorem; 57
`KomorebiPaperVIII.SmallCore.recompression_rank_zero`	theorem; 62
`KomorebiPaperVIII.SmallCore.recompression_full`	theorem; 70
`KomorebiPaperVIII.SmallCore.diagonalSVD`	def; 18
`KomorebiPaperVIII.SmallCore.zeroSVD`	def; 30

13.28 SchedulingContracts: 17 theorems; 0 construction audits

Source: `formalization/SchedulingContracts.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.fair_match_iff_eq`	theorem; 32
`KomorebiPaperVIII.fair_matching_equivalence`	theorem; 40

Audited declaration	Kind; line
KomorebiPaperVIII.contract_conversion	theorem; 54
KomorebiPaperVIII.direct_compressed_consumption	theorem; 62
KomorebiPaperVIII.omitted_decode_understates_time	theorem; 69
KomorebiPaperVIII.shape_launch_count	theorem; 80
KomorebiPaperVIII.each_block_has_shape	theorem; 84
KomorebiPaperVIII.grouped_execution_output	theorem; 89
KomorebiPaperVIII.grouped_execution_work	theorem; 95
KomorebiPaperVIII.amortization_threshold	theorem; 101
KomorebiPaperVIII.no_amortization	theorem; 107
KomorebiPaperVIII.integer_amortization_threshold	theorem; 117
KomorebiPaperVIII.minimum_reuse_attains	theorem; 125
KomorebiPaperVIII.graph_replay_threshold	theorem; 137
KomorebiPaperVIII.materialization_threshold	theorem; 142
KomorebiPaperVIII.zero_setup_first_use	theorem; 155
KomorebiPaperVIII.exact_break_even_regression	theorem; 159

13.29 SchurSupport: 4 theorems; 0 construction audits

Source: formalization/SchurSupport.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.schur_correction_expansion	theorem; 25
KomorebiPaperVIII.schur_correction_support	theorem; 33
KomorebiPaperVIII.schur_support_inclusion	theorem; 57
KomorebiPaperVIII.schur_zero_of_no_route	theorem; 76

13.30 SmallCoreNorms: 19 theorems; 4 construction audits

Source: formalization/SmallCoreNorms.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.compress_lift	theorem; 25
KomorebiPaperVIII.SmallCore.lift_sub	theorem; 33
KomorebiPaperVIII.SmallCore.compress_sub	theorem; 38
KomorebiPaperVIII.SmallCore.rank_compress_le	theorem; 43
KomorebiPaperVIII.SmallCore.rank_lift_le	theorem; 48
KomorebiPaperVIII.SmallCore.rank_lift	theorem; 53
KomorebiPaperVIII.SmallCore.spectral_left_isometry	theorem; 64
KomorebiPaperVIII.SmallCore.spectral_right_isometry	theorem; 73
KomorebiPaperVIII.SmallCore.spectral_isometry_bound	theorem; 82
KomorebiPaperVIII.SmallCore.spectral_lift	theorem; 90
KomorebiPaperVIII.SmallCore.spectral_compress_le	theorem; 96
KomorebiPaperVIII.SmallCore.frobenius_rect_sq_trace	theorem; 117
KomorebiPaperVIII.SmallCore.frobenius_left_isometry	theorem; 128
KomorebiPaperVIII.SmallCore.frobenius_right_isometry	theorem; 138

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.frobenius_projection_pythagoras	theorem; 148
KomorebiPaperVIII.SmallCore.frobenius_left_compress_le	theorem; 163
KomorebiPaperVIII.SmallCore.frobenius_right_compress_le	theorem; 169
KomorebiPaperVIII.SmallCore.frobenius_lift	theorem; 178
KomorebiPaperVIII.SmallCore.frobenius_compress_le	theorem; 185
KomorebiPaperVIII.SmallCore.compress	def; 21
KomorebiPaperVIII.SmallCore.frobeniusNorm	def; 114
KomorebiPaperVIII.SmallCore.lift	def; 18
KomorebiPaperVIII.SmallCore.spectralNorm	def; 61

13.31 SmallCoreRecompression: 6 theorems; 3 construction audits

Source: formalization/SmallCoreRecompression.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.isometry_comp	theorem; 40
KomorebiPaperVIII.SmallCore.lift_comp	theorem; 47
KomorebiPaperVIII.SmallCore.recompressed_product	theorem; 64
KomorebiPaperVIII.SmallCore.core_gram_eigenvectors	theorem; 72
KomorebiPaperVIII.SmallCore.recompression_optimality	theorem; 91
KomorebiPaperVIII.SmallCore.small_core_optimal_recompression	theorem; 124
KomorebiPaperVIII.SmallCore.OrderedThinSVD	structure; 29
KomorebiPaperVIII.SmallCore.recompressLeft	def; 54
KomorebiPaperVIII.SmallCore.recompressRight	def; 59

13.32 SmallCoreTransfer: 7 theorems; 2 construction audits

Source: formalization/SmallCoreTransfer.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.SmallCore.compressed_error	theorem; 20
KomorebiPaperVIII.SmallCore.spectral_best_lift	theorem; 25
KomorebiPaperVIII.SmallCore.frobenius_best_lift	theorem; 38
KomorebiPaperVIII.SmallCore.exact_factor_core	theorem; 51
KomorebiPaperVIII.SmallCore.compact_product_reconstruction	theorem; 61
KomorebiPaperVIII.SmallCore.compact_core_rank	theorem; 69
KomorebiPaperVIII.SmallCore.factorized_truncation	theorem; 77
KomorebiPaperVIII.SmallCore.BestRank	def; 16
KomorebiPaperVIII.SmallCore.factorCore	def; 57

13.33 SpectralOptimality: 7 theorems; 1 construction audits

Source: `formalization/SpectralOptimality.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.SmallCore.prefix_isometry`	theorem; 20
`KomorebiPaperVIII.SmallCore.prefix_support`	theorem; 25
`KomorebiPaperVIII.SmallCore.rank_lt_kernel_witness`	theorem; 38
`KomorebiPaperVIII.SmallCore.prefix_kernel_witness`	theorem; 55
`KomorebiPaperVIII.SmallCore.prefix_diagonal_lower`	theorem; 87
`KomorebiPaperVIII.SmallCore.diagonal_spectral_lower`	theorem; 103
`KomorebiPaperVIII.SmallCore.diagonal_spectral_best`	theorem; 120
`KomorebiPaperVIII.SmallCore.prefixEmbedding`	def; 16

13.34 StoredBudgetCompiler: 16 theorems; 2 construction audits

Source: `formalization/StoredBudgetCompiler.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.BudgetDP.vector_get_ofFn`	theorem; 34
`KomorebiPaperVIII.BudgetDP.lookup_tabulate`	theorem; 45
`KomorebiPaperVIII.BudgetDP.bestOver_congr`	theorem; 50
`KomorebiPaperVIII.BudgetDP.convolution_congr_local`	theorem; 60
`KomorebiPaperVIII.BudgetDP.combine_congr_local`	theorem; 69
`KomorebiPaperVIII.BudgetDP.combineStored_correct`	theorem; 92
`KomorebiPaperVIII.CandidateTree.storedTree_correct`	theorem; 120
`KomorebiPaperVIII.CandidateTree.storedBudget_correct`	theorem; 140
`KomorebiPaperVIII.CandidateTree.storedBudget_minimum`	theorem; 147
`KomorebiPaperVIII.BudgetDP.storedDiscrete_correct`	theorem; 167
`KomorebiPaperVIII.BudgetDP.storedDiscreteBudget_correct`	theorem; 186
`KomorebiPaperVIII.CandidateTree.mergeCount_add_one`	theorem; 207
`KomorebiPaperVIII.CandidateTree.merge_trial_count`	theorem; 222
`KomorebiPaperVIII.CandidateTree.merge_trial_bound`	theorem; 226
`KomorebiPaperVIII.CandidateTree.tabulation_work_bound`	theorem; 242
`KomorebiPaperVIII.CandidateTree.tabulation_pseudopolynomial`	theorem; 256
`KomorebiPaperVIII.CandidateTree.storedTree`	def; 109
`KomorebiPaperVIII.BudgetDP.storedDiscrete`	def; 159

13.35 StructuredTensor: 7 theorems; 3 construction audits

Source: `formalization/StructuredTensor.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.kronecker_modewise`	theorem; 19
`KomorebiPaperVIII.kronecker_modewise_work`	theorem; 27
`KomorebiPaperVIII.kronecker_factor_storage`	theorem; 32

Audited declaration	Kind; line
KomorebiPaperVIII.tensor_dimension	theorem; 51
KomorebiPaperVIII.tensor_apply_correct	theorem; 73
KomorebiPaperVIII.tensor_modewise_work	theorem; 112
KomorebiPaperVIII.tensor_factor_storage	theorem; 126
KomorebiPaperVIII.tensorMatrix	def; 59
KomorebiPaperVIII.tensorApply	def; 66
KomorebiPaperVIII.TensorOperation	def; 99

13.36 ToeplitzEmbedding: 6 theorems; 2 construction audits

Source: formalization/ToeplitzEmbedding.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.toeplitz_circulant_block	theorem; 28
KomorebiPaperVIII.zero_pad_left	theorem; 45
KomorebiPaperVIII.zero_pad_right	theorem; 50
KomorebiPaperVIII.toeplitz_embedding_action	theorem; 54
KomorebiPaperVIII.toeplitz_generator_first_half	theorem; 60
KomorebiPaperVIII.toeplitz_generator_middle	theorem; 64
KomorebiPaperVIII.toeplitzGenerator	def; 22
KomorebiPaperVIII.zeroPad	def; 26

13.37 TransferSupport: 10 theorems; 0 construction audits

Source: formalization/TransferSupport.lean.

Audited declaration	Kind; line
KomorebiPaperVIII.power_path_expansion	theorem; 37
KomorebiPaperVIII.active_path_sum_zero	theorem; 55
KomorebiPaperVIII.power_block_zero_of_no_path	theorem; 74
KomorebiPaperVIII.power_support_inclusion	theorem; 79
KomorebiPaperVIII.nilpotent_of_path_bound	theorem; 86
KomorebiPaperVIII.path_level_bound	theorem; 93
KomorebiPaperVIII.nilpotent_of_topological_levels	theorem; 109
KomorebiPaperVIII.block_smul	theorem; 121
KomorebiPaperVIII.polynomial_support_inclusion	theorem; 127
KomorebiPaperVIII.polynomial_zero_of_no_paths	theorem; 142

13.38 TreeCompilerUpdates: 9 theorems; 1 construction audits

Source: `formalization/TreeCompilerUpdates.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.CandidateTree.half_gap_order`	theorem; 18
`KomorebiPaperVIII.CandidateTree.compiler_stability`	theorem; 45
`KomorebiPaperVIII.CandidateTree.CachedContext.refresh_valid`	theorem; 95
`KomorebiPaperVIII.CandidateTree.CachedContext.refresh_fill`	theorem; 102
`KomorebiPaperVIII.CandidateTree.CachedContext.updateValue_correct`	theorem; 109
`KomorebiPaperVIII.CandidateTree.CachedContext.sibling_tree_unchanged`	theorem; 128
`KomorebiPaperVIII.CandidateTree.CachedContext.sibling_value_unchanged`	theorem; 134
`KomorebiPaperVIII.CandidateTree.CachedContext.local_recompilation`	theorem; 142
`KomorebiPaperVIII.CandidateTree.unsupported_pareto_example`	theorem; 155
`KomorebiPaperVIII.CandidateTree.CachedContext.updateValue`	def; 75

13.39 TriangularSolve: 7 theorems; 3 construction audits

Source: `formalization/TriangularSolve.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.DAG.Graph.solve_eq`	theorem; 32
`KomorebiPaperVIII.DAG.Graph.solve_correct`	theorem; 38
`KomorebiPaperVIII.DAG.Graph.solve_unique`	theorem; 46
`KomorebiPaperVIII.DAG.Graph.triangular_bijective`	theorem; 65
`KomorebiPaperVIII.DAG.Graph.local_solve_independent`	theorem; 75
`KomorebiPaperVIII.DAG.Graph.stagedSolve_correct_through`	theorem; 91
`KomorebiPaperVIII.DAG.Graph.level_schedule_solves`	theorem; 112
`KomorebiPaperVIII.DAG.Graph.triangularAction`	def; 24
`KomorebiPaperVIII.DAG.Graph.solve`	def; 28
`KomorebiPaperVIII.DAG.Graph.stagedSolve`	def; 85

13.40 WalshTransform: 7 theorems; 2 construction audits

Source: `formalization/WalshTransform.lean`.

Audited declaration	Kind; line
`KomorebiPaperVIII.walsh_two_orthogonal`	theorem; 21
`KomorebiPaperVIII.walsh_recursion`	theorem; 29
`KomorebiPaperVIII.walsh_orthogonal`	theorem; 33
`KomorebiPaperVIII.walsh_apply_correct`	theorem; 60
`KomorebiPaperVIII.walsh_operation_count`	theorem; 99
`KomorebiPaperVIII.walsh_inverse_action`	theorem; 110
`KomorebiPaperVIII.walsh_log_operation_count`	theorem; 115
`KomorebiPaperVIII.walshMatrix`	def; 25
`KomorebiPaperVIII.walshApply`	def; 54

14 Dependencies and source identities

Both public imports and import-all commands contribute to this graph. Import-all exposes implementation details needed by downstream proofs in these source modules. Each identity below is checked against the corresponding saved compiler record and log. The runtime and imported Mathlib corpus are dependencies of the inherited evidence, rather than newly rechecked results.

14.1 Direct local dependencies

Module	Direct local imports
BandedCount	Mathlib / Init only
BlockJacobi	Mathlib / Init only
BoundaryChecks	PacketIR; QuantumPacketCore; TransferSupport
BudgetConvolution	Mathlib / Init only
CirculantFourier	Mathlib / Init only
ClippedPrecision	ContinuousPrecision
CompactRangeFactors	SmallCoreNorms
CompactTruncation	DiagonalTruncation; SmallCoreNorms
ContinuousPrecision	Mathlib / Init only
DiagonalTruncation	SmallCoreNorms; SmallCoreTransfer
DiscoveryCertificate	PacketIR; QuantumPacketCore
DiscretePrecision	BudgetConvolution
ExactTreeCompiler	Mathlib / Init only
FrobeniusOptimality	DiagonalTruncation; KernelFrame; SmallCoreNorms; SmallCoreTransfer
GraphSchedules	Mathlib / Init only
InverseSupport	PacketIR; QuantumPacketCore; TransferSupport
KernelFrame	SmallCoreNorms
KrylovPackets	LowRankAlgebra; LowRankChains; ProductChainError; QuantumPacketCore
LowRankAlgebra	PacketIR; QuantumPacketCore
LowRankChains	LowRankAlgebra; PacketIR; ProductChainError; QuantumPacketCore
PacketCostModels	Mathlib / Init only
PacketIR	QuantumPacketCore
PacketPerturbation	Mathlib / Init only
ProductChainError	PacketPerturbation
QuantizedTreeCompiler	BudgetConvolution; ExactTreeCompiler
QuantumPacketCore	Mathlib / Init only
RecompressionEndpoints	CompactTruncation; DiagonalTruncation; SmallCoreNorms; SmallCoreRecompression; SmallCoreTransfer
SchedulingContracts	Mathlib / Init only
SchurSupport	InverseSupport; PacketIR; QuantumPacketCore
SmallCoreNorms	Mathlib / Init only
SmallCoreRecompression	CompactRangeFactors; CompactTruncation; DiagonalTruncation; FrobeniusOptimality; KernelFrame; SmallCoreNorms; SmallCoreTransfer; SpectralOptimality
SmallCoreTransfer	CompactRangeFactors; SmallCoreNorms
SpectralOptimality	DiagonalTruncation; KernelFrame; SmallCoreNorms; SmallCoreTransfer

Module	Direct local imports
StoredBudgetCompiler	BudgetConvolution; DiscretePrecision; ExactTreeCompiler; QuantizedTreeCompiler
StructuredTensor	Mathlib / Init only
ToeplitzEmbedding	Mathlib / Init only
TransferSupport	PacketIR; QuantumPacketCore
TreeCompilerUpdates	ExactTreeCompiler
TriangularSolve	GraphSchedules
WalshTransform	StructuredTensor

14.2 Source and compiler-log identities

BandedCount

Source SHA-256: b13124a6d12cdd0589895d3d12c07da364fd47976e28b2c9541b8828bb291a0d.

Log SHA-256: 8fe4bd93789d7d03432b770afeb52e31f2fe4802b84a7b1ac096df3f76fa1721.

BlockJacobi

Source SHA-256: 0b5b13491150bdab13be5cc72e6fb0040f39f06cc06398095c360bcb632ced71.

Log SHA-256: 208d359ecf6a62cc2e0ea02c60f340ab7acae41cbf8caf710e12582627d9f404.

BoundaryChecks

Source SHA-256: 18a64ef9bdd427697b5fc70417ff123dc9aed37a74be2fcc8cfc30b509ce32b9.

Log SHA-256: 952e994cd8e9e81a93cdb61b2d8d8b2c87407c23274f97fb9b96357e9ae82969.

BudgetConvolution

Source SHA-256: e6421dc70c6050e77e5d6c8987fa7dd515ef5d2e1c0ef10c3a98b2ea18344fd3.

Log SHA-256: d1fe718701677b79398cb32bcda5af5b3f1eb49456a0e65c45fd13f993747765.

CirculantFourier

Source SHA-256: 8db2cde5eec1565ff44c496d4e73a49a2ef3d28a5afc2e1ca0874ad25af3b02c.

Log SHA-256: 14c85764e294596e7ad47896ed2baf8a0e1d45d9c21f2b4385996b32ba21e722.

ClippedPrecision

Source SHA-256: aa86e409a530f0916da245bfe3cf81b7fba26e97c12f8fd7a7690351a08dffc9.

Log SHA-256: 35d8dd8e973deb2043deaaa73ad1c463477131541231f6c9ae3867b41e0a16f5.

CompactRangeFactors

Source SHA-256: 3ae44a0f80503e8455cc2cd4ad7de140e631c59e32a109cd5fae2366b34d2e1c.

Log SHA-256: 6840ecbef5562873065f69d71efdfbd8d1dd63e6c70dccb3bb1aff04e1413ed9.

CompactTruncation

Source SHA-256: e387ce9da3cf16879225207523cc0923ae5adaaa1c3e317c447b4923ec40b6a2.

Log SHA-256: 0162357f46179fd8d1c70d3ed0488c1cff7cf9290e6ac12be299e38e322def53.

ContinuousPrecision

Source SHA-256: 5a737f5aa8ed23525c225b48fc6cef46a623fe85b7ddf8ee4bf9115976667230.

Log SHA-256: 1473e51a2a09550739223cfe7c2ce4576daa3abf4ff700e9bc347ce7e4ae04ab.

DiagonalTruncation

Source SHA-256: 21594e839e131b8ae6a5ef4fe06a1640fdf2ef03e5b745985c3e5f21429809ad.

Log SHA-256: f689d98f40296f56dad426161e3636e287d3cfa624a4c885398a916a58009afb.

DiscoveryCertificate

Source SHA-256: `ae8a69d0022cb8258a75d91b5e7e98c9a4f0f60d4d67f2d976b093d8a5987e8f`.

Log SHA-256: `5fa99bbe29375b3e7ee48d27a08362201829200e888dcd37829334d7b74b2ec8`.

DiscretePrecision

Source SHA-256: `0b12bf6ecf1c9bc36bc735ee737c06db5a37651d12dca3ffcad230493bb7a654`.

Log SHA-256: `0a45c7a80f06fe0f55002efd9abc600a469d78d6242283eb14b6ee54dd293316`.

ExactTreeCompiler

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Log SHA-256: `8f4156453fc9298a64518da7dba7178c606d2269a7c3f47c2db53fa534c4c66d`.

FrobeniusOptimality

Source SHA-256: `3d28205f0088a2ebc9ca78141cefb73d4d14bdb4f1feb9cb860968cfa9b29f87`.

Log SHA-256: `576c50d4d646d0a75e382953b4e4c9cd1fcca2aa0763abe6bcf8125080698785`.

GraphSchedules

Source SHA-256: `1b845790a59fdf87562a2afa5dd56cda8ee0d65ae4cfbb8a1259364d4a01d4cf`.

Log SHA-256: `a8ce351f7fabb514b6a3de6cda7498541ca7491294274f6756f252b6abd46a6d`.

InverseSupport

Source SHA-256: `e8214096eab227677cdd7977840b02844bd0ec706df57dca405bcfae2f6e2063`.

Log SHA-256: `104d9d3d784e307a6e864540a07f4f60259b60b5d8b3765bee1919e38c7b0d53`.

KernelFrame

Source SHA-256: `dc4fd08f29d82779453ad49007b38339423f6c09aec0393eca5b56f0674af90c`.

Log SHA-256: `b5c78b4e160d3f33b1b1bb59ab2a222f6642f8cb034b163d3d483f46b7ba8548`.

KrylovPackets

Source SHA-256: `12fd73f08368ec528953e66e165ef4281f1a978052c21e0c61dbde6d12c508b5`.

Log SHA-256: `4b7a9affc66c399b35a54e9aca5483e78b99acf970cbb9435a029098fe6d5044`.

LowRankAlgebra

Source SHA-256: `61e29902b761b36930587c604cee5740c16fbb076fbde070df7ba80ce0c7984d`.

Log SHA-256: `d274abfbfd96bb5ece7a700a42eeb3bf7b0fbfbdcb9a54f569860b8b144653db`.

LowRankChains

Source SHA-256: `970c35e0219c27dedaf97e215a9868ebd4e6015b7f736b9dc0aea385b592212b`.

Log SHA-256: `c72865e5ff284eaba62955b9daa8f9da87a7589e81b6097de6af72978e2c8da4`.

PacketCostModels

Source SHA-256: `77aa5594485ec62c259e1a1d9493874481e72ceaecd6c2b6f654f42474f999b5`.

Log SHA-256: `2d53dec907e3286a529bb2de82d6cbc3f800a0ca31e01f91fb89505962623a88`.

PacketIR

Source SHA-256: `67e2e088b17bb95162585592da91c1ed506c5faf7f8d6574cc8b3af8ff0e47c7`.

Log SHA-256: `ab2bc3c6cbc4cce4f38ea6a4c285c596f496fb5759bb3f32c8ce7bb2db83b746`.

PacketPerturbation

Source SHA-256: `5fbb1026da706b46afaf2793e138fe503e00ed37a470c193388ad6e79a37bd12`.

Log SHA-256: `649c81ea6be13008f2b5d21194eaac3e67ac8915b270a5b3278298417d1b6411`.

ProductChainError

Source SHA-256: `5bad0d7ed04be1a94bb19e54c9371e3aea0dc69ff90e4afc33c6cdc0e4686a2c`.

Log SHA-256: `0634b1ccd6522518ee87aa2a00d71b1bcd9e44723ae646645c186c5d03b9121b`.

QuantizedTreeCompiler

Source SHA-256: `564ddda43334a4cd506e7bf990704b4fc604eeccb891084a223cae53b96268e4`.

Log SHA-256: `f88e40d19ee915fd044cc719c48d746bfb2b7b12c25a58c8f4d338d2aa1c5b2f`.

QuantumPacketCore

Source SHA-256: `92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595`.

Log SHA-256: `cde10a754cab43da66a12045d11df455573b5cc3f647497d29a54359e39f5798`.

RecompressionEndpoints

Source SHA-256: `d8e8dc3290e938198c524607904db67869659461b852964e46297de4690a3789`.

Log SHA-256: `f08cd5b88cff2be28ff88b1fb9923c8dbe6a3003e0a92c82bd0919cfc1c67e19`.

SchedulingContracts

Source SHA-256: `ea646fab14df1c6f6b4855745035b64835920a75fe2e66d168cb515f74854160`.

Log SHA-256: `79f97910adb6bce9043ffe0717877e663b87c479bcd07a35ee6f8f990e7b082a`.

SchurSupport

Source SHA-256: `36b368041215ef6857fb3d5809525c1f78436601ff1897e04de9ff1992acfceb`.

Log SHA-256: `fe9fbac8b5b20877f8cabf76b6d71488b1317fdec24ce9c4b395f6ef267530ca`.

SmallCoreNorms

Source SHA-256: `dd397aad6a2b83f9c292d50d201fea3719fc5831b0d982f4ab0036144c28d3e9`.

Log SHA-256: `7ae75ddba072bee8bddb8ed1d858f0f86f54bbeed4c9e5b082e716bb76600d1b`.

SmallCoreRecompression

Source SHA-256: `4a75c726f933e466bbbf14bdfa227def9509b6b891701e61e860f9702f9829f1`.

Log SHA-256: `e9fd9e09de7b3a24cac21183a4c82dd9e0461d70db55f46c3d534e82a77b53b1`.

SmallCoreTransfer

Source SHA-256: `d6040b4f192abcd15ee0f7aebdf81054208e2dae8c0a8a77a40ad473f2efff1b`.

Log SHA-256: `d0fc63e9344310c4d09db11a8697ffdd018300e7463ec80c1678faa7afa08e66`.

SpectralOptimality

Source SHA-256: `b7ae843f3c3dc73462b8761e69c6532225fe0e73040d029218ff65887cf6dba2`.

Log SHA-256: `780db5d54a9e37cce1ed34fb8c16e5d08e1db1403a25f400708e5005ed588383`.

StoredBudgetCompiler

Source SHA-256: `1cf96ed6762939a89b434e7ed6c07f5f696cc794461725bedf590f5eff4d9a89`.

Log SHA-256: `b10e398878b0630b8f7412cbcac6ca0676e06e52a673fc4b1b86e692c2f0155c`.

StructuredTensor

Source SHA-256: `659f010a9ad19e3470f98825b20934147c807aa7582e445d8735e82ff80b71e0`.

Log SHA-256: `50e13efc7deed747d3422504c03a6b830c4373376e5dafd07bffc953bcf97f37`.

ToeplitzEmbedding

Source SHA-256: `b40f87ab523e8fe12299eef50fb822839d13d4635e257b9a644aae397b83e8c5`.

Log SHA-256: `37c02a27a35fa9d66f4a4b8e86dc0e3aeac6a23d07bae989e54a766e17f6f58f`.

TransferSupport

Source SHA-256: `f984a44d1fd1f0b4d6b937ff6e6c4b69f58f7e8a4f29ccf61715252eec5f6cfe`.

Log SHA-256: `e6a96bcb9010fee9eca5e3dde81892729ec623ef4a85de6ba6e8a4146f5df194`.

TreeCompilerUpdates

Source SHA-256: `7d7f8569bef69bc8938e1e883789d72a9e9b8a82def59b79cb637b85274f5001`.

Log SHA-256: `8076a73777b0870b8e5640a9ae56c213685996814570283d00ab6729559c6e54`.

TriangularSolve

15 Exact excerpt register

Each selection is a contiguous block of the unchanged active source, retaining the declaration and proof. Module-level variables and type classes remain in force. The teaching chapters state the relevant context. The printed code applies a font fallback for the one unavailable monospaced glyph. The accompanying `.lean.txt` files preserve the raw UTF-8 bytes without substitutions.

packet_family

Source: `formalization/QuantumPacketCore.lean`.

Lines 22–25; `PacketFamily`.

Source SHA-256: `92e49bb023877ed1f1c2bb34968aa81ee7724fa3a8e1801a831cfc70f1de3595`.

Excerpt SHA-256: `94b93d351a97d973bf990a4a8d1fcd4a8d04e12ae60192ab2eaf58fd05722bde`.

active_evaluation

Source: `formalization/PacketIR.lean`.

Lines 41–49; `exact_active_evaluation`.

Source SHA-256: `67e2e088b17bb95162585592da91c1ed506c5faf7f8d6574cc8b3af8ff0e47c7`.

Excerpt SHA-256: `b8db812ffae61d1a5628229653be4d010699845fb7b5588e5997d9407d3f2d85`.

support_inclusion

Source: `formalization/PacketIR.lean`.

Lines 63–77; `product_support_inclusion`.

Source SHA-256: `67e2e088b17bb95162585592da91c1ed506c5faf7f8d6574cc8b3af8ff0e47c7`.

Excerpt SHA-256: `bd4107318d9f6faefcc4896c60f1995b1a410de7cb48159c39f0ba4c0acd41af`.

adjoint

Source: `formalization/PacketIR.lean`.

Lines 95–97; `exact_adjoint_block`.

Source SHA-256: `67e2e088b17bb95162585592da91c1ed506c5faf7f8d6574cc8b3af8ff0e47c7`.

Excerpt SHA-256: `be55df55acfc0c920766f0848b6325d49a2b00dfd58b78fe36897c8217786a5b`.

batch

Source: `formalization/PacketIR.lean`.

Lines 145–149; `exact_matrix_batch`.

Source SHA-256: `67e2e088b17bb95162585592da91c1ed506c5faf7f8d6574cc8b3af8ff0e47c7`.

Excerpt SHA-256: `55c9fd65ca205c50da1c090e08ba5220eb54d38cddebcad392e005fc5946e62a`.

bdlr_count

Source: formalization/PacketCostModels.lean.

Lines 68–73; bdlr_application_count.

Source SHA-256: 77aa5594485ec62c259e1a1d9493874481e72ceaecd6c2b6f654f42474f999b5.

Excerpt SHA-256: c8640596a7f0bb7aef34ac7b91d5efc23cfec34f8d2166bd00dd5537df37ced7.

roofline

Source: formalization/PacketCostModels.lean.

Lines 118–124; roofline_time_bound.

Source SHA-256: 77aa5594485ec62c259e1a1d9493874481e72ceaecd6c2b6f654f42474f999b5.

Excerpt SHA-256: ea76d943ec216b343c478c484ee260e81281349adf817f422945f448d29454a6.

band_count

Source: formalization/BandedCount.lean.

Lines 73–77; banded_block_count.

Source SHA-256: b13124a6d12cdd0589895d3d12c07da364fd47976e28b2c9541b8828bb291a0d.

Excerpt SHA-256: 215d38825da4198d4d652e7cc7460819c457b1ffee9139397540fe88b1236d7b.

bdlr_product

Source: formalization/LowRankAlgebra.lean.

Lines 57–65; bdlr_product.

Source SHA-256: 61e29902b761b36930587c604cee5740c16fbb076fbde070df7ba80ce0c7984d.

Excerpt SHA-256: 9264a6332bf3610bc332de96452aa893026990771d125af2c65fdabe90fe735e.

chain_factors

Source: formalization/LowRankChains.lean.

Lines 56–66; chainFactors_correct.

Source SHA-256: 970c35e0219c27dedaf97e215a9868ebd4e6015b7f736b9dc0aea385b592212b.

Excerpt SHA-256: 65b33e5cf658798a4651f3e8a83c1a60af9af6c303b799e3275b202c6a0fe756.

woodbury

Source: formalization/LowRankAlgebra.lean.

Lines 89–94; bdlr_woodbury.

Source SHA-256: 61e29902b761b36930587c604cee5740c16fbb076fbde070df7ba80ce0c7984d.

Excerpt SHA-256: b6e65a998993dadf81b79f505d629ab076210bc137f29932b27b6a62421f7ee4.

kronecker

Source: `formalization/StructuredTensor.lean`.

Lines 19–22; `kronecker_modewise`.

Source SHA-256: `659f010a9ad19e3470f98825b20934147c807aa7582e445d8735e82ff80b71e0`.

Excerpt SHA-256: `42d94ff97bf545b6b0abb297f76db13f7caec1ef8bcd501a96e2d8a19f6fc967`.

tensor_apply

Source: `formalization/StructuredTensor.lean`.

Lines 73–96; `tensor_apply_correct`.

Source SHA-256: `659f010a9ad19e3470f98825b20934147c807aa7582e445d8735e82ff80b71e0`.

Excerpt SHA-256: `87c9fee83936f6c3056c31d7bbdeede737219cb900935b1921dbdf777527e4a8`.

walsh_apply

Source: `formalization/WalshTransform.lean`.

Lines 60–84; `walsh_apply_correct`.

Source SHA-256: `fd74891c276e65c3bdade75cd37768243b5b1d3f61a6a4c3a8b03ed6236551e7`.

Excerpt SHA-256: `13305b25799d85ceee84f430c17c47158717349a71a5eef9aa90e34d5ee93431`.

toeplitz_apply

Source: `formalization/ToeplitzEmbedding.lean`.

Lines 54–58; `toeplitz_embedding_action`.

Source SHA-256: `b40f87ab523e8fe12299eef50fb822839d13d4635e257b9a644aae397b83e8c5`.

Excerpt SHA-256: `45b0c8e014eba5778271dc6127c731cf5b76fdccbe470570482a9f20ef341441`.

fourier

Source: `formalization/CirculantFourier.lean`.

Lines 150–153; `circulant_spectral_representation`.

Source SHA-256: `8db2cde5eec1565ff44c496d4e73a49a2ef3d28a5afc2e1ca0874ad25af3b02c`.

Excerpt SHA-256: `4d96344b6bfd07b20e24ed96afb28abf92fffc5f654d261025c97ec8c3018275`.

power_paths

Source: `formalization/TransferSupport.lean`.

Lines 37–52; `power_path_expansion`.

Source SHA-256: `f984a44d1fd1f0b4d6b937ff6e6c4b69f58f7e8a4f29ccf61715252eec5f6cfe`.

Excerpt SHA-256: `872dd779a4e39a29bd50c57cb1326c95870477e966e459372538ecbaf2d0938c`.

nilpotent

Source: `formalization/TransferSupport.lean`.

Lines 86–90; `nilpotent_of_path_bound`.

Source SHA-256: `f984a44d1fd1f0b4d6b937ff6e6c4b69f58f7e8a4f29ccf61715252eec5f6cfe`.

Excerpt SHA-256: `6874427d6bcf4444acc78d23c0f6b7ae953896e30607aa5b37b6641e90e08e77`.

inverse_support

Source: `formalization/InverseSupport.lean`.

Lines 94–100; `inverse_block_zero_of_no_path`.

Source SHA-256: `e8214096eab227677cdd7977840b02844bd0ec706df57dca405bcfae2f6e2063`.

Excerpt SHA-256: `49f48da9ac56e4b9f42a3624291f367fecb5ed88da26b521a5882edbc3895fb6`.

schur_support

Source: `formalization/SchurSupport.lean`.

Lines 57–73; `schur_support_inclusion`.

Source SHA-256: `36b368041215ef6857fb3d5809525c1f78436601ff1897e04de9ff1992acfceb`.

Excerpt SHA-256: `eb2070911268dd8d6f00ab3c496f5fa0f3983d3525362018d17dab441083e6df`.

solve_correct

Source: `formalization/TriangularSolve.lean`.

Lines 38–43; `solve_correct`.

Source SHA-256: `0f5939c4982d65b5b74c10455503686adc32923d5211561a17a291eae1e33a0e`.

Excerpt SHA-256: `16f88870ae9e77b3046abcd6f1986ba4992f59ff2254bffb42825467c9fcd4e4`.

schedule_depth

Source: `formalization/GraphSchedules.lean`.

Lines 150–154; `optimal_depth`.

Source SHA-256: `1b845790a59fdf87562a2afa5dd56cda8ee0d65ae4cfbb8a1259364d4a01d4cf`.

Excerpt SHA-256: `48fea4f397c61d44b2d4ce4ecb71402f7d3947c83b2338ed5624499b09eded57`.

packet_error

Source: `formalization/PacketPerturbation.lean`.

Lines 35–39; `packet_perturbation_bound`.

Source SHA-256: `5fbb1026da706b46afaf2793e138fe503e00ed37a470c193388ad6e79a37bd12`.

Excerpt SHA-256: `f9be290eddd13b0e1e78d379a60f35f7d7aa6f6b25d5b8e19ea8c158b24c27c5`.

chain_error

Source: formalization/ProductChainError.lean.

Lines 129–145; product_chain_error_bound.

Source SHA-256: 5bad0d7ed04be1a94bb19e54c9371e3aea0dc69ff90e4afc33c6cdc0e4686a2c.

Excerpt SHA-256: d5ce5f5abbf53f974cdaae1745fbbc526424537fc51b88c495f58fa6fec5a798.

recompression_error

Source: formalization/ProductChainError.lean.

Lines 100–111; repeated_recompression_bound.

Source SHA-256: 5bad0d7ed04be1a94bb19e54c9371e3aea0dc69ff90e4afc33c6cdc0e4686a2c.

Excerpt SHA-256: 1d5d7a1765f121c95ac8c2571665e1ccb9429065c71075df02f552ccb9816965.

continuous_precision

Source: formalization/ContinuousPrecision.lean.

Lines 80–97; precision_allocation_optimal.

Source SHA-256: 5a737f5aa8ed23525c225b48fc6cef46a623fe85b7ddf8ee4bf9115976667230.

Excerpt SHA-256: 56e7ccbc5bc16b61893e32284ae60ea67fd8f4241dbe4de8ce75f39d2e94b544.

clipped_precision

Source: formalization/ClippedPrecision.lean.

Lines 211–234; clipped_unique_minimizer.

Source SHA-256: aa86e409a530f0916da245bfe3cf81b7fba26e97c12f8fd7a7690351a08dffc9.

Excerpt SHA-256: c92e6134eb561e19b80ed28b63c178e1893a6a2f905b5628a7c1c633d45be8f3.

discrete_recurrence

Source: formalization/DiscretePrecision.lean.

Lines 38–42; discreteTable.

Source SHA-256: 0b12bf6ecf1c9bc36bc735ee737c06db5a37651d12dca3ffcad230493bb7a654.

Excerpt SHA-256: 6228e94e6ca74427d363ac3f162d2ad71b2aa3bb7dfdf857d23338416d090413.

discrete_endpoint

Source: formalization/DiscretePrecision.lean.

Lines 110–121; discreteBudget_some_iff.

Source SHA-256: 0b12bf6ecf1c9bc36bc735ee737c06db5a37651d12dca3ffcad230493bb7a654.

Excerpt SHA-256: 12a6e2e0aac33aa266dcebe1a641bf422a65954cd6cf9caaca44d92c07576adf.

candidate_tree

Source: `formalization/ExactTreeCompiler.lean`.

Lines 21–23; `CandidateTree`.

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Excerpt SHA-256: `6042f121d7e093d71bf88604481295fc3a44039b07719a96e678fb01847dfe08`.

cut

Source: `formalization/ExactTreeCompiler.lean`.

Lines 33–36; `Cut`.

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Excerpt SHA-256: `c39d677c2ec68517663c4047337d620a3fd2e9f7a76c3b6014e90d714b448125`.

tree_recurrence

Source: `formalization/ExactTreeCompiler.lean`.

Lines 45–47; `optimalValue`.

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Excerpt SHA-256: `bdd0f6e304db9f408f6700b56f5c28afe3b8e723b8e07f89b176e676535151a7`.

compile_attains

Source: `formalization/ExactTreeCompiler.lean`.

Lines 69–76; `compile_attains`.

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Excerpt SHA-256: `758756e03cc7b28620f65cfb1246578a05b283dbe8c19abe503e4c7bf140e9b8`.

tree_exact

Source: `formalization/ExactTreeCompiler.lean`.

Lines 79–83; `exact_tree_compiler`.

Source SHA-256: `680d0a063af7cba93ce7b29ac9bfa1ba65dea1b9cf42309c55550b394c3a3f5f`.

Excerpt SHA-256: `7638f7a2a4eb00eb9e8d12462200118215899cf68b16346b667fa72c94ebfc39`.

budget_recurrence

Source: `formalization/QuantizedTreeCompiler.lean`.

Lines 27–33; `budgetTable`.

Source SHA-256: `564ddda43334a4cd506e7bf990704b4fc604eeccb891084a223cae53b96268e4`.

Excerpt SHA-256: `c6e79eb95cd127303ec581bedf142a2d6fd275df9d5463d932e9e8e0813dc748`.

budget_endpoint

Source: formalization/QuantizedTreeCompiler.lean.

Lines 107–119; compileBudget_some_iff.

Source SHA-256: 564ddda43334a4cd506e7bf990704b4fc604eeccb891084a223cae53b96268e4.

Excerpt SHA-256: 36aa0f165da0f22adee78e1a2a896121e4aa17d33cbde46f6f4e79da2a53fed1.

budget_none

Source: formalization/QuantizedTreeCompiler.lean.

Lines 122–126; compileBudget_none_iff.

Source SHA-256: 564ddda43334a4cd506e7bf990704b4fc604eeccb891084a223cae53b96268e4.

Excerpt SHA-256: b3fc8f236ff026cdd83f7caf79190b0ad2c15d99c14ff943bf9aa6bf554fe6ba.

stored_tree

Source: formalization/StoredBudgetCompiler.lean.

Lines 109–118; storedTree.

Source SHA-256: 1cf96ed6762939a89b434e7ed6c07f5f696cc794461725bedf590f5eff4d9a89.

Excerpt SHA-256: c584d5dc39f3557a80946b4b00d454f949b0a032083c1bead34a2b3c7eaae365.

stored_correct

Source: formalization/StoredBudgetCompiler.lean.

Lines 120–133; storedTree_correct.

Source SHA-256: 1cf96ed6762939a89b434e7ed6c07f5f696cc794461725bedf590f5eff4d9a89.

Excerpt SHA-256: d0fb967877905890a693b75b63146bd23243caa565a1ab2640781370abfe278c.

pseudo_polynomial

Source: formalization/StoredBudgetCompiler.lean.

Lines 256–260; tabulation_pseudopolynomial.

Source SHA-256: 1cf96ed6762939a89b434e7ed6c07f5f696cc794461725bedf590f5eff4d9a89.

Excerpt SHA-256: 694efc5328fd56ccb8f37633c3f507172d6de874811ff601b23cee7eb9dbefe7.

compiler_stability

Source: formalization/TreeCompilerUpdates.lean.

Lines 45–59; compiler_stability.

Source SHA-256: 7d7f8569bef69bc8938e1e883789d72a9e9b8a82def59b79cb637b85274f5001.

Excerpt SHA-256: 20d720b1679d7fa385f677c73a0070be0fc0e6ec4d78100b1c96d206c426bf3d.

local_update

Source: `formalization/TreeCompilerUpdates.lean`.

Lines 142–150; `local_recompilation`.

Source SHA-256: `7d7f8569bef69bc8938e1e883789d72a9e9b8a82def59b79cb637b85274f5001`.

Excerpt SHA-256: `2f57ee7d82d42de016997621fdd1564a1ad6a29730d07461ed424e24fc9027e3`.

contract

Source: `formalization/SchedulingContracts.lean`.

Lines 54–60; `contract_conversion`.

Source SHA-256: `ea646fab14df1c6f6b4855745035b64835920a75fe2e66d168cb515f74854160`.

Excerpt SHA-256: `e9796ff3b4bd8914b18d8c74a03dc9d332f6c25c6e2e05d63e395be81feb5dda`.

grouping

Source: `formalization/SchedulingContracts.lean`.

Lines 89–93; `grouped_execution_output`.

Source SHA-256: `ea646fab14df1c6f6b4855745035b64835920a75fe2e66d168cb515f74854160`.

Excerpt SHA-256: `f5631a2725315cbf2b0f719019445a0049b43d600322868ac7f91353192d2243`.

amortization

Source: `formalization/SchedulingContracts.lean`.

Lines 117–123; `integer_amortization_threshold`.

Source SHA-256: `ea646fab14df1c6f6b4855745035b64835920a75fe2e66d168cb515f74854160`.

Excerpt SHA-256: `3228ffc9dda59871a65a0470e3d4c06a738ad5934a73f1e3bff9fad957aa1da5`.

krylov

Source: `formalization/KrylovPackets.lean`.

Lines 89–99; `invariant_packet_krylov`.

Source SHA-256: `12fd73f08368ec528953e66e165ef4281f1a978052c21e0c61dbde6d12c508b5`.

Excerpt SHA-256: `5493ac684d15f1329599b43e94cf1a98c0e78bf41c1af8a47e980968c09bb759`.

approx_krylov

Source: `formalization/KrylovPackets.lean`.

Lines 183–195; `approximate_packet_krylov_distance`.

Source SHA-256: `12fd73f08368ec528953e66e165ef4281f1a978052c21e0c61dbde6d12c508b5`.

Excerpt SHA-256: `f5b2dc70dee98e47baf485eef18ba8a584d181afc886b22d5b47022d1c6f4cb8`.

jacobi

Source: formalization/BlockJacobi.lean.

Lines 87–95; block_jacobi_condition.

Source SHA-256: 0b5b13491150bdab13be5cc72e6fb0040f39f06cc06398095c360bcb632ced71.

Excerpt SHA-256: f7dedba3028e599a34f27ec89a2847a54f3e3fc76babd443c4464d1dc5c9add1.

cg_bound

Source: formalization/BlockJacobi.lean.

Lines 134–141; certified_energy_error.

Source SHA-256: 0b5b13491150bdab13be5cc72e6fb0040f39f06cc06398095c360bcb632ced71.

Excerpt SHA-256: 9206b8018309d6b6fd6c47ce89cbaa62b10910c8912755a3eeb92d1bad64d930.

discovery

Source: formalization/DiscoveryCertificate.lean.

Lines 126–137; discovery_frobenius_certificate.

Source SHA-256: ae8a69d0022cb8258a75d91b5e7e98c9a4f0f60d4d67f2d976b093d8a5987e8f.

Excerpt SHA-256: ceb4c4d0d9e6d4cae8f5a47105676a6ea536874e27575476646e7ce647dcfc90.

compact_factor

Source: formalization/CompactRangeFactors.lean.

Lines 51–57; compact_factorization.

Source SHA-256: 3ae44a0f80503e8455cc2cd4ad7de140e631c59e32a109cd5fae2366b34d2e1c.

Excerpt SHA-256: 3c31204f3c15ded1aefab7a77dace1ccfa12800618b4d57bb0e25094518e2fe1.

best_rank

Source: formalization/SmallCoreTransfer.lean.

Lines 16–17; BestRank.

Source SHA-256: d6040b4f192abcd15ee0f7aebdf81054208e2dae8c0a8a77a40ad473f2efff1b.

Excerpt SHA-256: 152df4ae4bf5c843eba1d574336f437b77a7dc6482f91f2d68b5487f553dffe4.

ambient_transfer

Source: formalization/SmallCoreTransfer.lean.

Lines 25–35; spectral_best_lift.

Source SHA-256: d6040b4f192abcd15ee0f7aebdf81054208e2dae8c0a8a77a40ad473f2efff1b.

Excerpt SHA-256: 10fd13227a608516e1d94fe05a23e19a264c25dd3587d80d9b8c8f20bd5d88fc.

ordered_svd

Source: `formalization/SmallCoreRecompression.lean`.

Lines 29–37; `OrderedThinSVD`.

Source SHA-256: `4a75c726f933e466bbbf14bdfa227def9509b6b891701e61e860f9702f9829f1`.

Excerpt SHA-256: `7ffa30b8be7d8fcc025e5d89b1d85745eee3f58847d9ff10150482633ae170f7`.

kernel_witness

Source: `formalization/SpectralOptimality.lean`.

Lines 55–82; `prefix_kernel_witness`.

Source SHA-256: `b7ae843f3c3dc73462b8761e69c6532225fe0e73040d029218ff65887cf6dba2`.

Excerpt SHA-256: `b244da5b0dbb939ab440aea1192e76e0d87213a86ba223d011ca0057183c1143`.

spectral_best

Source: `formalization/SpectralOptimality.lean`.

Lines 120–125; `diagonal_spectral_best`.

Source SHA-256: `b7ae843f3c3dc73462b8761e69c6532225fe0e73040d029218ff65887cf6dba2`.

Excerpt SHA-256: `845fbd554f691fa869be101e0439cdf3a57b77d4cecae9db0f51d0ea0b26ab3a`.

weighted_tail

Source: `formalization/FrobeniusOptimality.lean`.

Lines 35–60; `weighted_tail_lower`.

Source SHA-256: `3d28205f0088a2ebc9ca78141cefb73d4d14bdb4f1feb9cb860968cfa9b29f87`.

Excerpt SHA-256: `06a99100943ba24f123308c7cbc9be31d6358ac68c4bf96738b538ef6270faac`.

frobenius_lower

Source: `formalization/FrobeniusOptimality.lean`.

Lines 76–104; `diagonal_frobenius_lower`.

Source SHA-256: `3d28205f0088a2ebc9ca78141cefb73d4d14bdb4f1feb9cb860968cfa9b29f87`.

Excerpt SHA-256: `ec9c2efbd1402edb36371b9234ce5986fd1c8d8912d9b117a4b720ca95e0806b`.

recompression_optimal

Source: `formalization/SmallCoreRecompression.lean`.

Lines 91–120; `recompression_optimality`.

Source SHA-256: `4a75c726f933e466bbbf14bdfa227def9509b6b891701e61e860f9702f9829f1`.

Excerpt SHA-256: `89293b779e3b624a8aa564e3650b986204d99e06c2633e0be689bdfbd8b0d7e3`.

small_core_endpoint

Source: `formalization/SmallCoreRecompression.lean`.

Lines 124–141; `small_core_optimal_recompression`.

Source SHA-256: `4a75c726f933e466bbbf14bdfa227def9509b6b891701e61e860f9702f9829f1`.

Excerpt SHA-256: `30ffd8ab1cfe1b7c7fd7c3867247600a5946e08d57e3b4c17e77f2d6df1c883c`.

full_retention

Source: `formalization/RecompressionEndpoints.lean`.

Lines 70–74; `recompression_full`.

Source SHA-256: `d8e8dc3290e938198c524607904db67869659461b852964e46297de4690a3789`.

Excerpt SHA-256: `e8689561b90aaef3b937ea6d2bd9c95e0bea3d1c1c44a0bfd5579e31fb3c3219`.

Euler Congruences

The formal companion to Chapter IX explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2221

Return to Chapter IX

Reading scope

This guide records the formal coverage of the Euler chapter and teaches the retained arithmetic, finite obstruction, polynomial tower, quantitative deformation, and analytic regularity proofs. The opening chapter maps the strengthened congruences to their existing Lean endpoints and identifies the new proofs that remain to be formalized. The complete original teaching chapters, exact source excerpts, and catalog of named sources follow.

The formal evidence consists of retained compiler records and matching source identities. The factorial and mixed-difference theorems have written proofs in the main text but no compiled Lean endpoint. Static source checks and finite computations do not replace a kernel check.

1 The strengthened Euler chapter and its exact formal coverage

The chapter combines Nishibuchi's conjecture, the four-packet theory at every row, and stronger written congruences. The capstone, step-sensitive four-packet bound, exact first-difference periods and isometry have Lean sources and compiler records. General prime-power closure, mixed factorial estimates, exact higher-order laws and Taylor radii have mathematical proofs but no compiled Lean endpoint.

The distinction is attached to claims, rather than to whole documents. A new proof may include a previously formalized special case. Conversely, an inherited endpoint about a polynomial analytic graph does not certify the new Euler interpolation radius merely because both use p-adic analysis. The tables and contracts below identify the actual overlap and the remaining work.

1.1 One normalization for the chapter and the guide

For a positive modulus N and an address $0 \le j < N$, define

$$G_{N,j}(t) = j! \sum_{n \ge 0} \frac{t^{Nn}}{(Nn+j)!}, \qquad G_{N,j}(t)^{-1} = \sum_{n \ge 0} C_{N,j}(n) \frac{t^{Nn}}{(Nn)!}.$$

The constant coefficient of both normalized series is one. The original generalized Euler number at degree Nn is

$$E_{Nn}^{(N,j)} = j!\, C_{N,j}(n).$$

Thus `Congruential.coeff N j n` corresponds to $C_{N,j}(n)$ and `Congruential.unnormalized N j n` corresponds to $E_{Nn}^{(N,j)}$. The source `NishibuchiCompletion.generalizedEuler N hN j d` instead uses the literal Euler degree d. Its theorem `generalizedEuler_at_packet` is the bridge between these indices.

Write $\Delta_H f(n) = f(n+H) - f(n)$. The source `Mahler.difference H k f n` is the homogeneous difference $\Delta_H^k f(n)$. A mixed difference applies several possibly unequal steps. Existing homogeneous endpoints cannot be substituted for a mixed endpoint without a proof relating those operators.

The notation $v_p(0) = +\infty$ makes lower bounds zero safe on paper. In the sources, `GlobalEulerBoundary.precision p r x` supplies the corresponding zero-safe rational predicate. Actual integral lifts express the same congruence as divisibility by $(p : \mathbb{Z}_p)^r$. Lean's rational valuation of zero is not itself an infinite-valued function. Exact valuation endpoints therefore include a separate nonzero conclusion or require a nonzero hypothesis. That distinction is essential when translating a lower bound into a typed target.

1.2 Current claim map

Topic in the main text	Retained Lean coverage	Present status
Nishibuchi Conjecture 3.9, all permitted primes and addresses	`NishibuchiCompletion.conjecture_3_9`, with the original integrality and degree-shift bridges	Compiled in the recorded source.

Topic in the main text	Retained Lean coverage	Present status
Four-packet all-row Kummer divisibility	`DyadicFourAllRows.normalized_precision`	Compiled, including row zero and every difference order.
Four-packet common-step precision, including the address factorial	`DyadicFourFinalStrengthening.normalized_step_precision` and `unnormalized_step_precision`	Compiled with exponent $k(1+v_2H)$, with $v_2(j!)$ for the original numbers.
Four-packet sharp first differences, least periods, and isometry	`DyadicFourLeastPeriods` and `DyadicFourIsometry`	Compiled under the stated address and positive-step hypotheses.
All prime-power moduli and all mixed steps	No matched general endpoint	Proved in the new continuation. Not formalized in Lean here.
Four-packet factorial gain and address surpluses	No matched higher-order endpoint	Proved in the new continuation. Not formalized in Lean here.
Exact higher differences at addresses zero and one and the mixed leading-digit criterion	No matched endpoint	Proved in the new continuation. Not formalized in Lean here.
Address-two stronger growth and exact power-of-two orders	No matched endpoint	Proved in the new continuation. Not formalized in Lean here.
Small-rank metric results beyond $N = 4$ and the all-rank lower-half metric theorem	No matched endpoint	Proved in the new continuation. Not formalized in Lean here.
Exact Euler Taylor radii at addresses zero, one, and two	No matched endpoint	Proved in the new continuation. Not implied by the retained polynomial regularity modules.
General exact higher-order law at addresses two and three and the unrestricted upper-half metric law	No completed endpoint	The stated unrestricted laws remain open or are refuted in the main text.

The table identifies mathematical topics and their formal coverage. The crosswalk links the corresponding monograph statements and source interfaces. Separate routes identify the strengthening statements and eight exact worked examples.

1.3 What the Nishibuchi capstone actually proves

Let p be prime, let $m > 0$ satisfy $m \mid p-1$ or $m = 2$, put $q = \operatorname{lcm}(2, p-1)$, and let $0 \le j < mp$. The capstone proves p-integrality of every original packet coefficient and, for every $r \ge 1$, the degree-indexed congruence

$$E^{(mp,j)}_{mpn+qp^r} \equiv E^{(mp,j)}_{mpn} \pmod{p^r} \qquad (n \ge r).$$

Its existential formulation supplies the positive threshold $n_0 = r$, so it also supplies $n_0 \le r$. It does not assert the smallest possible threshold. The source has no premise asking the caller to assume source flatness, a weighted certificate, or the desired congruence.

The divisor branch supplies the stronger sufficient row condition $r \le mn$. The arithmetic parameter reduction isolates the exceptional case $p = 2$, $m = 2$. At $p = 2$, $q = 2$. Replacing q by $p - 1 = 1$ would change the conjecture. The degree-shift theorem proves that the packet step

$$p^{r-1}\frac{q}{m}$$

produces exactly the Euler-degree increment qp^r.

The completion source contains 79 proof modules and 1,000 named axiom reports. The complete declaration index has 1,876 rows under 199 canonical files, including overlapping arithmetic and lifting developments. These are source counts and must not be added or treated as counts of mathematical discoveries.

The completion records include successful compilation, axiom checks and rejection tests. The capstone uses only propext, Classical.choice and Quot.sound. Static source and log checks do not rerun the Lean kernel.

1.4 The four-packet theorem already passed the row threshold

The earlier completion extracted a bound with exponent $\min(k, n)$. The retained strengthening subsequently replaced that cap by k using the unshifted operator

$$B = \frac{D^4 - 1}{2}.$$

Its charged reciprocal theorem proves that the source's flatness survives inversion, with the charge shift made explicit. The module `DyadicFourAllRows` then extracts the difference directly at degree $4n$. It has no $k \le n$ assumption and no artificial extraction at a negative degree.

The endpoint `DyadicFourFinalStrengthening.normalized_step_precision` states

$$v_2(\Delta_H^k C_{4,j}(n)) \ge k(1 + v_2 H) \qquad (j < 4,\ n, k \in \mathbb{N},\ H \ge 1),$$

with its zero-safe precision interpretation. The original-number endpoint adds $v_2(j!)$. For positive H, this is exactly the four-packet homogeneous specialization of the general mixed-step estimate in the new chapter. The source also includes zero step using its finite natural valuation `padicValNat 2 0 = 0`: every positive-order zero-step difference vanishes. At order zero, the statement is just the appropriate integrality assertion. Thus the source never forms the undefined paper expression $0 \cdot v_2(0)$.

For these four packets, the conjectured dyadic period already starts at row zero in the retained Lean source. The literal degree endpoint is `all_rows_period_degree`. The exact-period development goes further and proves

$$v_2(E^{(4,j)}_{4(n+H)} - E^{(4,j)}_{4n}) = v_2 H + B_j, \qquad (B_0, B_1, B_2, B_3) = (1, 1, 5, 3),$$

for positive H. It follows that the least positive row period modulo 2^r is

$$2^{\max(0, r - B_j)}.$$

The normalized bonuses are $(1, 1, 4, 2)$. The isometry endpoint uses a translated and rescaled packet, not the raw original sequence. Undoing its normalization gives the original image ball $j! + 2^{B_j}\mathbb{Z}_2$.

The written results extend the modulus range and operator patterns, strengthen higher differences by factorial and address gains, and give exact higher-order and analytic consequences. The already formalized four-packet statements retain their own coverage.

1.5 The strongest homogeneous congruence to formalize next

For $j \in \{0, 1, 2, 3\}$, $n \geq 0$, and $k, H \geq 1$, the final congruence in the main text is

$$\sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} E^{(4,j)}_{4(n+iH)} \equiv 0 \pmod{2^{\,v_2(k!)+kv_2H+A_j(k)}},$$

where

$$A_0(k) = A_1(k) = k, \qquad A_2(k) = \max\{k+4,\ 2k + \lfloor k/2 \rfloor + 1\}, \qquad A_3(k) = k + 2.$$

This displayed exponent includes the address factorial. Using it directly for the normalized $C_{4,j}$ would introduce an extra digit at addresses two and three. The lower bound holds at every base row, including zero. Equality holds for every positive order and positive homogeneous step at addresses zero and one. At address two, the exact laws proved in the chapter cover orders that are powers of two, including one and two, with the explicit small-order exceptions stated there. This is not a theorem asserting equality at every address-two order.

A direct formal target should use `Mahler.difference H k (Congruential.unnormalized 4 j) n`, the natural exponent `padicValNat 2 k.factorial + k * padicValNat 2 H + A j k`, and `GlobalEulerBoundary.precision`. Its hypotheses are $j < 4$, $0 < k$, and $0 < H$. That target reuses the original definitions and preserves the chapter's quantifiers. It should not replace the rational sequence by a separately named experimental recurrence without a coefficient-identity bridge.

The proof of the new factorial term is a divided-power argument. At addresses zero and one, its external arithmetic input is the exact classical Frobenius congruence for ordinary Euler numbers, in the form checked in the main text's source record. The Bernoulli central-factorial argument at address two proves its divided-power integrality internally. A Lean completion will need the exact ordinary-Euler input formalized or imported as a proved theorem with a matched statement. A custom axiom standing in for that input would change the formal coverage and must not be hidden under an unchanged completion label.

1.6 General prime-power closure: a reusable target

For $N = p^a$, $a \geq 1$, and $j < N$, put $C(n) = C_{N,j}(n)$. The general theorem is most naturally phrased for the sign-adjusted sequence

$$A(n) = (-1)^n C(n).$$

For positive steps $H_1, \ldots, H_k$, it proves

$$v_p(\Delta_{H_1} \cdots \Delta_{H_k} A(n)) \geq k + \sum_{i=1}^{k} v_p H_i.$$

For the raw sequence, define $T_H f(n) = f(n+H) - (-1)^H f(n)$. The equivalent original-number statement is

$$v_p(T_{H_1} \cdots T_{H_k} E^{(N,j)}(n)) \geq v_p(j!) + k + \sum_{i=1}^{k} v_p H_i, \qquad E^{(N,j)}(n) = E_{Nn}^{(N,j)}.$$

At $p = 2$, the sign adjustment and an integral charge shift give the ordinary mixed-difference theorem for $C_{N,j}$ itself. The degenerate dyadic modulus $N = 1$ has $C(n) = (-1)^n$ and is handled directly. At an odd prime, ordinary unsigned differences of the raw sequence need not satisfy the same estimate: $C_{N,0}(0) = 1$ and $C_{N,0}(1) = -1$, so the first unit-step difference is -2.

A useful formalization order is:

1. Prove integral Hurwitz coordinates are closed under products, derivatives, and inversion of a unit constant coefficient. Retained `HurwitzConvolution`, `HurwitzInverse`, and `DworkAlgebra` provide reusable constructions. Their exact coefficient domains must be matched before reuse.
2. Prove interior binomial divisibility $p \mid \binom{p^a}{r}$ for $0 < r < p^a$, and record the integral coefficients in the charged operator product formula.
3. Generalize the charged reciprocal closure from D^4 to D^N, keeping the source and inverse charges visible. The critical induction isolates the unique term containing the highest inverse order. No hypothesis $k \leq n$ is permitted.
4. Prove flatness of the actual source $g_{N,j}(n) = \prod_{i=1}^{j}(1 + (N/i)n)^{-1}$. Every slope N/i belongs to $p\mathbb{Z}_p$ because $1 \leq i < N = p^a$. Prove the reciprocal affine-factor finite-difference identity before taking products.
5. Decode the reciprocal coefficient at a nonnegative packed degree, then transfer to rational precision and to the original factorial normalization.
6. Prove mixed-step amplification, with independent step valuations added, and specialize to homogeneous steps only after the general target is established.

This sequence isolates the new universal arithmetic infrastructure. The retained $N = 4$ charged closure proves a special case, but its source signature does not quantify over every prime-power modulus. A future generalized file should remain a new endpoint until its statement and axiom closure have been checked.

1.7 Divided powers, factorial gains, and equality

For four-packet normalized coefficients, the factorial theorem states

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} C_{4,j}(n)) \geq k + v_2(k!) + \sum_{i=1}^{k} v_2 H_i + \beta_j, \qquad (\beta_0, \beta_1, \beta_2, \beta_3) = (0, 0, 3, 1),$$

for $k \geq 1$. Multiplication by $j!$ changes the address surpluses to $(0, 0, 4, 2)$ for the original numbers. At address two, the additional bound

$$v_2(\Delta_{H_1} \cdots \Delta_{H_k} C_{4,2}(n)) \geq 2k + v_2(k!) + \lfloor k/2 \rfloor + \sum_{i=1}^{k} v_2 H_i$$

combines with the first bound by taking their maximum.

The formal proof must distinguish a valuation lower bound from the evaluation of its leading unit. Divided-power integrality proves the lower bound. Exactness additionally requires a parity calculation

after dividing by the claimed power of two. That calculation is where unequal steps can cancel. At addresses zero and one, let

$$V = k + v_2(k!) + \sum_{i=1}^{k} v_2 H_i.$$

The new chapter proves exact valuation V if k is odd or the number of even steps is even. If k is even and the number of even steps is odd, the leading digit cancels and the valuation is at least $V + 1$. That last case does not carry a uniform formula for the deeper valuation.

For a homogeneous step, the number of even steps is either zero or k. The criterion therefore always gives exactness at addresses zero and one. This explains why a true homogeneous equality can fail when a single step is changed. For example,

$$\Delta_1\Delta_2 C_{4,0}(0) = -33728,$$

has valuation 6, whereas the mixed baseline is 4. The same pattern at address one has value $-150208/585$ and valuation 6. These examples should become exact rational negative controls for a proposed unconditional mixed equality. They should not be used as a substitute for the general parity proof.

At address two, the exact mixed-step exponent is $4 + \sum v_2 H_i$ at order one, $6 + \sum v_2 H_i$ at order two, and

$$2k + v_2(k!) + k/2 + \sum v_2 H_i$$

at powers of two $k \geq 4$. The split between these three targets is deliberate. A theorem with the final expression at order one would not state the correct sharp exponent. General exact higher-order laws at addresses two and three remain outside the completed formal and mathematical register.

The proof route uses two Gaussian-scaled Euler or Bernoulli umbrae over $\mathbb{Q}_2(i)$ and a descent back to rational coefficients. Lean obligations include the base change, the Gaussian scalar identities, the divided-power product and composition laws, and the rational descent. Treating the symbolic complex expression as though it were already a rational integral element would omit a substantive bridge. The unbounded exactness subsequence at address two also needs its parity proof. A computed prefix through order 512 does not supply that bridge.

1.8 Analytic consequences require their own bridge

The factorial Newton estimates are converted into Taylor coefficients using the Stirling change of basis, with convergence and uniformity justified. They imply an exact Taylor radius 2 at every integral center for addresses zero and one. The address-two bounds, together with the sharp power-of-two coefficient subsequence, give exact radius $2^{5/2}$. The address-three result gives a radius of at least 2. Its exact radius is not claimed.

A formal endpoint should specify the coefficient field, the absolute value normalization $|2|_2 = 1/2$, the chosen center, and the meaning of radius. If the radius is viewed through an extension containing points outside $\mathbb{Q}_2$, the relevant convergence statement needs that ambient extension specified. The root-test coefficient rate itself can be stated independently. A generic Kummer bound alone does not establish these analytic radii. The retained `PolynomialAnalyticEvaluation`, `PolynomialAnalyticBlocks`, and local implicit-graph modules concern different objects and do not close this argument.

A future Lean proof should separately verify Newton interpolation equals the original rational packet on natural inputs, the Newton-to-Taylor coefficient identity, the coefficient lower bounds, the

exact subsequence supplying the upper radius bound, and convergence on the asserted disk. No result about finitely many numerical coefficients can be promoted to the exact radius without that unbounded subsequence argument.

1.10 A clean completion criterion for the new formal layer

A theorem is formally checked only after its exact statement and source are fixed, its dependencies are identified, positive examples compile, relevant false strengthenings fail for the expected reason, and its axiom closure is recorded. The controls cover odd-prime raw differences, unconditional mixed equality, zero steps, the address factorial, and the higher-rank upper-half metric boundary.

Proof sources must not replace an argument with a placeholder, an axiom for the desired conclusion, or precomputed coefficients. Imported arithmetic needs a matched theorem and a justified bridge to the chapter's convention. Common-step and mixed-step bounds, and finite tests and universal statements, remain distinct. The inherited capstone does not automatically certify the stronger written results.

2 How to read the formal development

The development uses two complementary mathematical languages. Arithmetic packets begin with sequences and formal power series, then transport divisibility through exact coordinate changes. Polynomial lifting begins with a finite residual equation, passes to compatible digit towers, and finally enters complete p-adic and analytic spaces. The transition between these languages is explicit: a rational coefficient has to be shown integral before it becomes a p-adic integer. An integer congruence tower has to be shown Cauchy before it becomes a root. A formal polynomial derivative has to be identified with a strict Fréchet derivative before it supports an analytic inverse theorem.

This guide follows those transitions. Read the *contract* before the excerpt: it states the coefficient domain, ambient dimensions, and hypotheses that may be declared earlier in the source. The excerpt is an exact contiguous selection from a preserved Lean file. Its line range and source hash are supplied in the accompanying excerpt inventory. The proof reading then explains what the displayed tactics accomplish mathematically. A theorem imported from another module is a dependency, not a hidden new assumption. A hypothesis supplied to a theorem remains a mathematical assumption even when its name does not appear in the theorem's axiom closure.

2.1 Statements, constructions, audits, and identifiers

The retained preceding consolidated manuscript links 62 statements to 202 named source declarations. That index resolves names and source identities. It does not prove that every informal sentence equals a Lean statement. Several declarations may establish one displayed theorem, while one source theorem may support several expository consequences. The restored Bernoulli specialization has a classical formal-series derivation in the manuscript, but no dedicated Lean bridge was matched. The manuscript's open problems retain their open status.

The full declaration index has 1,876 rows. A row records a declaration kind, qualified name, and entire source-file hash. Byte-identical copies are collapsed. Distinct source files remain separate. There are 1,863 distinct kind/name pairs, including 1,542 distinct names whose command kind is theorem. These are identifier counts, not counts of independent mathematical results. The rows also contain constructions, statement contracts, examples and control support. Thirteen names have more than one source file.

The polynomial regularity development contains 322 theorem commands, 76 definitions and seven abbreviations in 55 proof modules. Eight support modules and thirteen contract modules bring the named audit to 515 declarations. Its 415 theorem commands include 61 support and 32 contract commands.

2.2 Reading the Lean notation

`Fin d → R` means a vector with exactly d coordinates over R. In rectangular lifting, d counts unknowns and e counts equations. In a parameter split, `Fin (e+k) → R` separates e solved coordinates and k free parameters. A theorem using `Fin (m+k)` may instead have m solved coordinates and e equations. The distinction matters when a fiber has p^{m-e} elements.

`A →ₗ[R] B` is a linear map, while `A →L[R] B` is a continuous linear map. An arrow decorated `≃L[R]` is a continuous linear equivalence. The analytic derivative is a continuous linear map. A raw matrix

becomes one through the explicitly defined `matrixCLM`. `IsUnit J` for a square matrix means it is invertible in the matrix ring. It does not mean each matrix entry is a scalar unit.

`ZMod p` is the residue ring. The source uses `Fact p.Prime` when it needs a finite field, rank, or p-adic objects. Some affine and integer polynomial statements require only $p > 0$ or $q > 0$. Their quotient algebra does not require primality. A hypothesis `1 ≤ L` supplies the depth needed for polynomial first-order linearization. The affine depth-zero case is valid because an affine map has no quadratic Taylor remainder.

`∃! b, P b` means there is exactly one object satisfying the full predicate P. A local root theorem's predicate includes its residue ball. It does not imply that the polynomial has only one root in the entire field. Likewise, an analytic graph theorem supplies open parameter and solved-coordinate neighborhoods. Global uniqueness is outside its conclusion.

3 Exact coordinates before arithmetic estimates

3.1 Residue projectors are coefficient filters

Contract. For an additive commutative monoid R, a positive modulus N, and a sequence $a : \mathbb{N} \to R$, the residue packets sum to a. The coefficient-level idempotent and orthogonality laws need only a zero element.

Lines 38–46; `KomorebiPaperIX.ResiduePackets.packet_reconstruction`.

```
theorem packet_reconstruction [AddCommMonoid R] {N : ℕ} (hN : 0 < N) (a : ℕ → R) :
    ∑ j ∈ Finset.range N, packet N j a = a := by
  funext n
  simp only [Finset.sum_apply, packet]
  rw [Finset.sum_eq_single (n % N)]
  · simp
  · intro b _ hb
    simp [Ne.symm hb]
  · simp [Nat.mod_lt n hN]
```

Proof reading. Function extensionality reduces the sequence identity to one coefficient n. The only nonzero summand is indexed by n mod N. `Finset.sum_eq_single` separates that summand from the rest. `Nat.mod_lt n hN` proves its address lies inside the chosen finite range. Positivity justifies the finite address set. There is no division by N and no root-of-unity hypothesis.

A residue projector is a selector on the degree address. Applying it twice keeps the same coefficients. Projectors with distinct addresses cannot both retain the same degree. Refinement to a multiple modulus groups fine addresses according to their coarse remainder. These are exact identities before reduction modulo a prime. The power-series versions use coefficient extensionality, so the same proof mechanism acts on `PowerSeries R`.

3.2 Triangular reconstruction constructs the solution

The source does not assume that an infinite triangular system already has a solution. It defines forward substitution recursively, using only indices $k < n$ at row n. The `termination_by n` declaration verifies the recursion decreases. Over a field the formula is defined even when a diagonal vanishes, but the row equation and uniqueness theorem explicitly require nonzero diagonal entries.

Contract. F is a field, T is an infinite coefficient array, every T_{nn} is nonzero, and a satisfies every finite row equation. Then a equals the recursively constructed solution.

Lines 35–49; `KomorebiPaperIX.Triangular.solve_unique`.

```
theorem solve_unique (T : ℕ → ℕ → F) (b a : ℕ → F)
    (hT : ∀ n, T n n ≠ 0)
    (ha : ∀ n, (∑ k : Fin n, T n k * a k) + T n n * a n = b n) :
```

```
    a = solve T b := by
  funext n
  induction n using Nat.strong_induction_on with
  | h n ih =>
    have hs : (∑ k : Fin n, T n k * a k) = ∑ k : Fin n, T n k * solve T b k := by
      apply Finset.sum_congr rfl
      intro k _
      rw [ih k k.isLt]
    have hr := ha n
    rw [hs] at hr
    have he := solve_row T b hT n
    exact (mul_left_cancel₀ (hT n)) (add_left_cancel (hr.trans he.symm))
```

Proof reading. Strong induction first replaces each earlier unknown a_k by the recursive value. The two row equations then have identical earlier sums. Cancelling the sum leaves equality after multiplication by T_{nn}. Nonzero diagonal cancellation gives equality of the current entries. Function extensionality assembles all rows. The argument needs no convergence or boundedness of an infinite matrix. Each row is a finite calculation.

Contract. For integer coefficients and integer right-hand side, the product $D_n = \prod_{r=0}^{n} T_{rr}$ clears the rational solution's denominator at row n. Every diagonal remains nonzero.

Lines 106–127; `KomorebiPaperIX.Triangular.solve_denominator_integral`.

```
theorem solve_denominator_integral (T : ℕ → ℕ → ℤ) (b : ℕ → ℤ)
    (hT : ∀ n, T n n ≠ 0) (n : ℕ) :
    Integral ((denominator T n : ℚ) * solve (fun i j => (T i j : ℚ)) (fun i => (b i : ℚ)) n) := by
  induction n using Nat.strong_induction_on with
  | h n ih =>
    cases n with
    | zero =>
      simp only [denominator_zero, solve_zero]
      have h0 : (T 0 0 : ℚ) ≠ 0 := Int.cast_ne_zero.mpr (hT 0)
      rw [mul_div_cancel₀ _ h0]
      exact integral_int _
    | succ n =>
      rw [solve_step, denominator_succ, Int.cast_mul]
      have hn : (T (n + 1) (n + 1) : ℚ) ≠ 0 := Int.cast_ne_zero.mpr (hT (n + 1))
      rw [mul_assoc, mul_div_cancel₀ _ hn]
      rw [mul_sub, Finset.mul_sum]
      apply integral_sub (integral_mul (integral_int _) (integral_int _))
      apply integral_sum
      intro k _
      rw [mul_left_comm]
      exact integral_mul (integral_int _)
        (integral_promote (denominator_dvd T (Nat.le_of_lt_succ k.isLt)) (ih k k.isLt))
```

Proof reading. The induction separates row zero from a successor row. At a successor, $D_{n+1} = D_n T_{n+1,n+1}$ cancels the new denominator. Earlier denominators divide D_n, allowing `integral_promote`

to place every earlier term under one common integer multiplier. The proof establishes an integral multiple of the actual recursive solution. It does not assert that the product is the smallest denominator. Negative diagonal entries are allowed, since only nonvanishing and divisibility are used.

3.3 EGF multiplication is Hurwitz convolution

Contract. Over a characteristic-zero field in the source coordinates, encoding divides the nth sequence entry by $n!$, and decoding multiplies the power-series coefficient by $n!$. The two operations are inverse.

Lines 52–60; `KomorebiPaperIX.EGF.egf_product`.

```
theorem egf_product (h a : ℕ → K) (n : ℕ) :
    decode (encode h * encode a) n =
      ∑ r ∈ Finset.range (n+1), (n.choose r : K) * h r * a (n-r) := by
  simp only [decode, coeff_mul, Finset.Nat.sum_antidiagonal_eq_sum_range_succ_mk,
    encode_coeff, Finset.sum_mul]
  apply Finset.sum_congr rfl
  intro r hr
  rw [Nat.cast_choose K (Finset.mem_range_succ_iff.mp hr)]
  ring
```

The binomial coefficient in the displayed convolution records the factorial change of coordinates. Ordinary power-series multiplication uses an unweighted coefficient convolution. The exponential coordinate multiplication uses $\binom{n}{k}$. A raw formal coefficient must therefore be distinguished from its degree-indexed generalized Euler number. Replacing one by the other without restoring the factorial changes p-adic precision.

Proof reading. The proof unfolds the product coefficient over finitely many pairs with sum n, substitutes the encoded entries, and proves the factorial identity behind the binomial coefficient. Nonzero factorial casts justify field cancellation. The resulting law transports exact inverse identities and reconstruction statements into sequence coordinates. The formal series need not converge analytically.

3.4 Finite differences and the Mahler basis

Contract. R is a commutative ring. The finite Mahler identity reconstructs any sequence value $f(n)$ from its first $n+1$ forward-difference coefficients. All sums are finite.

Lines 55–57; `KomorebiPaperIX.Mahler.finite_mahler_reconstruction`.

```
theorem finite_mahler_reconstruction (f : ℕ → R) (n : ℕ) :
    f n = ∑ r ∈ Finset.range (n+1), coefficient f r * (n.choose r : R) := by
  simpa [coefficient, difference, nsmul_eq_mul, mul_comm] using shift_eq_sum_fwdDiff_iter 1 f n 0
```

The change of basis is

$$f(n) = \sum_{k=0}^{n} \binom{n}{k} \Delta^k f(0).$$

The inverse relation expresses $\Delta^k f(0)$ as a signed binomial sum of the first $k+1$ values. No infinite Mahler-series convergence theorem is required for this finite identity. Later p-adic bounds put valuations on these exact coefficients and turn a finite coordinate theorem into precision estimates.

A weight condition on the kth coefficient controls higher differences by divisibility. The weighted difference modules formulate that condition explicitly and prove the translation by finite sums. A theorem about all orders therefore depends on the weight hypothesis for all the relevant coefficients. It cannot be inferred from a small computed prefix.

3.5 Finite units and Chinese remainders

Contract. For a prime p and $L \in \mathbb{N}$, the finite-unit module decomposes a unit modulo p^{L+1} into the source's torsion and principal components. Its uniqueness proof uses the uniquely lifted torsion representative and cancellation.

Lines 134–144; `KomorebiPaperIX.FiniteUnits.decomposition_unique`.

```
theorem decomposition_unique (L : ℕ) (a w v : (ZMod (p^(L+1)))ˣ)
    (ha : w*v=a) (hw : w^(p-1)=1) (hv : residueUnits p L v=1) :
    w=omega p L a ∧ v=principal p L a := by
  have hr : residueUnits p L w=residueUnits p L a := by
    rw [← ha, map_mul, hv, mul_one]
  have he := omega_unique p L a w hw hr
  refine ⟨he, ?_⟩
  subst w
  apply mul_left_cancel (a := omega p L a)
  rw [ha]
  exact (finite_unit_decomposition p L a).1.symm
```

The displayed source theorem applies to prime p, including $p = 2$. It compares the reduced residue of w with that of a, invokes the uniquely lifted torsion representative, and then cancels that factor to identify the principal component. For odd primes the familiar coprime torsion/principal orders explain this structure, but the Lean endpoint uses its proved lifted-root uniqueness. The dyadic development additionally supplies the finer sign/principal decomposition in its separate module.

Contract. The displayed CRT equivalence identifies addresses modulo $2^a M$ with addresses modulo 2^a and M, where M is odd. Oddness proves the coprimality needed by the ring equivalence.

Lines 13–17; `KomorebiPaperIX.CRTPackets.addressEquiv`.

```
def addressEquiv (a M : ℕ) (hM : Odd M) :
    ZMod (2^a*M) ≃+* ZMod (2^a) × ZMod M :=
  ZMod.chineseRemainder (by
    apply Nat.Coprime.pow_left
    exact (Nat.coprime_two_left).mpr hM)
```

Proof reading. The equivalence carries both directions and their inverse laws. Packet factorization then compares the retained degree address through that equivalence. The algebra concerns residue addresses and coefficients, not an analytic tensor-product completion. This is why the finite arithmetic layer can be reused without adding convergence claims.

4 From packet coefficients to Nishibuchi's conjecture

4.1 Rational precision includes zero

Contract. Fix a prime p. For a rational number x and an integer L, `precision p L x` means $x = 0$ or $v_p(x) \geq L$.

Lines 18–18; `KomorebiPaperIX.GlobalEulerBoundary.precision`.

```
def precision (L : ℤ) (x : ℚ) : Prop := x=0 ∨ L≤padicValRat p x
```

The explicit zero alternative avoids relying on a finite-valued implementation of the valuation at zero. It also allows negative precision exponents: dividing an integral p-adic quantity by an integer normalization can reduce the bound below zero. The source's `precision` predicate is not divisibility inside $\mathbb{Q}$, where every nonzero scalar is a unit. Divisibility statements live in $\mathbb{Z}$, $\mathbb{Z}_p$, or a stated coefficient ring, and a separate bridge transports them to rational valuations.

Let $N = mp$ with $m > 0$, $m \mid p-1$, and $0 \leq j < N$. The unnormalized sequence is lifted to $\mathbb{Z}_p$ only after integrality is proved. Its higher differences use a packet step that is a multiple of $(p-1)/m$. The arithmetic core supplies a bound of the form

$$p^{\min(k,mn)} \mid \Delta^k_{u(p-1)/m} A_{N,j}(n).$$

Contract. The prime hypothesis, divisor condition $m \mid p-1$, positive m, and address bound $j < mp$ are all required. The sequence in the theorem is the actual integral lift of the unnormalized generating-series coefficients.

Lines 79–85; `KomorebiPaperIX.AllResidueEuler.global_unnormalized_bound`.

```
theorem global_unnormalized_bound (m j u k n : ℕ)
    (hm : m|p-1) (hmpos : 0<m) (hj : j<m*p) :
    (p:ℤ_[p])^(min k (m*n)) |
      difference (u*((p-1)/m)) k (unnormalizedLift p m j hm hmpos hj) n := by
  exact packet_divisibility p m hm hmpos
    (unnormalizedSeries (m*p) (Nat.mul_pos hmpos (Fact.out : p.Prime).pos).ne' j)
    (all_residue_flat p m hm hmpos j hj) (unnormalized_support (m*p) _ j) u k n
```

Proof reading. The theorem invokes `packet_divisibility` with three distinct inputs: the actual unnormalized series, its all-residue flatness theorem, and its support theorem. The interface does not permit an arbitrary series to acquire this bound. The difficult recurrence/flatness propagation resides in its imported arithmetic modules, whose explicit declarations remain in the complete catalog. The short endpoint proof is a composition of those established inputs, not an independent numerical experiment.

The normalization transfer must retain its factorial loss. A bound on $A_{N,j}$ gives a bound on the normalized coefficient only after subtracting the valuation of the normalization factor. The retained normalization counterexample shows why a universal loss-free transfer is false. A theorem with a stronger hypothesis in a standard window must not be silently extended to every row without that transfer.

4.2 A Newton expansion converts differences into a period

Contract. R is a commutative ring, p is prime, $r > 0$, and every order-k difference at the chosen base point is divisible by $p^{\min(k,r)}$. The conclusion is one period displacement at that base point.

Lines 46–67; `KomorebiPaperIX.PeriodFromDifferences.prime_power_period`.

```
theorem prime_power_period (f : ℕ → R) (h n r : ℕ) (hr : 0<r)
    (H : ∀ k, (p:R)^(min k r) | difference h k f n) :
    (p:R)^r | f (n+p^(r-1)*h)-f n := by
  rw [displacement_expansion]
  apply Finset.dvd_sum
  intro k hk
  have hkr : k+1 ≤ p^(r-1) := by have := Finset.mem_range.mp hk; omega
  by_cases hsmall : k+1 < r
  · have hd := H (k+1)
    rw [min_eq_left (by omega : k+1≤r)] at hd
    obtain ⟨a,ha⟩ := hd
    rw [ha, ←mul_assoc]
    have hb := binomial_precision p (r-1) (k+1) (by omega) hkr
    rw [Nat.sub_add_cancel (by omega : 1≤r)] at hb
    have hcast : (p:R)^r | (((p^(r-1)).choose (k+1):ℕ):R)*(p:R)^(k+1) := by
      obtain ⟨b,hb⟩ := hb
      refine ⟨(b:R), ?_⟩
      simpa only [Nat.cast_mul, Nat.cast_pow] using congrArg (fun a:ℕ => (a:R)) hb
    exact dvd_mul_of_dvd_left hcast a
  · have hd := H (k+1)
    rw [min_eq_right (by omega : r≤k+1)] at hd
    exact dvd_mul_of_dvd_right hd _
```

Proof reading. The displacement expansion writes the change at p^{r-1} steps as a finite sum of binomial coefficients times higher differences. For an order $k < r$, the difference contributes p^k and the binomial coefficient contributes the missing digits. For $k \geq r$, the difference itself supplies the whole p^r. The proof explicitly splits these cases. It does not require division in R. Integer divisibility of binomial coefficients is cast into the ring.

When the packet bound supplies $\min(k, mn)$, a sufficient threshold $r \leq mn$ gives the hypothesis above. The endpoint uses the simpler threshold $n \geq r$ valid across the allowed parameter families. This is a sufficient starting index. The proof makes no assertion that the threshold is the smallest possible one.

4.3 Packet address and degree address are different indices

The generalized Euler sequence is degree-indexed, while a residue packet sequence is indexed by its packet row n. The source proves its reciprocal generating identity before presenting the external conjecture: the unnormalized series multiplied by the original exponential section is X^j. Thus the degree sequence is connected to the defining source series, not just assigned a suggestive name.

Set $q = \operatorname{lcm}(2, p-1)$. The allowed divisor branch has packet shift $p^{r-1}(q/m)$, which becomes degree shift qp^r after multiplication by mp. The exceptional allowed family is $p = 2, m = 2$. For odd primes, the option $m = 2$ already belongs to the divisor branch because $2 \mid p-1$.

Contract. The integral quotient q/m is justified by $m \mid p-1$ or $m = 2$. The degree identity also requires $r > 0$ so that $p^r = p^{r-1}p$.

Lines 89–96; `KomorebiPaperIX.NishibuchiCompletion.original_degree_shift`.

```
theorem original_degree_shift (m n r:ℕ) (h:m|p-1 ∨ m=2) (hr:0<r) :
    m*p*(n+p^(r-1)*(displacementFactor p/m)) =
      m*p*n+displacementFactor p*p^r := by
  have hq := Nat.mul_div_cancel' (modulus_divides_displacement p m h)
  have hp : p^r=p^(r-1)*p := by rw [←pow_succ,Nat.sub_add_cancel (by omega : 1≤r)]
  calc
    _ = m*p*n+(m*(displacementFactor p/m))*(p^(r-1)*p) := by ring
    _ = _ := by rw [hq,hp]
```

Proof reading. The proof establishes $m(q/m) = q$ and rewrites the exponent successor using $r \geq 1$. The remaining identity is ring arithmetic. Natural-number division is not an approximate operation here: its exact cancellation is justified by divisibility. Forgetting that hypothesis would change the intended displacement.

4.4 The completion theorem exposes its full range

Contract. p is prime, $m > 0$, $j < mp$, and $m \mid p-1$ or $m = 2$. Every degree-mpn coefficient is p-integral. For each $r > 0$, a positive starting index $n_0 \leq r$ exists, and all $n \geq n_0$ satisfy the stated degree displacement congruence.

Lines 128–141; `KomorebiPaperIX.NishibuchiCompletion.conjecture_3_9`.

```
theorem conjecture_3_9 (m j:ℕ) (hmpos:0<m) (hj:j<m*p) (h:m|p-1 ∨ m=2) :
    (∀n, ((generalizedEuler (m*p) (Nat.mul_ne_zero hmpos.ne' (Fact.out:p.Prime).ne_zero)
      j (m*p*n):ℚ):ℚ_[p]) ∈ PadicInt.subring p) ∧
    ∀r:ℕ, 0<r → ∃n0:ℕ, 0<n0 ∧ n0≤r ∧ ∀n:ℕ, n0≤n →
      precision p (r:ℤ)
        (generalizedEuler (m*p) (Nat.mul_ne_zero hmpos.ne' (Fact.out:p.Prime).ne_zero) j
          (m*p*n+displacementFactor p*p^r)-
         generalizedEuler (m*p) (Nat.mul_ne_zero hmpos.ne' (Fact.out:p.Prime).ne_zero) j (m*p*n))
          ↪  := by
  constructor
  · intro n
    rw [generalizedEuler_at_packet]
    exact full_integrality p m j n hmpos hj h
  · intro r hr
    exact ⟨r,hr,le_rfl,fun n hn => full_period_degree p m j r n hmpos hj h hr hn⟩
```

Proof reading. The conjunction separates integrality from eventual periodicity. Integrality first rewrites the degree coefficient at mpn into the packet coefficient, then invokes the proved integral lift. The period witness is exactly $n_0 = r$. The degree-period theorem is supplied for every later n, so the existential threshold is constructive at the statement level even though the ambient development uses classical mathematics.

This endpoint proves the original range recorded in the retained external comparison. The prime, parameter, address, and displacement conventions all remain visible. It supplies an eventual period and integrality, while the later dyadic modules prove sharper exact periods in their own specified family. It does not extend the conjecture to arbitrary m, identify an optimal starting index, or establish a least period in every prime family.

5 Reconstruction, sharp dyadic displacement, and isometry

5.1 Odd reconstruction separates identity from precision

Contract. The odd Euler reconstruction is an exact identity over $\mathbb{Z}$ for every odd length p, even when that length is composite. The prime-only corollary is a specialization.

Lines 80–87; `KomorebiPaperIX.OddEuler.odd_euler_reconstruction`.

```
theorem odd_euler_reconstruction (p n : ℕ) (hp : Odd p) :
    eulerEven n = ∑ m ∈ Finset.range (n+1),
      ((2*n).choose (2*m) : ℤ) * inverseEven m * (p:ℤ)^(2*m) * source p (2*(n-m)) := by
  have h := reconstruction_all_indices p (2*n) hp (even_two_mul n)
  rw [euler_even] at h
  rw [sum_even_terms (fun r => ((2*n).choose r : ℤ)*inverseCoeff r*(p:ℤ)^r*source p (2*n-r))
    (fun m => by rw [inverse_odd_zero _ (by exact ⟨m,rfl⟩)]; simp)] at h
  simpa only [inverseEven, Nat.mul_sub_left_distrib] using h
```

The series identity underlying this theorem multiplies a rescaled squared reciprocal series by the even part of the source series. It then extracts exponential coordinates, proves odd terms vanish, and changes from arbitrary indices to even indices. The coefficient $\binom{2n}{2m}$ and the factor p^{2m} come from that exact coordinate calculation. The source separately proves the signed zigzag identity for the inverse coefficient. The sign is not inferred from a plotted prefix.

Contract. Truncation includes $m = 0, \dots, \min(M, n)$. Every omitted summand has the stated even weight.

Lines 144–150; `KomorebiPaperIX.OddEuler.odd_euler_supercongruence`.

```
theorem odd_euler_supercongruence (p n M : ℕ) (hp : Odd p) :
    (p:ℤ)^(2*M+2) ∣ eulerEven n - truncation p n M := by
  rw [odd_euler_reconstruction p n hp]
  have h := Filtered.even_weight_tail (p:ℤ)
    (fun m => ((2*n).choose (2*m):ℤ)*inverseEven m*source p (2*(n-m))) n M
  rw [filter_range_le] at h
  simpa only [truncation, mul_assoc, mul_left_comm, mul_comm] using h
```

Proof reading. The full sum is split into retained and omitted indices. For an omitted m, $m \geq M+1$, so its factor p^{2m} contains p^{2M+2}. The tail theorem is therefore an integer divisibility statement. If $M \geq n$, there are no omitted terms and the truncation equals the full value exactly. No convergence question arises, because each row has only $n+1$ terms.

5.2 The dyadic transform has its own coefficients

Contract. The dyadic Euler reconstruction uses its specifically constructed dyadic source and integral transform coefficients. It is not obtained by substituting $p = 2$ into a theorem whose hypothesis is `Odd p`.

Lines 26–45; `KomorebiPaperIX.DyadicReconstruction.dyadic_reconstruction`.

```
theorem dyadic_reconstruction (n : ℕ) :
    (eulerEven n : ℚ) = ∑ m ∈ Finset.range (n+1), summand n m := by
  have hs := congrArg (fun f => EGF.decode f (2*n)) dyadic_reconstruction_series
  have hp := EGF.egf_product (EGF.decode (rescale (2:ℚ) inverseSeries))
    (EGF.decode DyadicSeries.sourceSeries) (2*n)
  simp only [EGF.encode_decode] at hp
  have he : EGF.decode OddPower.sech (2*n) = (eulerEven n : ℚ) := by
    rw [← euler_egf, EGF.decode_encode, euler_even]
  rw [hp,he] at hs
  rw [OddEuler.sum_even_terms _ (fun m => by
    simp [EGF.decode, coeff_rescale, inverse_odd_coefficient])] at hs
  rw [← hs]
  apply Finset.sum_congr rfl
  intro m hm
  simp only [EGF.decode, coeff_rescale, inverse_even_coefficient,
    ← Nat.mul_sub_left_distrib, source_even_coefficient]
  unfold summand
  rw [Nat.descFactorial_eq_factorial_mul_choose]
  push_cast
  field_simp
```

The dyadic coefficient analysis includes the valuation of the transform coefficients and of the source terms. Its sharper truncation bound is

$$2^{4M+3} \mid E_{2n} - T_M(n)$$

in the p-adic integral coordinates. That exponent comes from the omitted index bound and the valuation of the dyadic coefficient κ_m. The first omitted term is separately analyzed to show sharpness where the stated row/depth hypotheses apply.

Contract. The displayed uniform sharp tail bound holds for all natural n, M. Full-depth truncations have zero error. A nonzero exact valuation for the first omitted error has its own boundary hypotheses.

Source: `strengthening/formalization/DyadicSharpTower.lean`.

Lines 176–189; `KomorebiPaperIX.DyadicSharpTower.dyadic_supercongruence_sharp`.

```
theorem dyadic_supercongruence_sharp (n M : ℕ) :
    (2:ℤ_[2])^(4*M+3) | (EulerCoefficients.eulerEven n:ℤ_[2]) - truncationPadic n M := by
  rw [dyadic_reconstruction_padic, truncationPadic, ← OddEuler.filter_range_le]
  classical
  rw [← Finset.sum_filter_add_sum_filter_not (Finset.range (n+1)) (fun m => m ≤ M)
    (summandPadic n), add_sub_cancel_left]
  apply Finset.dvd_sum
  intro m hm
  obtain ⟨_, hm⟩ := Finset.mem_filter.mp hm
  have hpos : 0 < m := by omega
  have hpow : (2:ℤ_[2])^(4*M+3) | (2:ℤ_[2])^(4*m-1) := pow_dvd_pow 2 (by omega)
```

```
  rw [summandPadic_kappa]
  exact dvd_mul_of_dvd_left (dvd_mul_of_dvd_left
    (hpow.trans (kappaPadic_divisible m hpos)) _) _
```

Proof reading. The proof splits the finite reconstruction sum by the predicate $m \le M$. In the complement, $m > 0$, allowing the coefficient divisibility theorem. The exponent comparison $4M + 3 \le 4m - 1$ supplies the uniform bound. The calculation is a universal finite-sum proof. Checking numerical rows is independent bounded evidence about examples, not the reason this theorem holds.

5.3 Dominance gives an exact valuation

A lower bound on every difference does not itself determine a least period. The sharp modules isolate a nonzero leading Newton contribution whose valuation is smaller than all remaining contributions. In a nonarchimedean field, that strict separation forces the whole sum to have the leading term's valuation. A tie in the lowest valuations could allow cancellation, so the strict tail estimate is essential.

Contract. For each of the four normalized dyadic addresses $j < 4$ and each positive natural displacement T, the difference is nonzero and its valuation is $v_2(T) + b_j$, where b_j is the explicitly defined prefix bonus.

Source: `strengthening/formalization/DyadicFourLeastPeriods.lean`.

Lines 26–32; `KomorebiPaperIX.DyadicFourLeastPeriods.normalized_exact_displacement`.

```
theorem normalized_exact_displacement (j n T : ℕ) (hj : j < 4) (hT : 0 < T) :
    coeff 4 j (n + T) - coeff 4 j n ≠ 0 ∧
    padicValRat 2 (coeff 4 j (n + T) - coeff 4 j n) =
      (padicValNat 2 T : ℤ) + (DyadicFourPrefix.bonus j : ℤ) := by
  exact NewtonDominance.exact_displacement (coeff 4 j) (DyadicFourPrefix.bonus j)
    (DyadicFourPrefix.first_nonzero j hj) (DyadicFourPrefix.first_valuation j hj)
    (DyadicFourPrefix.tail_precision j hj) n T hT
```

Proof reading. The endpoint supplies four inputs to the generic Newton dominance theorem: first nonvanishing coefficient, its exact valuation, the strict tail-precision bound, and positivity of T. The conjunction records nonzero difference as well as valuation equality. At $T = 0$ the difference is zero, so the exact nonzero statement would be false. The theorem excludes that case.

The normalized bonus and the unnormalized bonus are distinguished. The latter incorporates the relevant factorial normalization. The exact-period classification is stated using the actual unnormalized sequence and its own bonus c_j.

Contract. A period in the source is a positive displacement. At precision r, the four-address unnormalized packet has period T exactly when $T > 0$ and $2^{\max(r - c_j, 0)} \mid T$. Natural subtraction in the source expresses the maximum with zero.

Source: `strengthening/formalization/DyadicFourLeastPeriods.lean`.

Lines 58–62; `KomorebiPaperIX.DyadicFourLeastPeriods.period_iff_dvd`.

```
theorem period_iff_dvd (j r T : ℕ) (hj : j < 4) :
    ExactPeriodClassification.IsPeriod 2 (unnormalized 4 j) r T ↔
      0 < T ∧ 2 ^ (r - originalBonus j) ∣ T :=
  ExactPeriodClassification.isPeriod_iff_dvd 2 (unnormalized 4 j)
    (originalBonus j) r T (fun n S hS => exact_displacement j n S hj hS)
```

Proof reading. Exact displacement reduces a universal congruence at every base row to an inequality $v_2(T) + c_j \geq r$. The generic classification translates that inequality into divisibility. If $r \leq c_j$, the exponent is zero and every positive displacement is a period. The least positive one is then one. If $r > c_j$, the least period is the indicated power of two. Zero remains excluded by the period predicate.

5.4 The normalized packet extends to a p-adic isometry

Contract. For $j < 4$, the appropriately rescaled rational packet extends uniquely to an isometric equivalence $\mathbb{Z}_2 \simeq \mathbb{Z}_2$ agreeing with its values at natural integers.

Source: `strengthening/formalization/DyadicFourIsometry.lean`.

Lines 78–84; `KomorebiPaperIX.DyadicFourIsometry.existsUnique_isometry`.

```
theorem existsUnique_isometry (j : ℕ) (hj : j < 4) :
    ∃! e : ℤ_[2] ≃ᵢ ℤ_[2], ∀ n : ℕ,
      (e (n : ℤ_[2]) : ℚ_[2]) = (DyadicPacketRational.packet j n : ℚ_[2]) := by
  refine ⟨isometry j hj, isometry_nat j hj, ?_⟩
  intro e he
  apply IsometryEquiv.ext
  exact congrFun (unique_continuous j hj e e.continuous he)
```

Proof reading. The source first constructs the isometry and proves agreement on natural inputs. Uniqueness then applies the theorem that continuous extensions agreeing on the dense natural subset agree everywhere. The isometric-equivalence conclusion includes surjectivity, so a merely distance-preserving embedding would not suffice. The rescaling and basepoint translation are part of the packet definition. The raw unnormalized sequence has a different image ball.

The concrete normalization is

$$(b_0, b_1, b_2, b_3) = (1, 1, 4, 2), \qquad (c_0, c_1, c_2, c_3) = (1, 1, 5, 3),$$

where b_j are normalized-coefficient bonuses and c_j are original-coefficient bonuses. Writing $U_{4,j}$ for the unnormalized packet, the isometric packet is

$$P_j(n) = \frac{U_{4,j}(n) - j!}{2^{c_j}}.$$

Its extension maps the unit ball onto the unit ball. Undoing that formula shows the original unnormalized extension has image $j! + 2^{c_j}\mathbb{Z}_2$.

This distinction explains the manuscript's affine-image consequence: undoing the normalization sends the unit ball to a specified affine residue ball. Neither the period theorem nor the isometry endpoint proves an unrestricted arithmetic statement for all moduli. Their exact hypotheses identify the four-address dyadic family being closed.

6 Finite obstructions and a common correction

6.1 Residuals become quotient classes

Let $A : \mathbb{Z}^d \to \mathbb{Z}^e$ be linear and suppose $Ax - b = p^L r$. A same-residue correction has the form $y = x + p^L t$. Its next residual is $p^L(r + At)$. Thus the next level exists precisely when $\bar{r} + \bar{A}\bar{t} = 0$ modulo p. The obstruction is the class of $\bar{r}$ in the cokernel of the digit map. Zero class means the residual can be cancelled by some digit.

Contract. The affine theorem requires $p > 0$, but neither primality nor $L \geq 1$. It supplies an integer residual, a finite obstruction space, and an equivalence between same-residue next-level lifts and zero obstruction.

Source: `polynomial_regularity/formalization/AffineObstructionCore.lean`.

Lines 152–171; `Komorebi.NativeAffine.finite_affine_obstruction`.

```
theorem finite_affine_obstruction (p L : ℕ) (hp : 0 < p)
    (A : (Fin d → ℤ) →ₗ[ℤ] (Fin e → ℤ)) (b : Fin e → ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (p : ℤ)^L ∣ (A x - b) i) :
    ∃ r : Fin e → ℤ,
      A x - b = (p : ℤ)^L • r ∧
      Finite (Obstruction p A) ∧
      ((∃ y : Fin d → ℤ,
        (∀ i, (p : ℤ)^L ∣ (y - x) i) ∧
        (∀ i, (p : ℤ)^(L+1) ∣ (A y - b) i)) ↔ obstruction p A r = 0) := by
  obtain ⟨r, hr⟩ := residual_exists p L A b x hx
  let : NeZero p := ⟨Nat.ne_of_gt hp⟩
  refine ⟨r, hr, obstruction_finite p A, ?_⟩
  rw [← affine_lift_iff_obstruction_zero p L hp A b r x hr]
  constructor
  · rintro ⟨y, hy, hAy⟩
    obtain ⟨t, rfl⟩ := (same_residue_iff ((p : ℤ)^L) x y).mp hy
    exact ⟨t, hAy⟩
  · rintro ⟨t, ht⟩
    exact ⟨x + (p : ℤ)^L • t,
      (same_residue_iff ((p : ℤ)^L) x _).mpr ⟨t, rfl⟩, ht⟩
```

Proof reading. Residual existence is coordinatewise integer divisibility. The source installs `NeZero p` to make the residue ring finite. It then rewrites arbitrary same-residue y as $x + p^L t$ using `same_residue_iff`. That rewriting supplies both directions: a successful lift gives a digit, and a successful digit constructs a lift. At $L = 0$, $p^L = 1$ and all integer points are in the same initial residue class. The affine calculation still works.

The finite quotient should not be confused with a rank formula. Finiteness over `ZMod p` works for any nonzero modulus. A dimension argument and the count $p^{d-\rho}$ need p prime so that the residue ring is a field.

Contract. Adding an image vector As to the residual does not change its obstruction class. No rank or surjectivity assumption is required.

Source: `polynomial_regularity/formalization/AffineObstructionCore.lean`.

Lines 70–79; `Komorebi.NativeAffine.obstruction_representative_independent`.

```
theorem obstruction_representative_independent (p : ℕ)
    (A : (Fin d → ℤ) →ₗ[ℤ] (Fin e → ℤ)) (r : Fin e → ℤ) (s : Fin d → ℤ) :
    obstruction p A (r + A s) = obstruction p A r := by
  change Submodule.Quotient.mk (reduction p e (r + A s)) = _
  rw [map_add, Submodule.Quotient.mk_add]
  have hs : (Submodule.Quotient.mk (reduction p e (A s)) : Obstruction p A) = 0 := by
    rw [Submodule.Quotient.mk_eq_zero]
    exact ⟨s, rfl⟩
  rw [hs, add_zero]
  rfl
```

Proof reading. The reduction map preserves addition. The quotient map then sends the reduced image vector to zero because it belongs to the digit-map range. Adding zero leaves the class unchanged. This is an exact statement about replacing a representative by a known image term. It does not establish the separate open problem of presentation invariance under every possible native equivalence.

6.2 Two solvable equations need one shared digit

For two maps A and B with separately soluble digit equations, choose individual correcting digits u and v. The first allows the coset $u + \ker \bar{A}$ and the second allows $v + \ker \bar{B}$. A common digit exists exactly when these cosets meet, equivalently when

$$u - v \in \ker \bar{A} + \ker \bar{B}.$$

The shared obstruction is the quotient of the common digit space by that sum. It is generally not the product of the two separate cokernels: those cokernels already vanish for the individually soluble pair, while the common-correction obstruction can still be nonzero.

Contract. Both chosen correcting digits must solve their own reduced residual equations. The theorem then identifies a joint next lift with equality of their classes in the shared quotient. Positivity of the modulus is retained. $L = 0$ is allowed.

Source: `polynomial_lifting/formalization/AffineSharedCorrection.lean`.

Lines 60–75; `Komorebi.AffinePair.joint_lift_iff_class_eq`.

```
theorem joint_lift_iff_class_eq (p L : ℕ) (hp : 0 < p)
    (A : (Fin d → ℤ) →ₗ[ℤ] (Fin e → ℤ))
    (B : (Fin d → ℤ) →ₗ[ℤ] (Fin f → ℤ))
    (b r : Fin e → ℤ) (c s : Fin f → ℤ) (x u v : Fin d → ℤ)
    (hr : A x - b = (p : ℤ)^L • r)
    (hs : B x - c = (p : ℤ)^L • s)
    (hu : reduction p e (r + A u) = 0)
    (hv : reduction p f (s + B v) = 0) :
    HasJointLift p L A B b c x ↔
    (Submodule.Quotient.mk u : SharedObstruction p A B) =
      Submodule.Quotient.mk v := by
  rw [joint_lift_iff_digit_correction p L hp A B b r c s x hr hs,
```

```
    quotient_mk_eq_iff_common]
  exact exists_congr fun t => and_congr
    (digit_correction_iff_same_image p A r u t hu)
    (digit_correction_iff_same_image p B s v t hv)
```

Proof reading. The first rewrite translates joint lifting into a simultaneous digit equation. The second identifies class equality with a common representative. The final `exists_congr` compares the two individual conditions by their image fibers. This separates the existence of separate corrections from their compatibility. A claim that separate solvability automatically gives a joint lift would drop the equality of quotient classes and is false.

Changing u by a digit in $\ker \bar{A}$ or changing v by one in $\ker \bar{B}$ leaves the shared class unchanged. The quotient has the expected universal property for maps annihilating both kernels, and its finite image/range formulations are proved in separate modules. Those endpoint statements retain their own exact coefficient conventions.

6.3 A nonempty fiber is a kernel torsor

Contract. $A : U \to V$ is linear over a ring, and a base point u has been supplied in the fiber over b. The fiber is equivalent to $\ker A$ by subtraction from u and translation back. The ambient U, V are modules and additive commutative groups.

Source: `polynomial_regularity/formalization/FiniteFiberCounts.lean`.

Lines 75–79; `Komorebi.FiniteFiberCounts.fiberEquivKernel`.

```
def fiberEquivKernel (A : U →ₗ[K] V) (b : V) (u : Fiber A b) : Fiber A b ≃ A.ker where
  toFun t := difference A b t u
  invFun v := translate A b v u
  left_inv t := translate_difference A b t u
  right_inv v := difference_translate A b v u
```

Proof reading. The forward map sends t to $t - u$ and uses linearity to prove it lies in the kernel. The inverse sends a kernel vector v to $u + v$ and proves its image is b. The two inverse identities are cancellation in the additive group. The chosen u is an origin for a torsor, not a distinguished zero of the original fiber. An empty fiber admits no such choice.

Contract. For a linear digit map $A : (\mathbb{F}_p)^d \to (\mathbb{F}_p)^e$ with rank ρ, the fiber over b has $p^{d-\rho}$ elements when nonempty and zero elements otherwise.

Source: `polynomial_regularity/formalization/FiniteFiberCounts.lean`.

Lines 138–144; `Komorebi.FiniteFiberCounts.fiber_card`.

```
theorem fiber_card (A : (Fin d → ZMod p) →ₗ[ZMod p] (Fin e → ZMod p))
    (b : Fin e → ZMod p) :
    Nat.card (Fiber A b) = if Nonempty (Fiber A b) then p ^ (d - rank p A) else 0 := by
  classical
  split_ifs with h
  · exact fiber_card_of_nonempty p A b h
  · exact fiber_card_of_empty A b h
```

Proof reading. The nonempty branch transports cardinality through the fiber–kernel equivalence and uses rank-nullity plus the finite-field cardinality theorem. The empty branch has cardinality zero. The `if Nonempty` guard cannot be discarded for a general right-hand side. In a surjective case every right-hand side enters the nonempty branch. In an injective square case the nonempty fiber has exactly one element.

7 Actual polynomial Taylor expansion and lift counts

7.1 A quadratic remainder is proved algebraically

Contract. P is an actual multivariate polynomial with integer coefficients, $x, t \in \mathbb{Z}^d$, and $a \in \mathbb{Z}$. The directional derivative uses the evaluated formal partial derivatives. Then a^2 divides $P(x+at) - P(x) - aDP(x)t$.

Source: `polynomial_regularity/formalization/PolynomialTaylor.lean`.

Lines 41–53; `Komorebi.PolynomialTaylor.taylor_dvd`.

```
theorem taylor_dvd {d : ℕ} (P : MvPolynomial (Fin d) ℤ)
    (x t : Fin d → ℤ) (a : ℤ) :
    a ^ 2 | eval (fun i => x i + a * t i) P - eval x P - a * directional P x t := by
  induction P using MvPolynomial.induction_on with
  | C c => simp [directional]
  | add P Q hP hQ =>
    simp only [map_add, directional_add]
    convert dvd_add hP hQ using 2 <;> first | rfl | ring
  | mul_X P j hP =>
    obtain ⟨r, hr⟩ := hP
    refine ⟨r * (x j + a * t j) + directional P x t * t j, ?_⟩
    simp only [map_mul, eval_X, directional_mul, directional_X]
    linear_combination (x j + a * t j) * hr
```

Proof reading. The proof inducts on the multivariate polynomial syntax supplied by Mathlib: a constant, an addition, or multiplication by a variable. Constants have zero remainder. Sums add remainders. In the multiplication case, the induction witness r is combined with the linear term and the new variable displacement. `linear_combination` verifies the resulting polynomial identity. This is exact integer algebra and requires no differentiability or limiting argument.

The quadratic factor is what permits one-digit lifting. Put $a = q^L$. The remainder is divisible by q^{2L}. To neglect it modulo q^{L+1}, one needs $2L \geq L + 1$, hence $L \geq 1$. An affine map has no quadratic remainder and works at $L = 0$. A general polynomial does not. The source does not assume that an arbitrary nonlinear map comes with the same exact expansion.

7.2 One nonlinear step becomes a finite digit equation

Write $F(x) = q^L r$ coordinatewise. The actual Jacobian $J_F(x)$ is defined by formal partial derivatives evaluated at x, assembled as an integer linear map. The next-level equation is

$$F(x + q^L t) \equiv 0 \pmod{q^{L+1}} \quad \Longleftrightarrow \quad r + J_F(x)t \equiv 0 \pmod{q}.$$

Contract. $q > 0$, $L \geq 1$, and the given residual identity is exact. No prime assumption is needed for this equivalence.

Source: `polynomial_regularity/formalization/PolynomialObstruction.lean`.

Lines 97–108; `Komorebi.PolynomialObstruction.next_level_iff_digit`.

```
theorem next_level_iff_digit (q L : ℕ) (hq : 0 < q) (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (r : Fin e → ℤ) (hr : evalSystem F x = (q : ℤ)^L • r)
    (t : Fin d → ℤ) :
    (∀ i, (q : ℤ)^(L+1) | evalSystem F (x + (q : ℤ)^L • t) i) ↔
      reduction q e (r + jacobian F x t) = 0 := by
  rw [reduction_zero_iff]
  apply forall_congr'
  intro i
  rw [dvd_iff_of_dvd_sub _ _ _ (taylor_residual_dvd q L hL F x r hr t i)]
  have hqL : (q : ℤ)^L ≠ 0 := pow_ne_zero _ (Int.natCast_ne_zero.mpr (Nat.ne_of_gt hq))
  rw [pow_succ, mul_dvd_mul_iff_left hqL]
```

Proof reading. The source rewrites reduced zero as coordinatewise divisibility. Taylor remainder divisibility replaces the actual polynomial value by its residual-plus-linear term. The remaining factor q^L can be cancelled because $q > 0$ implies $q^L \neq 0$ in $\mathbb{Z}$. This cancellation gives the modulo-q equation. Both directions are proved, so the linearization is an exact next-digit criterion rather than an approximation.

Contract. The finite polynomial obstruction theorem starts with an approximate root modulo q^L and constructs its residual. The obstruction space uses the range of the actual Jacobian digit map.

Source: `polynomial_regularity/formalization/PolynomialObstruction.lean`.

Lines 141–154; `Komorebi.PolynomialObstruction.finite_polynomial_obstruction`.

```
theorem finite_polynomial_obstruction (q L : ℕ) (hq : 0 < q) (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (q : ℤ)^L | evalSystem F x i) :
    ∃ r : Fin e → ℤ,
      evalSystem F x = (q : ℤ)^L • r ∧
      Finite (Obstruction q (jacobian F x)) ∧
      ((∃ y : Fin d → ℤ,
        (∀ i, (q : ℤ)^L | (y - x) i) ∧
        (∀ i, (q : ℤ)^(L+1) | evalSystem F y i)) ↔
          obstruction q (jacobian F x) r = 0) := by
  obtain ⟨r, hr⟩ := residual_exists q L F x hx
  let : NeZero q := ⟨Nat.ne_of_gt hq⟩
  exact ⟨r, hr, obstruction_finite q (jacobian F x),
    polynomial_same_residue_lift_iff q L hq hL F x r hr⟩
```

Proof reading. Residual existence supplies r. The finite quotient construction is reused from the affine layer, while the polynomial same-residue theorem supplies the nonlinear lifting equivalence. Reusing the quotient algebra does not require treating the polynomial as affine. Its application is justified precisely at this one-digit Taylor interface.

A nonzero obstruction means no same-residue next lift exists. A zero obstruction proves existence for that step. It does not by itself prove persistence at all later depths: the Jacobian and the residual may change, and singular digit maps can develop higher obstructions. The smooth tower theorem adds surjectivity and proves it persists along the constructed tower.

7.3 Digits correspond to residue lifts, not all integer representatives

The lift-counting development distinguishes an integer lift from its residue class modulo p^{L+1}. Without that quotient, a soluble congruence has infinitely many integer representatives. `ActualLifts` is a finite type of residue lifts, and `DigitSolutions` is the finite fiber of the digit equation.

Contract. $q > 0$, $L \geq 1$, and $F(x) = q^L r$. The residue digit equivalence takes a digit solution modulo q to a residue lift modulo q^{L+1}.

Source: `polynomial_regularity/formalization/PolynomialLiftCounts.lean`.

Lines 43–48; `Komorebi.PolynomialLiftCounts.digitEquivLifts`.

```
def digitEquivLifts (q L : ℕ) (hq : 0 < q) (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (r : Fin e → ℤ) (hr : evalSystem F x = (q : ℤ)^L • r) :
    DigitSolutions q F x r ≃ ActualLifts q L F x :=
  Equiv.subtypeEquiv (digitEquivResidue q L hq x)
    (fun t => (digit_residue_solution_iff q L hq hL F x r hr t).symm)
```

Proof reading. The source first has an equivalence between all digit addresses and all residue addresses above the initial class. `Equiv.subtypeEquiv` then restricts it to the subsets satisfying the equations. The previous next-digit theorem supplies the required equivalence of predicates. Thus cardinalities are preserved through a genuinely bijective map. A mere surjection would not establish the exact count.

Contract. Now p is prime. If the digit Jacobian has rank ρ, the number of residue lifts is zero for nonzero obstruction and $p^{d-\rho}$ for zero obstruction. The obstruction space has $p^{e-\rho}$ elements.

Source: `polynomial_regularity/formalization/PolynomialLiftCounts.lean`.

Lines 104–115; `Komorebi.PolynomialLiftCounts.finite_polynomial_lift_count`.

```
theorem finite_polynomial_lift_count (p L : ℕ) [Fact (Nat.Prime p)] (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (p : ℤ)^L | evalSystem F x i) :
    ∃ r : Fin e → ℤ,
      evalSystem F x = (p : ℤ)^L • r ∧
      Nat.card (ActualLifts p L F x) =
        (if obstruction p (jacobian F x) r = 0 then
          p ^ (d - FiniteFiberCounts.rank p (digitJacobian p F x)) else 0) ∧
      Nat.card (Obstruction p (jacobian F x)) =
        p ^ (e - FiniteFiberCounts.rank p (digitJacobian p F x)) := by
  obtain ⟨r, hr⟩ := Komorebi.PolynomialObstruction.residual_exists p L F x hx
  exact ⟨r, hr, lift_card p L hL F x r hr, obstruction_card p F x⟩
```

Proof reading. The endpoint reuses the digit–lift equivalence for the lift cardinality and the finite cokernel equivalence for the obstruction cardinality. Rank-nullity yields the two exponents in the input and output spaces respectively. The rank is bounded by both d and e, so natural subtraction expresses the ordinary nonnegative dimension difference in these hypotheses. Rectangular equations can have many lifts even when every obstruction vanishes.

At a surjective digit Jacobian, $\rho = e$ and each step has p^{d-e} residue lifts. When $d = e$, that number is one. This is the source of both smooth existence and square-case uniqueness. When $d > e$, existence is still valid but a claim of unique solved coordinates would discard the nontrivial kernel.

8 Compatible integer towers and p-adic endpoints

8.1 The tower is constructed, not supplied

Contract. $q > 0$, $L \geq 1$, F has integer coefficients, the starting point is a root modulo q^L, and its digit Jacobian over `ZMod q` is surjective. The theorem constructs a whole sequence z_n of integer points.

Source: `polynomial_regularity/formalization/PolynomialIntegerTower.lean`.

Lines 80–96; `Komorebi.PolynomialIntegerTower.exists_compatible_tower_strong`.

```
theorem exists_compatible_tower_strong (q L : ℕ) (hq : 0 < q) (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (q : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (digitJacobian q F x)) :
    ∃ z : ℕ → Fin d → ℤ,
      z 0 = x ∧
      (∀ n i, (q : ℤ)^(L+n) | evalSystem F (z n) i) ∧
      (∀ n i, (q : ℤ)^(L+n) | (z (n+1) - z n) i) ∧
      (∀ m n, m ≤ n → ∀ i, (q : ℤ)^(L+m) | (z n - z m) i) ∧
      (∀ n, digitJacobian q F (z n) = digitJacobian q F x) ∧
      (∀ n, Function.Surjective (digitJacobian q F (z n))) := by
  obtain ⟨z, hz0, hroot, hstep, hsurj⟩ := exists_compatible_tower q L hq hL F x hx hJ
  refine ⟨z, hz0, hroot, hstep, ?_, ?_, hsurj⟩
  · exact fun m n hmn => tower_pairwise_congruent q L z hstep m n hmn
  · intro n
    rw [← hz0]
    exact tower_digitJacobian_eq q L hL F z hstep n
```

The theorem returns six pieces of information together. It fixes $z_0 = x$. $F(z_n)$ vanishes modulo q^{L+n}. Successive points differ by multiples of q^{L+n}. Every later point agrees with an earlier point to that earlier depth. The digit Jacobian stays equal to the starting one. Surjectivity holds at every level. These statements keep existence, compatibility and smoothness separate.

Proof reading. The basic tower theorem chooses a correcting digit at each step using digit-map surjectivity. The stronger endpoint adds pairwise congruence by summing compatible increments and identifies every digit Jacobian with the initial map. Polynomial partial derivatives have integer coefficients, so same residue modulo q gives the same reduced derivative. Since $L \geq 1$, all constructed points stay in that initial residue class.

The construction is allowed to use classical choice. It proves an actual infinite sequence in the mathematical model, but it is not presented as extracted executable code. A finite loop that finds digits for a few depths is useful bounded evidence. It does not replace the compatible-tower theorem.

8.2 Completeness changes the coefficient domain

For a prime p, cast z_n coordinatewise into $\mathbb{Z}_p$. Pairwise congruence at depths tending to infinity makes the sequence Cauchy in each coordinate. Finite products of complete spaces are complete, so the vector sequence converges to a p-adic point b.

Contract. p is prime, $L \geq 1$, and the same initial root and surjectivity hypotheses hold. The theorem returns the integer tower, its p-adic limit, an exact polynomial root, and the retained congruence of the limit to every tower stage.

Source: polynomial_regularity/formalization/PolynomialPadicLift.lean.

Lines 57–77; Komorebi.PolynomialPadicLift.exists_padic_root_with_tower.

```
theorem exists_padic_root_with_tower (p L : ℕ) [Fact (Nat.Prime p)] (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (p : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (digitJacobian p F x)) :
    ∃ z : ℕ → Fin d → ℤ, ∃ a : Fin d → ℤ_[p],
      z 0 = x ∧
      (∀ n i, (p : ℤ)^(L+n) | evalSystem F (z n) i) ∧
      (∀ n i, (p : ℤ)^(L+n) | (z (n+1) - z n) i) ∧
      Tendsto (fun n i => (z n i : ℤ_[p])) atTop (𝓝 a) ∧
      (∀ n i, (p : ℤ_[p])^(L+n) | a i - (z n i : ℤ_[p])) ∧
      (∀ i, padicEval p F a i = 0) ∧
      (∀ n, digitJacobian p F (z n) = digitJacobian p F x) ∧
      (∀ n, Nat.card (PolynomialLiftCounts.ActualLifts p (L+n) F (z n)) = p^(d-e)) := by
  obtain ⟨z, hz0, hroot, hstep, _, hcount⟩ := exists_regular_tower p L hL F x hx hJ
  obtain ⟨a, ha, hprecision⟩ :=
    PadicTowerLimit.compatible_vector_tower_limit p L z hstep
  refine ⟨z, a, hz0, hroot, hstep, ha, hprecision,
    polynomial_tower_limit_root p L F z a hroot ha, ?_, hcount⟩
  intro n
  rw [← hz0]
  exact tower_digitJacobian_eq p L hL F z hstep n
```

Proof reading. The source combines the integer tower theorem with the general p-adic tower-limit theorem. Polynomial evaluation is continuous, so $F(z_n)$ converges to $F(b)$. Its divisibility by arbitrarily large powers of p forces its limit to zero. A fixed-depth congruence ball is closed, yielding the stronger precision statement $p^{L+n} \mid b - z_n$ for each fixed n.

This proof separates two limiting operations. Compatibility controls the input limit. Increasing residual depth controls the output limit. Neither alone suffices: a convergent sequence of approximate points whose residual depth does not increase need not converge to an exact root.

8.3 Uniqueness requires a square solved system

Contract. The p-adic unique-root endpoint uses d equations in d unknowns and a surjective digit Jacobian, hence a bijective digit map over $\mathbb{F}_p$. It asserts uniqueness only in the initial depth-L residue ball.

Source: polynomial_regularity/formalization/PolynomialPadicUnique.lean.

Lines 146–156; Komorebi.PolynomialPadicUnique.exists_unique_padic_root.

```
theorem exists_unique_padic_root (p L : ℕ) [Fact (Nat.Prime p)] (hL : 1 ≤ L)
    (F : Fin d → MvPolynomial (Fin d) ℤ) (x : Fin d → ℤ)
    (hx : ∀ i, (p : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (digitJacobian p F x)) :
    ∃! a : Fin d → ℤ_[p],
      (∀ i, MvPolynomial.eval₂ (Int.castRingHom ℤ_[p]) a (F i) = 0) ∧
      (∀ i, (p : ℤ_[p])^L | a i - (x i : ℤ_[p])) := by
  obtain ⟨a, ha, hax⟩ := exists_padic_root p L hL F x hx hJ
```

```
refine ⟨a, ⟨ha, hax⟩, ?_⟩
intro b hb
exact padic_root_unique p L hL F x hJ b a hb.1 ha hb.2 hax
```

Proof reading. Existence supplies one root. The uniqueness theorem compares any second root with that root by finite residues. At each successive depth, the digit equation has exactly one solution, so equal residues at the current depth imply equal residues at the next. Equality at all depths forces equality in $\mathbb{Z}_p$. The excerpt packages that comparison into `∃!`.

The uniqueness proof uses actual finite reductions of arbitrary p-adic roots, not an assumption that every candidate arose from the chosen integer tower. That distinction matters: uniqueness must quantify over every root in the specified ball. For a rectangular surjective system, the kernel has positive dimension and the same conclusion is generally false. The retained rectangular-uniqueness negative control guards this boundary.

8.4 Coefficient stability retains the promised precision

The deformation modules also compare a polynomial system with a coefficientwise congruent perturbation in the smooth, unit-Jacobian setting. Because evaluation preserves coefficient congruence at integral points, the original approximate root is an approximate root of the perturbed system. Its reduced Jacobian is unchanged when the coefficient perturbation has positive depth. The same smooth root theorem gives a nearby root, and the uniqueness comparison fixes the promised residue precision.

This reasoning supplies a bound at the declared perturbation depth. It does not imply an additional free digit. A linear equation with its constant term shifted by exactly p^M has its root shifted by exactly that precision. The retained `InvalidPerturbationExtraPrecision` control is based on that obstruction to strengthening the bound.

9 Local charts and prescribed parameter paths

9.1 The solved block is the correct Jacobian

Split the variables as (y, t), where y has m solved coordinates and t has k prescribed parameters. For e equations the reduced Jacobian has blocks

$$DF = (J\ K), \qquad J : \mathbb{F}_p^m \to \mathbb{F}_p^e, \quad K : \mathbb{F}_p^k \to \mathbb{F}_p^e.$$

A prescribed parameter digit v changes the residual by Kv. The solved digit u must solve

$$Ju = -\bar{r} - Kv.$$

Surjectivity of the full matrix $(J\ K)$ is insufficient when v has already been fixed. It is surjectivity of J that permits every prescribed digit.

Contract. For an integer polynomial system, the source defines the parameter residual $\bar{r} + Kv$, the solved digit map J, and the joined full digit (u, v). The split equation equals the full digit equation exactly.

Source: `polynomial_regularity/formalization/PolynomialPrescribedSystem.lean`.

Lines 77–82; `Komorebi.PolynomialPrescribedSystem.prescribed_equation_iff`.

```
theorem prescribed_equation_iff (q : ℕ) (F : Fin e → MvPolynomial (Fin (m + k)) ℤ)
    (x : Fin (m + k) → ℤ) (r : Fin e → ℤ) (v : Fin k → ZMod q) (u : Fin m → ZMod q) :
    parameterResidual q F x r v + solvedJacobian q F x u = 0 ↔
      reduction q e r + digitJacobian q F x (PolynomialParameterSystem.join u v) = 0 := by
  rw [digitJacobian_join]
  simp only [parameterResidual, add_assoc, add_comm (parameterJacobian q F x v)]
```

Proof reading. The block derivative theorem rewrites the full digit map on `join u v` as the sum of the solved and parameter actions. Associativity and commutativity then identify the residual equations. The parameter term has positive sign inside $\bar{r} + Kv + Ju = 0$, and consequently negative sign when solving for u. Losing this sign changes the actual root motion.

If the solved block has rank ρ, a soluble prescribed-digit fiber has $p^{m-\rho}$ elements. For a surjective J, this becomes p^{m-e} for every prescribed v. A square solved block gives one solved digit for each parameter digit. Rectangular solved blocks leave free solved directions, so prescribing the parameter alone does not generally select one root.

9.2 The continuous chart includes both directions

Contract. The chart endpoint uses e equations, e solved coordinates and q free parameters. The starting integer point is a root modulo p^L, $L \geq 1$, and the solved digit Jacobian is surjective. The homeomorphism is between the parameter residue ball and the exact root locus in the full residue ball.

Source: `polynomial_regularity/formalization/PolynomialParameterChart.lean`.

Lines 167–183; `Komorebi.PolynomialParameterChart.parameterHomeomorph`.

```
def parameterHomeomorph (p L : ℕ) [Fact (Nat.Prime p)] (hL : 1 ≤ L)
    (F : Fin e → MvPolynomial (Fin (e + q)) ℤ) (x : Fin (e + q) → ℤ)
    (hx : ∀ i, (p : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (solvedJacobian p F x)) :
    residueBall p L (freePart x) ≃ₜ {z : residueBall p L x // padicEval p F z.val = 0} where
  toFun t := ⟨chartPoint p L hL F x hx hJ t, chartPoint_root p L hL F x hx hJ t⟩
  invFun z := ⟨freePart z.val.val, fun j => z.val.property (Fin.natAdd e j)⟩
  left_inv t := Subtype.ext (chartPoint_free p L hL F x hx hJ t)
  right_inv z := by
    apply Subtype.ext
    exact (chartPoint_unique p L hL F x hx hJ _ z.val z.property rfl).symm
  continuous_toFun := by
    apply Continuous.subtype_mk
    exact continuous_chartPoint p L hL F x hx hJ
  continuous_invFun := by
    apply Continuous.subtype_mk
    exact continuous_freePart.comp (continuous_subtype_val.comp continuous_subtype_val)
```

Proof reading. The forward map takes a parameter to its uniquely solved root in the ball. The inverse is the free-coordinate projection. The first inverse identity says the constructed root has the requested parameter. The second invokes uniqueness of solved coordinates for any root in the ball. Continuity of the solution map and continuity of the projection establish a homeomorphism, not merely a set bijection.

This chart has explicit integral residue balls and strong congruence control. Its source proof precedes the analytic regularity layer. A homeomorphism of root spaces does not automatically identify an analytic derivative. The later polynomial/analytic bridge supplies that stronger conclusion under fraction-field nonsingularity.

9.3 An arbitrary p-adic parameter is reached by integer digits

The prescribed tower construction can begin with a supplied compatible integer parameter path t_n converging to τ. It constructs solved coordinates at every depth and tracks the actual parameter path, rather than guessing compatible roots independently. The endpoint also builds such a parameter path for every p-adic τ in the initial parameter ball.

Contract. p is prime, $L \geq 1$, J is the surjective solved digit map with m solved coordinates and e equations, and τ lies in the initial parameter residue ball. No solution tower or parameter path is an input to this endpoint.

Source: `polynomial_regularity/formalization/PolynomialPrescribedPadicTower.lean`.

Lines 57–76; `Komorebi.PolynomialPrescribedPadicTower.exists_prescribed_padic_root_with_tower`.

```
theorem exists_prescribed_padic_root_with_tower (p L : ℕ) [Fact (Nat.Prime p)]
    (hL : 1 ≤ L) (F : Fin e → MvPolynomial (Fin (m+k)) ℤ)
    (x : Fin (m+k) → ℤ) (hx : ∀ i, (p : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (solvedJacobian p F x))
    (τ : Fin k → ℤ_[p])
```

```
    (hτ : ∀ j, (p : ℤ_[p])^L | τ j - ((freePart x j : ℤ) : ℤ_[p])) :
    ∃ (z : ℕ → Fin (m+k) → ℤ) (b : Fin (m+k) → ℤ_[p]),
      z 0 = x ∧
      (∀ n i, (p : ℤ)^(L+n) | evalSystem F (z n) i) ∧
      (∀ n i, (p : ℤ)^(L+n) | (z (n+1) - z n) i) ∧
      (∀ n, solvedJacobian p F (z n) = solvedJacobian p F x) ∧
      Tendsto (fun n i => (z n i : ℤ_[p])) atTop (𝓝 b) ∧
      (∀ i, padicEval p F b i = 0) ∧ freePart b = τ ∧
      (∀ n i, (p : ℤ_[p])^(L+n) | b i - (z n i : ℤ_[p])) ∧
      (∀ n v, Nat.card (PrescribedLifts p (L+n) F (z n) v) = p^(m-e)) := by
  obtain ⟨t, ht0, ht, _, htlim⟩ :=
    PolynomialParameterPath.exists_parameter_path p L (freePart x) τ hτ
  obtain ⟨z, b, hz0, _, hroot, hstep, hfixed, hlim, hb, hbτ, hprecision, hcard⟩ :=
    exists_root_with_prescribed_path p L hL F x hx hJ t ht0 ht τ htlim
  exact ⟨z, b, hz0, hroot, hstep, hfixed, hlim, hb, hbτ, hprecision, hcard⟩
```

Proof reading. The parameter-path theorem supplies integer approximations with the correct initial value and increment divisibility. The prescribed integer tower fills in solved digits using J-surjectivity. A p-adic tower limit then gives the exact root with free part τ. The returned finite lift count p^{m-e} holds at every stage and for every next parameter digit, making the rectangular freedom explicit.

Contract. For $m = e$, the prescribed root in the initial ball is unique. The full predicate includes its exact parameter τ as well as its root equation and residue ball.

Source: `polynomial_regularity/formalization/PolynomialPrescribedPadicTower.lean`.

Lines 96–109; `Komorebi.PolynomialPrescribedPadicTower.exists_unique_prescribed_padic_root`.

```
theorem exists_unique_prescribed_padic_root (p L : ℕ) [Fact (Nat.Prime p)]
    (hL : 1 ≤ L) (F : Fin e → MvPolynomial (Fin (e+k)) ℤ)
    (x : Fin (e+k) → ℤ) (hx : ∀ i, (p : ℤ)^L | evalSystem F x i)
    (hJ : Function.Surjective (solvedJacobian p F x))
    (τ : Fin k → ℤ_[p])
    (hτ : ∀ j, (p : ℤ_[p])^L | τ j - ((freePart x j : ℤ) : ℤ_[p])) :
    ∃! b : Fin (e+k) → ℤ_[p],
      (∀ i, padicEval p F b i = 0) ∧ freePart b = τ ∧
      (∀ i, (p : ℤ_[p])^L | b i - (x i : ℤ_[p])) := by
  obtain ⟨b, hb, hbτ, hbx⟩ := exists_prescribed_padic_root p L hL F x hx hJ τ hτ
  refine ⟨b, ⟨hb, hbτ, hbx⟩, ?_⟩
  intro c hc
  exact PolynomialPrescribedTowerUnique.prescribed_root_unique p L hL F x hx hJ
    c b hc.1 hb hc.2.2 hbx (hc.2.1.trans hbτ.symm)
```

Proof reading. Existence and the prescribed-root comparison theorem are combined. Two candidates with the same parameter have zero parameter difference, so their next solved digit difference lies in the kernel of the bijective solved block. It vanishes at each depth. The theorem therefore controls every candidate root with that parameter, independent of the tower choices used to obtain the first one.

If $k = 0$, the parameter space has one vector and the theorem reduces to the square root theorem. If there are zero equations, the algebraic finite-product statements remain meaningful in dimension zero, subject to their stated typeclass and surjectivity hypotheses. The source does not impose a positive dimension merely to make a matrix display look conventional.

9.4 A full-rank system admits a suitable coordinate split

For a general e-equation system with surjective full reduced Jacobian, the minor-selection module chooses e independent columns. The coordinate-reordering module turns that choice into a permutation separating solved and free coordinates. It proves evaluation and derivative blocks transform according to that actual permutation. The general-chart theorem then applies the square solved-block chart in those coordinates and transports the result back.

This is existence of a good coordinate split. It does not assert that every externally prescribed split has a surjective solved block. If a parameter policy has already fixed particular coordinates, its prescribed-lifting theorem must still check that policy's own solved block. The finite dimensions and explicit reorder equivalence preserve the number of free parameters, $d - e$.

10 Weighted inverse bounds and quantitative root motion

10.1 A fraction-field bound has an integral certificate

Let J be the actual square Jacobian at an integral p-adic center. It may be singular modulo p while remaining invertible over $\mathbb{Q}_p$. The relevant size condition is that $p^s J^{-1}$ is an integral matrix. This condition records the chosen exponent s directly. The proof does not replace it by a possibly larger determinant valuation.

Contract. p is prime and s is natural. Fraction-field invertibility together with an integral matrix representing $p^s J^{-1}$ is equivalent to an integral two-sided certificate $AJ = JA = p^s I$.

Source: `polynomial_regularity/formalization/PolynomialWeightedInverse.lean`.

Lines 142–155; `Komorebi.PolynomialWeightedInverse.bounded_inverse_iff_certificate`.

```
theorem bounded_inverse_iff_certificate (p s : ℕ) [Fact (Nat.Prime p)]
    (J : Matrix (Fin d) (Fin d) ℤ_[p]) :
    (IsUnit (rationalMatrix p J) ∧
      ∃ A : Matrix (Fin d) (Fin d) ℤ_[p],
        rationalMatrix p A = (p : ℚ_[p])^s • (rationalMatrix p J)⁻¹) ↔
    (∃ A : Matrix (Fin d) (Fin d) ℤ_[p],
      A * J = (p : ℤ_[p])^s • (1 : Matrix (Fin d) (Fin d) ℤ_[p]) ∧
      J * A = (p : ℤ_[p])^s • (1 : Matrix (Fin d) (Fin d) ℤ_[p])) := by
  constructor
  · rintro ⟨hJ, hA⟩
    exact (integral_inverse_iff_certificate p s J hJ).mp hA
  · rintro ⟨A, hAJ, hJA⟩
    have hJ := rationalMatrix_isUnit_of_certificate p s J A hAJ
    exact ⟨hJ, (integral_inverse_iff_certificate p s J hJ).mpr ⟨A, hAJ, hJA⟩⟩
```

Proof reading. The embedding of integral matrices into rational p-adic matrices preserves multiplication and scalar action. An integral representation of the inverse therefore gives the two identities after injectivity of the embedding. Conversely, the two-sided identities establish invertibility in the fraction field because $p^s \neq 0$. Both products matter: they support normalization and root equivalence in different directions. A single entrywise valuation statement without an actual matrix inverse is not this certificate.

10.2 Polynomial maps become contractions after scaling

Integral polynomial evaluation is Lipschitz for the p-adic sup metric. Multiplying it by p yields a strict contraction. The contraction map has the form $u \mapsto c + pH(u)$, so the fixed point theorem acts on the complete space of integral p-adic vectors.

Contract. H is an arbitrary vector of actual multivariate polynomials with $\mathbb{Z}_p$ coefficients and $c \in \mathbb{Z}_p^d$. The source establishes a unique fixed point of its specified contraction map.

Source: `polynomial_regularity/formalization/PolynomialPadicContraction.lean`.

Lines 101–105; `Komorebi.PolynomialPadicContraction.exists_unique_fixedPoint`.

```
theorem exists_unique_fixedPoint (p : ℕ) [Fact p.Prime]
    (H : Fin d → MvPolynomial (Fin d) ℤ_[p]) (c : Fin d → ℤ_[p]) :
```

```
    ∃! u : Fin d → ℤ_[p], contractionMap p H c u = u := by
  let hf := contracting_contractionMap p H c
  exact ⟨hf.fixedPoint _, hf.fixedPoint_isFixedPt, fun u hu => hf.fixedPoint_unique hu⟩
```

Proof reading. The source first proves the Lipschitz estimate for polynomial evaluation and the contraction constant after multiplication by p. `contracting_contractionMap` packages that estimate into the complete-space fixed point machinery. The final three fields are the fixed point, its fixed-point equation, and uniqueness against every other fixed point. The source's separate divisible-fixed-point theorem controls precision when c has a specified p-power factor and $H(0) = 0$.

The condition $H(0) = 0$ is required for that additional precision argument for divisible fixed points. It is not a requirement of the bare fixed-point existence theorem displayed above. Keeping these two contracts separate prevents an accidental strengthening.

10.3 The threshold and the uniqueness ball are explicit

For $F(a)$ divisible by p^r and a certificate exponent s, the source imposes $r > 2s$. Write $b = a + p^{s+1}u$. Actual polynomial Taylor expansion and preconditioning put the equation into the normalized form

$$p^{2s+1}(C + u + pQ(u)) = 0, \qquad Q(0) = 0, \qquad p^{r-(2s+1)} \mid C.$$

The contraction solves for u, and its precision transfers back to b.

Contract. F is square with integral p-adic coefficients; $J_F(a)$ is invertible over $\mathbb{Q}_p$; $p^s J_F(a)^{-1}$ is integral; $F(a)$ is divisible by p^r; and $r > 2s$. The root is unique in the entire depth-$s+1$ ball and is congruent to a to depth $r - s$.

Source: `polynomial_regularity/formalization/PolynomialWeightedLift.lean`.

Lines 112–125; `Komorebi.PolynomialWeightedLift.exists_weighted_root`.

```
theorem exists_weighted_root (p s r : ℕ) [Fact p.Prime]
    (hr : 2 * s < r) (F : Fin d → MvPolynomial (Fin d) ℤ_[p])
    (a : Fin d → ℤ_[p]) (hJ : IsUnit (rationalMatrix p (jacobian F a)))
    (hbound : ∃ A : Matrix (Fin d) (Fin d) ℤ_[p],
      rationalMatrix p A = (p : ℚ_[p]) ^ s • (rationalMatrix p (jacobian F a))⁻¹)
    (hF : ∀ i, (p : ℤ_[p]) ^ r ∣ eval a (F i)) :
    ∃ b : Fin d → ℤ_[p],
      (∀ i, eval b (F i) = 0) ∧
      (∀ i, (p : ℤ_[p]) ^ (r - s) ∣ b i - a i) ∧
      (∀ i, (p : ℤ_[p]) ^ (s + 1) ∣ b i - a i) ∧
      (∀ z : Fin d → ℤ_[p], (∀ i, eval z (F i) = 0) →
        (∀ i, (p : ℤ_[p]) ^ (s + 1) ∣ z i - a i) → z = b) := by
  obtain ⟨A, hAJ, hJA⟩ := (bounded_inverse_iff_certificate p s (jacobian F a)).mp ⟨hJ, hbound⟩
  exact exists_weighted_root_certificate p s r hr F a A hAJ hJA hF
```

Proof reading. The source converts the fraction-field bound to the integral two-sided certificate. Its normalization theorem builds C, Q from the actual polynomials, so no Taylor expansion or contraction hypothesis is supplied by the caller. The normalized-root theorem obtains the fixed point and transfers its root equation, precision, and uniqueness back to the original coordinates. The root-equivalence step uses injectivity of the preconditioner in the fraction field.

At the least permitted depth $r = 2s + 1$, the precision $r - s = s + 1$ is exactly the uniqueness-ball depth. Larger r improves the retained root precision. The boundary $r = 2s$ is excluded. The retained weighted-boundary counterexample supplies an integral polynomial with that residual depth and no integral root. A linear example also shows the loss of s in the displacement is sharp. Thus neither replacing $>$ by $\geq$ nor adding a free precision digit follows from this theorem.

10.4 Coefficient perturbation has a first-order formula

Suppose $F(a) = 0$ and perturb the actual coefficient system to $F + p^M H$, with $M > 2s$. The same original inverse certificate supports the nearby root. The derivative of the perturbation is handled in the proof, rather than assumed to have exactly the old inverse.

Contract. The perturbed root is unique in the depth-$s + 1$ ball, has displacement divisible by p^{M-s}, and its rational p-adic first-order term is $-p^M J_F(a)^{-1} H(a)$. The integral error has depth $2M - 3s$.

Source: `polynomial_regularity/formalization/PolynomialRootMotion.lean`.

Lines 142–162; `Komorebi.PolynomialRootMotion.exists_unique_root_motion`.

```
theorem exists_unique_root_motion (p s M : ℕ) [Fact p.Prime]
    (hM : 2 * s < M) (F H : Fin d → MvPolynomial (Fin d) ℤ_[p])
    (a : Fin d → ℤ_[p]) (hFa : ∀ i, eval a (F i) = 0)
    (hJ : IsUnit (rationalMatrix p (jacobian F a)))
    (hbound : ∃ A : Matrix (Fin d) (Fin d) ℤ_[p],
      rationalMatrix p A = (p : ℚ_[p]) ^ s • (rationalMatrix p (jacobian F a))⁻¹) :
    (∃! b : Fin d → ℤ_[p], (∀ i, eval b (perturbSystem p M F H i) = 0) ∧
      (∀ i, (p : ℤ_[p]) ^ (s + 1) | b i - a i)) ∧
    (∀ b : Fin d → ℤ_[p], (∀ i, eval b (perturbSystem p M F H i) = 0) →
      (∀ i, (p : ℤ_[p]) ^ (s + 1) | b i - a i) →
      (∀ i, (p : ℤ_[p]) ^ (M - s) | b i - a i) ∧
      (∃ ε : Fin d → ℤ_[p], (∀ i, (p : ℤ_[p]) ^ (2 * M - 3 * s) | ε i) ∧
        ∀ i, (b i : ℚ_[p]) - (a i : ℚ_[p]) =
          -(p : ℚ_[p]) ^ M * ((rationalMatrix p (jacobian F a))⁻¹).mulVec
            (fun j => (eval a (H j) : ℚ_[p])) i + (ε i : ℚ_[p]))) := by
  obtain ⟨b, hb, hdepth, hball, huniq, herror⟩ := exists_root_motion p s M hM F H a hFa hJ hbound
  constructor
  · exact ⟨b, ⟨hb, hball⟩, fun z hz => huniq z hz.1 hz.2⟩
  · intro z hz hzb
    rw [huniq z hz hzb]
    exact ⟨hdepth, herror⟩
```

Proof reading. The existence theorem constructs a root using the original preconditioner. Its error estimate combines the quadratic Taylor remainder with the perturbation's change under the root displacement, then applies the inverse bound, costing the stated s digits. The unique-existence endpoint transports both the depth and the error formula to every root in the uniqueness ball. It does not assign a fixed linear map to a nonlinear finite displacement. It estimates that displacement by a linear leading term plus a rigorously bounded remainder.

10.5 Parameter and coefficient motion share one forcing term

Now $F(y,t)$ has a square solved block J and parameter block K. Perturb the coefficients by $p^M H$ and prescribe $t' = t + p^M v$. The solved-coordinate leading displacement is

$$y' - y = -p^M J^{-1}(H(a) + Kv) + \varepsilon, \qquad p^{2M-3s} \mid \varepsilon.$$

The extra Kv term is required even when $H = 0$. The source uses the actual evaluated partial derivatives in both blocks.

Contract. The theorem requires the same inverse bound on the solved block, $M > 2s$, an exact initial root, and the stated parameter displacement. Uniqueness is among roots with that exact new parameter in the depth-$s+1$ ball.

Source: `polynomial_regularity/formalization/PolynomialJointMotion.lean`.

Lines 76–105; `Komorebi.PolynomialJointMotion.exists_joint_root_motion`.

```
theorem exists_joint_root_motion (p s M : ℕ) [Fact p.Prime]
    (hM : 2 * s < M) (F H : Fin e → MvPolynomial (Fin (e + k)) ℤ_[p])
    (a : Fin (e + k) → ℤ_[p]) (v : Fin k → ℤ_[p])
    (hFa : ∀ i, eval a (F i) = 0)
    (hJ : IsUnit (rationalMatrix p (solvedJacobian F a)))
    (hbound : ∃ A : Matrix (Fin e) (Fin e) ℤ_[p],
      rationalMatrix p A = (p : ℚ_[p]) ^ s • (rationalMatrix p (solvedJacobian F a))⁻¹) :
    ∃ b : Fin (e + k) → ℤ_[p],
      (∀ i, eval b (F i) + (p : ℤ_[p]) ^ M * eval b (H i) = 0) ∧
      freePart b = freePart a + (p : ℤ_[p]) ^ M • v ∧
      (∀ i, (p : ℤ_[p]) ^ (M - s) | b i - a i) ∧
      (∀ i, (p : ℤ_[p]) ^ (s + 1) | b i - a i) ∧
      (∀ z : Fin (e + k) → ℤ_[p],
        (∀ i, eval z (F i) + (p : ℤ_[p]) ^ M * eval z (H i) = 0) →
        freePart z = freePart a + (p : ℤ_[p]) ^ M • v →
        (∀ i, (p : ℤ_[p]) ^ (s + 1) | z i - a i) → z = b) ∧
      (∃ ε : Fin e → ℤ_[p], (∀ i, (p : ℤ_[p]) ^ (2 * M - 3 * s) | ε i) ∧
        ∀ i, ((solvedPart b i : ℤ_[p]) : ℚ_[p]) - ((solvedPart a i : ℤ_[p]) : ℚ_[p]) =
          -(p : ℚ_[p]) ^ M * ((rationalMatrix p (solvedJacobian F a))⁻¹).mulVec
            (fun j => ((eval a (H j) + (parameterJacobian F a).mulVec v j : ℤ_[p]) : ℚ_[p])) i +
          (ε i : ℚ_[p])) := by
  obtain ⟨A, hA⟩ := hbound
  obtain ⟨hAJ, hJA⟩ := certificate_of_integral_inverse p s (solvedJacobian F a) A hJ hA
  obtain ⟨b, hb, ht, hdepth, hball, huniq, herror⟩ :=
    exists_joint_root_certificate p s M hM F H a v A hAJ hJA hFa
  obtain ⟨ε, hε, hformula⟩ := PolynomialMotionRational.exists_rational_error
    p s M (2 * M - 3 * s) (by omega) (solvedJacobian F a) A hA
    (solvedPart a) (solvedPart b)
    ((fun j => eval a (H j)) + (parameterJacobian F a).mulVec v) herror
  exact ⟨b, hb, ht, hdepth, hball, huniq, ε, hε, hformula⟩
```

Proof reading. The source converts the solved inverse bound into a certificate, constructs the augmented root, and obtains an integral error equation. `exists_rational_error` then divides the certificate relation in the fraction field and identifies the first-order term. The coordinate split ensures the final error concerns solved coordinates. Free coordinates already have their exact prescribed displacement.

The retained controls reject omitting the parameter forcing term, reversing its sign, and demanding an extra uniform error digit. They use explicit boundary systems, so those exclusions are mathematical constraints rather than stylistic choices. The resulting theorem is local quantitative deformation, not a global branch-selection theorem.

11 Actual derivatives and analytic local graphs

11.1 The polynomial Jacobian becomes a strict derivative

The earlier integer Taylor lemma proves divisibility. Analytic regularity needs a different bridge: actual polynomial evaluation must have the strict Fréchet derivative represented by the evaluated formal Jacobian. The source proves the scalar derivative by polynomial induction and assembles finite coordinates.

Contract. $\mathbb{K}$ is a complete nontrivially normed field. F is a vector of actual multivariate polynomials over $\mathbb{K}$. The strict derivative of evaluation at a is the continuous linear map represented by the actual Jacobian.

Source: `polynomial_regularity/formalization/PolynomialAnalyticEvaluation.lean`.

Lines 76–85; `Komorebi.PolynomialAnalyticEvaluation.hasStrictFDerivAt_evalSystem`.

```
theorem hasStrictFDerivAt_evalSystem (F : Fin m → MvPolynomial (Fin n) 𝕜)
    (a : Fin n → 𝕜) :
    HasStrictFDerivAt (evalSystem F) (matrixCLM (PolynomialWeightedInverse.jacobian F a)) a := by
  have h := hasStrictFDerivAt_pi.mpr (fun i => hasStrictFDerivAt_eval (F i) a)
  have hder : ContinuousLinearMap.pi (fun i => polynomialDeriv (F i) a) =
      matrixCLM (PolynomialWeightedInverse.jacobian F a) := by
    ext v i
    simp [PolynomialWeightedInverse.jacobian, Matrix.mulVec, dotProduct]
  rw [← hder]
  convert! h using 1
```

Proof reading. Each scalar polynomial has its proved derivative. `hasStrictFDerivAt_pi` assembles those coordinate derivatives. Extensionality of continuous linear maps compares their action on every vector and coordinate. Unfolding the matrix action yields the same finite dot product as the formal partial derivatives. The final conversion identifies the evaluated system with this coordinate assembly.

Analyticity of actual polynomial evaluation is supplied through the pinned Mathlib multivariate-polynomial theorem and finite coordinate assembly. It is not a hypothesis about a proposed solution map. The development also proves the derivative block identities for the actual augmented polynomial system. A user cannot supply an unrelated matrix and call it the Jacobian.

11.2 The strict local inverse is the analytic inverse

Contract. In the inverse helper, $\mathbb{K}$ is nontrivially normed; E, F are normed spaces over it; E is complete; and $i : E \simeq_L F$ is a continuous linear equivalence. The function is analytic at a and has strict derivative i there.

Source: `polynomial_regularity/formalization/PolynomialAnalyticInverse.lean`.

Lines 28–32; `Komorebi.PolynomialAnalyticInverse.analyticAt_inverse`.

```
theorem analyticAt_inverse (ha : AnalyticAt 𝕜 f a)
    (hd : HasStrictFDerivAt f (i : E →L[𝕜] F) a) :
    AnalyticAt 𝕜 (hd.localInverse f i a) (f a) := by
  exact (hd.toOpenPartialHomeomorph f).analyticAt_symm'
    hd.mem_toOpenPartialHomeomorph_source ha hd.hasFDerivAt.fderiv
```

Proof reading. The strict derivative's local inverse is the inverse of its associated open partial homeomorphism. The Mathlib analytic inverse theorem applies to that same inverse, using the derivative identity. The proof does not construct one analytic inverse and separately assume it agrees with the root-solving inverse. The two are linked by the explicit partial homeomorphism.

These helper hypotheses are slightly different from the polynomial endpoint's ambient hypotheses. In particular, the helper only declares completeness of its source space E, while the endpoint gets it from the complete field and finite-dimensional coordinates. Reading the full variable context prevents a short excerpt from concealing these assumptions.

11.3 The augmented derivative exposes the sign

For the map $\Phi(y,t) = (F(y,t),t)$, the actual derivative at a and its inverse are

$$D\Phi(a) = \begin{pmatrix} J & K \\ 0 & I \end{pmatrix}, \qquad D\Phi(a)^{-1} = \begin{pmatrix} J^{-1} & -J^{-1}K \\ 0 & I \end{pmatrix}.$$

The source constructs the continuous linear equivalence from these matrices and verifies both products. Inserting a pure parameter displacement $(0,v)$ into the inverse yields solved displacement $-J^{-1}Kv$.

Contract. J is a square invertible matrix over the normed field and K has the specified rectangular shape. The explicit augmented equivalence's inverse, restricted to parameter input and solved output, equals the continuous linear map of $-J^{-1}K$.

Source: `polynomial_regularity/formalization/PolynomialAnalyticBlocks.lean`.

Lines 59–66; `Komorebi.PolynomialAnalyticBlocks.inverse_solved_zero`.

```
theorem inverse_solved_zero (J : Matrix (Fin e) (Fin e) 𝕜)
    (K : Matrix (Fin e) (Fin k) 𝕜) (hJ : IsUnit J) (v : Fin k → 𝕜) :
    solvedPart ((upperEquiv J K hJ).symm (join (0 : Fin e → 𝕜) v)) =
      matrixCLM (-(J⁻¹ * K)) v := by
  rw [upperEquiv_symm_apply]
  simp only [blockCertificate, upperBlock_mulVec_join, Matrix.mulVec_zero,
    zero_add, solvedPart_join]
  rfl
```

Proof reading. The proof unfolds the actual inverse equivalence and its certificate, evaluates the upper block on the joined vector, and removes the zero solved input. The negative sign is therefore an algebraic consequence of the inverse block identities. The dedicated derivative-sign negative control checks a concrete contrary statement.

11.4 The graph is exact in a product neighborhood

Contract. $\mathbb{K}$ is a complete nontrivially normed field; $F \in \mathbb{K}[Y,T]^e$ is an actual polynomial system with e solved and k parameter coordinates; $F(a) = 0$; and the solved Jacobian J is invertible. The theorem constructs open neighborhoods U, V, an analytic solution ψ, and exact equivalence $F(y,t) = 0 \iff y = \psi(t)$ for $t \in U, y \in V$.

Source: `polynomial_regularity/formalization/PolynomialAnalyticImplicit.lean`.

Lines 151–196; `Komorebi.PolynomialAnalyticImplicit.exists_analytic_implicit`.

```
theorem exists_analytic_implicit
    (F : Fin e → MvPolynomial (Fin (e + k)) 𝕜)
    (a : Fin (e + k) → 𝕜) (hF : evalSystem F a = 0)
    (hJ : IsUnit (Komorebi.PolynomialJointSystem.solvedJacobian F a)) :
    ∃ (ψ : (Fin k → 𝕜) → (Fin e → 𝕜)) (U : Set (Fin k → 𝕜))
      (V : Set (Fin e → 𝕜)),
      IsOpen U ∧ IsOpen V ∧ freePart a ∈ U ∧ solvedPart a ∈ V ∧
      ψ (freePart a) = solvedPart a ∧ AnalyticOnNhd 𝕜 ψ U ∧
      (∀ t ∈ U, ψ t ∈ V) ∧
      (∀ t ∈ U, ∀ y ∈ V, evalSystem F (join y t) = 0 ↔ y = ψ t) ∧
      HasStrictFDerivAt ψ
        (matrixCLM (-((Komorebi.PolynomialJointSystem.solvedJacobian F a)⁻¹ *
          Komorebi.PolynomialJointSystem.parameterJacobian F a))) (freePart a) := by
  let J := Komorebi.PolynomialJointSystem.solvedJacobian F a
  let K := Komorebi.PolynomialJointSystem.parameterJacobian F a
  let A := upperEquiv J K hJ
  have hshape : evalSystem (Komorebi.PolynomialJointSystem.augmented F 0) =
      fun z => join (evalSystem F z) (freePart z) := by
    funext z
    change (fun i => MvPolynomial.eval z
      (Komorebi.PolynomialJointSystem.augmented F 0 i)) =
        join (fun i => MvPolynomial.eval z (F i)) (freePart z)
    simpa only [sub_zero] using
      Komorebi.PolynomialJointSystem.eval_augmented F (0 : Fin k → 𝕜) z
  have ha := analyticAt_evalSystem (Komorebi.PolynomialJointSystem.augmented F 0) a
  have hd : HasStrictFDerivAt
      (evalSystem (Komorebi.PolynomialJointSystem.augmented F 0))
      (A : (Fin (e + k) → 𝕜) →L[𝕜] (Fin (e + k) → 𝕜)) a := by
    rw [show (A : (Fin (e + k) → 𝕜) →L[𝕜] (Fin (e + k) → 𝕜)) =
      matrixCLM (Komorebi.PolynomialBlockInverse.upperBlock J K 1) from
      upperEquiv_toCLM J K hJ]
    simpa only [Komorebi.PolynomialJointSystem.jacobian_augmented] using
      hasStrictFDerivAt_evalSystem (Komorebi.PolynomialJointSystem.augmented F 0) a
  rw [hshape] at ha hd
  obtain ⟨ψ, U, V, hU, hV, ht, hy, hψa, hψan, hψV, hgraph, hder⟩ :=
    exists_analytic_graph (evalSystem F) a hF A ha hd
  refine ⟨ψ, U, V, hU, hV, ht, hy, hψa, hψan, hψV, hgraph, ?_⟩
  have heq : (solvedPartCLM (𝕜 := 𝕜) (e := e) (k := k)).comp
      ((A.symm : (Fin (e + k) → 𝕜) →L[𝕜] (Fin (e + k) → 𝕜)).comp
        (parameterInjectionCLM (e := e) (k := k))) = matrixCLM (-(J⁻¹ * K)) := by
    apply ContinuousLinearMap.ext
    intro v
    simp only [ContinuousLinearMap.comp_apply, solvedPartCLM_apply,
      parameterInjectionCLM_apply]
    exact inverse_solved_zero J K hJ v
  simpa only [heq] using hder
```

Proof reading. The proof names the actual solved and parameter blocks J, K, constructs their upper-block continuous equivalence, and identifies the augmented polynomial map with $z \mapsto (F(z), \pi_T z)$.

Actual polynomial analyticity and the strict derivative bridge provide the inputs to the analytic graph helper. The helper takes a local inverse g and defines $\psi(t) = \pi_Y g(0,t)$. The final linear-map identity restricts g's inverse derivative to obtain $-J^{-1}K$.

The graph helper shrinks the parameter neighborhood so that four requirements hold together: the inverse identities, analyticity, the requested product neighborhood, and $\psi(U) \subseteq V$. This shrinking is part of the proof. It is not enough to know an inverse exists at the base point and then claim every point in an arbitrary ball lies in its domain.

The function ψ is defined on the ambient parameter space, but the root and regularity conclusions are restricted to U. Uniqueness is among $y \in V$. Analyticity on the neighborhood implies differentiability there. The displayed strict derivative formula is at the specified base point. The open-set selection theorem later obtains the analogous derivative at every point by applying the local theorem repeatedly.

11.5 A supplied continuous root selection acquires regularity

Contract. Keep the same field and solved-block nonsingularity. A supplied map g is continuous at the base parameter, has the correct base value, and solves the exact polynomial equation on a neighborhood. No derivative or analyticity of g is assumed.

Source: `polynomial_regularity/formalization/PolynomialAnalyticSelection.lean`.

Lines 39–56; `Komorebi.PolynomialAnalyticSelection.continuous_selection_regularity`.

```
theorem continuous_selection_regularity
    (F : Fin e → MvPolynomial (Fin (e + k)) 𝕜)
    (a : Fin (e + k) → 𝕜)
    (hJ : IsUnit (PolynomialJointSystem.solvedJacobian F a))
    (g : (Fin k → 𝕜) → (Fin e → 𝕜))
    (hg : ContinuousAt g (freePart a)) (hg₀ : g (freePart a) = solvedPart a)
    (hroot : ∀ᶠ t in 𝒩 (freePart a), evalSystem F (join (g t) t) = 0) :
    AnalyticAt 𝕜 g (freePart a) ∧
      HasStrictFDerivAt g
        (matrixCLM (-((PolynomialJointSystem.solvedJacobian F a)⁻¹ *
          PolynomialJointSystem.parameterJacobian F a))) (freePart a) := by
  have hF : evalSystem F a = 0 := by
    simpa only [hg₀, join_parts] using hroot.self_of_nhds
  obtain ⟨ψ, U, V, hU, hV, ht, hy, _hψ₀, hψ, _hψV, hgraph, hdψ⟩ :=
    exists_analytic_implicit F a hF hJ
  have heq := continuous_selection_eventually_eq (evalSystem F) g ψ
    (freePart a) (solvedPart a) U V hU hV ht hy hgraph hg hg₀ hroot
  exact ⟨(hψ (freePart a) ht).congr heq.symm, hdψ.congr_of_eventuallyEq heq.symm⟩
```

Proof reading. The root equation at the base point follows from the eventual neighborhood equation and the given base value. The analytic graph theorem constructs ψ. Continuity places $g(t)$ in V for all sufficiently close t, while openness places t in U. The exact graph equivalence forces $g = \psi$ as germs. Analyticity and the strict derivative transfer through eventual equality.

For a continuous exact root selection on an open set, the source requires an invertible solved Jacobian at every point of that set. It then proves analyticity and the actual implicit derivative at every point. A single nonsingular point does not justify regularity of a branch through later singular points. Arbitrary global branch choices are outside this germ comparison.

11.6 Integral coefficients are evaluated in the fraction field

Contract. For any prime p, the coefficients and base point may lie in $\mathbb{Z}_p$. Only the solved Jacobian after inclusion in $\mathbb{Q}_p$ must be invertible. The resulting neighborhoods and analytic map live in the ambient $\mathbb{Q}_p$ vector spaces.

Source: `polynomial_regularity/formalization/PolynomialRegularityIntegral.lean`.

Lines 87–123; `Komorebi.PolynomialRegularityIntegral.exists_analytic_integral_implicit`.

```
theorem exists_analytic_integral_implicit (p : ℕ) [Fact (Nat.Prime p)]
    (F : Fin e → MvPolynomial (Fin (e + k)) ℤ_[p])
    (a : Fin (e + k) → ℤ_[p]) (hF : ∀ i, eval a (F i) = 0)
    (hJ : IsUnit (rationalMatrix p (Komorebi.PolynomialJointSystem.solvedJacobian F a))) :
    ∃ (ψ : (Fin k → ℚ_[p]) → (Fin e → ℚ_[p])) (U : Set (Fin k → ℚ_[p]))
      (V : Set (Fin e → ℚ_[p])),
      IsOpen U ∧ IsOpen V ∧ rationalPoint p (freePart a) ∈ U ∧
      rationalPoint p (solvedPart a) ∈ V ∧
      ψ (rationalPoint p (freePart a)) = rationalPoint p (solvedPart a) ∧
      AnalyticOnNhd ℚ_[p] ψ U ∧ (∀ t ∈ U, ψ t ∈ V) ∧
      (∀ t ∈ U, ∀ y ∈ V,
        (∀ i, eval₂ (algebraMap ℤ_[p] ℚ_[p]) (join y t) (F i) = 0) ↔ y = ψ t) ∧
      HasStrictFDerivAt ψ
        (matrixCLM (-((rationalMatrix p
          (Komorebi.PolynomialJointSystem.solvedJacobian F a))⁻¹ *
          Matrix.of (fun i j => algebraMap ℤ_[p] ℚ_[p] (parameterJacobian F a i j)))))
        (rationalPoint p (freePart a)) := by
  have hroot : evalSystem (rationalSystem p F) (rationalPoint p a) = 0 := by
    ext i
    exact rationalSystem_root p F a hF i
  have hunit : IsUnit (Komorebi.PolynomialJointSystem.solvedJacobian
      (rationalSystem p F) (rationalPoint p a)) := by
    simpa only [solvedJacobian_rationalSystem] using hJ
  obtain ⟨ψ, U, V, hU, hV, ht, hy, hψa, hψan, hψV, hgraph, hder⟩ :=
    Komorebi.PolynomialAnalyticImplicit.exists_analytic_implicit
      (rationalSystem p F) (rationalPoint p a) hroot hunit
  refine ⟨ψ, U, V, hU, hV, ht, hy, hψa, hψan, hψV, ?_, ?_⟩
  · intro t ht' y hy'
    have h := hgraph t ht' y hy'
    simpa only [evalSystem, rationalSystem, MvPolynomial.eval_map,
      funext_iff, Pi.zero_apply] using h
  · have hK : parameterJacobian (rationalSystem p F) (rationalPoint p a) =
        Matrix.of (fun i j => algebraMap ℤ_[p] ℚ_[p] (parameterJacobian F a i j)) := by
      ext i j
      exact eval_pderiv_rationalSystem p F a i (Fin.natAdd e j)
    rw [solvedJacobian_rationalSystem, hK, rationalPoint_freePart] at hder
    exact hder
```

Proof reading. The source proves that evaluation, formal partial derivatives, and both Jacobian blocks commute with the coefficient inclusion. It applies the actual polynomial implicit theorem to the included system and point, then rewrites its equation and derivative back in terms of the original integral coefficients. Invertibility modulo p is not required. The included solved Jacobian may therefore be singular modulo p while remaining invertible over $\mathbb{Q}_p$.

This is an ambient-field analytic theorem. It does not make $\mathbb{Z}_p$ a vector space over $\mathbb{Q}_p$, assert that every nearby analytic value is integral, or supply a new explicit integral radius. The earlier weighted and residue-ball theorems retain their own quantitative domains. Nor does it prove a branch through a point whose solved Jacobian is singular over the fraction field. General singular persistence and global compatibility remain separate research problems.

12 Formal evidence and its limits

12.2 The negative controls mark mathematical boundaries

Retained regularity control	Boundary being guarded
`InvalidGlobalUniqueness`	Local root uniqueness cannot be extended to all roots.
`InvalidRectangularUniqueness`	A surjective rectangular digit map can have a nontrivial kernel.
`InvalidPerturbationExtraPrecision`	A perturbation does not supply a free additional congruence digit.
`InvalidWeightedBoundary`	The weighted residual threshold requires $r > 2s$.
`InvalidWeightedExtraPrecision`	The inverse-bound loss in retained precision can be sharp.
`InvalidMotionExtraErrorDigit`	The first-order remainder exponent cannot be uniformly increased.
`InvalidPrescribedParameter`	A fixed parameter must satisfy its actual digit equation.
`InvalidPrescribedFullJacobian`	Full Jacobian surjectivity does not replace solved-block surjectivity.
`InvalidPrescribedRectangularUniqueness`	Prescribed parameters do not eliminate a solved-block kernel.
`InvalidJointMissingParameterTerm`	Joint motion needs the actual Kv forcing term.

Retained regularity control	Boundary being guarded
`InvalidJointParameterSign`	Solved motion has the negative inverse-Jacobian sign.
`InvalidRegularityDerivativeSign`	The analytic implicit derivative is $-J^{-1}K$.

The controls compile attempted statements using positive examples that contradict the strengthened conclusions. Their expected compiler rejection is recorded evidence that the false theorem application does not type-check in the inherited run. The mathematical explanation of the relevant examples appears in the manuscript and source control modules. A failed attempt alone does not prove the negation of an arbitrary unformalized conjecture.

12.4 Remaining questions

Chapter IX closes the recorded arithmetic, smooth finite lifting, prescribed towers, quantitative deformation, and local analytic regularity results under their stated hypotheses. It retains open unrestricted native presentation invariance, completed/global compatibility and descent, general singular persistence with higher obstructions, and other specified arithmetic extensions. The analytic inverse theorem does not close those programs.

The original mathematical text and the continuation sources have separate provenance. Inclusion in this collection does not assign a blanket formal certificate to every argument. The Bernoulli specialization has a classical derivation and no matched dedicated Lean bridge.

13 Current manuscript-to-guide navigation

This table resolves all 62 inherited statement entries against the current 232-page monograph. The paper column gives physical PDF pages, counting the cover. The guide column uses printed pages. The 202 exact formal source links are retained in the paired JSON without changing their source hashes. A separate classical example follows the indexed entries. Current monograph SHA-256: `1975d95d124cdb303452950c272917c2e3fc20c19fdc882ad39b8f9744d328ca`.

Statement	Paper	Teaching chapter	Guide
3.1 `def:arith-euler`	16	Arithmetic	19
3.4 `def:arith-precision`	18	Arithmetic	19
3.6 `thm:arith-nishibuchi`	19	Arithmetic	19
3.7 `thm:arith-divisor-base`	19	Arithmetic	19
3.10 `lem:arith-newton-period`	22	Arithmetic	19
3.11 `thm:arith-dyadic-differences`	22	Dyadic results	23
3.12 `cor:arith-amplified`	24	Dyadic results	23
3.13 `cor:arith-allrow-period`	24	Dyadic results	23
3.14 `lem:arith-dominance`	24	Dyadic results	23
3.15 `thm:arith-exact`	25	Dyadic results	23
3.16 `cor:arith-least-period`	26	Dyadic results	23
3.17 `def:arith-odd-source`	26	Dyadic results	23
3.18 `thm:arith-odd-reconstruction`	27	Dyadic results	23
3.19 `cor:arith-odd-ladder`	27	Dyadic results	23
3.20 `def:arith-truncation`	28	Dyadic results	23
3.21 `lem:arith-reconstruction`	28	Dyadic results	23
3.22 `lem:arith-kappa`	29	Dyadic results	23
3.23 `thm:arith-sharp-tower`	29	Dyadic results	23
3.24 `prop:arith-row-weight`	30	Dyadic results	23
3.25 `thm:arith-isometry`	30	Dyadic results	23
3.26 `cor:arith-affine-image`	31	Dyadic results	23
3.2 `prop:arith-denominator`	17	Arithmetic	19
3.5 `prop:arith-lucas-window`	18	Arithmetic	19
3.12 `eq:arith-signed-zigzag`	27	Dyadic results	23
3.8 `ex:arith-normalization-loss`	21	Arithmetic	19
3.9 `cor:arith-window-period`	21	Arithmetic	19
10.1 `lem:polynomial-taylor`	51	Polynomial digits	31
10.2 `thm:finite-polynomial-obstruction`	51	Polynomial digits	31
10.3 `prop:affine-obstruction`	52	Affine obstructions	27
10.4 `prop:representative-independence`	52	Affine obstructions	27
10.5 `thm:common-correction`	53	Affine obstructions	27
10.6 `cor:shared-polynomial`	53	Affine obstructions	27
10.7 `thm:lift-torsor-count`	54	Polynomial digits	31
10.8 `thm:smooth-tower`	55	Compatible towers	34
10.9 `thm:smooth-padic-root`	55	Compatible towers	34
10.11 `prop:smooth-coefficient-stability`	56	Compatible towers	34
10.12 `thm:continuous-parameter-chart`	56	Parameters and charts	37
10.13 `thm:prescribed-obstruction`	57	Parameters and charts	37
10.14 `thm:prescribed-integer-tower`	58	Parameters and charts	37
10.15 `thm:prescribed-padic-tower`	58	Parameters and charts	37
10.16 `thm:weighted-lifting`	59	Weighted lifting	41
10.18 `thm:coefficient-motion`	60	Weighted lifting	41

Statement	Paper	Teaching chapter	Guide
10.19 `thm:joint-deformation`	61	Weighted lifting	41
10.17 `ex:weighted-boundary`	60	Weighted lifting	41
10.20 `ex:motion-boundary`	62	Weighted lifting	41
2.1 `thm:residue-reconstruction`	10	Exact coordinates	14
2.2 `thm:egf-coordinates`	11	Exact coordinates	14
2.3 `thm:hurwitz-product`	11	Exact coordinates	14
2.4 `thm:triangular-foundation`	12	Exact coordinates	14
2.5 `thm:filtered-foundation`	12	Exact coordinates	14
2.6 `thm:mahler-foundation`	13	Exact coordinates	14
2.7 `thm:weighted-mahler-foundation`	13	Exact coordinates	14
2.8 `thm:unit-modes-foundation`	14	Exact coordinates	14
2.9 `thm:finite-unit-foundation`	14	Exact coordinates	14
2.10 `thm:companion-foundation`	15	Exact coordinates	14
2.11 `thm:crt-foundation`	15	Exact coordinates	14
11.1 `lem:actual-polynomial-derivative`	63	Analytic regularity	46
11.2 `lem:augmented-analytic-inverse`	63	Analytic regularity	46
11.3 `thm:analytic-local-graph`	64	Analytic regularity	46
11.4 `thm:selection-regularity`	64	Analytic regularity	46
11.5 `cor:selection-open-regularity`	65	Analytic regularity	46
11.6 `cor:integral-analytic-graph`	65	Analytic regularity	46
3.3 `ex:arith-bernoulli`	17	Classical derivation	12

13.1 Strengthening and worked-example routes

These locators navigate the newer written proofs and the exact worked atlas. Their formal coverage is discussed in Section 1. The page map does not turn a written proof or a finite example into a compiled Lean endpoint.

Current statement or example	Paper PDF page
4.2 `lem:translated-hurwitz-product`	32
4.3 `lem:translated-hurwitz-inverse`	33
4.4 `thm:all-prime-power-transfer`	34
4.5 `cor:all-dyadic-final`	34
5.1 `thm:four-factorial-bound`	35
5.2 `lem:bernoulli-central-pd`	39
6.1 `thm:zero-one-exact-mixed`	41
6.3 `thm:two-exact-power-orders`	42
7.1 `lem:newton-exact-isometry-criterion`	44
7.2 `thm:rank-four-exact-periods`	44
7.3 `cor:four-exact-images`	45
7.4 `thm:lower-half-all-dyadic-isometry`	46
7.5 `prop:two-four-eight-isometry-table`	46
8.1 `thm:four-taylor-radii`	47
9.1 `cor:final-euler-congruence`	49
F.1 `atlas:reciprocal`	223
F.2 `atlas:normalization`	224
F.3 `atlas:sign`	225
F.4 `atlas:newton`	226
F.5 `atlas:smooth`	227

Current statement or example	Paper PDF page
F.6 `atlas:blocked`	228
F.7 `atlas:shared`	229
F.8 `atlas:weighted`	230

14 The retained verification frontiers

Each row records a compilation scope. The polynomial rows reuse earlier sources, so their counts must not be added. Source and log checks are distinct from compilation.

Frontier	Proof	Support	Contracts	Negative	Audits
baseline	79	0	0	0	1000
strengthening	22	0	0	14	246
shared_correction	6	0	2	3	60
polynomial_lifting	9	0	4	3	91
polynomial_counts	9	0	3	3	106
polynomial_towers	14	0	3	3	127
polynomial_deformation	17	1	3	3	159
polynomial_charts	25	2	6	4	244
polynomial_weighted	30	3	8	6	304
polynomial_motion	35	4	9	7	326
polynomial_prescribed	39	5	10	8	375
polynomial_parameter_towers	45	6	11	9	408
polynomial_joint_motion	49	7	12	11	460
polynomial_regularity	55	8	13	12	515
native_affine	2	0	0	0	17

15 Latest polynomial dependency map

These are direct imports of the 55 polynomial proof modules. The full inventory also gives imports for control and contract sources. The external Mathlib dependency and its exact identity are recorded with the sources.

Module	Direct project imports
AffineObstructionCore	Mathlib only
FiniteFiberCounts	Mathlib only
PadicTowerLimit	Mathlib only
PolynomialAnalyticInverse	Mathlib only
PolynomialMinorSelection	Mathlib only
PolynomialPadicContraction	Mathlib only
PolynomialTaylor	Mathlib only
PolynomialTaylorRing	Mathlib only
PolynomialWeightedInverse	Mathlib only
ResidueLiftDigits	Mathlib only
IntegerFiniteCokernel	AffineObstructionCore
PolynomialAnalyticEvaluation	PolynomialWeightedInverse
PolynomialMotionRational	PolynomialWeightedInverse
PolynomialObstruction	AffineObstructionCore; PolynomialTaylor
PolynomialTaylorDisplacement	PolynomialTaylorRing
PolynomialWeightedNormalization	PolynomialTaylorRing; PolynomialWeightedInverse
PolynomialDigitBridge	PolynomialObstruction
PolynomialMotionEstimate	PolynomialTaylorDisplacement; PolynomialWeightedInverse
PolynomialMotionNormalization	PolynomialWeightedNormalization
PolynomialWeightedLift	PolynomialTaylorRing; PolynomialPadicContraction; PolynomialWeightedInverse; PolynomialWeightedNormalization
PolynomialLiftCounts	PolynomialDigitBridge; ResidueLiftDigits; FiniteFiberCounts; IntegerFiniteCokernel
PolynomialRootMotion	PolynomialWeightedLift; PolynomialMotionNormalization; PolynomialMotionEstimate; PolynomialMotionRational
PolynomialSurjectiveStep	PolynomialDigitBridge
PolynomialIntegerTower	PolynomialSurjectiveStep
PolynomialSmoothCounts	PolynomialIntegerTower; PolynomialLiftCounts
PolynomialTowerControls	PolynomialIntegerTower; PadicTowerLimit
PolynomialPadicLift	PolynomialSmoothCounts; PadicTowerLimit
PolynomialCoefficientPerturbation	PolynomialPadicLift
PolynomialPadicUnique	PolynomialPadicLift
PolynomialParameterSystem	PolynomialPadicLift
PolynomialBlockInverse	PolynomialWeightedInverse; PolynomialParameterSystem
PolynomialCoordinateReorder	PolynomialParameterSystem; PolynomialMinorSelection
PolynomialInverseCongruence	PolynomialPadicUnique
PolynomialLocalSurjective	PolynomialPadicUnique
PolynomialParameterPath	PolynomialPadicUnique
PolynomialPrescribedSystem	PolynomialParameterSystem; FiniteFiberCounts
PolynomialRootStability	PolynomialPadicUnique; PolynomialCoefficientPerturbation
PolynomialAnalyticBlocks	PolynomialAnalyticEvaluation; PolynomialBlockInverse
PolynomialJointSystem	PolynomialParameterSystem; PolynomialMotionNormalization; PolynomialBlockInverse
PolynomialLocalInverse	PolynomialInverseCongruence; PolynomialLocalSurjective

Module	Direct project imports
PolynomialPrescribedLifts	PolynomialPrescribedSystem; PolynomialLiftCounts
PolynomialPrescribedTowerLimit	PolynomialParameterPath; PolynomialParameterSystem
PolynomialAnalyticImplicit	PolynomialAnalyticInverse; PolynomialJointSystem; PolynomialAnalyticBlocks
PolynomialJointMotion	PolynomialJointSystem; PolynomialBlockInverse; PolynomialRootMotion
PolynomialParameterChart	PolynomialLocalInverse; PolynomialParameterSystem
PolynomialPrescribedCounts	PolynomialPrescribedLifts; FiniteFiberCounts
PolynomialAnalyticSelection	PolynomialAnalyticImplicit
PolynomialGeneralChart	PolynomialParameterChart; PolynomialCoordinateReorder
PolynomialJointDeformation	PolynomialJointMotion
PolynomialPrescribedObstruction	PolynomialPrescribedLifts; PolynomialPrescribedCounts
PolynomialPrescribedStep	PolynomialPrescribedCounts; PolynomialSurjectiveStep
PolynomialPrescribedTowerUnique	PolynomialParameterChart; PolynomialPrescribedSystem
PolynomialRegularityIntegral	PolynomialAnalyticImplicit
PolynomialPrescribedIntegerTower	PolynomialPrescribedStep; PolynomialIntegerTower
PolynomialPrescribedPadicTower	PolynomialPrescribedIntegerTower; PolynomialPrescribedTowerLimit; PolynomialPrescribedTowerUnique

The proof chain deliberately retains its interfaces: integer Taylor identities feed digit obstructions and towers; contraction and inverse certificates feed quantitative weighted lifting; actual polynomial strict derivatives and augmented inverse blocks feed the analytic graph. A direct import is a dependency relation; it is not by itself a theorem correspondence or a fresh compiler check.

16 Complete named-source catalog

This catalog contains all 1,876 declaration rows under 199 canonical source files. The accompanying CSV retains exact-copy locations, roles, file identities and axiom closures. A theorem command, a control, a construction and a source file are different objects. A new name alone does not establish a new mathematical result.

AllRatiosDwork

SHA-256: `26a89250a7cd30e0b8bf22aa1159553965c182997a21dfd2c1c257cbc88a1948`; 4 rows.

Namespace: `KomorebiPaperIX.AllRatiosDwork`

`all_ratios_flat` (theorem; proof_source)

`packet_window_flat` (theorem; proof_source)

`ratio_window_flat` (theorem; proof_source)

`source_window_flat` (theorem; proof_source)

AllResidueEuler

SHA-256: `173902631c8b2ba6e52eecbf01e27be748ffe1e863811c1a9ec23ba46c55b9a9`; 12 rows.

Namespace: `KomorebiPaperIX.AllResidueEuler`

`all_residue_flat` (theorem; proof_source)

`decode_unnormalized` (theorem; proof_source)

`global_normalized_precision` (theorem; proof_source)

`global_unnormalized_bound` (theorem; proof_source)

`global_unnormalized_order` (theorem; proof_source)

`global_unnormalized_precision` (theorem; proof_source)

`unnormalizedLift` (def; proof_source)

`unnormalizedLift_coe` (theorem; proof_source)

`unnormalizedSeries` (def; proof_source)

`unnormalized_integral` (theorem; proof_source)

`unnormalized_successor` (theorem; proof_source)

`unnormalized_support` (theorem; proof_source)

AllResidueExamples

SHA-256: `8928578ff4f2e956636043443dc66fdc6a147e7318565018b7f26f289ad8eea4`; 8 rows.

Namespace: `KomorebiPaperIX.AllResidueExamples`

`no_loss_still_false` (theorem; positive_control)

`singular_second_difference` (theorem; positive_control)
`singular_second_valuation` (theorem; positive_control)
`singular_third` (theorem; positive_control)
`singular_unnormalized_second` (theorem; positive_control)
`upper_regular_coefficients` (theorem; positive_control)
`upper_regular_global_bound` (theorem; positive_control)
`upper_regular_unnormalized_second` (theorem; positive_control)

BinomialLogSeries

SHA-256: `8370192c821ba82844293fac3beae8244d437ef97da65c2a57ea44b1d93b5bab`; 9 rows.

Namespace: `KomorebiPaperIX.BinomialLogSeries`

`coeff_logPullback` (theorem; proof_source)
`expAt` (def; proof_source)
`expAt_derivative` (theorem; proof_source)
`factorial_coeff_logPullback` (theorem; proof_source)
`fallingAt_succ` (theorem; proof_source)
`logPullback` (def; proof_source)
`logPullback_equation` (theorem; proof_source)
`logPullback_recurrence` (theorem; proof_source)
`logPullback_zero` (theorem; proof_source)

CRTPackets

SHA-256: `1613b875e76274f541f74804bc489f696caa636128eb73ffeec29dfeab439163`; 6 rows.

Namespace: `KomorebiPaperIX.CRTPackets`

`addressEquiv` (def; proof_source)
`address_bijective` (theorem; proof_source)
`address_components` (theorem; proof_source)
`packet_factorization` (theorem; proof_source)
`residue_equivalence` (theorem; proof_source)
`series_packet_factorization` (theorem; proof_source)

CompanionPackets

SHA-256: `651d904dd712bc8aa0d0a6ed18fa07f3716639ebfb5cb6412fbc8fb0188452d5`; 19 rows.

Namespace: `KomorebiPaperIX.Companion`

`charpoly_coeff_dimension` (theorem; proof_source)

`charpoly_sum_powers` (theorem; proof_source)
`companion_jump` (theorem; proof_source)
`orbit` (def; proof_source)
`orbit_of_recurrence` (theorem; proof_source)
`orbit_succ` (theorem; proof_source)
`orbit_zero` (theorem; proof_source)
`packet` (def; proof_source)
`packet_advance` (theorem; proof_source)
`packet_as_jump_orbit` (theorem; proof_source)
`packet_charpoly_annihilation` (theorem; proof_source)
`packet_charpoly_recurrence` (theorem; proof_source)
`packet_coordinate_recurrence` (theorem; proof_source)
`packet_observable_recurrence` (theorem; proof_source)
`packet_refinement` (theorem; proof_source)
`packet_step` (theorem; proof_source)
`packet_zero_step` (theorem; proof_source)
`source_charpoly_recurrence` (theorem; proof_source)
`source_companion_jump` (theorem; proof_source)

CongruentialEuler

SHA-256: `349af85472b8e503e57e27292629e6699e1637e3ff7a81fab77eb09c3ccf49b1`; 23 rows.

Namespace: `KomorebiPaperIX.Congruential`

`clearing` (def; proof_source)
`clearing_equals_generic` (theorem; proof_source)
`clearing_product` (theorem; proof_source)
`coeff` (def; proof_source)
`coeff_zero` (theorem; proof_source)
`coefficient_recurrence` (theorem; proof_source)
`coefficient_step` (theorem; proof_source)
`coefficient_unique` (theorem; proof_source)
`diagonal_ne_zero` (theorem; proof_source)
`diagonal_positive` (theorem; proof_source)
`factorial_coefficient_identity` (theorem; proof_source)
`kernel_constant_one` (theorem; proof_source)
`packetSeries` (def; proof_source)
`packet_is_actual_inverse` (theorem; proof_source)
`packet_series_coeff` (theorem; proof_source)
`packet_times_kernel` (theorem; proof_source)
`rhs` (def; proof_source)
`row` (def; proof_source)
`sectionKernel` (def; proof_source)

`triangular_inverse` (theorem; proof_source)
`universal_denominator_clearing` (theorem; proof_source)
`unnormalized` (def; proof_source)
`unnormalized_zero` (theorem; proof_source)

ConjectureExamples

SHA-256: `46651c5e47bd95f70deb09764634766e6a53f21e02b3585f4b62f092e702bfae`; 7 rows.

Namespace: `KomorebiPaperIX.ConjectureExamples`

`dyadic_all_residues` (theorem; positive_control)
`dyadic_modulus_two` (theorem; positive_control)
`dyadic_nonzero_address_error` (theorem; positive_control)
`dyadic_nonzero_address_valuation` (theorem; positive_control)
`four_first_coefficients` (theorem; positive_control)
`four_first_unnormalized` (theorem; positive_control)
`odd_singular_address` (theorem; positive_control)

CubicDworkAlgebra

SHA-256: `b102ebc4e04ed0c7bead62b87c653925e47a4c2d6359d699ccb6953e3e898310`; 2 rows.

Namespace: `KomorebiPaperIX.CubicDworkAlgebra`

`B` (def; proof_source)
`B_mul` (theorem; proof_source)

CubicDworkClosure

SHA-256: `9ec22df4b14920247743f017401f253de62ed7ed335511d85e17123d66dbb256`; 22 rows.

Namespace: `KomorebiPaperIX.CubicDworkClosure`

`B_deriv` (theorem; proof_source)
`B_one` (theorem; proof_source)
`bounded` (def; proof_source)
`bounded_add` (theorem; proof_source)
`bounded_deriv` (theorem; proof_source)
`bounded_inverse` (theorem; proof_source)
`bounded_middle` (theorem; proof_source)
`bounded_mono` (theorem; proof_source)
`bounded_mul` (theorem; proof_source)
`bounded_nat_mul` (theorem; proof_source)

`bounded_neg` (theorem; proof_source)
`bounded_sub` (theorem; proof_source)
`bounded_succ` (theorem; proof_source)
`bounded_sum` (theorem; proof_source)
`bounded_zero` (theorem; proof_source)
`deriv_add` (theorem; proof_source)
`deriv_one_eq_zero` (theorem; proof_source)
`flat` (def; proof_source)
`flat_inverse` (theorem; proof_source)
`flat_mul` (theorem; proof_source)
`inverse_defect` (theorem; proof_source)
`iterate_B_deriv` (theorem; proof_source)

CyclicAmiceCoefficients

SHA-256: `b964b86e9c28490a1c167c4e1e6f9eec20c4b1f9a8a87462a4be3eb56b92e271`; 14 rows.

Namespace: `KomorebiPaperIX.CyclicAmiceCoefficients`

`CyclicRing` (abbrev; proof_source)
`coeff_nat_mul` (theorem; proof_source)
`coeff_nat_pow_mul` (theorem; proof_source)
`coefficient` (def; proof_source)
`coefficient_weighted_integral` (theorem; proof_source)
`generator` (def; proof_source)
`generator_pow` (theorem; proof_source)
`generator_relation` (theorem; proof_source)
`normalizer` (def; proof_source)
`normalizer_integral` (theorem; proof_source)
`normalizer_nonzero` (theorem; proof_source)
`normalizer_valuation` (theorem; proof_source)
`numerator` (def; proof_source)
`numerator_divisibility` (theorem; proof_source)

CyclotomicFiltration

SHA-256: `cd237b7f6ca7687164778924b1e569e0bd361306732d161e41967bfca8947296`; 7 rows.

Namespace: `KomorebiPaperIX.CyclotomicFiltration`

`augmentation_mem` (theorem; proof_source)
`augmentation_pow_mem` (theorem; proof_source)
`factorial_power_bound` (theorem; proof_source)
`prime_mem` (theorem; proof_source)

`torsionIdeal` (def; proof_source)

`torsion_ideal_factorial_bound` (theorem; proof_source)

`torsion_ideal_pow` (theorem; proof_source)

DifferenceFiltration

SHA-256: `572ce1909ca441ae5478f0bfd4fc4af64142c7c948344ab259eb71b8d8dfc9d6`; 4 rows.

Namespace: `KomorebiPaperIX.DifferenceFiltration`

`difference_map` (theorem; proof_source)

`saturated_bound_iff` (theorem; proof_source)

`saturated_bound_of_triangular` (theorem; proof_source)

`tail_period_higher_difference` (theorem; proof_source)

DifferenceStepMultiples

SHA-256: `128fbae2dead2a970a25cb8ac21309520bdbfb7067e74f8489180d81c3946f62`; 6 rows.

Namespace: `KomorebiPaperIX.StepMultiples`

`difference_commute_one` (theorem; proof_source)

`difference_translate` (theorem; proof_source)

`divisibility_transfer` (theorem; proof_source)

`saturated_multiple_step` (theorem; proof_source)

`single_step_telescope` (theorem; proof_source)

`translateSum` (def; proof_source)

DworkAlgebra

SHA-256: `71164e13e361dffe632db205f5c1634a1aa031080d6c969e79ee1e3bbdd4fc6d`; 10 rows.

Namespace: `KomorebiPaperIX.DworkAlgebra`

`B` (def; proof_source)

`B_mul` (theorem; proof_source)

`deriv` (def; proof_source)

`deriv_leibniz` (theorem; proof_source)

`deriv_one` (theorem; proof_source)

`deriv_succ` (theorem; proof_source)

`deriv_succ_right` (theorem; proof_source)

`deriv_zero` (theorem; proof_source)

`middle` (def; proof_source)

`prime_power_leibniz` (theorem; proof_source)

DworkClosure

SHA-256: `c55b79b8105e0e276811fd6ef58e67dbcd3853183c7fff8c3e065f6c0f3fe9fc`; 22 rows.

Namespace: `KomorebiPaperIX.DworkClosure`

`B_deriv` (theorem; proof_source)

`B_one` (theorem; proof_source)

`bounded` (def; proof_source)

`bounded_add` (theorem; proof_source)

`bounded_deriv` (theorem; proof_source)

`bounded_inverse` (theorem; proof_source)

`bounded_middle` (theorem; proof_source)

`bounded_mono` (theorem; proof_source)

`bounded_mul` (theorem; proof_source)

`bounded_nat_mul` (theorem; proof_source)

`bounded_neg` (theorem; proof_source)

`bounded_sub` (theorem; proof_source)

`bounded_succ` (theorem; proof_source)

`bounded_sum` (theorem; proof_source)

`bounded_zero` (theorem; proof_source)

`deriv_add` (theorem; proof_source)

`deriv_one_eq_zero` (theorem; proof_source)

`flat` (def; proof_source)

`flat_inverse` (theorem; proof_source)

`flat_mul` (theorem; proof_source)

`inverse_defect` (theorem; proof_source)

`iterate_B_deriv` (theorem; proof_source)

DworkCriterion

SHA-256: `62db8d69f80034062ae976c9155fc80e25bf4881feececf1a0b1155c6e251e4a`; 11 rows.

Namespace: `KomorebiPaperIX.DworkCriterion`

`defect` (def; proof_source)

`defectIter` (def; proof_source)

`defectIter_succ` (theorem; proof_source)

`defectIter_zero` (theorem; proof_source)

`defect_is_shifted_difference` (theorem; proof_source)

`defect_packet_extraction` (theorem; proof_source)

`flat` (def; proof_source)

`packed` (def; proof_source)

`packed_multiple` (theorem; proof_source)

`packet_difference` (theorem; proof_source)
`saturated_bound_of_flat` (theorem; proof_source)

DworkWindowExamples

SHA-256: `daa9a6e4d11480fee843d6768568aa0fde3ca163fa430479e93277233d24f46c`; 8 rows.

Namespace: `KomorebiPaperIX.DworkWindowExamples`

`defect_cubic_constant` (theorem; positive_control)
`derivation_rhs_constant` (theorem; positive_control)
`modulus_three_all_addresses` (theorem; positive_control)
`normalized_defect_not_derivation` (theorem; positive_control)
`window_p3_j2_coefficients` (theorem; positive_control)
`window_second_difference` (theorem; positive_control)
`window_second_difference_bound` (theorem; positive_control)
`window_second_difference_valuation` (theorem; positive_control)

DyadicExamples

SHA-256: `13b9524deedd38e67666626673176a95699632fe1715ef420d76a109c91b1ef8`; 14 rows.

Namespace: `KomorebiPaperIX.DyadicExamples`

`beta_four` (theorem; positive_control)
`beta_one` (theorem; positive_control)
`beta_three` (theorem; positive_control)
`beta_two` (theorem; positive_control)
`manuscript_error_three` (theorem; positive_control)
`manuscript_error_three_valuation` (theorem; positive_control)
`manuscript_error_zero` (theorem; positive_control)
`manuscript_error_zero_valuation` (theorem; positive_control)
`manuscript_terms` (theorem; positive_control)
`manuscript_truncations` (theorem; positive_control)
`ordinary_coefficient_not_exponential` (theorem; positive_control)
`scale_factor_cannot_be_omitted` (theorem; positive_control)
`source_anchor_values` (theorem; positive_control)
`zero_row_all_depths` (theorem; positive_control)

DyadicFourCongruences

SHA-256: `8ee06fbfedc303d1a98a287f6edbadb9a5500593e45649e7b3fa0d0bf6f88b4d`; 18 rows.

Namespace: `KomorebiPaperIX.DyadicFourCongruences`

`base_step_bound` (theorem; proof_source)
`decode_packet` (theorem; proof_source)
`dyadic_period` (theorem; proof_source)
`dyadic_period_precision` (theorem; proof_source)
`lifted_flat` (theorem; proof_source)
`normalizedLift` (def; proof_source)
`normalizedLift_coe` (theorem; proof_source)
`normalized_bound` (theorem; proof_source)
`normalized_difference_coe` (theorem; proof_source)
`normalized_precision` (theorem; proof_source)
`packet_integral_series` (theorem; proof_source)
`packet_support` (theorem; proof_source)
`unnormalizedLift` (def; proof_source)
`unnormalizedLift_coe` (theorem; proof_source)
`unnormalized_bound` (theorem; proof_source)
`unnormalized_difference_coe` (theorem; proof_source)
`unnormalized_integral` (theorem; proof_source)
`unnormalized_precision` (theorem; proof_source)

DyadicFourDwork

SHA-256: `4d9a9782f70e31b2ce842deea60414412d2e907f6904520fb8e98e5958b40138`; 15 rows.

Namespace: `KomorebiPaperIX.DyadicFourDwork`

`B4` (abbrev; proof_source)
`CoeffFlat4` (def; proof_source)
`Flat4` (def; proof_source)
`actual_packet_flat` (theorem; proof_source)
`decode_B4` (theorem; proof_source)
`decode_B4_iter` (theorem; proof_source)
`defect_coe` (theorem; proof_source)
`flat4_encode_iff` (theorem; proof_source)
`flat4_reciprocal` (theorem; proof_source)
`packed_flat_of_kummer` (theorem; proof_source)
`packet_is_reciprocal` (theorem; proof_source)
`sourceSeries` (def; proof_source)
`sourceSeries_constant` (theorem; proof_source)
`sourceSeries_encoding` (theorem; proof_source)
`sourceSeries_flat` (theorem; proof_source)

DyadicFourKernel

SHA-256: `504a069c490a2b005fe5e8e585e283226fdd8b963b09884aa322cc979296db59`; 10 rows.

Namespace: `KomorebiPaperIX.DyadicFourKernel`

`affineUnit` (def; proof_source)

`affineUnit_coe` (theorem; proof_source)

`affine_isUnit` (theorem; proof_source)

`invAffine` (def; proof_source)

`invAffine_cancel` (theorem; proof_source)

`invAffine_coe` (theorem; proof_source)

`invAffine_kummer` (theorem; proof_source)

`sourcePacket` (def; proof_source)

`source_coe` (theorem; proof_source)

`source_kummer` (theorem; proof_source)

DyadicIntegrality

SHA-256: `513898a8b2ea87ed359a3bfaee2851c3da486e24f51b80a26e494ff485119eac`; 29 rows.

Namespace: `KomorebiPaperIX.Dyadic`

`beta` (def; proof_source)

`betaPadic` (def; proof_source)

`betaPadic_coe` (theorem; proof_source)

`beta_integral` (theorem; proof_source)

`beta_recurrence` (theorem; proof_source)

`beta_zero` (theorem; proof_source)

`integralRationals` (def; proof_source)

`integral_iff` (theorem; proof_source)

`integral_iff_completed_norm` (theorem; proof_source)

`integral_of_valuation` (theorem; proof_source)

`inverse_coefficients_in_subring` (theorem; proof_source)

`kernel` (def; proof_source)

`kernelCoeff` (def; proof_source)

`kernel_coefficient_integral` (theorem; proof_source)

`kernel_coefficient_valuation` (theorem; proof_source)

`kernel_constant` (theorem; proof_source)

`sourceEven` (def; proof_source)

`sourcePadic` (def; proof_source)

`sourcePadic_coe` (theorem; proof_source)

`sourcePadic_isUnit` (theorem; proof_source)

`source_denominator_valuation` (theorem; proof_source)

`source_integral` (theorem; proof_source)

`source_nonzero` (theorem; proof_source)

source_norm_one (theorem; proof_source)
source_numerator_nonzero (theorem; proof_source)
source_numerator_valuation (theorem; proof_source)
source_valuation (theorem; proof_source)
toPadicInt (def; proof_source)
toPadicInt_coe (theorem; proof_source)

DyadicReconstruction

SHA-256: 5c6d7f50a2fe03f186c37410e9cfdfebc88fd216557f4d180e6f8bb29eac4f0a; 14 rows.

Namespace: KomorebiPaperIX.DyadicReconstruction

dyadic_reconstruction (theorem; proof_source)
dyadic_reconstruction_even_degree (theorem; proof_source)
dyadic_reconstruction_padic (theorem; proof_source)
dyadic_supercongruence (theorem; proof_source)
falling_factorial_fraction (theorem; proof_source)
full_depth_exact (theorem; proof_source)
full_depth_exact_padic (theorem; proof_source)
normalized_error_integral (theorem; proof_source)
summand (def; proof_source)
summandPadic (def; proof_source)
summandPadic_coe (theorem; proof_source)
truncation (def; proof_source)
truncationPadic (def; proof_source)
truncationPadic_coe (theorem; proof_source)

DyadicSeries

SHA-256: ab56451fee07b38657f9143d81e0e520ce74a31510f706ee9d9f27297849c998; 15 rows.

Namespace: KomorebiPaperIX.DyadicSeries

beta_original_generating_function (theorem; proof_source)
cosh_product_identity (theorem; proof_source)
dyadic_reconstruction_series (theorem; proof_source)
fullKernel (def; proof_source)
inverseSeries (def; proof_source)
inverse_even_coefficient (theorem; proof_source)
inverse_odd_coefficient (theorem; proof_source)
kernel_shift_identity (theorem; proof_source)
sinhAt (def; proof_source)
sinh_square_identity (theorem; proof_source)

`sourceSeries` (def; proof_source)
`source_even_coefficient` (theorem; proof_source)
`source_odd_coefficient` (theorem; proof_source)
`source_shift_identity` (theorem; proof_source)
`source_times_cosh` (theorem; proof_source)

DyadicUnits

SHA-256: `8d55fea825056a646ec3ee1523ae5bfcba20cf41416695bc43737d5309645b95`; 14 rows.

Namespace: `KomorebiPaperIX.DyadicUnits`

`decomposition_unique` (theorem; proof_source)
`dyadic_unit_decomposition` (theorem; proof_source)
`four_unit_cases` (theorem; proof_source)
`modFour` (def; proof_source)
`modFourUnits` (def; proof_source)
`principal` (def; proof_source)
`principal_mod_four` (theorem; proof_source)
`sign` (def; proof_source)
`sign_cases` (theorem; proof_source)
`sign_invisible_mod_two` (theorem; proof_source)
`sign_mod_four` (theorem; proof_source)
`sign_square` (theorem; proof_source)
`sign_unique` (theorem; proof_source)
`signs_distinct_mod_four` (theorem; proof_source)

EGFCoordinates

SHA-256: `cba2da32413923f441e9491a8f6ef5c53eddafcd5c5e797abd4a36516e8ca05e`; 27 rows.

Namespace: `KomorebiPaperIX.EGF`

`coordinateEquiv` (def; proof_source)
`coordinate_forward` (theorem; proof_source)
`coordinate_inverse` (theorem; proof_source)
`decode` (def; proof_source)
`decode_add` (theorem; proof_source)
`decode_encode` (theorem; proof_source)
`decode_zero` (theorem; proof_source)
`egf_product` (theorem; proof_source)
`encode` (def; proof_source)
`encode_add` (theorem; proof_source)
`encode_coeff` (theorem; proof_source)

`encode_decode` (theorem; proof_source)
`encode_zero` (theorem; proof_source)
`inverse_constant_one` (theorem; proof_source)
`matrixEntry` (def; proof_source)
`matrix_above_diagonal` (theorem; proof_source)
`matrix_diagonal` (theorem; proof_source)
`transfer` (def; proof_source)
`transfer_add` (theorem; proof_source)
`transfer_comp` (theorem; proof_source)
`transfer_one` (theorem; proof_source)
`transfer_sum` (theorem; proof_source)
`transfer_zero` (theorem; proof_source)
`transportedPacket` (def; proof_source)
`transported_idempotent` (theorem; proof_source)
`transported_orthogonal` (theorem; proof_source)
`transported_reconstruction` (theorem; proof_source)

EulerCoefficients

SHA-256: `0456c97fcefd607b7add2abd9ca8e2a43fafa2a7a5c8cd8cbf3186ddad1d245c`; 18 rows.

Namespace: `KomorebiPaperIX.EulerCoefficients`

`clearing_even_one` (theorem; proof_source)
`cosh_coeff` (theorem; proof_source)
`euler` (def; proof_source)
`eulerEven` (def; proof_source)
`eulerEven_cast` (theorem; proof_source)
`euler_egf` (theorem; proof_source)
`euler_even` (theorem; proof_source)
`euler_odd` (theorem; proof_source)
`euler_series_expansion` (theorem; proof_source)
`euler_zero` (theorem; proof_source)
`inverseCoeff` (def; proof_source)
`inverseEven` (def; proof_source)
`inverse_coeff_cast` (theorem; proof_source)
`inverse_even_egf` (theorem; proof_source)
`inverse_odd_zero` (theorem; proof_source)
`inverse_zero` (theorem; proof_source)
`original_even_section` (theorem; proof_source)
`sech_square_coeff` (theorem; proof_source)

EulerDefect

SHA-256: `460a5627629e76a6ca2c42513293904b600e2b879d1656b04ebbea1bcc5b6846`; 6 rows.

Namespace: `KomorebiPaperIX.EulerDefect`

`correction` (def; proof_source)

`defect_support` (theorem; proof_source)

`eulerOperator` (def; proof_source)

`euler_pivot_unit` (theorem; proof_source)

`flat_of_euler_flat` (theorem; proof_source)

`iterated_commutator` (theorem; proof_source)

EulerSeriesInverse

SHA-256: `1d7e58820165bbf636774c8a1362209ffdebdf9d8e55e52442f8c72038d571d0`; 25 rows.

Namespace: `KomorebiPaperIX.EulerSeriesInverse`

`Flat` (def; proof_source)

`Supported` (def; proof_source)

`encode_lift` (theorem; proof_source)

`euler` (def; proof_source)

`euler_coeff` (theorem; proof_source)

`euler_lift_encoding` (theorem; proof_source)

`euler_solve` (theorem; proof_source)

`flat_add` (theorem; proof_source)

`flat_constant` (theorem; proof_source)

`flat_mul` (theorem; proof_source)

`flat_sub` (theorem; proof_source)

`flat_zero` (theorem; proof_source)

`lift` (def; proof_source)

`solve` (def; proof_source)

`solve_coeff` (theorem; proof_source)

`solve_constant_zero` (theorem; proof_source)

`solve_euler` (theorem; proof_source)

`solve_flat` (theorem; proof_source)

`solve_integral` (theorem; proof_source)

`solve_supported` (theorem; proof_source)

`supported_add` (theorem; proof_source)

`supported_constant` (theorem; proof_source)

`supported_mul` (theorem; proof_source)

`supported_sub` (theorem; proof_source)

`supported_zero` (theorem; proof_source)

ExponentialSection

SHA-256: `be0887adc23c60fe705ab0ddfe06eec999116de56035245a9f5a4ae82fbfe80e`; 8 rows.

Namespace: `KomorebiPaperIX.ExponentialSection`

`expandedPacket` (def; proof_source)

`expandedSource` (def; proof_source)

`originalSection` (def; proof_source)

`original_section_factorization` (theorem; proof_source)

`original_section_inverse_identity` (theorem; proof_source)

`packet_coefficient_at_address` (theorem; proof_source)

`source_coefficient_at_address` (theorem; proof_source)

`source_coefficient_off_address` (theorem; proof_source)

FibonacciPackets

SHA-256: `a2663018262ad9cdb601389d8b48de4d9b6f4fdd5329522c7f219eeb64d7f6bc`; 22 rows.

Namespace: `KomorebiPaperIX.FibonacciPackets`

`companion` (def; proof_source)

`companion_power_add_two` (theorem; proof_source)

`companion_square` (theorem; proof_source)

`det_companion` (theorem; proof_source)

`det_power` (theorem; proof_source)

`fibonacci_even_step` (theorem; proof_source)

`fibonacci_odd_step` (theorem; proof_source)

`fibonacci_packet_recurrence` (theorem; proof_source)

`fibonacci_packet_unique` (theorem; proof_source)

`fibonacci_state_packet_recurrence` (theorem; proof_source)

`initial` (def; proof_source)

`jump_charpoly` (theorem; proof_source)

`jump_coeff_one` (theorem; proof_source)

`jump_coeff_zero` (theorem; proof_source)

`lucas` (def; proof_source)

`lucas_is_fibonacci_combination` (theorem; proof_source)

`packet_is_fibonacci` (theorem; proof_source)

`state` (def; proof_source)

`state_is_orbit` (theorem; proof_source)

`state_step` (theorem; proof_source)

`state_zero` (theorem; proof_source)

`trace_power_is_lucas` (theorem; proof_source)

FilteredReconstruction

SHA-256: `44d692ff84dbd425ce72b1aee131f119269e99da000d5efe073ad2fb03f9cf85`; 8 rows.

Namespace: `KomorebiPaperIX.Filtered`

`even_weight_tail` (theorem; proof_source)

`filtered_triangular_reconstruction` (theorem; proof_source)

`omitted_tail_dvd` (theorem; proof_source)

`padic_filtered_reconstruction` (theorem; proof_source)

`power_dvd_weightPower` (theorem; proof_source)

`weightPower` (def; proof_source)

`weightPower_nat` (theorem; proof_source)

`weightPower_top` (theorem; proof_source)

FiniteArithmeticExamples

SHA-256: `0c3c7d513a811b8493fc3bcdd6ede9a1af5429fb780124ba744d3c0e52d0e2b7`; 15 rows.

Namespace: `KomorebiPaperIX.FiniteArithmeticExamples`

`composite_fermat_failure` (theorem; positive_control)

`dyadic_three_principal` (theorem; positive_control)

`dyadic_three_reconstruction` (theorem; positive_control)

`dyadic_three_sign` (theorem; positive_control)

`noncoprime_components_do_not_separate` (theorem; positive_control)

`ordinary_residue_is_not_lifted_root` (theorem; positive_control)

`outside_window_denominator` (theorem; positive_control)

`outside_window_not_integral` (theorem; positive_control)

`principal_two_mod_nine` (theorem; positive_control)

`reduced_first_row` (theorem; positive_control)

`reduced_second_row` (theorem; positive_control)

`standard_window_example` (theorem; positive_control)

`teichmuller_two_mod_nine` (theorem; positive_control)

`threeModEight` (def; positive_control)

`twoModNine` (def; positive_control)

FiniteMahler

SHA-256: `9fa96a23456f589891bb58cff4f22075375a4d3fe6a00db2e5236ddcbf3facec`; 22 rows.

Namespace: `KomorebiPaperIX.Mahler`

`coefficient` (def; proof_source)

`coefficient_difference` (theorem; proof_source)
`coefficient_eq_of_prefix` (theorem; proof_source)
`coefficient_interpolant` (theorem; proof_source)
`coefficient_zero` (theorem; proof_source)
`difference` (def; proof_source)
`difference_add` (theorem; proof_source)
`difference_eq_of_stencil` (theorem; proof_source)
`difference_explicit` (theorem; proof_source)
`difference_from_coefficients` (theorem; proof_source)
`difference_mul_const` (theorem; proof_source)
`difference_one` (theorem; proof_source)
`difference_succ` (theorem; proof_source)
`difference_sum` (theorem; proof_source)
`difference_zero` (theorem; proof_source)
`finite_mahler_reconstruction` (theorem; proof_source)
`finite_sample_difference` (theorem; proof_source)
`interpolant` (def; proof_source)
`interpolant_coefficients_unique` (theorem; proof_source)
`interpolant_difference` (theorem; proof_source)
`interpolant_high_difference` (theorem; proof_source)
`interpolant_reproduces` (theorem; proof_source)

FiniteUnits

SHA-256: `935bf62202a5c0e0f0af151ac1e3651281e1abb12af2d9307ea60d7547aefb4e`; 17 rows.

Namespace: `KomorebiPaperIX.FiniteUnits`

`decomposition_unique` (theorem; proof_source)
`finite_unit_decomposition` (theorem; proof_source)
`omega` (def; proof_source)
`omega_mul` (theorem; proof_source)
`omega_residue` (theorem; proof_source)
`omega_root` (theorem; proof_source)
`omega_unique` (theorem; proof_source)
`power_lift_congruence` (theorem; proof_source)
`principal` (def; proof_source)
`principal_mul` (theorem; proof_source)
`principal_residue` (theorem; proof_source)
`residue` (def; proof_source)
`residueUnits` (def; proof_source)
`residue_zero_iff` (theorem; proof_source)
`root_fixed_by_prime` (theorem; proof_source)
`root_fixed_by_prime_power` (theorem; proof_source)
`root_unique` (theorem; proof_source)

FrobeniusSquare

SHA-256: `50716048ec83db6ceaee9a9f7d99a194aa527673acc6bc7be381cee052bcd370`; 2 rows.

Namespace: `KomorebiPaperIX.FrobeniusSquare`

`normalized_defect_integral` (theorem; proof_source)

`power_defect_divisible` (theorem; proof_source)

GlobalEulerBoundary

SHA-256: `5fb8c720fbf51e90c2de2476c7b3c5977a25495bb6a5594bd8819949e97f0417`; 8 rows.

Namespace: `KomorebiPaperIX.GlobalEulerBoundary`

`factorial_loss_transfer` (theorem; proof_source)

`lifted_difference_coe` (theorem; proof_source)

`precision` (def; proof_source)

`precision_zero` (theorem; proof_source)

`standard_window_all_orders_mod_p` (theorem; proof_source)

`standard_window_all_orders_order` (theorem; proof_source)

`standard_window_loss_transfer` (theorem; proof_source)

`unnormalized_difference` (theorem; proof_source)

HurwitzConvolution

SHA-256: `6047f1dac605fe54aab71b846cd0ac03af8e124d707e28f530dfbdfd987b2eff`; 16 rows.

Namespace: `KomorebiPaperIX.Hurwitz`

`convolution` (def; proof_source)

`convolution_add_left` (theorem; proof_source)

`convolution_add_right` (theorem; proof_source)

`convolution_at_zero` (theorem; proof_source)

`convolution_left_eq_zero` (theorem; proof_source)

`convolution_sub_left` (theorem; proof_source)

`convolution_sub_right` (theorem; proof_source)

`convolution_zero_left` (theorem; proof_source)

`convolution_zero_right` (theorem; proof_source)

`oneSeries` (def; proof_source)

`prime_shift_convolution` (theorem; proof_source)

`shift` (def; proof_source)

`shift_add` (theorem; proof_source)

`shift_convolution` (theorem; proof_source)

`shift_one_convolution` (theorem; proof_source)
`shift_zero` (theorem; proof_source)

HurwitzDworkClosure

SHA-256: `b225c928c4c25c9f870a5b38d7f3fb3c7b2eaa619d88df4b208923f7314585d6`; 17 rows.

Namespace: `KomorebiPaperIX.HurwitzDworkClosure`

`decode_B` (theorem; proof_source)
`decode_B_iterate` (theorem; proof_source)
`decode_deriv` (theorem; proof_source)
`decode_derivative` (theorem; proof_source)
`decode_encodeIntegral` (theorem; proof_source)
`decode_one` (theorem; proof_source)
`defectIter_coe` (theorem; proof_source)
`encodeIntegral` (def; proof_source)
`flat_encode_iff` (theorem; proof_source)
`flat_product` (theorem; proof_source)
`flat_reciprocal` (theorem; proof_source)
`integralSeries` (def; proof_source)
`integral_derivative` (theorem; proof_source)
`integral_encode` (theorem; proof_source)
`integral_inverse` (theorem; proof_source)
`inverse_decode_step` (theorem; proof_source)
`normalized_mem_iff` (theorem; proof_source)

HurwitzInverse

SHA-256: `ef1bc0132dd3082f7bd5e60533178e7ea3ef33c83e61ec31637cfc4c07260e06`; 7 rows.

Namespace: `KomorebiPaperIX.HurwitzInverse`

`inverse` (def; proof_source)
`inverse_convolution` (theorem; proof_source)
`inverse_prime_difference` (theorem; proof_source)
`inverse_step` (theorem; proof_source)
`inverse_support` (theorem; proof_source)
`inverse_zero` (theorem; proof_source)
`sum_on_multiples` (theorem; proof_source)

KummerSequences

SHA-256: `602180b22312175b93fd3faef6b91299dddb27fd64de2af40a899a4858d8c946`; 7 rows.

Namespace: `KomorebiPaperIX.KummerSequences`

`Kummer` (def; proof_source)

`affine_kummer` (theorem; proof_source)

`constant_kummer` (theorem; proof_source)

`difference_constant` (theorem; proof_source)

`difference_product` (theorem; proof_source)

`kummer_inverse` (theorem; proof_source)

`kummer_mul` (theorem; proof_source)

MahlerExamples

SHA-256: `e2e4fcfd857b57a888698d3f0d3199d9f1f9ed2d2c3129c7ef9027d7e5f650b2`; 15 rows.

Namespace: `KomorebiPaperIX.MahlerExamples`

`constant_mode_infinite_valuation` (theorem; positive_control)

`constant_mode_zero` (theorem; positive_control)

`exampleWeight` (def; positive_control)

`finite_sample_boundary` (theorem; positive_control)

`nonunit_mode_counterexample` (theorem; positive_control)

`square_coefficients` (theorem; positive_control)

`square_reconstructed` (theorem; positive_control)

`unit_mode_sharp_integer_example` (theorem; positive_control)

`unrestricted_interpolant_difference_false` (theorem; positive_control)

`weightedCoefficients` (def; positive_control)

`weighted_example_bound` (theorem; positive_control)

`weighted_example_difference` (theorem; positive_control)

`weighted_example_padic_difference` (theorem; positive_control)

`weighted_example_padic_valuation` (theorem; positive_control)

`weighted_example_sharp` (theorem; positive_control)

NishibuchiCompletion

SHA-256: `2abaf31d61dd1fbf3db8998f6f48b854ed3ecc497ed1d58b2da8aaa21188c99e`; 13 rows.

Namespace: `KomorebiPaperIX.NishibuchiCompletion`

`allowed_parameters` (theorem; proof_source)

`conjecture_3_9` (theorem; proof_source)

`displacementFactor` (def; proof_source)

`divisor_period` (theorem; proof_source)

`full_integrality` (theorem; proof_source)

`full_period_degree` (theorem; proof_source)

`full_period_packet` (theorem; proof_source)
`generalizedEuler` (def; proof_source)
`generalizedEuler_at_packet` (theorem; proof_source)
`modulus_divides_displacement` (theorem; proof_source)
`original_degree_shift` (theorem; proof_source)
`original_generating_identity` (theorem; proof_source)
`packet_step_factor` (theorem; proof_source)

NormalizationBoundary

SHA-256: `4881988d3339abce4b1077f05fcef68a9e65853008dc9316c6f2df0e95fda768`; 12 rows.

Namespace: `KomorebiPaperIX.Normalization`

`counterexample_parameters` (theorem; proof_source)
`difference_nonzero` (theorem; proof_source)
`difference_valuation` (theorem; proof_source)
`factorial_loss_is_sharp` (theorem; proof_source)
`factorial_valuation` (theorem; proof_source)
`false_unrestricted_bound` (theorem; proof_source)
`normalized_difference` (theorem; proof_source)
`normalized_first` (theorem; proof_source)
`normalized_second` (theorem; proof_source)
`universal_no_loss_refuted` (theorem; proof_source)
`unnormalized_difference` (theorem; proof_source)
`unnormalized_difference_valuation` (theorem; proof_source)

OddEulerExamples

SHA-256: `ab3b62d33aa2edfacac0745ee9c2bf0177bc18b3c2140ef73a102b155f1ad7af`; 11 rows.

Namespace: `KomorebiPaperIX.OddEulerExamples`

`coefficient_four` (theorem; positive_control)
`coefficient_one` (theorem; positive_control)
`coefficient_three` (theorem; positive_control)
`coefficient_two` (theorem; positive_control)
`euler_anchor_values` (theorem; positive_control)
`even_length_reconstruction_fails` (theorem; positive_control)
`inverse_anchor_values` (theorem; positive_control)
`manuscript_error` (theorem; positive_control)
`manuscript_error_sharp` (theorem; positive_control)
`manuscript_truncation` (theorem; positive_control)
`source_anchor_values` (theorem; positive_control)

OddEulerReconstruction

SHA-256: `a462465cedc9ed789ce10a89d0a6f53e2bf6059b215f066ad2dfc6059637b699`; 18 rows.

Namespace: `KomorebiPaperIX.OddEuler`

`decode_even_source` (theorem; proof_source)

`decode_rescale` (theorem; proof_source)

`filter_range_le` (theorem; proof_source)

`full_depth_exact` (theorem; proof_source)

`odd_euler_reconstruction` (theorem; proof_source)

`odd_euler_reconstruction_zigzag` (theorem; proof_source)

`odd_euler_supercongruence` (theorem; proof_source)

`odd_prime_supercongruence` (theorem; proof_source)

`odd_reconstruction_series` (theorem; proof_source)

`reconstruction_all_indices` (theorem; proof_source)

`rowEntry` (def; proof_source)

`row_above_diagonal` (theorem; proof_source)

`row_action` (theorem; proof_source)

`row_diagonal` (theorem; proof_source)

`row_weight` (theorem; proof_source)

`scaled_sech_cosh` (theorem; proof_source)

`sum_even_terms` (theorem; proof_source)

`truncation` (def; proof_source)

OddPowerSeries

SHA-256: `4b1c80671d1c701ffa6f8f28ef6cdff16fa05c0e24766df5fe61cf73a35efb1f`; 24 rows.

Namespace: `KomorebiPaperIX.OddPower`

`coshAt` (def; proof_source)

`cosh_constant` (theorem; proof_source)

`cosh_mul_sech` (theorem; proof_source)

`cosh_neg` (theorem; proof_source)

`cosh_square` (theorem; proof_source)

`evenPart` (def; proof_source)

`evenPart_coeff` (theorem; proof_source)

`expAt` (def; proof_source)

`expAt_coeff` (theorem; proof_source)

`expAt_constant` (theorem; proof_source)

`expAt_mul` (theorem; proof_source)

`expAt_zero` (theorem; proof_source)

`odd_source_egf` (theorem; proof_source)

`odd_source_even_egf` (theorem; proof_source)
`rescale_constant` (theorem; proof_source)
`rescale_cosh` (theorem; proof_source)
`rescale_expAt` (theorem; proof_source)
`rescale_sech_neg` (theorem; proof_source)
`sech` (def; proof_source)
`sech_mul_cosh` (theorem; proof_source)
`source` (def; proof_source)
`sourceSeries` (def; proof_source)
`source_as_exponentials` (theorem; proof_source)
`source_times_cosh` (theorem; proof_source)

PadicDifferenceBounds

SHA-256: `aaf74728af8a0e97badebe3365af4545093c76d9d5af560ee5187f2e803ec9ba`; 14 rows.

Namespace: `KomorebiPaperIX.PadicBounds`

`coefficient_tail_bound` (theorem; proof_source)
`difference_dvd_of_coefficients` (theorem; proof_source)
`interpolant_weighted_bound` (theorem; proof_source)
`nat_le_order_iff` (theorem; proof_source)
`order` (def; proof_source)
`order_add_lower` (theorem; proof_source)
`order_mul` (theorem; proof_source)
`order_natCast` (theorem; proof_source)
`order_nonzero` (theorem; proof_source)
`order_sum_lower` (theorem; proof_source)
`order_zero` (theorem; proof_source)
`weightedBound` (def; proof_source)
`weighted_difference_bound` (theorem; proof_source)
`weighted_difference_divisibility` (theorem; proof_source)

PeriodExamples

SHA-256: `70655437a706347237eb9e8002f0859f6f1498af13479eb2b3c05b22fc97e3c3`; 12 rows.

Namespace: `KomorebiPaperIX.PeriodExamples`

`c60_difference` (theorem; positive_control)
`c60_first` (theorem; positive_control)
`c60_second` (theorem; positive_control)
`first_difference_precision_is_one` (theorem; positive_control)
`nondivisor_period_fails` (theorem; positive_control)

`nondivisor_reduced_third` (theorem; positive_control)
`p3_m2_positive_tail` (theorem; positive_control)
`p5_m2_positive_tail` (theorem; positive_control)
`period_can_fail_at_zero` (theorem; positive_control)
`reduced_one` (theorem; positive_control)
`reduced_two` (theorem; positive_control)
`zero_has_every_precision` (theorem; positive_control)

PeriodFromDifferences

SHA-256: `7434bbb6d142befe3708a7678248ccc48ced0b25d8beb8c6bef95dd460f468c4`; 4 rows.

Namespace: `KomorebiPaperIX.PeriodFromDifferences`

`binomial_precision` (theorem; proof_source)
`displacement_expansion` (theorem; proof_source)
`prime_power_period` (theorem; proof_source)
`valuation_lt` (theorem; proof_source)

PrimeBlockPolynomial

SHA-256: `4fd07e046a0bc4c8d0d0eef91a1780477115d4c957902139343abf8018c9ca73`; 10 rows.

Namespace: `KomorebiPaperIX.PrimeBlockPolynomial`

`block_mem` (theorem; proof_source)
`falling` (def; proof_source)
`fallingAt` (def; proof_source)
`fallingAt_add` (theorem; proof_source)
`falling_blocks_mem` (theorem; proof_source)
`falling_defect_divisible` (theorem; proof_source)
`falling_eval` (theorem; proof_source)
`falling_factorial_divisibility` (theorem; proof_source)
`falling_mem` (theorem; proof_source)
`prime_field_falling` (theorem; proof_source)

RatioRiccati

SHA-256: `feaa57b9c8c751f1dba6f72f7a09d17013a5f0cf6052a4516ae1a504a5db5e88`; 9 rows.

Namespace: `KomorebiPaperIX.RatioRiccati`

`centered_equation` (theorem; proof_source)
`next_ratio_flat` (theorem; proof_source)

`original_derivative` (theorem; proof_source)
`original_nonzero` (theorem; proof_source)
`ratio` (def; proof_source)
`ratio_p_support` (theorem; proof_source)
`ratio_riccati` (theorem; proof_source)
`ratio_riccati_padic` (theorem; proof_source)
`ratio_zero` (theorem; proof_source)

RatioSeriesSupport

SHA-256: `dd94ae985ab933d532ad1df434c6457c8a02811a1bc19800792c51556b160b56`; 7 rows.

Namespace: `KomorebiPaperIX.RatioSeriesSupport`

`expanded_inverse` (theorem; proof_source)
`ratioCoefficient` (def; proof_source)
`ratioCoefficient_zero` (theorem; proof_source)
`ratio_expansion` (theorem; proof_source)
`ratio_support` (theorem; proof_source)
`singular_ratio_integral` (theorem; proof_source)
`singular_ratio_precision` (theorem; proof_source)

RecurrenceExamples

SHA-256: `8282676bd72fea6471c1b76b114c7c76d6100cd94db01355235bd0ebfa1d936c`; 15 rows.

Namespace: `KomorebiPaperIX.RecurrenceExamples`

`empty_state_recurrence` (theorem; positive_control)
`even_step_wrong_sign` (theorem; positive_control)
`identity_packet_constant` (theorem; positive_control)
`identity_packet_first_order` (theorem; positive_control)
`lucas_anchor_values` (theorem; positive_control)
`manuscript_sequence` (theorem; positive_control)
`manuscript_step_three` (theorem; positive_control)
`nilpotent` (def; positive_control)
`nilpotent_charpoly` (theorem; positive_control)
`nilpotent_packet_zero` (theorem; positive_control)
`nilpotent_square` (theorem; positive_control)
`step_one` (theorem; positive_control)
`step_two` (theorem; positive_control)
`wrong_jump_matrix` (theorem; positive_control)
`zero_step` (theorem; positive_control)

ResiduePackets

SHA-256: `337ff6013b30e09336b0e0e8cadb092864a5f5a3b6ce18c801130c5b7afc0b73`; 10 rows.

Namespace: `KomorebiPaperIX.ResiduePackets`

`packet` (def; proof_source)

`packet_apply` (theorem; proof_source)

`packet_idempotent` (theorem; proof_source)

`packet_orthogonal` (theorem; proof_source)

`packet_reconstruction` (theorem; proof_source)

`packet_refinement` (theorem; proof_source)

`seriesPacket` (def; proof_source)

`seriesPacket_coeff` (theorem; proof_source)

`series_packet_reconstruction` (theorem; proof_source)

`series_packet_refinement` (theorem; proof_source)

ResidueZeroDwork

SHA-256: `bb1209803a3619553c0b17013332ce8e167491bba730335bfe9dec96680439e0`; 11 rows.

Namespace: `KomorebiPaperIX.ResidueZeroDwork`

`encode_convolution` (theorem; proof_source)

`sectionCoefficients` (def; proof_source)

`section_defect_zero` (theorem; proof_source)

`section_flat` (theorem; proof_source)

`section_zero` (theorem; proof_source)

`zeroPacket` (def; proof_source)

`zero_packet_coe` (theorem; proof_source)

`zero_packet_flat` (theorem; proof_source)

`zero_packet_global_bound` (theorem; proof_source)

`zero_packet_is_reciprocal` (theorem; proof_source)

`zero_packet_section_product` (theorem; proof_source)

RiccatiFlatPropagation

SHA-256: `bdb9adf5c152b1d2eafeb3e741c6ab1ba007d844d7e1266e08a930dc91f4c5fb`; 10 rows.

Namespace: `KomorebiPaperIX.RiccatiFlatPropagation`

`B_iterate_prefix` (theorem; proof_source)

`forcing` (def; proof_source)

`forcing_difference` (theorem; proof_source)

`picard` (def; proof_source)
`picard_constant` (theorem; proof_source)
`picard_flat` (theorem; proof_source)
`picard_matches` (theorem; proof_source)
`picard_supported` (theorem; proof_source)
`product_vanishing_prefix` (theorem; proof_source)
`riccati_solution_flat` (theorem; proof_source)

SectionLogBridge

SHA-256: `3b519fbb417aabac764fb101a31a5012a08faa64892c305978b9cf652c9477ff`; 11 rows.

Namespace: `KomorebiPaperIX.SectionLogBridge`

`coefficient_section` (theorem; proof_source)
`factorial_coeff_section` (theorem; proof_source)
`numerator_succ` (theorem; proof_source)
`numerator_zero` (theorem; proof_source)
`pullback_equation` (theorem; proof_source)
`pullback_recurrence` (theorem; proof_source)
`pullback_zero` (theorem; proof_source)
`sectionPullback` (def; proof_source)
`sectionSeries` (def; proof_source)
`section_derivative` (theorem; proof_source)
`section_weighted_integral` (theorem; proof_source)

SeriesDifferenceBridge

SHA-256: `8c48a5db9bda683b7885b98f04ebd5757621ae73a9153ea4255afae4775dfbac`; 7 rows.

Namespace: `KomorebiPaperIX.SeriesDifferenceBridge`

`decode_rationalMap` (theorem; proof_source)
`packetLift` (def; proof_source)
`packetLift_coe` (theorem; proof_source)
`packet_divisibility` (theorem; proof_source)
`packet_encoding` (theorem; proof_source)
`packet_precision` (theorem; proof_source)
`rationalMap` (def; proof_source)

SingularAmiceExamples

SHA-256: `fa578ab72fd4b39a74866b6ffb6159b5b6a427c26ecfd5453cfde5dc77a146f3`; 10 rows.

Namespace: `KomorebiPaperIX.SingularAmiceExamples`

`normalized_section_isUnit` (theorem; positive_control)

`p3_singular_ratio_constant` (theorem; positive_control)

`p3_singular_ratio_weighted` (theorem; positive_control)

`positive_section_zero_constant` (theorem; positive_control)

`section3_bound_sharp` (theorem; positive_control)

`section3_coefficient_valuation` (theorem; positive_control)

`section3_first` (theorem; positive_control)

`section3_first_nonconstant` (theorem; positive_control)

`section3_second_two` (theorem; positive_control)

`section3_second_zero` (theorem; positive_control)

SingularNormalizedSection

SHA-256: `d3f900e7689f612e6acd0a6c78457a9232b420e26dd8657f563ff11913473e14`; 15 rows.

Namespace: `KomorebiPaperIX.SingularNormalizedSection`

`kernelPadic` (def; proof_source)

`kernel_constant` (theorem; proof_source)

`logUnit` (def; proof_source)

`logUnit_constant` (theorem; proof_source)

`log_factorization` (theorem; proof_source)

`mapped_original_factorization` (theorem; proof_source)

`normalizedUnit` (def; proof_source)

`normalized_coefficient` (theorem; proof_source)

`normalized_constant` (theorem; proof_source)

`prime_power_dvd_singular_factorial` (theorem; proof_source)

`sectionSeries_is_original` (theorem; proof_source)

`section_factorization` (theorem; proof_source)

`weighted_normalized_singular` (theorem; proof_source)

`weighted_prime_logUnit` (theorem; proof_source)

`weighted_singular_leading` (theorem; proof_source)

SingularRatioAmice

SHA-256: `ddf3691149adc29c17f2a98e177a5d649c47e12c78dc45c2491f075fd4a143d0`; 19 rows.

Namespace: `KomorebiPaperIX.SingularRatioAmice`

`X_derivative_power` (theorem; proof_source)

`derivative_map_rat` (theorem; proof_source)

`normalized_log_derivative` (theorem; proof_source)

`paper_singular_amice_estimate` (theorem; proof_source)
`ratioPullback` (def; proof_source)
`ratioPullback_is_rational_map` (theorem; proof_source)
`ratioSeries` (def; proof_source)
`ratio_cleared` (theorem; proof_source)
`ratio_constant` (theorem; proof_source)
`ratio_pullback_cleared` (theorem; proof_source)
`ratio_pullback_formula` (theorem; proof_source)
`rationalRatioPullback` (def; proof_source)
`rational_pullback_constant` (theorem; proof_source)
`sectionPadic_as_subst` (theorem; proof_source)
`section_nonzero` (theorem; proof_source)
`section_theta` (theorem; proof_source)
`singular_ratio_weighted` (theorem; proof_source)
`singular_rational_coefficient_bound` (theorem; proof_source)
`valuation_of_scaled_integral` (theorem; proof_source)

StandardWindow

SHA-256: `5d9bba7b3ad262849f5a718d5d0916674a7620c45ec7c7a3b1e60a769159af1f`; 6 rows.

Namespace: `KomorebiPaperIX.StandardWindow`

`factorial_unit_window` (theorem; proof_source)
`factorial_valuation_zero` (theorem; proof_source)
`lucas_row_reduction` (theorem; proof_source)
`standard_diagonal_mod_one` (theorem; proof_source)
`standard_diagonal_not_divisible` (theorem; proof_source)
`standard_diagonal_unit` (theorem; proof_source)

StandardWindowDwork

SHA-256: `ddcef4f0da1cc65bb5e17fd752b5610acea5c0817eefdc4ca609634a092d28a4`; 15 rows.

Namespace: `KomorebiPaperIX.StandardWindowDwork`

`defectIter_mul_const` (theorem; proof_source)
`flat_mul_const` (theorem; proof_source)
`kernelCoefficients` (def; proof_source)
`kernel_coe` (theorem; proof_source)
`kernel_constant` (theorem; proof_source)
`kernel_encoding` (theorem; proof_source)
`kernel_euler_step` (theorem; proof_source)
`kernel_flat` (theorem; proof_source)

`kernel_support` (theorem; proof_source)
`kernel_zero_address` (theorem; proof_source)
`packet_encoding` (theorem; proof_source)
`standard_window_flat` (theorem; proof_source)
`standard_window_global_bound` (theorem; proof_source)
`standard_window_global_order` (theorem; proof_source)
`window_is_reciprocal` (theorem; proof_source)

StandardWindowPeriod

SHA-256: `29cedfb69eb21401800028d2939a7fef30e39086ace7bbe86080e0d48da78ea1`; 9 rows.

Namespace: `KomorebiPaperIX.WindowPeriod`

`inverse_off_support` (theorem; proof_source)
`inverse_on_support` (theorem; proof_source)
`multiple_period` (theorem; proof_source)
`sectionCoefficients` (def; proof_source)
`section_off_support` (theorem; proof_source)
`section_prime_shift` (theorem; proof_source)
`section_zero` (theorem; proof_source)
`standard_window_period_divisibility` (theorem; proof_source)
`uniform_standard_window_period` (theorem; proof_source)

StandardWindowPrecision

SHA-256: `5c90b1298e01e43085409cd1476565e16cb71c8a0253e9cd44e6d61342e8c243`; 5 rows.

Namespace: `KomorebiPaperIX.StandardWindowPrecision`

`modulus_p_complete_family` (theorem; proof_source)
`normalized_window_precision` (theorem; proof_source)
`rational_precision_of_divisibility` (theorem; proof_source)
`residue_zero_complete_bound` (theorem; proof_source)
`unnormalized_window_precision` (theorem; proof_source)

Theorem75

SHA-256: `4f18d608ec38c38f9773e2a58ed058595d523e3860944c65a78cc9e35da06a0a`; 2 rows.

Namespace: `KomorebiPaperIX.Theorem75`

`global_finite_difference` (theorem; proof_source)
`precision_at_threshold` (theorem; proof_source)

ThetaWeighted

SHA-256: `715a740572c34e784cbe239b83da0b26f6e80aafda4ed51b906255f68f760dd9`; 30 rows.

Namespace: `KomorebiPaperIX.ThetaWeighted`

`B_constant` (theorem; proof_source)

`B_iter_mem` (theorem; proof_source)

`B_mem` (theorem; proof_source)

`B_prefix` (theorem; proof_source)

`B_y_power` (theorem; proof_source)

`Good` (def; proof_source)

`X_binomial` (theorem; proof_source)

`deriv_mem` (theorem; proof_source)

`deriv_prefix` (theorem; proof_source)

`deriv_y_power` (theorem; proof_source)

`dividedGenerator` (def; proof_source)

`dividedGenerator_mem` (theorem; proof_source)

`divided_y_defect_mem` (theorem; proof_source)

`good_X` (theorem; proof_source)

`good_add` (theorem; proof_source)

`good_const` (theorem; proof_source)

`good_dividedGenerator` (theorem; proof_source)

`good_monomial` (theorem; proof_source)

`good_mul` (theorem; proof_source)

`good_neg` (theorem; proof_source)

`good_pow` (theorem; proof_source)

`good_sum` (theorem; proof_source)

`good_y` (theorem; proof_source)

`middle_mem` (theorem; proof_source)

`monomial_factorization` (theorem; proof_source)

`theta` (def; proof_source)

`theta_apply` (theorem; proof_source)

`theta_coeff` (theorem; proof_source)

`theta_mem` (theorem; proof_source)

`theta_y_power` (theorem; proof_source)

Triangular

SHA-256: `296b4aed921d061cf5579b0a7c532da2ac324fb7229f3c24988cb6e8b1d96ac1`; 17 rows.

Namespace: `KomorebiPaperIX.Triangular`

`Integral` (def; proof_source)

`denominator` (def; proof_source)
`denominator_dvd` (theorem; proof_source)
`denominator_succ` (theorem; proof_source)
`denominator_zero` (theorem; proof_source)
`integral_add` (theorem; proof_source)
`integral_int` (theorem; proof_source)
`integral_mul` (theorem; proof_source)
`integral_promote` (theorem; proof_source)
`integral_sub` (theorem; proof_source)
`integral_sum` (theorem; proof_source)
`solve` (def; proof_source)
`solve_denominator_integral` (theorem; proof_source)
`solve_row` (theorem; proof_source)
`solve_step` (theorem; proof_source)
`solve_unique` (theorem; proof_source)
`solve_zero` (theorem; proof_source)

UnitModeDifferences

SHA-256: `e1619f4ab356ac746fbf942ec6fedb4b196809290233d10e3437347d84c3b82f`; 9 rows.

Namespace: `KomorebiPaperIX.UnitModes`

`finite_mode_formula` (theorem; proof_source)
`finite_unit_modes_divisibility` (theorem; proof_source)
`finite_unit_modes_valuation_gain` (theorem; proof_source)
`geometric_difference` (theorem; proof_source)
`geometric_difference_divisibility` (theorem; proof_source)
`unit_fermat_divisibility` (theorem; proof_source)
`unit_mode_divisibility` (theorem; proof_source)
`unit_mode_valuation_gain` (theorem; proof_source)
`unit_multiple_step_divisibility` (theorem; proof_source)

UnitRefinement

SHA-256: `4bb96f1682473774c23e51955cc4d7e730f9aab3e4a985a6baa4fd859e066709`; 6 rows.

Namespace: `KomorebiPaperIX.UnitRefinement`

`base_precision_principal` (theorem; proof_source)
`omega_compatible` (theorem; proof_source)
`principal_compatible` (theorem; proof_source)
`principal_power` (theorem; proof_source)
`reduce` (def; proof_source)
`reduceUnits` (def; proof_source)

UnitTriangular

SHA-256: `ab7423623dd1c26b41b65077960aa5d056b4c29125e1947815867b5cc0ecd26b`; 6 rows.

Namespace: `KomorebiPaperIX.UnitTriangular`

`field_solve_map` (theorem; proof_source)

`solve` (def; proof_source)

`solve_map_to_field` (theorem; proof_source)

`solve_row` (theorem; proof_source)

`solve_step` (theorem; proof_source)

`solve_unique` (theorem; proof_source)

WeightedAmice

SHA-256: `7eac9c8d5d14f3f259821dc737e451b8fcb61dc04eab31c0a40d3174407aedb2`; 23 rows.

Namespace: `KomorebiPaperIX.WeightedAmice`

`Integral` (def; proof_source)

`Weighted` (def; proof_source)

`integral_add` (theorem; proof_source)

`integral_int` (theorem; proof_source)

`integral_mul` (theorem; proof_source)

`integral_nat` (theorem; proof_source)

`integral_neg` (theorem; proof_source)

`integral_one` (theorem; proof_source)

`integral_pow` (theorem; proof_source)

`integral_sum` (theorem; proof_source)

`integral_zero` (theorem; proof_source)

`scaled_mono` (theorem; proof_source)

`scaled_product` (theorem; proof_source)

`sectionPadic` (def; proof_source)

`weighted_add` (theorem; proof_source)

`weighted_constant` (theorem; proof_source)

`weighted_inverse` (theorem; proof_source)

`weighted_mul` (theorem; proof_source)

`weighted_neg` (theorem; proof_source)

`weighted_one` (theorem; proof_source)

`weighted_pow` (theorem; proof_source)

`weighted_section` (theorem; proof_source)

`weighted_zero` (theorem; proof_source)

WeightedAmiceDifferential

SHA-256: `6d6e1ce93e29fb6baff35f764f98c60fa89a2e9abd1d7fe8e360d69f40da22e2`; 8 rows.

Namespace: `KomorebiPaperIX.WeightedAmice`

`eulerLog` (def; proof_source)

`inverse_nat_integral` (theorem; proof_source)

`linear_le_power` (theorem; proof_source)

`valuation_le_floor` (theorem; proof_source)

`weighted_X` (theorem; proof_source)

`weighted_derivative` (theorem; proof_source)

`weighted_eulerLog` (theorem; proof_source)

`weighted_log` (theorem; proof_source)

WeightedAmiceRing

SHA-256: `ab28188695852c09425cb6b80798ad8cc95f1163dedd1e0b65729dfa0eb685f7`; 15 rows.

Namespace: `KomorebiPaperIX.WeightedAmice`

`IsWeighted` (def; proof_source)

`X_mem` (theorem; proof_source)

`constant_mem` (theorem; proof_source)

`derivative_mem` (theorem; proof_source)

`inverse_coefficient` (theorem; proof_source)

`inverse_mem` (theorem; proof_source)

`mem_ring` (theorem; proof_source)

`monomial_mem` (theorem; proof_source)

`ordinary_mem` (theorem; proof_source)

`raise_weight` (theorem; proof_source)

`ring` (def; proof_source)

`shifted_weight` (theorem; proof_source)

`tail` (def; proof_source)

`tail_coeff` (theorem; proof_source)

`weighted_product_term` (theorem; proof_source)

WeightedDworkTransfer

SHA-256: `d31afdf839ff4072edd9715cc2621c76d08716acbc4b8920b26d0b4ba10b1c7e`; 12 rows.

Namespace: `KomorebiPaperIX.WeightedDworkTransfer`

`B_expPull` (theorem; proof_source)

`constant_expPull` (theorem; proof_source)
`decode_expPull` (theorem; proof_source)
`deriv_expPull` (theorem; proof_source)
`derivative_expPull` (theorem; proof_source)
`expPull` (def; proof_source)
`expPull_flat` (theorem; proof_source)
`expPull_integral` (theorem; proof_source)
`expPull_log` (theorem; proof_source)
`flat_of_log_weighted` (theorem; proof_source)
`iterate_B_expPull` (theorem; proof_source)
`singular_ratio_flat` (theorem; proof_source)

WindowIndependence

SHA-256: `247f6420080b7c26ffa8c1a001b2d85e1f03ca024918927655eb02708d804b6b`; 14 rows.

Namespace: `KomorebiPaperIX.WindowIndependence`

`coeffPadic` (def; proof_source)
`coeffPadic_coe` (theorem; proof_source)
`coeffPadic_row` (theorem; proof_source)
`diagonalUnit` (def; proof_source)
`diagonalUnit_coe` (theorem; proof_source)
`diagonal_padic_isUnit` (theorem; proof_source)
`reduced` (def; proof_source)
`reduced_row` (theorem; proof_source)
`reduced_unique` (theorem; proof_source)
`reduced_zero` (theorem; proof_source)
`residue_independence` (theorem; proof_source)
`standard_window_integral` (theorem; proof_source)
`two_window_addresses_agree` (theorem; proof_source)
`two_window_addresses_congruent` (theorem; proof_source)

ZigzagCoefficients

SHA-256: `8fe52750980c04ce9dec3605facfcfbf9a9e98acf29948fa589e148d0f7b9d19`; 18 rows.

Namespace: `KomorebiPaperIX.Zigzag`

`cos_constant` (theorem; proof_source)
`cosine_sign_expansion` (theorem; proof_source)
`derivative_cos` (theorem; proof_source)
`derivative_sec` (theorem; proof_source)
`derivative_sin` (theorem; proof_source)

derivative_tan (theorem; proof_source)
inverse_is_signed_zigzag (theorem; proof_source)
odd_zigzag_coefficient (theorem; proof_source)
secSeries (def; proof_source)
sec_cos (theorem; proof_source)
secant_odd_coeff (theorem; proof_source)
secant_sign_expansion (theorem; proof_source)
secant_square_even_coeff (theorem; proof_source)
sech_expansion (theorem; proof_source)
sin_constant (theorem; proof_source)
sin_square_add_cos_square (theorem; proof_source)
tanSeries (def; proof_source)
zigzag (def; proof_source)

AffineObstructionControls

SHA-256: 625313196b0e109d793513ecd4f48581bf41493f83f1e65c41e0d709a52f7779; 4 rows.

Namespace: Komorebi.NativeAffine.Controls

identity_unobstructed (theorem; positive_control)
singular_lower_solution (theorem; positive_control)
singular_no_lift (theorem; positive_control)
zero_map_residual_nonzero (theorem; positive_control)

FiniteAffineObstruction

SHA-256: 8f7ffb9f8387cbb23949f65d8d79ef601dd677ed567208f0bdeff475609ba056; 13 rows.

Namespace: Komorebi.NativeAffine

Obstruction (abbrev; proof_source ; 2 source files)
affine_lift_iff_obstruction_zero (theorem; proof_source ; 2 source files)
affine_residual_transport (theorem; proof_source ; 2 source files)
digitMap (def; proof_source ; 2 source files)
finite_affine_obstruction (theorem; proof_source ; 2 source files)
obstruction (def; proof_source ; 2 source files)
obstruction_finite (theorem; proof_source ; 2 source files)
obstruction_representative_independent (theorem; proof_source ; 2 source files)
obstruction_zero_iff_correction (theorem; proof_source ; 2 source files)
reduction (def; proof_source ; 2 source files)
reduction_zero_iff (theorem; proof_source ; 2 source files)
residual_exists (theorem; proof_source ; 2 source files)
same_residue_iff (theorem; proof_source ; 2 source files)

FiniteFiberCountsContract

SHA-256: `587f3ee59982c40bc9ad04da94d5fd67aee2b4d7eced7ce9f3327a5f27c5f71d`; 2 rows.

Namespace: `Komorebi.FiniteFiberCountsContract`

`explicit_fiber_and_obstruction_counts` (theorem; statement_contract)

`explicit_free_transitive` (theorem; statement_contract)

PolynomialLiftCountContract

SHA-256: `11bf03c901c3d1353e269c240d58668340e159af1e8dab5e0ac450d624b2c84e`; 2 rows.

Namespace: `Komorebi.PolynomialLiftCountContract`

`explicit_actual_polynomial_count` (theorem; statement_contract)

`explicit_jacobian_formula` (theorem; statement_contract)

ResidueDigitsContract

SHA-256: `96c06bf4928003510b7d7f0028a49cf1edf84dc64485737c920cc7e30eb00668`; 2 rows.

Namespace: `Komorebi.ResidueDigitsContract`

`explicit_residue_digit_equivalence` (def; statement_contract)

`explicit_residue_digit_formula` (theorem; statement_contract)

PolynomialCountControls

SHA-256: `a4ddb1282ee57643fa17837bbc7fff9c0409bd7d092776b52e8f52e1fe23deea`; 13 rows.

Namespace: `Komorebi.PolynomialCountControls`

`linearSystem` (def; positive_control)

`linear_digit_count` (theorem; positive_control)

`linear_lift_count` (theorem; positive_control)

`obstructed_square_digit_count` (theorem; positive_control)

`obstructed_square_lift_count` (theorem; positive_control)

`singular_square_digit_count` (theorem; positive_control)

`singular_square_lift_count` (theorem; positive_control)

`squareMinusTwo` (def; positive_control)

`squareSystem` (def; positive_control)

`unrestricted_zero_root_count` (theorem; positive_control)

`zeroSystem` (def; positive_control)

`zero_digit_count` (theorem; positive_control)

`zero_lift_count` (theorem; positive_control)

PolynomialObstructionContract

SHA-256: `b5f7d5ea68aa85097939d3913cfa0a0699ee080efd389a62746e3e2a33116d16`; 1 rows.

Namespace: `Komorebi.PolynomialObstructionContract`

`explicit_finite_polynomial_contract` (theorem; statement_contract)

PolynomialRepresentativeContract

SHA-256: `8e1ab12346c822e070d52f4482d36ea5d947a2ac6e2082e2f31365b673e0c5db`; 1 rows.

Namespace: `Komorebi.PolynomialRepresentativeContract`

`explicit_representative_contract` (theorem; statement_contract)

PolynomialSharedContract

SHA-256: `dc013485992ecdcb51602ad758bbe2b52981f98417c951c74447dc81c2489c64`; 1 rows.

Namespace: `Komorebi.PolynomialSharedContract`

`explicit_shared_polynomial_contract` (theorem; statement_contract)

PolynomialTaylorContract

SHA-256: `d6bba92070ef41f5fe3da87b040eb0caf78f73120cfb322302981d158ce1fc69`; 1 rows.

Namespace: `Komorebi.PolynomialTaylorContract`

`explicit_integral_taylor_contract` (theorem; statement_contract)

AffinePairTransport

SHA-256: `d6359e8bd04afc212633e2f902e1ea39d32b6922d0cff77e1d021634b5ca6082`; 4 rows.

Namespace: `Komorebi.AffinePair`

`HasJointLift` (def; proof_source)

`digit_correction_iff_same_image` (theorem; proof_source)

`joint_lift_iff_digit_correction` (theorem; proof_source)

`next_level_iff_digit` (theorem; proof_source)

AffineSharedCorrection

SHA-256: `b4e412dbf811a08e0fb673dbf7776e653d1d16c946306fc81e8ddee7bf1687a2`; 7 rows.

Namespace: `Komorebi.AffinePair`

`SharedObstruction` (abbrev; proof_source)

finite_affine_pair_compatibility (theorem; proof_source)
joint_lift_iff_class_eq (theorem; proof_source)
joint_lift_iff_compatible_corrections (theorem; proof_source)
left_class_independent (theorem; proof_source)
right_class_independent (theorem; proof_source)
shared_obstruction_finite (theorem; proof_source)

PolynomialControls

SHA-256: edda90f9dcd62b96e293e0e201676ceae12aea672706cf0f518528e3af7d5b89; 9 rows.

Namespace: Komorebi.PolynomialControls

even_square_cannot_be_two_mod_four (theorem; positive_control)
linearTwo (def; positive_control)
linearZero (def; positive_control)
no_common_polynomial_lift (theorem; positive_control)
separate_polynomial_lifts (theorem; positive_control)
singular_root_without_lift (theorem; positive_control)
squareMinusTwo (def; positive_control)
squarePolynomial (def; positive_control)
zero_depth_taylor_failure (theorem; positive_control)

PolynomialRepresentative

SHA-256: 68818904b747cd7b845877777458138ab0bed5938cf938a69b8e27acb5150fb2; 5 rows.

Namespace: Komorebi.PolynomialRepresentative

digit_jacobian_eq (theorem; proof_source)
digit_jacobian_shift_eq (theorem; proof_source)
representative_independence (theorem; proof_source)
residual_class_independent (theorem; proof_source)
residual_digit_transport (theorem; proof_source)

PolynomialShared

SHA-256: 8df73989caf432acdf0b7278f95d0e3f0afa1e8804b77e56ff9b09e019ff0825; 9 rows.

Namespace: Komorebi.PolynomialShared

HasJointLift (def; proof_source)
SharedObstruction (abbrev; proof_source)
finite_polynomial_pair_compatibility (theorem; proof_source)
joint_lift_iff_class_eq (theorem; proof_source)
joint_lift_iff_compatible_corrections (theorem; proof_source)
joint_lift_iff_digit_correction (theorem; proof_source)

`left_class_independent` (theorem; proof_source)
`right_class_independent` (theorem; proof_source)
`shared_obstruction_finite` (theorem; proof_source)

SharedCorrection

SHA-256: `77c9ad37f9f56bc107396050edc9399af3f4b8f3194fee3dd8a7a6a1428d7d76`; 19 rows.

Namespace: `Komorebi.SharedCorrection`

`Common` (abbrev; proof_source)
`commonKernel` (def; proof_source)
`common_finite_of_left` (theorem; proof_source)
`common_finite_of_right` (theorem; proof_source)
`common_paired_universal` (theorem; proof_source)
`common_universal` (theorem; proof_source)
`fiber` (def; proof_source)
`joint` (def; proof_source)
`jointImageEquivFiber` (def; proof_source)
`joint_range_eq_fiber` (theorem; proof_source)
`labelJ` (def; proof_source)
`labelJ_image` (theorem; proof_source)
`labelJ_surjective` (theorem; proof_source)
`labelK` (def; proof_source)
`labelK_image` (theorem; proof_source)
`labelK_surjective` (theorem; proof_source)
`labels_eq_iff_common` (theorem; proof_source)
`mem_fiber` (theorem; proof_source)
`quotient_mk_eq_iff_common` (theorem; proof_source)

PolynomialCoefficientPerturbationContract

SHA-256: `bf8c6423f4b764b7adaf51ab6a52faeca32404cd7ebda30685f442f1bfefd115`; 2 rows.

Namespace: `Komorebi.PolynomialCoefficientPerturbationContract`

`explicit_perturbed_jacobian` (theorem; statement_contract)
`explicit_power_perturbation_root` (theorem; statement_contract)

PolynomialGeneralChartContract

SHA-256: `42a57317eb813cd5f29a9a58008153ad374bb606c573bd707d057bb36a5a8859`; 1 rows.

Namespace: `Komorebi.PolynomialGeneralChartContract`

`full_rectangular_chart` (theorem; statement_contract)

PolynomialHomeomorphContract

SHA-256: `93a559a213702a9470b150b6e074862049ce976c92764f67d40c0813103588d9`; 1 rows.

Namespace: `Komorebi.PolynomialHomeomorphContract`

`square_homeomorphism` (theorem; statement_contract)

PolynomialJointMotionContracts

SHA-256: `ef407d17d713a64f01ef69eaf1f1a798768cf3b543004d43f56f00b692248eef`; 2 rows.

Namespace: `Komorebi.PolynomialJointMotionContracts`

`actual_joint_certificate` (theorem; statement_contract)

`actual_joint_rational_motion` (theorem; statement_contract)

PolynomialLocalMapContract

SHA-256: `6203a1d89bbf7e079bbbac8167ae7a5ab0ada098e573500abec6e34900136648`; 2 rows.

Namespace: `Komorebi.PolynomialLocalMapContract`

`all_depths_reflected` (theorem; statement_contract)

`arbitrary_padic_target` (theorem; statement_contract)

PolynomialPadicUniqueContract

SHA-256: `62d3afb540bb019bc3d705683d1ec4e9fd6d279c61a29ec60c13dd19598ad6f1`; 2 rows.

Namespace: `Komorebi.PolynomialPadicUniqueContract`

`explicit_arbitrary_root_uniqueness` (theorem; statement_contract)

`explicit_unique_padic_root` (theorem; statement_contract)

PolynomialParameterTowerContracts

SHA-256: `40a450b93c3d686cfd7ea84081a0cb061e00730cc96ff542015e4e8751d48ae6`; 5 rows.

Namespace: `Komorebi.PolynomialParameterTowerContracts`

`actual_integer_path_tower` (theorem; statement_contract)

`actual_square_parameter_uniqueness` (theorem; statement_contract)

`arbitrary_padic_parameter_path` (theorem; statement_contract)

`arbitrary_padic_target_tower` (theorem; statement_contract)

`supplied_path_actual_endpoint` (theorem; statement_contract)

PolynomialPrescribedContract

SHA-256: `6598a12ba513de9a23f1745a50d241ae5ca5a5dade6771c312747171b166c6cc`; 5 rows.

Namespace: `Komorebi.PolynomialPrescribedContract`

`ActualResidues` (abbrev; statement_contract)

`actualResidueEquiv` (def; statement_contract)

`actualSolvedMap` (def; statement_contract)

`actual_lower_root_obstruction` (theorem; statement_contract)

`actual_lower_root_prime_count` (theorem; statement_contract)

PolynomialRegularityContracts

SHA-256: `4b3ef7ae03c57942325aaafc278848ac1c339c6131235c8e053c7cf7ffa5eff6`; 5 rows.

Namespace: `Komorebi.PolynomialRegularityContracts`

`actual_integral_analytic_root_graph` (theorem; statement_contract)

`actual_padic_analytic_root_graph` (theorem; statement_contract)

`actual_polynomial_evaluation` (theorem; statement_contract)

`actual_selection_analytic_on_open` (theorem; statement_contract)

`actual_selection_germ` (theorem; statement_contract)

PolynomialRootMotionContract

SHA-256: `c1f671c373ed386790176b517c19c65974c43fc0c952907adbd72c955bc3db5b`; 2 rows.

Namespace: `Komorebi.PolynomialRootMotionContract`

`original_jacobian_integral_error` (theorem; statement_contract)

`original_jacobian_rational_motion` (theorem; statement_contract)

PolynomialRootStabilityContract

SHA-256: `cb5dddb6def6a35ff96eccd9096c42a11476a4e0cbbb371dd8a552324f583c62`; 3 rows.

Namespace: `Komorebi.PolynomialRootStabilityContract`

`explicit_lower_root_and_unique_perturbation` (theorem; statement_contract)

`explicit_rectangular_nearby_root` (theorem; statement_contract)

`explicit_square_root_displacement` (theorem; statement_contract)

PolynomialWeightedAuxiliaryContract

SHA-256: `a734bcfd612bfbbd4ffe694d6795d5198ee9dc9124d499ac0076474a37f4c057`; 3 rows.

Namespace: `Komorebi.PolynomialWeightedAuxiliaryContract`

`actual_polynomial_contraction` (theorem; statement_contract)
`exact_inverse_denominator_certificate` (theorem; statement_contract)
`uniform_polynomial_taylor` (theorem; statement_contract)

PolynomialWeightedLiftContract

SHA-256: `f74ab51190f4e7e6a4fa503ec2216bebd5496c6bbd160054196b4ba0e17679ef`; 2 rows.

Namespace: `Komorebi.PolynomialWeightedLiftContract`

`actual_jacobian_bounded_inverse` (theorem; statement_contract)
`actual_jacobian_integral_certificate` (theorem; statement_contract)

PolynomialDeformationControls

SHA-256: `72809a109644db6ce09ffe1b1f7244e2759d4ccbca060f0357be82a6c8a80439`; 6 rows.

Namespace: `Komorebi.PolynomialDeformationControls`

`linear_shift_coefficient_perturbation` (theorem; positive_control)
`linear_shift_no_extra_padic_precision` (theorem; positive_control)
`squareMinusOne` (def; positive_control)
`square_minus_one_jacobian_bijective` (theorem; positive_control)
`square_minus_one_not_global_unique` (theorem; positive_control)
`square_minus_one_two_padic_roots` (theorem; positive_control)

PolynomialJointMotionControls

SHA-256: `7f564ab0c011cae38264f3d7f149864649fdd329bb28412bdb94e5e2e3cdb362`; 16 rows.

Namespace: `Komorebi.PolynomialJointMotionControls`

`exists_sharp_parameter_only_root` (theorem; positive_control)
`join_one_one` (theorem; positive_control)
`join_one_zero` (theorem; positive_control)
`linearPerturbation` (def; positive_control)
`linearSystem` (def; positive_control)
`linear_actual_jacobians` (theorem; positive_control)
`linear_augmented_root_depth_two` (theorem; positive_control)
`linear_base_data` (theorem; positive_control)
`linear_joint_forcing_and_error` (theorem; positive_control)
`linear_joint_root` (theorem; positive_control)
`linear_joint_unique` (theorem; positive_control)
`no_uniform_extra_parameter_error_digit` (theorem; positive_control)
`omitted_parameter_term_fails` (theorem; positive_control)
`quadraticParameterSystem` (def; positive_control)
`quadratic_parameter_base_data` (theorem; positive_control)
`reversed_parameter_sign_fails` (theorem; positive_control)

PolynomialMotionControls

SHA-256: `4c90b8177833dc905f9cd6996805682dce59e9084dcaf1e4e333d5370a14d883`; 6 rows.

Namespace: `Komorebi.PolynomialMotionControls`

`exists_sharp_motion_root` (theorem; positive_control)

`no_uniform_extra_error_digit` (theorem; positive_control)

`sharpQuadratic` (def; positive_control)

`sharpQuadratic_base` (theorem; positive_control)

`sharp_error_of_even_root` (theorem; positive_control)

`sharp_mod_eight` (theorem; positive_control)

PolynomialParameterControls

SHA-256: `43a88e22897c64c0403330d0189827c18eec60bf45c0a85e9ac0eb45d38b7a83`; 6 rows.

Namespace: `Komorebi.PolynomialParameterControls`

`full_jacobian_surjective` (theorem; positive_control)

`no_root_at_parameter_two` (theorem; positive_control)

`parameter_two_near_origin` (theorem; positive_control)

`solved_jacobian_not_surjective` (theorem; positive_control)

`solved_jacobian_zero` (theorem; positive_control)

`squareMinusParameter` (def; positive_control)

PolynomialParameterTowerControls

SHA-256: `41037b0fbd33d89ee7c4601a05455441d4d618ae341bea9a2789b8730bdee5bb`; 3 rows.

Namespace: `Komorebi.PolynomialParameterTowerControls`

`firstSolved` (def; positive_control)

`firstSolved_surjective` (theorem; positive_control)

`rectangular_same_parameter_nonunique` (theorem; positive_control)

PolynomialPrescribedControls

SHA-256: `5cce4a5e5295ba98e65ea52efe72741b92e2607bfbe4844e8d5ad647c8be8a5a`; 15 rows.

Namespace: `Komorebi.PolynomialPrescribedControls`

`digit_one_empty` (theorem; positive_control)

`digit_one_for_three_empty` (theorem; positive_control)

`digit_two_for_three_exists` (theorem; positive_control)

`full_jacobian_surjective` (theorem; positive_control)

`parameterOnly` (def; positive_control)

`parameterPlusThree` (def; positive_control)

`prescribed_one_for_three_no_actual_lift` (theorem; positive_control)
`prescribed_one_no_actual_lift` (theorem; positive_control)
`prescribed_two_for_three_actual_lift` (theorem; positive_control)
`prescribed_zero_actual_card` (theorem; positive_control)
`solved_jacobian_zero` (theorem; positive_control)
`three_residual` (theorem; positive_control)
`zeroDigitEquiv` (def; positive_control)
`zero_digit_card` (theorem; positive_control)
`zero_residual` (theorem; positive_control)

PolynomialRegularityControls

SHA-256: `a84a69f9c32efb43b4bfbccf86b7879002f4327ad8be06da3bf03c75793605e2`; 12 rows.

Namespace: `Komorebi.PolynomialRegularityControls`

`linearSystem` (def; positive_control)
`linear_analytic` (theorem; positive_control)
`linear_derivative_not_positive` (theorem; positive_control)
`linear_parameterJacobian` (theorem; positive_control)
`linear_root_graph` (theorem; positive_control)
`linear_solvedJacobian` (theorem; positive_control)
`linear_strict_derivative` (theorem; positive_control)
`quadraticSystem` (def; positive_control)
`quadratic_analytic_branch` (theorem; positive_control)
`quadratic_parameterJacobian` (theorem; positive_control)
`quadratic_root` (theorem; positive_control)
`quadratic_solvedJacobian` (theorem; positive_control)

PolynomialWeightedControls

SHA-256: `88fea6bc068832ba84339af00cd98b0f9b458219dd3df71c1578f445cb7fdf04`; 11 rows.

Namespace: `Komorebi.PolynomialWeightedControls`

`boundary_exact_residual_depth` (theorem; positive_control)
`boundary_no_mod_eight` (theorem; positive_control)
`boundary_no_padic_root` (theorem; positive_control)
`boundary_no_root` (theorem; positive_control)
`boundary_residual_derivative` (theorem; positive_control)
`boundary_scaled_inverse` (theorem; positive_control)
`linear_displacement_sharp` (theorem; positive_control)
`linear_perturbation_root` (theorem; positive_control)
`prime_power_exact_depth` (theorem; positive_control)
`squarePlusThree` (def; positive_control)
`weightedLinear` (def; positive_control)

AffineObstructionCore

SHA-256: `8af3cd815f87519dfa90aec72244a7924f19d4291d29f116a97fcce31af9f80f`; 13 rows.

Namespace: `Komorebi.NativeAffine`

Obstruction (abbrev; proof_source ; 2 source files)

affine_lift_iff_obstruction_zero (theorem; proof_source ; 2 source files)

affine_residual_transport (theorem; proof_source ; 2 source files)

digitMap (def; proof_source ; 2 source files)

finite_affine_obstruction (theorem; proof_source ; 2 source files)

obstruction (def; proof_source ; 2 source files)

obstruction_finite (theorem; proof_source ; 2 source files)

obstruction_representative_independent (theorem; proof_source ; 2 source files)

obstruction_zero_iff_correction (theorem; proof_source ; 2 source files)

reduction (def; proof_source ; 2 source files)

reduction_zero_iff (theorem; proof_source ; 2 source files)

residual_exists (theorem; proof_source ; 2 source files)

same_residue_iff (theorem; proof_source ; 2 source files)

FiniteFiberCounts

SHA-256: `4e7e7fcc6350236114543d1db3c3b290da45b1bcd0b92b630cb9581ec08c6528`; 20 rows.

Namespace: `Komorebi.FiniteFiberCounts`

`Fiber` (def; proof_source)

`cokernel_card` (theorem; proof_source)

`cokernel_finrank` (theorem; proof_source)

`difference` (def; proof_source)

`difference_translate` (theorem; proof_source)

`fiberAddTorsor` (def; proof_source)

`fiberEquivKernel` (def; proof_source)

`fiber_card` (theorem; proof_source)

`fiber_card_eq_kernel` (theorem; proof_source)

`fiber_card_of_empty` (theorem; proof_source)

`fiber_card_of_nonempty` (theorem; proof_source)

`fiber_nonempty_iff_class_zero` (theorem; proof_source)

`kernel_card` (theorem; proof_source)

`kernel_finrank` (theorem; proof_source)

`rank` (def; proof_source)

`translate` (def; proof_source)

`translate_add` (theorem; proof_source)

`translate_difference` (theorem; proof_source)

`translate_exists_unique` (theorem; proof_source)

`translate_zero` (theorem; proof_source)

IntegerFiniteCokernel

SHA-256: `ca844b9373b49e64974717ab73eddf56ddfa6df45b11e9c4334c9ac2c1bae8e3`; 6 rows.

Namespace: `Komorebi.IntegerFiniteCokernel`

`cokernelEquiv` (def; proof_source)

`cokernelEquiv_mk` (theorem; proof_source)

`cokernelEquiv_obstruction` (theorem; proof_source)

`cokernel_card_eq` (theorem; proof_source)

`digit_range_eq_restrict` (theorem; proof_source)

`reduction_surjective` (theorem; proof_source)

PadicTowerLimit

SHA-256: `43d4cab78912d49bb0c23645d3b1af4d4148d188781c31957e611902ff603528`; 6 rows.

Namespace: `Komorebi.PadicTowerLimit`

`compatible_int_tower_cauchy` (theorem; proof_source)

`compatible_int_tower_limit` (theorem; proof_source)

`compatible_int_tower_unique_limit` (theorem; proof_source)

`compatible_vector_tower_limit` (theorem; proof_source)

`divisible_int_tendsto_zero` (theorem; proof_source)

`tower_pair_dvd` (theorem; proof_source)

PolynomialAnalyticBlocks

SHA-256: `19fa21ab3877afe7925003f2c142267b214159d9c217f94204e5765e8e1d5955`; 6 rows.

Namespace: `Komorebi.PolynomialAnalyticBlocks`

`inverse_free_zero` (theorem; proof_source)

`inverse_solved_zero` (theorem; proof_source)

`upperEquiv` (def; proof_source)

`upperEquiv_apply` (theorem; proof_source)

`upperEquiv_symm_apply` (theorem; proof_source)

`upperEquiv_toCLM` (theorem; proof_source)

PolynomialAnalyticEvaluation

SHA-256: `f8d75e73ac2df1e336c55889da4e81184f54ff6bffe94eee0112d90e905ff57f`; 9 rows.

Namespace: `Komorebi.PolynomialAnalyticEvaluation`

`analyticAt_evalSystem` (theorem; proof_source)

`analyticOnNhd_evalSystem` (theorem; proof_source)

`evalSystem` (def; proof_source)

`hasStrictFDerivAt_eval` (theorem; proof_source)
`hasStrictFDerivAt_evalSystem` (theorem; proof_source)
`matrixCLM` (def; proof_source)
`matrixCLM_apply` (theorem; proof_source)
`polynomialDeriv` (def; proof_source)
`polynomialDeriv_apply` (theorem; proof_source)

PolynomialAnalyticImplicit

SHA-256: `4fb68eb1f5417a3ecc46a281d5eddb23afa551ac3cb4c65354b8b907c681bb7a`; 7 rows.

Namespace: `Komorebi.PolynomialAnalyticImplicit`

`differentiableOn_of_analytic_graph` (theorem; proof_source)
`exists_analytic_graph` (theorem; proof_source)
`exists_analytic_implicit` (theorem; proof_source)
`parameterInjectionCLM` (def; proof_source)
`parameterInjectionCLM_apply` (theorem; proof_source)
`solvedPartCLM` (def; proof_source)
`solvedPartCLM_apply` (theorem; proof_source)

PolynomialAnalyticInverse

SHA-256: `4139bc9fe166b3f9aa926fa1f5d4b83d5155c2a31fa0826dac8e48fad422b3c7`; 2 rows.

Namespace: `Komorebi.PolynomialAnalyticInverse`

`analyticAt_inverse` (theorem; proof_source)
`exists_open_analytic_inverse` (theorem; proof_source)

PolynomialAnalyticSelection

SHA-256: `ed23cb0c55b662bfb8270fa17e2af27c4dff8c5605c6a1e65c4fa35bbca0ba82`; 3 rows.

Namespace: `Komorebi.PolynomialAnalyticSelection`

`continuous_selection_analyticOnNhd` (theorem; proof_source)
`continuous_selection_eventually_eq` (theorem; proof_source)
`continuous_selection_regularity` (theorem; proof_source)

PolynomialBlockInverse

SHA-256: `398f1123b76a2d458bac4eb6d671d15ad41c36d329c70cac8a3fdc0bca3a945a`; 12 rows.

Namespace: `Komorebi.PolynomialBlockInverse`

`blockCertificate` (def; proof_source)
`blockCertificate_mul` (theorem; proof_source)

`blockCertificate_mulVec_join_neg` (theorem; proof_source)
`blockCertificate_two_sided` (theorem; proof_source)
`bounded_inverse_upperBlock` (theorem; proof_source)
`mul_blockCertificate` (theorem; proof_source)
`upperBlock` (def; proof_source)
`upperBlock_free_free` (theorem; proof_source)
`upperBlock_free_solved` (theorem; proof_source)
`upperBlock_mulVec_join` (theorem; proof_source)
`upperBlock_solved_free` (theorem; proof_source)
`upperBlock_solved_solved` (theorem; proof_source)

PolynomialCoefficientPerturbation

SHA-256: `fbb6841c84bf08cd074cd3ecdb40ef9fec4cc65708272abd0fd034f7ccbc5198`; 9 rows.

Namespace: `Komorebi.PolynomialCoefficientPerturbation`

`digitJacobian_perturb_eq` (theorem; proof_source)
`digitJacobian_perturb_point_eq` (theorem; proof_source)
`evalSystem_perturb` (theorem; proof_source)
`exists_padic_root_perturb` (theorem; proof_source)
`exists_padic_root_power_perturb` (theorem; proof_source)
`lower_root_perturb` (theorem; proof_source)
`padicEval_perturb` (theorem; proof_source)
`pderiv_perturb` (theorem; proof_source)
`perturb` (def; proof_source)

PolynomialCoordinateReorder

SHA-256: `d8a5e2e71e5c56103816a083a914f8ff01e24b881ecae3e9c04a97a98a7f2652`; 14 rows.

Namespace: `Komorebi.PolynomialCoordinateReorder`

`continuous_reorderPoint` (theorem; proof_source)
`digitJacobian_single` (theorem; proof_source)
`evalSystem_reorder` (theorem; proof_source)
`evalSystem_reorder_at` (theorem; proof_source)
`exists_reorder_solvedJacobian_surjective` (theorem; proof_source)
`padicEval_reorder` (theorem; proof_source)
`padicEval_reorder_at` (theorem; proof_source)
`pderiv_eval_reorder` (theorem; proof_source)
`pderiv_reorder` (theorem; proof_source)
`reorderPoint` (def; proof_source)
`reorderPoint_inverse` (theorem; proof_source)
`reorderPoint_inverse'` (theorem; proof_source)
`reorderSystem` (def; proof_source)
`solvedJacobian_reorder_apply` (theorem; proof_source)

PolynomialDigitBridge

SHA-256: `6a985c2c6308910d148283297e71b0cd30b1abdc86056898aa2b5294fedc27e2`; 7 rows.

Namespace: `Komorebi.PolynomialDigitBridge`

`DigitSolutions` (abbrev; proof_source)

`digitJacobian` (def; proof_source)

`digit_solution_iff` (theorem; proof_source)

`digit_solutions_nonempty_iff` (theorem; proof_source)

`integerLift` (def; proof_source)

`reduction_integerLift` (theorem; proof_source)

`reduction_jacobian` (theorem; proof_source)

PolynomialGeneralChart

SHA-256: `492bccc855312c5b262c31bbfe1b7521dc03f30ef15afc3739d05874ff6062c8`; 2 rows.

Namespace: `Komorebi.PolynomialGeneralChart`

`exists_parameter_homeomorph` (theorem; proof_source)

`reorderRootHomeomorph` (def; proof_source)

PolynomialIntegerTower

SHA-256: `1b36300b65e2771cb785fce035044c2b80e4210412133077be7788785a9c16a8`; 5 rows.

Namespace: `Komorebi.PolynomialIntegerTower`

`exists_compatible_tower` (theorem; proof_source)

`exists_compatible_tower_strong` (theorem; proof_source)

`tower_digitJacobian_eq` (theorem; proof_source)

`tower_initial_congruent` (theorem; proof_source)

`tower_pairwise_congruent` (theorem; proof_source)

PolynomialInverseCongruence

SHA-256: `298bba9f501f5dc5a983a3eb7a34fa6790d7ec16279b220101fae23f5bf32ebf`; 6 rows.

Namespace: `Komorebi.PolynomialInverseCongruence`

`integer_inverse_congruence` (theorem; proof_source)

`integer_inverse_congruence_step` (theorem; proof_source)

`padicEval_congruent` (theorem; proof_source)

`padic_congruence_iff` (theorem; proof_source)

`padic_inverse_congruence` (theorem; proof_source)

`padic_map_injective` (theorem; proof_source)

PolynomialJointDeformation

SHA-256: `5cef4d16ad05e035326f23b1ec62468b4e1e5708e29218513d9fb22528450345`; 3 rows.

Namespace: `Komorebi.PolynomialJointDeformation`

`exists_joint_deformation` (theorem; proof_source)

`exists_unique_joint_deformation` (theorem; proof_source)

`join_deformed_ball` (theorem; proof_source)

PolynomialJointMotion

SHA-256: `946ca8e3e4f17c71e66db136a1ab70b1012c60dbc1f87ded33f2d7defb1b84e4`; 2 rows.

Namespace: `Komorebi.PolynomialJointMotion`

`exists_joint_root_certificate` (theorem; proof_source)

`exists_joint_root_motion` (theorem; proof_source)

PolynomialJointSystem

SHA-256: `01aec0f34cdcf5a6f9288f4af342bb804678ecb1713b1aef40cf6fa227e767df`; 17 rows.

Namespace: `Komorebi.PolynomialJointSystem`

`augmented` (def; proof_source)

`augmentedPerturbation` (def; proof_source)

`augmentedPerturbation_free` (theorem; proof_source)

`augmentedPerturbation_solved` (theorem; proof_source)

`augmented_free` (theorem; proof_source)

`augmented_root` (theorem; proof_source)

`augmented_solved` (theorem; proof_source)

`eval_augmented` (theorem; proof_source)

`eval_augmentedPerturbation` (theorem; proof_source)

`jacobian_augmented` (theorem; proof_source)

`jacobian_augmented_free_free` (theorem; proof_source)

`jacobian_augmented_free_solved` (theorem; proof_source)

`jacobian_augmented_solved_free` (theorem; proof_source)

`jacobian_augmented_solved_solved` (theorem; proof_source)

`parameterJacobian` (def; proof_source)

`perturb_augmented_root_iff` (theorem; proof_source)

`solvedJacobian` (def; proof_source)

PolynomialLiftCounts

SHA-256: `10c01bfa8a892f0b6aacb7f36e1c0c86cef6ddfa51917ae5c0a84be32f963931`; 10 rows.

Namespace: `Komorebi.PolynomialLiftCounts`

`ActualLifts` (abbrev; proof_source)

`digitAddTorsor` (def; proof_source)

`digitEquivLifts` (def; proof_source)

`digit_residue_solution_iff` (theorem; proof_source)

`finite_polynomial_lift_count` (theorem; proof_source)

`liftEquivKernel` (def; proof_source)

`lift_card` (theorem; proof_source)

`obstructionEquiv` (def; proof_source)

`obstructionEquiv_class` (theorem; proof_source)

`obstruction_card` (theorem; proof_source)

PolynomialLocalInverse

SHA-256: `11d6dda31c7932a5175a1ecfb99f18e0a9d287098997bfc5a43789af8385f635`; 7 rows.

Namespace: `Komorebi.PolynomialLocalInverse`

`ballMap` (def; proof_source)

`ballMap_bijective` (theorem; proof_source)

`continuous_ballMap` (theorem; proof_source)

`localHomeomorph` (def; proof_source)

`localHomeomorph_apply` (theorem; proof_source)

`localHomeomorph_symm_congruent` (theorem; proof_source)

`localHomeomorph_symm_eval` (theorem; proof_source)

PolynomialLocalSurjective

SHA-256: `c4ddf4985f7739b5e8555d103c85e131524d7c04d855bbbe94826ba58824c937`; 7 rows.

Namespace: `Komorebi.PolynomialLocalSurjective`

`digitJacobian_sub_C` (theorem; proof_source)

`exists_integer_target_preimage` (theorem; proof_source)

`exists_padic_preimage` (theorem; proof_source)

`isClosed_residueBall` (theorem; proof_source)

`isCompact_residueBall` (theorem; proof_source)

`residueBall` (def; proof_source)

`truncInt_tendsto` (theorem; proof_source)

PolynomialMinorSelection

SHA-256: `00ca2ec49dcaf069c90d882e7edf4b088c68f7d4a37f42fcd728e39f351e4c3e`; 3 rows.

Namespace: `Komorebi.PolynomialMinorSelection`

`exists_permuted_columns` (theorem; proof_source)

`exists_prime_permuted_columns` (theorem; proof_source)
`exists_selected_columns` (theorem; proof_source)

PolynomialMotionEstimate

SHA-256: `b805b3ce934dc0eeda54be631e7a1210c1919d4c231b9794cdf6eeb64e6e2f4b`; 1 rows.

Namespace: `Komorebi.PolynomialMotionEstimate`

`first_order_error_dvd` (theorem; proof_source)

PolynomialMotionNormalization

SHA-256: `488c028384e45fe3e9a2f3365227bf4f2583beddc73b7ad76faf8e5b605249c2`; 5 rows.

Namespace: `Komorebi.PolynomialMotionNormalization`

`eval_perturbSystem` (theorem; proof_source)
`exists_eval_scaled_difference` (theorem; proof_source)
`exists_motion_normalization` (theorem; proof_source)
`perturbSystem` (def; proof_source)
`precondition_perturbSystem` (theorem; proof_source)

PolynomialMotionRational

SHA-256: `17d45ac3501b93146f907f59702e96fabee320fb15449756685fe3e9fb480beb`; 3 rows.

Namespace: `Komorebi.PolynomialMotionRational`

`coe_mulVec` (theorem; proof_source)
`exists_rational_error` (theorem; proof_source)
`scaled_inverse_mulVec` (theorem; proof_source)

PolynomialObstruction

SHA-256: `ab0670c8d9c84aaf36e0e83f396c26f0514b8279f694d7b3653c3ab43410c653`; 12 rows.

Namespace: `Komorebi.PolynomialObstruction`

`dvd_iff_of_dvd_sub` (theorem; proof_source)
`evalSystem` (def; proof_source)
`evalSystem_apply` (theorem; proof_source)
`finite_polynomial_obstruction` (theorem; proof_source)
`jacobian` (def; proof_source)
`jacobian_apply` (theorem; proof_source)
`next_level_iff_digit` (theorem; proof_source)
`polynomial_lift_iff_obstruction_zero` (theorem; proof_source)
`polynomial_same_residue_lift_iff` (theorem; proof_source)

`residual_exists` (theorem; proof_source)
`residual_unique` (theorem; proof_source)
`taylor_residual_dvd` (theorem; proof_source)

PolynomialPadicContraction

SHA-256: `e667bd03d6577c8a7180659ef5fa95f068ccc172a7cf43d0f7f2508156f8e7ce`; 9 rows.

Namespace: `Komorebi.PolynomialPadicContraction`

`contracting_contractionMap` (theorem; proof_source)
`contractionMap` (def; proof_source)
`evalSystem` (def; proof_source)
`eval_dist_le` (theorem; proof_source)
`eval_dvd_of_dvd` (theorem; proof_source)
`exists_unique_fixedPoint` (theorem; proof_source)
`exists_unique_fixedPoint_divisible` (theorem; proof_source)
`fixedPoint_divisible` (theorem; proof_source)
`lipschitz_evalSystem` (theorem; proof_source)

PolynomialPadicLift

SHA-256: `072b6ad7c76a858d63d674a9eabd6d0604a9d2903ce7f0d81bf5c7ab3eb6cda6`; 6 rows.

Namespace: `Komorebi.PolynomialPadicLift`

`continuous_padicEval` (theorem; proof_source)
`exists_padic_root` (theorem; proof_source)
`exists_padic_root_with_tower` (theorem; proof_source)
`padicEval` (def; proof_source)
`padicEval_intCast` (theorem; proof_source)
`polynomial_tower_limit_root` (theorem; proof_source)

PolynomialPadicUnique

SHA-256: `51a734f436ec118fd39cf20ad228eedaffe5225eeb989b5a1040deb701c0f4cd`; 10 rows.

Namespace: `Komorebi.PolynomialPadicUnique`

`exists_unique_padic_root` (theorem; proof_source)
`int_pow_dvd_iff` (theorem; proof_source)
`padic_root_unique` (theorem; proof_source)
`rootResidueLift` (def; proof_source)
`root_next_residue_eq` (theorem; proof_source)
`toZModPow_eq_iff_dvd_sub` (theorem; proof_source)
`truncInt` (def; proof_source)
`truncInt_eval_residue` (theorem; proof_source)

`truncInt_precision` (theorem; proof_source)
`truncInt_root` (theorem; proof_source)

PolynomialParameterChart

SHA-256: `77fa77bfce3b270fe3388ffc70e3567caac9d9bea62baf70fe4cc1d604c01559`; 15 rows.

Namespace: `Komorebi.PolynomialParameterChart`

`chartGraph_iff` (theorem; proof_source)
`chartPhi` (def; proof_source)
`chartPoint` (def; proof_source)
`chartPoint_congruent` (theorem; proof_source)
`chartPoint_congruent_iff` (theorem; proof_source)
`chartPoint_free` (theorem; proof_source)
`chartPoint_graph` (theorem; proof_source)
`chartPoint_root` (theorem; proof_source)
`chartPoint_spec` (theorem; proof_source)
`chartPoint_unique` (theorem; proof_source)
`chartTarget` (def; proof_source)
`continuous_chartPhi` (theorem; proof_source)
`continuous_chartPoint` (theorem; proof_source)
`continuous_chartTarget` (theorem; proof_source)
`parameterHomeomorph` (def; proof_source)

PolynomialParameterPath

SHA-256: `8ce7df0bc47d4207bd6a6ad1f19afb82343efeb9573de8c415d16f6d92ec39ff`; 6 rows.

Namespace: `Komorebi.PolynomialParameterPath`

`exists_parameter_path` (theorem; proof_source)
`parameterPath` (def; proof_source)
`parameterPath_compatible` (theorem; proof_source)
`parameterPath_precision` (theorem; proof_source)
`parameterPath_tendsto` (theorem; proof_source)
`parameterPath_zero` (theorem; proof_source)

PolynomialParameterSystem

SHA-256: `ec8fc723a7fd5a6c1e945c7dab27a20c4a5e42e6db5ab7fe0fb981b2f2e4ee70`; 21 rows.

Namespace: `Komorebi.PolynomialParameterSystem`

`augmented` (def; proof_source)
`augmented_free` (theorem; proof_source)
`augmented_jacobian_surjective` (theorem; proof_source)

`augmented_solved` (theorem; proof_source)
`continuous_freePart` (theorem; proof_source)
`continuous_join` (theorem; proof_source)
`continuous_solvedPart` (theorem; proof_source)
`digitJacobian_augmented` (theorem; proof_source)
`digitJacobian_join_zero` (theorem; proof_source)
`evalSystem_augmented` (theorem; proof_source)
`freePart` (def; proof_source)
`freePart_join` (theorem; proof_source)
`join` (def; proof_source)
`join_free` (theorem; proof_source)
`join_parts` (theorem; proof_source)
`join_solved` (theorem; proof_source)
`padicEval_augmented` (theorem; proof_source)
`padicEval_augmented_iff` (theorem; proof_source)
`solvedJacobian` (def; proof_source)
`solvedPart` (def; proof_source)
`solvedPart_join` (theorem; proof_source)

PolynomialPrescribedCounts

SHA-256: `a7db903a0ba00e05d84059639c78df0625635a946b603775e23768fafbad903c`; 8 rows.

Namespace: `Komorebi.PolynomialPrescribedCounts`

`digitAddTorsor` (def; proof_source)
`digit_card` (theorem; proof_source)
`liftAddTorsor` (def; proof_source)
`liftEquivKernel` (def; proof_source)
`lift_card` (theorem; proof_source)
`lift_card_eq_kernel` (theorem; proof_source)
`obstruction_card` (theorem; proof_source)
`obstruction_finrank` (theorem; proof_source)

PolynomialPrescribedIntegerTower

SHA-256: `d0a11598d31484d7e5ca7a34b26b7bdfd7b71de52691b2dafbc213f52f1bdbe5`; 3 rows.

Namespace: `Komorebi.PolynomialPrescribedIntegerTower`

`exists_prescribed_tower` (theorem; proof_source)
`exists_prescribed_tower_strong` (theorem; proof_source)
`tower_solvedJacobian_eq` (theorem; proof_source)

PolynomialPrescribedLifts

SHA-256: af964eafecd3d84148aba6baa5c7ade302051f97bff40b09645ddd9b2309dc6b; 6 rows.

Namespace: Komorebi.PolynomialPrescribedLifts

PrescribedLifts (abbrev; proof_source)

freePart_digitResidue (theorem; proof_source)

prescribedDigitEquivLifts (def; proof_source)

prescribedDigitEquivLifts_apply (theorem; proof_source)

prescribedDigitEquivRestricted (def; proof_source)

restrictedDigitEquivLifts (def; proof_source)

PolynomialPrescribedObstruction

SHA-256: 4e1a7312fe5dbfd49762bf54022568e3321224111b0314b09d6f754eaabde140; 5 rows.

Namespace: Komorebi.PolynomialPrescribedObstruction

finite_prescribed_lift_count (theorem; proof_source)

finite_prescribed_obstruction (theorem; proof_source)

prescribed_lift_iff_obstruction_zero (theorem; proof_source)

prescribed_obstruction_finite (theorem; proof_source)

prescribed_residue_exists_iff (theorem; proof_source)

PolynomialPrescribedPadicTower

SHA-256: ccab254952dd2648be03304011fec384c8eb479b468da8c36bebcc9a16e6882f; 4 rows.

Namespace: Komorebi.PolynomialPrescribedPadicTower

exists_prescribed_padic_root (theorem; proof_source)

exists_prescribed_padic_root_with_tower (theorem; proof_source)

exists_root_with_prescribed_path (theorem; proof_source)

exists_unique_prescribed_padic_root (theorem; proof_source)

PolynomialPrescribedStep

SHA-256: 248587ef2bb1663331fc0163385df5144400bae44c274fa0f3d6b10c730ed34d; 7 rows.

Namespace: Komorebi.PolynomialPrescribedStep

obstruction_zero (theorem; proof_source)

one_step_lift (theorem; proof_source)

prescribed_lift_card (theorem; proof_source)

solvedJacobian_eq (theorem; proof_source)

solvedJacobian_shift_eq (theorem; proof_source)

solved_dimension (theorem; proof_source)

solved_rank (theorem; proof_source)

PolynomialPrescribedSystem

SHA-256: `87d8d3542994ae04a705ea6ba282ce00adda6d64fc1c61573c0b7d3592fb9978`; 10 rows.

Namespace: `Komorebi.PolynomialPrescribedSystem`

`PrescribedDigits` (abbrev; proof_source)

`PrescribedObstruction` (abbrev; proof_source)

`digitJacobian_join` (theorem; proof_source)

`obstruction` (def; proof_source)

`parameterJacobian` (def; proof_source)

`parameterResidual` (def; proof_source)

`prescribed_digits_finite` (theorem; proof_source)

`prescribed_digits_nonempty_iff` (theorem; proof_source)

`prescribed_equation_iff` (theorem; proof_source)

`solvedJacobian` (def; proof_source)

PolynomialPrescribedTowerLimit

SHA-256: `4559f09410a4d76900bf4670747b83c8dbd5648a7f080856b8014d885c5e81d2`; 1 rows.

Namespace: `Komorebi.PolynomialPrescribedTowerLimit`

`prescribed_tower_limit` (theorem; proof_source)

PolynomialPrescribedTowerUnique

SHA-256: `0c060ec7a92f7ca84bb1970fc1e85340fc64f438813fde33a054d215a98399c1`; 4 rows.

Namespace: `Komorebi.PolynomialPrescribedTowerUnique`

`prescribed_root_eq_chartPoint` (theorem; proof_source)

`prescribed_root_unique` (theorem; proof_source)

`prescribed_tower_endpoints_unique` (theorem; proof_source)

`solvedJacobian_square` (theorem; proof_source)

PolynomialRegularityIntegral

SHA-256: `ef886787a4cbb17a992c554f1ce6258abdc17ac045f45c02cfb0fe66f4d35c6d`; 11 rows.

Namespace: `Komorebi.PolynomialRegularityIntegral`

`eval_pderiv_rationalSystem` (theorem; proof_source)

`eval_rationalSystem` (theorem; proof_source)

`exists_analytic_integral_implicit` (theorem; proof_source)

`parameterJacobian_rationalSystem` (theorem; proof_source)

`rationalPoint` (def; proof_source)

`rationalPoint_freePart` (theorem; proof_source)

`rationalPoint_join` (theorem; proof_source)
`rationalPoint_solvedPart` (theorem; proof_source)
`rationalSystem` (def; proof_source)
`rationalSystem_root` (theorem; proof_source)
`solvedJacobian_rationalSystem` (theorem; proof_source)

PolynomialRootMotion

SHA-256: `76a6b1a6cdeab1306809881a58eccb6885911c7f2dbb4dd10b92e874da4b367a`; 4 rows.

Namespace: `Komorebi.PolynomialRootMotion`

`exists_perturbed_root_certificate` (theorem; proof_source)
`exists_root_from_original_preconditioner` (theorem; proof_source)
`exists_root_motion` (theorem; proof_source)
`exists_unique_root_motion` (theorem; proof_source)

PolynomialRootStability

SHA-256: `77eb7db2cc0e93d3f57f0248cf7e19f30932416441caa369f625f15a2869a2db`; 5 rows.

Namespace: `Komorebi.PolynomialRootStability`

`exists_nearby_perturbed_root` (theorem; proof_source)
`exists_root_and_unique_perturbation` (theorem; proof_source)
`exists_unique_nearby_perturbed_root` (theorem; proof_source)
`integer_congruent_of_common_padic` (theorem; proof_source)
`perturbed_roots_congruent` (theorem; proof_source)

PolynomialSmoothCounts

SHA-256: `bb4d79e92006e03f8bff12b3faa7eb5e87d9b0a25b5a7cf5265f16cfd670fd35`; 5 rows.

Namespace: `Komorebi.PolynomialSmoothCounts`

`exists_regular_tower` (theorem; proof_source)
`smooth_dimension` (theorem; proof_source)
`smooth_lift_card` (theorem; proof_source)
`smooth_obstruction_card` (theorem; proof_source)
`smooth_rank` (theorem; proof_source)

PolynomialSurjectiveStep

SHA-256: `d66d6635f63ca82be65e2a86aa55c2c4dc486796e6a3a1222fa039be162e6417`; 3 rows.

Namespace: `Komorebi.PolynomialSurjectiveStep`

`digitJacobian_eq` (theorem; proof_source)

`digitJacobian_shift_eq` (theorem; proof_source)

`one_step_lift` (theorem; proof_source)

PolynomialTaylor

SHA-256: `192667cea6c00131792d164ef6d802a689d4a34f8c465235a42d9319c57e699b`; 9 rows.

Namespace: `Komorebi.PolynomialTaylor`

`directional` (def; proof_source)

`directional_C` (theorem; proof_source)

`directional_X` (theorem; proof_source)

`directional_add` (theorem; proof_source)

`directional_mul` (theorem; proof_source)

`eval_dvd_of_dvd` (theorem; proof_source)

`pderiv_eval_dvd_of_dvd` (theorem; proof_source)

`taylor_dvd` (theorem; proof_source)

`taylor_step_dvd` (theorem; proof_source)

PolynomialTaylorDisplacement

SHA-256: `a283ae6bbcfb681a386165e562130765d5601d6c02a200010040174711e01e9d`; 1 rows.

Namespace: `Komorebi.PolynomialTaylorDisplacement`

`taylor_remainder_dvd` (theorem; proof_source)

PolynomialTaylorRing

SHA-256: `344502b0ee6d50e602ae1a23a2b6685c3992a08a6366c029b4e64d2c1f9d1c32`; 11 rows.

Namespace: `Komorebi.PolynomialTaylorRing`

`eval_dvd_of_dvd` (theorem; proof_source)

`eval_linearPart` (theorem; proof_source)

`eval_scaled_bind` (theorem; proof_source)

`eval_zero_linearPart` (theorem; proof_source)

`exists_eval_scaled_remainder` (theorem; proof_source)

`exists_scaled_remainder` (theorem; proof_source)

`linearPart` (def; proof_source)

`linearPart_C` (theorem; proof_source)

`linearPart_X` (theorem; proof_source)

`linearPart_add` (theorem; proof_source)

`linearPart_mul` (theorem; proof_source)

PolynomialTowerControls

SHA-256: `369da958792083c935d91b72f781b3a1ee624c50bfddb7f2500404248aedbb2b`; 10 rows.

Namespace: `Komorebi.PolynomialTowerControls`

`linearMinusTwo` (def; positive_control)

`linear_shift_no_extra_precision` (theorem; positive_control)

`linear_shift_nonsingular_lower_root` (theorem; positive_control)

`projectionSystem` (def; positive_control)

`projection_jacobian_surjective` (theorem; positive_control)

`projection_two_padic_roots` (theorem; positive_control)

`squareMinusFive` (def; positive_control)

`square_minus_five_lower_steps` (theorem; positive_control)

`square_minus_five_no_mod_eight` (theorem; positive_control)

`square_minus_five_no_tower` (theorem; positive_control)

PolynomialWeightedInverse

SHA-256: `f809cd360810e51a7179d27080643bf3827066cb9ddc8a34deaaed74e814c1b8`; 17 rows.

Namespace: `Komorebi.PolynomialWeightedInverse`

`bounded_inverse_iff_certificate` (theorem; proof_source)

`certificate_of_integral_inverse` (theorem; proof_source)

`eval_precondition` (theorem; proof_source)

`integral_inverse_iff_certificate` (theorem; proof_source)

`integral_inverse_of_certificate` (theorem; proof_source)

`jacobian` (def; proof_source)

`jacobian_precondition` (theorem; proof_source)

`mulVec_injective_of_scalar_certificate` (theorem; proof_source)

`mulVec_preserves_dvd` (theorem; proof_source)

`padic_mulVec_injective` (theorem; proof_source)

`precondition` (def; proof_source)

`precondition_root_iff` (theorem; proof_source)

`rationalMatrix` (def; proof_source)

`rationalMatrix_injective` (theorem; proof_source)

`rationalMatrix_isUnit_of_certificate` (theorem; proof_source)

`rationalMatrix_mul` (theorem; proof_source)

`rationalMatrix_scalar` (theorem; proof_source)

PolynomialWeightedLift

SHA-256: `8b30f4b2015e6f12cabdb123972ef5b40378034c5c3b8f7947336a2307148845`; 5 rows.

Namespace: `Komorebi.PolynomialWeightedLift`

`exists_root_of_normalization` (theorem; proof_source)

`exists_unique_weighted_root` (theorem; proof_source)

`exists_weighted_root` (theorem; proof_source)
`exists_weighted_root_certificate` (theorem; proof_source)
`shiftedPoint` (def; proof_source)

PolynomialWeightedNormalization

SHA-256: `3b7ac03ae8c15ff0150f4cc98fc5ced67f285e16ec659400dcddec5585614411`; 2 rows.

Namespace: `Komorebi.PolynomialWeightedNormalization`

`exists_normalization` (theorem; proof_source)
`precondition_directional` (theorem; proof_source)

ResidueLiftDigits

SHA-256: `378dadff08814a05565baea228042017e58753f5a490d398655d4c95df8fb6d8`; 10 rows.

Namespace: `Komorebi.ResidueLiftDigits`

`ResidueFiber` (abbrev; proof_source)
`digitEquivResidue` (def; proof_source)
`digitEquivResidue_apply` (theorem; proof_source)
`digitResidue` (def; proof_source)
`digitResidue_injective` (theorem; proof_source)
`digitResidue_mem` (theorem; proof_source)
`digitResidue_surjective` (theorem; proof_source)
`next_modulus_dvd_val_sub_shift` (theorem; proof_source)
`residueFiber_card` (theorem; proof_source)
`shift_cast_eq_iff` (theorem; proof_source)

PadicTowerLimitContract

SHA-256: `6d81199904dec30c4e6d0ce263e7a69ae9a2bae7ffa70b4ab86ef5a6ad5a5c1e`; 1 rows.

Namespace: `Komorebi.PadicTowerLimitContract`

`explicit_vector_limit` (theorem; statement_contract)

PolynomialPadicLiftContract

SHA-256: `acb5f14247e13111819e76a510950c6b5ab122cd4b01c570e662c6ec64f0133d`; 2 rows.

Namespace: `Komorebi.PolynomialPadicLiftContract`

`explicit_padic_root` (theorem; statement_contract)
`explicit_padic_root_with_tower` (theorem; statement_contract)

PolynomialTowerContract

SHA-256: `b53f89c3151e95d920e2ad8e06c2e4bb11313f6ad113e61b905097980517421e`; 2 rows.

Namespace: `Komorebi.PolynomialTowerContract`

`explicit_jacobian` (theorem; statement_contract)

`explicit_polynomial_tower` (theorem; statement_contract)

SharedCorrectionContract

SHA-256: `4aec4fc0ee27d21e0c98de6ebf66e2c331a278347a95b60c7988ec3e75eca89f`; 1 rows.

Namespace: `Komorebi.SharedCorrectionContract`

`explicit_quotient_contract` (theorem; statement_contract)

AffinePairControls

SHA-256: `ef7312826db247cc709fbf0fde10b49fc014b6cbec97addf9421789b8787533e`; 5 rows.

Namespace: `Komorebi.AffinePairControls`

`composite_level_zero` (theorem; positive_control)

`incompatible_mod_four` (theorem; positive_control)

`modulus_one` (theorem; positive_control)

`oneId` (abbrev; positive_control)

`two_individual_lifts` (theorem; positive_control)

SharedCorrectionControls

SHA-256: `6c08654acf2c8b51863630c97618d779dd953084db61e3d9cf05b7cce9b2caac`; 11 rows.

Namespace: `Komorebi.SharedCorrectionControls`

`Two` (abbrev; positive_control)

`evaluateCoordinate` (def; positive_control)

`identical_maps_distinguish_targets` (theorem; positive_control)

`identity` (abbrev; positive_control)

`independent_joint_range_top` (theorem; positive_control)

`independent_targets_have_equal_labels` (theorem; positive_control)

`leftCoordinate` (abbrev; positive_control)

`rightCoordinate` (abbrev; positive_control)

`separately_solvable_but_incompatible` (theorem; positive_control)

`sequence_domain_common_finite` (theorem; positive_control)

`zero_left_identifies_all` (theorem; positive_control)

AllResidueStepAmplification

SHA-256: `a16c419ab66d69068d54e3e47751c4684e21febd90232601e1c72c37e4177e2b`; 3 rows.

Namespace: `KomorebiPaperIX.AllResidueStepAmplification`

`normalized_precision` (theorem; proof_source)

`unnormalized_bound` (theorem; proof_source)

`unnormalized_precision` (theorem; proof_source)

AllRowsControls

SHA-256: `5c0b80c3b45af5cd396e18b76235aa069f5ce7aaf4fa638fd428636955fe2031`; 3 rows.

Namespace: `KomorebiPaperIX.AllRowsControls`

`zero_step_every_order` (theorem; positive_control)

`zeroth_row_every_order` (theorem; positive_control)

`zeroth_row_every_step` (theorem; positive_control)

ChargedDworkClosure

SHA-256: `d1db5c67f6ac3e73ca1914e7b6b394082aa01c2575a9ab3714b7fd7cc997831e`; 24 rows.

Namespace: `KomorebiPaperIX.ChargedDworkClosure`

`B` (def; proof_source)

`B_charge_shift` (theorem; proof_source)

`B_deriv` (theorem; proof_source)

`B_mul` (theorem; proof_source)

`B_zero_one` (theorem; proof_source)

`bounded` (def; proof_source)

`bounded_add` (theorem; proof_source)

`bounded_charge_shift` (theorem; proof_source)

`bounded_deriv` (theorem; proof_source)

`bounded_inverse` (theorem; proof_source)

`bounded_middle` (theorem; proof_source)

`bounded_mono` (theorem; proof_source)

`bounded_mul` (theorem; proof_source)

`bounded_nat_mul` (theorem; proof_source)

`bounded_neg` (theorem; proof_source)

`bounded_sub` (theorem; proof_source)

`bounded_succ` (theorem; proof_source)

`bounded_sum` (theorem; proof_source)

`bounded_zero` (theorem; proof_source)

`flat` (def; proof_source)

`flat_charge_shift` (theorem; proof_source)

`flat_inverse` (theorem; proof_source)
`inverse_defect` (theorem; proof_source)
`iterate_B_deriv` (theorem; proof_source)

CompactSelfIsometry

SHA-256: `3a81b435e899be582f1bcec59c561de1c57b7d7a7c6150a3eeafbe2ca8723c44`; 3 rows.

Namespace: `KomorebiPaperIX.CompactSelfIsometry`

`denseRange` (theorem; proof_source)
`iterate_isometry` (theorem; proof_source)
`surjective` (theorem; proof_source)

DensePadicIsometry

SHA-256: `5ff66895f2ef98dad15abe1af104aa00285c05466d87ba384f7f5af795fbc136`; 15 rows.

Namespace: `KomorebiPaperIX.DensePadicIsometry`

`existsUnique_extensionEquiv` (theorem; proof_source)
`extension` (def; proof_source)
`extensionEquiv` (def; proof_source)
`extensionEquiv_apply` (theorem; proof_source)
`extensionEquiv_nat` (theorem; proof_source)
`extensionEquiv_unique` (theorem; proof_source)
`extensionEquiv_unique_continuous` (theorem; proof_source)
`extension_continuous` (theorem; proof_source)
`extension_isometry` (theorem; proof_source)
`extension_nat` (theorem; proof_source)
`extension_surjective` (theorem; proof_source)
`extension_unique` (theorem; proof_source)
`onNaturalRange` (def; proof_source)
`onNaturalRange_isometry` (theorem; proof_source)
`onNaturalRange_nat` (theorem; proof_source)

DyadicFourAllRows

SHA-256: `e90316dce8f7976fed0333f72fbea9043542c0b6ea188f9acfa8a6084cffa3a4`; 18 rows.

Namespace: `KomorebiPaperIX.DyadicFourAllRows`

`B` (abbrev; proof_source)
`CoeffFlat` (def; proof_source)
`Flat` (def; proof_source)
`actual_packet_flat` (theorem; proof_source)
`base_step_bound` (theorem; proof_source)

`decode_B` (theorem; proof_source)
`decode_B_iter` (theorem; proof_source)
`difference_coe` (theorem; proof_source)
`flat_encode_iff` (theorem; proof_source)
`flat_reciprocal` (theorem; proof_source)
`lifted_flat` (theorem; proof_source)
`normalized_bound` (theorem; proof_source)
`normalized_precision` (theorem; proof_source)
`packed_difference_support` (theorem; proof_source)
`packed_flat_of_kummer` (theorem; proof_source)
`source_flat` (theorem; proof_source)
`unnormalized_bound` (theorem; proof_source)
`unnormalized_precision` (theorem; proof_source)

DyadicFourFinalStrengthening

SHA-256: `23e7b814ac843f882fc99c0429bdac6b4542f041a43cb0ba8e92b601a2dcb179`; 11 rows.

Namespace: `KomorebiPaperIX.DyadicFourFinalStrengthening`

`all_rows_half_period` (theorem; proof_source)
`all_rows_half_period_degree` (theorem; proof_source)
`all_rows_half_period_precision` (theorem; proof_source)
`all_rows_period` (theorem; proof_source)
`all_rows_period_degree` (theorem; proof_source)
`all_rows_period_precision` (theorem; proof_source)
`normalized_period` (theorem; proof_source)
`normalized_step_bound` (theorem; proof_source)
`normalized_step_precision` (theorem; proof_source)
`unnormalized_step_bound` (theorem; proof_source)
`unnormalized_step_precision` (theorem; proof_source)

DyadicFourIsometry

SHA-256: `22ee30fdfe665602b491ef9ff43b3aa8fe2bacfc503724e19e6623b227ef4da8`; 19 rows.

Namespace: `KomorebiPaperIX.DyadicFourIsometry`

`existsUnique_isometry` (theorem; proof_source)
`isometry` (def; proof_source)
`isometry_continuous` (theorem; proof_source)
`isometry_dist` (theorem; proof_source)
`isometry_nat` (theorem; proof_source)
`isometry_surjective` (theorem; proof_source)
`isometry_zero` (theorem; proof_source)
`originalExtension` (def; proof_source)

`originalExtension_continuous` (theorem; proof_source)
`originalExtension_dist` (theorem; proof_source)
`originalExtension_injective` (theorem; proof_source)
`originalExtension_nat` (theorem; proof_source)
`originalExtension_range` (theorem; proof_source)
`samples` (def; proof_source)
`samples_coe` (theorem; proof_source)
`samples_dist` (theorem; proof_source)
`unique_continuous` (theorem; proof_source)
`unique_original_preimage` (theorem; proof_source)
`unique_preimage` (theorem; proof_source)

DyadicFourLeastPeriods

SHA-256: `1073c2debcd4f11ac2478b89549c18682cc05683c078d9378d86991da23d7286`; 8 rows.

Namespace: `KomorebiPaperIX.DyadicFourLeastPeriods`

`exact_displacement` (theorem; proof_source)
`least_packet_period` (theorem; proof_source)
`normalized_exact_displacement` (theorem; proof_source)
`originalBonus` (def; proof_source)
`originalBonus_values` (theorem; proof_source)
`original_degree_exact_displacement` (theorem; proof_source)
`period_iff_dvd` (theorem; proof_source)
`precision_iff_dvd` (theorem; proof_source)

DyadicFourPrefix

SHA-256: `3fc459e037e333e6c3680753ac591502061ef0b2e2a83073f5f7892335ef5ca2`; 49 rows.

Namespace: `KomorebiPaperIX.DyadicFourPrefix`

`bonus` (def; proof_source)
`coeff_0_1` (theorem; proof_source)
`coeff_0_2` (theorem; proof_source)
`coeff_1_1` (theorem; proof_source)
`coeff_1_2` (theorem; proof_source)
`coeff_2_1` (theorem; proof_source)
`coeff_2_2` (theorem; proof_source)
`coeff_2_3` (theorem; proof_source)
`coeff_2_4` (theorem; proof_source)
`coeff_2_5` (theorem; proof_source)
`coeff_2_6` (theorem; proof_source)
`coeff_3_1` (theorem; proof_source)
`coeff_3_2` (theorem; proof_source)

`coeff_3_3` (theorem; proof_source)
`coeff_3_4` (theorem; proof_source)
`first_nonzero` (theorem; proof_source)
`first_valuation` (theorem; proof_source)
`mahlerCoeff` (def; proof_source)
`mahler_0_1` (theorem; proof_source)
`mahler_0_2` (theorem; proof_source)
`mahler_1_1` (theorem; proof_source)
`mahler_1_2` (theorem; proof_source)
`mahler_2_1` (theorem; proof_source)
`mahler_2_2` (theorem; proof_source)
`mahler_2_3` (theorem; proof_source)
`mahler_2_4` (theorem; proof_source)
`mahler_2_5` (theorem; proof_source)
`mahler_2_6` (theorem; proof_source)
`mahler_3_1` (theorem; proof_source)
`mahler_3_2` (theorem; proof_source)
`mahler_3_3` (theorem; proof_source)
`mahler_3_4` (theorem; proof_source)
`mahler_valuation_0_1` (theorem; proof_source)
`mahler_valuation_0_2` (theorem; proof_source)
`mahler_valuation_1_1` (theorem; proof_source)
`mahler_valuation_1_2` (theorem; proof_source)
`mahler_valuation_2_1` (theorem; proof_source)
`mahler_valuation_2_2` (theorem; proof_source)
`mahler_valuation_2_3` (theorem; proof_source)
`mahler_valuation_2_4` (theorem; proof_source)
`mahler_valuation_2_5` (theorem; proof_source)
`mahler_valuation_2_6` (theorem; proof_source)
`mahler_valuation_3_1` (theorem; proof_source)
`mahler_valuation_3_2` (theorem; proof_source)
`mahler_valuation_3_3` (theorem; proof_source)
`mahler_valuation_3_4` (theorem; proof_source)
`precision_of_order_le` (theorem; proof_source)
`tail_precision` (theorem; proof_source)
`valuation_two_power_odd` (theorem; proof_source)

DyadicFourSharpening

SHA-256: `eda98f64f8525d210569db34177a5795f9745939dc59d9cd98887eac52df57ac`; 9 rows.

Namespace: `KomorebiPaperIX.DyadicFourSharpening`

`all_rows_even` (theorem; proof_source)
`factorial_difference_bound` (theorem; proof_source)

`factorial_difference_precision` (theorem; proof_source)
`factorial_even` (theorem; proof_source)
`half_period` (theorem; proof_source)
`half_period_degree` (theorem; proof_source)
`half_period_precision` (theorem; proof_source)
`normalized_period` (theorem; proof_source)
`unnormalized_difference_factorial` (theorem; proof_source)

DyadicIsometryControls

SHA-256: `b4f813406c69b1d9c37766c388be658d82e900f2efdccc864a5d41458b6625b3`; 6 rows.

Namespace: `KomorebiPaperIX.DyadicIsometryControls`

`address_zero_at_one` (theorem; positive_control)
`address_zero_at_two` (theorem; positive_control)
`address_zero_inverse_at_thirty_four` (theorem; positive_control)
`address_zero_not_additive` (theorem; positive_control)
`original_address_zero_ne_zero` (theorem; positive_control)
`original_address_zero_not_surjective` (theorem; positive_control)

DyadicLogThresholds

SHA-256: `3c5230cbad55877095ab8a5b45cb99082ed724c1b0720f90d77fc5f9d4917eae`; 5 rows.

Namespace: `KomorebiPaperIX.DyadicLogThresholds`

`five_add_log_le` (theorem; proof_source)
`log_two_succ_le` (theorem; proof_source)
`offset_log_bound_of_le` (theorem; proof_source)
`three_add_log_le` (theorem; proof_source)
`two_add_log_le` (theorem; proof_source)

DyadicPacketRational

SHA-256: `7a29429898faaa4d9d0d8d9ee931be6ba38644b061e967d6c36a69a3d49dc9e1`; 7 rows.

Namespace: `KomorebiPaperIX.DyadicPacketRational`

`exact_displacement` (theorem; proof_source)
`norm_displacement` (theorem; proof_source)
`norm_le_one` (theorem; proof_source)
`norm_sub` (theorem; proof_source)
`packet` (def; proof_source)
`packet_sub` (theorem; proof_source)
`packet_zero` (theorem; proof_source)

DyadicSharpTower

SHA-256: `70f43e0621368af3bb3871e102deb238e68fb610fd73d81cae1486d7a95c3a2b`; 27 rows.

Namespace: `KomorebiPaperIX.DyadicSharpTower`

`bernoulli_dyadic_valuation` (theorem; proof_source)

`bernoulli_even_nonzero` (theorem; proof_source)

`beta_bernoulli` (theorem; proof_source)

`cosh_derivative` (theorem; proof_source)

`cothNormalized` (def; proof_source)

`coth_times_sinh` (theorem; proof_source)

`dyadic_supercongruence_row_weight` (theorem; proof_source)

`dyadic_supercongruence_sharp` (theorem; proof_source)

`expAt_derivative` (theorem; proof_source)

`first_omitted_error` (theorem; proof_source)

`first_omitted_error_valuation` (theorem; proof_source)

`hyperbolic_identity` (theorem; proof_source)

`inverse_eq_coth_derivative` (theorem; proof_source)

`kappa` (def; proof_source)

`kappaPadic` (def; proof_source)

`kappaPadic_coe` (theorem; proof_source)

`kappaPadic_divisible` (theorem; proof_source)

`kappaPadic_norm` (theorem; proof_source)

`kappa_bernoulli` (theorem; proof_source)

`kappa_nonzero` (theorem; proof_source)

`kappa_valuation` (theorem; proof_source)

`odd_factor_valuation` (theorem; proof_source)

`sinh_derivative` (theorem; proof_source)

`summandPadic_divisible` (theorem; proof_source)

`summandPadic_kappa` (theorem; proof_source)

`summand_kappa` (theorem; proof_source)

`summand_valuation` (theorem; proof_source)

DyadicStepAmplification

SHA-256: `e5d7937d20e2664578d066e4df1c843190da3018a2677ccd5dd9f72e117f1eab`; 5 rows.

Namespace: `KomorebiPaperIX.DyadicStepAmplification`

`normalized_bound` (theorem; proof_source)

`normalized_precision` (theorem; proof_source)

`prime_power_step_bound` (theorem; proof_source)

`unnormalized_factorial_bound` (theorem; proof_source)

`unnormalized_factorial_precision` (theorem; proof_source)

ExactPeriodClassification

SHA-256: `86fb5cc83b77e3889d004503916aaa12c981d9dc6e9566c544a0136664856eb5`; 4 rows.

Namespace: `KomorebiPaperIX.ExactPeriodClassification`

`IsPeriod` (def; proof_source)

`isPeriod_iff_dvd` (theorem; proof_source)

`least_period` (theorem; proof_source)

`precision_iff_dvd` (theorem; proof_source)

ExactPeriodControls

SHA-256: `13237d0c0b681c160c1c1dcf4e1158e156ad3f28378d800dc68196f54ca6f3f4`; 6 rows.

Namespace: `KomorebiPaperIX.ExactPeriodControls`

`address_three_mod1024_least` (theorem; positive_control)

`address_two_mod1024_period96` (theorem; positive_control)

`address_two_mod32` (theorem; positive_control)

`address_two_mod64_least` (theorem; positive_control)

`address_two_period_one_fails_mod64` (theorem; positive_control)

`positive_shift_nonzero` (theorem; positive_control)

NewtonDominance

SHA-256: `57973ae0db7d02891a2d2048d5b68530252d47a354cb7569edb4922f7b73da05`; 11 rows.

Namespace: `KomorebiPaperIX.NewtonDominance`

`choose_precision` (lemma; proof_source)

`displacement_expansion` (lemma; proof_source)

`exact_displacement` (theorem; proof_source)

`first_difference_tail` (lemma; proof_source)

`precision_add` (lemma; proof_source)

`precision_mono` (lemma; proof_source)

`precision_mul` (lemma; proof_source)

`precision_mul_nat` (lemma; proof_source)

`precision_sum` (lemma; proof_source)

`strict_dominance` (lemma; proof_source)

`translated_tail` (lemma; proof_source)

RationalPadicEmbedding

SHA-256: `234880a6a053a143265ab9b474f489ff7f4107356135b7038518b757d617f6dc`; 4 rows.

Namespace: `KomorebiPaperIX.RationalPadicEmbedding`

`norm_le_one` (theorem; proof_source)
`samples` (def; proof_source)
`samples_coe` (theorem; proof_source)
`samples_dist` (theorem; proof_source)

StepAmplification

SHA-256: `c4f9ecbcf2cffa821a25db744f8b52e2cb88c98cbafd89014f266f5286b47900`; 5 rows.

Namespace: `KomorebiPaperIX.StepAmplification`

`binomial_step_precision` (theorem; proof_source)
`difference_compose` (theorem; proof_source)
`offset_displacement` (theorem; proof_source)
`repeated_offset` (theorem; proof_source)
`repeated_precision` (theorem; proof_source)

StepAmplificationExamples

SHA-256: `286d99f1bbe2876a6339f0672daf49dd9d20b3f2d3cd57ac19def50c3bad68ac`; 4 rows.

Namespace: `KomorebiPaperIX.StepAmplificationExamples`

`actual_first_difference` (theorem; positive_control)
`actual_first_difference_valuation` (theorem; positive_control)
`actual_second_coefficient` (theorem; positive_control)
`no_uniform_extra_digit` (theorem; positive_control)

Universal Residual Seeds

The formal companion to Chapter X explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2351

Return to Chapter X

Abstract

This guide explains the source contracts behind Chapter X rather than treating its terminal signature as a stand-in for the mathematics. It provides a concordance of the twenty source records, follows the two independent construction branches, and distinguishes arithmetic occurrence, finite trace recovery, carrier presentations, local comparison data, and nonzero occupancy. Four complete production modules are printed from preserved source. Their verification scope is stated separately.

1 Formal verification and scope

The recorded compiler checks cover 122 project and support modules, including the 97-module terminal closure. Two clean builds agreed on 392 artifacts. The matching Lean toolchain and all nine dependency identities were checked. External Mathlib caches were reused. This is not a cold dependency rebuild.

Owned declarations and explicit targets. The fresh audit checked all 2,846 declarations owned by the 122 selected modules, including 2,127 theorems, 444 private declarations, and generated declarations. All 45 explicit target contracts resolved with the required kind and owner. Their complete elaborated types are retained. The union of transitive axioms was exactly `propext`, `Classical.choice`, and `Quot.sound`. No forbidden dependency was found. The source records give the checking procedure and its limits. The 18 compiler fixtures and 64 unit tests check tooling and are separate from these production counts.

Correspondence requires mathematical reading. The twenty-row register is a reading concordance. Its three incorrect source routes have been corrected to the actual residual-obstruction, reconstruction, and empty-level declarations. The independent bounded source review checked the amended terminal composition and finite-copy carrier closure under their explicit inputs. It is not a new proof audit of every historical theorem. Classical Inputs 3.1 and 4.1 and the conjectures remain outside the twenty-row concordance. Compiler acceptance and name resolution alone do not prove manuscript equivalence.

Finite checks have separate bounds. The current replay passed the original 28 companion tests and 13 reader regression methods, including 3,123 canonical residue towers through depth four over bases 2, 3, 5 and 7. The visual suite passed 263 positive assertions and rejected 14 false controls. Six further finite suites retain their actual bounds, seeds and logs separately. These counts are not combined into a theorem count.

The mathematical boundary is unchanged. The transported target family and recovered carrier summand are independent existential conclusions. Their identification requires another comparison. The literal constructed arithmetic separator is over `GQ`. Local comparisons, carrier presentations and source-family data remain inputs. Zero multiplicities are admitted. Nonzero occupancy needs its own premise. Universal occupancy and natural finite seed rank remain conjectures. The later stronger residual-system attempt remains separately preserved and unpromoted.

2 How to read the formal evidence

Chapter X has a finite and formal core, a substantial arithmetic development, and a terminal conditional assembly. These are not three names for one assertion. The source register below gives the twenty numbered proof records in the main manuscript. The checked endpoint concerns their recorded amended scope. Later arithmetic prose can still require hypotheses which an application has not supplied.

A Lean declaration proves its type, including implicit parameters and typeclass assumptions. A short proof can therefore depend on a long contract. Here the terminal proof is only six lines: it constructs a conjunction and invokes two previous theorems. The mathematical content is distributed across those theorems and their dependencies. Reading only the final six lines would miss the source family, all-depth naturality, inverse-limit closure, carrier presentations, and constituent local comparisons.

2.1 Three different records

Record	Interpretation
Base scalar/Mellin	Base 17/20: Claims 9.1, 9.2 and 11.1 are partial in that contract.
Conditional assembly	Recorded 20/20 under an explicitly amended contract. The target family and carrier summand are independent conclusions.
Stronger residual contract	19/20 under a stronger contract with an unproved terminal claim. This does not invalidate the conditional assembly.

The records distinguish the accepted conditional theorem from the incomplete attempt at a stronger conclusion. The latter neither replaces nor invalidates the former.

2.2 What each kind of check can establish

An exact Python calculation verifies the displayed finite instance or enumerated range. A source hash verifies identity of bytes. A successful typeset build verifies that the document can be generated. A saved compiler log reports a past compilation. None of these is a fresh proof of every theorem in the collection. A transitive axiom report also has a specific declaration set and environment. Its scope is not enlarged by binding it into a longer monograph.

3 The twenty source records

The numbered entries below are the main manuscript's twenty theorem, proposition, and corollary records. Classical Inputs 3.1 and 4.1 are not counted as new proof records. Definitions, examples, numerical certificates, and the conjecture that the seed rank is finite are also excluded from this count.

This is a reading concordance, not a claim that twenty lines of a table replace twenty theorem-to-source audits. The earlier seventeen entries retain their hypotheses. The final three arithmetic entries must be read through the amended source contract explained in Sections 7–11. Links target named destinations in the companion monograph. Keep the three PDFs in the same directory for relative file links to resolve.

Record	Statement and source route	Hypothesis to retain
2.1	Residual partition. `Residual`	A supplied, single-valued reduction map.
2.2	No crossing under positive-depth perturbation. `Residual`, `SeedConsequences`	Root-preserving operations. The infinite statement assumes convergence.
2.3	Zeroth-order obstruction. `Residual.zeroth_order_obstruction`	Different roots cannot be reached by genuine positive-depth operations.
3.2	Formal inverse-limit reconstruction. `Reconstruction.represented_local_algebra_reconstruction`	Represented complete local problems with compatible finite maps.
3.3	Finite-branch compactness. `Compactness`, `RepresentationForests`	Finite nonempty levels and the stated total reduction maps.
3.4	Smooth formal branch count. `SmoothMaps`, `IntegralCharts`	An actual power-series chart and coefficient-preserving local maps.
4.2	Cohomology packet decomposition. `ComplexCohomology`, `PacketCohomology`	A finite complete orthogonal family of chain projectors.
5.3	Weighted smooth-chart count. `Counting`, `FormalCharts`	Independent coordinates with the stated valuation lower bounds.
6.2	Smooth seed zeta function. `Zeta`, `SmoothModels`	The smooth-chart count, as an infinite formal power series.
6.3	Weighted seed zeta function. `Zeta`	The stated ordered positive weights, with finite initial levels separated.
6.4	Singular power seed. `Singular`, `SingularZeta`	The scalar equation x to the e equals zero at finite precision.
6.6	Fractional singular dimension. `BranchingDimension`, `Singular`	A finite-count growth exponent, not the dimension of a limit space.
7.3	Directed minimum-seed theorem. `DirectedSeeds`	A finite supplied graph. Arithmetic admissibility of edges is separate.
8.2	Adelic finite-depth completion. `ClassCommonSpine`, `SystemCompletion`	Common data, finite-depth nonemptiness, and inverse-limit-closed conditions.
8.3	Residual companion obstruction. `SystemCompletion.system_empty_level_obstruction`	An empty required finite level rules out the specified realization.
9.1	Finite Frobenius separation. `Claim91ConstructedClosure`	Later constructed rational-base separator, with its literal compatible-family inputs.
9.2	Compatible-carrier summand completion. `StrictCompatibleCarrierCompletion`	Explicit copy presentations, natural multiplicities within capacity, and supplied local data.

Record	Statement and source route	Hypothesis to retain
10.1	Exact finite occupancy trace. `SpectralProjectors`, `JointOccupancy`	Commuting split semisimple operators. The positive-trace interpretation uses characteristic zero.
11.1	Residual-system completion criterion. `ConditionalOccupiedSystemCompletion`	The later amended two-conjunct contract, not universal residual existence.
11.2	Exact location of the remaining existence problem. `SpineExistenceCriterion`, `RequiredPacketOccupancy`	All required finite levels and all supplied packet occupancy premises.

The monograph retains the general-base mathematical discussion. In contrast, the printed constructed arithmetic separator and the terminal assembly use `GQ`, the rational absolute Galois group. That literal base is part of the formal statement. An arbitrary-number-field version requires its own attachment and proof, not a change in a heading.

4 Residual roots and reconstruction

The fiber decomposition begins with a function, the reduction map. For an object x in a coefficient algebra A, its residual root is $\mathrm{red}_A(x)$. Distinct fibers are disjoint because a function has one value at each input. No topological or representation-theoretic theorem is needed for that partition.

The no-crossing theorem adds a hypothesis about the allowed operation. A positive-depth correction is trivial after residual reduction. A finite sequence of such corrections therefore stays in one fiber. An infinite sequence needs a convergence assumption. The sequence $0, 3, 0, 3, \ldots$ has one residue modulo three but does not converge modulo nine. Residual invariance is weaker than being Cauchy.

4.1 A point in an inverse limit is a coherent family

For a represented complete deformation problem, compatible finite maps determine one map into the inverse-limit coefficient ring. The argument does not assert that every finite object can be extended. Nor does it produce a map when the finite levels are empty. Reconstruction and existence remain distinct.

The finite-branch argument supplies existence when all levels are finite and nonempty and the transition maps are the stated reductions. Surjectivity of every transition map is unnecessary. Some nodes may die. The surviving branch is obtained from nodes with descendants at arbitrarily large depths. Finiteness prevents indefinitely moving the root to a new unrelated component.

Plate X.9 displays the infinite-level countermodel $X_m = \{n \geq m\}$ with inclusion maps. Each level is nonempty, but a compatible family would have to be one integer n lying in every X_m. No such integer exists. Its infinite levels violate the compactness hypotheses. This is a countermodel to dropping finiteness, not to the finite-level theorem.

4.2 The corrected digit formula

For compatible canonical residues $0 \leq x_m < p^m$, the digit in position $m-1$ is

$$a_{m-1} = \left\lfloor \frac{x_m}{p^{m-1}} \right\rfloor \pmod{p}.$$

Set $x_0 = 0$. Compatibility also makes

$$a_{m-1} = \frac{x_m - x_{m-1}}{p^{m-1}} \in \{0, \ldots, p-1\}.$$

The earlier algorithm display omitted the floor. The executable implementation already used integer division. For the base-five residues $4, 19$, the new digit is 3. Dividing 19 by 5 in the rationals does not recover a digit. The new regression checks compare the displayed formulas with the inherited executable on exhaustive bounded towers, including noncanonical integer representatives for the floor formula.

5 Counting finite points without inventing limit points

In a smooth chart $\mathcal{O}[[t_1, \dots, t_d]]$, local maps into $\mathcal{O}/\varpi^m$ send each parameter into the maximal ideal. There are q^{m-1} possibilities for each independent coordinate, giving $q^{d(m-1)}$ maps. The word independent is justified by the power-series presentation, not by the appearance of a grid in a figure.

Weighted coordinates with $t_i \in \varpi^{w_i}\mathcal{O}$ have $\max(m - w_i, 0)$ free residue digits. Thus

$$N_{\mathbf{w}}(m) = q^{\sum_i \max(m-w_i,0)}.$$

For weights $(1, 2, 4)$ over $\mathbb{Z}_2$, the counts begin $1, 2, 8, 32, 256, 2048$. The coordinate of weight four is not nontrivial at depth four. It first has a free digit at depth five. The weight and first visible depth are different quantities.

5.1 The singular count is not a count of infinite branches

For $x^e = 0$ modulo ϖ^m, the condition is $e\, v_{\varpi}(x) \geq m$. Its finite cardinality is $q^{m-\lceil m/e \rceil}$. Splitting $m = ek + r$, $1 \leq r \leq e$, gives

$$Z_e(T) = \frac{\sum_{r=1}^{e} q^{r-1} T^r}{1 - q^{e-1} T^e}.$$

This is an identity of formal power series. Its growth exponent $1 - 1/e$ measures the finite thickenings. It does not imply that the complete solution space has a fractional geometric dimension. In the integral domain $\mathcal{O}$, the only complete root of $x^e = 0$ is zero. Equivalently, the reduction at depth m of a root at depth em is zero. Every coherent infinite branch is therefore the zero branch.

The smooth, weighted, and singular cases teach different tests. A complete chart can have many genuine branches. A weighted chart delays their finite visibility. A singular thickening can have many finite roots and only one complete branch. The generating series must always be interpreted for the object it actually counts.

6 Cohomology packets and directed transport

A degreewise idempotent is not automatically a map of complexes. The required identity is $d_r P_j^r = P_j^{r+1} d_r$. Together with orthogonality and a sum equal to the identity, this makes the images into subcomplexes. The induced maps on cycles and boundaries then descend to cohomology. Their inverse identities give

$$H^r(C^\bullet) \simeq \bigoplus_j H^r(P_j C^\bullet).$$

The comparison with tangent or obstruction spaces of a particular deformation problem is an additional identification. The packet theorem itself is a theorem about a complex with supplied chain projectors.

Plates X.10 and X.11 use a complex over $\mathbb{F}_5$ with dimensions $(2, 4, 2)$. The valid packet family gives cohomology dimensions $(1, 2, 1)$. A modified idempotent deletes the image of a retained source vector and fails the chain-map identity. All its individual matrices remain idempotent. The missing condition is compatibility with the differential, not idempotence.

6.1 A transport graph is data before it is an optimization problem

For a finite directed graph, contract strongly connected components. The resulting graph is acyclic. Every generating seed set must meet each source component, since no other component can reach into that source. Conversely, one vertex from every source reaches all components. This proves the directed minimum-seed formula.

The theorem optimizes a supplied graph. An arithmetic twist, induction, or coefficient change is an edge only after its admissibility has been established. Adding an unjustified arrow can lower the computed seed rank without proving anything about the original arithmetic problem. Likewise, a finite graph theorem does not settle the conjectural finite seed rank of an unrestricted arithmetic transport category.

6.2 Transport must respect every reduction square

For roots of $x^2 + 1$ modulo powers of five, negation preserves the equation and commutes with every reduction. Translation by one maps the root two to the root three modulo five, but maps three to four, which is not a root. One successful pair of endpoints does not define a natural operation on the whole tower. The formal interface in the next section expresses exactly this distinction.

7 Horizontal completion, hypothesis by hypothesis

The module `HorizontalTransportCompletion` works with actual change-of-basis classes. Its structure `AdmissibleClassTransport` has three fields: a map at every precision, preservation of allowed classes, and compatibility with reduction. Those fields are the data of a map of inverse towers.

A completed source representation ρ first gives a coherent branch of its finite classes. Applying the transport at each depth gives a coherent target branch. The predecessor reconstruction theorem then supplies an actual target representation. This is stronger than mapping one residual root, but it is still conditional on the existence of the completed source representation.

7.1 The local profile is not hidden in the word reconstruction

`TransportedAlong` records, for every depth, equality between the reduced target class and the transported source class. An arbitrary semantic property of the complete representation does not follow from that equality alone. The theorem therefore accepts `InverseLimitClosed` for the selected target profile and finite-level conditions. Applying the closure theorem to the reconstructed target establishes that profile.

For example, nontriviality is not a closed condition under the relevant limit. The matrices

$$\rho_n(g) = \begin{pmatrix} 1 & 3^n g \\ 0 & 1 \end{pmatrix}$$

are nontrivial representations of the additive integers, while they converge pointwise to the trivial representation in the three-adic topology. A proof of nontriviality for every approximation is not a proof of nontriviality for its limit. This example does not decide the closedness of any particular arithmetic local condition. That condition needs its own theorem.

7.2 What changes when coefficient places are indexed

The indexed theorem takes a source member at every λ. It first proves an existence statement at each place, then uses choice to select a family of target members. The source code explicitly verifies three outputs at every place: finite-depth admissibility, the relation `TransportedAlong`, and membership in the target profile.

This is an indexed family statement. Any interpretation as a compatible arithmetic system depends on the shared data encoded by the conditions and profiles. Merely placing unrelated local objects in one function indexed by λ is not a new compatibility theorem. The abstract horizontal module is general in the source group, but the terminal assembly printed here specializes that group to `GQ`.

8 From arithmetic occurrence to a finite separator

The constructed Claim 9.1 module has two declarations. The first provides its `ExistentialArithmeticC` overage proposition by calling the constructed Chebotarev bridge. The second applies the arithmetic system-separation theorem to that proposition. In this later source there is no free external coverage parameter in the final separator signature. The scalar/Mellin interface retains that parameter. Both sources remain in the archive.

8.1 What the separator accepts

The family consists of finite-dimensional irreducible representations, compatible character values in a common field, continuous characters at each realization, and pairwise nonisomorphism in the stated memberwise sense. At each coefficient place the bad-prime set is finite. The caller selects a distinguished place and may forbid an additional finite set of rational primes.

The result selects distinct allowed rational primes. The common sample matrix has nonzero determinant, and that remains true after each coefficient-field embedding. Matrix entries identify with the representation characters wherever the required good-prime condition holds. The terminal wrapper then converts this matrix identity into the literal trace determinant needed by the carrier theorem.

Distinct primes alone would not suffice. Samples could repeat one character row and leave the matrix singular. Nonisomorphism, arithmetic occurrence, and the finite-dimensional separation argument do different work.

8.2 The finite character-table model

With columns trivial, sign, and standard, use

$$C = \begin{pmatrix} 1 & 1 & 2 \\ 1 & -1 & 0 \\ 1 & 1 & -1 \end{pmatrix}, \qquad \det C = 6, \qquad C \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}.$$

The table explains multiplicity recovery after separating samples exist. It does not produce rational primes with prescribed Frobenius behavior. The arithmetic occurrence construction is represented by the preserved source and its inherited verification record, not by this finite example.

8.3 Exact recovery can still lose finite precision

Over characteristic zero, C is invertible. Modulo eight, $\delta m = (4, 4, 0)$ is nonzero but $C\delta m = (8, 0, 8)$ vanishes. The denominator valuation consumes one binary digit. Traces modulo sixteen distinguish this particular ambiguity. The example separates invertibility over a field from invertibility over a residue ring. Plate X.17 gives the complete small model.

9 Carrier recovery is a construction inside supplied data

Let the supplied carrier be equivariantly presented as $\bigoplus_i W_i^{\oplus c_i}$, and let the chosen multiplicities satisfy $m_i \le c_i$. The copy inclusion and projection give maps j and r with $r \circ j = \mathrm{id}$. Transporting these through the carrier presentation produces an actual split summand. An arbitrary vector of traces does not provide that presentation.

The initial subrepresentation is also supplied with an equivariant presentation by the same selected copies. The nonsingular sample matrix proves that no other natural multiplicity vector has those sample traces. Thus the theorem constructs splittings and proves uniqueness within the declared constituent family. It does not discover an unknown irreducible decomposition from raw observations.

9.1 Three tests after solving a linear equation

For capacities $(2, 1, 1)$, the vector $(2, 0, 1)$ is admissible. The trace vector $(1, 0, 0)$ instead gives $(1/6, 1/6, 1/3)$, which is not integral. The vector $(-1, 1, 1)$ is integral but negative. The vector $(3, 0, 1)$ exceeds capacity. Each has a perfectly well-defined rational trace vector under the same matrix. A nonsingular system alone does not turn its solution into a representation.

9.2 What happens to local data

The source constructs finite copies of supplied local packets. Characteristic polynomials multiply with multiplicities. Relevant dimensions and traces add. The supplied period comparisons similarly produce the selected finite-copy comparison and weight-multiplicity formulas.

These identities do not attach Weil–Deligne or period data to an arbitrary new representation. The data and comparisons are input. An extra semantic condition is handled separately by `FiniteCopyClosed`. The terminal `CarrierRecoveryAt` packages specific polynomial, local, and period identities. It does not universally quantify over every possible local predicate. The separate `selected_local_condition` theorem applies only after its finite-copy-closure premise has been proved.

A fixed nontrivial determinant character illustrates the distinction. Two one-dimensional constituents can each have determinant χ, while their direct sum has determinant χ^2, not χ. A determinant target must be combined according to its actual product law. The word local does not make a condition stable under arbitrary sums.

10 Occupancy and the missing nonzero premise

The identity $\mathrm{Tr}(P) = \mathrm{rank}(P)$ for a projector over a characteristic-zero field is interpreted through the natural-number rank cast into that field. Positivity means positivity of this integer, not an order on an arbitrary coefficient field. In positive characteristic, a nonzero projector can have field-valued trace zero. The rank remains the correct nonemptiness test.

For commuting split semisimple operators, products of their exact spectral projectors detect joint eigenspaces. Pairwise occupancy need not imply joint occupancy. Plate X.18 uses six binary labels for which each pair of the target conditions $(0, 1, 1)$ is witnessed, while the complete triple is absent. The product projector is zero although all pairwise tests pass.

10.1 Why the terminal assembly permits an empty selection

The multiplicities are natural numbers. The vector $m_i = 0$ satisfies $m_i \le c_i$ for every carrier. Its selected copy space is zero, its splitting is valid, and its polynomial products are empty products with value one. All of this is consistent with the formal theorem.

To infer a nonzero summand, an application needs an index i with $m_i > 0$ and a nonzero constituent W_i. Under the finite-dimensional field assumptions,

$$\dim\Big(\bigoplus_i W_i^{\oplus m_i}\Big) = \sum_i m_i \dim W_i > 0$$

then gives the desired conclusion. Dropping that premise would strengthen the theorem to a false statement.

The indexed required-packet criterion is a different statement. It requires a nonzero vector in every supplied required packet. A finite sample of packet indices need not test them all. It also does not produce one common vector belonging to several unrelated carriers. Quantifiers and carrier types must remain visible in the application.

11 The final conjunction and its exact logical strength

Write $A(\sigma)$ for the transported-family conclusion and $B(S)$ for carrier recovery. The assembly has the form

$$(\exists \sigma\ A(\sigma))\ \wedge\ (\exists S\ B(S)).$$

It does not assert an isomorphism between the two witnesses. In a finite logical model, A may hold only at zero and B only at one. Both existential statements are true, but no object satisfies both. This elementary countermodel explains why conjoining proofs cannot create a missing comparison.

In the actual source, the separation is stronger than different names: the transported family is represented by matrices over coefficient rings, whereas carrier recovery is formulated through constituents and presentations over coefficient fields. A comparison may need scalar extension, an equivariant isomorphism, and compatibility with the local structures. There is no canonical identification merely because both outputs are called a system or a packet.

11.1 Read the six proof lines

The terminal declaration first uses `constructor`. The left branch calls `exists_transported_system_of_inverseLimitClosed`. The right branch calls `carrier_recovery_from_arithmetic_separator`. The latter obtains a trace separator and invokes the strict compatible-carrier theorem. The dependency pattern is acyclic and the two branches share only the parameters they actually use.

The carrier statement is at an arbitrary distinguished coefficient place. Repeating the theorem at all places requires the corresponding carrier and initial-subrepresentation presentations there. Choice can select witnesses once those pointwise existence statements have been proved. It cannot fill a place at which their hypotheses are unavailable.

11.2 A practical application contract

An application should first identify its complete source members and the actual finite-level transports. It must then prove the target profile’s closure and supply the constituent family, carrier presentations, and compatible sample data. Only after applying the two construction branches should it attempt any desired identification or nonzero conclusion. This order keeps construction, comparison, and occupancy separate rather than using one as an unstated substitute for another.

14 The complete terminal module

File: `ConditionalOccupiedSystemCompletion.lean`. SHA-256: `ba15fd1ea175fecfcc705243c75d1061c8bdf500221dafe86f29fe1c5d21c5e0`.

```
import PaperX.HorizontalTransportCompletion
import PaperX.Claim91ConstructedClosure
import PaperX.StrictCompatibleCarrierCompletion

/-!
# Conditional occupied-system completion

This is the final assembly theorem for Paper X.  It keeps the two conclusions
of the manuscript separate:

* vertical/horizontal completion constructs a simultaneous target family with
  the declared inverse-limit-closed profile;
* arithmetic Frobenius separation and compatible-carrier closure recover the
  declared constituent packet in the known carrier.

No residual companion, carrier, or local comparison is manufactured by this
assembly theorem.  Those are exactly the explicit hypotheses of the conditional
source statement.
-/
set_option autoImplicit false
noncomputable section

namespace PaperX.ConditionalOccupiedSystemCompletion

open scoped BigOperators
open Polynomial
open PaperX.RepresentationLimits
open PaperX.RepresentationClasses
open PaperX.ClassCompletion
open PaperX.InverseLimitLocalConditions
open PaperX.HorizontalTransportCompletion
open PaperX.AbsoluteFrobeniusWitnesses
open PaperX.AbsoluteIntegralFrobenius
open PaperX.CompatibleSystemRigidity
open PaperX.FiniteSeparation
open PaperX.TraceRecovery
open PaperX.CarrierSummands
open PaperX.WeilData
open PaperX.WeilCopies
open PaperX.WeilSmooth
open PaperX.WeilEuler
open PaperX.WeilProfileClosure
open PaperX.SemilinearTateComparison
open PaperX.SemilinearPeriods

universe u z uB

/-- The exact local carrier conclusion proved by Claim 9.2 at one arbitrary
coefficient realization.  Since the coefficient place is arbitrary and the
same `m` is used, this is the pointwise form of the compatible subsystem. -/
def CarrierRecoveryAt
    {K : Type u} [Field K] [CharZero K] [TopologicalSpace K]
    {GG : Type} [Monoid GG] [TopologicalSpace GG]
    {CIdx : Type} [Fintype CIdx] [DecidableEq CIdx]
    {V : CIdx → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)] [∀ i, TopologicalSpace (V i)]
    (ρ : ∀ i, Representation K GG (V i))
    {W : Type z} [AddCommGroup W] [Module K W]
```

```
    (carrier : Representation K GG W)
    (m : CIdx → ℕ)
    {U₀ : Type z} [AddCommGroup U₀] [Module K U₀]
    [FiniteDimensional K U₀]
    (ρU₀ : Representation K GG U₀)
    (sample : CIdx → GG)
    {E : Type uB} [CommRing E] (σE : E →+* K)
    (g : GG) (P : CIdx → E[X])
    {GW : Type} [Group GW] [TopologicalSpace GW]
    {ν : GW →* Kˣ} (H : Subgroup GW) [H.Normal]
    (Dwd : ∀ i, InertialDatum (ν := ν) H (V i)) (φ : GW)
    {Bper : Type uB} [Field Bper] [Algebra K Bper]
    {GP : Type} [Group GP]
    (σB : GP →* (Bper ≃ₐ[K] Bper)) (tate : ℤ → GP →* Kˣ)
    (ρp : ∀ i, Representation K GP (V i))
    (Cper : ∀ i, Comparison σB tate (ρp i)) : Prop :=
  (∃ j : Representation.IntertwiningMap ρU₀ carrier,
    ∃ r : Representation.IntertwiningMap carrier ρU₀,
      ∀ x, r (j x) = x) ∧
  (∀ n : CIdx → ℕ,
    (∀ j, LinearMap.trace K U₀ (ρU₀ (sample j)) =
      LinearMap.trace K _ (copiesRep ρ n (sample j))) → n = m) ∧
  (∃ j : Representation.IntertwiningMap (copiesRep ρ m) carrier,
    ∃ r : Representation.IntertwiningMap carrier (copiesRep ρ m),
      (∀ x, r (j x) = x) ∧
      Continuous (fun gx : GG × Copies V m => copiesRep ρ m gx.1 gx.2) ∧
      (copiesRep ρ m g).charpoly = (∏ i, P i ^ m i).map σE) ∧
  (let S := copiesInertialDatum H Dwd m;
    S.toDatum = copiesDatum (fun i => (Dwd i).toDatum) m ∧
    localCharpoly S.toDatum H φ =
      ∏ i, localCharpoly (Dwd i).toDatum H φ ^ m i ∧
    eulerPolynomial S.toDatum H φ =
      ∏ i, eulerPolynomial (Dwd i).toDatum H φ ^ m i ∧
    Module.finrank K (fixedKernel S.toDatum H) =
      ∑ i, m i * Module.finrank K (fixedKernel (Dwd i).toDatum H) ∧
    LinearMap.trace K _ (fixedRep S.toDatum H φ) =
      ∑ i, (m i : K) * LinearMap.trace K _ (fixedRep (Dwd i).toDatum H φ)) ∧
  (∃ Csel : Comparison σB tate (copiesRep ρp m),
    (∀ q : ℤ,
      weightMultiplicity σB tate Csel q =
        ∑ i, m i * weightMultiplicity σB tate (Cper i) q) ∧
    (∀ q : ℤ, Nonempty (
      periodSpace σB tate (copiesRep ρp m) q ≃ₗ[K]
        Copies (fun i => periodSpace σB tate (ρp i) q) m)))

/-- Claim 9.1's constructed arithmetic separator gives the literal nonsingular
trace sample required by Claim 9.2 at any selected coefficient place. -/
theorem finite_trace_separator_at
    {Λ CIdx : Type} {E : Type uB} [Field E]
    [Fintype CIdx] [DecidableEq CIdx]
    {K : Λ → Type u} [∀ l, Field (K l)] [∀ l, CharZero (K l)]
    [∀ l, TopologicalSpace (K l)] [∀ l, IsTopologicalRing (K l)]
    [∀ l, T2Space (K l)]
    {V : Λ → CIdx → Type u}
    [∀ l i, AddCommGroup (V l i)] [∀ l i, Module (K l) (V l i)]
    [∀ l i, FiniteDimensional (K l) (V l i)]
    (ρ : (l : Λ) → (i : CIdx) → Representation (K l) GQ (V l i))
    [∀ l i, (ρ l i).IsIrreducible]
    (hc : ∀ l i, Continuous (ρ l i).character)
    (emb : (l : Λ) → E →+* K l)
    (t : CIdx → RationalPrime → E)
    (good : Λ → RationalPrime → Prop)
    (hgood : ∀ l, Set.Finite {p | ¬ good l p})
    (compat : ∀ l i p, good l p →
      (ρ l i).character (chosenWitness p).element = emb l (t i p))
    (hne : ∀ i j, MemberwiseEquiv ρ i j → i = j)
```

```
    (l₀ : Λ) (bad : Finset RationalPrime) :
    ∃ v : CIdx → RationalPrime,
      Function.Injective v ∧
      (∀ i, good l₀ (v i)) ∧
      (∀ i, v i ∉ bad) ∧
      Matrix.det (fun j i =>
        LinearMap.trace (K l₀) (V l₀ i)
          (ρ l₀ i (chosenWitness (v j)).element)) ≠ 0 := by
  obtain ⟨v, hv, hvgood, hvbad, _, hmapdet, hentries⟩ :=
    PaperX.Claim91ConstructedClosure.finite_arithmetic_system_separator
      ρ hc emb t good hgood compat hne l₀ bad
  refine ⟨v, hv, hvgood, hvbad, ?_⟩
  have hmatrix :
      (emb l₀).mapMatrix (sampleMatrix t v) =
        (fun j i => LinearMap.trace (K l₀) (V l₀ i)
          (ρ l₀ i (chosenWitness (v j)).element)) := by
    ext j i
    simpa [Representation.character] using hentries l₀ j i (hvgood j)
  rw [← hmatrix]
  exact hmapdet l₀

/-- Claims 9.1 and 9.2 compose: arithmetic occurrence constructs the finite
separator, and the compatible-carrier theorem then recovers the declared packet
and all supplied strict-compatible local data. -/
theorem carrier_recovery_from_arithmetic_separator
    {Λ CIdx : Type} {E : Type uB} [Field E]
    [Fintype CIdx] [DecidableEq CIdx]
    {K : Λ → Type u} [∀ l, Field (K l)] [∀ l, CharZero (K l)]
    [∀ l, TopologicalSpace (K l)] [∀ l, IsTopologicalRing (K l)]
    [∀ l, T2Space (K l)]
    {V : Λ → CIdx → Type u}
    [∀ l i, AddCommGroup (V l i)] [∀ l i, Module (K l) (V l i)]
    [∀ l i, FiniteDimensional (K l) (V l i)]
    [∀ l i, TopologicalSpace (V l i)]
    (ρ : (l : Λ) → (i : CIdx) → Representation (K l) GQ (V l i))
    [∀ l i, (ρ l i).IsIrreducible]
    (hc : ∀ l i, Continuous (ρ l i).character)
    (emb : (l : Λ) → E →+* K l)
    (t : CIdx → RationalPrime → E)
    (good : Λ → RationalPrime → Prop)
    (hgood : ∀ l, Set.Finite {p | ¬ good l p})
    (compat : ∀ l i p, good l p →
      (ρ l i).character (chosenWitness p).element = emb l (t i p))
    (hne : ∀ i j, MemberwiseEquiv ρ i j → i = j)
    (l₀ : Λ) (bad : Finset RationalPrime)
    (hρ : ∀ i, Continuous (fun gx : GQ × V l₀ i => ρ l₀ i gx.1 gx.2))
    {W : Type z} [AddCommGroup W] [Module (K l₀) W]
    (carrier : Representation (K l₀) GQ W)
    (m c : CIdx → ℕ) (hmc : ∀ i, m i ≤ c i)
    (Bcar : W ≃ₗ[K l₀] Copies (V l₀) c)
    (hBcar : ∀ g x, Bcar (carrier g x) = copiesRep (ρ l₀) c g (Bcar x))
    {U₀ : Type z} [AddCommGroup U₀] [Module (K l₀) U₀]
    [FiniteDimensional (K l₀) U₀]
    (ρU₀ : Representation (K l₀) GQ U₀)
    (e₀ : U₀ ≃ₗ[K l₀] Copies (V l₀) m)
    (he₀ : ∀ g x, e₀ (ρU₀ g x) = copiesRep (ρ l₀) m g (e₀ x))
    (g : GQ) (P : CIdx → E[X])
    (hP : ∀ i, (ρ l₀ i g).charpoly = (P i).map (emb l₀))
    {GW : Type} [Group GW] [TopologicalSpace GW]
    {ν : GW →* (K l₀)ˣ} (H : Subgroup GW) [H.Normal]
    (Dwd : ∀ i, InertialDatum (ν := ν) H (V l₀ i)) (φ : GW)
    {Bper : Type uB} [Field Bper] [Algebra (K l₀) Bper]
    {GP : Type} [Group GP]
    (σB : GP →* (Bper ≃ₐ[K l₀] Bper))
    (tate : ℤ → GP →* (K l₀)ˣ)
    (ρp : ∀ i, Representation (K l₀) GP (V l₀ i))
```

```
    (Cper : ∀ i, Comparison σB tate (ρp i)) :
    ∃ ν : CIdx → RationalPrime,
      Function.Injective ν ∧
      (∀ i, good l₀ (ν i)) ∧
      (∀ i, ν i ∉ bad) ∧
      CarrierRecoveryAt (K := K l₀) (GG := GQ) (CIdx := CIdx) (V := V l₀)
        (W := W) (U₀ := U₀) (E := E) (GW := GW) (ν := ν)
        (Bper := Bper) (GP := GP)
        (ρ l₀) carrier m ρU₀ (fun j => (chosenWitness (ν j)).element)
        (emb l₀) g P H Dwd φ σB tate ρp Cper := by
  obtain ⟨ν, hν, hνgood, hνbad, hdet⟩ :=
    finite_trace_separator_at ρ hc emb t good hgood compat hne l₀ bad
  refine ⟨ν, hν, hνgood, hνbad, ?_⟩
  simpa [CarrierRecoveryAt] using
    (PaperX.StrictCompatibleCarrierCompletion.compatible_carrier_summand_completion
      (ρ l₀) hρ carrier m c hmc Bcar hBcar ρU₀ e₀ he₀
      (fun j => (chosenWitness (ν j)).element) hdet
      (emb l₀) g P hP H Dwd φ σB tate ρp Cper)

/-- Final Paper X assembly at an arbitrary coefficient place.  The first output
is the simultaneous family supplied by vertical/horizontal completion and
inverse-limit-closed local conditions.  The second, logically independent,
output is the compatible carrier summand recovered by finite Frobenius traces.
This matches the two-clause conclusion of the manuscript's conditional
occupied-system completion theorem. -/
theorem conditional_occupied_system_completion
    {Λ CIdx : Type} {E : Type uB} [Field E]
    {ι : Type} [Fintype ι] [DecidableEq ι]
    (O : Λ → Type) [∀ l, CommRing (O l)]
    (J : (l : Λ) → Ideal (O l)) [∀ l, IsAdicComplete (J l) (O l)]
    (Csrc Ctgt : (l : Λ) → ClassConditions (G := GQ) (ι := ι) (J l))
    (finiteO : ∀ l k, Finite (O l ⧸ (J l) ^ k))
    (T : ∀ l, AdmissibleClassTransport (J l) (Csrc l) (Ctgt l))
    (TargetProfile : (l : Λ) → MatrixRep (O l) GQ ι → Prop)
    (closed : ∀ l, InverseLimitClosed (J l) (Ctgt l) (TargetProfile l))
    (ρsrc : (l : Λ) → MatrixRep (O l) GQ ι)
    (hρsrc : ∀ l k,
      (Csrc l).allowed k (classOf (reduce (J l) (ρsrc l) k)))
    [Fintype CIdx] [DecidableEq CIdx]
    {K : Λ → Type u} [∀ l, Field (K l)] [∀ l, CharZero (K l)]
    [∀ l, TopologicalSpace (K l)] [∀ l, IsTopologicalRing (K l)]
    [∀ l, T2Space (K l)]
    {V : Λ → CIdx → Type u}
    [∀ l i, AddCommGroup (V l i)] [∀ l i, Module (K l) (V l i)]
    [∀ l i, FiniteDimensional (K l) (V l i)]
    [∀ l i, TopologicalSpace (V l i)]
    (ρ : (l : Λ) → (i : CIdx) → Representation (K l) GQ (V l i))
    [∀ l i, (ρ l i).IsIrreducible]
    (hc : ∀ l i, Continuous (ρ l i).character)
    (emb : (l : Λ) → E →+* K l)
    (t : CIdx → RationalPrime → E)
    (good : Λ → RationalPrime → Prop)
    (hgood : ∀ l, Set.Finite {p | ¬ good l p})
    (compat : ∀ l i p, good l p →
      (ρ l i).character (chosenWitness p).element = emb l (t i p))
    (hne : ∀ i j, MemberwiseEquiv ρ i j → i = j)
    (l₀ : Λ) (bad : Finset RationalPrime)
    (hρ : ∀ i, Continuous (fun gx : GQ × V l₀ i => ρ l₀ i gx.1 gx.2))
    {W : Type z} [AddCommGroup W] [Module (K l₀) W]
    (carrier : Representation (K l₀) GQ W)
    (m c : CIdx → ℕ) (hmc : ∀ i, m i ≤ c i)
    (Bcar : W ≃ₗ[K l₀] Copies (V l₀) c)
    (hBcar : ∀ g x, Bcar (carrier g x) = copiesRep (ρ l₀) c g (Bcar x))
    {U₀ : Type z} [AddCommGroup U₀] [Module (K l₀) U₀]
    [FiniteDimensional (K l₀) U₀]
    (ρU₀ : Representation (K l₀) GQ U₀)
```

```
    (e₀ : U₀ ≃ₗ[K l₀] Copies (V l₀) m)
    (he₀ : ∀ g x, e₀ (ρU₀ g x) = copiesRep (ρ l₀) m g (e₀ x))
    (g : GQ) (P : CIdx → E[X])
    (hP : ∀ i, (ρ l₀ i g).charpoly = (P i).map (emb l₀))
    {GW : Type} [Group GW] [TopologicalSpace GW]
    {ν : GW →* (K l₀)ˣ} (H : Subgroup GW) [H.Normal]
    (Dwd : ∀ i, InertialDatum (ν := ν) H (V l₀ i)) (φ : GW)
    {Bper : Type uB} [Field Bper] [Algebra (K l₀) Bper]
    {GP : Type} [Group GP]
    (σB : GP →* (Bper ≃ₐ[K l₀] Bper))
    (tate : ℤ → GP →* (K l₀)ˣ)
    (ρp : ∀ i, Representation (K l₀) GP (V l₀ i))
    (Cper : ∀ i, Comparison σB tate (ρp i)) :
    (∃ σ : (l : Λ) → MatrixRep (O l) GQ ι,
      (∀ l k, (Ctgt l).allowed k (classOf (reduce (J l) (σ l) k))) ∧
      (∀ l, TransportedAlong (J l) (T l) (ρsrc l) (σ l)) ∧
      ∀ l, TargetProfile l (σ l)) ∧
    (∃ ν : CIdx → RationalPrime,
      Function.Injective ν ∧
      (∀ i, good l₀ (ν i)) ∧
      (∀ i, ν i ∉ bad) ∧
      CarrierRecoveryAt (K := K l₀) (GG := GQ) (CIdx := CIdx) (V := V l₀)
        (W := W) (U₀ := U₀) (E := E) (GW := GW) (ν := ν)
        (Bper := Bper) (GP := GP)
        (ρ l₀) carrier m ρU₀ (fun j => (chosenWitness (ν j)).element)
        (emb l₀) g P H Dwd φ σB tate ρp Cper) := by
  constructor
  · exact exists_transported_system_of_inverseLimitClosed O J Csrc Ctgt finiteO T
      TargetProfile closed ρsrc hρsrc
  · exact carrier_recovery_from_arithmetic_separator ρ hc emb t good hgood compat hne
      l₀ bad hρ carrier m c hmc Bcar hBcar ρU₀ e₀ he₀ g P hP
      H Dwd φ σB tate ρp Cper

end PaperX.ConditionalOccupiedSystemCompletion
```

15 Direct dependency: horizontal transport

File: source/HorizontalTransportCompletion.lean. SHA-256: 91c7652afda255943a57ac983414bd6ee2e786e26899175e51e6b390e1fbbfda.

```
import PaperX.ClassBranchLifting
import PaperX.ClassCompletion
import PaperX.FiniteFiberBranches
import PaperX.InverseLimitLocalConditions

/-!
Paper X: horizontal transport completion block.

This formalizes the finite-depth content of an admissible residual transport:
a map on representation classes at every precision, preservation of the selected
admissible conditions, and naturality with respect to coefficient reduction.

The main theorem proves that such a transport sends a completed source branch
to a compatible target branch and, over an adically complete coefficient ring
with finite coefficient quotients, reconstructs an actual target
representation realizing that transported branch.  No arithmetic transport is
asserted unless supplied through the `allowed` predicates and the transport
fields below.
-/
set_option autoImplicit false

namespace PaperX.HorizontalTransportCompletion

open PaperX.Compactness
open PaperX.FiniteFiberBranches
open PaperX.RepresentationLimits
open PaperX.RepresentationClasses
open PaperX.ClassCompletion
open PaperX.ClassBranchLifting
open PaperX.InverseLimitLocalConditions

variable {R G ι : Type} [CommRing R] [Group G]
  [Fintype ι] [DecidableEq ι]
variable (I : Ideal R)

/-- Finite-depth form of an admissible residual transport.  It acts on actual
change-of-basis classes, carries every selected source condition to the target
condition, and commutes with reduction between arbitrary precisions. -/
structure AdmissibleClassTransport
    (C D : ClassConditions (G := G) (ι := ι) I) where
  map : (k : ℕ) → ClassRep (R ⧸ I ^ k) G ι → ClassRep (R ⧸ I ^ k) G ι
  allowed : ∀ k c, C.allowed k c → D.allowed k (map k c)
  reduction : ∀ {j k : ℕ} (h : j ≤ k) c,
    transitionClass I h (map k c) = map j (transitionClass I h c)

/-- The inverse system of admissible representation classes for one selected
condition family. -/
def conditionTower (C : ClassConditions (G := G) (ι := ι) I) : Tower where
  Level k := {c : ClassRep (R ⧸ I ^ k) G ι // C.allowed k c}
  parent k c :=
    (transitionClass I (Nat.le_succ k) c.val,
      C.downward (Nat.le_succ k) c.val c.property)

/-- An admissible transport is literally a map of the corresponding inverse
towers. -/
def AdmissibleClassTransport.asTowerMap
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D) :
    TowerMap (conditionTower I C) (conditionTower I D) where
```

```
  map k c := ⟨T.map k c.val, T.allowed k c.val c.property⟩
  parent_commutes k c := by
    apply Subtype.ext
    exact (T.reduction (Nat.le_succ k) c.val).symm

/-- Identity is an admissible transport. -/
def AdmissibleClassTransport.id
    (C : ClassConditions (G := G) (ι := ι) I) :
    AdmissibleClassTransport I C C where
  map := fun _ c => c
  allowed := fun _ _ h => h
  reduction := by
    intro j k h c
    rfl

/-- Admissible transports compose without losing reduction naturality. -/
def AdmissibleClassTransport.comp
    {C D E : ClassConditions (G := G) (ι := ι) I}
    (U : AdmissibleClassTransport I D E)
    (T : AdmissibleClassTransport I C D) :
    AdmissibleClassTransport I C E where
  map := fun k c => U.map k (T.map k c)
  allowed := fun k c hc => U.allowed k (T.map k c) (T.allowed k c hc)
  reduction := by
    intro j k h c
    calc
      transitionClass I h (U.map k (T.map k c))
          = U.map j (transitionClass I h (T.map k c)) :=
            U.reduction h (T.map k c)
      _ = U.map j (T.map j (transitionClass I h c)) :=
            congrArg (U.map j) (T.reduction h c)

/-- A completed integral representation determines its exact admissible class
branch. -/
def realizationBranch
    (C : ClassConditions (G := G) (ι := ι) I)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k))) :
    Branch (conditionTower I C) where
  value k := ⟨classOf (reduce I ρ k), hρ k⟩
  compatible k := by
    apply Subtype.ext
    simp [conditionTower]

/-- Transport an entire compatible branch, not merely one residual root. -/
def transportBranch
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (b : Branch (conditionTower I C)) : Branch (conditionTower I D) :=
  mapBranch (T.asTowerMap I) b

@[simp] theorem transportBranch_value
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (b : Branch (conditionTower I C)) (k : ℕ) :
    (transportBranch I T b).value k =
      ⟨T.map k (b.value k).val, T.allowed k (b.value k).val (b.value k).property⟩ :=
  rfl

/-- Forget the admissibility witness while retaining the compatible class
branch. -/
def forgetConditionTower
    (C : ClassConditions (G := G) (ι := ι) I) :
    TowerMap (conditionTower I C) (classTower (G := G) (ι := ι) I) where
  map := fun _ c => c.val
  parent_commutes := by
```

```
    intro k c
    rfl

/-- The transported admissible branch viewed as an ordinary branch of
representation classes. -/
def transportedClassBranch
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k))) :
    Branch (classTower (G := G) (ι := ι) I) :=
  mapBranch (forgetConditionTower I D)
    (transportBranch I T (realizationBranch I C ρ hρ))

@[simp] theorem transportedClassBranch_value
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k))) (k : ℕ) :
    (transportedClassBranch I T ρ hρ).value k =
      T.map k (classOf (reduce I ρ k)) :=
  rfl

/-- Horizontal transport plus vertical completeness reconstructs an actual
integral target realization.  The conclusion remembers the exact transported
class at every finite precision; it is stronger than target nonemptiness. -/
theorem exists_transported_realization [IsAdicComplete I R]
    {C D : ClassConditions (G := G) (ι := ι) I}
    (finiteR : ∀ k, Finite (R ⧸ I ^ k))
    (T : AdmissibleClassTransport I C D)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k))) :
    ∃ σ : MatrixRep R G ι,
      (∀ k, D.allowed k (classOf (reduce I σ k))) ∧
      ∀ k, classOf (reduce I σ k) = T.map k (classOf (reduce I ρ k)) := by
  let b := transportedClassBranch I T ρ hρ
  obtain ⟨σ, hσ⟩ := realization_of_class_branch I finiteR b
  refine ⟨σ, ?_, ?_⟩
  · intro k
    rw [hσ k]
    exact ((transportBranch I T (realizationBranch I C ρ hρ)).value k).property
  · intro k
    exact hσ k

variable [TopologicalSpace G]

/-- If every finite representative of an allowed target class is continuous,
the transported realization can be chosen adically continuous as well. -/
theorem exists_continuous_transported_realization [IsAdicComplete I R]
    {C D : ClassConditions (G := G) (ι := ι) I}
    (finiteR : ∀ k, Finite (R ⧸ I ^ k))
    (T : AdmissibleClassTransport I C D)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k)))
    (continuous : ∀ k (τ : MatrixRep (R ⧸ I ^ k) G ι),
      D.allowed k (classOf τ) → PaperX.RepresentationTopology.FiniteRepContinuous τ) :
    ∃ σ : MatrixRep R G ι,
      (∀ k, D.allowed k (classOf (reduce I σ k))) ∧
      PaperX.RepresentationTopology.AdicRepContinuous I σ ∧
      ∀ k, classOf (reduce I σ k) = T.map k (classOf (reduce I ρ k)) := by
  let b := transportedClassBranch I T ρ hρ
  obtain ⟨σ, hσ, hc⟩ := continuous_realization_of_class_branch I finiteR b
    (fun k τ hτ => continuous k τ (by
      rw [hτ]
      exact ((transportBranch I T (realizationBranch I C ρ hρ)).value k).property))
  refine ⟨σ, ?_, hc, ?_⟩
```

```
  · intro k
    rw [hσ k]
    exact ((transportBranch I T (realizationBranch I C ρ hρ)).value k).property
  · intro k
    exact hσ k

end PaperX.HorizontalTransportCompletion

namespace PaperX.HorizontalTransportCompletion

open PaperX.InverseLimitLocalConditions

open PaperX.RepresentationLimits
open PaperX.RepresentationClasses
open PaperX.ClassCompletion

variable {R G ι : Type} [CommRing R] [Group G]
  [Fintype ι] [DecidableEq ι]
variable (I : Ideal R)

/-- Exact relation between complete realizations induced by a finite-depth
horizontal transport.  It records the whole compatible system of transported
classes, not only the residual class. -/
def TransportedAlong
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (ρ σ : MatrixRep R G ι) : Prop :=
  ∀ k, classOf (reduce I σ k) = T.map k (classOf (reduce I ρ k))


/-- If the target complete property is known to be inverse-limit closed for the
selected target finite-level conditions, the transported reconstruction satisfies
that property automatically.  This is the literal closure hypothesis appearing
in Claim 11.1; no separate complete-level transport axiom is needed. -/
theorem exists_transported_realization_of_inverseLimitClosed [IsAdicComplete I R]
    {C D : ClassConditions (G := G) (ι := ι) I}
    (finiteR : ∀ k, Finite (R ⧸ I ^ k))
    (T : AdmissibleClassTransport I C D)
    (TargetProfile : MatrixRep R G ι → Prop)
    (closed : InverseLimitClosed I D TargetProfile)
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k))) :
    ∃ σ : MatrixRep R G ι,
      (∀ k, D.allowed k (classOf (reduce I σ k))) ∧
      TransportedAlong I T ρ σ ∧ TargetProfile σ := by
  obtain ⟨σ, hD, hT⟩ := exists_transported_realization I finiteR T ρ hρ
  exact ⟨σ, hD, hT,
    (inverseLimitClosed_iff_realizations I D TargetProfile).mp closed σ hD⟩

/-- Arithmetic semantics are intentionally separated from the class-level
transport.  A selected complete profile is preserved only when this implication
is actually supplied/proved for the intended arithmetic realization. -/
def PreservesCompleteProfile
    {C D : ClassConditions (G := G) (ι := ι) I}
    (T : AdmissibleClassTransport I C D)
    (SourceProfile TargetProfile : MatrixRep R G ι → Prop) : Prop :=
  ∀ ρ σ, TransportedAlong I T ρ σ → SourceProfile ρ → TargetProfile σ

/-- The class transport and an explicit complete-profile preservation theorem
compose without conflating finite congruence data with arithmetic semantics. -/
theorem exists_transported_realization_with_profile [IsAdicComplete I R]
    {C D : ClassConditions (G := G) (ι := ι) I}
    (finiteR : ∀ k, Finite (R ⧸ I ^ k))
    (T : AdmissibleClassTransport I C D)
    (SourceProfile TargetProfile : MatrixRep R G ι → Prop)
    (preserves : PreservesCompleteProfile I T SourceProfile TargetProfile)
```

```
    (ρ : MatrixRep R G ι)
    (hρ : ∀ k, C.allowed k (classOf (reduce I ρ k)))
    (hSource : SourceProfile ρ) :
    ∃ σ : MatrixRep R G ι,
      (∀ k, D.allowed k (classOf (reduce I σ k))) ∧
      TransportedAlong I T ρ σ ∧ TargetProfile σ := by
  obtain ⟨σ, hD, hT⟩ := exists_transported_realization I finiteR T ρ hρ
  exact ⟨σ, hD, hT, preserves ρ σ hT hSource⟩

section Indexed

variable {Λ : Type}
variable (O : Λ → Type) [∀ l, CommRing (O l)]
variable {G ι : Type} [Group G] [Fintype ι] [DecidableEq ι]
variable (J : (l : Λ) → Ideal (O l))

/-- Coefficient-indexed horizontal completion.  Once one completed source
member is supplied at each coefficient place, admissible horizontal transports
produce actual target members at every place with exact finite-depth class
matching.  This is the system-level algebraic content of the horizontal half of
Claim 11.1; no Frobenius separation or WD/Hodge--Tate comparison is hidden in
this statement. -/
theorem exists_transported_system
    [∀ l, IsAdicComplete (J l) (O l)]
    (C D : (l : Λ) → ClassConditions (G := G) (ι := ι) (J l))
    (finiteO : ∀ l k, Finite (O l ⧸ (J l) ^ k))
    (T : ∀ l, AdmissibleClassTransport (J l) (C l) (D l))
    (ρ : (l : Λ) → MatrixRep (O l) G ι)
    (hρ : ∀ l k, (C l).allowed k (classOf (reduce (J l) (ρ l) k))) :
    ∃ σ : (l : Λ) → MatrixRep (O l) G ι,
      (∀ l k, (D l).allowed k (classOf (reduce (J l) (σ l) k))) ∧
      ∀ l, TransportedAlong (J l) (T l) (ρ l) (σ l) := by
  classical
  have H : ∀ l, ∃ σ : MatrixRep (O l) G ι,
      (∀ k, (D l).allowed k (classOf (reduce (J l) σ k))) ∧
      TransportedAlong (J l) (T l) (ρ l) σ := by
    intro l
    exact exists_transported_realization (J l) (finiteO l) (T l) (ρ l) (hρ l)
  refine (fun l => Classical.choose (H l), ?_, ?_)
  · intro l k
    exact (Classical.choose_spec (H l)).1 k
  · intro l
    exact (Classical.choose_spec (H l)).2


/-- Coefficient-indexed form using exactly the inverse-limit-closure premise of
Claim 11.1.  Every target member reconstructed from transported finite classes
inherits the supplied closed local/arithmetic predicate. -/
theorem exists_transported_system_of_inverseLimitClosed
    [∀ l, IsAdicComplete (J l) (O l)]
    (C D : (l : Λ) → ClassConditions (G := G) (ι := ι) (J l))
    (finiteO : ∀ l k, Finite (O l ⧸ (J l) ^ k))
    (T : ∀ l, AdmissibleClassTransport (J l) (C l) (D l))
    (TargetProfile : (l : Λ) → MatrixRep (O l) G ι → Prop)
    (closed : ∀ l, InverseLimitClosed (J l) (D l) (TargetProfile l))
    (ρ : (l : Λ) → MatrixRep (O l) G ι)
    (hρ : ∀ l k, (C l).allowed k (classOf (reduce (J l) (ρ l) k))) :
    ∃ σ : (l : Λ) → MatrixRep (O l) G ι,
      (∀ l k, (D l).allowed k (classOf (reduce (J l) (σ l) k))) ∧
      (∀ l, TransportedAlong (J l) (T l) (ρ l) (σ l)) ∧
      ∀ l, TargetProfile l (σ l) := by
  classical
  have H : ∀ l, ∃ σ : MatrixRep (O l) G ι,
      (∀ k, (D l).allowed k (classOf (reduce (J l) σ k))) ∧
      TransportedAlong (J l) (T l) (ρ l) σ ∧ TargetProfile l σ := by
    intro l
```

```
    exact exists_transported_realization_of_inverseLimitClosed (J l) (finiteO l)
      (T l) (TargetProfile l) (closed l) (ρ l) (hρ l)
  refine ⟨fun l => Classical.choose (H l), ?_, ?_, ?_⟩
  · intro l k
    exact (Classical.choose_spec (H l)).1 k
  · intro l
    exact (Classical.choose_spec (H l)).2.1
  · intro l
    exact (Classical.choose_spec (H l)).2.2

/-- Indexed version with an explicit arithmetic/local semantic profile.  The
only semantic premise is the named preservation implication; the compactness
and reconstruction layer does not manufacture Hodge--Tate, Weil--Deligne,
fixed-determinant, or other deformation properties. -/
theorem exists_transported_system_with_profile
    [∀ l, IsAdicComplete (J l) (O l)]
    (C D : (l : Λ) → ClassConditions (G := G) (ι := ι) (J l))
    (finiteO : ∀ l k, Finite (O l ⧸ (J l) ^ k))
    (T : ∀ l, AdmissibleClassTransport (J l) (C l) (D l))
    (SourceProfile TargetProfile : (l : Λ) → MatrixRep (O l) G ι → Prop)
    (preserves : ∀ l,
      PreservesCompleteProfile (J l) (T l) (SourceProfile l) (TargetProfile l))
    (ρ : (l : Λ) → MatrixRep (O l) G ι)
    (hρ : ∀ l k, (C l).allowed k (classOf (reduce (J l) (ρ l) k)))
    (hSource : ∀ l, SourceProfile l (ρ l)) :
    ∃ σ : (l : Λ) → MatrixRep (O l) G ι,
      (∀ l k, (D l).allowed k (classOf (reduce (J l) (σ l) k))) ∧
      (∀ l, TransportedAlong (J l) (T l) (ρ l) (σ l)) ∧
      ∀ l, TargetProfile l (σ l) := by
  classical
  have H : ∀ l, ∃ σ : MatrixRep (O l) G ι,
      (∀ k, (D l).allowed k (classOf (reduce (J l) σ k))) ∧
      TransportedAlong (J l) (T l) (ρ l) σ ∧ TargetProfile l σ := by
    intro l
    exact exists_transported_realization_with_profile (J l) (finiteO l) (T l)
      (SourceProfile l) (TargetProfile l) (preserves l) (ρ l) (hρ l) (hSource l)
  refine ⟨fun l => Classical.choose (H l), ?_, ?_, ?_⟩
  · intro l k
    exact (Classical.choose_spec (H l)).1 k
  · intro l
    exact (Classical.choose_spec (H l)).2.1
  · intro l
    exact (Classical.choose_spec (H l)).2.2

end Indexed

end PaperX.HorizontalTransportCompletion
```

16 Direct dependency: constructed arithmetic separation

File: source/Claim91ConstructedClosure.lean. SHA-256: f2d9bca535fe4fc8c2c6702e64e034cc3447de698cd29c0931f8ceb4642d751a.

```
import PaperX.ConstructedChebotarevBridge
import PaperX.ArithmeticSystemSeparation

/-!
# Constructed Claim 9.1 endpoint
The original Block37 separator statement, with the external API parameter
discharged by the actual general Frobenius occurrence construction.
-/
set_option autoImplicit false
noncomputable section

namespace PaperX.Claim91ConstructedClosure

open PaperX.AbsoluteFrobeniusWitnesses
open PaperX.AbsoluteIntegralFrobenius
open PaperX.ChebotarevExternalAPIAdapter
open PaperX.CompatibleSystemRigidity
open PaperX.FiniteSeparation PaperX.TraceRecovery

/-- The constructed Chebotarev implementation closes the exact
arithmetic occurrence obligation in Paper X Claim 9.1. -/
theorem existentialArithmeticCoverage :
    PaperX.GeneralArithmeticCoverage.ExistentialArithmeticCoverage :=
  PaperX.ConstructedChebotarevBridge.existentialArithmeticCoverage

variable {Λ I E : Type*} [Field E]
variable [Fintype I] [DecidableEq I]
variable {K : Λ → Type*} [∀ l, Field (K l)] [∀ l, CharZero (K l)]
  [∀ l, TopologicalSpace (K l)] [∀ l, IsTopologicalRing (K l)] [∀ l, T2Space (K l)]
variable {W : Λ → I → Type*} [∀ l i, AddCommGroup (W l i)]
  [∀ l i, Module (K l) (W l i)] [∀ l i, FiniteDimensional (K l) (W l i)]
variable (ρ : (l : Λ) → (i : I) → Representation (K l) GQ (W l i))
variable [∀ l i, (ρ l i).IsIrreducible]
variable (hc : ∀ l i, Continuous (ρ l i).character)
variable (emb : (l : Λ) → E →+* K l)
variable (t : I → RationalPrime → E)
variable (good : Λ → RationalPrime → Prop)
variable (hgood : ∀ l, Set.Finite {p | ¬ good l p})
variable (compat : ∀ l i p, good l p →
  (ρ l i).character (chosenWitness p).element = emb l (t i p))
variable (hne : ∀ i j, MemberwiseEquiv ρ i j → i = j)

include hc hgood compat hne in
/-- Full recorded Claim 9.1 separator conclusion with no external API premise.  The selected rational primes avoid every
prescribed finite forbidden set. -/
theorem finite_arithmetic_system_separator
    (l₀ : Λ) (bad : Finset RationalPrime) :
    ∃ v : I → RationalPrime, Function.Injective v ∧
      (∀ i, good l₀ (v i)) ∧ (∀ i, v i ∉ bad) ∧
      (sampleMatrix t v).det ≠ 0 ∧
      (∀ l, ((emb l).mapMatrix (sampleMatrix t v)).det ≠ 0) ∧
      ∀ l j i, good l (v j) →
        (emb l).mapMatrix (sampleMatrix t v) j i =
          (ρ l i).character (chosenWitness (v j)).element := by
  exact PaperX.ArithmeticSystemSeparation.finite_arithmetic_system_separator
    ρ hc emb t good hgood compat hne
    existentialArithmeticCoverage l₀ bad

end PaperX.Claim91ConstructedClosure
```

17 Direct dependency: strict compatible carrier closure

File: source/StrictCompatibleCarrierCompletion.lean. SHA-256: 4fa6b2fd0c212d75e1158049e777f94217b1dab5ce5f84c2690dfcb826406388.

```
import PaperX.CarrierCompatibility
import PaperX.WeilProfileClosure
import PaperX.SemilinearTateComparison
import PaperX.SemilinearPeriods

/-!
# Compatible-carrier summand completion

This is the source-faithful closure theorem for Paper X Claim 9.2.  The
constituent representations, their strict local Weil packets, and their genuine
semilinear Tate comparisons are inputs.  The selected multiplicities are then
used at every coefficient realization.  The theorem constructs the actual split
summand and proves closure of all local data under those finite copies.

It does not construct a compatible carrier or prove that an arbitrary p-adic
representation is Hodge--Tate.  Those claims are not part of the conditional
carrier theorem.
-/
set_option autoImplicit false
noncomputable section

namespace PaperX.StrictCompatibleCarrierCompletion

open scoped BigOperators
open Polynomial
open PaperX.CarrierSummands
open PaperX.CarrierCompatibility
open PaperX.WeilData
open PaperX.WeilCopies
open PaperX.WeilSmooth
open PaperX.WeilEuler
open PaperX.WeilProfileClosure
open PaperX.SemilinearTateComparison
open PaperX.SemilinearPeriods

universe u v w z uB


/-- A semantic local condition on representations over `K`. -/
def LocalCondition (K : Type u) (G : Type v) [CommRing K] [Monoid G] :=
  ∀ (X : Type u), [AddCommGroup X] → [Module K X] → Representation K G X → Prop

/-- A local condition is finite-copy closed when constituent instances imply
its instance on every finite copy carrier.  This explicit hypothesis is
necessary for conditions, such as fixed determinant, which are not automatically
stable under direct sum. -/
def FiniteCopyClosed
    {K : Type u} {G : Type v} [CommRing K] [Monoid G]
    (Q : LocalCondition K G) : Prop :=
  ∀ {I : Type} [Fintype I] {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    (ρ : ∀ i, Representation K G (V i)) (m : I → ℕ),
      (∀ i, Q (V i) (ρ i)) → Q (Copies V m) (copiesRep ρ m)

/-- Every explicitly finite-copy-closed local condition is inherited by the
selected compatible summand. -/
theorem selected_local_condition
    {K : Type u} {G : Type v} [CommRing K] [Monoid G]
    {I : Type} [Fintype I] {V : I → Type u}
```

```
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    (ρ : ∀ i, Representation K G (V i)) (m : I → ℕ)
    (Q : LocalCondition K G) (hclosed : FiniteCopyClosed Q)
    (hconst : ∀ i, Q (V i) (ρ i)) :
    Q (Copies V m) (copiesRep ρ m) :=
  hclosed ρ m hconst

/-- At one coefficient realization, a multiplicity vector bounded by the
carrier capacities produces an actual equivariant split summand, a continuous
selected action, and the expected common Frobenius polynomial. -/
theorem selected_global_carrier
    {K : Type u} [Field K] [TopologicalSpace K]
    {G : Type v} [Monoid G] [TopologicalSpace G]
    {I : Type} [Fintype I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)] [∀ i, TopologicalSpace (V i)]
    (ρ : ∀ i, Representation K G (V i))
    (hρ : ∀ i, Continuous (fun gx : G × V i => ρ i gx.1 gx.2))
    {W : Type z} [AddCommGroup W] [Module K W]
    (carrier : Representation K G W)
    (m c : I → ℕ) (hmc : ∀ i, m i ≤ c i)
    (B : W ≃ₗ[K] Copies V c)
    (hB : ∀ g x, B (carrier g x) = copiesRep ρ c g (B x))
    {E : Type uB} [CommRing E] (σE : E →+* K)
    (g : G) (P : I → E[X])
    (hP : ∀ i, (ρ i g).charpoly = (P i).map σE) :
    ∃ j : Representation.IntertwiningMap (copiesRep ρ m) carrier,
      ∃ r : Representation.IntertwiningMap carrier (copiesRep ρ m),
        (∀ x, r (j x) = x) ∧
        Continuous (fun gx : G × Copies V m => copiesRep ρ m gx.1 gx.2) ∧
        (copiesRep ρ m g).charpoly = (∏ i, P i ^ m i).map σE := by
  let jlin : Copies V m →ₗ[K] W := transportedInclusion hmc B
  let rlin : W →ₗ[K] Copies V m := transportedRetraction hmc B
  let j : Representation.IntertwiningMap (copiesRep ρ m) carrier :=
    jlin.intertwiningMap_of_isIntertwiningMap _ _ (by
      intro h x
      exact transported_inclusion_equivariant ρ carrier hmc B hB h x)
  let r : Representation.IntertwiningMap carrier (copiesRep ρ m) :=
    rlin.intertwiningMap_of_isIntertwiningMap _ _ (by
      intro h x
      exact transported_retraction_equivariant ρ carrier hmc B hB h x)
  refine ⟨j, r, ?_, ?_, ?_⟩
  · intro x
    exact transported_split hmc B x
  · exact continuous_copies_action ρ m hρ
  · exact common_polynomial σE ρ m g P hP

/-- Under an explicit nonsingular character sample, the coefficient-independent
multiplicity vector used by the completion is unique.  This is the formal
version of uniqueness in the semisimple carrier decomposition. -/
theorem selected_multiplicities_unique
    {K : Type u} [Field K] [CharZero K]
    {G : Type v} [Monoid G]
    {I : Type} [Fintype I] [DecidableEq I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)]
    (ρ : ∀ i, Representation K G (V i))
    (m : I → ℕ) (sample : I → G)
    (hdet : Matrix.det (fun j i => LinearMap.trace K _ (ρ i (sample j))) ≠ 0) :
    ∀ n : I → ℕ,
      (∀ j, LinearMap.trace K _ (copiesRep ρ m (sample j)) =
        LinearMap.trace K _ (copiesRep ρ n (sample j))) → n = m := by
  intro n htrace
  exact (multiplicities_unique_from_actual_traces ρ m n sample hdet htrace).symm
```

```
/-- Supplied constituent Weil--monodromy packets are closed under the selected
finite copies, including smooth inertia and the exact local polynomial,
dimension, and trace formulas. -/
theorem selected_weil_packets
    {K : Type u} [Field K]
    {GW : Type v} [Group GW] [TopologicalSpace GW]
    {I : Type} [Fintype I]
    {ν : GW →* Kˣ}
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)]
    (H : Subgroup GW) [H.Normal]
    (D : ∀ i, InertialDatum (ν := ν) H (V i))
    (m : I → ℕ) (φ : GW) :
    let S := copiesInertialDatum H D m
    S.toDatum = copiesDatum (fun i => (D i).toDatum) m ∧
    localCharpoly S.toDatum H φ =
      ∏ i, localCharpoly (D i).toDatum H φ ^ m i ∧
    eulerPolynomial S.toDatum H φ =
      ∏ i, eulerPolynomial (D i).toDatum H φ ^ m i ∧
    Module.finrank K (fixedKernel S.toDatum H) =
      ∑ i, m i * Module.finrank K (fixedKernel (D i).toDatum H) ∧
    LinearMap.trace K _ (fixedRep S.toDatum H φ) =
      ∑ i, (m i : K) * LinearMap.trace K _ (fixedRep (D i).toDatum H φ) := by
  exact copies_local_profile H D m φ

/-- Genuine semilinear comparison data on every constituent produce the actual
comparison on the selected finite copies.  The same construction gives the
intrinsic period-space equivalence, so the result is not merely a census of
formal weight labels. -/
theorem selected_semilinear_period_packets
    {R : Type u} {B : Type uB} {GP : Type v}
    [Field R] [Field B] [Algebra R B] [Group GP]
    (σB : GP →* (B ≃ₐ[R] B)) (tate : ℤ → GP →* Rˣ)
    {I : Type} [Fintype I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module R (V i)]
    (ρ : ∀ i, Representation R GP (V i))
    (C : ∀ i, Comparison σB tate (ρ i))
    (m : I → ℕ) :
    ∃ Csel : Comparison σB tate (copiesRep ρ m),
      (∀ z : ℤ,
        weightMultiplicity σB tate Csel z =
          ∑ i, m i * weightMultiplicity σB tate (C i) z) ∧
      (∀ z : ℤ, Nonempty (
        periodSpace σB tate (copiesRep ρ m) z ≃ₗ[R]
          Copies (fun i => periodSpace σB tate (ρ i) z) m)) := by
  refine ⟨copiesComparison σB tate ρ C m, ?_, ?_⟩
  · intro z
    exact copies_weightMultiplicity σB tate ρ C m z
  · intro z
    exact ⟨periodCopiesEquiv σB tate ρ m z⟩

/-- One-place capstone.  It combines the actual global split summand with the
constructed Weil packet and the genuine semilinear comparison.  Applying this
pointwise at every coefficient place, with the same `m`, is exactly the
compatible-carrier completion theorem. -/
theorem one_place_completion
    {K : Type u} [Field K] [TopologicalSpace K]
    {GG : Type v} [Monoid GG] [TopologicalSpace GG]
    {I : Type} [Fintype I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)] [∀ i, TopologicalSpace (V i)]
    (ρ : ∀ i, Representation K GG (V i))
```

```
    (hρ : ∀ i, Continuous (fun gx : GG × V i => ρ i gx.1 gx.2))
    {W : Type z} [AddCommGroup W] [Module K W]
    (carrier : Representation K GG W)
    (m c : I → ℕ) (hmc : ∀ i, m i ≤ c i)
    (Bcar : W ≃ₗ[K] Copies V c)
    (hBcar : ∀ g x, Bcar (carrier g x) = copiesRep ρ c g (Bcar x))
    {E : Type uB} [CommRing E] (σE : E →+* K)
    (g : GG) (P : I → E[X])
    (hP : ∀ i, (ρ i g).charpoly = (P i).map σE)
    {GW : Type v} [Group GW] [TopologicalSpace GW]
    {ν : GW →* Kˣ} (H : Subgroup GW) [H.Normal]
    (D : ∀ i, InertialDatum (ν := ν) H (V i)) (φ : GW)
    {Bper : Type uB} [Field Bper] [Algebra K Bper]
    {GP : Type v} [Group GP]
    (σB : GP →* (Bper ≃ₐ[K] Bper)) (tate : ℤ → GP →* Kˣ)
    (ρp : ∀ i, Representation K GP (V i))
    (C : ∀ i, Comparison σB tate (ρp i)) :
    (∃ j : Representation.IntertwiningMap (copiesRep ρ m) carrier,
      ∃ r : Representation.IntertwiningMap carrier (copiesRep ρ m),
        (∀ x, r (j x) = x) ∧
        Continuous (fun gx : GG × Copies V m => copiesRep ρ m gx.1 gx.2) ∧
        (copiesRep ρ m g).charpoly = (∏ i, P i ^ m i).map σE) ∧
    (let S := copiesInertialDatum H D m;
      S.toDatum = copiesDatum (fun i => (D i).toDatum) m ∧
      localCharpoly S.toDatum H φ =
        ∏ i, localCharpoly (D i).toDatum H φ ^ m i ∧
      eulerPolynomial S.toDatum H φ =
        ∏ i, eulerPolynomial (D i).toDatum H φ ^ m i ∧
      Module.finrank K (fixedKernel S.toDatum H) =
        ∑ i, m i * Module.finrank K (fixedKernel (D i).toDatum H) ∧
      LinearMap.trace K _ (fixedRep S.toDatum H φ) =
        ∑ i, (m i : K) * LinearMap.trace K _ (fixedRep (D i).toDatum H φ)) ∧
    (∃ Csel : Comparison σB tate (copiesRep ρp m),
      (∀ z : ℤ,
        weightMultiplicity σB tate Csel z =
          ∑ i, m i * weightMultiplicity σB tate (C i) z) ∧
      (∀ z : ℤ, Nonempty (
        periodSpace σB tate (copiesRep ρp m) z ≃ₗ[K]
          Copies (fun i => periodSpace σB tate (ρp i) z) m))) := by
  refine ⟨?_, ?_, ?_⟩
  · exact selected_global_carrier ρ hρ carrier m c hmc Bcar hBcar σE g P hP
  · exact selected_weil_packets H D m φ
  · exact selected_semilinear_period_packets σB tate ρp C m


/-- The supplied initial semisimple summand is transported to the canonical
selected-copy realization through its explicit equivariant decomposition.  This
is the formal version of the source theorem's statement
`U_{λ₀} ≃ ⨁ᵢ W_{i,λ₀}^{⊕ mᵢ}`. -/
theorem initial_summand_transport
    {K : Type u} [Field K]
    {G : Type v} [Monoid G]
    {I : Type} [Fintype I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    (ρ : ∀ i, Representation K G (V i))
    {W : Type z} [AddCommGroup W] [Module K W]
    (carrier : Representation K G W)
    (m c : I → ℕ) (hmc : ∀ i, m i ≤ c i)
    (Bcar : W ≃ₗ[K] Copies V c)
    (hBcar : ∀ g x, Bcar (carrier g x) = copiesRep ρ c g (Bcar x))
    {U₀ : Type z} [AddCommGroup U₀] [Module K U₀]
    (ρU₀ : Representation K G U₀)
    (e₀ : U₀ ≃ₗ[K] Copies V m)
    (he₀ : ∀ g x, e₀ (ρU₀ g x) = copiesRep ρ m g (e₀ x)) :
    ∃ j : Representation.IntertwiningMap ρU₀ carrier,
```

```
      ∃ r : Representation.IntertwiningMap carrier ρU₀,
        ∀ x, r (j x) = x := by
  let jlin : U₀ →ₗ[K] W :=
    (transportedInclusion hmc Bcar).comp e₀.toLinearMap
  let rlin : W →ₗ[K] U₀ :=
    e₀.symm.toLinearMap.comp (transportedRetraction hmc Bcar)
  let j : Representation.IntertwiningMap ρU₀ carrier :=
    jlin.intertwiningMap_of_isIntertwiningMap _ _ (by
      intro g x
      change transportedInclusion hmc Bcar (e₀ (ρU₀ g x)) =
        carrier g (transportedInclusion hmc Bcar (e₀ x))
      rw [he₀]
      exact transported_inclusion_equivariant ρ carrier hmc Bcar hBcar g (e₀ x))
  let r : Representation.IntertwiningMap carrier ρU₀ :=
    rlin.intertwiningMap_of_isIntertwiningMap _ _ (by
      intro g x
      apply e₀.injective
      simpa only [rlin, LinearMap.comp_apply, LinearEquiv.coe_coe,
        e₀.apply_symm_apply, he₀] using
        transported_retraction_equivariant ρ carrier hmc Bcar hBcar g x)
  refine ⟨j, r, ?_⟩
  intro x
  change rlin (jlin x) = x
  change e₀.symm (transportedRetraction hmc Bcar
      (transportedInclusion hmc Bcar (e₀ x))) = x
  rw [transported_split, e₀.symm_apply_apply]

/-- Frobenius traces of the supplied initial summand determine the same unique
multiplicity packet.  The trace comparison is proved from the explicit
intertwining linear equivalence rather than assumed separately. -/
theorem initial_summand_multiplicities_unique
    {K : Type u} [Field K] [CharZero K]
    {G : Type v} [Monoid G]
    {I : Type} [Fintype I] [DecidableEq I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)]
    (ρ : ∀ i, Representation K G (V i))
    (m : I → ℕ)
    {U₀ : Type z} [AddCommGroup U₀] [Module K U₀]
    [FiniteDimensional K U₀]
    (ρU₀ : Representation K G U₀)
    (e₀ : U₀ ≃ₗ[K] Copies V m)
    (he₀ : ∀ g x, e₀ (ρU₀ g x) = copiesRep ρ m g (e₀ x))
    (sample : I → G)
    (hdet : Matrix.det (fun j i => LinearMap.trace K _ (ρ i (sample j))) ≠ 0) :
    ∀ n : I → ℕ,
      (∀ j, LinearMap.trace K U₀ (ρU₀ (sample j)) =
        LinearMap.trace K _ (copiesRep ρ n (sample j))) → n = m := by
  intro n htrace
  apply selected_multiplicities_unique ρ m sample hdet n
  intro j
  have hconj : e₀.conj (ρU₀ (sample j)) = copiesRep ρ m (sample j) := by
    apply LinearMap.ext
    intro x
    change e₀ (ρU₀ (sample j) (e₀.symm x)) = copiesRep ρ m (sample j) x
    rw [he₀, e₀.apply_symm_apply]
  calc
    LinearMap.trace K _ (copiesRep ρ m (sample j)) =
        LinearMap.trace K _ (e₀.conj (ρU₀ (sample j))) := by rw [hconj]
    _ = LinearMap.trace K U₀ (ρU₀ (sample j)) :=
      LinearMap.trace_conj' (ρU₀ (sample j)) e₀
    _ = LinearMap.trace K _ (copiesRep ρ n (sample j)) := htrace j


/-- Source-level capstone for Paper X Claim 9.2.  The initial semisimple
```

```
summand is supplied in the explicit constituent-copy form stated in the amended
manuscript.  The theorem proves its split realization in the carrier, uniqueness
of the multiplicity vector from the finite Frobenius separator, and preservation
of the strict-compatible local packets at an arbitrary coefficient place.
Applying the universally quantified local conclusion at every coefficient place
with the same `m` constructs the compatible subsystem. -/
theorem compatible_carrier_summand_completion
    {K : Type u} [Field K] [CharZero K] [TopologicalSpace K]
    {GG : Type v} [Monoid GG] [TopologicalSpace GG]
    {I : Type} [Fintype I] [DecidableEq I]
    {V : I → Type u}
    [∀ i, AddCommGroup (V i)] [∀ i, Module K (V i)]
    [∀ i, FiniteDimensional K (V i)] [∀ i, TopologicalSpace (V i)]
    (ρ : ∀ i, Representation K GG (V i))
    (hρ : ∀ i, Continuous (fun gx : GG × V i => ρ i gx.1 gx.2))
    {W : Type z} [AddCommGroup W] [Module K W]
    (carrier : Representation K GG W)
    (m c : I → ℕ) (hmc : ∀ i, m i ≤ c i)
    (Bcar : W ≃ₗ[K] Copies V c)
    (hBcar : ∀ g x, Bcar (carrier g x) = copiesRep ρ c g (Bcar x))
    {U₀ : Type z} [AddCommGroup U₀] [Module K U₀]
    [FiniteDimensional K U₀]
    (ρU₀ : Representation K GG U₀)
    (e₀ : U₀ ≃ₗ[K] Copies V m)
    (he₀ : ∀ g x, e₀ (ρU₀ g x) = copiesRep ρ m g (e₀ x))
    (sample : I → GG)
    (hdet : Matrix.det (fun j i => LinearMap.trace K _ (ρ i (sample j))) ≠ 0)
    {E : Type uB} [CommRing E] (σE : E →+* K)
    (g : GG) (P : I → E[X])
    (hP : ∀ i, (ρ i g).charpoly = (P i).map σE)
    {GW : Type v} [Group GW] [TopologicalSpace GW]
    {ν : GW →* Kˣ} (H : Subgroup GW) [H.Normal]
    (D : ∀ i, InertialDatum (ν := ν) H (V i)) (φ : GW)
    {Bper : Type uB} [Field Bper] [Algebra K Bper]
    {GP : Type v} [Group GP]
    (σB : GP →* (Bper ≃ₐ[K] Bper)) (tate : ℤ → GP →* Kˣ)
    (ρp : ∀ i, Representation K GP (V i))
    (C : ∀ i, Comparison σB tate (ρp i)) :
    (∃ j : Representation.IntertwiningMap ρU₀ carrier,
      ∃ r : Representation.IntertwiningMap carrier ρU₀,
        ∀ x, r (j x) = x) ∧
    (∀ n : I → ℕ,
      (∀ j, LinearMap.trace K U₀ (ρU₀ (sample j)) =
        LinearMap.trace K _ (copiesRep ρ n (sample j))) → n = m) ∧
    (∃ j : Representation.IntertwiningMap (copiesRep ρ m) carrier,
      ∃ r : Representation.IntertwiningMap carrier (copiesRep ρ m),
        (∀ x, r (j x) = x) ∧
        Continuous (fun gx : GG × Copies V m => copiesRep ρ m gx.1 gx.2) ∧
        (copiesRep ρ m g).charpoly = (∏ i, P i ^ m i).map σE) ∧
    (let S := copiesInertialDatum H D m;
      S.toDatum = copiesDatum (fun i => (D i).toDatum) m ∧
      localCharpoly S.toDatum H φ =
        ∏ i, localCharpoly (D i).toDatum H φ ^ m i ∧
      eulerPolynomial S.toDatum H φ =
        ∏ i, eulerPolynomial (D i).toDatum H φ ^ m i ∧
      Module.finrank K (fixedKernel S.toDatum H) =
        ∑ i, m i * Module.finrank K (fixedKernel (D i).toDatum H) ∧
      LinearMap.trace K _ (fixedRep S.toDatum H φ) =
        ∑ i, (m i : K) * LinearMap.trace K _ (fixedRep (D i).toDatum H φ)) ∧
    (∃ Csel : Comparison σB tate (copiesRep ρp m),
      (∀ z : ℤ,
        weightMultiplicity σB tate Csel z =
          ∑ i, m i * weightMultiplicity σB tate (C i) z) ∧
      (∀ z : ℤ, Nonempty (
        periodSpace σB tate (copiesRep ρp m) z ≃ₗ[K]
          Copies (fun i => periodSpace σB tate (ρp i) z) m))) := by
```

```
  refine ⟨?_, ?_, ?_, ?_, ?_⟩
  · exact initial_summand_transport ρ carrier m c hmc Bcar hBcar ρU₀ e₀ he₀
  · exact initial_summand_multiplicities_unique ρ m ρU₀ e₀ he₀ sample hdet
  · exact selected_global_carrier ρ hρ carrier m c hmc Bcar hBcar σE g P hP
  · exact selected_weil_packets H D m φ
  · exact selected_semilinear_period_packets σB tate ρp C m

end PaperX.StrictCompatibleCarrierCompletion
```

18 Companion records and limits

Read `REPORT.md`, `PROOF_AUDIT.md`, `CLAIM_MAP.md`, `CLAIM_STATUS.json`, and `REPRODUCE.md` under `formal_preservation/readable/checked_conditional/`. The later residual attempt remains separately under `formal_preservation/readable/later_residual_attempt/` and `status_records/`. The inherited Block 27 and residue-counting materials remain in `inherited_baseline/`.

The principal reader scripts are `publication/build_publication.py` and `verification/run_checks.py`. Package integrity is checked by `verify_release.py`. The dated final review records failed attempts as well as completed checks. A successful package verification does not replace a mathematical proof review or a new pinned Lean build.

The current monograph includes the recovered 23-entry bibliography with separately recorded identity checks. This guide adds exact internal source routes. Bibliography identity checks do not certify every external theorem. Universal companion existence, unknown carrier existence, unrestricted local attachment, and universal automorphic occupancy are outside the delivered conditional theorem. No new result about those existence problems is claimed.

Tame and Wild Packet Geometry

The formal companion to Chapter XI explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2385

Return to Chapter XI

Reading scope

This companion explains the formal mechanisms and their arithmetic interfaces. The general realization endpoint is Conjecture 66. Detailed proofs, construction notes and evidence records accompany the sources.

CHAPTER 1

Reading the formal companion

Chapter XI develops packet decomposition, finite representation comparisons, actual arithmetic/type adapters, and a conditional comparison interface. This guide makes the logical boundaries visible in the source language. The mathematical text gives the exposition. The visual guide illustrates the constructions. The retained Lean sources provide the formal statements and proofs at their printed hypotheses.

The general arithmetic realization endpoint is presented as **Conjecture 66**. Conjecture 66 isolates the missing realization claim from historical Claim 16.2, indexed as original obligation 66. This separation preserves the proved finite comparison and the proved conditional implication. It records that finite assembly and coherent transport have not yet supplied an actual supercuspidal realization in the original generality.

The source collection includes detailed proofs and construction notes. It distinguishes established results, conditional results and the remaining obligations of Conjecture 66.

1.1 What Lean certifies

Acceptance by Lean checks that the declared conclusion follows from its binders, imported declarations, and foundational axioms. It does not silently instantiate a local field, an arithmetic type, a reference correspondence, or a normalized local-factor convention. Those objects must appear in the source or in a separate mathematical attachment.

A structure may contain an existence field. A theorem that consumes that structure proves an implication using the field. It is still a substantive theorem when the comparison map, descent or uniqueness is constructed. It is not a proof of the structure's arithmetic existence. In particular, `Realization.exists_atom` is an existence input in the formal comparison interface.

1.2 Evidence available for this guide

The catalog identifies the selected modules, source hashes, imports and named declarations. Compiler and axiom records specify the exact formal checks. Producing the PDF is a separate operation and does not itself execute Lean.

> The compiler record covers all 423 selected modules with unchanged source hashes. All 4,342 catalogued names resolved to their recorded owners, including four private or local source names matched through Lean's private-name map. The terminal and the single-candidate comparison supplement are separate checked targets.

Population	Modules	Cataloged names	Logical declarations
Selected sources	423	4,342	7,672
Recovered terminal	1	0	51
One-candidate supplement	1	1	1

The last column is the complete population of logical declarations owned by the named modules in the imported private kernel environment. It includes generated and private logical declarations. It is distinct from the retained lexical catalog, and is not a count of independent mathematical discoveries. The audit separately lists 1,636 compiler-only auxiliary names for the selected sources. These are absent from the kernel constant map.

Every audited declaration's transitive axiom dependencies belong to `Classical.choice`, `Quot.sound`, and `propext`. No forbidden dependency was found. All three aggregate imports passed. Two isolated negative controls were rejected as intended: uncataloged public and private custom axioms in a module with a safely cataloged theorem, and a deliberately unresolved catalog name. These fixtures are not Paper XI modules. The supplement's observed axioms were `Quot.sound` and `propext`.

1.3 How the catalog is selected

The selected formal source contains 423 distinct module paths and 4,342 named lexical declaration commands. Identical repeated files are collapsed. The catalog records the origin and exact bytes of each module.

This is a source census, not a count of independent mathematical discoveries or kernel-generated declarations. Constructors, recursors and unnamed instances are outside this lexical count. The historical “65/66” number concerns original paper obligations, whereas the appendix includes many supporting definitions and lemmas. The conjecture is not counted as a Lean theorem.

1.4 Supplement: the comparison of one supplied candidate

The publication comparison fixes one admissible candidate and one admissible reference. A separating factor system identifies their isomorphism classes if their declared factor profiles agree. The additional module `PaperXI.SingleCandidateComparison` applies `LocalFactorIdentification.class_eq_iff` to exactly these supplied objects.

The expected factor profile is a parameter. H5 is assumed only for this candidate. The proof does not consume `Realization.exists_atom`, does not quantify H5 over every realizer, and does not instantiate an arithmetic correspondence. The separating factor system and admissibility conditions remain explicit inputs. This is a formal presentation of the chapter's conditional implication, not an additional arithmetic existence result.

The complete supplementary module follows. It is counted separately from the 423-module catalog and forty teaching excerpts. Its compiler and axiom record accompanies the sources.

```
import PaperXI.LocalFactorIdentification

/-! The one-candidate local-factor comparison used in the manuscript.
The candidate and its admissibility are supplied. Matching factors are required
only for this candidate. No atom-realization existence field is consumed.
The separating factor system remains an explicit classical input. -/
set_option autoImplicit false
noncomputable section

namespace PaperXI.SingleCandidateComparison
open LocalFactorIdentification

variable {Atom Test Value : Type*} {s : Setoid Atom}

/-- Equality of the declared factor profile identifies this supplied candidate
with the reference in the declared isomorphism quotient. -/
theorem source (S : FactorSystem s Test Value)
    (candidate reference : Atom)
    (hcandidate : S.admissible candidate) (hreference : S.admissible reference)
    (expected : Test → Value)
    (h5 : S.profile candidate = expected)
    (href : S.profile reference = expected) :
    Quotient.mk s candidate = Quotient.mk s reference := by
  apply (class_eq_iff S hcandidate hreference).mpr
  exact h5.trans href.symm

end PaperXI.SingleCandidateComparison
```

1.5 The corrected moving-space phase domain

In odd residue characteristic, the monograph's breakwise formula now applies to $M = \mathrm{im}(g - I)$ and its break summands, with g explicitly symplectic and fixed-point-free on M. The selected theorem `PaperXI.BreakOddPhase.source` already proves this product of moving phases. `BreakMoving` supplies the complement and nondegeneracy facts. A zero moving space contributes one. This source alignment does not add an order-prime-to-p hypothesis and does not modify the retained Lean module.

1.6 Reading literal excerpts

A displayed declaration excerpt retains the exact source text through a top-level `:=`, omitting its subsequent proof or value. Named-argument assignments inside binders do not terminate the excerpt. Structures retain all their fields. The complete declaration is available in the source file. A proof excerpt is a contiguous source slice. Ambient assumptions are explained beside it and retained as raw setup lines in `EXCERPTS.json`. Imported definitions remain necessary.

The display is not a separately compiled replacement module. Unicode supported by the monospace font appears directly. Unsupported glyphs use reversible `[U+XXXX]` tokens. The raw excerpt, display file, conversion map and SHA-256 hashes are included.

Navigation through the five readers

The formal core, the written arithmetic constructions and the research arguments have different evidence. The companion links them while keeping the selected source, terminal replay and single-candidate lemma distinct. None supplies an axiom of arithmetic existence or a count of new discoveries.

Question	Reading route and formal boundary
The finite comparison	Monograph finite tame package. Its finite statement and its arithmetic interpretation retain distinct hypotheses.
The existence field	Monograph Conjecture 66, progress Section 1 and Plate XI.1. The source field `Realization.exists_atom` remains an input.
The pointwise comparison	Comparison of one realized candidate. H5 is about that candidate. The more general formal interface's all-realizers binder is not a stronger publication conjecture.
Ordinary arithmetic progress	Monograph Section 19 and the separate progress paper. The explicit parameter, character, conductor and marking arguments are not newly Lean-certified here.
Worked interpretation	Core visual guide and research atlas. Abstract finite examples are not certificates for an unspecified arithmetic type.

The source catalog identifies all 423 module paths, 4,342 named lexical declarations and forty teaching excerpts. Compiler and axiom records state their own checked targets. A source census and a kernel check have different scopes.

Evidence boundaries

Compiler and same-kernel records certify specified source inputs and dependencies. Producing and reviewing the readers is a separate operation. No independently implemented proof checker is claimed.

The terminal's checked source makes its imports, implicit arguments and standing hypotheses explicit. Its status is separate from the 423-module selection. The target is restricted tame Weil-side attachment. It does not construct a general wild automorphic realizer.

The selected classical-source gap has been closed. The Henniart–Lemaire induction interface has now been checked against the full official text, including direct visual inspection of the relevant normalization formulas. The bounded review preserves its characteristic, rank, genericity and orbit hypotheses. Other Bushnell–Kutzko, conductor, tower and descent inputs retain their own evidence. This source check is not H5 or corrected-character occurrence.

Four open obligations, not four missing formal lemmas

General arithmetic realization requires an actual common host with the declared actions and transports. H5 requires the characterizing factors for that same realizer. The corrected-character branch still needs occurrence in its fixed candidate. Where the stronger native contract is prescribed, the actual square, section and honest-phase certificate must be computed on the chosen type. The research certificate plate deliberately leaves those arithmetic fields uncomputed.

The positive transported-marking contract remains separate. Its hypotheses are not silently strengthened by the native-marking task. Likewise, a failed original carrier does not exclude every permitted enlarged inducing carrier.

Reproducible reading routes

CHAPTER 2

Fourier projectors and Frobenius transport

The cyclic packet is constructed by averaging against an additive character. Idempotence, orthogonality and completeness are consequences of the representation law and character orthogonality. The packet space is the actual range of the computed projector, and Frobenius transports that range by its conjugation law.

2.0.1 The averaging operator

The ambient cyclic representation and hypotheses on the additive character are declared in the source module. The index is a residue label in $\mathbb{Z}/M\mathbb{Z}$. The scalar field is C, so the finite order is invertible. The definition sums actual operators. Later results compute their products rather than postulating a family of orthogonal idempotents.

E01: `PaperXI.FourierProjectors`, source lines 13–15.

Contiguous source proof excerpt.

Source SHA-256: `02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52`.

```
def projector (ρ : Action (K := K) (V := V) (M := M))
    (ψ : AddChar (ZMod M) K) (j : ZMod M) : Module.End K V :=
  PaperXI.FourierCovariance.fourierOperator (operator ρ) ψ j

```

2.0.2 Completeness of the packet resolution

The finite cyclic group and primitive character remain fixed. The sum ranges over all residue labels. Completeness says that every vector is the sum of its packet components. A vanishing packet is permitted, and the theorem does not assert that every residue label occurs.

E02: `PaperXI.FourierProjectors`, source lines 68–77.

Contiguous source proof excerpt.

Source SHA-256: `02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52`.

```
theorem projector_complete (ρ : Action (K := K) (V := V) (M := M))
    (ψ : AddChar (ZMod M) K) (hψ : ψ.IsPrimitive) (hM : (M : K) ≠ 0) :
    (∑ j : ZMod M, projector ρ ψ j) = 1 := by
  have hs (a : ZMod M) : (∑ j : ZMod M, ψ (-j * a)) =
      if a = 0 then (M : K) else 0 := by
    simpa [mul_comm, neg_mul, mul_neg] using character_sum ψ hψ (-a)
  simp only [projector_formula, ← Finset.smul_sum]
  rw [Finset.sum_comm]
  simp_rw [← Finset.sum_smul, hs]
  simp [hM, smul_smul]

```

2.0.3 An actual packet equivalence

The supplied invertible Frobenius operator satisfies the declared conjugation relation with the cyclic inertia action. The multiplication direction in the residue label is fixed by that relation. The operator and its inverse restrict to the two computed packet ranges. Their inverse identities give a linear equivalence, so equal packet dimensions follow from a constructed transport.

E03: `PaperXI.PacketTransport`, source lines 62–66.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dc7282e77c935ca61b8807fe87f4fe2fa372d720a28d0a1c8982c6c749e1a910`.

```
def packetEquiv (ρ : Action (K := ℂ) (V := V) (M := M))
    (F : V ≃ₗ[ℂ] V) (q : (ZMod M)ˣ)
    (h : F.symm.conj (operator ρ 1) = operator ρ (q : ZMod M)) (j : ZMod M) :
    packet ρ j ≃ₗ[ℂ] packet ρ ((q : ZMod M) * j) where
  toFun v :=
```

2.1 Geometric Frobenius convention and the publication repair

The geometric convention is

$$\Phi^{-1} t \Phi = t^q, \qquad \Phi P_j \Phi^{-1} = P_{qj}.$$

Under the other relation $\Phi t \Phi^{-1} = t^q$, the correct label is $q^{-1}j$. The original Proposition 3.1 proof paired that other relation with the geometric label qj. The mathematical text repairs the displayed relation to the geometric convention, preserving its intended packet transport. The checked Lean module already states both conventions separately. No new Lean proof is claimed for this prose convention repair.

2.1.1 The existing geometric covariance theorem

The cyclic order is nonzero. The element q is a unit modulo that order, and the source hypothesis uses the conjugation by F^{-1}. This exact signature and proof support the publication correction. They distinguish the Frobenius direction from its action on projector labels, rather than changing both directions simultaneously.

E04: `PaperXI.TameProjectors`, source lines 45–52.

Contiguous source proof excerpt.

Source SHA-256: `d5881299706cbe60a5cbefc9dc60bb9bdffbc1365228529c181b9a5120b71d8e`.

```
theorem geometric_covariance (ρ : Action (K := ℂ) (V := V) (M := M))
    (F : V ≃ₗ[ℂ] V) (q : (ZMod M)ˣ)
    (h : F.symm.conj (operator ρ 1) = operator ρ (q : ZMod M)) (j : ZMod M) :
    F.conj (projector ρ j) = projector ρ ((q : ZMod M) * j) := by
  have hs := forward_generator_covariance ρ F.symm q h ((q : ZMod M) * j)
  simp only [← mul_assoc, Units.inv_mul, one_mul] at hs
  have hh := congrArg F.conj hs
  simpa only [LinearEquiv.conj_conj_symm] using hh.symm

```

CHAPTER 3

Induction, return scalars and tensor interactions

The induced representation branch uses a degree homomorphism and its degree subgroups. A normalized degree-one element gives coset representatives. Its cyclic action has a closing scalar determined by the inducing character. Tensor interactions are constructed as actual representation equivalences indexed by greatest-common-divisor classes.

3.0.1 The closing scalar is derived

An actual character of the degree subgroup and a degree-one Frobenius element are supplied. The degree n is positive. The scalar evaluates the character at the full return element. It is not an independent certificate whose agreement with the action is assumed.

E05: `PaperXI.InducedCycleInvariants`, source lines 33–33.

Contiguous source proof excerpt.

Source SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
33 def returnScalar : ℂˣ := ξ (degreePeriod φ δ hφ)
34
```

3.0.2 Dimension and characteristic polynomial

The basis of the genuine degree-induced space has already been constructed. Finite dimensionality and the nonzero degree hypotheses occur in the module's setup. The resulting dimension and polynomial record the entire cyclic block, including its closing scalar. Their interpretation as arithmetic local data needs the corresponding group and normalization attachments.

E06: `PaperXI.InducedCycleInvariants`, source lines 75–77.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
75 theorem cycle_invariants :
76     (degreeInduced δ n ξ φ).charpoly = X ^ n - C (returnScalar δ φ hφ n ξ : ℂ) ∧
77     (degreeInduced δ n ξ φ).det = (-1) ^ (n - 1) * (returnScalar δ φ hφ n ξ : ℂ) :=
```

3.0.3 The tensor decomposition is an equivalence

Two positive degrees and two characters of their degree subgroups are supplied. Representatives use φ^{-i}, fixing the direction of the shifted interaction character. There are $\gcd(n,m)$ interaction blocks of degree $\operatorname{lcm}(n,m)$. The source builds an intertwining equivalence of representations, not merely the dimension identity $\gcd(n,m)\operatorname{lcm}(n,m) = nm$.

E07: `PaperXI.TameTensorSource`, source lines 66–69.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `b52fef0aac236edae7a22103dcf267e147fa13606783529fa64e3c4c4e954d3e`.

```
def decomposition : Representation.Equiv
    ((CyclicInduced.degreeInduced ν n ξ).tprod (CyclicInduced.degreeInduced ν m η))
    (Representation.directSum (fun i : Fin (Nat.gcd n m) =>
      CyclicInduced.degreeInduced ν (Nat.lcm n m) (interactionCharacter ν φ n m ξ η i))) :=
```

CHAPTER 4

Finite conductors and the tame comparison package

The finite conductors are sums of codimensions of fixed spaces for a declared subgroup family. Arithmetic Swan and Artin conductors require that this family and its weights match the arithmetic ramification conventions. The finite residue Gauss phase is also a declared character sum before its arithmetic interpretation.

4.0.1 A declared finite ramification sum

The representation, a finite set of depths, a subgroup at each depth and rational weights are supplied. The source does not construct an absolute Weil ramification filtration here. These inputs determine the exact finite sum. The tame vanishing theorem applies when the selected positive-depth groups act trivially.

E08: `PaperXI.TameConductor`, source lines 56–58.

Contiguous source proof excerpt.

Source SHA-256: `808eb443d2a292fc79cd70ce323c369acd62c38e9393ec4c19269dc06d575dcf`.

```
def finiteSwan (π : Representation ℂ G V) (S : Finset ℕ)
    (J : ℕ → Subgroup G) (w : ℕ → ℚ) : ℚ :=
  ∑ r ∈ S, w r * (fixedCodimension π (J r) : ℚ)

```

4.0.2 Proposition 16.1 at finite scope

The exact induced spaces, degree character, cyclic relation, inertia label, finite-depth subgroup data and residue characters are supplied. The additive character is nontrivial. The package derives observations, a tensor equivalence, the total tensor dimension and the magnitude of the Gauss phase. Its magnitude formula retains the exception for the trivial multiplicative character. This finite statement does not certify standard local-Weil factors or an arithmetic epsilon-factor normalization.

E09: `PaperXI.FiniteTameComparison`, source lines 93–107.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `1c4357da32eaade5be0c88670eaa605f47fbc4269253888bbf9f72509d0e428d`.

```
theorem finite_comparison_package (P : Subgroup G)
    (hP : ∀ g ∈ P, ∀ v, degreeInduced δ n ξ g v = v)
    (S : Finset ℕ) (J : ℕ → Subgroup G) (w : ℕ → ℚ)
    (hJP : ∀ r ∈ S, J r ≤ P)
    (m : ℕ) [NeZero m] (η : TameDegree.degreeSubgroup δ m →* ℂˣ)
    (χ : kˣ →* ℂˣ) (ψ : AddChar k ℂ) (hψ : ψ ≠ 1) :
    observed δ ι φ n ξ q hrel S J w χ ψ =
      fromCoordinates n q j (InducedCycleInvariants.returnScalar δ φ hφ n ξ) χ ψ ∧
    Nonempty (Representation.Equiv ((degreeInduced δ n ξ).tprod (degreeInduced δ m η))
      (Representation.directSum (fun i : Fin (Nat.gcd n m) =>
        degreeInduced δ (Nat.lcm n m) (TameTensorSource.interactionCharacter δ φ n m ξ η i)))) ∧
    (∑ i : Fin (Nat.gcd n m), Module.finrank ℂ (DegreeIndSpace δ (Nat.lcm n m)
      (TameTensorSource.interactionCharacter δ φ n m ξ η i))) = n*m ∧
    ‖GaussCharacters.phase χ ψ‖ =
      if χ = 1 then (Real.sqrt (Fintype.card k))⁻¹ else 1 :=
```

CHAPTER 5

Heisenberg actions, central products and phase comparison

A finite Heisenberg action combines translations and modulation operators. Irreducibility is proved from those actual operators. Break assembly uses a genuine quotient central product and a tensor carrier. To extend the bridge through a stabilizer, the retained action on the whole central extension and the scalar discrepancy must both be controlled.

5.0.1 The finite Schrodinger action

The finite Heisenberg group and its central character are fixed by the source's preceding constructions. The carrier is literally a function space. The multiplication law is checked for the displayed operators. Invertibility, central action and irreducibility are subsequent consequences.

E10: `PaperXI.HeisenbergRepresentation`, source lines 14–16.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492`.

```
14 def representation (ψ : AddChar K ℂ) (h2 : (2 : K) ≠ 0) :
15     Representation ℂ (Element K X) (X → ℂ) where
16   toFun a :=
```

5.0.2 Irreducibility from actual translations and modulations

The source supplies the required nondegenerate character pairing. Its invariant-subspace proof uses coordinate projections and spans the full function space. This finite representation theorem supplies a building block. It does not by itself embed the block in a canonical arithmetic simple type.

E11: `PaperXI.HeisenbergRepresentation`, source lines 58–59.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492`.

```
58 theorem irreducible [Fintype X] (ψ : AddChar K ℂ) (hψ : ψ ≠ 1)
59     (h2 : (2 : K) ≠ 0) : Representation.IsIrreducible (representation (X := X) ψ h2) :=
```

5.0.3 The assembled group is a quotient

A finite indexed family of central-extension data shares a declared center. The balanced central subgroup is defined earlier in the module. The quotient construction retains an explicit center even for an empty family. The quotient's representation must descend through the balancing relation, and that descent is proved in the supporting modules.

E12: `PaperXI.CentralProduct`, source lines 39–41.

Contiguous source proof excerpt.

Source SHA-256: `9d1e6a4ece84a862ef75383b66b97af694c9ac2298fc2047a6c4844bc6aa34af`.

```
39 def Group := Ambient (Z := Z) (H := H) [U+29F8] relation D
40
41 instance : _root_.Group (Group D) := inferInstanceAs (_root_.Group (_ [U+29F8] relation D))
42
```

5.0.4 Precisely what break assembly consumes

The finite family includes the actual groups, quotient groups and complex carriers. Its record keeps central extensions, retained lifts, central characters and relevant pairings. An initial tensor intertwiner and a final representation equivalence are not input fields. They are constructed. Dyadic square and retained-group data remain part of the input and cannot be recovered from an alternating commutator form alone.

E13: `PaperXI.FiniteBreakAssembly`, source lines 22–37.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c`.

```
structure Data where
  source : ∀ i, CentralExtension.Datum Z (H i) (A i)
  target : ∀ i, CentralExtension.Datum Z (H' i) (A' i)
  lift : ∀ i, H i ≃* H' i
  lift_center : ∀ i z, lift i ((source i).i z) = (target i).i z
  centralCharacter : Z →* ℂˣ
  nondegenerate : ∀ i, CocycleCommutator.radical
    (CentralExtension.scalarCocycle (source i) centralCharacter) = ⊥
  leftAction : ∀ i, Representation ℂ (H i) (V i)
  rightAction : ∀ i, Representation ℂ (H' i) (W i)
  leftIrreducible : ∀ i, Representation.IsIrreducible (leftAction i)
  rightIrreducible : ∀ i, Representation.IsIrreducible (rightAction i)
  leftCentral : ∀ i z, leftAction i ((source i).i z) =
    (centralCharacter z : ℂ) • LinearMap.id
  rightCentral : ∀ i z, rightAction i ((target i).i z) =
    (centralCharacter z : ℂ) • LinearMap.id

```

5.0.5 Closing the honest stabilizer action

Honest source and target stabilizer actions are supplied together with covariance. The discrepancy is tested on a generating set. The construction kills the scalar discrepancy through the generator phase hypothesis. Matching quotient actions alone would leave this scalar issue unresolved.

E14: `PaperXI.FiniteBreakAssembly`, source lines 111–115.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c`.

```
def equivOfPhase (S : Set C) (hgen : Subgroup.closure S = ⊤)
    (hp : ∀ g ∈ S, discrepancy d α R T hR hT g = 1) :
    Representation.Equiv
      (CyclicBridge.semidirectRepresentation α (sourceAction d) R hR)
      (CyclicBridge.semidirectRepresentation α (targetAction d) T hT) :=
```

CHAPTER 6

Odd phases and dyadic quadratic refinements

The chapter separates odd phases defined by a determinant from dyadic phases defined by a quadratic refinement. The formal certificate computes the declared determinant phase through its characteristic polynomial. Identifying that value with an arithmetic Weil trace or a canonical beta extension requires an additional argument. In residue characteristic two, an alternating form alone does not retain the quadratic data needed for the phase.

6.0.1 The actual polynomial phase certificate

The residue characteristic is an odd prime. A finite-dimensional vector space carries an actual nondegenerate alternating form. The supplied invertible linear operator need not be fixed-point-free for this algebraic identity. The form derives even dimension. Evaluation of the actual characteristic polynomial at one gives the determinant of the difference operator with the declared sign convention. This certifies the defined phase, without assuming that it equals a separately normalized arithmetic phase.

E15: `PaperXI.OddPhaseCertificate`, source lines 46–50.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `783d84147c4f09cbca432ad799b1e57d9bd47da13f3841e92d4f071d49709f2e`.

```
theorem characteristic_certificate (hp : p ≠ 2)
    (b : LinearMap.BilinForm (ZMod p) V) (hb : b.IsAlt) (hbn : b.Nondegenerate)
    (g : V ≃ₗ[ZMod p] V) :
    phase g = quadraticChar (ZMod p) (-1) ^ (Module.finrank (ZMod p) V / 2) *
      quadraticChar (ZMod p) (g.toLinearMap.charpoly.eval 1) :=
```

6.0.2 Refinements form an affine space

Two quadratic refinements have the same bilinear form over the field with two elements. Nondegeneracy is not required for the affine-difference statement. Their difference is uniquely linear. Hence the bilinear form does not choose one refinement. When the arithmetic contract prescribes square data or a whole marked finite extension, that datum must be retained.

E16: `PaperXI.DyadicAffine`, source lines 74–75.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `06fd5b5a57b17060abd22a11fb29592fcd53b06aa09388b92314fe2fd23692e3`.

```
theorem exists_unique_linear_difference (q q₀ : Refinement b) :
    ∃! l : V →ₗ[ZMod 2] ZMod 2, addLinear q₀ l = q :=
```

6.0.3 The exact Arf Gauss sign

The finite index set parametrizes symplectic planes. Each plane has the explicit coefficients of its quadratic refinement in the field with two elements. The integer character sum is $(-1)^{\mathrm{Arf}(q)}2^d$. The proof factors the complete finite sum into plane sums. It is an exact theorem, not an asymptotic or fitted numerical rule.

E17: `PaperXI.DyadicGauss`, source lines 51–58.

Contiguous source proof excerpt.

Source SHA-256: `634fe67a476f515f627455fa43a1ec870a4ccc66b87867cc5d40ee1227c91dea`.

```
theorem gauss_sum (a : ι → F2 × F2) :
    (∑ x, sign (quadratic a x)) = sign (arf a) * 2 ^ Fintype.card ι := by
  classical
  simp only [quadratic, sign_sum]
  rw [← Fintype.prod_sum (fun (i : ι) (x : F2 × F2) => sign (plane (a i) x))]
  simp_rw [plane_gauss]
  rw [Finset.prod_mul_distrib, ← sign_sum]
  simp [arf]

```

6.0.4 Counting refinements with a prescribed Arf invariant

The supplied symplectic basis identifies the finite-dimensional nondegenerate alternating space. The source separately retains dimension zero and the odd-Arf count. The formulas count the complete refinement space, not a sampled collection. An isometry preserving only the bilinear form need not preserve the prescribed Arf invariant or its phase.

E18: `PaperXI.DyadicRefinementCounts`, source lines 42–44.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `5d9cdc46b7bde9d4f7ceed89b2a7a68ec38f36fee3142300f40b6a8d5301838b`.

```
theorem even_refinement_count :
    (Fintype.card {q : Refinement (symplectic (ι := ι)) // basisArf q = 0} : ℚ) =
      (2 : ℚ) ^ ((Fintype.card ι : ℤ) - 1) * ((2 : ℚ) ^ Fintype.card ι + 1) :=
```

CHAPTER 7

Matrix holonomy and global section equivalence

Local packet comparisons can be transported around a graph. A rooted spanning tree determines root paths. The remaining original edges become chords. Full matrix compatibility on the resulting fundamental cycles is the necessary and sufficient obstruction to a based global gauge.

7.0.1 The rooted chord criterion

The graph is a connected quiver with invertible edge transports. A root comparison is fixed. The cycle condition is an equality of the full transported linear maps. Trace, determinant and projective cycle tests are weaker than this matrix identity. The source constructs a gauge precisely when the chord tests hold and derives compatibility on every edge.

E19: `PaperXI.GraphGauge`, source lines 155–159.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `1750d323d264d4760717c62ab3aa154cbebefcdd24e80cc76ef22d7d8f5a572b`.

```
theorem connected_based_iff_chords (U : V r ≃ₗ[K] W r)
    (hc : GraphTree.Connected (α := α)) :
    (∃ F : ∀ v, V v ≃ₗ[K] W v, F r = U ∧ EdgeCompatible A B F) ↔
      ∀ e : GraphTree.Chord (GraphTree.ofConnected r hc),
        CycleCompatible A B U (GraphTree.rootPath (GraphTree.ofConnected r hc)) e.val :=
```

7.0.2 The space of compatible sections

Each vertex has a vector space fiber and each directed edge carries a linear equivalence. A section satisfies every edge equation. The space is an actual linear subspace of the product of fibers. It can be zero. A nonzero section space, an irreducible section representation, and an arithmetic supercuspidal atom are separate assertions.

E20: `PaperXI.GluedPacketSections`, source lines 22–26.

Contiguous source proof excerpt.

Source SHA-256: `5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf`.

```
def sections : Submodule K (∀ v, V v) where
  carrier s := ∀ {u v : α} (e : u → v), A e (s u) = s v
  zero_mem' := by intro u v e; simp
  add_mem' := by intro a b ha hb u v e; simpa using (map_add (A e) (a u) (b u)).trans (congrArg₂ (· + ·) (ha e) (hb e))
  smul_mem' := by intro c a ha u v e; simpa using (map_smul (A e) c (a u)).trans (congrArg (c • ·) (ha e))

```

7.0.3 Gluing produces a representation equivalence

Covariance of the source and target edge systems is supplied. A root intertwiner and the full fundamental-cycle compatibility equations are supplied. The resulting equivalence acts on the global spaces of compatible sections. Its existence is proved by transporting the root intertwiner, rather than stored as a field in a comparison record.

E21: `PaperXI.GluedPacketSections`, source lines 138–144.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf`.

```
def fromChords {r : α} (hc : GraphTree.Connected (α := α))
    (hA : RepresentationGraph.EdgeEquivariant A ρ)
    (hB : RepresentationGraph.EdgeEquivariant B τ)
    (U : Representation.Equiv (ρ r) (τ r))
    (hch : ∀ e : GraphTree.Chord (GraphTree.ofConnected r hc),
      CycleCompatible A B U.toLinearEquiv (GraphTree.rootPath (GraphTree.ofConnected r hc)) e.val) :
    Representation.Equiv (sectionRepresentation A ρ hA) (sectionRepresentation B τ hB) :=
```

CHAPTER 8

Monodromy and conditional atomic reconstruction

Finite atomic results can propagate through a supplied decomposition and a supplied ordered-quotient interface. The reconstruction theorem constructs sorting, atomic replacement and descent. The external decomposition and ordered Langlands-quotient existence statements remain printed hypotheses.

8.0.1 The monodromy normalization

A twisting character ν and a representation are fixed. Copy i has the twisted action $\nu(g)^i\rho(g)$, and the nilpotent operator shifts copy index. The declared relation is $R(g)NR(g)^{-1} = \nu(g)^{-1}N$. The inverse on the twisting character fixes the convention. Changing it would change the normalization.

E22: `PaperXI.MonodromyAtom`, source lines 64–66.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `f53d79ddc74c43995955bc45c8bce600d821bb6d118e9426d85e8478c0829bf2`.

```
theorem covariance (ν : G →* ℂˣ) (ρ : Representation ℂ G V) (m : ℕ) (g : G) :
    action ν ρ m g * monodromy m =
      (ν g : ℂ)⁻¹ • (monodromy m * action ν ρ m g) :=
```

8.0.2 Source decomposition is an explicit interface

An isomorphism relation on source blocks is fixed. The record specifies the allowed displays and their intrinsic classwise uniqueness. The formal reconstruction does not derive classification of all arithmetic source objects from arbitrary finite data. It uses this interface to make replacement independent of display choices.

E23: `PaperXI.AtomicReconstruction`, source lines 17–21.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
structure Decomposition {P : Type*} (parameterIso : Setoid P) (A : Type*) where
  displays : P → List (Block A) → Prop
  exists_display : ∀ p, ∃ xs, displays p xs
  intrinsic : ∀ p q xs ys, parameterIso.r p q →
    displays p xs → displays q ys → xs.Perm ys

```

8.0.3 The reconstructed map and its uniqueness

The source decomposition interface, ordered target-quotient interface and atomic replacement map are supplied. Ordering uses the specified target data. The candidate and choice independence are constructed. This conditional theorem remains useful when each interface is instantiated by separate mathematics, but it is not an unrestricted correspondence existence theorem.

E24: `PaperXI.AtomicReconstruction`, source lines 143–144.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
theorem formal_atomic_reconstruction (f : A → B) (D : Decomposition sP A)
    (L : OrderedQuotient B T) : ∃! F : Quotient sP → T, Reconstructs f D L F :=
```

CHAPTER 9

Local rings, ramification projectors and normalized Frobenius

The later local-field branch attaches finite residue extensions and normalized geometric Frobenius to actual algebraic objects. Lower ramification projectors average genuine finite groups. Their successive differences form depth-detail channels. A degree-only quotient forgets inertia, and the source proves exactly which detail information then vanishes.

9.0.1 Lower congruence groups

The action is on a supplied local ring, with the powers of its maximal ideal. In a finite Galois DVR application these agree with the usual nonnegative lower groups under the stated dictionary. The generic definition is based on congruences of the actual action. It is not asserted to be an absolute Weil ramification filtration without the arithmetic attachment.

E25: `PaperXI.ValuationRamification`, source lines 15–28.

Contiguous source proof excerpt.

Source SHA-256: `92ddb509dc327389c2168f44ddfd5f7e16f4315af16835a54398077fe81146ca`.

```
def lower (i : ℕ) : Subgroup D where
  carrier := {g | ∀ x : R, g • x - x ∈ (IsLocalRing.maximalIdeal R) ^ (i + 1)}
  one_mem' := by intro x; simp
  mul_mem' := by
    intro g h hg hh x
    have ha := hg (h • x)
    have hb := hh x
    have hab := Ideal.add_mem ((IsLocalRing.maximalIdeal R) ^ (i+1)) ha hb
    simpa only [sub_add_sub_cancel, mul_smul] using hab
  inv_mem' := by
    intro g hg x
    have hh := hg (g⁻¹ • x)
    have hh' := ((IsLocalRing.maximalIdeal R) ^ (i+1)).neg_mem hh
    simpa using hh'

```

9.0.2 An exact depth-detail resolution

The finite-group ramification filtration, representation and stabilization hypothesis are supplied. Averaging is over actual finite subgroups. Nested fixed spaces give nested projectors. Their successive differences are orthogonal detail projectors, and the telescope reconstructs the identity at the stated stabilized depth.

E26: `PaperXI.RamificationPolyphase`, source lines 311–313.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `2f34bd694def457221d6f00b874c8b54547dcc95ee249e3afcf34d71b5b64549`.

```
theorem identity_eq_inertia_add_details {N : ℕ} (hN : lower R D N = ⊥) :
    (1 : Module.End ℂ V) = depthProjector (R := R) (D := D) (F := F) (ρ := ρ) 0 +
      ∑ i ∈ Finset.range N, detailProjector (R := R) (D := D) (F := F) (ρ := ρ) i :=
```

9.0.3 The unconditional quotient target is the degree range

The residue–Frobenius pullback has a degree homomorphism whose kernel is inertia. No degree-one lift is needed for this first-isomorphism statement. The quotient is equivalent to the image of degree. To replace that image by all integer degrees, the source adds a hypothesis of a degree-one Frobenius lift or a subsequent construction that supplies it.

E27: `PaperXI.RamifiedUnramifiedQuotient`, source lines 42–43.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `755d457b0f11dfbac94eaee10326fa290bd5b6a331d8883bb8ce26811f00a747`.

```
42 noncomputable def quotientEquivDegreeRange :
43     Pullback r F [U+29F8] Inertia r F ≃* (degree r F).range :=
```

9.0.4 The constructed normalized quotient

The normalized-Frobenius continuation uses the geometric lift constructed in its preceding modules. Its exact local-field and residue hypotheses are in the source setup. This checked continuation fills the bounded normalized quotient step. Its retained report explicitly leaves full arithmetic/automorphic integration unestablished.

E28: `PaperXI.NormalizedFrobeniusQuotient`, source lines 79–79.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `917cfa62d1b914d7d47a3c31f682a52d3338dc040a1606450f1cff71ca075b1f`.

```
79 def quotientEquivIntegerDegree : Weil E L [U+29F8] Inertia E L ≃* Multiplicative ℤ :=
```

CHAPTER 10

Arithmetic type and induction adapters

The arithmetic branch uses actual local fields, continuous coordinate equivalences, literal subgroups, original characters, and induction carriers. Generic support and intertwining arguments derive irreducibility once the type dictionary is supplied. Later continuations reduce some dictionary leaves to quotient forms and actual lattice algebra.

10.0.1 A candidate from three arithmetic profile leaves

The ambient setup fixes local fields F, E, coordinates, the literal subgroup D, its character θ, the prescribed κ, residue representation σ, and the original tensor carrier. The displayed hypotheses keep trace support, character covariance and the oriented ambient index identity explicit. The proof derives the required Heisenberg intersection dimension and feeds it into the inherited candidate constructor. It preserves the original carrier and representations. The existence of canonical type data and the actual truth of its retained source hypotheses are separate mathematical attachments.

E29: `PaperXI.ActualProfileCandidate110`, source lines 34–88.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dd434e182eec32397e64a3e6a43193cc56a3c965eff65c4e36e14f4bd0b2c27b`.

```
theorem candidate_exists_from_original_profiles
    (hF : Continuous (algebraMap F E))
    (hDo : IsOpen (D : Set (GL n F)))
    (hB : ∀ u : GL m [U+1D4AA][E],
      (coordinateAut e).symm (integralMatrix F E m u) ∈ CharacterNormalizer61.normalizer D θ)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E m n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E m n e) a * (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E m n e D θ).subtype)] [IsIrreducible σ]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E m n e D θ) κ g)
    (hσc : CuspidalCartan105.Cuspidal σ)
    (hc : ActualCharacterData108.Contains F E m n e D θ hDc κ)
    (hθsupport : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ j₁ j₂ : compact F E m n e D θ, ∃ y : GL m E,
        g = (j₁ : GL n F) * centralizerMap F E m n e y * (j₂ : GL n F))
    (hz : ∀ x : firstSubgroup F E m n e D θ,
      x ∉ actualScalarSubgroup F E m n e D θ →
      Representation.character (ActualIntersection108.restriction F E m n e D θ κ) x = 0)
    (hcov : ∀ y : GL m E,
      ∀ p : pairs (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y),
      ∀ h₁ : left (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y) p ∈ actualScalarSubgroup F E m n e D θ,
      ∀ h₂ : right (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y) p ∈ actualScalarSubgroup F E m n e D θ,
      actualScalarCharacter F E m n e D θ
        (left (ActualIntersection108.embedding F E m n e D θ)
          (centralizerMap F E m n e y) p, h₁) =
      actualScalarCharacter F E m n e D θ
        (right (ActualIntersection108.embedding F E m n e D θ)
          (centralizerMap F E m n e y) p, h₂))
    (hindex : ∀ y : GL m E,
      ((ActualIntersection108.embedding F E m n e D θ).range ⊓
        (ActualIntersection108.embedding F E m n e D θ).range.comap
          (MulAut.conj (centralizerMap F E m n e y)).toMonoidHom).relIndex
```

```
            (ActualIntersection108.embedding F E m n e D θ).range =
      (D ⊓ D.comap (MulAut.conj (centralizerMap F E m n e y)).toMonoidHom).relIndex D) :
    FiniteDimensional ℂ (W ⊗[ℂ] U) ∧
    ∃ Λ : Representation ℂ (extended F E m n e D θ) (W ⊗[ℂ] U),
      (∀ j : compact F E m n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n (scalarMatrixEmbedding F E m n e)
          (compact F E m n e D θ) j) =
          CompactTypeTensor58.action (reduction F E m n e D θ hmodel) κ σ j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E m n e D θ) Λ =
        (extended F E m n e D θ : Set (GL n F)) ∧
      Function.Injective (TensorTypeOperators70.realize (extended F E m n e D θ) Λ) ∧
      Nontrivial (IndV (extended F E m n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E m n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E m n e D θ).subtype Λ) :=
```

10.0.2 The residue-rank-one specialization

The residue rank is Fin 1. The original character, carrier, support dictionary, beta-extension intertwining and occurrence input remain fixed. The finite cuspidal residue factor and several normalization/support leaves are derived in this branch. It still does not manufacture every canonical simple character or establish the entire original claim.

E30: `PaperXI.RankOneSourceCandidate110`, source lines 29–60.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `7f57ede141636d625efea3d9b67e905b1f34bcd5dd6c475f31a7e2ec2a7b2928`.

```
theorem candidate_exists
    (hF : Continuous (algebraMap F E)) (hDo : IsOpen (D : Set (GL n F)))
    (hJ : J₁ ≤ matrixCongruence F E (Fin 1) n e)
    (hJN : J₁ ≤ CharacterNormalizer61.normalizer D θ)
    (hθcentralizer : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ a : J₁, ∃ b : (centralizerMap F E (Fin 1) n e).range,
      ∃ c : J₁, (a : GL n F) * (b : GL n F) * (c : GL n F) = g)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E (Fin 1) n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E (Fin 1) n e) a *
            (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E (Fin 1) n e D θ).subtype)]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E (Fin 1) n e D θ) κ g)
    (hc : ActualCharacterData108.Contains F E (Fin 1) n e D θ hDc κ) :
    FiniteDimensional ℂ (W ⊗[ℂ] ℂ) ∧
    ∃ Λ : Representation ℂ (extended F E (Fin 1) n e D θ) (W ⊗[ℂ] ℂ),
      (∀ j : compact F E (Fin 1) n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n
          (scalarMatrixEmbedding F E (Fin 1) n e)
          (compact F E (Fin 1) n e D θ) j) =
          (κ.tprod (Representation.trivial ℂ (compact F E (Fin 1) n e D θ) ℂ)) j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E (Fin 1) n e D θ) Λ =
        (extended F E (Fin 1) n e D θ : Set (GL n F)) ∧
      Nontrivial (IndV (extended F E (Fin 1) n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E (Fin 1) n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E (Fin 1) n e D θ).subtype Λ) :=
```

10.1 Why orientation matters

For the literal pair group, $\iota(b) = g\iota(a)g^{-1}$. The first coordinate therefore belongs to

$$I_g = \{a : g\iota(a)g^{-1} \in \iota(K)\}, \qquad \iota(I_g) = \iota(K) \cap g^{-1}\iota(K)g.$$

The scalar intersection uses the same orientation. Under whole-action irreducibility and the original scalar/trace profile,

$$d^2 = [K : H].$$

The positive oriented index identity $[K : I_g] = [H : C_g]$ then yields $[I_g : C_g] = d^2$, giving the one-dimensional intertwining space. Schur is applied to the whole irreducible action, not to arbitrary intersection restrictions. Infinite subgroup index is encoded by zero in the relevant natural-valued index API, so positivity is essential to cancellation.

CHAPTER 11

Morita lattices and exact one-plus comparisons

The centralizer coefficient ring acts inside a potentially noncommutative endomorphism ring. It must not be treated as central in the full ambient algebra. Morita corner arguments produce bases, coordinate quotient equivalences and actual additive indices. Multiplicative one-plus indices require genuine group closure and a common deep ideal.

11.0.1 The actual Cartan index

Actual ring homomorphisms embed centralizer matrices in the endomorphism ring. The finite/full lattice and radical-commutation conditions are explicit. The index follows from actual coordinate conjugation and the preceding quotient computation. A numerical index formula is not inserted as a premise, but the arithmetic lattice instance must satisfy the printed hypotheses.

E31: `PaperXI.MoritaActualCartan110`, source lines 137–152.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dd18d3a60333f0e737ef4ff22c34db295f49460de701a9fddf9fcd3dc22d293a`.

```
theorem relIndex_of_actual_cartan [Nonempty m]
    (L : Submodule [U+1D4AA][E] (Matrix m m (Module.End F E)))
    (hL : IsMatrixBimodule (coefficientMap F E) L.toAddSubgroup)
    (hfg : L.FG)
    (hfull : Submodule.span F (L : Set (Matrix m m (Module.End F E))) = ⊤)
    (hcomm : RadicalCommutes F E m L.toAddSubgroup)
    (y : GL m E) (c : Eˣ) (l r : GL m [U+1D4AA][E]) (a : m → ℕ)
    (hy : y = GeneralLinearGroup.scalar m c *
      (ActualIntegralGeneralLinear.embedding E m l *
        PaperXI.SplitRadical105.cartan (ActualLocalUnitDecomposition.uniformizer E : E)
          (ActualLocalUnitDecomposition.uniformizer_ne_zero E) a *
            ActualIntegralGeneralLinear.embedding E m r)) :
    (L.toAddSubgroup ⊓ L.toAddSubgroup.map
      (unitConjugation (centralizerEmbedding F E m y)).toAddMonoidHom).relIndex
        L.toAddSubgroup =
      Nat.card [U+1D4C0][E] ^ (Module.finrank F E * ∑ i, ∑ j, ((a i : ℤ) - (a j : ℤ)).toNat) :=
```

11.0.2 From additive lattices to unit groups

The source keeps the original integral-order ring and its actual one-plus subgroups. A common ideal and a finite quotient allow the exact coset comparison. An arbitrary additive lattice is not automatically a subgroup under $1 + x$, and a conjugate lattice need not be an ideal of the full field algebra. The theorem's carrier and group hypotheses prevent either substitution.

E32: `PaperXI.OnePlusIntersection110`, source lines 179–189.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `512e3b7a894d7db7474f8586c4c652dddd45bfd98197f637f7be8e5a8fec3e7e`.

```
theorem conjugate_intersection_relIndex_eq (f : R →+* S) (y : Sˣ) (I : Ideal R)
    (L : AddSubgroup R) (K : Subgroup Rˣ)
    (hK : ∀ g : K, ((g : Rˣ) : R) - 1 ∈ L)
    (hL : ∀ x : L, ∃ g : K, ((g : Rˣ) : R) = 1 + (x : R))
    (hI : I.toAddSubgroup ≤ pullbackLattice f L
```

```
      (L.map ((innerRing y).toRingHom.comp f).toAddMonoidHom))
    [Finite (L [U+29F8] I.toAddSubgroup.addSubgroupOf L)] :
    (K.map (Units.map ((innerRing y).toRingHom.comp f).toMonoidHom)).relIndex
       (K.map (Units.map f.toMonoidHom)) =
      (L.map ((innerRing y).toRingHom.comp f).toAddMonoidHom).relIndex
        (L.map f.toAddMonoidHom) :=
```

11.0.3 The original carrier after dictionary reductions

The literal type quotient/form, actual one-plus groups, Morita lattices and their coordinate matches are supplied. The source fixes the original character and beta extension. Trace support, covariance, finite quotient, degree-square and index equality are derived from those lower-level instances. The canonical type dictionary and arithmetic category/factor interpretation still require an actual instance.

E33: `PaperXI.ActualSourceCandidate111`, source lines 40–84.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `c802c3b3d1be72ca3d961a4edaa42cca066eb81cbc902a0a0abe9eee58d60005`.

```
theorem candidate_exists_from_source_dictionary
    (hF : Continuous (algebraMap F E))
    (hDo : IsOpen (D : Set (GL n F)))
    (hB : ∀ u : GL m [U+1D4AA][E],
      (coordinateAut e).symm (integralMatrix F E m u) ∈ CharacterNormalizer61.normalizer D θ)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E m n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E m n e) a * (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E m n e D θ).subtype)] [IsIrreducible σ]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E m n e D θ) κ g)
    (hσc : CuspidalCartan105.Cuspidal σ)
    (hc : ActualCharacterData108.Contains F E m n e D θ hDc κ)
    (hθsupport : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ j₁ j₂ : compact F E m n e D θ, ∃ y : GL m E,
        g = (j₁ : GL n F) * centralizerMap F E m n e y * (j₂ : GL n F))
    (τ : ((firstSubgroup F E m n e D θ) [U+29F8]
      actualScalarSubgroup F E m n e D θ) ≃* Multiplicative X)
    (B : X →ₗ[k] X →ₗ[k] k) (ψ : AddChar k ℂˣ) (hψ : ψ ≠ 1)
    (hnd : ∀ v : X, (∀ u : X, B u v = 0) → v = 0)
    (hform : ∀ x y : firstSubgroup F E m n e D θ,
      ψ (B (coordinate (actualScalarSubgroup F E m n e D θ) τ x)
        (coordinate (actualScalarSubgroup F E m n e D θ) τ y)) =
      characterCommutator (actualScalarSubgroup F E m n e D θ) τ
        (actualScalarCharacter F E m n e D θ) x y)
    (hθcentralizer : ∀ y : GL m E, NormalizerIntertwining60.IntertwinedAt D
      (MonomialInduction.scalarRepresentation θ) (centralizerMap F E m n e y))
    (data : ActualLatticeData F E m n e D θ) :
    FiniteDimensional ℂ (W ⊗[ℂ] U) ∧
    ∃ Λ : Representation ℂ (extended F E m n e D θ) (W ⊗[ℂ] U),
      (∀ j : compact F E m n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n (scalarMatrixEmbedding F E m n e)
          (compact F E m n e D θ) j) =
          CompactTypeTensor58.action (reduction F E m n e D θ hmodel) κ σ j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E m n e D θ) Λ =
        (extended F E m n e D θ : Set (GL n F)) ∧
      Function.Injective (TensorTypeOperators70.realize (extended F E m n e D θ) Λ) ∧
      Nontrivial (IndV (extended F E m n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E m n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E m n e D θ).subtype Λ) :=
```

CHAPTER 12

Complex scalar cochains and honest cyclic actions

Equal scalar commutators determine a complex scalar cocycle class under the stated divisible-coefficient hypotheses. This can construct an actual representation of a supplied group. The original retained finite-group lift and the arithmetic source attachment remain separate when more structure is prescribed.

12.0.1 What equal commutators supply over complex units

The two normalized cocycles have coefficients in $\mathbb{C}^\times$. Their commutators agree. A cochain is constructed whose coboundary gives the cocycle ratio. This scalar-pushout result does not identify distinct marked finite central extensions such as the dyadic retained-group examples.

E34: `PaperXI.ScalarCocycle112`, source lines 171–174.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `351ac3d1e21afeb0a8897f6eea1ba99f83bb63eaeee390eebefa44c1dfbb0f4e`.

```
theorem same_commutator_exists_cochain
    (c d : Cocycle A ℂˣ) (h : ∀ a b, commutator c a b = commutator d a b) :
    ∃ s : A → ℂˣ, s 1 = 1 ∧
      ∀ a b, c.coeff a b = (s a * s b / s (a * b)) * d.coeff a b :=
```

12.0.2 The equivariant actual-group action

The actual groups and their full automorphisms are supplied. The scalar bridge has been corrected using the explicit cyclic coprimality conditions. The result retains the original theta restriction and intertwines the prescribed packet generator. It requires full actions, not only an action on the symplectic quotient.

E35: `PaperXI.EquivariantRepresentation113`, source lines 67–91.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `f1cf8ff3343f2647c0994270f6de5bc3478925a25406ea3a221861807b08b6c9`.

```
theorem exists_equivariant_actual_action
    (D : Subgroup G) [D.Normal] (θ : D →* ℂˣ)
    (hinv : TypeSideDescent.Invariant D θ)
    (q : G →* A) (hker : q.ker = D)
    (t : A → G) (ht : ∀ a, q (t a) = a) (ht1 : t 1 = 1)
    (H : Subgroup P) [H.Normal] (χ : H →* ℂˣ)
    (hχinv : TypeSideDescent.Invariant H χ)
    (r : P →* A) (hkerP : r.ker = H)
    (u : A → P) (hu : ∀ a, r (u a) = a) (hu1 : u 1 = 1)
    (ρ : Representation ℂ P V) [IsIrreducible ρ]
    (hp : ∀ h : H, ρ h = (χ h : ℂ) • LinearMap.id)
    (α : G ≃* G) (γ : P ≃* P) (β : A ≃* A)
    (hq : ∀ g, q (α g) = β (q g)) (hr : ∀ p, r (γ p) = β (r p))
    (hθ : ∀ d : D, θ (subgroupMap D q hker α β hq d) = θ d)
    (hχ : ∀ h : H, χ (subgroupMap H r hkerP γ β hr h) = χ h)
    (U : V ≃ₗ[ℂ] V) (hU : ∀ p v, U (ρ p v) = ρ (γ p) (U v))
    (n m : ℕ) (hα : α ^ n = 1) (hγ : γ ^ n = 1)
    (hm : ∀ a : A, a ^ m = 1) (hcop : n.Coprime m)
    (hcomm : ∀ a b,
      sectionCocycle D θ q hker t ht a b / sectionCocycle D θ q hker t ht b a =
        sectionCocycle H χ r hkerP u hu a b / sectionCocycle H χ r hkerP u hu b a) :
```

```
    ∃ Ξ : Representation ℂ G V,
      IsIrreducible Ξ ∧ TypeSideDescent.Contains D θ Ξ ∧
        (∀ d : D, Ξ d = (θ d : ℂ) • LinearMap.id) ∧
        (∀ g v, U (Ξ g v) = Ξ (α g) (U v)) :=
```

12.0.3 Determinant-one normalization

The generator implements the supplied full automorphism on the same irreducible carrier. Its cyclic order and carrier dimension satisfy the required coprimality. Scalar rescaling can impose order and determinant normalization with uniqueness at the stated scope. The theorem does not identify the implementer with the independently prescribed arithmetic beta extension.

E36: `PaperXI.CyclicNormalization113`, source lines 167–173.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `607587d94a544d2c4e014d81b21e4c9241d852feddfaa826df57f9715d396ce7`.

```
theorem normalized_generator_existsUnique (ρ : Representation ℂ P V) [IsIrreducible ρ]
    (α : MulAut P) (A : V ≃ₗ[ℂ] V)
    (hA : ∀ p v, A (ρ p v) = ρ (α p) (A v))
    (n : ℕ) (hn : 0 < n) (hα : α ^ n = 1)
    (hcop : Nat.Coprime n (Module.finrank ℂ V)) :
    ∃! B : V ≃ₗ[ℂ] V, B ^ n = 1 ∧ LinearEquiv.det B = 1 ∧
      ∀ p v, B (ρ p v) = ρ (α p) (B v) :=
```

CHAPTER 13

Conjecture 66 and the surviving conditional comparison

The finite bridge, coherent graph equivalence, arithmetic realization and local-factor characterization are distinct assertions. Conjecture 66 concerns general arithmetic realization. Conditional comparison is a theorem once actual realization and the stated factor characterization are supplied.

13.0.1 The factor profile must characterize the declared range

An atom type, its isomorphism setoid, a test type and value type are fixed. The record contains admissibility, invariant profiles and separation on that admissible range. The separation field is the external characterization input. In an arithmetic application, all normalizations and the original characterizing family must be instantiated. A finite-group character instance is an example of the abstract interface, not a local-Weil converse theorem.

E37: `PaperXI.LocalFactorIdentification`, source lines 14–18.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
structure FactorSystem {Atom : Type*} (s : Setoid Atom) (Test Value : Type*) where
  admissible : Atom → Prop
  profile : Atom → Test → Value
  invariant : ∀ a b, s.r a b → profile a = profile b
  separates : ∀ a b, admissible a → admissible b → profile a = profile b → s.r a b

```

13.0.2 Existence is an explicit arithmetic field

The factor system and allowed datum type are fixed. A realizes relation connects each datum to possible actual atoms. The record's `exists_atom` supplies existence. Neither atom uniqueness nor reference agreement is stored here. General arithmetic existence of this record is the unresolved endpoint.

E38: `PaperXI.LocalFactorIdentification`, source lines 25–28.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
structure Realization (D : Type*) where
  realizes : D → Atom → Prop
  exists_atom : ∀ d, ∃ a, realizes d a
  admissible : ∀ d a, realizes d a → S.admissible a

```

13.0.3 H5 acts on the same realized candidate

An actual realization is supplied. The realized atom's profile and the reference atom's profile agree with the same expected profile on the original test family. Separation then proves equality of atom classes. H5 must refer to this same chosen completion. Replacing it with a factor identity for another candidate would not discharge the displayed hypothesis.

E39: `PaperXI.LocalFactorIdentification`, source lines 49–55.

Contiguous source proof excerpt.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
theorem candidate_eq_reference (d : D) (reference : Atom)
    (hr : S.admissible reference) (expected : Test → Value)
    (hreal : ∀ a, R.realizes d a → S.profile a = expected)
    (href : S.profile reference = expected) :
    candidate S R d = Quotient.mk s reference := by
  apply (class_eq_iff S (R.admissible d _ (chosenAtom_realizes S R d)) hr).mpr
  exact (hreal _ (chosenAtom_realizes S R d)).trans href.symm
```

13.0.4 The full formal conditional signature

The finite source data construct a tensor bridge. Honest actions, phase compatibility and matrix holonomy supply the coherent finite comparison. The allowed target diagram, an arithmetic realization, admissibility and factor profiles are then supplied explicitly. The conclusion contains a concrete compatible gauge, an equivalence of global section representations, and equality of the realized atom class with the reference. The source does not assert realization of every arbitrary finite datum. Its theorem name and successful Lean status must be read together with these hypotheses.

E40: `PaperXI.ConditionalAtomicComparison`, source lines 127–143.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ce8d1ea0f8658d4d968aeb6665d5f1d3e4cc6bb1d472ebfbfbce6215f1103a58`.

```
theorem source {Atom Test Value : Type*} {atomIso : Setoid Atom}
    (sys : FactorSystem atomIso Test Value)
    (allowed : Set (Diagram X ℂ (CentralProduct.Group d.source [U+22CA][α] C)
      (PiTensorProduct ℂ Wb)))
    (ha : targetDiagram a0 ∈ allowed)
    (realization : Realization sys allowed)
    (reference : Atom) (href : sys.admissible reference)
    (expected : Test → Value)
    (hfactors : ∀ b, realization.realizes (targetDiagram a0, ha) b →
      sys.profile b = expected)
    (hreference : sys.profile reference = expected) :
    (∃ F : X → PiTensorProduct ℂ Vb ≃ₗ[ℂ] PiTensorProduct ℂ Wb,
      F a0.root = FiniteBreakAssembly.tensorBridge d ∧
      EdgeCompatible a0.sourceEdge a0.targetEdge F ∧
      RepresentationGraph.Intertwines a0.left a0.right F) ∧
    Nonempty (Representation.Equiv (leftSections a0) (rightSections a0)) ∧
      candidate sys realization (targetDiagram a0, ha) = Quotient.mk atomIso reference :=
```

13.1 The publication statement

Fix the declared class $\mathcal{A}$ of wild atoms and the paper's normalizations. For every $\rho \in \mathcal{A}$ with supplied H1–H4 assembled finite packet datum d_ρ, Conjecture 66 asserts that there exists one actual admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(F)$ together with an arithmetic realization witness $R(d_\rho, \pi)$, recovering the prescribed packet and preserving the declared stabilizer action, central-extension/phase data, and compatible transports. This witness uses actual arithmetic inducing data and explicit comparison/recovery. It does not require the whole assembled packet to be the original maximal-simple-type quotient.

Given that same π and realization witness, the conditional comparison proposition assumes original H5 for π on the stipulated characterizing family with the fixed normalization. The supplied characterization then gives $\pi \simeq \mathrm{LLC}(\rho)$. This proposition has no claim that H5 produces a candidate, and no requirement that H5 hold for all possible realizers.

The retained general Lean interface has an *all-realizers* profile binder because its `Realization` relation may have several realizers. The publication proposition needs the factor equality only for its one supplied candidate. The pointwise separation lemma `class_eq_iff` gives that implication directly. The broader interface should not be misread as strengthening the publication conjecture.

The mathematical text fixes the realization definition, admitted source class and finite hypothesis data. General arithmetic realization is a conjecture. Factor comparison is conditional. The restricted odd-prime and dyadic constructions are ordinary mathematical results under their stated hypotheses. The formal companion adds neither an axiom for the conjecture nor a placeholder proof.

The restricted constructions are not Lean-certified local-field and simple-character developments. They retain the same-candidate H5 boundary. The dyadic low-rank enlargement changes the arithmetic assembly carrier while preserving the candidate through induction in stages. It does not enlarge the fixed maximal-type quotient obstructed by the dimension calculation.

APPENDIX A

Complete selected named-source catalog

Every selected module below is copied unchanged into the editable guide's `sources/PaperXI` folder. The catalog gives module identity, selected layer, SHA-256, direct project imports, module documentation and every named lexical declaration with its raw source line. These line pins locate declarations in the complete source, not independently compiled snippets. Earlier repeated source-layer identities remain in `SOURCE_VERSIONS.json`.

A.1 `PaperXI.AbstractHeisenberg`

Source lines: 184.

SHA-256: `d83045df228867234c49fb9dd9e838bbf810394614dd22dce2401c6744e3dacd`.

```
Actual Heisenberg groups on abstract bilinear spaces, their central extensions,
and center-preserving lifts of arbitrary form-preserving linear equivalences.
No choice of symplectic coordinates is used.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
18	structure	`PaperXI.AbstractHeisenberg.Element`
24	instance	`PaperXI.AbstractHeisenberg.instMul`
28	instance	`PaperXI.AbstractHeisenberg.instOne`
33	instance	`PaperXI.AbstractHeisenberg.instInv`
36	theorem	`PaperXI.AbstractHeisenberg.mul_vector`
37	theorem	`PaperXI.AbstractHeisenberg.mul_central`
39	theorem	`PaperXI.AbstractHeisenberg.one_vector`
40	theorem	`PaperXI.AbstractHeisenberg.one_central`
41	theorem	`PaperXI.AbstractHeisenberg.inv_vector`
42	theorem	`PaperXI.AbstractHeisenberg.inv_central`
45	instance	`PaperXI.AbstractHeisenberg.instGroup`
55	def	`PaperXI.AbstractHeisenberg.coordinates`
61	instance	`PaperXI.AbstractHeisenberg.instFinite`
65	theorem	`PaperXI.AbstractHeisenberg.finite_of_finite_field`
70	def	`PaperXI.AbstractHeisenberg.centralElement`
72	theorem	`PaperXI.AbstractHeisenberg.centralElement_vector`
73	theorem	`PaperXI.AbstractHeisenberg.centralElement_central`
74	theorem	`PaperXI.AbstractHeisenberg.centralElement_zero`
75	theorem	`PaperXI.AbstractHeisenberg.centralElement_mul`
79	theorem	`PaperXI.AbstractHeisenberg.centralElement_commute`
84	def	`PaperXI.AbstractHeisenberg.inclusion`
90	def	`PaperXI.AbstractHeisenberg.quotient`
95	theorem	`PaperXI.AbstractHeisenberg.inclusion_apply`
97	theorem	`PaperXI.AbstractHeisenberg.quotient_apply`
101	def	`PaperXI.AbstractHeisenberg.datum`
121	theorem	`PaperXI.AbstractHeisenberg.datum_i`

Line	Kind	Name
122	theorem	`PaperXI.AbstractHeisenberg.datum_q`
125	theorem	`PaperXI.AbstractHeisenberg.inv_central_of_isAlt`
129	theorem	`PaperXI.AbstractHeisenberg.mul_swap`
140	theorem	`PaperXI.AbstractHeisenberg.commutator`
151	def	`PaperXI.AbstractHeisenberg.lift`
163	theorem	`PaperXI.AbstractHeisenberg.lift_vector`
167	theorem	`PaperXI.AbstractHeisenberg.lift_central`
171	theorem	`PaperXI.AbstractHeisenberg.lift_inclusion`
177	theorem	`PaperXI.AbstractHeisenberg.lift_quotient`

A.2 `PaperXI.AbstractHeisenbergExtension`

Source lines: 58.

SHA-256: `64c993ffe68d2705763bdb7173dcdb8096ddb3540a82363620d5ac0e820f8d06`.

```
The actual commutator of the abstract odd-characteristic Heisenberg extension
is the supplied alternating form, independently of the chosen quotient section.
```

Direct project imports: `PaperXI.AbstractHeisenberg`, `PaperXI.CentralExtensionCocycle`, `PaperXI.SymplecticCentralExtension`.

Line	Kind	Name
15	theorem	`PaperXI.AbstractHeisenberg.centralCommutator_eq`
23	theorem	`PaperXI.AbstractHeisenberg.scalar_commutator`
33	theorem	`PaperXI.AbstractHeisenberg.scalar_radical_eq_bot`
43	theorem	`PaperXI.AbstractHeisenberg.scalar_radical_eq_bot_scalarCharacter`
52	theorem	`PaperXI.AbstractHeisenberg.center_eq_inclusion_range`

A.3 `PaperXI.ActualCartanCandidate106`

Source lines: 79.

SHA-256: `e87ad679e0338ba7b5ffda904079152531d758cd29a472151aa21431bc543d06`.

```
The same candidate constructor with Cartan decomposition discharged from
actual local-field structure. The BK tensor reduction remains explicit.
```

Direct project imports: `PaperXI.ActualLocalCartan106`, `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
31	theorem	`PaperXI.ActualCartanCandidate106.candidate_exists`

A.4 `PaperXI.ActualCharacterCandidate108`

Source lines: 89.

SHA-256: `a311e3796f48d7d7a6c2cafd95e7ff962b8fb7543a0cdb7a81d04d9d12dbe4f2`.

```
Character attachment removes a redundant beta-support input. Character
support and the Heisenberg Hom dimension are still arithmetic hypotheses.
```

Direct project imports: `PaperXI.ActualCharacterData108`.

Line	Kind	Name
31	theorem	`PaperXI.ActualCharacterCandidate108.candidate_exists`

A.5 `PaperXI.ActualCharacterData108`

Source lines: 49.

SHA-256: `adcb8f31671a6ce4587678730fd689cdda4c87ebd3f1c8c67391cf42ee921bc4`.

```
Character occurrence on the actual prescribed kappa carrier, with its
normality and invariance derived from the literal character normalizer.
```

Direct project imports: `PaperXI.ActualHeisenbergCandidate107`, `PaperXI.CharacterSupport108`.

Line	Kind	Name
22	theorem	`PaperXI.ActualCharacterData108.domain_le_compact`
26	def	`PaperXI.ActualCharacterData108.intoCompact`
30	theorem	`PaperXI.ActualCharacterData108.intoCompact_mem_first`
37	def	`PaperXI.ActualCharacterData108.Contains`
41	theorem	`PaperXI.ActualCharacterData108.scalar_restriction`

A.6 `PaperXI.ActualCharacterIdentification61`

Source lines: 100.

SHA-256: `8507c7ccc941069745d0352c942ee3b600e8fb68441dc2ecd4d458f31ce367fc`.

```
# Identification with the supplied genuine compact type group

The exact standard-theory input is the actual full character-normalizer
formula N(theta)=E^× J0, together with integral E-units in J0 and preservation
of the lattice chain. These imply that the constructed compact intersection
is literally J0. No equality J1=U_a1 is used or concluded.
```

Direct project imports: `PaperXI.ActualCharacterType61`.

Line	Kind	Name
27	theorem	`PaperXI.ActualCharacterIdentification61.compactGroup_eq_of_full_normalizer`
59	theorem	`PaperXI.ActualCharacterIdentification61.uniformizer_mem_full_normalizer`
69	theorem	`PaperXI.ActualCharacterIdentification61.extendedSubgroup_eq_full_normalizer`

A.7 `PaperXI.ActualCharacterResidue61`

Source lines: 133.

SHA-256: `401e10d5858bab78cc6a399bac1bd8c3d594a891773157e9ff975c2a7f0b805f`.

```
# Actual residue reduction on the theta-dependent compact group

The map is the restriction of the already constructed literal lattice residue
map. Its kernel is the actual intersection C(theta)∩U_a1. Containment in the
lattice-product group and normalization of theta by integral centralizer
matrices are explicit standard-type inputs. They do not assert C(theta)=B U_a1.
```

Direct project imports: `PaperXI.ActualCharacterType61`, `PaperXI.ActualELatticeFieldJoin59`.

Line	Kind	Name
29	def	`PaperXI.ActualCharacterResidue61.intoModel`
35	def	`PaperXI.ActualCharacterResidue61.reduction`
40	theorem	`PaperXI.ActualCharacterResidue61.reduction_kernel`
48	theorem	`PaperXI.ActualCharacterResidue61.integralMatrix_mem_compactGroup`
60	theorem	`PaperXI.ActualCharacterResidue61.integralUnit_mem_compactGroup`
76	theorem	`PaperXI.ActualCharacterResidue61.reduction_surjective`
110	theorem	`PaperXI.ActualCharacterResidue61.reduction_conjugation`

A.8 `PaperXI.ActualCharacterType61`

Source lines: 163.

SHA-256: `31cbff49441e1406a497ce67cbba0965509582286af1f2fc8337525909e25c1f`.

```
# The compact lattice part of an actual character normalizer

This group depends on the actual subgroup-character pair. It is not the larger
literal lattice-product group from Block59. Its first subgroup is its actual
intersection with adjacent-lattice congruence, never an equality with all U1.
```

Direct project imports: `PaperXI.CharacterNormalizer61`, `PaperXI.ActualELatticeTopology60`.

Line	Kind	Name
23	theorem	`PaperXI.ActualCharacterType61.power_normalizes_chainStabilizer`
53	def	`PaperXI.ActualCharacterType61.chainGroup`
58	theorem	`PaperXI.ActualCharacterType61.matrixCongruence_le_chainGroup`
62	theorem	`PaperXI.ActualCharacterType61.chainGroup_open`
68	theorem	`PaperXI.ActualCharacterType61.chainGroup_compact`
83	theorem	`PaperXI.ActualCharacterType61.coordinateAut_unitsEmbedding`
96	theorem	`PaperXI.ActualCharacterType61.coordinateAut_embeddedUniformizer`
110	theorem	`PaperXI.ActualCharacterType61.uniformizer_normalizes_chainGroup`
126	def	`PaperXI.ActualCharacterType61.compactGroup`
130	def	`PaperXI.ActualCharacterType61.firstSubgroup`
135	theorem	`PaperXI.ActualCharacterType61.domain_le_compactGroup`
140	theorem	`PaperXI.ActualCharacterType61.compactGroup_compact_open`
148	theorem	`PaperXI.ActualCharacterType61.firstSubgroup_open`
154	theorem	`PaperXI.ActualCharacterType61.uniformizer_normalizes_compactGroup`

A.9 `PaperXI.ActualCompactTypeCompletion58`

Source lines: 97.

SHA-256: `9c0e0c63cd7aa7f6c5ce5841cf9dbd3b97709ee0300314cb808dae8caef66b2a`.

```
# Same-factor actual compact type extension

The literal tensor action retains the prescribed Heisenberg factor. Its
irreducibility and smoothness, and the whole-type normalizer intertwiner, are
constructed. Actual embedded-field arithmetic supplies cyclic freeness.
Normalization of the actual compact subgroup, quotient-triviality, and one
nonzero first-factor intertwiner remain explicit type-theoretic inputs.
```

Direct project imports: `PaperXI.ActualNormalizerTypeExtension58`, `PaperXI.CompactTypeTensorNormalizer58`.

Line	Kind	Name
25	theorem	`PaperXI.ActualCompactTypeCompletion58.firstFactor_irreducible`
53	theorem	`PaperXI.ActualCompactTypeCompletion58.completed`

A.10 `PaperXI.ActualCoordinateDeepIdeal111`

Source lines: 80.

SHA-256: `1478145467bf2f189641de88b717440f1439dc649d84164ad972b32966afeb91`.

```
The actual integral coordinate-order instance and its one-plus index
application. Ideal containment and quotient finiteness are conclusions.
```

Direct project imports: `PaperXI.CoordinateDeepIdeal111`.

Line	Kind	Name
14	def	`PaperXI.ActualCoordinateDeepIdeal111.innerHomeomorph`
26	def	`PaperXI.ActualCoordinateDeepIdeal111.integralEmbedding`
29	theorem	`PaperXI.ActualCoordinateDeepIdeal111.integralEmbedding_open`
37	theorem	`PaperXI.ActualCoordinateDeepIdeal111.conjugate_lattice_open`
49	theorem	`PaperXI.ActualCoordinateDeepIdeal111.exists_conjugate_common_ideal`
66	theorem	`PaperXI.ActualCoordinateDeepIdeal111.onePlus_conjugate_relIndex`

A.11 `PaperXI.ActualCuspidalCandidate105`

Source lines: 118.

SHA-256: `b37fd23618fec1d17f60baca5392c3d0d4e8c255da68dfff131b5ca746840100`.

```
The actual character-dependent constructor of Block 70 now returns an
irreducible induced candidate, not a conditional promise requiring its full
intertwining locus. The residual cuspidality, arithmetic Cartan decomposition,
and BK tensor-to-residue reduction remain explicit.
```

Direct project imports: `PaperXI.FullTypeIntertwining105`, `PaperXI.ActualPacketTypeRealization70`.

Line	Kind	Name
26	def	`PaperXI.ActualCuspidalCandidate105.centralizerMap`
29	abbrev	`PaperXI.ActualCuspidalCandidate105.compact`
30	abbrev	`PaperXI.ActualCuspidalCandidate105.extended`
36	theorem	`PaperXI.ActualCuspidalCandidate105.centralizer_scalar`
54	theorem	`PaperXI.ActualCuspidalCandidate105.candidate_exists`

A.12 `PaperXI.ActualDepthZeroExtension`

Source lines: 155.

SHA-256: `ff5750ec70106f6aa0ce85b66fc16095d1d17973b914f7b37b2e2c1efdcda17c`.

```
# Actual depth-zero inducing data extended by the scalar uniformizer

The subgroup, degree, free central direction and smooth inducing representation
are constructed from the given local field. The supplied finite representation
is inflated through literal residue reduction. Induced irreducibility is a
separate theorem: no Mackey vanishing is assumed in this construction.
```

Direct project imports: `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualLocalUnitDecomposition`, `PaperXI.CentralExtensionInductionConsequences`.

Line	Kind	Name
25	def	`PaperXI.ActualDepthZeroExtension.determinantDegree`
29	def	`PaperXI.ActualDepthZeroExtension.scalarUniformizer`
32	theorem	`PaperXI.ActualDepthZeroExtension.scalarUniformizer_central`
35	theorem	`PaperXI.ActualDepthZeroExtension.determinantDegree_embedding`
41	theorem	`PaperXI.ActualDepthZeroExtension.integralSubgroup_le_degree_kernel`
46	theorem	`PaperXI.ActualDepthZeroExtension.determinantDegree_scalarUniformizer`
56	theorem	`PaperXI.ActualDepthZeroExtension.scalarUniformizer_free`
66	abbrev	`PaperXI.ActualDepthZeroExtension.extendedSubgroup`
71	theorem	`PaperXI.ActualDepthZeroExtension.integral_le_extended`
76	theorem	`PaperXI.ActualDepthZeroExtension.scalar_unit_mem_integral`
83	theorem	`PaperXI.ActualDepthZeroExtension.extendedSubgroup_eq_scalar_sup`
106	theorem	`PaperXI.ActualDepthZeroExtension.extendedSubgroup_open`
111	abbrev	`PaperXI.ActualDepthZeroExtension.inclusion`
118	abbrev	`PaperXI.ActualDepthZeroExtension.extendedRepresentation`
122	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_restrict`
126	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_irreducible`
131	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_smooth`
137	abbrev	`PaperXI.ActualDepthZeroExtension.inducedRepresentation`
142	theorem	`PaperXI.ActualDepthZeroExtension.induced_smooth`
146	theorem	`PaperXI.ActualDepthZeroExtension.induced_nontrivial`
151	theorem	`PaperXI.ActualDepthZeroExtension.induced_scalar`

A.13 `PaperXI.ActualELatticeFieldJoin59`

Source lines: 80.

SHA-256: `cd8801f07f19c6130f8e0cb4b83972ac28c399481be71c0f8cee79f7e9bbd660`.

```
# The literal field-times-lattice inducing subgroup

Integral extension-field units already lie in the actual integral matrix
factor. The unit/uniformizer decomposition of E× therefore identifies the
subgroup generated by J and pi_E with the subgroup generated by J and E×.
```

Direct project imports: `PaperXI.ActualELatticeQuotient59`, `PaperXI.ActualELatticeFreeness59`.

Line	Kind	Name
24	theorem	`PaperXI.ActualELatticeFieldJoin59.integralMatrix_scalar`
34	theorem	`PaperXI.ActualELatticeFieldJoin59.integralUnit_mem_typeGroup`
40	theorem	`PaperXI.ActualELatticeFieldJoin59.extendedSubgroup_eq_field_join`
60	theorem	`PaperXI.ActualELatticeFieldJoin59.field_normalizes_typeGroup`
71	theorem	`PaperXI.ActualELatticeFieldJoin59.extendedSubgroup_coe_eq_field_mul`

A.14 PaperXI.ActualELatticeFreeness59

Source lines: 73.

SHA-256: b9c837f72cdde593dc7a1bf64f366e0e5453e390448eedf225b0024251c84a47.

```
# Uniformizer freeness for the actual lattice-product group

Freeness is proved directly from the integral lattice and the strict valuation
of the actual extension-field uniformizer. It does not assume compactness of
the group or a matrix determinant identity.
```

Direct project imports: PaperXI.ActualELatticeTypeGroup59.

Line	Kind	Name
21	def	PaperXI.ActualELatticeFreeness59.scalarAction
28	theorem	PaperXI.ActualELatticeFreeness59.scalarAction_apply
33	theorem	PaperXI.ActualELatticeFreeness59.uniformizerAction_zpow_apply
41	theorem	PaperXI.ActualELatticeFreeness59.typeGroup_le_stabilizer
49	theorem	PaperXI.ActualELatticeFreeness59.uniformizerAction_exponent_nonneg
66	theorem	PaperXI.ActualELatticeFreeness59.uniformizerAction_free

A.15 PaperXI.ActualELatticeNormalizer59

Source lines: 129.

SHA-256: 7328521c5eb95fe067aaae8fbb771a976d75965543e6cb1619720f6df9d68c17.

```
# Periodic lattices and their literal first congruence group

The chain used below is the actual sequence of scalar-uniformizer translates
of the integral lattice in `E^m`, viewed as a vector space over `K`. Its first
congruence group consists of automorphisms preserving every lattice and acting
as the identity modulo the next lattice. Uniformizer normalization is proved
from the coordinate definitions. This is not an identification with the
simple-character group of an arbitrary stratum.
```

Direct project imports: PaperXI.ActualIntegralGeneralLinear, PaperXI.ActualLocalUnitDecomposition, PaperXI.NormalizerExtensionInducing58.

Line	Kind	Name
24	def	PaperXI.ActualELatticeNormalizer59.chainStabilizer
35	def	PaperXI.ActualELatticeNormalizer59.firstCongruence
59	theorem	PaperXI.ActualELatticeNormalizer59.firstCongruence_le
63	theorem	PaperXI.ActualELatticeNormalizer59.stabilizer_normalizes_firstCongruence
84	def	PaperXI.ActualELatticeNormalizer59.periodicChain
88	theorem	PaperXI.ActualELatticeNormalizer59.periodicChain_shift
96	theorem	PaperXI.ActualELatticeNormalizer59.power_normalizes_firstCongruence

A.16 PaperXI.ActualELatticeQuotient59

Source lines: 89.

SHA-256: a49f9630deecf93df96b4f831fb702cd48b0154c5563d3fe4f919ad2c9f47d24.

```
# Finite residue quotient of the literal extension-field lattice product

For the actual subgroup `GL_n(O_E) U^1` in `Aut_K(E^n)`, the quotient by
literal adjacent-lattice congruence is `GL_n(k_E)`. Conjugation by the
extension-field uniformizer is the identity on this quotient.
```

Direct project imports: `PaperXI.ActualELatticeTypeGroup59`, `PaperXI.ProductNormalizerQuotient59`.

Line	Kind	Name
22	def	`PaperXI.ActualELatticeQuotient59.typeCongruence`
25	instance	`PaperXI.ActualELatticeQuotient59.typeCongruence_normal`
29	def	`PaperXI.ActualELatticeQuotient59.residueEquiv`
37	def	`PaperXI.ActualELatticeQuotient59.reduction`
40	theorem	`PaperXI.ActualELatticeQuotient59.reduction_surjective`
43	theorem	`PaperXI.ActualELatticeQuotient59.reduction_kernel`
50	theorem	`PaperXI.ActualELatticeQuotient59.finite_quotient`
54	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_centralizes`
62	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_normalizes_congruence`
68	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_normalizes`
75	def	`PaperXI.ActualELatticeQuotient59.uniformizerConjugation`
81	theorem	`PaperXI.ActualELatticeQuotient59.reduction_conjugation`

A.17 `PaperXI.ActualELatticeResidue59`

Source lines: 157.

SHA-256: `7adc864addc36f19b3d94431dcbaa62f118e45633f9bc356692c1c8c0698d5a3`.

```
# Literal residue action on an extension-field lattice

The group and the congruence kernel below are the actual integral matrices on
`E^n`, viewed as `K`-linear operators. The next lattice is multiplication of the
integral lattice by the chosen actual uniformizer.
```

Direct project imports: `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualLocalUnitDecomposition`.

Line	Kind	Name
20	def	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix`
25	theorem	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix_apply`
28	theorem	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix_injective`
37	def	`PaperXI.ActualELatticeResidue59.integralMatrix`
40	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_injective`
44	def	`PaperXI.ActualELatticeResidue59.integralLattice`
51	theorem	`PaperXI.ActualELatticeResidue59.mem_integralLattice`
55	theorem	`PaperXI.ActualELatticeResidue59.integralLattice_coe`
58	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_coe`
64	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_preserves`
74	def	`PaperXI.ActualELatticeResidue59.nextLattice`
79	theorem	`PaperXI.ActualELatticeResidue59.mem_nextLattice`
84	theorem	`PaperXI.ActualELatticeResidue59.coe_mem_nextLattice_iff`
111	theorem	`PaperXI.ActualELatticeResidue59.reduction_kernel_iff`
139	def	`PaperXI.ActualELatticeResidue59.integralImage`
141	def	`PaperXI.ActualELatticeResidue59.integralImageEquiv`
144	def	`PaperXI.ActualELatticeResidue59.residueAction`

Line	Kind	Name
147	theorem	`PaperXI.ActualELatticeResidue59.residueAction_surjective`
152	def	`PaperXI.ActualELatticeResidue59.residueQuotientEquiv`

A.18 `PaperXI.ActualELatticeScalarPeriod60`

Source lines: 87.

SHA-256: `10bfffe3207aaacbc9291830df80f34c502d7383c79a8a9e1b7b723e7e3021f0`.

```
# A base-field scalar period for the literal extension-field lattice chain

The actual base-field uniformizer has nonzero extension-field degree. Otherwise
its image would be an integral extension-field unit, whose matrix determinant
degree vanishes, contradicting its known scalar-matrix degree. Choosing it or
its inverse supplies a positive period without assuming a ramification formula.
```

Direct project imports: `PaperXI.ActualELatticeFieldJoin59`, `PaperXI.ActualEmbeddedUniformizer58`.

Line	Kind	Name
26	theorem	`PaperXI.ActualELatticeScalarPeriod60.base_uniformizer_degree_ne_one`
42	theorem	`PaperXI.ActualELatticeScalarPeriod60.exists_positive_base_normalForm`
67	theorem	`PaperXI.ActualELatticeScalarPeriod60.chain_period_of_normalForm`
81	theorem	`PaperXI.ActualELatticeScalarPeriod60.exists_chain_period`

A.19 `PaperXI.ActualELatticeTopology60`

Source lines: 170.

SHA-256: `4936918f1210fcc205c45cc2b2e787c6dfa51e1d1399c1c9e3988e6d9e791981`.

```
# Compact open topology for the literal extension-field lattice type group

The exact Block59 lattice groups are realized in actual `GL_d(K)` through
finite continuous coordinates on `E^n`. Compactness, openness, and the open
kernel of actual residue reduction are proved. No identification with the
simple-character group of an arbitrary stratum is asserted.
```

Direct project imports: `PaperXI.LatticeCongruenceTopology60`, `PaperXI.ActualELatticeScalarPeriod60`.

Line	Kind	Name
29	theorem	`PaperXI.ActualELatticeTopology60.field_nhdsNE_neBot`
41	def	`PaperXI.ActualELatticeTopology60.coordinatesOfBasis`
56	def	`PaperXI.ActualELatticeTopology60.scalarMatrixEmbedding`
62	theorem	`PaperXI.ActualELatticeTopology60.integralLattice_compact`
67	theorem	`PaperXI.ActualELatticeTopology60.integralLattice_open`
74	theorem	`PaperXI.ActualELatticeTopology60.latticeChain_compact`
86	theorem	`PaperXI.ActualELatticeTopology60.latticeChain_open`
95	def	`PaperXI.ActualELatticeTopology60.matrixType`
99	def	`PaperXI.ActualELatticeTopology60.matrixCongruence`
105	theorem	`PaperXI.ActualELatticeTopology60.matrixCongruence_open`
115	theorem	`PaperXI.ActualELatticeTopology60.matrixType_open`
124	theorem	`PaperXI.ActualELatticeTopology60.matrixType_compact`

Line	Kind	Name
139	theorem	`PaperXI.ActualELatticeTopology60.matrixCongruence_compact`
148	theorem	`PaperXI.ActualELatticeTopology60.compact_open_of_basis`
155	def	`PaperXI.ActualELatticeTopology60.finiteExtensionCoordinates`
161	theorem	`PaperXI.ActualELatticeTopology60.finiteExtension_compact_open`

A.20 `PaperXI.ActualELatticeTypeGroup59`

Source lines: 136.

SHA-256: `985e4b5cffb7f0083ef712004a55d0dd95409019eac2f83d57c77cd971ef0455`.

```
# The actual lattice-product type group

The lattice and its first congruence subgroup are literal subgroups of
`Aut_K(E^n)`. The centralizer integral matrices normalize congruence, and their
intersection with congruence is exactly their residue kernel.
```

Direct project imports: `PaperXI.ActualELatticeNormalizer59`, `PaperXI.ActualELatticeResidue59`.

Line	Kind	Name
22	def	`PaperXI.ActualELatticeTypeGroup59.uniformizerAction`
27	theorem	`PaperXI.ActualELatticeTypeGroup59.uniformizerAction_apply`
31	def	`PaperXI.ActualELatticeTypeGroup59.latticeChain`
35	theorem	`PaperXI.ActualELatticeTypeGroup59.latticeChain_zero`
40	theorem	`PaperXI.ActualELatticeTypeGroup59.latticeChain_one`
56	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_preserves_iff`
65	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_commute`
74	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_mem_stabilizer`
83	def	`PaperXI.ActualELatticeTypeGroup59.congruence`
87	theorem	`PaperXI.ActualELatticeTypeGroup59.integralImage_normalizes`
94	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_mem_congruence_iff`
124	theorem	`PaperXI.ActualELatticeTypeGroup59.congruence_subgroupOf_integralImage`
133	def	`PaperXI.ActualELatticeTypeGroup59.typeGroup`

A.21 `PaperXI.ActualEmbeddedUniformizer58`

Source lines: 177.

SHA-256: `3f8d9f340d7a6f79785e6b82d82a800344788509d9b97b51c77aa96429018a33`.

```
# Uniformizers of actual matrix-embedded local fields

For a continuous algebra embedding of a local field `E` into matrices over
`F`, its actual uniformizer has nonzero determinant degree over `F`.
The proof uses actual compact integral units and actual uniformizer coordinates;
it does not assume a determinant-norm formula or a ramification index.
Consequently no nonzero power belongs to any compact matrix subgroup.
The embedding may be noncentral. No simple-type realization is asserted here.
```

Direct project imports: `PaperXI.ActualDepthZeroExtension`, `PaperXI.CompactDegreeSeparation`.

Line	Kind	Name
30	def	PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding
34	def	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer
39	theorem	PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding_continuous
42	theorem	PaperXI.ActualEmbeddedUniformizer58.determinantDegree_continuous
48	theorem	PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding_baseScalar
57	theorem	PaperXI.ActualEmbeddedUniformizer58.determinantDegree_integral_unit
71	theorem	PaperXI.ActualEmbeddedUniformizer58.algebraEmbedding_continuous
89	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_degree_ne_one
107	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_free
117	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_free_of_algebraMap
126	theorem	PaperXI.ActualEmbeddedUniformizer58.normalForm_injective
152	theorem	PaperXI.ActualEmbeddedUniformizer58.field_product_normalForm
166	theorem	PaperXI.ActualEmbeddedUniformizer58.field_product_unique_normalForm

A.22 PaperXI.ActualFieldTypeQuotient69

Source lines: 157.

SHA-256: fd2a01056fe19cf73010c917683c3bcdac14d241c1d9f3bc3accb3584b709bd6.

```
Compactness modulo the actual base-field scalar matrices for E-times-J.
```

Direct project imports: PaperXI.ActualNormalizerTypeExtension58, PaperXI.CompactQuotientPeriod69.

Line	Kind	Name
21	abbrev	PaperXI.ActualFieldTypeQuotient69.fieldGroup
25	def	PaperXI.ActualFieldTypeQuotient69.scalarHom
31	def	PaperXI.ActualFieldTypeQuotient69.scalarSubgroup
36	theorem	PaperXI.ActualFieldTypeQuotient69.scalar_central
43	instance	PaperXI.ActualFieldTypeQuotient69.scalar_normal
51	def	PaperXI.ActualFieldTypeQuotient69.compactPart
55	theorem	PaperXI.ActualFieldTypeQuotient69.degree_kernel
98	theorem	PaperXI.ActualFieldTypeQuotient69.quotient_compact

A.23 PaperXI.ActualGroupIndex111

Source lines: 273.

SHA-256: 569cdbec7bc0fcd987fb8fc614db2c4369221abc0007eb98bfdff0896e0800c2.

```
The exact ambient group index from two supplied open coordinate lattices
and two finite full Morita lattices. Canonical H1/J1 identifications remain
source matches; no index identity or common ideal is assumed.
```

Direct project imports: PaperXI.ActualCoordinateDeepIdeal111, PaperXI.ActualOnePlusTopology111, PaperXI.MoritaEndomorphismCoordinates111, PaperXI.MoritaPairIndex110, PaperXI.MoritaRadicalDictionary110, PaperXI.DegreeIndex110.

Line	Kind	Name
24	theorem	`PaperXI.ActualGroupIndex111.units_innerRing_inv`
33	theorem	`PaperXI.ActualGroupIndex111.map_inner_inv_eq_comap`
47	theorem	`PaperXI.ActualGroupIndex111.integralUnitMap_eq_actualEmbedding`
53	theorem	`PaperXI.ActualGroupIndex111.onePlus_backward_relIndex`
93	theorem	`PaperXI.ActualGroupIndex111.onePlus_index_of_Morita_image`
125	theorem	`PaperXI.ActualGroupIndex111.ambient_pair_index`
163	theorem	`PaperXI.ActualGroupIndex111.ambient_pair_index_of_radical_products`
202	structure	`PaperXI.ActualGroupIndex111.ActualLatticeData`
244	theorem	`PaperXI.ActualGroupIndex111.ActualLatticeData.actual_index`

A.24 `PaperXI.ActualHeisenbergCandidate107`

Source lines: 89.

SHA-256: `7dd358423f2d83b2201f0e8b9eed0484c2e577f041d171fe447c59806e11d971`.

```
The same arithmetic candidate, with the BK tensor-to-residue implication
proved internally from explicit Heisenberg support and intertwining-line inputs.
Those arithmetic Heisenberg inputs remain unproved here.
```

Direct project imports: `PaperXI.ActualCartanCandidate106`, `PaperXI.ResidueRestriction107`, `PaperXI.HeisenbergReduction107`, `PaperXI.IntertwiningSpace107`.

Line	Kind	Name
34	theorem	`PaperXI.ActualHeisenbergCandidate107.candidate_exists`

A.25 `PaperXI.ActualIntegerTower111`

Source lines: 178.

SHA-256: `6effc42145758945edec994d27d6de01bfae48a6752ba7b8f53ac9f8c9ee0300`.

```
Actual integer-ring compatibility from the original continuous local-field
embedding. Scalar towers are constructed; none is a supplied arithmetic input.
```

Direct project imports: `PaperXI.MoritaActualCartan110`.

Line	Kind	Name
20	theorem	`PaperXI.ActualIntegerTower111.map_integer_of_continuous`
43	def	`PaperXI.ActualIntegerTower111.integerMap`
47	theorem	`PaperXI.ActualIntegerTower111.integerMap_coe`
51	theorem	`PaperXI.ActualIntegerTower111.integer_of_map_integer`
79	theorem	`PaperXI.ActualIntegerTower111.integer_comap_eq`
88	theorem	`PaperXI.ActualIntegerTower111.valuationExtension`
93	def	`PaperXI.ActualIntegerTower111.integerAlgebra`
97	theorem	`PaperXI.ActualIntegerTower111.integerAlgebra_map_coe`
102	theorem	`PaperXI.ActualIntegerTower111.integer_tower`
109	theorem	`PaperXI.ActualIntegerTower111.integerAlgebra_map_eq`
117	theorem	`PaperXI.ActualIntegerTower111.map_valuation_le_iff`
126	theorem	`PaperXI.ActualIntegerTower111.integer_end_smul_apply`

Line	Kind	Name
131	theorem	PaperXI.ActualIntegerTower111.integer_end_tower
143	theorem	PaperXI.ActualIntegerTower111.integer_matrix_smul_apply
149	theorem	PaperXI.ActualIntegerTower111.integer_matrix_tower
162	theorem	PaperXI.ActualIntegerTower111.exists_actual_cartan_index_of_integer_lattice

A.26 PaperXI.ActualIntegralGeneralLinear

Source lines: 219.

SHA-256: 37fadf54dfd99b1032584b0bd3b561824077296be369ecc43f74deffb3847996.

```
# The actual compact open integral subgroup of `GL_n(K)`

The subgroup is the image of `GL_n([U+1D4AA][K])` in the given local field's matrix
group. Its compactness, openness and actual residue quotient are derived from
the entrywise maps. Inflation therefore gives smooth irreducible actions on
this actual subgroup from actual irreducible finite residue representations.
```

Direct project imports: PaperXI.ActualResidueGeneralLinear, PaperXI.QuotientRepresentation, PaperXI.SmoothDegreeQuotient, PaperXI.FiniteRepresentation.

Line	Kind	Name
27	def	PaperXI.ActualIntegralGeneralLinear.embedding
30	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_apply
33	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_injective
41	def	PaperXI.ActualIntegralGeneralLinear.integralSubgroup
44	def	PaperXI.ActualIntegralGeneralLinear.integralEquiv
47	theorem	PaperXI.ActualIntegralGeneralLinear.integralEquiv_coe
50	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_integralEquiv_symm
55	theorem	PaperXI.ActualIntegralGeneralLinear.mem_integralSubgroup
84	theorem	PaperXI.ActualIntegralGeneralLinear.field_t2Space
92	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_isEmbedding
96	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_continuous
100	theorem	PaperXI.ActualIntegralGeneralLinear.integralEquiv_symm_continuous
111	theorem	PaperXI.ActualIntegralGeneralLinear.integralSubgroup_compact
120	theorem	PaperXI.ActualIntegralGeneralLinear.integralSubgroup_open
131	def	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup
134	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_integralEquiv
138	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_surjective
142	theorem	PaperXI.ActualIntegralGeneralLinear.mem_reduction_kernel
156	theorem	PaperXI.ActualIntegralGeneralLinear.reduction_kernel_open
171	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_kernel_open
177	def	PaperXI.ActualIntegralGeneralLinear.residueQuotientEquiv
182	theorem	PaperXI.ActualIntegralGeneralLinear.finite_residue_quotient
191	def	PaperXI.ActualIntegralGeneralLinear.inflated
194	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_integralEquiv
199	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_irreducible
205	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_isSmooth
216	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_finiteDimensional

A.27 PaperXI.ActualIntersection108

Source lines: 59.

SHA-256: fede966a6ccc302109f79282ebd6c8a1b66b3c9dcbafc6b1cccd28b1160bd34b.

```
The canonical finite quotient for the actual first-subgroup intertwining
space. This proves existence of the finite model, not its arithmetic profile.
```

Direct project imports: PaperXI.ActualCharacterCandidate108, PaperXI.FiniteIntersectionCriterion108.

Line	Kind	Name
20	abbrev	PaperXI.ActualIntersection108.embedding
23	abbrev	PaperXI.ActualIntersection108.restriction
25	abbrev	PaperXI.ActualIntersection108.quotient
30	theorem	PaperXI.ActualIntersection108.finite
47	theorem	PaperXI.ActualIntersection108.dimension

A.28 PaperXI.ActualLocalCartan106

Source lines: 44.

SHA-256: ad8205ed61e0d305b7023ba61e8f17b266407a62ec64e846e7993a90d62882b2.

```
The arithmetic Cartan and residue support theorems on actual local-field
matrices, with a uniformizer chosen from the proved valuation-ring structure.
```

Direct project imports: PaperXI.DVRCartan106, PaperXI.ActualLocalUnitDecomposition, PaperXI.ActualIntegralGeneralLinear, PaperXI.ActualResidueGeneralLinear.

Line	Kind	Name
16	theorem	PaperXI.ActualLocalCartan106.uniformizer_residue_zero
22	theorem	PaperXI.ActualLocalCartan106.hasCartan
31	theorem	PaperXI.ActualLocalCartan106.intertwining_support

A.29 PaperXI.ActualLocalUnitDecomposition

Source lines: 128.

SHA-256: a8395974d47aefc05e157314ff5c0f41b5ec8a33c9bfb98579de8b62ab5e273d.

```
# Actual local-field unit and uniformizer coordinates

The uniformizer is chosen from the proved DVR structure of the actual ring of
integers. Existence of coordinates comes from fraction-field factorization;
uniqueness comes from the actual valuation. No split group is supplied.
```

Direct project imports: PaperXI.ActualLocalUnitTopology.

Line	Kind	Name
22	def	PaperXI.ActualLocalUnitDecomposition.uniformizer
24	theorem	PaperXI.ActualLocalUnitDecomposition.uniformizer_irreducible
27	theorem	PaperXI.ActualLocalUnitDecomposition.uniformizer_ne_zero

Line	Kind	Name
31	def	PaperXI.ActualLocalUnitDecomposition.uniformizerUnit
33	theorem	PaperXI.ActualLocalUnitDecomposition.uniformizerUnit_coe
35	theorem	PaperXI.ActualLocalUnitDecomposition.uniformizer_value_pos
38	theorem	PaperXI.ActualLocalUnitDecomposition.uniformizer_value_lt_one
42	def	PaperXI.ActualLocalUnitDecomposition.normalFormHom
50	theorem	PaperXI.ActualLocalUnitDecomposition.normalFormHom_apply
54	theorem	PaperXI.ActualLocalUnitDecomposition.normalForm_surjective
65	theorem	PaperXI.ActualLocalUnitDecomposition.normalForm_injective
82	def	PaperXI.ActualLocalUnitDecomposition.normalFormEquiv
85	theorem	PaperXI.ActualLocalUnitDecomposition.normalFormEquiv_apply
89	def	PaperXI.ActualLocalUnitDecomposition.degree
92	theorem	PaperXI.ActualLocalUnitDecomposition.degree_normalForm
97	theorem	PaperXI.ActualLocalUnitDecomposition.degree_unit
100	theorem	PaperXI.ActualLocalUnitDecomposition.degree_unitEmbedding
103	theorem	PaperXI.ActualLocalUnitDecomposition.degree_uniformizer
107	theorem	PaperXI.ActualLocalUnitDecomposition.ker_degree
119	theorem	PaperXI.ActualLocalUnitDecomposition.degree_continuous

A.30 PaperXI.ActualLocalUnitTopology

Source lines: 102.

SHA-256: `2d87df54c86bef03b5bf3277d309f08733cde2c39d287c2226b9d04bcbe388ea`.

```
# The actual compact open unit subgroup of a nonarchimedean local field

The subgroup is the image of the unit group of the valuation ring under its
canonical embedding in the field. Its topological properties are derived from
the local-field hypotheses; they are not assumptions on an abstract subgroup.
No characteristic-zero hypothesis is used.
```

Direct project imports: none.

Line	Kind	Name
23	def	PaperXI.ActualLocalUnitTopology.unitEmbedding
27	def	PaperXI.ActualLocalUnitTopology.compactUnitSubgroup
29	theorem	PaperXI.ActualLocalUnitTopology.coe_unitEmbedding
32	theorem	PaperXI.ActualLocalUnitTopology.unitEmbedding_injective
35	theorem	PaperXI.ActualLocalUnitTopology.mem_compactUnitSubgroup_iff
50	theorem	PaperXI.ActualLocalUnitTopology.compactUnitSubgroup_eq_units
62	theorem	PaperXI.ActualLocalUnitTopology.continuous_unitEmbedding
67	instance	PaperXI.ActualLocalUnitTopology.localFieldT2Space
73	theorem	PaperXI.ActualLocalUnitTopology.isCompact_compactUnitSubgroup
78	theorem	PaperXI.ActualLocalUnitTopology.isOpen_compactUnitSubgroup
84	def	PaperXI.ActualLocalUnitTopology.unitEmbeddingEquiv
92	theorem	PaperXI.ActualLocalUnitTopology.coe_unitEmbeddingEquiv
96	theorem	PaperXI.ActualLocalUnitTopology.unitEmbedding_isOpenMap

A.31 PaperXI.ActualNormalizerTypeExtension58

Source lines: 82.

SHA-256: `eed55102f4bdce2532e92a4655f4da687a9fa494d89917d61f2a8cef613b46f1`.

```
# Actual embedded-field normalizer extension

The noncentral matrix uniformizer's freeness is proved from compactness and
actual local-field arithmetic. Given its actual normalization of the compact
subgroup and one nonzero intertwiner, the literal representation is extended.
Simple-character normalization and beta-extension intertwining are explicit
remaining arithmetic inputs; no claim that they have been constructed is made.
```

Direct project imports: `PaperXI.ActualEmbeddedUniformizer58`, `PaperXI.NormalizerExtensionInducing58`.

Line	Kind	Name
32	theorem	`PaperXI.ActualNormalizerTypeExtension58.extendedSubgroup_eq_field_join`
60	theorem	`PaperXI.ActualNormalizerTypeExtension58.exists_smooth_irreducible_extension`

A.32 `PaperXI.ActualOnePlusTopology111`

Source lines: 114.

SHA-256: `cdef6d0ce5c93ad213cc8af1fef105ca1375f9566fdf088cdaad7677198cb9ac`.

```
Openness of an actual one-plus group supplies additive lattice openness.
```

Direct project imports: `PaperXI.ActualCoordinateDeepIdeal111`, `PaperXI.ActualIntegralGeneralLinear`.

Line	Kind	Name
19	def	`PaperXI.ActualOnePlusTopology111.matrixUnitsHomeomorph`
38	theorem	`PaperXI.ActualOnePlusTopology111.matrix_isOpenUnits`
53	theorem	`PaperXI.ActualOnePlusTopology111.actual_matrix_isOpenUnits`
59	theorem	`PaperXI.ActualOnePlusTopology111.additive_open_of_image_group_open`
100	theorem	`PaperXI.ActualOnePlusTopology111.onePlus_conjugate_relIndex_of_group_open`

A.33 `PaperXI.ActualPacketTypeRealization70`

Source lines: 109.

SHA-256: `079fe5114db1fe95bd51e370b7932a6394bce0b44bad32671fa820774e806037`.

```
Same-theta, same-kappa construction at the actual residue map and field join.
This replaces the old compact-subgroup Mackey premise by a full-locus condition
on the newly constructed extension. Standard character/type leaves remain
explicit; this is not a formalization of their classification theorem.
```

Direct project imports: `PaperXI.PacketTypeAttachment70`, `PaperXI.PrescribedFieldType70`, `PaperXI.ArithmeticTypeCertificate69`, `PaperXI.ActualCharacterIdentification61`, `PaperXI.ActualCharacterResidue61`.

Line	Kind	Name
36	theorem	`PaperXI.ActualPacketTypeRealization70.candidate_exists`

A.34 PaperXI.ActualProfileCandidate110

Source lines: 94.

SHA-256: dd434e182eec32397e64a3e6a43193cc56a3c965eff65c4e36e14f4bd0b2c27b.

```
The existing actual candidate from original eta trace support, centralizer
character covariance and the oriented ambient index identity. The degree,
finite quotient and Heisenberg Hom dimension are derived, not supplied.
Actual simple-character and category instantiation remains separate.
```

Direct project imports: PaperXI.DegreeIndex110.

Line	Kind	Name
34	theorem	PaperXI.ActualProfileCandidate110.candidate_exists_from_original_profiles

A.35 PaperXI.ActualResidueGeneralLinear

Source lines: 69.

SHA-256: 33f125bdc926fb2a521cd18f1d46ba7c0d70d277e4d96bf0827d0fc610cd3459.

```
Entrywise reduction of actual integral general linear groups.

The lifting argument works over any commutative local ring: lift each entry,
then use the nonzero reduced determinant to prove that the lifted determinant
is a unit. The final declarations specialize this to the given nonarchimedean
local field's own ring of integers and finite residue field.
```

Direct project imports: none.

Line	Kind	Name
21	def	PaperXI.ActualResidueGeneralLinear.residueMap
25	theorem	PaperXI.ActualResidueGeneralLinear.residueMap_apply
30	theorem	PaperXI.ActualResidueGeneralLinear.residueMap_surjective
52	def	PaperXI.ActualResidueGeneralLinear.reduction
55	theorem	PaperXI.ActualResidueGeneralLinear.reduction_apply
59	theorem	PaperXI.ActualResidueGeneralLinear.reduction_surjective
65	theorem	PaperXI.ActualResidueGeneralLinear.finite_reduction_target

A.36 PaperXI.ActualScalarLift113

Source lines: 221.

SHA-256: 9dca2abac60d683f19c4b2bd2bd257b8f18e5b5f459cf39df9948de4ccf41ce0.

```
Honest actions on an actual group induce honest scalar-pushout actions.
The section phase and its multiplication law are conclusions. The original
theta, its literal kernel subgroup, and the quotient labels are retained.
```

Direct project imports: PaperXI.ScalarRepresentation112, PaperXI.CyclicDefect113.

Line	Kind	Name
19	def	PaperXI.ActualScalarLift113.actualPushout

Line	Kind	Name
30	theorem	`PaperXI.ActualScalarLift113.actualPushout_section`
37	theorem	`PaperXI.ActualScalarLift113.actualPushout_subgroup`
50	theorem	`PaperXI.ActualScalarLift113.quotient_action_power`
62	theorem	`PaperXI.ActualScalarLift113.quotient_power_eq_one`
72	def	`PaperXI.ActualScalarLift113.subgroupMap`
88	def	`PaperXI.ActualScalarLift113.actualPhase`
92	theorem	`PaperXI.ActualScalarLift113.actualPhase_one`
98	theorem	`PaperXI.ActualScalarLift113.sectionScalar_action`
127	theorem	`PaperXI.ActualScalarLift113.actualPhase_mul`
137	def	`PaperXI.ActualScalarLift113.actualLift`
161	theorem	`PaperXI.ActualScalarLift113.actualLift_phase`
167	theorem	`PaperXI.ActualScalarLift113.actualLift_intertwines`
175	theorem	`PaperXI.ActualScalarLift113.actualLift_power_intertwines`
191	theorem	`PaperXI.ActualScalarLift113.actualLift_power_eq_one`

A.37 `PaperXI.ActualSourceCandidate111`

Source lines: 102.

SHA-256: `c802c3b3d1be72ca3d961a4edaa42cca066eb81cbc902a0a0abe9eee58d60005`.

```
The original candidate from exact type quotient/form data and literal
one-plus/Morita lattice matches. Trace support, covariance, finite quotient,
degree-square and the ambient index equality are derived. The input record
still requires its canonical type-theoretic instance; this is not Claim16.2.
```

Direct project imports: `PaperXI.ActualProfileCandidate110`, `PaperXI.HeisenbergCharacterProfile111`, `PaperXI.ActualGroupIndex111`.

Line	Kind	Name
40	theorem	`PaperXI.ActualSourceCandidate111.candidate_exists_from_source_dictionary`

A.38 `PaperXI.ActualUnramifiedBaseChange`

Source lines: 115.

SHA-256: `8f4e3c5117f404df075772a6e9b3782e2463b7e84973d380f248d9c9c5c4f87f`.

```
# Actual unramified base change through the recursive local-field tower

Block 40 proved Frobenius-power and Euler-factor formulas for an internal degree
dilation of the integer-degree unramified Weil model. Block 41 constructs an
actual recursive local-field tower and its restriction map. This module proves
that the two constructions give the same Frobenius polynomials and local
factors, and records conductor zero on both sides of the actual tower.

This remains scoped to the bounded unramified Weil models, not absolute Weil
groups or automorphic base change.
```

Direct project imports: `PaperXI.UnramifiedFieldTower`.

Line	Kind	Name
43	theorem	`PaperXI.ActualUnramifiedBaseChange.frobeniusPolynomial_eq_modelBaseChange`

Line	Kind	Name
52	theorem	PaperXI.ActualUnramifiedBaseChange.localFactor_eq_modelBaseChange
62	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_inertiaCodimension_eq_zero
71	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_swanConductorExponent_eq_zero
79	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_artinConductorExponent_eq_zero
87	theorem	PaperXI.ActualUnramifiedBaseChange.artinConductorExponent_restriction
98	theorem	PaperXI.ActualUnramifiedBaseChange.irreducible_restriction_frobeniusPolynomial
107	theorem	PaperXI.ActualUnramifiedBaseChange.irreducible_restriction_localFactor

A.39 PaperXI.AlgebraActionModule

Source lines: 250.

SHA-256: 5ceb841756c4ca3a00e4fe21b5969470175a83df02d2c417316405eced836820.

```
An algebra action gives an actual module on a separate copy of its vector space.
Submodules and intertwiners are identified with the corresponding complex-linear data.
```

Direct project imports: none.

Line	Kind	Name
19	def	PaperXI.AlgebraActionModule.Space
25	def	PaperXI.AlgebraActionModule.equiv
27	instance	PaperXI.AlgebraActionModule.instModule
29	theorem	PaperXI.AlgebraActionModule.equiv_smul
32	theorem	PaperXI.AlgebraActionModule.equiv_symm_apply
34	instance	PaperXI.AlgebraActionModule.instScalarTower
41	def	PaperXI.AlgebraActionModule.Stable
45	def	PaperXI.AlgebraActionModule.toSubspace
47	theorem	PaperXI.AlgebraActionModule.mem_toSubspace
50	theorem	PaperXI.AlgebraActionModule.stable_toSubspace
55	def	PaperXI.AlgebraActionModule.ofSubspace
61	theorem	PaperXI.AlgebraActionModule.mem_ofSubspace
64	theorem	PaperXI.AlgebraActionModule.toSubspace_ofSubspace
67	theorem	PaperXI.AlgebraActionModule.ofSubspace_toSubspace
71	def	PaperXI.AlgebraActionModule.submoduleOrderIso
78	theorem	PaperXI.AlgebraActionModule.toSubspace_bot
79	theorem	PaperXI.AlgebraActionModule.toSubspace_top
82	theorem	PaperXI.AlgebraActionModule.simple_iff
119	def	PaperXI.AlgebraActionModule.Intertwines
123	def	PaperXI.AlgebraActionModule.toModuleMap
130	def	PaperXI.AlgebraActionModule.toComplexMap
133	theorem	PaperXI.AlgebraActionModule.toModuleMap_apply
136	theorem	PaperXI.AlgebraActionModule.toComplexMap_apply
139	theorem	PaperXI.AlgebraActionModule.intertwines_toComplexMap
143	def	PaperXI.AlgebraActionModule.intertwinerEquiv
150	theorem	PaperXI.AlgebraActionModule.toComplexMap_toModuleMap
153	theorem	PaperXI.AlgebraActionModule.toModuleMap_toComplexMap
159	theorem	PaperXI.AlgebraActionModule.toComplexMap_id
165	theorem	PaperXI.AlgebraActionModule.toComplexMap_comp

Line	Kind	Name
173	def	PaperXI.AlgebraActionModule.toModuleEquiv
183	def	PaperXI.AlgebraActionModule.toComplexEquiv
186	theorem	PaperXI.AlgebraActionModule.toModuleEquiv_apply
190	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_apply
193	theorem	PaperXI.AlgebraActionModule.intertwines_toComplexEquiv
197	def	PaperXI.AlgebraActionModule.equivalenceEquiv
204	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_toModuleEquiv
208	theorem	PaperXI.AlgebraActionModule.toModuleEquiv_toComplexEquiv
213	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_refl
218	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_trans
232	def	PaperXI.AlgebraActionModule.ofModuleEquiv
240	theorem	PaperXI.AlgebraActionModule.ofModuleEquiv_apply
244	theorem	PaperXI.AlgebraActionModule.ofModule_simple_iff

A.40 PaperXI.ArithmeticAttachment110

Source lines: 161.

SHA-256: e67a6b640cb79d7ea347d00621361e950ea439519e53e1f472ef6758460a8693.

```
A genuine residue-rank-one specialization. The finite residue representation
is constructed, and the full extended support follows directly from character
support because GL_1(E) consists of field scalars. No Heisenberg intersection
dimension, BK lattice index, or supplied finite cuspidal representation occurs
as an input. The actual simple-character/model/category leaves remain explicit.
```

Direct project imports: PaperXI.ActualCharacterCandidate108.

Line	Kind	Name
18	def	PaperXI.ArithmeticAttachment110.residueOne
21	theorem	PaperXI.ArithmeticAttachment110.residueOne_irreducible
27	theorem	PaperXI.ArithmeticAttachment110.residueOne_cuspidal
37	theorem	PaperXI.ArithmeticAttachment110.gl_one_scalar
45	def	PaperXI.ArithmeticAttachment110.tensorUnitEquiv
66	theorem	PaperXI.ArithmeticAttachment110.integral_normalization_of_full_rank_one
88	theorem	PaperXI.ArithmeticAttachment110.rank_one_candidate_exists

A.41 PaperXI.ArithmeticFieldInduction69

Source lines: 83.

SHA-256: 84d9fdbbdab35cebcb07e0b60f4363cd4e176ac7b0483a850db3af4ebe56588e.

```
The compact-mod-center induction theorem specialized to literal local-field matrices.
The actual field scalar subgroup and compact quotient are constructed. The common
scalar character is derived from finite-dimensional irreducibility, not supplied.
```

Direct project imports: PaperXI.ActualFieldTypeQuotient69, PaperXI.DerivedCentralCharacter69, PaperXI.CompactModCenterInduction68.

Line	Kind	Name
27	theorem	`PaperXI.ArithmeticFieldInduction69.fieldGroup_open`
32	def	`PaperXI.ArithmeticFieldInduction69.scalarCharacter`
38	theorem	`PaperXI.ArithmeticFieldInduction69.scalarCharacter_action`
46	theorem	`PaperXI.ArithmeticFieldInduction69.induction_package`
72	theorem	`PaperXI.ArithmeticFieldInduction69.induced_scalar`

A.42 `PaperXI.ArithmeticTypeCertificate69`

Source lines: 106.

SHA-256: `a360d478cc7c93440aa4806bf37892a9cbc4c847d393a8be3079c7fc9da439ac`.

```
The actual base-field central character and the full intertwining-locus
application. No auxiliary scalar action or quotient compactness is assumed.
```

Direct project imports: `PaperXI.ArithmeticFieldInduction69`, `PaperXI.FullIntertwiningLocus69`.

Line	Kind	Name
24	def	`PaperXI.ArithmeticTypeCertificate69.centralCharacter`
29	theorem	`PaperXI.ArithmeticTypeCertificate69.centralCharacter_action`
35	theorem	`PaperXI.ArithmeticTypeCertificate69.induced_centralCharacter`
41	theorem	`PaperXI.ArithmeticTypeCertificate69.scalarHom_continuous`
56	theorem	`PaperXI.ArithmeticTypeCertificate69.centralCharacter_locallyConstant`
87	theorem	`PaperXI.ArithmeticTypeCertificate69.from_full_intertwining_locus`

A.43 `PaperXI.AtomicMultiset`

Source lines: 122.

SHA-256: `e9b0e86f94713a76638c5bde3c53ba97140aec5c543aad941c9bba61ede4d3f6`.

```
The block-multiset part of Paper XI, Theorem 2.5.
An atom is an isomorphism class. A block retains its strictly positive monodromy
length. Mapping atoms does not discard lengths, multiplicities, or repetitions.
This module contains no local-Weil or Langlands-classification existence axiom.
```

Direct project imports: none.

Line	Kind	Name
14	structure	`PaperXI.AtomicMultiset.Block`
21	def	`PaperXI.AtomicMultiset.blockMap`
24	theorem	`PaperXI.AtomicMultiset.blockMap_atom`
27	theorem	`PaperXI.AtomicMultiset.blockMap_length`
30	theorem	`PaperXI.AtomicMultiset.blockMap_id`
33	theorem	`PaperXI.AtomicMultiset.blockMap_comp`
36	def	`PaperXI.AtomicMultiset.extend`
41	theorem	`PaperXI.AtomicMultiset.extend_singleton`
44	theorem	`PaperXI.AtomicMultiset.extend_cons`
47	theorem	`PaperXI.AtomicMultiset.extend_card`
50	theorem	`PaperXI.AtomicMultiset.extend_id`

Line	Kind	Name
54	theorem	`PaperXI.AtomicMultiset.extend_comp`
59	theorem	`PaperXI.AtomicMultiset.extend_unique`
70	theorem	`PaperXI.AtomicMultiset.extend_injective`
80	def	`PaperXI.AtomicMultiset.extendEquiv`
94	def	`PaperXI.AtomicMultiset.dimension`
97	theorem	`PaperXI.AtomicMultiset.dimension_extend`
107	def	`PaperXI.AtomicMultiset.descend`
111	theorem	`PaperXI.AtomicMultiset.descend_mk`
116	theorem	`PaperXI.AtomicMultiset.extend_natural`

A.44 `PaperXI.AtomicNaturality`

Source lines: 106.

SHA-256: `ba747e172fb8b6398cc4ce4076c4f8dc2c3f5fcff98a6a24f1e3f1b4a4584873`.

```
Naturality and the exact information-preservation boundary for Theorem 2.5.
Twist laws are attached to the primitive decomposition and ordered quotient
relations. Equivariance of the final candidate is then a theorem, not an input.
```

Direct project imports: `PaperXI.AtomicReconstruction`.

Line	Kind	Name
15	theorem	`PaperXI.AtomicNaturality.reconstruction_from_representatives`
22	theorem	`PaperXI.AtomicNaturality.blocks_twist`
36	theorem	`PaperXI.AtomicNaturality.realize_twist`
53	theorem	`PaperXI.AtomicNaturality.candidate_twist`
67	theorem	`PaperXI.AtomicNaturality.extend_injective_iff`
84	theorem	`PaperXI.AtomicNaturality.candidate_injective`
92	theorem	`PaperXI.AtomicNaturality.decomposition_independence`

A.45 `PaperXI.AtomicReconstruction`

Source lines: 153.

SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
Paper XI, Theorem 2.5: choice-independent conditional reconstruction.
External interfaces here are exactly (1) existence and intrinsic uniqueness of
indecomposable source displays and (2) existence and uniqueness of irreducible
Langlands quotients of ordered target factors. They do not contain the final
candidate map. Sorting, atomic replacement, descent, and that map are constructed.
The theorem is conditional on these classical interfaces; it does not formalize
the construction/classification of local Weil groups or p-adic representations.
```

Direct project imports: `PaperXI.AtomicMultiset`.

Line	Kind	Name
17	structure	`PaperXI.AtomicReconstruction.Decomposition`
25	def	`PaperXI.AtomicReconstruction.rawBlocks`
28	theorem	`PaperXI.AtomicReconstruction.rawBlocks_eq`
33	theorem	`PaperXI.AtomicReconstruction.rawBlocks_iso`

Line	Kind	Name
41	def	PaperXI.AtomicReconstruction.blocks
44	theorem	PaperXI.AtomicReconstruction.blocks_of_display
52	structure	PaperXI.AtomicReconstruction.OrderedQuotient
63	def	PaperXI.AtomicReconstruction.ordered
67	theorem	PaperXI.AtomicReconstruction.ordered_pairwise
74	theorem	PaperXI.AtomicReconstruction.ordered_multiset
83	def	PaperXI.AtomicReconstruction.realize
86	theorem	PaperXI.AtomicReconstruction.realize_eq
95	theorem	PaperXI.AtomicReconstruction.realize_spec
103	def	PaperXI.AtomicReconstruction.candidate
108	def	PaperXI.AtomicReconstruction.Reconstructs
114	theorem	PaperXI.AtomicReconstruction.candidate_spec
121	theorem	PaperXI.AtomicReconstruction.candidate_eq
130	theorem	PaperXI.AtomicReconstruction.candidate_unique
143	theorem	PaperXI.AtomicReconstruction.formal_atomic_reconstruction
148	theorem	PaperXI.AtomicReconstruction.candidate_isomorphism

A.46 PaperXI.BilinearHeisenberg

Source lines: 152.

SHA-256: `d4615e543af67b750697c7fdbf5f7d3b790aa0b4d6992aa2246a3aa00399e7d9`.

```
Actual finite central extensions from bilinear factor sets.  The extension,
commutator, central character and Stone--von Neumann endpoint are distinct data.
No multiplication law, irrep, dimension or initial intertwiner is assumed.
```

Direct project imports: `PaperXI.SymplecticHeisenbergUniqueness`.

Line	Kind	Name
13	structure	PaperXI.BilinearHeisenberg.Element
19	instance	PaperXI.BilinearHeisenberg.instMul
21	instance	PaperXI.BilinearHeisenberg.instOne
22	instance	PaperXI.BilinearHeisenberg.instInv
25	theorem	PaperXI.BilinearHeisenberg.mul_vector
27	theorem	PaperXI.BilinearHeisenberg.mul_central
29	theorem	PaperXI.BilinearHeisenberg.one_vector
30	theorem	PaperXI.BilinearHeisenberg.one_central
31	theorem	PaperXI.BilinearHeisenberg.inv_vector
32	theorem	PaperXI.BilinearHeisenberg.inv_central
35	instance	PaperXI.BilinearHeisenberg.instGroup
44	def	PaperXI.BilinearHeisenberg.i
49	def	PaperXI.BilinearHeisenberg.q
54	theorem	PaperXI.BilinearHeisenberg.i_vector
55	theorem	PaperXI.BilinearHeisenberg.i_central
56	theorem	PaperXI.BilinearHeisenberg.q_apply
59	def	PaperXI.BilinearHeisenberg.datum
78	theorem	PaperXI.BilinearHeisenberg.commutator
88	theorem	PaperXI.BilinearHeisenberg.centralCommutator
96	instance	PaperXI.BilinearHeisenberg.instFinite

Line	Kind	Name
100	theorem	`PaperXI.BilinearHeisenberg.order`
109	theorem	`PaperXI.BilinearHeisenberg.source_commutator`
121	theorem	`PaperXI.BilinearHeisenberg.exists_irreducible`
136	theorem	`PaperXI.BilinearHeisenberg.exists_equiv`
145	theorem	`PaperXI.BilinearHeisenberg.finrank_sqrt`

A.47 `PaperXI.BoundaryExamples`

Source lines: 30.

SHA-256: `d847590d5166021de5e552840a8b6bac45da991bbac45860b3cd96c3aaeabf89`.

Direct project imports: `PaperXI.GeneralChain`, `PaperXI.ChainSharpness`, `PaperXI.ScalarCovariance`, `PaperXI.FourierCovariance`, `PaperXI.HolonomyResidual`.

Line	Kind	Name
11	theorem	`PaperXI.BoundaryExamples.empty_chain`
14	theorem	`PaperXI.BoundaryExamples.singleton_chain`
17	theorem	`PaperXI.BoundaryExamples.length_three_square_nonzero`
21	theorem	`PaperXI.BoundaryExamples.length_three_cube_zero`
24	theorem	`PaperXI.BoundaryExamples.inverse_two_mod_five`
26	theorem	`PaperXI.BoundaryExamples.inverse_label_differs`

A.48 `PaperXI.BreakBridge`

Source lines: 104.

SHA-256: `8867bca4f28ab57140255d385f80f0d940f50f9d7bfd3de435c9b4b6385937d9`.

```
Finite break tensor bridges on the actual central quotient. The total
stabilizer phase is the product of the actual factor discrepancies.
```

Direct project imports: `PaperXI.BreakTensorMap`, `PaperXI.HeisenbergTensor`, `PaperXI.CharacterGenerators`.

Line	Kind	Name
18	theorem	`PaperXI.BreakBridge.ambient_intertwines`
30	theorem	`PaperXI.BreakBridge.quotient_intertwines`
47	theorem	`PaperXI.BreakBridge.stabilizer_intertwines`
61	theorem	`PaperXI.BreakBridge.stabilizer_intertwines_of_factor_phases`
75	theorem	`PaperXI.BreakBridge.semidirect_intertwines`

A.49 `PaperXI.BreakBridgeAlong`

Source lines: 80.

SHA-256: `cc313014e54d45de8e7391509928e831d857cc92c21b22d029f4c5541b823658`.

```
Tensor bridges between different actual break extensions, with discrepancy
characters derived from the chosen honest stabilizer actions.
```

Direct project imports: `PaperXI.BreakBridge`, `PaperXI.CentralProductMap`, `PaperXI.HeisenbergTensorFactorization`.

Line	Kind	Name
19	theorem	`PaperXI.BreakBridgeAlong.quotient_intertwines`
45	theorem	`PaperXI.BreakBridgeAlong.source`

A.50 `PaperXI.BreakDyadicPhase`

Source lines: 115.

SHA-256: `6b938157b4b523f51398fb5d57351690ba2200647a98c90f9d715a12030349df`.

```
Paper XI 15.3, characteristic two: finite-family products of actual
normalized Gauss sums, and transfer through the supplied quadratic decomposition.
```

Direct project imports: `PaperXI.BreakQuadratic`.

Line	Kind	Name
20	theorem	`PaperXI.BreakDyadicPhase.gaussSum_product`
31	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_product`
42	theorem	`PaperXI.BreakDyadicPhase.arf_sum`
53	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_empty`
66	theorem	`PaperXI.BreakDyadicPhase.transport_refinement`
77	theorem	`PaperXI.BreakDyadicPhase.arf_of_decomposition`
95	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_of_decomposition`
110	theorem	`PaperXI.BreakDyadicPhase.half_finrank_of_decomposition`

A.51 `PaperXI.BreakHeisenbergSum`

Source lines: 143.

SHA-256: `8f7d2102714c520e91a760b6fe970dd48fb99ed979d8d4cd14280b6ffc18e50c`.

```
Actual half-form Heisenberg groups of a finite orthogonal sum are the
balanced central product of the summand groups. The retained center handles
an empty family. The group isomorphism is constructed, not assumed.
```

Direct project imports: `PaperXI.AbstractHeisenberg`, `PaperXI.CentralProduct`.

Line	Kind	Name
19	def	`PaperXI.BreakHeisenbergSum.form`
27	theorem	`PaperXI.BreakHeisenbergSum.form_apply`
30	theorem	`PaperXI.BreakHeisenbergSum.form_isAlt`
36	def	`PaperXI.BreakHeisenbergSum.fromProduct`
49	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_vector`
53	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_central`
58	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_surjective`
65	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_balanced`
72	def	`PaperXI.BreakHeisenbergSum.sumMap`
76	theorem	`PaperXI.BreakHeisenbergSum.sumMap_mk`

Line	Kind	Name
80	theorem	`PaperXI.BreakHeisenbergSum.sumMap_center`
84	theorem	`PaperXI.BreakHeisenbergSum.sumMap_surjective`
91	theorem	`PaperXI.BreakHeisenbergSum.sumMap_injective`
116	def	`PaperXI.BreakHeisenbergSum.sumEquiv`
120	theorem	`PaperXI.BreakHeisenbergSum.sumEquiv_mk`
124	theorem	`PaperXI.BreakHeisenbergSum.sumEquiv_center`
130	def	`PaperXI.BreakHeisenbergSum.decompositionEquiv`
136	theorem	`PaperXI.BreakHeisenbergSum.decompositionEquiv_center`

A.52 `PaperXI.BreakMoving`

Source lines: 116.

SHA-256: `e72bb01b58c7e900a66b0919ad0b78a71e7ae979c3ed55025b524b2d80519da9`.

```
Actual finite-family moving spaces. The maps below use the image of the
actual difference, not an independently supplied vector space of the same size.
```

Direct project imports: `PaperXI.MovingPhase`, `PaperXI.OrthogonalPhase`.

Line	Kind	Name
17	def	`PaperXI.BreakMoving.productEquiv`
30	theorem	`PaperXI.BreakMoving.productEquiv_coe`
34	theorem	`PaperXI.BreakMoving.productEquiv_symm_coe`
39	theorem	`PaperXI.BreakMoving.productEquiv_intertwines`
51	def	`PaperXI.BreakMoving.equiv`
57	theorem	`PaperXI.BreakMoving.equiv_coe`
62	theorem	`PaperXI.BreakMoving.equiv_intertwines`
72	theorem	`PaperXI.BreakMoving.equiv_preserves`
89	theorem	`PaperXI.BreakMoving.isCompl_of_moving_fixedPointFree`
109	theorem	`PaperXI.BreakMoving.movingForm_nondegenerate`

A.53 `PaperXI.BreakOddPhase`

Source lines: 164.

SHA-256: `32575366712c4c41dec1087b72f1a84ef94ced9b0af536673412a053eec700cf`.

```
Source 15.3 in odd characteristic, using the moving phase of Definition
12.7. Fixed vectors do not enter a determinant on the full singular space.
```

Direct project imports: `PaperXI.BreakMoving`.

Line	Kind	Name
20	theorem	`PaperXI.BreakOddPhase.component_fixedPointFree`
42	theorem	`PaperXI.BreakOddPhase.product_of_moving_forms`
58	theorem	`PaperXI.BreakOddPhase.product`
74	theorem	`PaperXI.BreakOddPhase.determinants_ne_zero`
85	theorem	`PaperXI.BreakOddPhase.empty`
92	theorem	`PaperXI.BreakOddPhase.zero_moving_factor`

Line	Kind	Name
100	theorem	`PaperXI.BreakOddPhase.internal_intertwines`
112	theorem	`PaperXI.BreakOddPhase.internal_form`
126	theorem	`PaperXI.BreakOddPhase.internal_moving_form`
147	theorem	`PaperXI.BreakOddPhase.source`

A.54 `PaperXI.BreakQuadratic`

Source lines: 104.

SHA-256: `37bf741c07a7c2c90e5139d5bd490d61a60eccc0d2573bcf0d51e59ce81f4f6a`.

```
Actual finite orthogonal products of dyadic quadratic refinements.
The index type may be empty and the factors may have dimension zero.
```

Direct project imports: `PaperXI.DyadicAbstractOrthogonal`.

Line	Kind	Name
16	def	`PaperXI.BreakQuadratic.form`
25	theorem	`PaperXI.BreakQuadratic.form_apply`
28	theorem	`PaperXI.BreakQuadratic.form_alternating`
32	theorem	`PaperXI.BreakQuadratic.form_single_right`
42	theorem	`PaperXI.BreakQuadratic.form_nondegenerate`
51	def	`PaperXI.BreakQuadratic.refinement`
57	theorem	`PaperXI.BreakQuadratic.refinement_apply`
63	theorem	`PaperXI.BreakQuadratic.refinement_polar`
71	theorem	`PaperXI.BreakQuadratic.gauss_sum_product`
82	theorem	`PaperXI.BreakQuadratic.finrank_eq_twice_half`
91	theorem	`PaperXI.BreakQuadratic.half_finrank_product`

A.55 `PaperXI.BreakTensorFactorization`

Source lines: 77.

SHA-256: `9201c286d4ce3d4c53eda5ba1a756940f87a36ce7ddfb768a007ec158398ec24`.

```
Source 15.2: actual tensor factorization on the declared full group.
Compatible factor embeddings construct the group comparison. In the odd
symplectic branch both this comparison and scalar nondegeneracy are derived.
```

Direct project imports: `PaperXI.CentralProductComparison`, `PaperXI.HeisenbergTensorComparison`, `PaperXI.BreakHeisenbergSum`, `PaperXI.AbstractHeisenbergExtension`.

Line	Kind	Name
23	def	`PaperXI.BreakTensorFactorization.source`
53	theorem	`PaperXI.BreakTensorFactorization.odd_source`

A.56 `PaperXI.BreakTensorMap`

Source lines: 100.

SHA-256: `52fc48e66b2d78e5a85d528b3d8eccdee4b0668c3dc10b4ba49f3c44482caebb`.

```
Actual finite tensor maps of the supplied break intertwiners.
```

Direct project imports: `PaperXI.FiniteTensor`.

Line	Kind	Name
14	def	`PaperXI.BreakTensorMap.tensor`
17	theorem	`PaperXI.BreakTensorMap.tensor_tprod`
22	theorem	`PaperXI.BreakTensorMap.tensor_symm_tprod`
27	theorem	`PaperXI.BreakTensorMap.tensor_symm`
35	theorem	`PaperXI.BreakTensorMap.intertwines`
49	theorem	`PaperXI.BreakTensorMap.descends`
62	def	`PaperXI.BreakTensorMap.stabilizer`
66	theorem	`PaperXI.BreakTensorMap.stabilizer_tprod`
73	def	`PaperXI.BreakTensorMap.productCharacter`
78	theorem	`PaperXI.BreakTensorMap.productCharacter_apply`
82	theorem	`PaperXI.BreakTensorMap.scalar_intertwines`

A.57 `PaperXI.CanonicalDVRValuation`

Source lines: 225.

SHA-256: `d30233e7e4f5fffd0c9ee97fc3e54a765389da26cf8f917bd7c158e249eae6a1`.

```
# The canonical adic valuation of the constructed upper DVR

The fraction field constructed in `LocalFieldResidueExtension` is literally the
fraction field of the constructed DVR.  Consequently Mathlib's maximal-ideal
valuation identifies that DVR with the valuation unit ball.  This file records
that identification and proves the unit/uniformizer normalization required by
the local-field attachment.
```

Direct project imports: `PaperXI.LocalFieldResidueExtension`.

Line	Kind	Name
28	abbrev	`PaperXI.CanonicalDVRValuation.canonicalValuation`
32	def	`PaperXI.CanonicalDVRValuation.integerRingEquiv`
37	theorem	`PaperXI.CanonicalDVRValuation.integerRingEquiv_coe`
43	theorem	`PaperXI.CanonicalDVRValuation.integerRing_image`
50	theorem	`PaperXI.CanonicalDVRValuation.integer_iff_exists_lift`
59	theorem	`PaperXI.CanonicalDVRValuation.unit_value`
65	theorem	`PaperXI.CanonicalDVRValuation.maximalIdeal_iff_value_lt_one`
72	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_isUniformizer`
85	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_value`
92	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_value_lt_one`
108	abbrev	`PaperXI.CanonicalDVRValuation.upperRing`
109	abbrev	`PaperXI.CanonicalDVRValuation.upperField`
110	abbrev	`PaperXI.CanonicalDVRValuation.upperValuation`
115	def	`PaperXI.CanonicalDVRValuation.upperIntegerRingEquiv`
121	theorem	`PaperXI.CanonicalDVRValuation.upperIntegerRingEquiv_coe`
129	theorem	`PaperXI.CanonicalDVRValuation.upper_integerRing_image`
136	theorem	`PaperXI.CanonicalDVRValuation.upper_integer_iff_exists_lift`
144	theorem	`PaperXI.CanonicalDVRValuation.upper_unit_value`
152	theorem	`PaperXI.CanonicalDVRValuation.base_unit_value`

Line	Kind	Name
161	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_generates_upper`
171	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_isUniformizer`
182	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_value`
191	def	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer`
195	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_irreducible`
200	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_span`
207	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_isUpperUniformizer`
216	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_value`

A.58 `PaperXI.CentralCharacterAction`

Source lines: 111.

SHA-256: `7d5028051c4d9345e5ff3724fa65990af1cb4900a25053f54c82f6e7dafcf04f`.

```
An actual representation with a prescribed central character yields an
actual action of its section's twisted group algebra.
```

Direct project imports: `PaperXI.CentralExtensionCocycle`, `PaperXI.TwistedAction`.

Line	Kind	Name
15	theorem	`PaperXI.CentralCharacter.section_operators_one`
20	theorem	`PaperXI.CentralCharacter.section_operators_mul`
32	abbrev	`PaperXI.CentralCharacter.Algebra`
38	abbrev	`PaperXI.CentralCharacter.basis`
42	def	`PaperXI.CentralCharacter.toAlgebraAction`
50	theorem	`PaperXI.CentralCharacter.toAlgebraAction_basis`
59	theorem	`PaperXI.CentralCharacter.action_normal_form`
76	theorem	`PaperXI.CentralCharacter.action_stable_iff`
93	theorem	`PaperXI.CentralCharacter.action_intertwines_iff`

A.59 `PaperXI.CentralCharacterInduction`

Source lines: 62.

SHA-256: `26563eced498d52375622ece6a9b1081ff7d338c5f94399babdd60a81e749349`.

```
# The central action of actual induction

The scalar on a central element of the inducing subgroup remains the scalar on
the actual induced representation. Consequently, two different such scalars
cannot be related by a nonzero ambient intertwiner. No irreducibility hypothesis
or finite-index hypothesis is used.
```

Direct project imports: `PaperXI.OpenSubgroupSmoothInduction`.

Line	Kind	Name
22	theorem	`PaperXI.CentralCharacterInduction.ind_central_scalar`
37	theorem	`PaperXI.CentralCharacterInduction.intertwiner_eq_zero_of_scalar_ne`
51	theorem	`PaperXI.CentralCharacterInduction.ind_intertwiner_eq_zero`

A.60 PaperXI.CentralCharacterIrreducible

Source lines: 49.

SHA-256: `bb4d95bc7100a200f2b5e915893df883a1e7768c06b5e5006432e5b6c4b3d367`.

```
Irreducibility of an actual central-character representation is equivalent
to simplicity of its constructed twisted-algebra module.
```

Direct project imports: `PaperXI.CentralCharacterAction`, `PaperXI.CentralCharacterRepresentation`, `PaperXI.AlgebraActionModule`, `PaperXI.RepresentationStability`.

Line	Kind	Name
18	theorem	`PaperXI.CentralCharacter.irreducible_iff_simple`
34	theorem	`PaperXI.CentralCharacter.simpleModule`
42	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_irreducible_iff`

A.61 PaperXI.CentralCharacterRepresentation

Source lines: 83.

SHA-256: `115c7bf9c958e23517b9990efc2179bb7595d5c416e839a96ee9705c1448d3fe`.

```
Reconstruction of the actual central-character representation from an
algebra action, with both action roundtrips.
```

Direct project imports: `PaperXI.CentralCharacterAction`.

Line	Kind	Name
14	def	`PaperXI.CentralCharacter.ofAlgebraAction`
27	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_apply`
33	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_central`
40	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_section`
46	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_isUnit`
53	theorem	`PaperXI.CentralCharacter.of_toAlgebraAction`
62	theorem	`PaperXI.CentralCharacter.to_ofAlgebraAction`
74	def	`PaperXI.CentralCharacter.actionEquiv`

A.62 PaperXI.CentralExtension

Source lines: 85.

SHA-256: `738dea99a4dea853d5968a45499d8cd3804c8aae571749faa7ecd0f2beada19a`.

```
An actual central extension and normalized coordinates on its middle group.
No characteristic, finiteness, or scalar-character assumption is used.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
11	structure	`PaperXI.CentralExtension.Datum`
22	theorem	`PaperXI.CentralExtension.q_i`

Line	Kind	Name
28	def	`PaperXI.CentralExtension.sectionMap`
32	theorem	`PaperXI.CentralExtension.section_one`
35	theorem	`PaperXI.CentralExtension.q_section`
41	theorem	`PaperXI.CentralExtension.exists_coordinate`
49	def	`PaperXI.CentralExtension.coordinate`
51	theorem	`PaperXI.CentralExtension.coordinate_spec`
56	theorem	`PaperXI.CentralExtension.normal_form`
60	theorem	`PaperXI.CentralExtension.coordinate_eq_of_eq`
67	theorem	`PaperXI.CentralExtension.coordinate_one`
70	theorem	`PaperXI.CentralExtension.coordinate_section`
73	theorem	`PaperXI.CentralExtension.coordinate_i`
77	theorem	`PaperXI.CentralExtension.normal_form_unique`

A.63 `PaperXI.CentralExtensionCocycle`

Source lines: 127.

SHA-256: `20408e0fc0cb3e5f4edced01e3f8829ed5b80a905ed55a06d4545e27170151fa`.

```
The normalized central and scalar cocycles of an actual central extension.
The defect and its cocycle law are derived from group multiplication.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
15	def	`PaperXI.CentralExtension.defect`
17	theorem	`PaperXI.CentralExtension.section_mul`
22	theorem	`PaperXI.CentralExtension.defect_one_left`
25	theorem	`PaperXI.CentralExtension.defect_one_right`
28	theorem	`PaperXI.CentralExtension.normal_form_mul`
38	theorem	`PaperXI.CentralExtension.coordinate_mul`
49	theorem	`PaperXI.CentralExtension.defect_assoc`
56	def	`PaperXI.CentralExtension.centralCocycle`
60	theorem	`PaperXI.CentralExtension.normalized_centralCocycle`
64	def	`PaperXI.CentralExtension.scalarCocycle`
69	theorem	`PaperXI.CentralExtension.scalarCocycle_coeff`
72	theorem	`PaperXI.CentralExtension.normalized_scalarCocycle`
76	theorem	`PaperXI.CentralExtension.product_normal_form`
82	theorem	`PaperXI.CentralExtension.lift_commutation`
100	def	`PaperXI.CentralExtension.centralCommutator`
102	theorem	`PaperXI.CentralExtension.centralCommutator_eq`
110	theorem	`PaperXI.CentralExtension.i_centralCommutator`
116	theorem	`PaperXI.CentralExtension.centralCommutator_lift_independent`
123	theorem	`PaperXI.CentralExtension.scalar_commutator`

A.64 `PaperXI.CentralExtensionComparison`

Source lines: 61.

SHA-256: `e36d9e9e7da551d6db27596259bc69841e32a06e82d7b1c79f3bb69942a35082`.

```
The initial representation comparison along an actual extension lift,
without an odd-characteristic or coordinate-group restriction.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`, `PaperXI.HeisenbergTransport`.

Line	Kind	Name
15	def	`PaperXI.CentralExtensionComparison.equivAlong`
25	theorem	`PaperXI.CentralExtensionComparison.equivAlong_intertwines`
38	theorem	`PaperXI.CentralExtensionComparison.exists_unique_along`

A.65 `PaperXI.CentralExtensionExistence`

Source lines: 43.

SHA-256: `c073b326ebaed1aafad9eeaf1e4cbb0707dfa8112ae901e0b421f9d114227a2a`.

```
Existence is obtained from an actual simple ideal of the constructed
twisted algebra, then the reverse action construction.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`, `PaperXI.CentralCharacterRepresentation`.

Line	Kind	Name
16	theorem	`PaperXI.CentralExtensionExistence.exists_irreducible`

A.66 `PaperXI.CentralExtensionInducing`

Source lines: 208.

SHA-256: `a435de848de5be66c076c64e660fe71344453fd66a68dc1d2f842854b984317f`.

```
# Extending inducing data by a central infinite cyclic direction

The extension is a representation of an actual subgroup of the ambient group.
Its hypotheses require that the adjoined central element has no nonzero power
in the original subgroup.  No assertion about simple types, intertwining, or
admissibility is supplied by this construction.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`, `PaperXI.RepresentationStability`.

Line	Kind	Name
21	theorem	`PaperXI.CentralExtensionInducing.hfree_from_degree`
32	def	`PaperXI.CentralExtensionInducing.normalFormHom`
45	theorem	`PaperXI.CentralExtensionInducing.normalFormHom_apply`
50	def	`PaperXI.CentralExtensionInducing.extendedSubgroup`
53	theorem	`PaperXI.CentralExtensionInducing.le_extendedSubgroup`
58	theorem	`PaperXI.CentralExtensionInducing.central_mem_extendedSubgroup`
63	theorem	`PaperXI.CentralExtensionInducing.extendedSubgroup_eq_sup`
75	theorem	`PaperXI.CentralExtensionInducing.normalFormHom_injective`
91	def	`PaperXI.CentralExtensionInducing.normalFormEquiv`
97	theorem	`PaperXI.CentralExtensionInducing.normalFormEquiv_coe`
104	def	`PaperXI.CentralExtensionInducing.subgroupInclusion`
109	def	`PaperXI.CentralExtensionInducing.centralElement`

Line	Kind	Name
116	def	PaperXI.CentralExtensionInducing.productRepresentation
127	def	PaperXI.CentralExtensionInducing.extendedRepresentation
135	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_normalForm
143	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_restrict
153	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_central
167	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_irreducible
183	theorem	PaperXI.CentralExtensionInducing.extendedSubgroup_isOpen
189	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_smooth

A.67 PaperXI.CentralExtensionInductionConsequences

Source lines: 95.

SHA-256: 6e51c61426161f53ded9cdde06049b7358cf387ee47e88de993af2d98a6838f1.

```
# Actual induced candidates with prescribed central scalar

The central extension constructs nonzero smooth induced representations with
an arbitrary unit scalar on the adjoined central element.  Different scalars
give no nonzero ambient intertwiner.  These statements do not assert that the
induced representations are irreducible, admissible, or realizations of a
specified supercuspidal parameter.
```

Direct project imports: PaperXI.CentralExtensionInducing, PaperXI.CentralCharacterInduction.

Line	Kind	Name
22	def	PaperXI.CentralExtensionInductionConsequences.inducedRepresentation
31	theorem	PaperXI.CentralExtensionInductionConsequences.induced_central_scalar
41	theorem	PaperXI.CentralExtensionInductionConsequences.induced_nontrivial
50	theorem	PaperXI.CentralExtensionInductionConsequences.induced_intertwiner_eq_zero
62	theorem	PaperXI.CentralExtensionInductionConsequences.induced_not_equiv
84	theorem	PaperXI.CentralExtensionInductionConsequences.induced_isSmooth

A.68 PaperXI.CentralExtensionSectionChange

Source lines: 267.

SHA-256: 5360a7116c397b9588d620f7febc89b633f697bd8cec091dd4ecc76468e03f4f.

```
Independence of normalized quotient section for the actual cocycle and
twisted algebra of a central extension. The alternate defect is computed
in the middle group, then identified with the derived coboundary.
```

Direct project imports: PaperXI.CentralExtensionCocycle, PaperXI.TwistedRescaling, PaperXI.CentralCharacterAction.

Line	Kind	Name
18	theorem	PaperXI.CentralExtension.cocycle_eq_of_coeff_eq
27	def	PaperXI.CentralExtension.sectionFactor
29	theorem	PaperXI.CentralExtension.section_factor
33	theorem	PaperXI.CentralExtension.sectionFactor_one
38	def	PaperXI.CentralExtension.sectionDefect

Line	Kind	Name
41	theorem	`PaperXI.CentralExtension.i_sectionDefect`
46	theorem	`PaperXI.CentralExtension.sectionDefect_mul`
51	theorem	`PaperXI.CentralExtension.sectionDefect_eq_rescale`
70	def	`PaperXI.CentralExtension.sectionCocycle`
76	theorem	`PaperXI.CentralExtension.sectionCocycle_eq_rescale`
81	theorem	`PaperXI.CentralExtension.normalized_sectionCocycle`
88	def	`PaperXI.CentralExtension.sectionScalarCocycle`
94	theorem	`PaperXI.CentralExtension.sectionScalarCocycle_eq_rescale`
106	theorem	`PaperXI.CentralExtension.normalized_sectionScalarCocycle`
114	theorem	`PaperXI.CentralExtension.sectionScalarCocycle_commutator`
121	theorem	`PaperXI.CentralExtension.sectionScalarCocycle_radical`
130	def	`PaperXI.CentralExtension.sectionAlgebraEquiv`
142	theorem	`PaperXI.CentralExtension.sectionAlgebraEquiv_basis`
152	theorem	`PaperXI.CentralExtension.sectionAlgebraEquiv_symm_basis`
165	def	`PaperXI.CentralExtension.betweenSectionsEquiv`
174	theorem	`PaperXI.CentralExtension.betweenSectionsEquiv_basis`
191	theorem	`PaperXI.CentralExtension.sectionOperators_factor`
202	theorem	`PaperXI.CentralExtension.sectionOperators_mul`
214	def	`PaperXI.CentralExtension.sectionAction`
224	theorem	`PaperXI.CentralExtension.sectionAction_basis`
235	theorem	`PaperXI.CentralExtension.sectionAction_eq_transport`
254	theorem	`PaperXI.CentralExtension.sectionAction_betweenSections`

A.69 `PaperXI.CentralExtensionStructure`

Source lines: 117.

SHA-256: `601bab4de431a620e5b9f2c84ea154cd63b7336382468bdf4dd6cca2a0a1ee35`.

```
Coordinates, cardinality, and intrinsic scalar nondegeneracy of an actual
central extension. Faithfulness is an explicit hypothesis when a central-valued
nondegeneracy statement is transported through a central character.
```

Direct project imports: `PaperXI.CentralExtensionCocycle`.

Line	Kind	Name
17	def	`PaperXI.CentralExtension.coordinates`
26	theorem	`PaperXI.CentralExtension.coordinates_apply`
29	theorem	`PaperXI.CentralExtension.coordinates_symm_apply`
33	theorem	`PaperXI.CentralExtension.finite_middle`
38	theorem	`PaperXI.CentralExtension.natCard_eq`
42	theorem	`PaperXI.CentralExtension.card_eq`
48	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_iff`
69	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_of_faithful`
81	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_of_injective`
95	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_of_central`
102	theorem	`PaperXI.CentralExtension.center_eq_range`

A.70 `PaperXI.CentralExtensionUniqueness`

Source lines: 99.

SHA-256: `2e5ae550279047849c064e92382606a085429ff1dabb04de593c14006d39a9d2`.

```
Finite Stone–von Neumann for actual central extensions with nondegenerate
scalar commutator. Only the quotient is required finite. The argument works
in every characteristic and derives the dimensions of the given irreducibles.
```

Direct project imports: `PaperXI.CentralCharacterIrreducible`, `PaperXI.TwistedModules`, `PaperXI.SchurScalar`.

Line	Kind	Name
18	theorem	`PaperXI.CentralExtensionUniqueness.finiteDimensional`
33	theorem	`PaperXI.CentralExtensionUniqueness.finrank_sqrt`
47	theorem	`PaperXI.CentralExtensionUniqueness.exists_equiv`
68	def	`PaperXI.CentralExtensionUniqueness.equiv`
76	theorem	`PaperXI.CentralExtensionUniqueness.equiv_intertwines`
87	theorem	`PaperXI.CentralExtensionUniqueness.unique_scalar`

A.71 `PaperXI.CentralProduct`

Source lines: 159.

SHA-256: `9d1e6a4ece84a862ef75383b66b97af694c9ac2298fc2047a6c4844bc6aa34af`.

```
The finite central product, with an explicit central factor so that the empty
family has the correct tensor-unit center. The group is an actual quotient.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
15	abbrev	`PaperXI.CentralProduct.Ambient`
18	def	`PaperXI.CentralProduct.balanced`
23	def	`PaperXI.CentralProduct.relation`
25	theorem	`PaperXI.CentralProduct.balanced_central`
32	instance	`PaperXI.CentralProduct.relation_normal`
39	def	`PaperXI.CentralProduct.Group`
43	def	`PaperXI.CentralProduct.mk`
45	theorem	`PaperXI.CentralProduct.mk_surjective`
47	theorem	`PaperXI.CentralProduct.mk_balanced`
50	def	`PaperXI.CentralProduct.center`
52	theorem	`PaperXI.CentralProduct.center_apply`
54	theorem	`PaperXI.CentralProduct.center_injective`
69	def	`PaperXI.CentralProduct.ambientProjection`
74	theorem	`PaperXI.CentralProduct.relation_le_ker_projection`
79	def	`PaperXI.CentralProduct.projection`
82	theorem	`PaperXI.CentralProduct.projection_mk`
85	theorem	`PaperXI.CentralProduct.projection_center`
89	theorem	`PaperXI.CentralProduct.projection_surjective`
96	theorem	`PaperXI.CentralProduct.mk_centers`
105	theorem	`PaperXI.CentralProduct.range_center_eq_ker_projection`
122	theorem	`PaperXI.CentralProduct.center_commutes`
131	def	`PaperXI.CentralProduct.datum`
139	theorem	`PaperXI.CentralProduct.datum_i`

Line	Kind	Name
141	theorem	`PaperXI.CentralProduct.datum_q`
144	def	`PaperXI.CentralProduct.descend`
148	theorem	`PaperXI.CentralProduct.descend_mk`
153	theorem	`PaperXI.CentralProduct.hom_ext`

A.72 `PaperXI.CentralProductCommutator`

Source lines: 72.

SHA-256: `00857ad77d53b9578db693336a58114a88ad7c58f683a74132a7730fff7c4dbe`.

```
The actual quotient commutator is the product of the factor commutators.
Scalar nondegeneracy is inherited without assuming a faithful central character.
```

Direct project imports: `PaperXI.CentralProduct`, `PaperXI.CentralExtensionStructure`.

Line	Kind	Name
16	theorem	`PaperXI.CentralProduct.commutator_mk`
32	theorem	`PaperXI.CentralProduct.scalar_commutator_product`
48	theorem	`PaperXI.CentralProduct.radical_eq_bot`
67	theorem	`PaperXI.CentralProduct.full_center`

A.73 `PaperXI.CentralProductComparison`

Source lines: 149.

SHA-256: `0c1567c5e43684d459663ec1d8c0995f0cdd4d26a9b3b84b56e719e76dcf1a67`.

```
Actual compatible factor embeddings construct the comparison with a declared
full central extension. The desired group isomorphism is derived from exactness.
```

Direct project imports: `PaperXI.CentralProductFactors`.

Line	Kind	Name
22	def	`PaperXI.CentralProductComparison.factors`
24	theorem	`PaperXI.CentralProductComparison.factors_single`
30	theorem	`PaperXI.CentralProductComparison.factors_projection`
45	def	`PaperXI.CentralProductComparison.centerProduct`
52	theorem	`PaperXI.CentralProductComparison.factors_centers`
72	def	`PaperXI.CentralProductComparison.ambient`
76	theorem	`PaperXI.CentralProductComparison.ambient_apply`
81	theorem	`PaperXI.CentralProductComparison.ambient_balanced`
87	def	`PaperXI.CentralProductComparison.comparison`
90	theorem	`PaperXI.CentralProductComparison.comparison_mk`
93	theorem	`PaperXI.CentralProductComparison.comparison_center`
99	theorem	`PaperXI.CentralProductComparison.comparison_projection`
108	theorem	`PaperXI.CentralProductComparison.comparison_injective`
126	theorem	`PaperXI.CentralProductComparison.comparison_surjective`
138	def	`PaperXI.CentralProductComparison.equiv`
142	theorem	`PaperXI.CentralProductComparison.equiv_center`

Line	Kind	Name
146	theorem	`PaperXI.CentralProductComparison.equiv_mk`

A.74 `PaperXI.CentralProductFactors`

Source lines: 89.

SHA-256: `2c019b264e9f537a7886668193c3e1777ac631b16c10cf02acbd25e8f30b3e82`.

The factor injections and the empty-family center of the actual finite central product.

Direct project imports: `PaperXI.CentralProduct`.

Line	Kind	Name
13	def	`PaperXI.CentralProduct.factor`
16	theorem	`PaperXI.CentralProduct.factor_apply`
20	theorem	`PaperXI.CentralProduct.projection_factor`
28	theorem	`PaperXI.CentralProduct.factor_center`
43	theorem	`PaperXI.CentralProduct.factor_injective`
61	theorem	`PaperXI.CentralProduct.factor_commute`
72	theorem	`PaperXI.CentralProduct.center_surjective_of_isEmpty`
81	def	`PaperXI.CentralProduct.emptyEquiv`
86	theorem	`PaperXI.CentralProduct.emptyEquiv_center`

A.75 `PaperXI.CentralProductMap`

Source lines: 64.

SHA-256: `a5dad822167784fc5033f5187a85a8e99a172466c1f65742380c20fa01d6e4d8`.

Actual maps and isomorphisms of the central quotients induced by compatible factor maps.

Direct project imports: `PaperXI.CentralProduct`.

Line	Kind	Name
15	def	`PaperXI.CentralProduct.ambientMap`
21	def	`PaperXI.CentralProduct.map`
29	theorem	`PaperXI.CentralProduct.map_mk`
34	theorem	`PaperXI.CentralProduct.map_center`
40	def	`PaperXI.CentralProduct.congr`
55	theorem	`PaperXI.CentralProduct.congr_mk`
60	theorem	`PaperXI.CentralProduct.congr_center`

A.76 `PaperXI.ChainSharpness`

Source lines: 25.

SHA-256: `9dc676cc958252b1a0d2ad618ee5f85bcded1dbd8813022e05f9a32c85129ece`.

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
9	theorem	`PaperXI.MonodromyChain.identity_path`
16	theorem	`PaperXI.MonodromyChain.identity_shift_power_ne_zero`

A.77 `PaperXI.CharacterAttachment108`

Source lines: 73.

SHA-256: `3ab5684d139d5f725153a9bc2d64d616384a28734b3839f0bc3c1f89856219fe`.

```
Attach the actual character to the prescribed carrier. A single character
vector and character normalization imply scalar action, by irreducibility.
```

Direct project imports: `PaperXI.TypeSideDescent`, `PaperXI.CharacterNormalizer61`, `PaperXI.RepresentationStability`, `PaperXI.IntertwiningAlong105`.

Line	Kind	Name
14	theorem	`PaperXI.CharacterAttachment108.irreducible_of_restriction`
23	def	`PaperXI.CharacterAttachment108.insideCharacter`
27	theorem	`PaperXI.CharacterAttachment108.scalar_of_contains`
50	theorem	`PaperXI.CharacterAttachment108.character_of_scalar`

A.78 `PaperXI.CharacterGenerators`

Source lines: 59.

SHA-256: `5f499da97dbae7d5948989b633f51e14f7adac1468a62e22af5ec653fbc2b1a6`.

```
Triviality of the actual stabilizer discrepancy on a generating set.
```

Direct project imports: `PaperXI.CyclicBridge`.

Line	Kind	Name
12	theorem	`PaperXI.CharacterGenerators.eq_one_on_closure`
18	theorem	`PaperXI.CharacterGenerators.trivial`
25	theorem	`PaperXI.CharacterGenerators.trivial_iff`
35	theorem	`PaperXI.CharacterGenerators.bridge_from_actions`

A.79 `PaperXI.CharacterNormalizer61`

Source lines: 104.

SHA-256: `4d1fb07d5d106c49247b3900dea666ce7859054359e7328de46b86d3cf3cdf1f`.

```
# The literal normalizer of a subgroup-character pair

The group is defined from the graph of the actual character. It is therefore
sensitive to both its domain and its values, rather than a separately supplied
group label. Compact parts are formed by intersection with a compact subgroup.
```

Direct project imports: `PaperXI.NormalizerIntertwining60`.

Line	Kind	Name
18	def	`PaperXI.CharacterNormalizer61.graph`
21	def	`PaperXI.CharacterNormalizer61.normalizer`
25	theorem	`PaperXI.CharacterNormalizer61.conjugation_mem`
34	theorem	`PaperXI.CharacterNormalizer61.character_covariance`
45	theorem	`PaperXI.CharacterNormalizer61.normalizes_domain`
57	theorem	`PaperXI.CharacterNormalizer61.domain_le_normalizer`
76	theorem	`PaperXI.CharacterNormalizer61.character_intertwinedAt`
85	def	`PaperXI.CharacterNormalizer61.compactPart`
91	theorem	`PaperXI.CharacterNormalizer61.normalizer_open`
95	theorem	`PaperXI.CharacterNormalizer61.compactPart_compact_open`

A.80 `PaperXI.CharacterPairing108`

Source lines: 78.

SHA-256: `e86637f400e8ad723c2b13db11b5fc48fb9110e11a1d764d1925fed4a1e06dfb`.

```
Exact character pairing on a finite intersection. The index equality used
for dimension one is an explicit, separate arithmetic obligation.
```

Direct project imports: `PaperXI.HeisenbergTrace108`.

Line	Kind	Name
22	theorem	`PaperXI.CharacterPairing108.dimension_formula`
65	theorem	`PaperXI.CharacterPairing108.dimension_one_iff`

A.81 `PaperXI.CharacterProjector`

Source lines: 100.

SHA-256: `c3d3017695299f3ef67a28bee5c519bbe37aa9c0768e52c4fdbb8c9be912d7cc`.

```
The literal character element of a finite complex group algebra.
Centrality and scalar action are proved from character orthogonality.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.CharacterProjector.element`
20	theorem	`PaperXI.CharacterProjector.coeff_element`
26	theorem	`PaperXI.CharacterProjector.central`
42	theorem	`PaperXI.CharacterProjector.action`
50	theorem	`PaperXI.CharacterProjector.central_scalar`
58	theorem	`PaperXI.CharacterProjector.nontrivial_space`
66	theorem	`PaperXI.CharacterProjector.degree_ne_zero`
72	theorem	`PaperXI.CharacterProjector.irreducible_action`

A.82 PaperXI.CharacterSupport108

Source lines: 71.

SHA-256: `4a1a02bf6685a09772dbbe49ad1f29037975b2bbc299e736b1eba53df4aa8960`.

```
Equality of actual intertwining supports after character attachment. The
centralizer beta-intertwining input follows from positive Heisenberg dimension.
```

Direct project imports: `PaperXI.CharacterAttachment108`, `PaperXI.FullTypeIntertwining105`, `PaperXI.InterwiningSpace107`.

Line	Kind	Name
16	def	`PaperXI.CharacterSupport108.intoFirst`
26	theorem	`PaperXI.CharacterSupport108.restricted_iff_character`
38	theorem	`PaperXI.CharacterSupport108.full_iff_character`
48	theorem	`PaperXI.CharacterSupport108.full_iff_restricted`
54	theorem	`PaperXI.CharacterSupport108.existsAt_of_dimension_pos`
64	theorem	`PaperXI.CharacterSupport108.beta_of_dimension_pos`

A.83 PaperXI.CharacterTwist

Source lines: 54.

SHA-256: `2f04977b3efe71419a28d3985c9ca035ba1e9f43cccb4779638ea69098b5f58e`.

```
Scalar twisting and its literal tensor comparison.
```

Direct project imports: `PaperXI.CyclicInducedCoordinates`.

Line	Kind	Name
11	def	`PaperXI.CharacterTwist.twist`
16	theorem	`PaperXI.CharacterTwist.twist_apply`
20	def	`PaperXI.CharacterTwist.tensorEquiv`
27	def	`PaperXI.CharacterTwist.mapEquiv`
37	theorem	`PaperXI.CharacterTwist.restrict_eq`
42	theorem	`PaperXI.CharacterTwist.power_action`
48	theorem	`PaperXI.CharacterTwist.closing_scalar`

A.84 PaperXI.CliffordAction

Source lines: 86.

SHA-256: `aff52bad200146645c61ebf8647615481ee1f908eb4fa8aa5ccdd39bf79393bf`.

```
The full Clifford choice comparison and irreducibility for every
section, intertwiner family, and equivariant tensor identification.
```

Direct project imports: `PaperXI.CliffordIrreducible`, `PaperXI.CliffordChoices`.

Line	Kind	Name
31	theorem	`PaperXI.CliffordAction.all_choices`
56	theorem	`PaperXI.CliffordAction.irreducible_all_choices`

A.85 `PaperXI.CliffordChoices`

Source lines: 148.

SHA-256: `1e5599eae92df49fe01d3a846e8a21e070e97ca8aa79461ebafdfaf666c9a9e0`.

```
Choice comparisons on the literal Clifford multiplicity operators.
The comparing scalar is derived from Schur's lemma.
```

Direct project imports: `PaperXI.CliffordMultiplicity`, `PaperXI.ProjectiveCocycle`.

Line	Kind	Name
16	theorem	`PaperXI.CliffordChoices.operator_rescale`
32	theorem	`PaperXI.CliffordChoices.existsUnique_intertwiner_scalar`
46	def	`PaperXI.CliffordChoices.intertwinerScalar`
52	theorem	`PaperXI.CliffordChoices.intertwinerScalar_spec`
61	theorem	`PaperXI.CliffordChoices.operator_choice`
72	def	`PaperXI.CliffordChoices.representativeFactor`
79	theorem	`PaperXI.CliffordChoices.representative_factorization`
85	def	`PaperXI.CliffordChoices.representativeIntertwiner`
103	theorem	`PaperXI.CliffordChoices.representativeIntertwiner_apply`
110	theorem	`PaperXI.CliffordChoices.operator_representative`
126	def	`PaperXI.CliffordChoices.choiceCochain`
135	theorem	`PaperXI.CliffordChoices.arbitrary_choices`

A.86 `PaperXI.CliffordCocycle`

Source lines: 98.

SHA-256: `915d42076be1fed3ce0abb96e2e74367ebbd9d51d78dd6220a3cd73f37a04354`.

```
The Clifford cocycle is extracted from the literal multiplicity action.
The scalar on the multiplicity space is the inverse of the Schur scalar
comparing the subgroup intertwiners.
```

Direct project imports: `PaperXI.CliffordChoices`.

Line	Kind	Name
16	def	`PaperXI.CliffordCocycle.scalar`
25	theorem	`PaperXI.CliffordCocycle.scalar_spec`
38	theorem	`PaperXI.CliffordCocycle.operator_defect`
52	theorem	`PaperXI.CliffordCocycle.sectionOperator_mul`
67	theorem	`PaperXI.CliffordCocycle.sectionOperator_ne_zero`
77	def	`PaperXI.CliffordCocycle.cocycle`
85	theorem	`PaperXI.CliffordCocycle.cocycle_coeff`
91	theorem	`PaperXI.CliffordCocycle.normalized`

A.87 PaperXI.CliffordCocycleChoices

Source lines: 87.

SHA-256: `b81243b34012e08c729b15becd2058df413240d7047d7975e8f6d119746f4bd8`.

```
Explicit cocycle and commutator comparisons for all actual section and
intertwiner choices in the Clifford construction.
```

Direct project imports: `PaperXI.CliffordCocycle`.

Line	Kind	Name
18	def	`PaperXI.CliffordCocycleChoices.choiceCocycle`
23	theorem	`PaperXI.CliffordCocycleChoices.choiceOperator_mul`
35	theorem	`PaperXI.CliffordCocycleChoices.choiceCocycle_unique`
55	theorem	`PaperXI.CliffordCocycleChoices.commutator_choice`
62	theorem	`PaperXI.CliffordCocycleChoices.radical_choice`
68	theorem	`PaperXI.CliffordCocycleChoices.choiceCochain_one`
80	theorem	`PaperXI.CliffordCocycleChoices.choiceCocycle_normalized`

A.88 PaperXI.CliffordCocycleClass

Source lines: 111.

SHA-256: `88207bfc7c10216386294ac5631533d2b07f6a592136a204d91b41553bd7711c`.

```
The class of the actual Clifford multiplicity cocycle in the explicit
quotient of normalized cocycles by normalized scalar coboundaries.
```

Direct project imports: `PaperXI.CocycleClass`, `PaperXI.CliffordCocycleChoices`, `PaperXI.CliffordTransport`, `PaperXI.CliffordSource`.

Line	Kind	Name
21	def	`PaperXI.CliffordCocycleClass.value`
24	theorem	`PaperXI.CliffordCocycleClass.commutator_value`
28	theorem	`PaperXI.CliffordCocycleClass.radical_value`
32	theorem	`PaperXI.CliffordCocycleClass.choice_class`
42	theorem	`PaperXI.CliffordCocycleClass.class_of_multiplication`
67	theorem	`PaperXI.CliffordCocycleClass.value_transport`
82	def	`PaperXI.CliffordCocycleClass.sourceValue`
101	theorem	`PaperXI.CliffordCocycleClass.source_transport`

A.89 PaperXI.CliffordDatum

Source lines: 112.

SHA-256: `7321b347ed121e9981fdb5e48e5cdcc56ffa0b1ed4c556fb547dcc4a7eab8b5b`.

```
Actual restriction, conjugation and normalized quotient representatives
for the Clifford multiplicity construction.
```

Direct project imports: `PaperXI.QuotientRepresentation`.

Line	Kind	Name
14	abbrev	`PaperXI.CliffordDatum.restriction`
18	def	`PaperXI.CliffordDatum.conjugation`
23	theorem	`PaperXI.CliffordDatum.conjugation_coe`
26	theorem	`PaperXI.CliffordDatum.conjugation_one`
29	theorem	`PaperXI.CliffordDatum.conjugation_mul`
34	theorem	`PaperXI.CliffordDatum.conjugation_inv_cancel`
40	abbrev	`PaperXI.CliffordDatum.conjugate`
44	def	`PaperXI.CliffordDatum.Invariant`
48	def	`PaperXI.CliffordDatum.sectionMap`
53	theorem	`PaperXI.CliffordDatum.section_one`
56	theorem	`PaperXI.CliffordDatum.quotient_section`
64	def	`PaperXI.CliffordDatum.defect`
71	theorem	`PaperXI.CliffordDatum.defect_coe`
74	theorem	`PaperXI.CliffordDatum.section_mul`
78	theorem	`PaperXI.CliffordDatum.defect_one_left`
81	theorem	`PaperXI.CliffordDatum.defect_one_right`
85	def	`PaperXI.CliffordDatum.coordinate`
92	theorem	`PaperXI.CliffordDatum.normal_form`
97	def	`PaperXI.CliffordDatum.sectionIntertwiner`
106	theorem	`PaperXI.CliffordDatum.sectionIntertwiner_one`

A.90 `PaperXI.CliffordEnvelope`

Source lines: 147.

SHA-256: `550b0e257a986100dd3ddf95022f7766f8366363760787b3b28b03c08cc04548`.

```
The actual elementary commutator envelope of the source Clifford datum.
No cocycle, projective operator family, bilinear form or transport on the
multiplicity space is supplied in place of its construction.
```

Direct project imports: `PaperXI.CliffordSource`, `PaperXI.CliffordTransport`, `PaperXI.ElementaryEnvelopeTransport`.

Line	Kind	Name
23	def	`PaperXI.CliffordEnvelope.pairing`
26	theorem	`PaperXI.CliffordEnvelope.pairing_coe`
31	theorem	`PaperXI.CliffordEnvelope.pairing_self`
35	abbrev	`PaperXI.CliffordEnvelope.Space`
40	def	`PaperXI.CliffordEnvelope.form`
43	theorem	`PaperXI.CliffordEnvelope.form_isAlt`
46	theorem	`PaperXI.CliffordEnvelope.form_nondegenerate`
49	theorem	`PaperXI.CliffordEnvelope.finiteDimensional`
54	theorem	`PaperXI.CliffordEnvelope.form_on_classes`
77	theorem	`PaperXI.CliffordEnvelope.commutator_preserved`
92	theorem	`PaperXI.CliffordEnvelope.exists_normalized_cochain`
110	theorem	`PaperXI.CliffordEnvelope.pairing_preserved`
118	def	`PaperXI.CliffordEnvelope.equiv`
124	theorem	`PaperXI.CliffordEnvelope.equiv_mk`
131	theorem	`PaperXI.CliffordEnvelope.equiv_form`

Line	Kind	Name
139	theorem	PaperXI.CliffordEnvelope.equiv_independent

A.91 PaperXI.CliffordEvaluation

Source lines: 242.

SHA-256: 405e9bf8ba1bdf8eba00ab3d6063ec48594e3b24d4d919d5a5b862e76bd4effd.

Direct project imports: PaperXI.IsotypicEvaluation, PaperXI.FiniteRepresentation, PaperXI.CliffordMultiplicity.

Line	Kind	Name
19	def	PaperXI.CliffordEvaluation.map
28	theorem	PaperXI.CliffordEvaluation.map_tmul
32	theorem	PaperXI.CliffordEvaluation.map_action
40	theorem	PaperXI.CliffordEvaluation.bijective
55	def	PaperXI.CliffordEvaluation.equiv
60	theorem	PaperXI.CliffordEvaluation.equiv_tmul
64	theorem	PaperXI.CliffordEvaluation.equiv_action
71	theorem	PaperXI.CliffordEvaluation.multiplicity_nontrivial
80	def	PaperXI.CliffordEvaluation.choiceMap
90	theorem	PaperXI.CliffordEvaluation.choiceMap_apply
96	theorem	PaperXI.CliffordEvaluation.choiceMap_injective
107	def	PaperXI.CliffordEvaluation.choiceEquiv
116	theorem	PaperXI.CliffordEvaluation.choiceEquiv_apply
123	theorem	PaperXI.CliffordEvaluation.exists_unique_choice
146	theorem	PaperXI.CliffordEvaluation.map_operator
156	theorem	PaperXI.CliffordEvaluation.operator_unique
166	theorem	PaperXI.CliffordEvaluation.choice_operator_conjugation
184	def	PaperXI.CliffordEvaluation.choiceOperator
194	theorem	PaperXI.CliffordEvaluation.choiceOperator_tensor
206	theorem	PaperXI.CliffordEvaluation.exists_unique_choice_operator
227	theorem	PaperXI.CliffordEvaluation.restriction_bijective
234	theorem	PaperXI.CliffordEvaluation.restriction_multiplicity_nontrivial

A.92 PaperXI.CliffordHeisenbergAttachment

Source lines: 144.

SHA-256: bfc127d6190ce5ad509d5e37b1b9507b92996f620a90657c855d90516c6648e7.

```
Source-facing Corollary 9.9. The form comes from the actual Clifford
multiplicity cocycle. Odd characteristic uses its intrinsic half-form group;
characteristic two retains a declared quadratic refinement and constructs a
bilinear lift. No irreducible Heisenberg representation is supplied as input.
```

Direct project imports: PaperXI.OddEnvelopeHeisenberg, PaperXI.DyadicHeisenbergLift, PaperXI.CliffordEnvelope.

Line	Kind	Name
27	def	PaperXI.CliffordHeisenbergAttachment.oddFactor
32	theorem	PaperXI.CliffordHeisenbergAttachment.odd_commutator_on_classes
50	theorem	PaperXI.CliffordHeisenbergAttachment.odd_exists
72	theorem	PaperXI.CliffordHeisenbergAttachment.odd_unique
91	def	PaperXI.CliffordHeisenbergAttachment.dyadicFactor
98	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_square
106	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_exists
128	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_unique

A.93 PaperXI.CliffordIrreducible

Source lines: 98.

SHA-256: 39476ad63852c2ef28b547eed56fe8805910a7beabbdde97f4cdbef6b5b83b63.

Canonical evaluation and irreducibility on the actual Clifford multiplicity space. All finite dimensions and tensor-image properties are derived from the finite-group irreducibles and genuine module isotypy.

Direct project imports: PaperXI.CliffordMultiplicity, PaperXI.CliffordEvaluation, PaperXI.TensorSubspace.

Line	Kind	Name
21	abbrev	PaperXI.CliffordIrreducible.Isotypic
25	theorem	PaperXI.CliffordIrreducible.multiplicity_finiteDimensional
31	theorem	PaperXI.CliffordIrreducible.multiplicity_nontrivial
37	def	PaperXI.CliffordIrreducible.evaluation
41	theorem	PaperXI.CliffordIrreducible.evaluation_tmul
47	theorem	PaperXI.CliffordIrreducible.section_action
64	theorem	PaperXI.CliffordIrreducible.group_action
80	theorem	PaperXI.CliffordIrreducible.irreducible

A.94 PaperXI.CliffordMultiplicity

Source lines: 241.

SHA-256: 9ad23300239f1a6cb777e10ea3a8efb66596640c0644862baf2db01c0089b1df.

Invertible operators on the actual space of subgroup intertwiners.

Direct project imports: PaperXI.CliffordDatum.

Line	Kind	Name
13	abbrev	PaperXI.CliffordMultiplicity.Space
16	theorem	PaperXI.CliffordMultiplicity.space_intertwines
20	theorem	PaperXI.CliffordMultiplicity.inverse_intertwines
27	theorem	PaperXI.CliffordMultiplicity.action_conjugation
35	def	PaperXI.CliffordMultiplicity.operatorMap
48	theorem	PaperXI.CliffordMultiplicity.operatorMap_apply
54	def	PaperXI.CliffordMultiplicity.inverseMap

Line	Kind	Name
74	theorem	`PaperXI.CliffordMultiplicity.inverseMap_apply`
80	def	`PaperXI.CliffordMultiplicity.operator`
100	theorem	`PaperXI.CliffordMultiplicity.operator_apply`
105	theorem	`PaperXI.CliffordMultiplicity.operator_symm_apply`
111	def	`PaperXI.CliffordMultiplicity.sectionOperator`
116	theorem	`PaperXI.CliffordMultiplicity.sectionOperator_one`
128	theorem	`PaperXI.CliffordMultiplicity.intertwiner_apply`
134	def	`PaperXI.CliffordMultiplicity.productIntertwiner`
145	theorem	`PaperXI.CliffordMultiplicity.productIntertwiner_apply`
152	def	`PaperXI.CliffordMultiplicity.actionEquiv`
165	theorem	`PaperXI.CliffordMultiplicity.actionEquiv_apply`
169	def	`PaperXI.CliffordMultiplicity.subgroupIntertwiner`
185	theorem	`PaperXI.CliffordMultiplicity.subgroupIntertwiner_apply`
191	def	`PaperXI.CliffordMultiplicity.defectIntertwiner`
210	theorem	`PaperXI.CliffordMultiplicity.defectIntertwiner_apply`
215	theorem	`PaperXI.CliffordMultiplicity.operator_product`
227	theorem	`PaperXI.CliffordMultiplicity.operator_subgroup_mul`

A.95 `PaperXI.CliffordPacket`

Source lines: 68.

SHA-256: `620bdf1b1eaca5bc5f86831a95fd342e72fe43e7fb697fd0590d56a03cc92533`.

```
Source 9.6: the actual Clifford action algebra map is an isomorphism,
and the actual multiplicity dimension is the square root of the quotient order.
```

Direct project imports: `PaperXI.CliffordSource`, `PaperXI.TwistedActionEquivalence`.

Line	Kind	Name
20	def	`PaperXI.CliffordPacket.action`
27	theorem	`PaperXI.CliffordPacket.action_basis`
38	theorem	`PaperXI.CliffordPacket.action_bijective`
44	def	`PaperXI.CliffordPacket.actionEquiv`
49	theorem	`PaperXI.CliffordPacket.actionEquiv_apply`
55	theorem	`PaperXI.CliffordPacket.finrank_sq`
65	theorem	`PaperXI.CliffordPacket.finrank_sqrt`

A.96 `PaperXI.CliffordSource`

Source lines: 48.

SHA-256: `f0b150418de3d56a1fcd7799169a9937e6b9746b2dc6a75c79de625bde828c88`.

```
The normalized cocycle and literal projective action for the source
Clifford datum, with all auxiliary dimensions and nontriviality derived.
```

Direct project imports: `PaperXI.CliffordIrreducible`, `PaperXI.CliffordCocycle`.

Line	Kind	Name
20	def	PaperXI.CliffordSource.cocycle
26	theorem	PaperXI.CliffordSource.normalized
32	theorem	PaperXI.CliffordSource.operator_mul
42	theorem	PaperXI.CliffordSource.projective_irreducible

A.97 PaperXI.CliffordStabilizer

Source lines: 216.

SHA-256: 6bc6cae8489f5025d88e5dfe8375db8a056c899365557a772e5188998ff689fc.

```
The honest action induced on a normal quotient by ambient group
automorphisms. No coherent choices of representation intertwiners enter.
```

Direct project imports: PaperXI.CliffordDatum, PaperXI.CliffordSource, PaperXI.CliffordEnvelope, PaperXI.CliffordTransport, PaperXI.ElementaryEnvelopeTransport.

Line	Kind	Name
20	def	PaperXI.CliffordStabilizer.quotientAction
36	theorem	PaperXI.CliffordStabilizer.quotientAction_mk
41	theorem	PaperXI.CliffordStabilizer.quotientAction_comp
47	theorem	PaperXI.CliffordStabilizer.quotientAction_unique
67	def	PaperXI.CliffordStabilizer.envelopeAction
70	theorem	PaperXI.CliffordStabilizer.envelopeAction_mk
76	theorem	PaperXI.CliffordStabilizer.envelopeAction_form
89	theorem	PaperXI.CliffordStabilizer.envelopeAction_equivariant
137	theorem	PaperXI.CliffordStabilizer.source_commutator_preserved
151	def	PaperXI.CliffordStabilizer.sourceAction
157	theorem	PaperXI.CliffordStabilizer.sourceAction_mk
163	theorem	PaperXI.CliffordStabilizer.sourceAction_form
174	theorem	PaperXI.CliffordStabilizer.sourceAction_independent
200	theorem	PaperXI.CliffordStabilizer.sourceAction_equivariant

A.98 PaperXI.CliffordTransport

Source lines: 280.

SHA-256: 3b10aa513a5ba4f75eee3ca0e288cb4683b0a0044f3c22cf1bfaf031a568a995.

```
Actual transport of subgroup intertwiners and multiplicity operators
along isomorphisms of the original representation data.
```

Direct project imports: PaperXI.CliffordCocycle.

Line	Kind	Name
15	def	PaperXI.CliffordTransport.subgroupEquiv
26	theorem	PaperXI.CliffordTransport.subgroupEquiv_coe
30	theorem	PaperXI.CliffordTransport.subgroupEquiv_conjugation
39	def	PaperXI.CliffordTransport.quotientEquiv
43	theorem	PaperXI.CliffordTransport.quotientEquiv_mk

Line	Kind	Name
50	theorem	`PaperXI.CliffordTransport.quotientEquiv_refl`
59	theorem	`PaperXI.CliffordTransport.quotientEquiv_comp`
76	theorem	`PaperXI.CliffordTransport.source_apply`
80	theorem	`PaperXI.CliffordTransport.source_inverse_apply`
88	theorem	`PaperXI.CliffordTransport.target_apply`
91	theorem	`PaperXI.CliffordTransport.target_inverse_apply`
99	def	`PaperXI.CliffordTransport.homMap`
113	theorem	`PaperXI.CliffordTransport.homMap_apply`
117	def	`PaperXI.CliffordTransport.inverseHomMap`
130	def	`PaperXI.CliffordTransport.homEquiv`
142	theorem	`PaperXI.CliffordTransport.homEquiv_apply`
145	theorem	`PaperXI.CliffordTransport.homEquiv_symm_apply`
151	def	`PaperXI.CliffordTransport.transportedIntertwiner`
162	theorem	`PaperXI.CliffordTransport.transportedIntertwiner_apply`
167	theorem	`PaperXI.CliffordTransport.operator_transport`
178	theorem	`PaperXI.CliffordTransport.operator_transport_conj`
189	theorem	`PaperXI.CliffordTransport.transported_representative`
198	def	`PaperXI.CliffordTransport.cochain`
206	theorem	`PaperXI.CliffordTransport.sectionOperator_transport`
230	theorem	`PaperXI.CliffordTransport.cochain_one`
243	theorem	`PaperXI.CliffordTransport.cocycle_transport`
265	theorem	`PaperXI.CliffordTransport.commutator_preserved`

A.99 `PaperXI.CocycleBasis`

Source lines: 33.

SHA-256: `5ba62f3c665922695e62725e35c969a9b223859a7dee0cad1163de238e51213e`.

```
Paper XI, Proposition 8.1: the coefficient change in an actual algebra basis.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
15	theorem	`PaperXI.CocycleCommutator.rescaled_basis_mul`
29	theorem	`PaperXI.CocycleCommutator.rescaled_basis_commutator`

A.100 `PaperXI.CocycleClass`

Source lines: 168.

SHA-256: `393677b19c265df8ad023c34ede0aa8a8cb38f60e1bb050ef0b4282876a302bd`.

```
The explicit quotient of normalized cocycles by normalized scalar coboundaries.
This is a concrete cohomological-class model; no identification with a separate
Mathlib group-cohomology construction is assumed.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
14	theorem	`PaperXI.CocycleClass.cocycle_ext`
21	theorem	`PaperXI.CocycleClass.rescale_one`
27	theorem	`PaperXI.CocycleClass.rescale_trans`
34	abbrev	`PaperXI.CocycleClass.NormalizedCocycle`
38	def	`PaperXI.CocycleClass.Related`
41	theorem	`PaperXI.CocycleClass.related_refl`
44	theorem	`PaperXI.CocycleClass.related_symm`
50	theorem	`PaperXI.CocycleClass.related_trans`
58	def	`PaperXI.CocycleClass.coboundarySetoid`
64	def	`PaperXI.CocycleClass.Class`
67	def	`PaperXI.CocycleClass.of`
71	theorem	`PaperXI.CocycleClass.of_eq_iff`
76	theorem	`PaperXI.CocycleClass.of_eq_of_rescale`
80	theorem	`PaperXI.CocycleClass.of_rescale`
86	def	`PaperXI.CocycleClass.commutator`
95	theorem	`PaperXI.CocycleClass.commutator_of`
99	def	`PaperXI.CocycleClass.radical`
107	theorem	`PaperXI.CocycleClass.radical_of`
111	def	`PaperXI.CocycleClass.pullbackCocycle`
115	theorem	`PaperXI.CocycleClass.normalized_pullback`
119	theorem	`PaperXI.CocycleClass.pullback_rescale`
126	def	`PaperXI.CocycleClass.pullback`
136	theorem	`PaperXI.CocycleClass.pullback_of`
140	def	`PaperXI.CocycleClass.congrEquiv`
164	theorem	`PaperXI.CocycleClass.congrEquiv_of`

A.101 `PaperXI.CocycleCommutator`

Source lines: 122.

SHA-256: `0b8b7f046bf6d69122f05e1838481020360574431927 62a7e89fa4e374f6d4df.`

```
Paper XI, Propositions 8.1–8.2. Multiplicative notation for the abelian
index group. Values may lie in any commutative group, including ℂˣ.
```

Direct project imports: none.

Line	Kind	Name
14	structure	`PaperXI.CocycleCommutator.Cocycle`
18	def	`PaperXI.CocycleCommutator.Normalized`
22	def	`PaperXI.CocycleCommutator.rescale`
35	def	`PaperXI.CocycleCommutator.commutator`
37	theorem	`PaperXI.CocycleCommutator.commutator_self`
40	theorem	`PaperXI.CocycleCommutator.commutator_swap`
43	theorem	`PaperXI.CocycleCommutator.commutator_rescale`
48	theorem	`PaperXI.CocycleCommutator.commutator_mul_left`
68	theorem	`PaperXI.CocycleCommutator.commutator_mul_right`
73	theorem	`PaperXI.CocycleCommutator.commutator_one_left`
78	theorem	`PaperXI.CocycleCommutator.commutator_one_right`
82	def	`PaperXI.CocycleCommutator.characterMap`

Line	Kind	Name
90	theorem	`PaperXI.CocycleCommutator.characterMap_apply`
94	def	`PaperXI.CocycleCommutator.radical`
96	theorem	`PaperXI.CocycleCommutator.mem_radical_iff`
103	theorem	`PaperXI.CocycleCommutator.radical_rescale`
110	theorem	`PaperXI.CocycleCommutator.normalized_rescale`
114	theorem	`PaperXI.CocycleCommutator.normalized_rescale_iff`

A.102 `PaperXI.CocycleQuotient`

Source lines: 85.

SHA-256: `e4fd2024e7a2b19c99fe508d778300fc161a53761c5ed2db8f0e5c0091101b51`.

```
Paper XI, Proposition 8.2: the actual quotient by the commutator radical.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
11	def	`PaperXI.CocycleCommutator.quotientCharacter`
14	theorem	`PaperXI.CocycleCommutator.quotientCharacter_mk`
17	theorem	`PaperXI.CocycleCommutator.radical_le_quotientCharacter_ker`
25	def	`PaperXI.CocycleCommutator.quotientPairing`
43	theorem	`PaperXI.CocycleCommutator.quotientPairing_mk`
47	theorem	`PaperXI.CocycleCommutator.quotientPairing_self`
52	theorem	`PaperXI.CocycleCommutator.quotientPairing_swap`
60	theorem	`PaperXI.CocycleCommutator.quotientPairing_nondegenerate_left`
74	theorem	`PaperXI.CocycleCommutator.quotientPairing_nondegenerate_right`
82	theorem	`PaperXI.CocycleCommutator.quotientCharacter_injective`

A.103 `PaperXI.CommutatorEnvelope`

Source lines: 160.

SHA-256: `bc6e6fc84b1c63fc1bb70568bc6eb1c3137c5f9f781783990c9c1bc51a6453b8`.

```
Exact universal properties of the quotient detected by the scalar
commutator. A pairing descends through A/L precisely when L lies in its
radical; that descended pairing is nondegenerate precisely at the radical.
```

Direct project imports: `PaperXI.CocycleQuotient`.

Line	Kind	Name
14	def	`PaperXI.CommutatorEnvelope.descendedCharacter`
17	theorem	`PaperXI.CommutatorEnvelope.descendedCharacter_mk`
21	theorem	`PaperXI.CommutatorEnvelope.le_descendedCharacter_ker`
29	def	`PaperXI.CommutatorEnvelope.descendedPairing`
48	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_mk`
54	theorem	`PaperXI.CommutatorEnvelope.descends_iff`
68	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_unique`
81	theorem	`PaperXI.CommutatorEnvelope.existsUnique_descendedPairing`

Line	Kind	Name
89	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_self`
95	theorem	`PaperXI.CommutatorEnvelope.nondegenerate_iff`
116	def	`PaperXI.CommutatorEnvelope.quotientFactor`
120	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_mk`
125	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_surjective`
132	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_pairing`
140	theorem	`PaperXI.CommutatorEnvelope.existsUnique_quotientFactor`
151	theorem	`PaperXI.CommutatorEnvelope.existsUnique_injective_characterFactor`

A.104 `PaperXI.CompactDegreeSeparation`

Source lines: 37.

SHA-256: `c69f0838adb9db017a926ff1a66606634858bfb0aa89710e147b19b44a60d561`.

```
# Compact subgroups have zero integer degree

A continuous integer degree has finite image on a compact subgroup, because
the target has its usual discrete topology.  The image is a finite subgroup of
the torsion-free group `Multiplicative ℤ`, so it is trivial.  This supplies the
separation hypothesis for adjoining any element of nonzero degree.
```

Direct project imports: `PaperXI.CentralExtensionInducing`.

Line	Kind	Name
20	theorem	`PaperXI.CompactDegreeSeparation.compact_le_degree_ker`
32	theorem	`PaperXI.CompactDegreeSeparation.hfree_from_compact_degree`

A.105 `PaperXI.CompactIntersection108`

Source lines: 55.

SHA-256: `318e23f5dfa82f3cf42743e82c7783ac44508ce10be1a856aa6fc60407b2fa37`.

```
The finite model of the actual conjugate-intersection Hom space exists for
any continuous embedded compact group with a smooth finite-dimensional action.
```

Direct project imports: `PaperXI.IntersectionModel108`, `PaperXI.FiniteJointQuotient108`.

Line	Kind	Name
17	theorem	`PaperXI.CompactIntersection108.compact_pairs`
26	theorem	`PaperXI.CompactIntersection108.smooth_comp`
35	theorem	`PaperXI.CompactIntersection108.finite_quotient`
46	theorem	`PaperXI.CompactIntersection108.finrank_quotient`

A.106 `PaperXI.CompactModCenterInduction68`

Source lines: 126.

SHA-256: `b1765821b4681b903bdbaabe6a76b05200fba99f81d94cc260e9ecc3484a4065`.

```
# Full-subgroup Mackey irreducibility for compact-mod-central induction

The ambient induction and its coefficient maps are the actual Mathlib carriers.
No compact-restriction intertwining criterion is substituted for the given
full-subgroup criterion. Smoothness, scalar cancellation and compact quotient
supply the needed equivariant section by a finite orbit average.
```

Direct project imports: `PaperXI.EquivariantSplitting68`, `PaperXI.InducedTypeSpace68`.

Line	Kind	Name
25	theorem	`PaperXI.CompactModCenterInduction68.induced_central_scalar`
40	def	`PaperXI.CompactModCenterInduction68.subInclusion`
52	theorem	`PaperXI.CompactModCenterInduction68.irreducible_of_compact_mod_center_mackey`
111	theorem	`PaperXI.CompactModCenterInduction68.induction_package`

A.107 `PaperXI.CompactQuotientPeriod69`

Source lines: 52.

SHA-256: `db609bc27e26e3964e4eaa9f6739b066d277b23dd9c0b595f4b51fc80baa975a`.

```
Compact fundamental domains from a finite period modulo a compact subgroup.
```

Direct project imports: none.

Line	Kind	Name
15	theorem	`PaperXI.CompactQuotientPeriod69.compact_quotient_of_period`

A.108 `PaperXI.CompactSmoothSemisimple`

Source lines: 92.

SHA-256: `8f8329834876b5913b22197a22cc58d386a26429cb21fcb23d51947c33d93e74`.

```
Smooth representations of compact groups are locally finite dimensional.
This uses compactness of the actual group, not finiteness of the ambient group.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`, `PaperXI.LocallyFiniteSemisimple`.

Line	Kind	Name
19	def	`PaperXI.CompactSmoothSemisimple.orbitQuotient`
27	def	`PaperXI.CompactSmoothSemisimple.orbitSpan`
40	theorem	`PaperXI.CompactSmoothSemisimple.mem_orbitSpan`
48	theorem	`PaperXI.CompactSmoothSemisimple.subrepresentation_smooth`
61	theorem	`PaperXI.CompactSmoothSemisimple.orbit_finite`
70	theorem	`PaperXI.CompactSmoothSemisimple.orbitSpan_finiteDimensional`
75	theorem	`PaperXI.CompactSmoothSemisimple.subrepresentation_kernel_quotient_finite`
83	theorem	`PaperXI.CompactSmoothSemisimple.smooth_semisimple`

A.109 PaperXI.CompactTypeTensor58

Source lines: 93.

SHA-256: 885a6eef6b9a66f86ddf833e1077fdd3de5fe94a004487d6b539d70cba73bc6b.

```
# The prescribed compact factor with an actual finite quotient factor

An irreducible quotient representation is tensored with the supplied extension
of an irreducible subgroup action. The resulting compact inducing action is
constructed on the literal tensor space, is irreducible and smooth, and retains
the supplied subgroup action on the first factor exactly. No type-theoretic
compact-induction theorem or arithmetic comparison is asserted.
```

Direct project imports: PaperXI.GallagherTensorIrreducibility58, PaperXI.SmoothDegreeQuotient, PaperXI.FiniteRepresentation.

Line	Kind	Name
27	abbrev	PaperXI.CompactTypeTensor58.action
32	theorem	PaperXI.CompactTypeTensor58.restriction
39	theorem	PaperXI.CompactTypeTensor58.subgroup_action
48	theorem	PaperXI.CompactTypeTensor58.irreducible
55	theorem	PaperXI.CompactTypeTensor58.finiteDimensional
66	theorem	PaperXI.CompactTypeTensor58.smooth
82	theorem	PaperXI.CompactTypeTensor58.completed

A.110 PaperXI.CompactTypeTensorNormalizer58

Source lines: 54.

SHA-256: 1058ac054e5f073f587c3012d307d04d2924924594cd5ec137685c0bfb3e10b5.

```
# Lifting a prescribed normalizer intertwiner through the quotient tensor factor

When an actual normalizer automorphism acts trivially on the quotient, its
invertible intertwiner on the prescribed first factor supplies an invertible
intertwiner on the literal compact tensor action. The latter covariance is a
conclusion, not a hypothesis.
```

Direct project imports: PaperXI.CompactTypeTensor58.

Line	Kind	Name
26	def	PaperXI.CompactTypeTensorNormalizer58.tensorOperator
29	theorem	PaperXI.CompactTypeTensorNormalizer58.tensorOperator_tmul
34	theorem	PaperXI.CompactTypeTensorNormalizer58.tensorOperator_covariance
43	def	PaperXI.CompactTypeTensorNormalizer58.normalizerEquiv
51	theorem	PaperXI.CompactTypeTensorNormalizer58.normalizerEquiv_linear

A.111 PaperXI.ConditionalAtomicComparison

Source lines: 151.

SHA-256: ce8d1ea0f8658d4d968aeb6665d5f1d3e4cc6bb1d472ebfbfbce6215f1103a58.

```
The last comparison pipeline. Actual finite tensor intertwiners and global
parallel-section equivalences are constructed. Atom realization and normalized
pair-factor characterization remain explicit external interfaces. They are not
consequences of finite phases or of zero holonomy.
```

Direct project imports: `PaperXI.FiniteBreakAssembly`, `PaperXI.GluedPacketSections`, `PaperXI.LocalFactorIdentification`.

Line	Kind	Name
19	structure	`PaperXI.ConditionalAtomicComparison.Diagram`
27	structure	`PaperXI.ConditionalAtomicComparison.Atlas`
42	def	`PaperXI.ConditionalAtomicComparison.targetDiagram`
45	def	`PaperXI.ConditionalAtomicComparison.sourceDiagram`
49	def	`PaperXI.ConditionalAtomicComparison.atRoot`
56	theorem	`PaperXI.ConditionalAtomicComparison.atRoot_linear`
60	def	`PaperXI.ConditionalAtomicComparison.Holonomy`
65	def	`PaperXI.ConditionalAtomicComparison.leftSections`
67	def	`PaperXI.ConditionalAtomicComparison.rightSections`
70	def	`PaperXI.ConditionalAtomicComparison.glue`
75	theorem	`PaperXI.ConditionalAtomicComparison.glue_apply`
84	theorem	`PaperXI.ConditionalAtomicComparison.holonomy_iff_gauge`
119	def	`PaperXI.ConditionalAtomicComparison.finiteComparison`
127	theorem	`PaperXI.ConditionalAtomicComparison.source`

A.112 `PaperXI.ConstructedFieldFrobenius`

Source lines: 65.

SHA-256: `37abb8f28a5e22530d0f0296eaef1880e5fca49629014ef7cd762d606916b5bc`.

```
# Specialization to Paper XI's constructed spectral local field

The base of the finite Galois extension is the literal UpperField(K,s) from the
verified predecessor. The base residue cardinality in geometric normalization
is identified with the residue cardinality of that same field, by equality of
integer subrings and the induced residue equivalence.
```

Direct project imports: `PaperXI.SpectralGeometricFrobenius`, `PaperXI.SpectralLocalField`.

Line	Kind	Name
22	abbrev	`PaperXI.ConstructedFieldFrobenius.BaseField`
28	theorem	`PaperXI.ConstructedFieldFrobenius.restricted_integer_subring`
36	def	`PaperXI.ConstructedFieldFrobenius.baseIntegerEquiv`
43	def	`PaperXI.ConstructedFieldFrobenius.baseResidueEquiv`
46	theorem	`PaperXI.ConstructedFieldFrobenius.baseResidue_card`
54	theorem	`PaperXI.ConstructedFieldFrobenius.exists_normalized_field_automorphism`

A.113 `PaperXI.ContinuousHolonomyResidual`

Source lines: 132.

SHA-256: `ef0ef5f7cdf0b952317f838c65b6f9e0229b32e424b9f384ab96634cd955f060`.

```
Continuous forms of the exact residual identity and its conditioning bound.
The maps are actual continuous linear equivalences. No isometry is assumed.
```

Direct project imports: `PaperXI.HolonomyResidual`.

Line	Kind	Name
18	def	`PaperXI.ContinuousHolonomyResidual.transported`
21	theorem	`PaperXI.ContinuousHolonomyResidual.transported_apply`
24	theorem	`PaperXI.ContinuousHolonomyResidual.transported_toLinearEquiv`
28	def	`PaperXI.ContinuousHolonomyResidual.cycleDefect`
37	def	`PaperXI.ContinuousHolonomyResidual.edgeResidual`
44	theorem	`PaperXI.ContinuousHolonomyResidual.cycleDefect_toLinearMap`
52	theorem	`PaperXI.ContinuousHolonomyResidual.edgeResidual_toLinearMap`
60	theorem	`PaperXI.ContinuousHolonomyResidual.residual_factorization`
72	theorem	`PaperXI.ContinuousHolonomyResidual.residual_zero_iff`
86	theorem	`PaperXI.ContinuousHolonomyResidual.residual_norm_bound`
103	theorem	`PaperXI.ContinuousHolonomyResidual.uniform_bound`
119	theorem	`PaperXI.ContinuousHolonomyResidual.product_conditioning_bound`

A.114 `PaperXI.CoordinateDeepIdeal111`

Source lines: 79.

SHA-256: `00a4e4aef33f78b2682431bc4370b2158cb1628f7d1c702f634861314e19cb09`.

```
A common open ideal in an actual coordinate integral matrix order.
The ideal lives in the integral order, not the full field matrix algebra.
```

Direct project imports: `PaperXI.OnePlusIntersection110`.

Line	Kind	Name
21	def	`PaperXI.CoordinateDeepIdeal111.uncurryHomeomorph`
31	theorem	`PaperXI.CoordinateDeepIdeal111.exists_open_matrix_ideal_subset`
45	theorem	`PaperXI.CoordinateDeepIdeal111.matrix_ideal_open`
64	theorem	`PaperXI.CoordinateDeepIdeal111.exists_common_ideal_finite`

A.115 `PaperXI.CuspidalAnnihilation`

Source lines: 87.

SHA-256: `e56bdeb22b0d449550cd009852c029feed4d614bdc02a53d49648784b4a6abd2`.

```
Subgroup averages and annihilation of an actual central character block.
The cuspidal input is the vanishing of genuine restricted invariants.
```

Direct project imports: `PaperXI.PrimitiveCharacterBlock`, `PaperXI.FiniteAverage`.

Line	Kind	Name
16	def	`PaperXI.CuspidalAnnihilation.subgroupAverage`
20	theorem	`PaperXI.CuspidalAnnihilation.average_action`
25	theorem	`PaperXI.CuspidalAnnihilation.average_range`
31	theorem	`PaperXI.CuspidalAnnihilation.average_idempotent`
38	theorem	`PaperXI.CuspidalAnnihilation.average_zero_iff`
45	theorem	`PaperXI.CuspidalAnnihilation.own_block_action`
51	theorem	`PaperXI.CuspidalAnnihilation.left_annihilation_iff`
66	theorem	`PaperXI.CuspidalAnnihilation.right_annihilation_iff`

Line	Kind	Name
73	theorem	`PaperXI.CuspidalAnnihilation.parabolic_annihilation`
79	theorem	`PaperXI.CuspidalAnnihilation.nonzero_of_fixed_vector`

A.116 `PaperXI.CuspidalCartan105`

Source lines: 82.

SHA-256: `b2194352b6debd4ae75e3489078283c8f99b6c759d153356b9853900799a8087`.

```
An inflated finite cuspidal representation has no noncentral Cartan
intertwiners. The arithmetic Cartan decomposition remains a separate explicit
input; its cuspidal exclusion is proved on actual matrices.
```

Direct project imports: `PaperXI.CuspidalContraction105`, `PaperXI.IntertwiningAlong105`, `PaperXI.SplitRadical105`.

Line	Kind	Name
19	def	`PaperXI.CuspidalCartan105.Cuspidal`
23	theorem	`PaperXI.CuspidalCartan105.diagonal_intertwiner_zero`
48	def	`PaperXI.CuspidalCartan105.HasCartan`
56	theorem	`PaperXI.CuspidalCartan105.intertwining_support`

A.117 `PaperXI.CuspidalContraction105`

Source lines: 37.

SHA-256: `cf19708af1422fb89f6f41404483d37fb54258b7a6b1b18428ac52a31995dcbd`.

```
Finite cuspidality excludes a map on which an actual radical contracts to
identity. The averaging sum and its vanishing are derived, not supplied.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.CuspidalContraction105.NoFixed`
17	theorem	`PaperXI.CuspidalContraction105.sum_fixed`
23	theorem	`PaperXI.CuspidalContraction105.sum_eq_zero`
26	theorem	`PaperXI.CuspidalContraction105.map_eq_zero`

A.118 `PaperXI.CycleCoordinates`

Source lines: 97.

SHA-256: `0e63d65fbd8eecf42832697e5cbb2b1c5e6c7fda9b165030352c34bb3ec19004`.

```
Transfer formulas in an actual basis, including independent rescalings.
```

Direct project imports: `PaperXI.CyclicPolynomial`.

Line	Kind	Name
13	theorem	`PaperXI.CycleCoordinates.basis_eq_powers`
28	theorem	`PaperXI.CycleCoordinates.closing_of_last_step`
45	theorem	`PaperXI.CycleCoordinates.cycle_formula`
55	theorem	`PaperXI.CycleCoordinates.rescaled_entry`
62	theorem	`PaperXI.CycleCoordinates.rescaled_charpoly`
68	theorem	`PaperXI.CycleCoordinates.rescaled_det`
74	theorem	`PaperXI.CycleCoordinates.cycle_orientation`
77	theorem	`PaperXI.CycleCoordinates.cycle_local_polynomial`
87	theorem	`PaperXI.CycleCoordinates.rescaled_cycle_formula`

A.119 `PaperXI.CycleFixed`

Source lines: 138.

SHA-256: `008caab8c32493eb96d7225cdc6ff9c2b44718f2cbf85360a0ebef91be48e3dd`.

```
Fixed vectors of an actual cyclic basis, valid in every characteristic.
```

Direct project imports: `PaperXI.CycleCoordinates`.

Line	Kind	Name
11	def	`PaperXI.CycleFixed.fixedSpace`
13	theorem	`PaperXI.CycleFixed.mem_fixedSpace`
16	theorem	`PaperXI.CycleFixed.coordinate_step`
42	theorem	`PaperXI.CycleFixed.fixed_coordinates_constant`
59	theorem	`PaperXI.CycleFixed.basis_sum_ne_zero`
65	theorem	`PaperXI.CycleFixed.fixed_eq_smul_sum`
75	theorem	`PaperXI.CycleFixed.map_basis_sum`
90	theorem	`PaperXI.CycleFixed.fixedSpace_eq`
126	theorem	`PaperXI.CycleFixed.fixedSpace_finrank`

A.120 `PaperXI.CycleSpan`

Source lines: 76.

SHA-256: `a96cff77ae41d6de99c41763e94a925f42820c3e7646ee438e6fe3bbb67f8054`.

```
The span of a finite Frobenius cycle with a nonzero closing scalar.
```

Direct project imports: `PaperXI.PacketReturn`.

Line	Kind	Name
10	def	`PaperXI.CycleSpan.cycleVector`
12	def	`PaperXI.CycleSpan.cycleSpan`
15	theorem	`PaperXI.CycleSpan.power_mem`
18	theorem	`PaperXI.CycleSpan.base_mem`
21	theorem	`PaperXI.CycleSpan.map_mem`
30	theorem	`PaperXI.CycleSpan.forward_power`
33	theorem	`PaperXI.CycleSpan.backward_power`

Line	Kind	Name
37	theorem	`PaperXI.CycleSpan.inverse_wrap`
43	theorem	`PaperXI.CycleSpan.forward_stable`
55	theorem	`PaperXI.CycleSpan.backward_stable`
69	theorem	`PaperXI.CycleSpan.span_ne_bot`

A.121 `PaperXI.CyclicAction`

Source lines: 55.

SHA-256: `581f94955c014edf5c55919aa1a8a85b4fc7ec5ec07b9110e81dea08ad0648cf`.

```
Actual representations of the additive finite cyclic group.
```

Direct project imports: `PaperXI.FourierCovariance`.

Line	Kind	Name
12	abbrev	`PaperXI.CyclicAction.Action`
14	def	`PaperXI.CyclicAction.operator`
17	theorem	`PaperXI.CyclicAction.operator_zero`
20	theorem	`PaperXI.CyclicAction.operator_add`
23	theorem	`PaperXI.CyclicAction.operator_nsmul`
31	theorem	`PaperXI.CyclicAction.operator_mul`
36	theorem	`PaperXI.CyclicAction.operator_eq_generator_pow`
40	theorem	`PaperXI.CyclicAction.generator_period`
46	theorem	`PaperXI.CyclicAction.conjugation_from_generator`

A.122 `PaperXI.CyclicBridge`

Source lines: 111.

SHA-256: `c24882a79bb6079e3ec2c68c8abfdd1db6c00634d6dd9045b2be41aa67cf601b`.

```
A trivial scalar discrepancy on one cyclic generator closes an actual
semidirect-product bridge. Phase means the scalar discrepancy, not trace.
```

Direct project imports: `PaperXI.DiscrepancyCharacter`.

Line	Kind	Name
12	theorem	`PaperXI.CyclicBridge.character_trivial`
18	theorem	`PaperXI.CyclicBridge.semidirect_intertwining`
28	theorem	`PaperXI.CyclicBridge.stabilizer_covariance`
41	theorem	`PaperXI.CyclicBridge.one_phase_bridge`
63	def	`PaperXI.CyclicBridge.semidirectRepresentation`
74	theorem	`PaperXI.CyclicBridge.semidirectRepresentation_inl`
80	theorem	`PaperXI.CyclicBridge.semidirectRepresentation_inr`
88	theorem	`PaperXI.CyclicBridge.one_phase_bridge_from_actions`

A.123 PaperXI.CyclicDefect113

Source lines: 339.

SHA-256: b6fe373c44ebd8725be1683e7f851da03bcc708c896db2103e02a5a31fbb3a19.

```
Explicit coprime correction of the character defect left by scalar
cocycle comparison. Arithmetic realization is not asserted here.
```

Direct project imports: PaperXI.ScalarCocycle112.

Line	Kind	Name
17	def	PaperXI.CyclicDefect113.orbitNorm
21	def	PaperXI.CyclicDefect113.weightedCharacter
24	theorem	PaperXI.CyclicDefect113.orbitNorm_zero
27	theorem	PaperXI.CyclicDefect113.orbitNorm_succ
31	theorem	PaperXI.CyclicDefect113.weightedCharacter_zero
34	theorem	PaperXI.CyclicDefect113.weightedCharacter_succ
39	theorem	PaperXI.CyclicDefect113.power_apply_shift
45	theorem	PaperXI.CyclicDefect113.weightedCharacter_shift
59	theorem	PaperXI.CyclicDefect113.bezout_power
71	def	PaperXI.CyclicDefect113.correctingCharacter
76	theorem	PaperXI.CyclicDefect113.correctingCharacter_coboundary
92	structure	PaperXI.CyclicDefect113.ScalarLift
102	def	PaperXI.CyclicDefect113.ScalarLift.phase
104	theorem	PaperXI.CyclicDefect113.ScalarLift.phase_one
108	theorem	PaperXI.CyclicDefect113.ScalarLift.scalar_formula
121	theorem	PaperXI.CyclicDefect113.ScalarLift.phase_mul
131	theorem	PaperXI.CyclicDefect113.ScalarLift.power_base
141	theorem	PaperXI.CyclicDefect113.ScalarLift.power_scalar
152	theorem	PaperXI.CyclicDefect113.ScalarLift.norm_eq_one
162	theorem	PaperXI.CyclicDefect113.orbitNorm_coboundary
172	theorem	PaperXI.CyclicDefect113.orbitNorm_mul
176	theorem	PaperXI.CyclicDefect113.orbitNorm_div
187	def	PaperXI.CyclicDefect113.actionDefect
190	theorem	PaperXI.CyclicDefect113.actionDefect_one
195	theorem	PaperXI.CyclicDefect113.actionDefect_mul
205	def	PaperXI.CyclicDefect113.defectCharacter
211	theorem	PaperXI.CyclicDefect113.actionDefect_norm
223	def	PaperXI.CyclicDefect113.correctedCochain
227	theorem	PaperXI.CyclicDefect113.correctedCochain_one
232	theorem	PaperXI.CyclicDefect113.correctedCochain_cocycle
243	theorem	PaperXI.CyclicDefect113.correctedCochain_covariant
264	def	PaperXI.CyclicDefect113.scalarBridge
279	theorem	PaperXI.CyclicDefect113.scalarBridge_base
284	theorem	PaperXI.CyclicDefect113.scalarBridge_fixes_scalar
290	theorem	PaperXI.CyclicDefect113.scalarBridge_equivariant
308	theorem	PaperXI.CyclicDefect113.exists_equivariant_cochain
325	theorem	PaperXI.CyclicDefect113.exists_equivariant_scalarBridge

A.124 PaperXI.CyclicFrobeniusCount

Source lines: 120.

SHA-256: `b97169de1bf48a86a58bf00236bad78341081a6db9ae46f6a8e49f95c87a4f82`.

```
Fixed points and regular orbit counts for the power map on a finite cyclic
 group. No fixed-point cardinality is assumed.
```

Direct project imports: `PaperXI.FrobeniusPeriodCount`.

Line	Kind	Name
16	theorem	`PaperXI.CyclicFrobeniusCount.iterate_power`
24	theorem	`PaperXI.CyclicFrobeniusCount.fixed_iff_kernel`
40	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_gcd`
48	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_pow_gcd`
54	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_of_dvd`
60	theorem	`PaperXI.CyclicFrobeniusCount.exactCount_formula`
72	theorem	`PaperXI.CyclicFrobeniusCount.card_coprime_q`
79	def	`PaperXI.CyclicFrobeniusCount.frobeniusPerm`
84	theorem	`PaperXI.CyclicFrobeniusCount.frobeniusPerm_apply`
89	theorem	`PaperXI.CyclicFrobeniusCount.orbitCount_mul_formula`
100	theorem	`PaperXI.CyclicFrobeniusCount.moebius_sum_dvd`
108	theorem	`PaperXI.CyclicFrobeniusCount.orbitCount_formula`

A.125 `PaperXI.CyclicInducedBasis`

Source lines: 83.

SHA-256: `a90ba613fe359b594d103fb19aa93529a4e4cb8584c38b335950bcb1ceaf0eb1`.

```
The Frobenius-indexed basis of an actual induction from a cyclic degree kernel.
```

Direct project imports: `PaperXI.CyclicInducedCoordinates`.

Line	Kind	Name
15	def	`PaperXI.CyclicInduced.cyclicRep`
19	theorem	`PaperXI.CyclicInduced.cyclicRep_degree`
25	theorem	`PaperXI.CyclicInduced.cyclicRep_one`
30	def	`PaperXI.CyclicInduced.cyclicBasis`
34	def	`PaperXI.CyclicInduced.cyclicWeight`
37	theorem	`PaperXI.CyclicInduced.cyclicBasis_apply`
43	theorem	`PaperXI.CyclicInduced.cyclicBasis_action`
51	def	`PaperXI.CyclicInduced.cyclicCharacter`
54	theorem	`PaperXI.CyclicInduced.cyclicCharacter_apply`
64	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_kernel`
71	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_zero`
78	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_phi`

A.126 `PaperXI.CyclicInducedCharacter`

Source lines: 60.

SHA-256: `15707518268a187bd5d4f486516c842c236f5385de68af3fb7ef63eb83fcd4f5`.

```
Exact subgroup character values of the monomial coefficients of actual induction.
```

Direct project imports: `PaperXI.CyclicInducedDegree`.

Line	Kind	Name
13	theorem	`PaperXI.CyclicInduced.degreeWeight_kernel`
19	theorem	`PaperXI.CyclicInduced.degreeCharacter_apply`
28	theorem	`PaperXI.CyclicInduced.degreeCharacter_zero`
36	theorem	`PaperXI.CyclicInduced.degreeWeight_zero`
43	theorem	`PaperXI.CyclicInduced.degreeWeight_restrict`
54	theorem	`PaperXI.CyclicInduced.degreeWeight_restrict_zero`

A.127 `PaperXI.CyclicInducedCoordinates`

Source lines: 179.

SHA-256: `302c38129316e682653aa44bbff88450ec77341b005fbdbce392a31a475f2d90`.

```
Coordinates for genuine induction from the kernel of a finite quotient.
The representative indexed by `q` lies over `q⁻¹`, so the basis action is left translation.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.CyclicInduced.scalar`
22	theorem	`PaperXI.CyclicInduced.scalar_apply`
26	def	`PaperXI.CyclicInduced.residual`
29	theorem	`PaperXI.CyclicInduced.residual_rep`
33	theorem	`PaperXI.CyclicInduced.residual_mul`
40	abbrev	`PaperXI.CyclicInduced.CoindSpace`
41	abbrev	`PaperXI.CyclicInduced.coinduced`
42	abbrev	`PaperXI.CyclicInduced.IndSpace`
43	abbrev	`PaperXI.CyclicInduced.induced`
46	def	`PaperXI.CyclicInduced.extend`
52	theorem	`PaperXI.CyclicInduced.extend_rep`
57	def	`PaperXI.CyclicInduced.coordinates`
70	theorem	`PaperXI.CyclicInduced.coordinates_apply`
74	def	`PaperXI.CyclicInduced.weight`
80	def	`PaperXI.CyclicInduced.coindBasis`
83	theorem	`PaperXI.CyclicInduced.coindBasis_rep`
94	theorem	`PaperXI.CyclicInduced.coindBasis_action`
120	def	`PaperXI.CyclicInduced.indCoind`
125	def	`PaperXI.CyclicInduced.basis`
129	theorem	`PaperXI.CyclicInduced.basis_action`
142	def	`PaperXI.CyclicInduced.conjugate`
150	def	`PaperXI.CyclicInduced.conjugateCharacter`
154	theorem	`PaperXI.CyclicInduced.weight_kernel`
163	theorem	`PaperXI.CyclicInduced.basis_action_kernel`
170	theorem	`PaperXI.CyclicInduced.basis_action_origin`

A.128 `PaperXI.CyclicInducedDegree`

Source lines: 59.

SHA-256: `e60c559b41a83de4e58e4a01b0633216f888976f7748d01b1b05f7aa42aaf085`.

Source-facing degree-subgroup coordinates for tame cyclic induction.

Direct project imports: `PaperXI.CyclicInducedBasis`, `PaperXI.TameDegree`.

Line	Kind	Name
15	theorem	`PaperXI.CyclicInduced.degreeMod_phi`
20	abbrev	`PaperXI.CyclicInduced.degreeInduced`
21	abbrev	`PaperXI.CyclicInduced.DegreeIndSpace`
24	def	`PaperXI.CyclicInduced.degreeBasis`
27	def	`PaperXI.CyclicInduced.degreeWeight`
31	def	`PaperXI.CyclicInduced.degreeCharacter`
35	theorem	`PaperXI.CyclicInduced.degreeBasis_action`
42	theorem	`PaperXI.CyclicInduced.degreeBasis_action_kernel`
48	theorem	`PaperXI.CyclicInduced.degreeBasis_action_zero`
54	theorem	`PaperXI.CyclicInduced.degreeBasis_action_phi`

A.129 `PaperXI.CyclicInducedWrap`

Source lines: 119.

SHA-256: `4dec20ab681bfc9225f0bf157b406ac02ea6272612eb84376cb4b647af558bb9`.

Explicit wrap coefficients in the Frobenius-indexed induced basis.

Direct project imports: `PaperXI.CyclicInducedDegree`.

Line	Kind	Name
14	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_value`
38	def	`PaperXI.CyclicInduced.period`
44	theorem	`PaperXI.CyclicInduced.val_succ_of_lt`
50	theorem	`PaperXI.CyclicInduced.val_succ_of_last`
57	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_of_lt`
69	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_of_last`
85	def	`PaperXI.CyclicInduced.degreePeriod`
89	theorem	`PaperXI.CyclicInduced.degreePeriod_coe`
92	theorem	`PaperXI.CyclicInduced.degreeWeight_phi_of_lt`
97	theorem	`PaperXI.CyclicInduced.degreeWeight_phi_of_last`
102	theorem	`PaperXI.CyclicInduced.degreeBasis_phi_of_lt`
109	theorem	`PaperXI.CyclicInduced.degreeBasis_phi_of_last`

A.130 `PaperXI.CyclicLinearization`

Source lines: 83.

SHA-256: `77c69c57b811f5321306490bb3bb4aef92a7c262d5d875a613ef8d63e5170d0c`.

A genuine projective representation of a finite cyclic group lifts to GL.
The projective representation is an actual homomorphism to Mathlib's PGL,
and equality after projection is the conclusion, not a hypothesis.

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.CyclicLinearization.unit_root`
29	theorem	`PaperXI.CyclicLinearization.rescale_generator`
50	theorem	`PaperXI.CyclicLinearization.lift_of_generator`
65	theorem	`PaperXI.CyclicLinearization.finite_cyclic_linearization`
73	theorem	`PaperXI.CyclicLinearization.finite_cyclic_scalar_equivalence`

A.131 `PaperXI.CyclicNormalization113`

Source lines: 386.

SHA-256: `607587d94a544d2c4e014d81b21e4c9241d852feddfaa826df57f9715d396ce7`.

```
Same-carrier existence and uniqueness of determinant-one cyclic
implementers. The full automorphism and prescribed central character are
retained. No arithmetic realization or identification of the original kappa
is asserted.
```

Direct project imports: `PaperXI.GlobalHeisenbergBridge112`.

Line	Kind	Name
18	def	`PaperXI.CyclicNormalization113.scale`
21	theorem	`PaperXI.CyclicNormalization113.scale_apply`
24	theorem	`PaperXI.CyclicNormalization113.scale_pow_apply`
33	theorem	`PaperXI.CyclicNormalization113.scalar_correction_exists`
48	theorem	`PaperXI.CyclicNormalization113.det_scale`
58	theorem	`PaperXI.CyclicNormalization113.normalize_honest`
78	theorem	`PaperXI.CyclicNormalization113.implements_pow`
90	theorem	`PaperXI.CyclicNormalization113.scale_implements`
101	theorem	`PaperXI.CyclicNormalization113.implementer_power_scalar`
111	theorem	`PaperXI.CyclicNormalization113.unit_nth_root`
124	theorem	`PaperXI.CyclicNormalization113.normalize_order`
140	theorem	`PaperXI.CyclicNormalization113.normalized_generator_exists`
153	theorem	`PaperXI.CyclicNormalization113.normalized_generator_unique`
167	theorem	`PaperXI.CyclicNormalization113.normalized_generator_existsUnique`
189	theorem	`PaperXI.CyclicNormalization113.implementer_of_actual_profile`
219	theorem	`PaperXI.CyclicNormalization113.canonical_generator_of_actual_profile`
239	theorem	`PaperXI.CyclicNormalization113.canonical_generator_of_actual_form`
268	def	`PaperXI.CyclicNormalization113.cyclicAction`
276	theorem	`PaperXI.CyclicNormalization113.cyclicAction_generator`
286	theorem	`PaperXI.CyclicNormalization113.canonical_action_of_implementer`
325	theorem	`PaperXI.CyclicNormalization113.canonical_action_of_actual_profile`
347	theorem	`PaperXI.CyclicNormalization113.canonical_equivariant_bridge_of_actual_profile`

A.132 `PaperXI.CyclicPolynomial`

Source lines: 81.

SHA-256: `0b7bf89b38ea2b0bc63675d4707494ec562ab9f04b8218c17fc092d9e82ea6ab`.

Characteristic polynomials from a proved cyclic basis.

Direct project imports: none.

Line	Kind	Name
13	theorem	PaperXI.CyclicPolynomial.aeval_apply_eq_sum
20	theorem	PaperXI.CyclicPolynomial.no_short_annihilator
31	theorem	PaperXI.CyclicPolynomial.charpoly_of_orbit_basis
45	theorem	PaperXI.CyclicPolynomial.det_of_orbit_basis
59	theorem	PaperXI.CyclicPolynomial.eval_charpoly_of_orbit_basis
67	theorem	PaperXI.CyclicPolynomial.reverse_cycle_polynomial
72	theorem	PaperXI.CyclicPolynomial.local_polynomial_of_orbit_basis

A.133 PaperXI.DVRCartan106

Source lines: 68.

SHA-256: 450b28f31d92b480d5a52f75464a037a0aaf3ebd66ce7753b50f9e474951eeda.

Normalized Cartan decomposition over the fraction field of a DVR.

Direct project imports: PaperXI.FractionMatrix106, PaperXI.CuspidalCartan105.

Line	Kind	Name
15	def	PaperXI.DVRCartan106.unitsDiagonal
21	theorem	PaperXI.DVRCartan106.normalize_diagonal
55	theorem	PaperXI.DVRCartan106.hasCartan

A.134 PaperXI.DegreeIndex110

Source lines: 432.

SHA-256: 0f87f63a5f0f5c28adbb1a04732fbe110b63a5f45cdde03e98830271c7c6a6c5.

The degree-square identity from the original irreducible character profile,
and the positive-index cancellation that supplies the intersection Hom line.
Arithmetic trace support, character covariance, and the oriented lattice-index
identity remain explicit hypotheses. Schur is used only on the entire action.

Direct project imports: PaperXI.FiniteProfileBridge109, PaperXI.CompactIntersection108, PaperXI.ActualIntersection108, PaperXI.PairIndex110.

Line	Kind	Name
20	theorem	PaperXI.DegreeIndex110.degree_square_of_finite_model
44	theorem	PaperXI.DegreeIndex110.degree_square_of_finite_kernel_quotient
58	theorem	PaperXI.DegreeIndex110.degree_square
71	theorem	PaperXI.DegreeIndex110.intersection_index_of_oriented_identity
85	theorem	PaperXI.DegreeIndex110.dimension_one_iff_original_index
120	theorem	PaperXI.DegreeIndex110.intersection_dimension_one_of_index
154	theorem	PaperXI.DegreeIndex110.intersection_dimension_one
186	theorem	PaperXI.DegreeIndex110.intersection_index_pos

Line	Kind	Name
199	theorem	`PaperXI.DegreeIndex110.intersection_index_ambient`
205	theorem	`PaperXI.DegreeIndex110.scalarIntersection_index_ambient`
214	theorem	`PaperXI.DegreeIndex110.oriented_identity_of_ambient`
236	def	`PaperXI.DegreeIndex110.actualScalarSubgroup`
240	def	`PaperXI.DegreeIndex110.actualScalarCharacter`
251	theorem	`PaperXI.DegreeIndex110.actual_scalar`
264	theorem	`PaperXI.DegreeIndex110.actualScalarSubgroup_image`
275	theorem	`PaperXI.DegreeIndex110.actual_embedding_range_open`
295	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension_one`
343	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension`
390	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension_ambient`

A.135 `PaperXI.DerivedCentralCharacter69`

Source lines: 41.

SHA-256: `861257cc1f0b38c19e73ec7fec5cb12d5ee9e3affca53ab38d5287c502d56d0a`.

```
Construct the central character of the given irreducible finite-dimensional representation.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.DerivedCentralCharacter69.centralAction`
25	def	`PaperXI.DerivedCentralCharacter69.scalarEquiv`
30	def	`PaperXI.DerivedCentralCharacter69.character`
34	theorem	`PaperXI.DerivedCentralCharacter69.character_apply`

A.136 `PaperXI.DiscrepancyCharacter`

Source lines: 50.

SHA-256: `cf201fbd9c03f12816213b45d8e418ea5649df147dd993af122f826106f798af`.

```
Discrepancy of honest stabilizer actions is a character, derived from Schur.
```

Direct project imports: `PaperXI.SchurScalar`.

Line	Kind	Name
13	theorem	`PaperXI.DiscrepancyCharacter.exists_character`

A.137 `PaperXI.DualCharacterSums`

Source lines: 55.

SHA-256: `cd4cf6b9b43305d2e09f09566db4519b5821134b6fac03ac8e92c83fad7c0532`.

```
Literal finite character orthogonality on a vector space and its full dual.
```

Direct project imports: `PaperXI.SchrodingerTrace`, `PaperXI.SchrodingerSeparation`, `PaperXI.HeisenbergDimension`.

Line	Kind	Name
14	theorem	`PaperXI.DualCharacters.sum_over_points`
22	def	`PaperXI.DualCharacters.evaluationCharacter`
28	theorem	`PaperXI.DualCharacters.evaluationCharacter_apply`
31	theorem	`PaperXI.DualCharacters.evaluationCharacter_ne_one`
40	theorem	`PaperXI.DualCharacters.sum_over_dual`
50	theorem	`PaperXI.DualCharacters.card_dual`

A.138 `PaperXI.DualityLinear`

Source lines: 94.

SHA-256: `0f856e352f57db7831276fa70666d08465b1a79f9ac97e0b4a31265892c47e12`.

```
Actual dual representations, invariant spaces and tensor/Hom comparison.
No fixed-space dimension, invariant-space equivalence or dual irreducibility
is supplied as a hypothesis.
```

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
17	def	`PaperXI.DualityLinear.invariantsEquiv`
35	def	`PaperXI.DualityLinear.bidualEquiv`
43	theorem	`PaperXI.DualityLinear.coannihilator_stable`
54	theorem	`PaperXI.DualityLinear.dual_irreducible`
71	def	`PaperXI.DualityLinear.tensorHomEquiv`
82	def	`PaperXI.DualityLinear.fixedTensorHomEquiv`
88	theorem	`PaperXI.DualityLinear.hom_finrank_eq_tensor_invariants`

A.139 `PaperXI.DyadicAbstract`

Source lines: 159.

SHA-256: `1e70f115444ed1310c3bdae48b5dd98530fa9d8f2eae45684430b7cec5e836d0`.

```
Intrinsic Arf invariants and dyadic formulas under only the abstract
finite-dimensional alternating nondegenerate hypotheses.
```

Direct project imports: `PaperXI.SymplecticCoordinates`, `PaperXI.DyadicBasisConsequences`, `PaperXI.DyadicBasisCounts`.

Line	Kind	Name
15	def	`PaperXI.DyadicAbstract.coordinates`
20	theorem	`PaperXI.DyadicAbstract.coordinates_form`
25	def	`PaperXI.DyadicAbstract.arf`
32	theorem	`PaperXI.DyadicAbstract.arf_eq_basis`
43	def	`PaperXI.DyadicAbstract.gaussSum`
50	theorem	`PaperXI.DyadicAbstract.gaussSum_eq_sum`

Line	Kind	Name
58	theorem	`PaperXI.DyadicAbstract.gauss_sum`
68	def	`PaperXI.DyadicAbstract.normalizedGaussSign`
73	theorem	`PaperXI.DyadicAbstract.normalizedGaussSign_eq_sign`
83	theorem	`PaperXI.DyadicAbstract.arf_translation`
92	theorem	`PaperXI.DyadicAbstract.opposite_refinements`
113	theorem	`PaperXI.DyadicAbstract.even_count`
123	theorem	`PaperXI.DyadicAbstract.odd_count`
133	theorem	`PaperXI.DyadicAbstract.gauss_sum_of_finrank`
142	theorem	`PaperXI.DyadicAbstract.even_count_of_finrank`
151	theorem	`PaperXI.DyadicAbstract.odd_count_of_finrank`

A.140 `PaperXI.DyadicAbstractOrthogonal`

Source lines: 141.

SHA-256: `84c944337500a19941c2a2a1b839131b3c7ffd8b54f89c118ebb29f3416b811f`.

```
Intrinsic Arf and Gauss-sign laws for actual orthogonal sums.
Symplectic coordinates are constructed from alternation and nondegeneracy.
```

Direct project imports: `PaperXI.DyadicAbstract`, `PaperXI.DyadicIsometry`, `PaperXI.DyadicOrthogonal`.

Line	Kind	Name
17	theorem	`PaperXI.DyadicAbstractOrthogonal.form_alternating`
21	theorem	`PaperXI.DyadicAbstractOrthogonal.form_nondegenerate`
38	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_transport`
52	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_add`
71	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_sign_product`
84	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_of_decomposition`
107	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_sign_of_decomposition`
116	theorem	`PaperXI.DyadicAbstractOrthogonal.normalized_sign_product`
129	theorem	`PaperXI.DyadicAbstractOrthogonal.normalized_sign_of_decomposition`

A.141 `PaperXI.DyadicAffine`

Source lines: 81.

SHA-256: `06fd5b5a57b17060abd22a11fb29592fcd53b06aa09388b92314fe2fd23692e3`.

```
Paper XI, Proposition 13.2: the affine space of refinements of a fixed
bilinear form over the field with two elements. No nondegeneracy is required.
```

Direct project imports: none.

Line	Kind	Name
13	structure	`PaperXI.DyadicAffine.Refinement`
19	theorem	`PaperXI.DyadicAffine.Refinement.ext`
25	theorem	`PaperXI.DyadicAffine.Refinement.value_zero`
31	theorem	`PaperXI.DyadicAffine.difference_add`
38	def	`PaperXI.DyadicAffine.linearDifference`

Line	Kind	Name
43	theorem	`PaperXI.DyadicAffine.linearDifference_apply`
47	def	`PaperXI.DyadicAffine.addLinear`
53	theorem	`PaperXI.DyadicAffine.addLinear_apply`
56	theorem	`PaperXI.DyadicAffine.addLinear_linearDifference`
61	theorem	`PaperXI.DyadicAffine.linearDifference_addLinear`
67	def	`PaperXI.DyadicAffine.refinementEquiv`
74	theorem	`PaperXI.DyadicAffine.exists_unique_linear_difference`

A.142 `PaperXI.DyadicBasis`

Source lines: 83.

SHA-256: `18c632203cbd0c023aa7aaf6fcc10af54acb0271d09db3fa9a38d74308c5546f`.

```
Transfer to arbitrary vector spaces with supplied symplectic coordinates.
The existence of those coordinates is explicitly separate.
```

Direct project imports: `PaperXI.DyadicCoordinates`.

Line	Kind	Name
14	def	`PaperXI.DyadicBasis.toCoordinates`
22	def	`PaperXI.DyadicBasis.fromCoordinates`
30	def	`PaperXI.DyadicBasis.refinementEquiv`
42	theorem	`PaperXI.DyadicBasis.toCoordinates_apply`
47	theorem	`PaperXI.DyadicBasis.fromCoordinates_apply`
53	def	`PaperXI.DyadicBasis.basisArf`
56	theorem	`PaperXI.DyadicBasis.basisArf_eq_coordinate_arf`
62	theorem	`PaperXI.DyadicBasis.gauss_sum`
74	theorem	`PaperXI.DyadicBasis.basisArf_independent`

A.143 `PaperXI.DyadicBasisConsequences`

Source lines: 68.

SHA-256: `d45fa0f60b8166b49c7962a6afc195f6e7cb92dccabb9ee3103ff488062a4031`.

```
Translation and opposite Arf phases on arbitrary spaces with supplied
symplectic coordinates.
```

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicRefinement`.

Line	Kind	Name
16	theorem	`PaperXI.DyadicBasisConsequences.toCoordinates_addLinear`
25	theorem	`PaperXI.DyadicBasisConsequences.arf_translation`
42	theorem	`PaperXI.DyadicBasisConsequences.basisArf_fromCoordinates`
48	theorem	`PaperXI.DyadicBasisConsequences.gauss_fromCoordinates`
57	theorem	`PaperXI.DyadicBasisConsequences.opposite_refinements`

A.144 PaperXI.DyadicBasisCounts

Source lines: 35.

SHA-256: `2b9d123b889d163bf29be6dfbab3e510195d98591db8098fb4d773a4a05b6093`.

```
Counts of every actual refinement on an arbitrary space, conditional on
supplied symplectic coordinates.
```

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicRefinementCounts`.

Line	Kind	Name
14	def	`PaperXI.DyadicBasisCounts.arfFiberEquiv`
21	theorem	`PaperXI.DyadicBasisCounts.even_count`
28	theorem	`PaperXI.DyadicBasisCounts.odd_count`

A.145 PaperXI.DyadicCoordinates

Source lines: 162.

SHA-256: `fc1a2a3a6569e264cfddcb51fd16e91c22ff91ed4a741634239798f18cd83004`.

```
Every quadratic refinement in symplectic coordinates has unique linear
coefficients. This identifies all refinements, not only a chosen family.
```

Direct project imports: `PaperXI.DyadicAffine`, `PaperXI.DyadicGauss`.

Line	Kind	Name
15	def	`PaperXI.DyadicCoordinates.symplectic`
56	theorem	`PaperXI.DyadicCoordinates.symplectic_apply`
59	theorem	`PaperXI.DyadicCoordinates.quadratic_polar`
67	def	`PaperXI.DyadicCoordinates.coordinateRefinement`
71	def	`PaperXI.DyadicCoordinates.firstAxis`
72	def	`PaperXI.DyadicCoordinates.secondAxis`
75	theorem	`PaperXI.DyadicCoordinates.single_axes`
79	def	`PaperXI.DyadicCoordinates.linearCoefficients`
82	theorem	`PaperXI.DyadicCoordinates.linear_coordinate_formula`
93	theorem	`PaperXI.DyadicCoordinates.quadratic_firstAxis`
102	theorem	`PaperXI.DyadicCoordinates.quadratic_secondAxis`
111	def	`PaperXI.DyadicCoordinates.refinementCoefficients`
114	theorem	`PaperXI.DyadicCoordinates.refinement_coordinate_formula`
125	theorem	`PaperXI.DyadicCoordinates.refinementCoefficients_coordinateRefinement`
130	def	`PaperXI.DyadicCoordinates.coordinateEquiv`
138	theorem	`PaperXI.DyadicCoordinates.refinementCoefficients_apply`
145	theorem	`PaperXI.DyadicCoordinates.arf_basis_formula`
149	theorem	`PaperXI.DyadicCoordinates.refinement_arf_basis_formula`
155	theorem	`PaperXI.DyadicCoordinates.refinement_gauss_sum`

A.146 PaperXI.DyadicCounts

Source lines: 135.

SHA-256: `b1155d1f3103b197ed16cea0c4c297e29da3f6e917e6006f486e47727af34b5d`.

Exact counts for binary quadratic coefficient spaces.

Direct project imports: `PaperXI.DyadicGauss`.

Line	Kind	Name
13	theorem	`PaperXI.DyadicCounts.signed_sum_eq_card_difference`
19	theorem	`PaperXI.DyadicCounts.card_partition`
27	def	`PaperXI.DyadicCounts.evenCount`
30	def	`PaperXI.DyadicCounts.oddCount`
33	theorem	`PaperXI.DyadicCounts.f2_ne_zero_iff_eq_one`
47	theorem	`PaperXI.DyadicCounts.parameter_card`
50	theorem	`PaperXI.DyadicCounts.even_add_odd`
54	theorem	`PaperXI.DyadicCounts.even_sub_odd`
63	theorem	`PaperXI.DyadicCounts.twice_even`
71	theorem	`PaperXI.DyadicCounts.twice_odd`
78	theorem	`PaperXI.DyadicCounts.even_factored`
90	theorem	`PaperXI.DyadicCounts.odd_factored`
102	theorem	`PaperXI.DyadicCounts.counts_rank_zero`
110	theorem	`PaperXI.DyadicCounts.even_factored_rational`
123	theorem	`PaperXI.DyadicCounts.odd_factored_rational`

A.147 `PaperXI.DyadicDimension`

Source lines: 49.

SHA-256: `295c12f1fe30abb310b8e54856cc51fb1f11bee0bd9f7d4258386bd30271944a`.

Dimensions and Arf comparison for arbitrary finite symplectic index types.
All coordinate maps in this module are explicit inputs.

Direct project imports: `PaperXI.DyadicBasis`.

Line	Kind	Name
15	theorem	`PaperXI.DyadicDimension.finrank_eq_twice_card`
20	theorem	`PaperXI.DyadicDimension.card_eq_finrank_div_two`
25	theorem	`PaperXI.DyadicDimension.card_pos`
32	theorem	`PaperXI.DyadicDimension.coordinate_card_eq`
38	theorem	`PaperXI.DyadicDimension.basisArf_independent`

A.148 `PaperXI.DyadicExtensionClass`

Source lines: 93.

SHA-256: `af6825148cbf10fa7e036416c6b158a32f9be04ed3a83f3767394273839b2656`.

A prescribed square profile determines the bilinear F2-extension up to an
actual center- and quotient-preserving isomorphism. We construct the coboundary
with the upper-triangular part in an ordinary basis.

Direct project imports: `PaperXI.DyadicHeisenbergLift`.

Line	Kind	Name
14	def	PaperXI.DyadicExtensionClass.upper
18	theorem	PaperXI.DyadicExtensionClass.upper_basis
24	theorem	PaperXI.DyadicExtensionClass.upper_symmetrization
45	theorem	PaperXI.DyadicExtensionClass.exists_coboundary
60	theorem	PaperXI.DyadicExtensionClass.exists_equiv_of_diagonal
74	theorem	PaperXI.DyadicExtensionClass.exists_equiv_iff_diagonal
85	theorem	PaperXI.DyadicExtensionClass.refinement_model_independent

A.149 PaperXI.DyadicGauss

Source lines: 60.

SHA-256: 634fe67a476f515f627455fa43a1ec870a4ccc66b87867cc5d40ee1227c91dea.

```
Dyadic quadratic refinements in explicit symplectic coordinates.
```

Direct project imports: none.

Line	Kind	Name
10	abbrev	PaperXI.Dyadic.F2
12	def	PaperXI.Dyadic.sign
14	theorem	PaperXI.Dyadic.sign_zero
15	theorem	PaperXI.Dyadic.sign_one
17	theorem	PaperXI.Dyadic.sign_add
20	theorem	PaperXI.Dyadic.sign_injective
30	theorem	PaperXI.Dyadic.sign_sum
37	def	PaperXI.Dyadic.plane
39	theorem	PaperXI.Dyadic.plane_gauss
47	def	PaperXI.Dyadic.quadratic
49	def	PaperXI.Dyadic.arf
51	theorem	PaperXI.Dyadic.gauss_sum

A.150 PaperXI.DyadicHeisenbergLift

Source lines: 155.

SHA-256: 916986a499526e1343b2b63906fe133ac8db3cb95e6cbbbe31d6ada8e17229b1.

```
A given quadratic refinement on an intrinsic finite F2-space supplies an
actual bilinear factor set, hence a finite Heisenberg extension. Coordinates are
constructed from the form; its square profile is the given refinement, not erased.
```

Direct project imports: PaperXI.HeisenbergCochain, PaperXI.SymplecticCoordinates.

Line	Kind	Name
15	def	PaperXI.DyadicHeisenbergLift.coordinateFactor
65	theorem	PaperXI.DyadicHeisenbergLift.coordinateFactor_apply
69	theorem	PaperXI.DyadicHeisenbergLift.coordinateFactor_commutator
78	theorem	PaperXI.DyadicHeisenbergLift.coordinateFactor_diagonal
91	def	PaperXI.DyadicHeisenbergLift.coordinateRefinementOf

Line	Kind	Name
100	def	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates`
106	theorem	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates_commutator`
113	theorem	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates_diagonal`
125	theorem	`PaperXI.DyadicHeisenbergLift.exists_factor`
135	def	`PaperXI.DyadicHeisenbergLift.factor`
139	theorem	`PaperXI.DyadicHeisenbergLift.factor_commutator`
144	theorem	`PaperXI.DyadicHeisenbergLift.factor_diagonal`
150	theorem	`PaperXI.DyadicHeisenbergLift.square_profile`

A.151 `PaperXI.DyadicIsometry`

Source lines: 52.

SHA-256: `48d9678ba58d442aa451540502f01d5e1fb8a3257e8bdf4d64fef974305c651f`.

Quadratic refinements and their Arf values transport along actual isometries.

Direct project imports: `PaperXI.DyadicDimension`.

Line	Kind	Name
14	def	`PaperXI.DyadicIsometry.transportRefinement`
21	theorem	`PaperXI.DyadicIsometry.transportRefinement_apply`
25	def	`PaperXI.DyadicIsometry.transportRefinementEquiv`
33	theorem	`PaperXI.DyadicIsometry.gauss_sum_transport`
41	theorem	`PaperXI.DyadicIsometry.basisArf_transport`

A.152 `PaperXI.DyadicOrthogonal`

Source lines: 76.

SHA-256: `3ec83d3eeac7e69ef4986a573ba818eb9cfe33282530bb3e8f85d7ed4323d671`.

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicTranslation`.

Line	Kind	Name
14	def	`PaperXI.DyadicOrthogonal.form`
18	theorem	`PaperXI.DyadicOrthogonal.form_apply`
20	def	`PaperXI.DyadicOrthogonal.refinement`
28	theorem	`PaperXI.DyadicOrthogonal.gauss_product`
35	def	`PaperXI.DyadicOrthogonal.coordinates`
40	theorem	`PaperXI.DyadicOrthogonal.coordinates_form`
48	theorem	`PaperXI.DyadicOrthogonal.arf_add`
67	theorem	`PaperXI.DyadicOrthogonal.normalized_sign_product`

A.153 `PaperXI.DyadicRefinement`

Source lines: 48.

SHA-256: `4056c45a4a85c4fbaf44d651ce11f83f50656138b52fb6f18d95e9b8eafa01e5`.

```
Translation and opposite phases for all refinements of the actual
coordinate symplectic form, with the Arf value evaluated on basis vectors.
```

Direct project imports: `PaperXI.DyadicCoordinates`, `PaperXI.DyadicTranslation`.

Line	Kind	Name
13	def	`PaperXI.DyadicRefinement.basisArf`
16	theorem	`PaperXI.DyadicRefinement.basisArf_coordinateRefinement`
20	theorem	`PaperXI.DyadicRefinement.translation_refinement`
29	theorem	`PaperXI.DyadicRefinement.arf_translation`
36	theorem	`PaperXI.DyadicRefinement.opposite_refinements`
45	theorem	`PaperXI.DyadicRefinement.sign_eq_neg_one_pow`

A.154 `PaperXI.DyadicRefinementCounts`

Source lines: 55.

SHA-256: `5d9cdc46b7bde9d4f7ceed89b2a7a68ec38f36fee3142300f40b6a8d5301838b`.

```
Cardinalities of the complete spaces of quadratic refinements with each
Arf invariant, including dimension zero.
```

Direct project imports: `PaperXI.DyadicCounts`, `PaperXI.DyadicCoordinates`.

Line	Kind	Name
13	instance	`PaperXI.DyadicRefinementCounts.refinementFintype`
16	def	`PaperXI.DyadicRefinementCounts.basisArf`
19	theorem	`PaperXI.DyadicRefinementCounts.basisArf_coordinate`
24	def	`PaperXI.DyadicRefinementCounts.arfFiberEquiv`
29	theorem	`PaperXI.DyadicRefinementCounts.even_fiber_card`
35	theorem	`PaperXI.DyadicRefinementCounts.odd_fiber_card`
42	theorem	`PaperXI.DyadicRefinementCounts.even_refinement_count`
49	theorem	`PaperXI.DyadicRefinementCounts.odd_refinement_count`

A.155 `PaperXI.DyadicTorsor`

Source lines: 70.

SHA-256: `da8d7de85105943a07b9492568783dfc1df76d3789bb3444d344259b0ed8bb6b`.

```
The canonical affine-space structure on a nonempty family of refinements.
The action and subtraction are the actual addition and difference of functions.
```

Direct project imports: `PaperXI.DyadicAffine`.

Line	Kind	Name
14	theorem	`PaperXI.DyadicTorsor.addLinear_zero`
18	theorem	`PaperXI.DyadicTorsor.addLinear_add`
24	theorem	`PaperXI.DyadicTorsor.addLinear_injective`
30	theorem	`PaperXI.DyadicTorsor.addLinear_free`
39	theorem	`PaperXI.DyadicTorsor.addLinear_transitive`

Line	Kind	Name
44	instance	PaperXI.DyadicTorsor.instAddAction
49	instance	PaperXI.DyadicTorsor.instVSub
54	instance	PaperXI.DyadicTorsor.instAddTorsor
61	abbrev	PaperXI.DyadicTorsor.canonicalTorsor
64	theorem	PaperXI.DyadicTorsor.vadd_apply
67	theorem	PaperXI.DyadicTorsor.vsub_apply

A.156 PaperXI.DyadicTranslation

Source lines: 89.

SHA-256: 098a6b0ee464537aaea2cef020e535b28ba9572b8079134076fdd5ee6fb54969.

Direct project imports: PaperXI.DyadicGauss.

Line	Kind	Name
9	def	PaperXI.Dyadic.pairing
12	def	PaperXI.Dyadic.translate
15	theorem	PaperXI.Dyadic.quadratic_translate
23	theorem	PaperXI.Dyadic.arf_translate
31	theorem	PaperXI.Dyadic.gauss_translate
37	theorem	PaperXI.Dyadic.arf_orthogonal
41	theorem	PaperXI.Dyadic.gauss_orthogonal
57	theorem	PaperXI.Dyadic.sign_orthogonal
61	theorem	PaperXI.Dyadic.arf_zero
64	def	PaperXI.Dyadic.anisotropic
66	theorem	PaperXI.Dyadic.arf_anisotropic
76	theorem	PaperXI.Dyadic.opposite_gauss
82	theorem	PaperXI.Dyadic.both_arf

A.157 PaperXI.ElementaryEnvelope

Source lines: 139.

SHA-256: 76c36ba11aabeef9351becc4f99f59db800649c73d1dc0f0ff57d59bc7e5d6c4.

```
Paper XI, 8.4 and the elementary-abelian portion of 9.4–9.5.
The complex cocycle is arbitrary. Only its commutator is forced to have
values in the group of pth roots of unity.
```

Direct project imports: PaperXI.CocycleQuotient.

Line	Kind	Name
18	theorem	PaperXI.ElementaryEnvelope.commutator_pow
25	abbrev	PaperXI.ElementaryEnvelope.groupModule
28	theorem	PaperXI.ElementaryEnvelope.finiteDimensional_group
36	def	PaperXI.ElementaryEnvelope.Space
39	theorem	PaperXI.ElementaryEnvelope.quotient_pow
45	instance	PaperXI.ElementaryEnvelope.spaceModule
50	instance	PaperXI.ElementaryEnvelope.finiteSpace

Line	Kind	Name
53	instance	`PaperXI.ElementaryEnvelope.finiteDimensionalSpace`
58	def	`PaperXI.ElementaryEnvelope.rootCharacter`
69	def	`PaperXI.ElementaryEnvelope.rootEquiv`
76	theorem	`PaperXI.ElementaryEnvelope.rootEquiv_coe`
80	def	`PaperXI.ElementaryEnvelope.rootLog`
83	theorem	`PaperXI.ElementaryEnvelope.rootLog_character`
86	theorem	`PaperXI.ElementaryEnvelope.character_rootLog`
90	theorem	`PaperXI.ElementaryEnvelope.quotientPairing_pow`
97	def	`PaperXI.ElementaryEnvelope.rootPairing`
115	theorem	`PaperXI.ElementaryEnvelope.rootPairing_coe`
119	def	`PaperXI.ElementaryEnvelope.rootCommutator`
133	theorem	`PaperXI.ElementaryEnvelope.rootCommutator_coe`
136	theorem	`PaperXI.ElementaryEnvelope.rootCommutator_self`

A.158 `PaperXI.ElementaryEnvelopeForm`

Source lines: 137.

SHA-256: `8bfa652c46358cbe6ee15446b6d435521aeac0f75cc96a605a5fb41b3bf5bc6f`.

```
The actual alternating prime-field form on the commutator radical quotient.
The logarithm equivalence is explicit: the standard character supplies one,
and all alternative choices have a proved scalar comparison.
```

Direct project imports: `PaperXI.ElementaryEnvelope`.

Line	Kind	Name
15	def	`PaperXI.ElementaryEnvelope.logarithmicPairing`
37	def	`PaperXI.ElementaryEnvelope.formWith`
42	theorem	`PaperXI.ElementaryEnvelope.formWith_apply`
47	theorem	`PaperXI.ElementaryEnvelope.formWith_character`
54	theorem	`PaperXI.ElementaryEnvelope.formWith_smul_left`
59	theorem	`PaperXI.ElementaryEnvelope.formWith_smul_right`
64	theorem	`PaperXI.ElementaryEnvelope.formWith_isAlt`
73	theorem	`PaperXI.ElementaryEnvelope.formWith_eq_zero_iff`
80	theorem	`PaperXI.ElementaryEnvelope.formWith_nondegenerate`
93	def	`PaperXI.ElementaryEnvelope.form`
95	theorem	`PaperXI.ElementaryEnvelope.form_isAlt`
97	theorem	`PaperXI.ElementaryEnvelope.form_nondegenerate`
101	theorem	`PaperXI.ElementaryEnvelope.form_stdAddChar`
109	theorem	`PaperXI.ElementaryEnvelope.choice_scalar_ne_zero`
115	theorem	`PaperXI.ElementaryEnvelope.choice_map_eq_mul`
124	theorem	`PaperXI.ElementaryEnvelope.formWith_choice`
131	theorem	`PaperXI.ElementaryEnvelope.exists_choice_scalar`

A.159 `PaperXI.ElementaryEnvelopeTransport`

Source lines: 148.

SHA-256: `56b022844ede8e64c96a72cadf7393135d60c5c9601cd457d49f95322e3fcf18`.

```
Functoriality of the actual elementary commutator envelope.
All maps here descend the given group maps, and preserve the form for one
common chosen root coordinate.
```

Direct project imports: `PaperXI.ElementaryEnvelopeForm`.

Line	Kind	Name
15	theorem	`PaperXI.ElementaryEnvelope.radical_map`
34	def	`PaperXI.ElementaryEnvelope.quotientEquiv`
39	theorem	`PaperXI.ElementaryEnvelope.quotientEquiv_mk`
43	theorem	`PaperXI.ElementaryEnvelope.quotientEquiv_pairing`
56	def	`PaperXI.ElementaryEnvelope.linearEquiv`
62	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_mk`
68	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_formWith`
79	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_form`
85	theorem	`PaperXI.ElementaryEnvelope.space_induction_on`
89	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_refl`
98	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_trans`
113	def	`PaperXI.ElementaryEnvelope.action`
135	theorem	`PaperXI.ElementaryEnvelope.action_mk`
142	theorem	`PaperXI.ElementaryEnvelope.action_form`

A.160 `PaperXI.EquivariantRepresentation113`

Source lines: 117.

SHA-256: `f1cf8ff3343f2647c0994270f6de5bc3478925a25406ea3a221861807b08b6c9`.

```
The corrected scalar cochain constructs an actual representation that
retains the original theta and intertwines the original packet generator.
Actual groups and their full automorphisms remain explicit inputs.
```

Direct project imports: `PaperXI.ActualScalarLift113`.

Line	Kind	Name
38	theorem	`PaperXI.EquivariantRepresentation113.scalarTransport_covariant`
67	theorem	`PaperXI.EquivariantRepresentation113.exists_equivariant_actual_action`

A.161 `PaperXI.EquivariantSplitting68`

Source lines: 35.

SHA-256: `561e6ca7bb701f95ce183d25a1b39b3dbf2a5d389924851f78373fa3f52e4460`.

```
A compact-mod-scalar equivariant splitting, constructed by a finite orbit sum.
```

Direct project imports: `PaperXI.SmoothHomOrbit68`, `PaperXI.FiniteOrbitAverage68`.

Line	Kind	Name
20	theorem	`PaperXI.EquivariantSplitting68.exists_equivariant_section`

A.162 PaperXI.FiniteAverage

Source lines: 69.

SHA-256: e960b8793c2a7e76fef815c456c4e61db736706716741679c3d423053722c977.

Literal finite group averages and their invariant and orthogonal projection properties.

Direct project imports: none.

Line	Kind	Name
11	def	PaperXI.FiniteAverage.average
14	theorem	PaperXI.FiniteAverage.average_apply
18	theorem	PaperXI.FiniteAverage.card_ne_zero
21	theorem	PaperXI.FiniteAverage.invariant
27	theorem	PaperXI.FiniteAverage.fixes
34	theorem	PaperXI.FiniteAverage.idempotent
39	theorem	PaperXI.FiniteAverage.range_eq
51	theorem	PaperXI.FiniteAverage.self_adjoint
62	theorem	PaperXI.FiniteAverage.residual_orthogonal

A.163 PaperXI.FiniteBreakAssembly

Source lines: 175.

SHA-256: ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c.

Construct the initial tensor bridge and close its honest stabilizer action.
The central extensions (including dyadic square data), their actual lifts,
central characters and actions are inputs. No initial intertwiner, discrepancy
character, finite-dimensional certificate, or final comparison is an input.

Direct project imports: PaperXI.BreakBridgeAlong, PaperXI.CentralExtensionComparison, PaperXI.TraceDiscrepancy.

Line	Kind	Name
22	structure	PaperXI.FiniteBreakAssembly.Data
43	def	PaperXI.FiniteBreakAssembly.factor
51	theorem	PaperXI.FiniteBreakAssembly.factor_intertwines
60	def	PaperXI.FiniteBreakAssembly.tensorBridge
63	def	PaperXI.FiniteBreakAssembly.sourceAction
67	def	PaperXI.FiniteBreakAssembly.targetAction
71	theorem	PaperXI.FiniteBreakAssembly.tensor_intertwines
79	theorem	PaperXI.FiniteBreakAssembly.source_finiteDimensional
84	theorem	PaperXI.FiniteBreakAssembly.source_irreducible
96	def	PaperXI.FiniteBreakAssembly.discrepancy
102	theorem	PaperXI.FiniteBreakAssembly.discrepancy_law
111	def	PaperXI.FiniteBreakAssembly.equivOfPhase
132	theorem	PaperXI.FiniteBreakAssembly.discrepancy_and_closure
151	def	PaperXI.FiniteBreakAssembly.equivOfTraces
169	theorem	PaperXI.FiniteBreakAssembly.equivOfTraces_map

A.164 PaperXI.FiniteFieldAverage

Source lines: 186.

SHA-256: da2bd3e2ade03fa988a3b50efe5b53ef4ed47f3636c86fc5f6841cc8b868f81d.

```
Actual finite-order averages and their fixed/moving decomposition over a field.
The construction uses the invertibility of the order scalar, with no initial
complementary decomposition or finite-dimensionality assumption.
```

Direct project imports: none.

Line	Kind	Name
16	def	PaperXI.FiniteFieldAverage.difference
19	def	PaperXI.FiniteFieldAverage.average
23	def	PaperXI.FiniteFieldAverage.companion
26	theorem	PaperXI.FiniteFieldAverage.difference_apply
28	theorem	PaperXI.FiniteFieldAverage.average_apply
32	theorem	PaperXI.FiniteFieldAverage.companion_apply
37	theorem	PaperXI.FiniteFieldAverage.weighted_identity
50	theorem	PaperXI.FiniteFieldAverage.power_eq_one
58	theorem	PaperXI.FiniteFieldAverage.average_mul_difference
65	theorem	PaperXI.FiniteFieldAverage.difference_mul_average
72	theorem	PaperXI.FiniteFieldAverage.moving_identity
79	theorem	PaperXI.FiniteFieldAverage.power_fixes
86	theorem	PaperXI.FiniteFieldAverage.average_fixes
94	theorem	PaperXI.FiniteFieldAverage.range_average
104	theorem	PaperXI.FiniteFieldAverage.ker_average
117	theorem	PaperXI.FiniteFieldAverage.idempotent
124	theorem	PaperXI.FiniteFieldAverage.isCompl
131	def	PaperXI.FiniteFieldAverage.decomposition
136	theorem	PaperXI.FiniteFieldAverage.decomposition_apply
142	theorem	PaperXI.FiniteFieldAverage.decomposition_symm_fst
154	theorem	PaperXI.FiniteFieldAverage.decomposition_symm_snd
162	theorem	PaperXI.FiniteFieldAverage.average_mul_action
170	theorem	PaperXI.FiniteFieldAverage.action_mul_average
178	theorem	PaperXI.FiniteFieldAverage.average_independent

A.165 PaperXI.FiniteFieldOrbitCount

Source lines: 93.

SHA-256: 624a4a0234ebd63e8062471cb1439ed52dc73ed82e5d8f6400549c16196dde42.

```
Proposition 3.6 on actual complex-valued multiplicative characters.
The cyclicity and cardinality of the character group are derived, not assumed.
```

Direct project imports: PaperXI.CyclicFrobeniusCount.

Line	Kind	Name
13	instance	PaperXI.FiniteFieldOrbitCount.complexEnoughRoots
19	theorem	PaperXI.FiniteFieldOrbitCount.characterGroup_isCyclic
25	theorem	PaperXI.FiniteFieldOrbitCount.character_card

Line	Kind	Name
28	def	PaperXI.FiniteFieldOrbitCount.characterFrobenius
33	theorem	PaperXI.FiniteFieldOrbitCount.characterFrobenius_apply
41	theorem	PaperXI.FiniteFieldOrbitCount.fixed_character_count
47	theorem	PaperXI.FiniteFieldOrbitCount.exact_character_count
54	theorem	PaperXI.FiniteFieldOrbitCount.regular_character_orbits
62	theorem	PaperXI.FiniteFieldOrbitCount.regular_numerator_divisible
70	def	PaperXI.FiniteFieldOrbitCount.extensionFrobenius
77	theorem	PaperXI.FiniteFieldOrbitCount.extension_regular_orbits
88	theorem	PaperXI.FiniteFieldOrbitCount.extensionFrobenius_apply

A.166 PaperXI.FiniteFreeCompletion

Source lines: 75.

SHA-256: 218a5b9e952143a9e26b36d2bcf93529baca8ae45f986c76903854d7c18aeead.

```
Completeness is derived in finite free coordinates, not assumed of the
constructed root ring.
```

Direct project imports: PaperXI.ResidueExtensionRealization.

Line	Kind	Name
14	theorem	PaperXI.FiniteFreeCompletion.linear_congruence
25	theorem	PaperXI.FiniteFreeCompletion.precomplete
42	theorem	PaperXI.FiniteFreeCompletion.hausdorff
53	theorem	PaperXI.FiniteFreeCompletion.complete
65	theorem	PaperXI.CompleteResidueRealization.complete

A.167 PaperXI.FiniteFunctionAverage

Source lines: 54.

SHA-256: a7d776b604ef43cade5a813a977a7ce80ebd96a2b7906759e3b9fc89c7420d01.

```
Orthogonal averaging for the actual finite-sum Hermitian pairing on functions.
```

Direct project imports: PaperXI.TranslationAverage.

Line	Kind	Name
11	theorem	PaperXI.FiniteFunctionAverage.pairing_average_left
24	theorem	PaperXI.FiniteFunctionAverage.pairing_average_right
36	theorem	PaperXI.FiniteFunctionAverage.self_adjoint
46	theorem	PaperXI.FiniteFunctionAverage.residual_orthogonal

A.168 PaperXI.FiniteIntersectionCriterion108

Source lines: 30.

SHA-256: a428d271392d27755c20bb0b6dac61b71398d4cc6370fe0642c4e5ed41364f51.

```
An exact criterion on a literal finite model. The scalar character profile,
compatibility and index identity must still be proved for the arithmetic data.
```

Direct project imports: `PaperXI.CompactIntersection108`.

Line	Kind	Name
14	theorem	`PaperXI.FiniteIntersectionCriterion108.dimension_one_iff`

A.169 `PaperXI.FiniteJointQuotient108`

Source lines: 47.

SHA-256: `bd7174c5b31be9c6750454626e3e5627651dbea5f5465b49f2f3cbfbedc846df`.

```
A single finite quotient for two smooth actions, constructed from their
actual joint kernel. No faithful quotient or finite image is assumed.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`.

Line	Kind	Name
13	def	`PaperXI.FiniteJointQuotient108.kernel`
16	instance	`PaperXI.FiniteJointQuotient108.kernel_normal`
19	abbrev	`PaperXI.FiniteJointQuotient108.quotient`
21	def	`PaperXI.FiniteJointQuotient108.first`
25	def	`PaperXI.FiniteJointQuotient108.second`
29	theorem	`PaperXI.FiniteJointQuotient108.first_comp`
32	theorem	`PaperXI.FiniteJointQuotient108.second_comp`
40	theorem	`PaperXI.FiniteJointQuotient108.finite`

A.170 `PaperXI.FiniteOrbitAverage68`

Source lines: 101.

SHA-256: `41b8b3e87046ef73167fbebd34c7c4a62a6bf13714ff572356c87c64a17b3de6`.

```
Finite-orbit averaging with preservation of a genuine affine constraint.
```

Direct project imports: none.

Line	Kind	Name
16	abbrev	`PaperXI.FiniteOrbitAverage68.Orbit`
20	def	`PaperXI.FiniteOrbitAverage68.orbitEquiv`
32	def	`PaperXI.FiniteOrbitAverage68.average`
40	theorem	`PaperXI.FiniteOrbitAverage68.average_invariant`
51	theorem	`PaperXI.FiniteOrbitAverage68.average_constraint`
68	def	`PaperXI.FiniteOrbitAverage68.invariantHom`
79	theorem	`PaperXI.FiniteOrbitAverage68.equivariant_section_of_finite_orbit`

A.171 PaperXI.FiniteProductDeterminant

Source lines: 64.

SHA-256: `07340f6176b2b67b41bba32f65bdb41e2805856a67877d7e70b73585b9c7b556`.

```
Determinants of actual finite products with varying coordinate spaces.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.FiniteProductDeterminant.map`
19	theorem	`PaperXI.FiniteProductDeterminant.map_apply`
24	theorem	`PaperXI.FiniteProductDeterminant.determinant`

A.172 PaperXI.FiniteProfileBridge109

Source lines: 195.

SHA-256: `d829f6afedc938f83bdb668850c4a684874773efe807983b4ae769844588cf0a`.

```
Bounded auxiliary results for the next finite-profile bridge. These prove
generic algebraic descent prerequisites, not the actual simple-character
profile or the BK lattice-index equality.
```

Direct project imports: `PaperXI.HeisenbergTrace108`, `PaperXI.CharacterPairing108`, `PaperXI.FiniteJointQuotient108`.

Line	Kind	Name
19	theorem	`PaperXI.FiniteProfileBridge109.kernel_le_of_trace_support`
33	theorem	`PaperXI.FiniteProfileBridge109.scalar_character_eq_one_on_kernel`
48	theorem	`PaperXI.FiniteProfileBridge109.joint_kernel_le`
60	def	`PaperXI.FiniteProfileBridge109.imageMap`
65	theorem	`PaperXI.FiniteProfileBridge109.imageMap_surjective`
73	theorem	`PaperXI.FiniteProfileBridge109.imageMap_kernel_le_character_kernel`
90	def	`PaperXI.FiniteProfileBridge109.imageCharacter`
94	theorem	`PaperXI.FiniteProfileBridge109.imageCharacter_apply`
102	theorem	`PaperXI.FiniteProfileBridge109.image_scalar`
113	theorem	`PaperXI.FiniteProfileBridge109.image_trace_zero`
120	theorem	`PaperXI.FiniteProfileBridge109.image_intersection`
130	theorem	`PaperXI.FiniteProfileBridge109.image_index`
134	theorem	`PaperXI.FiniteProfileBridge109.image_characters_compatible`
158	theorem	`PaperXI.FiniteProfileBridge109.quotient_kernel_le_action_kernel`
168	theorem	`PaperXI.FiniteProfileBridge109.dimension_one_iff_of_original_profiles`

A.173 PaperXI.FiniteRepresentation

Source lines: 43.

SHA-256: `b9ac153884bbf3b17a2701a8b50ee49c6f936899b7e807fa3a1cae5006e44d47`.

```
Finite dimension is derived from the actual finite group action.
The generic input is the cyclicity of a nonzero simple module, not a
semisimple decomposition or an assumed target dimension.
```

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
16	theorem	`PaperXI.FiniteRepresentation.simple_finiteDimensional`
26	theorem	`PaperXI.FiniteRepresentation.finiteDimensional`
35	theorem	`PaperXI.FiniteRepresentation.finrank_pos`

A.174 `PaperXI.FiniteResidueBaseChange`

Source lines: 91.

SHA-256: `559f2e07321f7991ab4790ffc84edb044865ea96de4a2c240148b8fb39aea272`.

```
Genuine finite-residue Frobenius, base-field change, and period detection.
The chosen embeddings are inputs, not a constructed unramified extension of a
local field. Residue Frobenius is never replaced by an abstract cyclic label.
```

Direct project imports: `PaperXI.ResidueDegreeBaseChange`.

Line	Kind	Name
16	def	`PaperXI.FiniteResidueBaseChange.arithmetic`
20	def	`PaperXI.FiniteResidueBaseChange.geometric`
22	theorem	`PaperXI.FiniteResidueBaseChange.arithmetic_apply`
24	theorem	`PaperXI.FiniteResidueBaseChange.arithmetic_pow_apply`
33	theorem	`PaperXI.FiniteResidueBaseChange.geometric_inverse_power`
42	theorem	`PaperXI.FiniteResidueBaseChange.arithmetic_tower`
50	theorem	`PaperXI.FiniteResidueBaseChange.geometric_tower`
56	def	`PaperXI.FiniteResidueBaseChange.pullbackEquiv`
67	def	`PaperXI.FiniteResidueBaseChange.forget`
73	theorem	`PaperXI.FiniteResidueBaseChange.forget_injective`
80	theorem	`PaperXI.FiniteResidueBaseChange.geometric_order`
86	theorem	`PaperXI.FiniteResidueBaseChange.geometric_zpow_eq_one`

A.175 `PaperXI.FiniteResidueSubgroup`

Source lines: 128.

SHA-256: `2da76b0ad80ae46f7b05f39967fee2a9247b7c80f1a9e0ef61b0423a026f8a72`.

```
A residue-field fixing subgroup, not an abstract subgroup assigned index f.
Its pullback for geometric |k'|-Frobenius is identified with the degree-f subgroup
of the |k|-Frobenius pullback. No local-field lifting theorem is asserted.
```

Direct project imports: `PaperXI.FiniteResidueBaseChange`.

Line	Kind	Name
14	theorem	`PaperXI.FiniteResidueSubgroup.arithmetic_map`
19	theorem	`PaperXI.FiniteResidueSubgroup.geometric_map`
26	theorem	`PaperXI.FiniteResidueSubgroup.geometric_pow_map`
38	theorem	`PaperXI.FiniteResidueSubgroup.geometric_zpow_map`
51	theorem	`PaperXI.FiniteResidueSubgroup.fixes_subfield_iff`

Line	Kind	Name
70	def	PaperXI.FiniteResidueSubgroup.fixing
87	theorem	PaperXI.FiniteResidueSubgroup.kernel_le_fixing
94	theorem	PaperXI.FiniteResidueSubgroup.projection_mem_iff
104	theorem	PaperXI.FiniteResidueSubgroup.projection_fixes
111	def	PaperXI.FiniteResidueSubgroup.fixingEquiv
122	def	PaperXI.FiniteResidueSubgroup.unramifiedEquiv

A.176 PaperXI.FiniteTameComparison

Source lines: 113.

SHA-256: 1c4357da32eaade5be0c88670eaa605f47fbc4269253888bbf9f72509d0e428d.

```
Proposition 16.1: an explicit finite comparison package on genuine induced
representations. The local-Weil and standard Artin/epsilon-factor identifications
are intentionally NOT fields of this finite record and are not proved here.
```

Direct project imports: PaperXI.InducedTameProfile, PaperXI.TameGaussNaturality, PaperXI.TameTensorSource.

Line	Kind	Name
17	structure	PaperXI.FiniteTameComparison.Observables
40	def	PaperXI.FiniteTameComparison.observed
53	def	PaperXI.FiniteTameComparison.fromCoordinates
64	theorem	PaperXI.FiniteTameComparison.observations_eq_coordinates
83	def	PaperXI.FiniteTameComparison.interactionEquiv
93	theorem	PaperXI.FiniteTameComparison.finite_comparison_package

A.177 PaperXI.FiniteTensor

Source lines: 69.

SHA-256: 8a628f3f2be9e4cbc308b13abbc71ef858a82cf897523b60554829ed81ac0782.

```
Actual tensor products indexed by an arbitrary finite family, with their
external product actions. The empty family retains its scalar tensor unit.
```

Direct project imports: none.

Line	Kind	Name
17	def	PaperXI.FiniteTensor.representation
25	theorem	PaperXI.FiniteTensor.representation_tprod
32	theorem	PaperXI.FiniteTensor.finrank
40	theorem	PaperXI.FiniteTensor.finrank_empty
46	theorem	PaperXI.FiniteTensor.nontrivial
53	theorem	PaperXI.FiniteTensor.empty_action
60	theorem	PaperXI.FiniteTensor.scalar_map

A.178 PaperXI.FiniteUnramifiedWeil

Source lines: 198.

SHA-256: `73d632d598c02e6eced406cf8bd8f48422a3b79634bfb718dea82cc440d29f40`.

```
# Finite unramified Weil quotient of the constructed local field
```

Direct project imports: `PaperXI.ResidueFrobeniusAttachment`, `PaperXI.ResidueDegreeBaseChange`.

Line	Kind	Name
26	abbrev	`PaperXI.FiniteUnramifiedWeil.UnramifiedWeil`
30	def	`PaperXI.FiniteUnramifiedWeil.degreeEquiv`
44	theorem	`PaperXI.FiniteUnramifiedWeil.degreeEquiv_apply`
50	theorem	`PaperXI.FiniteUnramifiedWeil.projection_eq_geometric_zpow_degree`
57	theorem	`PaperXI.FiniteUnramifiedWeil.degree_bijective`
62	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_eq_bot`
67	def	`PaperXI.FiniteUnramifiedWeil.geometricLift`
71	theorem	`PaperXI.FiniteUnramifiedWeil.degree_geometricLift`
77	theorem	`PaperXI.FiniteUnramifiedWeil.projection_geometricLift`
84	def	`PaperXI.FiniteUnramifiedWeil.arithmeticLift`
88	theorem	`PaperXI.FiniteUnramifiedWeil.degree_arithmeticLift`
95	theorem	`PaperXI.FiniteUnramifiedWeil.projection_arithmeticLift`
104	theorem	`PaperXI.FiniteUnramifiedWeil.eq_geometricLift_zpow_degree`
118	theorem	`PaperXI.FiniteUnramifiedWeil.geometricLift_zpowers_eq_top`
128	instance	`PaperXI.FiniteUnramifiedWeil.unramifiedWeil_isMulCommutative`
136	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_finrank_eq_one`
144	theorem	`PaperXI.FiniteUnramifiedWeil.projection_surjective`
156	theorem	`PaperXI.FiniteUnramifiedWeil.projection_ker_eq_period`
189	def	`PaperXI.FiniteUnramifiedWeil.finiteResidueQuotientEquiv`

A.179 PaperXI.FixedMoving

Source lines: 198.

SHA-256: `c5159f03721e3f1446641781dcf303a90ea160dafb3eed4edac5906238153c67`.

```
Actual fixed and moving subspaces, restricted symplectic forms and
the restricted moving automorphism.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.FixedMoving.fixed`
16	def	`PaperXI.FixedMoving.moving`
18	theorem	`PaperXI.FixedMoving.mem_fixed`
21	theorem	`PaperXI.FixedMoving.mem_moving`
25	theorem	`PaperXI.FixedMoving.orthogonal`
37	theorem	`PaperXI.FixedMoving.orthogonal_reverse`
43	def	`PaperXI.FixedMoving.fixedForm`
46	def	`PaperXI.FixedMoving.movingForm`
48	theorem	`PaperXI.FixedMoving.fixedForm_alt`

Line	Kind	Name
51	theorem	`PaperXI.FixedMoving.movingForm_alt`
55	theorem	`PaperXI.FixedMoving.restrict_nondegenerate`
77	theorem	`PaperXI.FixedMoving.movingForm_nondegenerate_of_isCompl`
84	theorem	`PaperXI.FixedMoving.fixedForm_nondegenerate_of_isCompl`
91	theorem	`PaperXI.FixedMoving.moving_stable`
97	theorem	`PaperXI.FixedMoving.moving_inverse_stable`
105	def	`PaperXI.FixedMoving.movingAction`
113	theorem	`PaperXI.FixedMoving.movingAction_coe`
116	theorem	`PaperXI.FixedMoving.movingAction_symm_coe`
119	theorem	`PaperXI.FixedMoving.movingAction_preserves`
124	theorem	`PaperXI.FixedMoving.movingAction_fixedPointFree_of_isCompl`
133	theorem	`PaperXI.FixedMoving.moving_difference_injective`
145	def	`PaperXI.FixedMoving.movingDifferenceEquiv`
151	theorem	`PaperXI.FixedMoving.movingDifferenceEquiv_apply`
159	def	`PaperXI.FixedMoving.movingEquiv`
179	theorem	`PaperXI.FixedMoving.movingEquiv_coe`
184	theorem	`PaperXI.FixedMoving.movingEquiv_intertwines`
191	theorem	`PaperXI.FixedMoving.movingEquiv_preserves`

A.180 `PaperXI.FixedMovingSource`

Source lines: 111.

SHA-256: `2591738676c9f4830f6e41ce811376e7b6b2c17f1afe99cfbf94cffebdb7aa22`.

```
The source fixed/moving theorem, with the decomposition and restricted
invertibility derived from the actual finite order prime to the characteristic.
```

Direct project imports: `PaperXI.FixedMoving`, `PaperXI.FiniteFieldAverage`, `PaperXI.OddPhase`.

Line	Kind	Name
15	theorem	`PaperXI.FixedMoving.isCompl_of_period`
19	def	`PaperXI.FixedMoving.decomposition`
22	theorem	`PaperXI.FixedMoving.decomposition_apply`
27	theorem	`PaperXI.FixedMoving.decomposition_fixed`
33	theorem	`PaperXI.FixedMoving.decomposition_moving`
38	theorem	`PaperXI.FixedMoving.movingForm_nondegenerate`
44	theorem	`PaperXI.FixedMoving.fixedForm_nondegenerate`
55	theorem	`PaperXI.FixedMoving.source_order_pos`
63	theorem	`PaperXI.FixedMoving.source_order_ne_zero`
68	theorem	`PaperXI.FixedMoving.source_isCompl`
73	def	`PaperXI.FixedMoving.sourceDecomposition`
77	theorem	`PaperXI.FixedMoving.sourceDecomposition_apply`
81	theorem	`PaperXI.FixedMoving.source_moving_fixedPointFree`
86	def	`PaperXI.FixedMoving.sourceMovingDifference`
89	theorem	`PaperXI.FixedMoving.sourceMovingDifference_apply`
95	theorem	`PaperXI.FixedMoving.source`

A.181 PaperXI.FourierCovariance

Source lines: 40.

SHA-256: 83b273abc08207d3b23d39152863370e2345bc7a04f422578b880b39ffb6bb6c.

```
The corrected covariance half of Proposition 3.1 under its printed
relation F T(a) F⁻¹ = T(q*a). Idempotence and completeness are not asserted here.
```

Direct project imports: PaperXI.ProjectorTransport.

Line	Kind	Name
13	def	PaperXI.FourierCovariance.mulPermutation
19	def	PaperXI.FourierCovariance.fourierOperator
25	theorem	PaperXI.FourierCovariance.inverse_exponent_covariance

A.182 PaperXI.FourierPackets

Source lines: 50.

SHA-256: 3521693ff5ede2869607d8308db0f63ce2cb6a736c5d728fd43d9b18c942cade.

```
The ranges of the computed projectors are precisely the declared character
spaces. This identifies the algebraic operators with the paper's tame packets.
```

Direct project imports: PaperXI.FourierProjectors.

Line	Kind	Name
13	def	PaperXI.FourierPackets.packet
20	theorem	PaperXI.FourierPackets.projector_mem_packet
27	theorem	PaperXI.FourierPackets.projector_on_packet
36	theorem	PaperXI.FourierPackets.projector_range
44	theorem	PaperXI.FourierPackets.vector_reconstruction

A.183 PaperXI.FourierProjectors

Source lines: 79.

SHA-256: 02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52.

```
Finite Fourier projectors derived from cyclic representation laws and a
primitive additive character. No idempotence or orthogonality is assumed.
```

Direct project imports: PaperXI.CyclicAction.

Line	Kind	Name
13	def	PaperXI.FourierProjectors.projector
17	theorem	PaperXI.FourierProjectors.projector_formula
21	theorem	PaperXI.FourierProjectors.character_sum
26	theorem	PaperXI.FourierProjectors.action_projector
44	theorem	PaperXI.FourierProjectors.projector_product
58	theorem	PaperXI.FourierProjectors.projector_idempotent

Line	Kind	Name
63	theorem	`PaperXI.FourierProjectors.projector_orthogonal`
68	theorem	`PaperXI.FourierProjectors.projector_complete`

A.184 `PaperXI.FractionMatrix106`

Source lines: 57.

SHA-256: `37125ed47f602558238d6c6b2e9ef6abc4f83592ef58525b03697009a9ac54b6`.

```
Clear denominators and retain both integral basis changes.
```

Direct project imports: `PaperXI.SmithMatrix106`, `PaperXI.SplitRadical105`.

Line	Kind	Name
17	theorem	`PaperXI.FractionMatrix106.exists_integral_multiple`
38	theorem	`PaperXI.FractionMatrix106.exists_integral_diagonal`

A.185 `PaperXI.FreeLatticeCard110`

Source lines: 55.

SHA-256: `07d5f21b13996d68c3d71abd71dab1052fd72ae292beba461aee4f571f6bcd08`.

```
Cardinality of ideal quotients of an actual finite free module.
This supplies the numerical lattice leaf once its corner basis is constructed;
it does not assert that an arbitrary additive lattice has that basis.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.FreeLatticeCard110.ideal_smul_top_pi`
34	def	`PaperXI.FreeLatticeCard110.quotientEquiv`
41	theorem	`PaperXI.FreeLatticeCard110.card_quotient`
46	theorem	`PaperXI.FreeLatticeCard110.card_quotient_prime_pow`

A.186 `PaperXI.FrobeniusOrbit`

Source lines: 42.

SHA-256: `7d1f7d09d4d67afe3a6e5721cf265b76d2ff69a13ef63bb6a753133817f2f7c3`.

```
Frobenius orbits with integer powers, including both transport directions.
```

Direct project imports: `PaperXI.PacketTransport`.

Line	Kind	Name
8	def	`PaperXI.FrobeniusOrbit.Related`
11	theorem	`PaperXI.FrobeniusOrbit.refl`
14	theorem	`PaperXI.FrobeniusOrbit.trans`

Line	Kind	Name
20	theorem	`PaperXI.FrobeniusOrbit.symm`
26	theorem	`PaperXI.FrobeniusOrbit.step`
29	theorem	`PaperXI.FrobeniusOrbit.back`
32	theorem	`PaperXI.FrobeniusOrbit.step_mem`
35	theorem	`PaperXI.FrobeniusOrbit.back_mem`
38	theorem	`PaperXI.FrobeniusOrbit.same_orbit_iff`

A.187 `PaperXI.FrobeniusPeriod`

Source lines: 97.

SHA-256: `8e34bfa6b4636a8b392bdff1eb6f25eb5c9061f592ae1bff13c3179e48ec14fb`.

```
Minimal return period, exact finite orbit size, and stabilizing powers.
```

Direct project imports: `PaperXI.PacketSupport`.

Line	Kind	Name
12	def	`PaperXI.FrobeniusPeriod.permutation`
14	def	`PaperXI.FrobeniusPeriod.period`
16	theorem	`PaperXI.FrobeniusPeriod.iterate_label`
28	theorem	`PaperXI.FrobeniusPeriod.period_pos`
31	theorem	`PaperXI.FrobeniusPeriod.period_le`
35	theorem	`PaperXI.FrobeniusPeriod.return_label`
43	theorem	`PaperXI.FrobeniusPeriod.stabilizing_power_iff`
49	theorem	`PaperXI.FrobeniusPeriod.no_short_return`
56	theorem	`PaperXI.FrobeniusPeriod.related_iff_nat`
68	theorem	`PaperXI.FrobeniusPeriod.related_iff_bounded`
79	def	`PaperXI.FrobeniusPeriod.orbitEquiv`
93	theorem	`PaperXI.FrobeniusPeriod.orbit_card`

A.188 `PaperXI.FrobeniusPeriodCount`

Source lines: 125.

SHA-256: `53f45d0eb9fed4c2ad104695c3730de6b59560120f164630b865c30a3ccb0749`.

```
Exact-period counting for finite dynamical systems.  The count is the
cardinality of an actual subtype, not a recurrence defined to force inversion.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.FrobeniusPeriodCount.exactCount`
18	def	`PaperXI.FrobeniusPeriodCount.fixedCount`
22	lemma	`PaperXI.FrobeniusPeriodCount.exactCount_eq_filter`
28	lemma	`PaperXI.FrobeniusPeriodCount.fixedCount_eq_filter`
34	theorem	`PaperXI.FrobeniusPeriodCount.sum_exactCount`
59	theorem	`PaperXI.FrobeniusPeriodCount.moebius_exactCount`

Line	Kind	Name
75	def	`PaperXI.FrobeniusPeriodCount.regularPerm`
80	def	`PaperXI.FrobeniusPeriodCount.orbitCount`
84	theorem	`PaperXI.FrobeniusPeriodCount.regularPerm_minimalPeriod`
96	theorem	`PaperXI.FrobeniusPeriodCount.exactCount_eq_orbitCount_mul`
119	theorem	`PaperXI.FrobeniusPeriodCount.period_dvd_exactCount`

A.189 `PaperXI.FullBridgeTransport`

Source lines: 50.

SHA-256: `af018ebd737c4d182525a72c4eddc2561a4ca9a2a228aabd0bac83dae1a9d816`.

```
Actual semidirect and carrier transport of a finite bridge. These endpoints
connect the tensor model's normal and stabilizer equations to full operators.
```

Direct project imports: `PaperXI.CyclicBridge`.

Line	Kind	Name
14	theorem	`PaperXI.FullBridgeTransport.semidirect_along`
36	theorem	`PaperXI.FullBridgeTransport.through_factorizations`

A.190 `PaperXI.FullIntertwiningLocus69`

Source lines: 95.

SHA-256: `d1f43873562d510630801d569413b3ad3b4d659e2b4403e1d74714196488c12a`.

```
Exact convention adapter for the full extended-type intertwining locus.
```

Direct project imports: `PaperXI.InductionIrreducibility`.

Line	Kind	Name
14	def	`PaperXI.FullIntertwiningLocus69.locus`
19	def	`PaperXI.FullIntertwiningLocus69.internalIntertwiner`
30	theorem	`PaperXI.FullIntertwiningLocus69.internalIntertwiner_ne_zero`
41	theorem	`PaperXI.FullIntertwiningLocus69.subgroup_subset`
48	theorem	`PaperXI.FullIntertwiningLocus69.locus_eq_iff`
71	def	`PaperXI.FullIntertwiningLocus69.standardLocus`
75	theorem	`PaperXI.FullIntertwiningLocus69.standardLocus_eq_iff`
88	theorem	`PaperXI.FullIntertwiningLocus69.off_subgroup_of_standard_locus`

A.191 `PaperXI.FullTypeIntertwining105`

Source lines: 151.

SHA-256: `77dddac0fa33899fcac3fba052d3a5bf97fd3f7c3e9443409e6fd3be1e7f7be6`.

```
From the actual finite cuspidal factor to full extended-type Mackey vanishing.
The BK tensor-to-residue reduction and arithmetic Cartan decomposition remain
explicit inputs. The full-H intertwining bound and induced irreducibility do not.
```

Direct project imports: `PaperXI.CuspidalCartan105`, `PaperXI.FullIntertwiningLocus69`, `PaperXI.CompactModCenterInduction68`.

Line	Kind	Name
20	def	`PaperXI.FullTypeIntertwining105.mapOfAlong`
30	theorem	`PaperXI.FullTypeIntertwining105.alongOfMap`
47	theorem	`PaperXI.FullTypeIntertwining105.existsAt_iff_locus`
63	theorem	`PaperXI.FullTypeIntertwining105.restrict_existsAt`
72	theorem	`PaperXI.FullTypeIntertwining105.extension_irreducible`
92	theorem	`PaperXI.FullTypeIntertwining105.full_locus`
120	theorem	`PaperXI.FullTypeIntertwining105.induction_package`

A.192 `PaperXI.GallagherTensorIrreducibility58`

Source lines: 165.

SHA-256: `8bf42d852c9fc0f54c67a1dbdfb96f9df652d86138597dd1db0006bcf90e552b`.

```
# Reverse Clifford tensor irreducibility

The prescribed extension factor is tensored with an irreducible quotient
representation. The resulting action is irreducible; no irreducibility of the
tensor product is assumed. The argument uses Jacobson density on the actual
normal-subgroup action.
```

Direct project imports: `PaperXI.SchurScalar`, `PaperXI.RepresentationStability`.

Line	Kind	Name
25	theorem	`PaperXI.GallagherTensorIrreducibility58.algebraAction_surjective`
59	theorem	`PaperXI.GallagherTensorIrreducibility58.stable_tensor_algebra`
73	theorem	`PaperXI.GallagherTensorIrreducibility58.stable_tensor_end`
83	theorem	`PaperXI.GallagherTensorIrreducibility58.map_rankOne`
95	def	`PaperXI.GallagherTensorIrreducibility58.multiplicitySubspace`
104	theorem	`PaperXI.GallagherTensorIrreducibility58.tensor_irreducible`
158	theorem	`PaperXI.GallagherTensorIrreducibility58.quotient_tensor_irreducible`

A.193 `PaperXI.GaussCharacters`

Source lines: 67.

SHA-256: `f49f317f384804ecff8614197f0acd360d1575046498c320b2027bd7ba16d8ca`.

```
Translation to the manuscript's multiplicative characters kˣ → ℂˣ.
No additional unitarity assumption is needed over a finite field.
```

Direct project imports: `PaperXI.GaussSums`.

Line	Kind	Name
12	def	`PaperXI.GaussCharacters.sum`
16	theorem	`PaperXI.GaussCharacters.sum_eq_unitSum`
22	theorem	`PaperXI.GaussCharacters.extension_ne_one`
33	theorem	`PaperXI.GaussCharacters.extension_inverse`

Line	Kind	Name
38	theorem	`PaperXI.GaussCharacters.magnitude`
45	def	`PaperXI.GaussCharacters.phase`
49	theorem	`PaperXI.GaussCharacters.phase_eq`
54	theorem	`PaperXI.GaussCharacters.phase_norm`
60	theorem	`PaperXI.GaussCharacters.phase_duality`

A.194 `PaperXI.GaussSums`

Source lines: 90.

SHA-256: `ace300219a85a3337ec026bb367507f5088f887306c309e17af938fdfca02d02`.

```
Finite-field Gauss sums with the manuscript's sum over the unit group.
The additive character is explicitly required to be nontrivial.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.GaussSums.unitSum`
20	theorem	`PaperXI.GaussSums.unitSum_eq_gaussSum`
28	theorem	`PaperXI.GaussSums.sum_mul_inverse`
35	theorem	`PaperXI.GaussSums.conjugate_sum`
40	theorem	`PaperXI.GaussSums.norm_sq`
50	theorem	`PaperXI.GaussSums.magnitude`
56	theorem	`PaperXI.GaussSums.sum_duality`
65	def	`PaperXI.GaussSums.phase`
69	theorem	`PaperXI.GaussSums.sqrt_card_pos`
73	theorem	`PaperXI.GaussSums.phase_norm`
80	theorem	`PaperXI.GaussSums.phase_duality`

A.195 `PaperXI.GeneralChain`

Source lines: 89.

SHA-256: `0e5a7f76765c2e3b753f9db36e3886906a9eac283a0583ebb7f07bcb404e33d7`.

```
Proposition 2.3 for any operator satisfying the stated block support.
The transfer model is constructed from the operator; it is not an extra hypothesis.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
12	def	`PaperXI.MonodromyChain.ChainSupported`
15	theorem	`PaperXI.MonodromyChain.arbitrary_block_reconstruction`
29	theorem	`PaperXI.MonodromyChain.projector_eq_single`
36	def	`PaperXI.MonodromyChain.extractedTransfers`
42	theorem	`PaperXI.MonodromyChain.coordinate_block_zero`
49	theorem	`PaperXI.MonodromyChain.eq_extracted_shift`
75	theorem	`PaperXI.MonodromyChain.arbitrary_nilpotent_bound`

Line	Kind	Name
80	theorem	PaperXI.MonodromyChain.arbitrary_power_paths

A.196 PaperXI.GeneratedIrreducibility

Source lines: 55.

SHA-256: 3aca9c09ed2ae98d3df1627466cdadb860be6807be1e7555d2743c1b534bd5b2.

```
Pass from stability under explicit group generators to Mathlib's actual
representation irreducibility. No orbit-transitivity assumption occurs here.
```

Direct project imports: PaperXI.CyclicAction.

Line	Kind	Name
11	theorem	PaperXI.GeneratedIrreducibility.nontrivial_space
20	def	PaperXI.GeneratedIrreducibility.actionEquiv
26	theorem	PaperXI.GeneratedIrreducibility.actionEquiv_apply
29	theorem	PaperXI.GeneratedIrreducibility.actionEquiv_symm_apply
32	theorem	PaperXI.GeneratedIrreducibility.stable_of_generators
47	theorem	PaperXI.GeneratedIrreducibility.eq_bot_or_top

A.197 PaperXI.GlobalHeisenbergBridge112

Source lines: 301.

SHA-256: b44f5fc5e576c356e04e33a27cfbcd1d0f70946a32dafba8389de23b83d83998.

```
A literal global central lift produces the Heisenberg representation bridge;
the target representation equivalence is a conclusion. The actual quotient
endpoint retains the original theta and eta on its original carrier. Existence
of the canonical global quotient lift is not asserted here.
```

Direct project imports: PaperXI.HeisenbergCharacterProfile111, PaperXI.FiniteRepresentation, PaperXI.TypeSideDescent, PaperXI.SchurScalar.

Line	Kind	Name
23	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_common_profile
54	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_common_quotient_form
72	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_global_central_lift
105	theorem	PaperXI.GlobalHeisenbergBridge112.actual_eta_bridge_of_global_lift
146	theorem	PaperXI.GlobalHeisenbergBridge112.generator_discrepancy
168	theorem	PaperXI.GlobalHeisenbergBridge112.generator_discrepancy_pow
179	theorem	PaperXI.GlobalHeisenbergBridge112.unit_eq_one_of_coprime_powers
188	theorem	PaperXI.GlobalHeisenbergBridge112.generator_phase_eq_one_of_det_coprime
224	theorem	PaperXI.GlobalHeisenbergBridge112.generator_equivariant_of_det_coprime
247	theorem	PaperXI.GlobalHeisenbergBridge112.cyclic_equivariant_of_generator
268	theorem	PaperXI.GlobalHeisenbergBridge112.equivariant_bridge_of_common_quotient_form

A.198 PaperXI.GluedPacketSections

Source lines: 170.

SHA-256: 5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf.

```
Actual linear spaces of compatible sections, their representation, and the
map obtained from root holonomy tests. The final section equivalence is built;
a compatible family of comparison maps is not part of the input.
```

Direct project imports: PaperXI.RepresentationGraph.

Line	Kind	Name
22	def	PaperXI.GluedPacketSections.sections
28	theorem	PaperXI.GluedPacketSections.section_edge
37	theorem	PaperXI.GluedPacketSections.section_path
46	theorem	PaperXI.GluedPacketSections.root_evaluation_injective
56	def	PaperXI.GluedPacketSections.loopFixed
63	def	PaperXI.GluedPacketSections.extendRoot
71	def	PaperXI.GluedPacketSections.evaluationEquiv
90	def	PaperXI.GluedPacketSections.sectionRepresentation
101	theorem	PaperXI.GluedPacketSections.sectionRepresentation_apply
106	def	PaperXI.GluedPacketSections.ofGauge
123	def	PaperXI.GluedPacketSections.equivOfGauge
134	def	PaperXI.GluedPacketSections.gluedGauge
138	def	PaperXI.GluedPacketSections.fromChords
152	theorem	PaperXI.GluedPacketSections.fromChords_apply
163	theorem	PaperXI.GluedPacketSections.gauge_unique_at_root

A.199 PaperXI.GraphChordCharacters

Source lines: 144.

SHA-256: 037a0bd69fdb62bbe05ce9760a8f4bcf242c6c01d167c45f6d9bf591cf5bcca0.

```
# Independent scalar tests on fundamental graph cycles

Every assignment of unit scalars to the chords of the constructed rooted tree
is realized by actual one-dimensional edge transport: tree edges carry one and
each chord carries its prescribed scalar. Thus the scalar cycle tests can vary
independently. The same realization gives characters on the actual reduced loop
group and its abelianization.
```

Direct project imports: PaperXI.GraphTree, PaperXI.GraphLoopRepresentation, PaperXI.ScalarHolonomy.

Line	Kind	Name
24	def	PaperXI.GraphChordCharacters.weight
29	theorem	PaperXI.GraphChordCharacters.weight_tree
33	theorem	PaperXI.GraphChordCharacters.weight_chord
39	def	PaperXI.GraphChordCharacters.edgeSystem
43	theorem	PaperXI.GraphChordCharacters.edgeSystem_chord
49	theorem	PaperXI.GraphChordCharacters.traversal_tree
66	theorem	PaperXI.GraphChordCharacters.rootPath_transport
81	def	PaperXI.GraphChordCharacters.chordLoop
87	theorem	PaperXI.GraphChordCharacters.chordLoop_transport

Line	Kind	Name
94	theorem	`PaperXI.GraphChordCharacters.chordLoop_holonomy`
101	theorem	`PaperXI.GraphChordCharacters.field_equiv_scalar`
111	theorem	`PaperXI.GraphChordCharacters.holonomy_isScalar`
117	def	`PaperXI.GraphChordCharacters.character`
121	theorem	`PaperXI.GraphChordCharacters.character_chord`
128	theorem	`PaperXI.GraphChordCharacters.character_tests_surjective`
135	theorem	`PaperXI.GraphChordCharacters.homology_tests_surjective`

A.200 `PaperXI.GraphGauge`

Source lines: 190.

SHA-256: `1750d323d264d4760717c62ab3aa154cbebefcdd24e80cc76ef22d7d8f5a572b`.

```
Actual graph gauges and the rooted matrix holonomy criterion, Paper XI 14.2–14.3.
The cycle tests are the based loops determined by a constructed rooted tree.
```

Direct project imports: `PaperXI.GraphTransport`, `PaperXI.GraphTree`, `PaperXI.GraphLoopRepresentation`, `PaperXI.HolonomyResidual`.

Line	Kind	Name
27	def	`PaperXI.GraphGauge.cycle`
31	def	`PaperXI.GraphGauge.rootGauge`
35	def	`PaperXI.GraphGauge.EdgeCompatible`
39	def	`PaperXI.GraphGauge.LoopCompatible`
43	def	`PaperXI.GraphGauge.CycleCompatible`
47	theorem	`PaperXI.GraphGauge.rootGauge_root`
53	theorem	`PaperXI.GraphGauge.rootGauge_edge_iff`
66	theorem	`PaperXI.GraphGauge.edgeCompatible_loops`
72	theorem	`PaperXI.GraphGauge.rootGauge_compatible_of_loops`
79	theorem	`PaperXI.GraphGauge.gauge_unique`
88	theorem	`PaperXI.GraphGauge.based_iff_loops`
101	theorem	`PaperXI.GraphGauge.rootGauge_parent`
110	theorem	`PaperXI.GraphGauge.rootGauge_tree`
127	theorem	`PaperXI.GraphGauge.rootGauge_compatible_iff_chords`
141	theorem	`PaperXI.GraphGauge.based_iff_chords`
155	theorem	`PaperXI.GraphGauge.connected_based_iff_chords`
163	theorem	`PaperXI.GraphGauge.chords_iff_loops`
170	theorem	`PaperXI.GraphGauge.tree_independence`
177	theorem	`PaperXI.GraphGauge.unbased_iff_conjugates`

A.201 `PaperXI.GraphLoop`

Source lines: 106.

SHA-256: `f38f6d71925623d713432f41101158265cdfa0733f6bd3242f335f62bf01d198`.

```
# Reduced graph loops and their abelianization

Edges belong to the original quiver. A path in its symmetrification can traverse
an edge in either direction. The free groupoid identifies paths under insertion
and deletion of an edge followed by its reverse. Based endomorphisms therefore
```

```
give the combinatorial fundamental group, with multiplication ordered as function
composition: `x * y` traverses `y` before `x`.

`H1` is the abelianization of this actual path quotient. No comparison with a
topological realization or singular homology is asserted here.
```

Direct project imports: none.

Line	Kind	Name
27	abbrev	`PaperXI.GraphLoop.vertex`
31	def	`PaperXI.GraphLoop.pathClass`
35	theorem	`PaperXI.GraphLoop.pathClass_nil`
38	theorem	`PaperXI.GraphLoop.pathClass_comp`
43	theorem	`PaperXI.GraphLoop.pathClass_reverse`
48	theorem	`PaperXI.GraphLoop.pathClass_surjective`
53	abbrev	`PaperXI.GraphLoop.Loop`
56	def	`PaperXI.GraphLoop.loopClass`
59	theorem	`PaperXI.GraphLoop.loopClass_nil`
63	theorem	`PaperXI.GraphLoop.loopClass_comp`
67	theorem	`PaperXI.GraphLoop.loopClass_reverse`
71	theorem	`PaperXI.GraphLoop.loopClass_surjective`
75	abbrev	`PaperXI.GraphLoop.H1`
78	def	`PaperXI.GraphLoop.homologyClass`
80	theorem	`PaperXI.GraphLoop.homologyClass_mul`
84	def	`PaperXI.GraphLoop.factorCharacter`
87	theorem	`PaperXI.GraphLoop.factorCharacter_apply_class`
92	theorem	`PaperXI.GraphLoop.factorCharacter_unique`
99	theorem	`PaperXI.GraphLoop.character_ext_paths`

A.202 `PaperXI.GraphLoopRepresentation`

Source lines: 138.

SHA-256: `8e6bb58b55ba016228d5ae3cd85b19e40ccde0abd116f4a13101a8d8939a4e95`.

```
# Holonomy as a homomorphism on actual reduced graph loops

The free-groupoid extension of the edge system agrees with the recursively
defined path transport on every represented path. Surjectivity of path classes
then identifies intertwining all based paths with conjugacy of the resulting
honest group homomorphisms.
```

Direct project imports: `PaperXI.GraphLoop`, `PaperXI.GraphTransport`.

Line	Kind	Name
25	structure	`PaperXI.GraphLoop.Fiber`
28	instance	`PaperXI.GraphLoop.fiberGroupoid`
40	def	`PaperXI.GraphLoop.edgePrefunctor`
46	def	`PaperXI.GraphLoop.transportFunctor`
51	def	`PaperXI.GraphLoop.transport`
55	theorem	`PaperXI.GraphLoop.transport_id`
60	theorem	`PaperXI.GraphLoop.transport_comp`
66	theorem	`PaperXI.GraphLoop.transport_edge`
73	theorem	`PaperXI.GraphLoop.transport_pathClass`

Line	Kind	Name
90	def	PaperXI.GraphLoop.holonomy
96	theorem	PaperXI.GraphLoop.holonomy_loopClass
105	def	PaperXI.GraphLoop.Conjugates
112	theorem	PaperXI.GraphLoop.conjugates_iff_paths
129	theorem	PaperXI.GraphLoop.exists_conjugates_iff_paths

A.203 PaperXI.GraphResidual

Source lines: 274.

SHA-256: ffc35ff3cd92cf45144e94c501a36af8567e4bb7531e9dc21eb81401385adbf8.

```
Paper XI, 14.6 and 14.7 on the actual rooted graph paths. Finite dimensionality
makes the given edge equivalences and their actual inverses continuous for the
specified norms. Parallel edges and self-loops keep their identities.
```

Direct project imports: PaperXI.ContinuousHolonomyResidual, PaperXI.GraphTransport, PaperXI.GraphTree.

Line	Kind	Name
24	def	PaperXI.GraphResidual.pathL
28	theorem	PaperXI.GraphResidual.pathL_toLinearEquiv
33	theorem	PaperXI.GraphResidual.pathL_apply
38	theorem	PaperXI.GraphResidual.pathL_symm_apply
43	theorem	PaperXI.GraphResidual.pathL_reverse
50	theorem	PaperXI.GraphResidual.pathL_comp
60	def	PaperXI.GraphResidual.gauge
67	theorem	PaperXI.GraphResidual.gauge_toLinearEquiv
74	theorem	PaperXI.GraphResidual.gauge_apply
81	theorem	PaperXI.GraphResidual.gauge_root
90	theorem	PaperXI.GraphResidual.gauge_parent
106	def	PaperXI.GraphResidual.edgeResidual
116	def	PaperXI.GraphResidual.cycleDefect
125	theorem	PaperXI.GraphResidual.cycleDefect_toLinearMap
134	theorem	PaperXI.GraphResidual.edgeResidual_toLinearMap
144	theorem	PaperXI.GraphResidual.cycleDefect_loop
162	theorem	PaperXI.GraphResidual.residual_factorization
173	theorem	PaperXI.GraphResidual.residual_zero_iff
184	theorem	PaperXI.GraphResidual.residual_norm_bound
196	theorem	PaperXI.GraphResidual.uniform_bound
212	theorem	PaperXI.GraphResidual.uniform_product_bound
228	theorem	PaperXI.GraphResidual.uniform_chord_bound
246	theorem	PaperXI.GraphResidual.uniform_chord_product_bound
264	theorem	PaperXI.GraphResidual.connected_residual_bound

A.204 PaperXI.GraphTransport

Source lines: 150.

SHA-256: 72c2feeee0c9f9e98eeed01a1ff511d439bdfdac6c3bdb2732290dcc1f8151fb.

```
Paper XI, Section 14. Actual typed path transport on the symmetrification
of a quiver. An original arrow is one chosen orientation of an underlying edge;
parallel arrows and self-loops keep their identities. Concatenation traverses
its left argument first.
```

Direct project imports: none.

Line	Kind	Name
17	abbrev	`PaperXI.GraphTransport.Traversal`
20	abbrev	`PaperXI.GraphTransport.Walk`
25	def	`PaperXI.GraphTransport.symmetrifyAdd`
29	def	`PaperXI.GraphTransport.symmetrifyModule`
33	def	`PaperXI.GraphTransport.edge`
39	theorem	`PaperXI.GraphTransport.edge_inl`
42	theorem	`PaperXI.GraphTransport.edge_inr`
45	theorem	`PaperXI.GraphTransport.edge_reverse`
51	def	`PaperXI.GraphTransport.path`
56	theorem	`PaperXI.GraphTransport.nil`
59	theorem	`PaperXI.GraphTransport.cons`
63	theorem	`PaperXI.GraphTransport.toPath`
68	theorem	`PaperXI.GraphTransport.comp`
77	theorem	`PaperXI.GraphTransport.reverse`
85	theorem	`PaperXI.GraphTransport.path_reverse_cancel`
92	theorem	`PaperXI.GraphTransport.reverse_path_cancel`
100	theorem	`PaperXI.GraphTransport.cancel`
110	theorem	`PaperXI.GraphTransport.gauge_edge`
125	theorem	`PaperXI.GraphTransport.gauge_path`
139	theorem	`PaperXI.GraphTransport.gauge_path_equiv`

A.205 `PaperXI.GraphTree`

Source lines: 245.

SHA-256: `37a1cea3da4966b2d631d1f44a997095037c88ca15a4b3001a6b8efa607dd486`.

```
A rooted spanning tree of a connected quiver, with the identities of the
original edges retained. The construction selects a predecessor along a
shortest path in the symmetrified quiver.
```

Direct project imports: none.

Line	Kind	Name
18	abbrev	`PaperXI.GraphTree.Edge`
21	abbrev	`PaperXI.GraphTree.Traversal`
24	abbrev	`PaperXI.GraphTree.Path`
27	def	`PaperXI.GraphTree.Connected`
30	structure	`PaperXI.GraphTree.Rooted`
40	theorem	`PaperXI.GraphTree.exists_length`
45	def	`PaperXI.GraphTree.distance`
49	theorem	`PaperXI.GraphTree.distance_le`
53	theorem	`PaperXI.GraphTree.distance_root`
57	theorem	`PaperXI.GraphTree.exists_predecessor`
71	def	`PaperXI.GraphTree.ofConnected`

Line	Kind	Name
83	def	PaperXI.GraphTree.rootPath
92	theorem	PaperXI.GraphTree.rootPath_root
96	theorem	PaperXI.GraphTree.rootPath_parent
102	theorem	PaperXI.GraphTree.rooted_induction
112	def	PaperXI.GraphTree.underlying
118	def	PaperXI.GraphTree.selected
120	theorem	PaperXI.GraphTree.underlying_endpoints
125	theorem	PaperXI.GraphTree.selected_injective
152	def	PaperXI.GraphTree.treeEdges
155	def	PaperXI.GraphTree.Chord
157	theorem	PaperXI.GraphTree.mem_treeEdges
160	theorem	PaperXI.GraphTree.treeEdges_induction
167	theorem	PaperXI.GraphTree.selected_not_loop
183	def	PaperXI.GraphTree.Supported
188	theorem	PaperXI.GraphTree.rootPath_supported
196	theorem	PaperXI.GraphTree.treeEdges_no_loop
200	theorem	PaperXI.GraphTree.tree_card
211	theorem	PaperXI.GraphTree.chord_count
226	theorem	PaperXI.GraphTree.edge_count_lower_bound
233	theorem	PaperXI.GraphTree.chord_card
239	theorem	PaperXI.GraphTree.chord_card_int

A.206 PaperXI.GraphTreePaths

Source lines: 87.

SHA-256: 732095eebf2b794f7d31d503a7928bfde9b7d40da777fd5c99b3d1cccd1e2e18.

```
Actual tree paths intertwine every reversible system of edge transports.
Tree membership refers to the chosen original edge identity.
```

Direct project imports: PaperXI.GraphTree, PaperXI.GraphTransport.

Line	Kind	Name
20	theorem	PaperXI.GraphTreePaths.parent_transport
27	theorem	PaperXI.GraphTreePaths.tree_transport
55	theorem	PaperXI.GraphTreePaths.traversal_transport
70	theorem	PaperXI.GraphTreePaths.supported_transport
82	theorem	PaperXI.GraphTreePaths.tree_loop_identity

A.207 PaperXI.GroupFixed

Source lines: 85.

SHA-256: 32a4e3c21abaf0e27169d70976a894f79172987de4954913d8fc607fd925e51b.

```
Fixedness under an actual group, including supplied inertia quotient data.
```

Direct project imports: PaperXI.CycleFixed, PaperXI.InertiaQuotient.

Line	Kind	Name
12	theorem	`PaperXI.GroupFixed.fixed_of_generators`
22	theorem	`PaperXI.GroupFixed.invariants_eq_common`
46	theorem	`PaperXI.GroupFixed.orbit_basis_steps`
63	theorem	`PaperXI.GroupFixed.packet_finrank`

A.208 `PaperXI.HeisenbergCardinality`

Source lines: 65.

SHA-256: `3325cc5f2651ba031ec6ab41f813ec4f9e63afea875a3e4cac9827997fef3225`.

```
Dimension and cardinality of a maximal isotropic subspace in the standard
Heisenberg symplectic space. No cardinality or self-orthogonality conclusion is
assumed as an input.
```

Direct project imports: `PaperXI.HeisenbergForm`, `PaperXI.HeisenbergGeometry`.

Line	Kind	Name
17	theorem	`PaperXI.HeisenbergCardinality.form_isAlt`
19	theorem	`PaperXI.HeisenbergCardinality.form_separatingLeft`
29	theorem	`PaperXI.HeisenbergCardinality.form_nondegenerate`
37	theorem	`PaperXI.HeisenbergCardinality.form_forall_eq_zero_iff`
41	theorem	`PaperXI.HeisenbergCardinality.maximal_orthogonal_eq`
48	theorem	`PaperXI.HeisenbergCardinality.finrank_eq`
54	theorem	`PaperXI.HeisenbergCardinality.natCard_eq_pow`
59	theorem	`PaperXI.HeisenbergCardinality.natCard_eq`

A.209 `PaperXI.HeisenbergCenter`

Source lines: 64.

SHA-256: `f2e8ae8af020e8557e7b08bb957b7145733776c6b5c83c37218c1e0e184f4966`.

```
The center is proved to consist exactly of the central-coordinate elements.
```

Direct project imports: `PaperXI.HeisenbergGroup`, `PaperXI.HeisenbergCardinality`.

Line	Kind	Name
15	theorem	`PaperXI.HeisenbergCenter.commute_iff`
35	theorem	`PaperXI.HeisenbergCenter.mem_center_iff`
51	theorem	`PaperXI.HeisenbergCenter.center_eq_centralElement`

A.210 `PaperXI.HeisenbergCharacterProfile111`

Source lines: 317.

SHA-256: `1b222ac6ed229dac1c3c2682d5340cc880a2699c3c0ca3c9cbc80ae11bcb712e`.

```
The original type-side quotient/form datum derives trace support through
actual commutators. Character intertwining of the original one-dimensional
theta is equivalent to its covariance on the literal intersection. Neither
result assumes the desired eta intertwiner or eta intersection dimension.
```

Direct project imports: `PaperXI.DegreeIndex110`.

Line	Kind	Name
21	def	`PaperXI.HeisenbergCharacterProfile111.coordinate`
24	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_surjective`
31	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_eq_zero_iff`
44	theorem	`PaperXI.HeisenbergCharacterProfile111.commutator_mem`
52	def	`PaperXI.HeisenbergCharacterProfile111.characterCommutator`
59	theorem	`PaperXI.HeisenbergCharacterProfile111.commutator_detection_of_nondegenerate_form`
94	theorem	`PaperXI.HeisenbergCharacterProfile111.trace_support_of_nondegenerate_form`
116	def	`PaperXI.HeisenbergCharacterProfile111.transportedQuotient`
120	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_transport`
124	def	`PaperXI.HeisenbergCharacterProfile111.pulledBackCharacter`
128	theorem	`PaperXI.HeisenbergCharacterProfile111.characterCommutator_transport`
140	theorem	`PaperXI.HeisenbergCharacterProfile111.form_transport`
160	theorem	`PaperXI.HeisenbergCharacterProfile111.character_intertwinedAt_iff_covariance`
187	theorem	`PaperXI.HeisenbergCharacterProfile111.character_values_of_intertwinedAt`
216	instance	`PaperXI.HeisenbergCharacterProfile111.actualScalarSubgroup_normal`
237	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_trace_support_of_quotient_form`
261	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_covariance_of_character_intertwining`
283	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_eta_dimension_from_quotient_form`

A.211 `PaperXI.HeisenbergCochain`

Source lines: 69.

SHA-256: `ba7c5f4ea39d1e68eadd6f852cf159e67a2b0c454eb6083d92f7d48ca9e92f8f`.

```
Actual changes of central section and the square obstruction in characteristic
two. These statements concern the middle groups, not just their commutator forms.
```

Direct project imports: `PaperXI.BilinearHeisenberg`.

Line	Kind	Name
14	def	`PaperXI.HeisenbergCochain.equiv`
28	theorem	`PaperXI.HeisenbergCochain.equiv_center`
33	theorem	`PaperXI.HeisenbergCochain.equiv_vector`
41	theorem	`PaperXI.HeisenbergCochain.square`
51	theorem	`PaperXI.HeisenbergCochain.diagonal_preserved`
62	theorem	`PaperXI.HeisenbergCochain.no_equiv_of_diagonal_mismatch`

A.212 `PaperXI.HeisenbergComparison`

Source lines: 92.

SHA-256: `545124ccd81c40a4378e2d854940a0ddb3402c6081775d92751fa13fa0a7f96b`.

```
The first actual intertwiner between arbitrary irreducible actions along
a group equivalence from the odd-characteristic coordinate Heisenberg group.
```

```
The coordinate isometry specialization constructs that group equivalence.
Abstract symplectic coordinates, source quotient actions, and the dyadic
extension branch are not supplied by this module.
```

Direct project imports: `PaperXI.HeisenbergTransport`, `PaperXI.SchrodingerUniversal`.

Line	Kind	Name
21	def	`PaperXI.HeisenbergComparison.equivAlong`
32	theorem	`PaperXI.HeisenbergComparison.equivAlong_intertwines`
46	theorem	`PaperXI.HeisenbergComparison.exists_unique_along`
75	theorem	`PaperXI.HeisenbergComparison.exists_unique_of_isometry`

A.213 `PaperXI.HeisenbergDimension`

Source lines: 41.

SHA-256: `08109693bdd53c46678bff188224863dda5f276ce70dee638238a08c28317a02`.

```
Dimension and square-root cardinality in the finite Schrödinger model.
```

Direct project imports: `PaperXI.HeisenbergCardinality`.

Line	Kind	Name
16	theorem	`PaperXI.HeisenbergDimension.natCard_dual`
21	theorem	`PaperXI.HeisenbergDimension.natCard_symplectic`
24	theorem	`PaperXI.HeisenbergDimension.sqrt_natCard_symplectic`
31	theorem	`PaperXI.HeisenbergDimension.finrank_function`
34	theorem	`PaperXI.HeisenbergDimension.finrank_function_pow`
37	theorem	`PaperXI.HeisenbergDimension.finrank_function_sqrt`

A.214 `PaperXI.HeisenbergFinite`

Source lines: 32.

SHA-256: `d3703379c8db33d8675a342dc898f778aa75b1599529098f11b2af1df156e553`.

```
Finiteness and order of the actual Heisenberg group.
```

Direct project imports: `PaperXI.HeisenbergGroup`, `PaperXI.HeisenbergDimension`.

Line	Kind	Name
16	def	`PaperXI.HeisenbergFinite.coordinates`
22	instance	`PaperXI.HeisenbergFinite.instFinite`
27	theorem	`PaperXI.HeisenbergFinite.natCard_eq`

A.215 `PaperXI.HeisenbergForm`

Source lines: 50.

SHA-256: `b9d055bf58ee5326c45c49e9267dcd9b113d6b1d2b3f0f82d5e7cd909890230c`.

```
The manuscript's standard alternating form on X plus its linear dual.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.HeisenbergForm.form`
41	theorem	`PaperXI.HeisenbergForm.form_apply`
44	theorem	`PaperXI.HeisenbergForm.form_self`
47	theorem	`PaperXI.HeisenbergForm.form_skew`

A.216 `PaperXI.HeisenbergFrequency`

Source lines: 197.

SHA-256: `511658b8ac1ca7f93b8002ee2a626418b076f2cd34875bd57bbad47a4deb6b59`.

```
Frequency projectors in an arbitrary representation of the actual Heisenberg group.
Completeness and translation produce a nonzero modulation-fixed vector without any
assumption on the dimension of the target representation.
```

Direct project imports: `PaperXI.DualCharacterSums`, `PaperXI.HeisenbergNormalForm`, `PaperXI.HeisenbergRepresentation`.

Line	Kind	Name
19	def	`PaperXI.HeisenbergFrequency.projector`
24	theorem	`PaperXI.HeisenbergFrequency.projector_apply`
31	theorem	`PaperXI.HeisenbergFrequency.eigen`
63	theorem	`PaperXI.HeisenbergFrequency.complete`
92	theorem	`PaperXI.HeisenbergFrequency.projector_of_eigen`
114	theorem	`PaperXI.HeisenbergFrequency.orthogonal_idempotents`
127	theorem	`PaperXI.HeisenbergFrequency.modulation_translation_action`
147	theorem	`PaperXI.HeisenbergFrequency.translation_covariance`
169	theorem	`PaperXI.HeisenbergFrequency.nonzero_zero_fiber`

A.217 `PaperXI.HeisenbergGeometry`

Source lines: 82.

SHA-256: `df27cc30cbdcf1f4f8ac3ac85c90e2266dcc866bcd81ea3fbdc18d9ac3022c7c`.

```
# Maximal isotropic subspaces

Maximality is by inclusion among subspaces on which the alternating form vanishes.
The equality with the orthogonal complement is derived from that hypothesis.
```

Direct project imports: none.

Line	Kind	Name
21	def	`PaperXI.HeisenbergGeometry.IsIsotropic`
25	def	`PaperXI.HeisenbergGeometry.IsMaximalIsotropic`
28	theorem	`PaperXI.HeisenbergGeometry.isotropic_iff_le_orthogonal`
37	theorem	`PaperXI.HeisenbergGeometry.isotropic_sup_span`

Line	Kind	Name
50	theorem	`PaperXI.HeisenbergGeometry.orthogonal_eq`
60	theorem	`PaperXI.HeisenbergGeometry.maximal_iff_orthogonal_eq`
72	theorem	`PaperXI.HeisenbergGeometry.two_mul_finrank`

A.218 `PaperXI.HeisenbergGroup`

Source lines: 109.

SHA-256: `1a146672c602d8cacb7184a8bf364365676d1766a0ffd7af7a1e1c56652776dd`.

```
The Heisenberg group in position, frequency, and central coordinates.
The alternating bilinear cocycle defines a group in every characteristic.
Identifying its commutator with the symplectic pairing additionally needs 2 ≠ 0.
```

Direct project imports: none.

Line	Kind	Name
12	structure	`PaperXI.Heisenberg.Element`
19	instance	`PaperXI.Heisenberg.instMul`
23	instance	`PaperXI.Heisenberg.instOne`
26	instance	`PaperXI.Heisenberg.instInv`
29	theorem	`PaperXI.Heisenberg.mul_position`
32	theorem	`PaperXI.Heisenberg.mul_frequency`
35	theorem	`PaperXI.Heisenberg.mul_central`
39	theorem	`PaperXI.Heisenberg.one_position`
40	theorem	`PaperXI.Heisenberg.one_frequency`
41	theorem	`PaperXI.Heisenberg.one_central`
42	theorem	`PaperXI.Heisenberg.inv_position`
43	theorem	`PaperXI.Heisenberg.inv_frequency`
44	theorem	`PaperXI.Heisenberg.inv_central`
47	instance	`PaperXI.Heisenberg.instGroup`
59	def	`PaperXI.Heisenberg.centralElement`
62	def	`PaperXI.Heisenberg.translationElement`
65	def	`PaperXI.Heisenberg.modulationElement`
67	theorem	`PaperXI.Heisenberg.centralElement_position`
69	theorem	`PaperXI.Heisenberg.centralElement_frequency`
71	theorem	`PaperXI.Heisenberg.centralElement_central`
73	theorem	`PaperXI.Heisenberg.translationElement_position`
75	theorem	`PaperXI.Heisenberg.translationElement_frequency`
77	theorem	`PaperXI.Heisenberg.translationElement_central`
79	theorem	`PaperXI.Heisenberg.modulationElement_position`
81	theorem	`PaperXI.Heisenberg.modulationElement_frequency`
83	theorem	`PaperXI.Heisenberg.modulationElement_central`
86	theorem	`PaperXI.Heisenberg.centralElement_zero`
87	theorem	`PaperXI.Heisenberg.translationElement_zero`
88	theorem	`PaperXI.Heisenberg.modulationElement_zero`
91	theorem	`PaperXI.Heisenberg.centralElement_mul`
95	theorem	`PaperXI.Heisenberg.translationElement_mul`
99	theorem	`PaperXI.Heisenberg.modulationElement_mul`
104	theorem	`PaperXI.Heisenberg.centralElement_commute`

A.219 PaperXI.HeisenbergNormalForm

Source lines: 33.

SHA-256: 61b84ce5c3a491de98e96f9f6edee89760c3d6516081c9372b8cc2d4cb98c1b2.

Normal forms and signed commutation in the actual coordinate Heisenberg group.

Direct project imports: PaperXI.HeisenbergGroup.

Line	Kind	Name
10	theorem	PaperXI.Heisenberg.normal_form
16	theorem	PaperXI.Heisenberg.modulation_translation
25	theorem	PaperXI.Heisenberg.translation_modulation

A.220 PaperXI.HeisenbergPrime

Source lines: 29.

SHA-256: 5f481efb90ff5aa531e502311d9145cf59b1546638b05905f8dbce01b4be4dc9.

The odd-prime specialization uses exactly the manuscript's exponential
additive character, with nontriviality and invertibility of two proved.

Direct project imports: PaperXI.SchrodingerOperator.

Line	Kind	Name
12	theorem	PaperXI.HeisenbergPrime.two_ne_zero
17	theorem	PaperXI.HeisenbergPrime.character_nontrivial
24	theorem	PaperXI.HeisenbergPrime.character_formula

A.221 PaperXI.HeisenbergProjector

Source lines: 230.

SHA-256: 8d41365d8376b7d75e5d8bd5fed0875e6d650020b0341126c480b93f5bfe0d6a.

Actual zero-central isotropic lifts and their orthogonal averaging projectors.

Direct project imports: PaperXI.HeisenbergRepresentation, PaperXI.HeisenbergForm, PaperXI.HeisenbergGeometry, PaperXI.SchrodingerUnitary, PaperXI.FiniteFunctionAverage.

Line	Kind	Name
16	def	PaperXI.HeisenbergProjector.liftHom
25	theorem	PaperXI.HeisenbergProjector.lift_position
29	theorem	PaperXI.HeisenbergProjector.lift_frequency
33	theorem	PaperXI.HeisenbergProjector.lift_central
37	theorem	PaperXI.HeisenbergProjector.lift_injective
45	def	PaperXI.HeisenbergProjector.liftedSubgroup
48	theorem	PaperXI.HeisenbergProjector.mem_liftedSubgroup_iff
58	def	PaperXI.HeisenbergProjector.liftEquiv

Line	Kind	Name
67	instance	`PaperXI.HeisenbergProjector.liftedFintype`
71	theorem	`PaperXI.HeisenbergProjector.liftEquiv_coe`
75	def	`PaperXI.HeisenbergProjector.restriction`
80	theorem	`PaperXI.HeisenbergProjector.restriction_apply`
85	def	`PaperXI.HeisenbergProjector.projector`
89	theorem	`PaperXI.HeisenbergProjector.projector_apply`
101	def	`PaperXI.HeisenbergProjector.subgroupProjector`
106	theorem	`PaperXI.HeisenbergProjector.projector_eq_subgroupProjector`
121	theorem	`PaperXI.HeisenbergProjector.idempotent`
127	theorem	`PaperXI.HeisenbergProjector.range_eq_invariants`
136	theorem	`PaperXI.HeisenbergProjector.pairing_invariant`
143	theorem	`PaperXI.HeisenbergProjector.self_adjoint`
155	theorem	`PaperXI.HeisenbergProjector.residual_orthogonal`
173	def	`PaperXI.HeisenbergProjector.positionSpace`
179	def	`PaperXI.HeisenbergProjector.positionEquiv`
187	instance	`PaperXI.HeisenbergProjector.positionFintype`
190	theorem	`PaperXI.HeisenbergProjector.position_isotropic`
196	theorem	`PaperXI.HeisenbergProjector.position_projector_apply`
214	theorem	`PaperXI.HeisenbergProjector.position_projector_eq`
220	theorem	`PaperXI.HeisenbergProjector.position_range_eq`
225	theorem	`PaperXI.HeisenbergProjector.position_rank_one`

A.222 `PaperXI.HeisenbergReduction107`

Source lines: 61.

SHA-256: `7bb3cef0f423eb861a688bbf662334632a9bfeaf015ddd1b429d554fc2c19934`.

```
Assemble the operator proof and two-sided compact transport. The only
arithmetic inputs are support and line structure for the restricted first factor.
```

Direct project imports: `PaperXI.TensorIntertwining107`.

Line	Kind	Name
15	theorem	`PaperXI.HeisenbergReduction107.pullback`
27	theorem	`PaperXI.HeisenbergReduction107.double_coset_reduction`
43	theorem	`PaperXI.HeisenbergReduction107.integral_reduction`

A.223 `PaperXI.HeisenbergRepresentation`

Source lines: 72.

SHA-256: `386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492`.

```
The actual finite Schrödinger representation of the Heisenberg group,
its invertible operators, central character, and irreducibility.
```

Direct project imports: `PaperXI.HeisenbergGroup`, `PaperXI.SchrodingerOperator`, `PaperXI.SchrodingerIrreducible`.

Line	Kind	Name
14	def	`PaperXI.Heisenberg.representation`
21	theorem	`PaperXI.Heisenberg.representation_apply`
28	def	`PaperXI.Heisenberg.representationGL`
32	theorem	`PaperXI.Heisenberg.representationGL_coe`
38	theorem	`PaperXI.Heisenberg.representation_isUnit`
43	theorem	`PaperXI.Heisenberg.central_action`
47	theorem	`PaperXI.Heisenberg.translation_action`
51	theorem	`PaperXI.Heisenberg.modulation_action`
58	theorem	`PaperXI.Heisenberg.irreducible`

A.224 `PaperXI.HeisenbergSelector`

Source lines: 109.

SHA-256: `89d25c89f133dcf4a56a48864b73092bcf942c81c09adf27fcd7b8f8149c6643`.

```
Trace and rank of the literal maximal isotropic averaging projector.
```

Direct project imports: `PaperXI.FiniteAverage`, `PaperXI.SchrodingerTrace`, `PaperXI.HeisenbergCardinality`, `PaperXI.HeisenbergProjector`.

Line	Kind	Name
17	theorem	`PaperXI.HeisenbergSelector.trace_average`
23	theorem	`PaperXI.HeisenbergSelector.rank_average`
30	theorem	`PaperXI.HeisenbergSelector.invariants_rank_one_of_trace_one`
42	theorem	`PaperXI.HeisenbergSelector.coordinates_zero_iff`
57	theorem	`PaperXI.HeisenbergSelector.trace_restriction`
66	theorem	`PaperXI.HeisenbergSelector.trace_projector`
77	theorem	`PaperXI.HeisenbergSelector.maximal_trace_one`
88	theorem	`PaperXI.HeisenbergSelector.maximal_rank_one`
100	theorem	`PaperXI.HeisenbergSelector.maximal_fixed_line`

A.225 `PaperXI.HeisenbergSource`

Source lines: 63.

SHA-256: `ffb52975ff4f261a1791da759b032fd5c736abf36257053961a5b5b84c963686`.

```
Source propositions 7.2 and 7.3 for odd prime p and the canonical
exponential character, with the manuscript's dimension parameter.
```

Direct project imports: `PaperXI.HeisenbergPrime`, `PaperXI.HeisenbergRepresentation`, `PaperXI.HeisenbergDimension`, `PaperXI.HeisenbergFinite`, `PaperXI.HeisenbergCenter`, `PaperXI.SchrodingerTrace`.

Line	Kind	Name
19	def	`PaperXI.HeisenbergSource.model`
22	theorem	`PaperXI.HeisenbergSource.model_formula`
27	theorem	`PaperXI.HeisenbergSource.center_character`
33	theorem	`PaperXI.HeisenbergSource.model_invertible`

Line	Kind	Name
38	theorem	PaperXI.HeisenbergSource.model_irreducible
44	theorem	PaperXI.HeisenbergSource.dimension_power
48	theorem	PaperXI.HeisenbergSource.dimension_sqrt
52	theorem	PaperXI.HeisenbergSource.character_support

A.226 PaperXI.HeisenbergSourceProjector

Source lines: 118.

SHA-256: 5e8ee76ad84809af2c219cb594f6ae86e0538557cd45997e05b8df888155396e.

```
Source propositions 7.5 and 7.6 for the canonical odd-prime model.
The finite enumeration of each isotropic subspace is constructed here.
```

Direct project imports: PaperXI.HeisenbergSource, PaperXI.HeisenbergSelector, PaperXI.HeisenbergProjector.

Line	Kind	Name
17	def	PaperXI.HeisenbergSourceProjector.projector
24	theorem	PaperXI.HeisenbergSourceProjector.idempotent
34	theorem	PaperXI.HeisenbergSourceProjector.range_eq_invariants
46	theorem	PaperXI.HeisenbergSourceProjector.self_adjoint
57	theorem	PaperXI.HeisenbergSourceProjector.residual_orthogonal
68	theorem	PaperXI.HeisenbergSourceProjector.lift_central_iff_identity
76	theorem	PaperXI.HeisenbergSourceProjector.position_range
87	theorem	PaperXI.HeisenbergSourceProjector.position_rank_one
97	theorem	PaperXI.HeisenbergSourceProjector.maximal_rank_one
107	theorem	PaperXI.HeisenbergSourceProjector.maximal_fixed_line

A.227 PaperXI.HeisenbergTensor

Source lines: 99.

SHA-256: c869c46b15622f6c072689647c34c28046d5f9ff18bbc8191683f0588727d4e7.

```
The actual tensor action descends from the direct product, with one retained
central factor, to the balanced central product. Its central character occurs once.
```

Direct project imports: PaperXI.FiniteTensor, PaperXI.CentralProduct.

Line	Kind	Name
17	def	PaperXI.HeisenbergTensor.ambient
27	theorem	PaperXI.HeisenbergTensor.ambient_apply
33	theorem	PaperXI.HeisenbergTensor.ambient_tprod
41	theorem	PaperXI.HeisenbergTensor.ambient_balanced
55	def	PaperXI.HeisenbergTensor.descend
61	theorem	PaperXI.HeisenbergTensor.descend_mk
68	theorem	PaperXI.HeisenbergTensor.descend_tprod
77	theorem	PaperXI.HeisenbergTensor.central

Line	Kind	Name
88	theorem	PaperXI.HeisenbergTensor.empty_central

A.228 PaperXI.HeisenbergTensorComparison

Source lines: 70.

SHA-256: cc8aa01b6a54bea28296549299ad41b7873e08c01703c5aca621d700c4469661.

```
Tensor factorization transported along an actual equivalence to a declared
full central extension. The comparisons intertwine actual group actions, not
only vector-space dimensions.
```

Direct project imports: PaperXI.HeisenbergTensorFactorization, PaperXI.CentralExtensionComparison.

Line	Kind	Name
23	def	PaperXI.HeisenbergTensor.factorizationAlong
33	theorem	PaperXI.HeisenbergTensor.factorizationAlong_intertwines
43	def	PaperXI.HeisenbergTensor.onFullEnvelope
48	def	PaperXI.HeisenbergTensor.fullEnvelopeFactorization
60	theorem	PaperXI.HeisenbergTensor.fullEnvelopeFactorization_intertwines

A.229 PaperXI.HeisenbergTensorFactorization

Source lines: 87.

SHA-256: 03fec0d85721da724851da3e533aa76b616b5ba39ddfc3fe9a3ab5b513399b16.

```
Finite Heisenberg tensor factorization for the actual central product.
All factor spaces are actual irreducible representation spaces. Their finite
dimensions and nontriviality are derived, including zero quotient factors and
the empty family. The descended tensor action is proved irreducible before
Stone--von Neumann uniqueness is used.
```

Direct project imports: PaperXI.HeisenbergTensor, PaperXI.CentralProductCommutator, PaperXI.TensorIrreducible.

Line	Kind	Name
27	theorem	PaperXI.HeisenbergTensor.factor_finiteDimensional
31	theorem	PaperXI.HeisenbergTensor.finiteDimensional
36	theorem	PaperXI.HeisenbergTensor.nontrivial
42	theorem	PaperXI.HeisenbergTensor.finrank_sq
51	theorem	PaperXI.HeisenbergTensor.irreducible
61	theorem	PaperXI.HeisenbergTensor.exists_factorization
71	def	PaperXI.HeisenbergTensor.factorization
78	theorem	PaperXI.HeisenbergTensor.factorization_intertwines

A.230 PaperXI.HeisenbergTrace108

Source lines: 47.

SHA-256: 045edb4a7a7672920cc0a4fdfc44e0b7b64d78563fe3cf3c625335b1e09240c8.

```
The finite Heisenberg character formula, derived from scalar restriction
and detection by actual commutators. No trace-support hypothesis is supplied.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.HeisenbergTrace108.character_on_subgroup`
23	theorem	`PaperXI.HeisenbergTrace108.character_zero_of_commutator`
40	theorem	`PaperXI.HeisenbergTrace108.character_off_subgroup`

A.231 `PaperXI.HeisenbergTransport`

Source lines: 180.

SHA-256: `7be38801f777c48277b8229987482c3b7e8c656c351a2e5201c7a067740572e2`.

```
Transport of the coordinate Heisenberg group and actual representations.
The lift preserves the alternating cocycle in every characteristic. Its use
as the nondegenerate Heisenberg extension requires odd characteristic.
No coordinates on an abstract symplectic space are asserted here.
```

Direct project imports: `PaperXI.HeisenbergRepresentation`, `PaperXI.HeisenbergForm`.

Line	Kind	Name
20	def	`PaperXI.HeisenbergTransport.vector`
22	theorem	`PaperXI.HeisenbergTransport.vector_mul`
25	def	`PaperXI.HeisenbergTransport.PreservesForm`
28	theorem	`PaperXI.HeisenbergTransport.preservesForm_refl`
31	theorem	`PaperXI.HeisenbergTransport.PreservesForm.symm`
37	theorem	`PaperXI.HeisenbergTransport.PreservesForm.trans`
46	def	`PaperXI.HeisenbergTransport.liftIsometry`
69	theorem	`PaperXI.HeisenbergTransport.liftIsometry_vector`
74	theorem	`PaperXI.HeisenbergTransport.liftIsometry_central`
79	theorem	`PaperXI.HeisenbergTransport.liftIsometry_centralElement`
90	theorem	`PaperXI.HeisenbergTransport.liftIsometry_refl`
95	theorem	`PaperXI.HeisenbergTransport.liftIsometry_symm`
101	theorem	`PaperXI.HeisenbergTransport.liftIsometry_trans`
115	def	`PaperXI.HeisenbergTransport.pullback`
118	theorem	`PaperXI.HeisenbergTransport.pullback_apply`
121	theorem	`PaperXI.HeisenbergTransport.pullback_refl`
126	theorem	`PaperXI.HeisenbergTransport.pullback_trans`
130	theorem	`PaperXI.HeisenbergTransport.pullback_symm`
136	def	`PaperXI.HeisenbergTransport.pullbackSubrepresentationEquiv`
145	theorem	`PaperXI.HeisenbergTransport.pullback_irreducible_iff`
149	theorem	`PaperXI.HeisenbergTransport.pullback_irreducible`
155	theorem	`PaperXI.HeisenbergTransport.pullback_scalar_action`
170	theorem	`PaperXI.HeisenbergTransport.pullback_central_action`

A.232 `PaperXI.HenselianConductorTransport`

Source lines: 120.

SHA-256: `4195ca7a0238b26ebca438814c59c9712c09d21fd211f8468e3738fadebcb48a`.

```
Conductor invariance for the actual base change to the Hensel-generated field.
The valuation ring is unchanged upstairs. This is not ramified quotient invariance.
```

Direct project imports: `PaperXI.HenselianWeilAttachment`.

Line	Kind	Name
17	def	`PaperXI.HenselianConductorTransport.localLowerEquiv`
31	def	`PaperXI.HenselianConductorTransport.lowerEquiv`
39	theorem	`PaperXI.HenselianConductorTransport.lower_card`
45	theorem	`PaperXI.HenselianConductorTransport.embedding_lift`
58	theorem	`PaperXI.HenselianConductorTransport.fixed_eq`
78	theorem	`PaperXI.HenselianConductorTransport.summand_eq`
87	theorem	`PaperXI.HenselianConductorTransport.swan_eq`
98	theorem	`PaperXI.HenselianConductorTransport.artin_eq`
108	theorem	`PaperXI.HenselianConductorTransport.summable_source`

A.233 `PaperXI.HenselianFieldLift`

Source lines: 95.

SHA-256: `af945609f76e7f115f367f158515f7c5053cb2b0290dbc8baf21c20ea5f12e09`.

```
Construct an actual finite intermediate field from finite residue-field
Hensel lifts. The fixed-field statement concerns literal automorphisms of L.
```

Direct project imports: `PaperXI.HenselianResidueLift`.

Line	Kind	Name
16	def	`PaperXI.HenselianFieldLift.element`
19	def	`PaperXI.HenselianFieldLift.field`
21	theorem	`PaperXI.HenselianFieldLift.element_mem`
24	theorem	`PaperXI.HenselianFieldLift.element_integral`
31	instance	`PaperXI.HenselianFieldLift.finiteDimensional`
38	theorem	`PaperXI.HenselianFieldLift.residue_element`
43	def	`PaperXI.HenselianFieldLift.fixing`
47	theorem	`PaperXI.HenselianFieldLift.mem_fixing_iff`
63	theorem	`PaperXI.HenselianFieldLift.fixing_iff_residue`
81	theorem	`PaperXI.HenselianFieldLift.inertia_le_fixing`
89	theorem	`PaperXI.HenselianFieldLift.contained_in_inertia_fixed`

A.234 `PaperXI.HenselianGaloisResidue`

Source lines: 75.

SHA-256: `4a35a55e8f7a231bdd6f1f2affbe1d44e00c6423dea5d53811260120032d96a0`.

```
Derive residue-automorphism lifting in a finite Galois extension with invariant
Henselian valuation and finite residue field. No surjectivity is assumed.
```

Direct project imports: `PaperXI.ValuationResidueEmbedding`.

Line	Kind	Name
13	def	`PaperXI.HenselianGaloisResidue.decompositionMap`
17	def	`PaperXI.HenselianGaloisResidue.residueAction`
20	theorem	`PaperXI.HenselianGaloisResidue.residueAction_residue`
23	theorem	`PaperXI.HenselianGaloisResidue.fixes_base`
35	theorem	`PaperXI.HenselianGaloisResidue.fixed_iff_base`
59	theorem	`PaperXI.HenselianGaloisResidue.lift_automorphism`

A.235 `PaperXI.HenselianLiftTower`

Source lines: 71.

SHA-256: `3ba24216ac193a2da00a39d6d731e8e750f4d8932c2ea761c52b6de0657f9cc7`.

```
Completion implies the Hensel hypothesis, and the unique lifts respect
finite residue-field towers. Multiplicativity is not additivity.
```

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
12	theorem	`PaperXI.HenselianLiftTower.henselian_of_complete`
24	theorem	`PaperXI.HenselianLiftTower.root_power_tower`
32	theorem	`PaperXI.HenselianLiftTower.lift_tower`
43	theorem	`PaperXI.HenselianLiftTower.lift_equiv`
58	theorem	`PaperXI.HenselianLiftTower.field_tower`

A.236 `PaperXI.HenselianResidueLift`

Source lines: 115.

SHA-256: `52a3d30f9b5773234105b18f548446e88910a055cc08896c493da1c5bed39370`.

```
Unique multiplicative lifts of an embedded finite residue field. These are
actual Hensel roots in the given local ring, not an assumed ring section.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.HenselianResidueLift.poly`
17	theorem	`PaperXI.HenselianResidueLift.poly_monic`
22	theorem	`PaperXI.HenselianResidueLift.card_residue_zero`
26	theorem	`PaperXI.HenselianResidueLift.derivative_unit`
31	theorem	`PaperXI.HenselianResidueLift.exists_lift`
43	theorem	`PaperXI.HenselianResidueLift.unique_lift`
53	def	`PaperXI.HenselianResidueLift.lift`
55	theorem	`PaperXI.HenselianResidueLift.lift_pow`
58	theorem	`PaperXI.HenselianResidueLift.residue_lift`
61	theorem	`PaperXI.HenselianResidueLift.lift_injective`
66	theorem	`PaperXI.HenselianResidueLift.lift_zero`
71	theorem	`PaperXI.HenselianResidueLift.lift_one`
76	theorem	`PaperXI.HenselianResidueLift.lift_mul`

Line	Kind	Name
82	def	PaperXI.HenselianResidueLift.sectionMonoid
88	theorem	PaperXI.HenselianResidueLift.lift_nat_pow
91	theorem	PaperXI.HenselianResidueLift.lift_unit
96	theorem	PaperXI.HenselianResidueLift.automorphism_fix_iff
107	theorem	PaperXI.HenselianResidueLift.automorphism_transport

A.237 PaperXI.HenselianWeilAttachment

Source lines: 140.

SHA-256: af3f6d521f61881ee03e54dcc204f5911df3f47dc1b0201fa77efd687608ba2d.

```
Replace the previous residue stabilizer by the decomposition group over an
actual, constructed finite intermediate field. This does not assert its full
local-field unramified classification or an absolute Weil identification.
```

Direct project imports: PaperXI.HenselianLiftTower, PaperXI.ValuationFixingField, PaperXI.UnramifiedValuation.

Line	Kind	Name
19	abbrev	PaperXI.HenselianWeilAttachment.liftedField
23	theorem	PaperXI.HenselianWeilAttachment.field_fixing_eq
30	def	PaperXI.HenselianWeilAttachment.decompositionEquiv
37	theorem	PaperXI.HenselianWeilAttachment.ring_action
42	theorem	PaperXI.HenselianWeilAttachment.residue_action
50	theorem	PaperXI.HenselianWeilAttachment.residue_hom
58	def	PaperXI.HenselianWeilAttachment.pullbackEquiv
84	def	PaperXI.HenselianWeilAttachment.sourceEquiv
92	def	PaperXI.HenselianWeilAttachment.embedding
100	theorem	PaperXI.HenselianWeilAttachment.embedding_injective
104	theorem	PaperXI.HenselianWeilAttachment.degree_formula
111	theorem	PaperXI.HenselianWeilAttachment.embedding_range
130	theorem	PaperXI.HenselianWeilAttachment.fixes_liftedField_iff

A.238 PaperXI.HolonomyResidual

Source lines: 57.

SHA-256: d3e383de6f107b77fc33648b64a84751caa375bf3e468908ec513f79b1e11d52.

```
Paper XI, Proposition 14.6. The supplied equivalences are actual
root-to-vertex transports. Constructing the graph/tree is a separate obligation.
```

Direct project imports: none.

Line	Kind	Name
14	def	PaperXI.HolonomyResidual.transported
17	theorem	PaperXI.HolonomyResidual.transported_apply
20	def	PaperXI.HolonomyResidual.cycleDefect
27	def	PaperXI.HolonomyResidual.edgeResidual

Line	Kind	Name
34	theorem	`PaperXI.HolonomyResidual.residual_factorization`
43	theorem	`PaperXI.HolonomyResidual.residual_zero_iff`

A.239 `PaperXI.HyperbolicDeterminant`

Source lines: 66.

SHA-256: `36feef061c87d2e39afe6bfc3e8a246b942f70904fff07a77ab5dd5cff6fd275`.

```
Determinants of the literal inverse-dual and hyperbolic difference maps.
All identities are established for the actual endomorphisms.
```

Direct project imports: `PaperXI.HyperbolicSpace`.

Line	Kind	Name
13	theorem	`PaperXI.HyperbolicDeterminant.inverse_difference`
21	theorem	`PaperXI.HyperbolicDeterminant.dual_difference`
33	theorem	`PaperXI.HyperbolicDeterminant.dual_difference_det`
43	theorem	`PaperXI.HyperbolicDeterminant.action_difference`
51	theorem	`PaperXI.HyperbolicDeterminant.total_difference_det`
59	theorem	`PaperXI.HyperbolicDeterminant.action_det`

A.240 `PaperXI.HyperbolicPhase`

Source lines: 56.

SHA-256: `419bca65c3efe41e14cf7793f6d4defd5c05e15201eb00e4161a916f3cb975f6`.

```
Source 12.8 for the literal hyperbolic action. The calculation also holds
for the defined determinant phase in characteristic two; no Weil trace
interpretation or normalization is asserted here.
```

Direct project imports: `PaperXI.HyperbolicDeterminant`, `PaperXI.OddPhase`.

Line	Kind	Name
16	theorem	`PaperXI.HyperbolicPhase.fixedPointFree`
22	theorem	`PaperXI.HyperbolicPhase.source`
47	theorem	`PaperXI.HyperbolicPhase.symplectic_fixedPointFree`

A.241 `PaperXI.HyperbolicSpace`

Source lines: 120.

SHA-256: `1532ea2542009051b6a09e99d9353d460de621615cf408b4dff48af6a072c501`.

```
The actual hyperbolic space U × U*, its evaluation form, and its
contragredient action. No bilinear form or action identities are assumed.
```

Direct project imports: none.

Line	Kind	Name
16	abbrev	`PaperXI.HyperbolicSpace.Space`
19	def	`PaperXI.HyperbolicSpace.form`
39	theorem	`PaperXI.HyperbolicSpace.form_apply`
43	theorem	`PaperXI.HyperbolicSpace.form_isAlt`
48	theorem	`PaperXI.HyperbolicSpace.form_nondegenerate`
72	def	`PaperXI.HyperbolicSpace.action`
75	theorem	`PaperXI.HyperbolicSpace.action_apply`
79	theorem	`PaperXI.HyperbolicSpace.action_form`
85	theorem	`PaperXI.HyperbolicSpace.dual_fixedPointFree`
107	theorem	`PaperXI.HyperbolicSpace.action_fixedPointFree`
115	theorem	`PaperXI.HyperbolicSpace.finrank`

A.242 `PaperXI.InducedClassification`

Source lines: 136.

SHA-256: `5700629a23c61b9f8d38d3ca69ba94c5efdaf0c847404c0c162aa3c7e54321a0`.

```
Classification uses the full subgroup characters, including return values.
No assertion identifies an abstract degree subgroup with a local Weil group.
```

Direct project imports: `PaperXI.InducedDual`, `PaperXI.CyclicInducedWrap`.

Line	Kind	Name
16	theorem	`PaperXI.InducedClassification.index_transitive`
22	theorem	`PaperXI.InducedClassification.index_stabilizer`
29	def	`PaperXI.InducedClassification.conjugateEquiv`
38	theorem	`PaperXI.InducedClassification.coefficient_action`
56	theorem	`PaperXI.InducedClassification.character_of_equiv`
84	theorem	`PaperXI.InducedClassification.equiv_iff`
93	theorem	`PaperXI.InducedClassification.selfDual_iff`
110	def	`PaperXI.InducedClassification.degreeDualEquiv`
120	theorem	`PaperXI.InducedClassification.degreeSelfDual_iff`

A.243 `PaperXI.InducedCycleInvariants`

Source lines: 107.

SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
Frobenius invariants of genuine degree-kernel induction. The basis and its
closing scalar are constructed from the inducing character. No cyclic matrix,
characteristic polynomial or return-map certificate is assumed.
```

Direct project imports: `PaperXI.CyclicInducedWrap`, `PaperXI.CycleCoordinates`.

Line	Kind	Name
18	def	`PaperXI.InducedCycleInvariants.indexEquiv`
25	def	`PaperXI.InducedCycleInvariants.basis`
28	theorem	`PaperXI.InducedCycleInvariants.basis_apply`

Line	Kind	Name
33	def	PaperXI.InducedCycleInvariants.returnScalar
37	theorem	PaperXI.InducedCycleInvariants.dimension
41	theorem	PaperXI.InducedCycleInvariants.step
51	theorem	PaperXI.InducedCycleInvariants.closing_step
61	theorem	PaperXI.InducedCycleInvariants.basis_powers
69	theorem	PaperXI.InducedCycleInvariants.finiteSpace
75	theorem	PaperXI.InducedCycleInvariants.cycle_invariants
83	theorem	PaperXI.InducedCycleInvariants.returnScalar_from_charpoly
89	theorem	PaperXI.InducedCycleInvariants.ambient_polynomial
99	theorem	PaperXI.InducedCycleInvariants.return_operator

A.244 PaperXI.InducedDual

Source lines: 42.

SHA-256: e915fa1d7389f923ae824272ca73b059424964f89c7e46b0aa503af596137995.

```
Duality and the full-character classification for induction from a finite normal quotient.
```

Direct project imports: PaperXI.MonomialDual, PaperXI.CyclicInducedCoordinates, PaperXI.MonomialInductionSubgroup.

Line	Kind	Name
15	def	PaperXI.InducedDual.indexAction
21	theorem	PaperXI.InducedDual.indexAction_apply
26	def	PaperXI.InducedDual.inverseCharacterEquiv

A.245 PaperXI.InducedTameProfile

Source lines: 187.

SHA-256: 76a61c7e77722f469f734ee21b80e55786a2aec4a1916c23c4dc48a92249d576.

```
All inertia and determinant data are read from the actual induced space.
The finite ramification expression is kept distinct from an arithmetically
identified Artin conductor. No irreducibility or regularity is needed here.
```

Direct project imports: PaperXI.InducedCycleInvariants, PaperXI.TameConductor, PaperXI.InvariantPolynomial.

Line	Kind	Name
18	theorem	PaperXI.InducedTameProfile.inertia_degree
33	def	PaperXI.InducedTameProfile.inertiaInDegree
51	theorem	PaperXI.InducedTameProfile.origin_mem
63	theorem	PaperXI.InducedTameProfile.basis_packet
77	theorem	PaperXI.InducedTameProfile.support_of_basis
104	theorem	PaperXI.InducedTameProfile.inertia_support
112	theorem	PaperXI.InducedTameProfile.inertia_invariants
128	theorem	PaperXI.InducedTameProfile.invariant_dimension
149	theorem	PaperXI.InducedTameProfile.local_polynomial

Line	Kind	Name
173	theorem	`PaperXI.InducedTameProfile.finite_conductors`

A.246 `PaperXI.InducedTwist`

Source lines: 103.

SHA-256: `53e7034ad1f3dacf431f99ae0d0334e94296550244011390bb7f137f09a4ebec`.

```
The projection formula on actual induced spaces, with inertia and return scalars.
```

Direct project imports: `PaperXI.CharacterTwist`, `PaperXI.CyclicInducedWrap`.

Line	Kind	Name
14	def	`PaperXI.InducedTwist.character`
16	theorem	`PaperXI.InducedTwist.character_apply`
20	def	`PaperXI.InducedTwist.forward`
31	def	`PaperXI.InducedTwist.backward`
43	def	`PaperXI.InducedTwist.coindLinearEquiv`
52	def	`PaperXI.InducedTwist.coindEquiv`
62	def	`PaperXI.InducedTwist.inducedEquiv`
74	def	`PaperXI.InducedTwist.degreeEquiv`
81	theorem	`PaperXI.InducedTwist.period_value`
88	theorem	`PaperXI.InducedTwist.closing_basis`
97	theorem	`PaperXI.InducedTwist.inertia_unchanged`

A.247 `PaperXI.InducedTypeSpace68`

Source lines: 142.

SHA-256: `fe12e8899453c20460a4e73b84995d88a3551bb34ad86ab9b31fc3b83a89b409`.

```
# The canonical inducing-type subspace of actual induction

The full off-H Mackey condition gives factorization of every H-map through the
literal identity-coset inclusion.  No topological or semisimplicity hypothesis
is needed for this factorization.
```

Direct project imports: `PaperXI.InductionIrreducibility`.

Line	Kind	Name
21	theorem	`PaperXI.InducedTypeSpace68.factorization`
35	theorem	`PaperXI.InducedTypeSpace68.factorization_apply`
45	def	`PaperXI.InducedTypeSpace68.homEquiv`
60	theorem	`PaperXI.InducedTypeSpace68.range_eq_identity`
85	def	`PaperXI.InducedTypeSpace68.typeSpace`
89	def	`PaperXI.InducedTypeSpace68.typeEquiv`
96	def	`PaperXI.InducedTypeSpace68.typeOperators`
99	theorem	`PaperXI.InducedTypeSpace68.typeOperators_mul`
103	theorem	`PaperXI.InducedTypeSpace68.typeOperators_add`
107	theorem	`PaperXI.InducedTypeSpace68.typeOperators_eq_zero_iff`

Line	Kind	Name
112	theorem	PaperXI.InducedTypeSpace68.typeOperators_idempotent_iff
119	theorem	PaperXI.InducedTypeSpace68.type_map_scalar
134	theorem	PaperXI.InducedTypeSpace68.conjugation_eq_of_scalar

A.248 PaperXI.InductionIrreducibility

Source lines: 237.

SHA-256: b099fd6ad458401e85a83ae0289bec6be2de9575eb7b73c496fafb98d13088ed.

```
# Irreducibility of actual subgroup induction from a local coefficient projector

All spaces and actions below are Mathlib's actual `IndV` and `ind`, at arbitrary
subgroup index.  The projection hypothesis is a local group-algebra condition:
each vector's identity-coset coefficient can be isolated by a finite linear
combination of subgroup operators.  The conclusion is actual irreducibility,
not scalar endomorphisms or an assumed simple subrepresentation lattice.
```

Direct project imports: PaperXI.OpenSubgroupSmoothInduction, PaperXI.RepresentationStability, PaperXI.SemisimpleProjection, PaperXI.CompactSmoothSemisimple.

Line	Kind	Name
26	def	PaperXI.InductionIrreducibility.identityInclusion
36	theorem	PaperXI.InductionIrreducibility.evaluate_translate
42	def	PaperXI.InductionIrreducibility.identityEvaluation
53	theorem	PaperXI.InductionIrreducibility.identityEvaluation_inclusion
57	def	PaperXI.InductionIrreducibility.identityProjection
61	theorem	PaperXI.InductionIrreducibility.exists_nonzero_coefficient
72	theorem	PaperXI.InductionIrreducibility.eq_top_of_contains_identity
88	theorem	PaperXI.InductionIrreducibility.irreducible_of_projection
126	theorem	PaperXI.InductionIrreducibility.algebra_action_mem
137	theorem	PaperXI.InductionIrreducibility.irreducible_of_local_groupAlgebra
150	def	PaperXI.InductionIrreducibility.intersectionSubgroup
154	def	PaperXI.InductionIrreducibility.intersectionConjugation
161	def	PaperXI.InductionIrreducibility.coefficientIntertwiner
182	theorem	PaperXI.InductionIrreducibility.intertwiner_eq_zero_of_mackey
205	theorem	PaperXI.InductionIrreducibility.irreducible_of_mackey
221	theorem	PaperXI.InductionIrreducibility.irreducible_of_compact_open_mackey

A.249 PaperXI.InertiaFixed

Source lines: 90.

SHA-256: ac85a017c6c2462073de65b7304984cdc44ecebf7b201d52cad8f4a5e916e967.

```
Actual inertia invariants of a cyclic packet, without irreducibility or
an assumption that its inertia labels are distinct.
```

Direct project imports: PaperXI.PacketReturn.

Line	Kind	Name
13	theorem	`PaperXI.InertiaFixed.invariants_eq_packet_zero`
23	theorem	`PaperXI.InertiaFixed.orbit_label_eq_zero_iff`
32	theorem	`PaperXI.InertiaFixed.invariants_eq_top_of_basis`
40	theorem	`PaperXI.InertiaFixed.invariants_eq_bot_of_basis`
56	theorem	`PaperXI.InertiaFixed.fixed_forward`
63	theorem	`PaperXI.InertiaFixed.fixed_backward`
70	theorem	`PaperXI.InertiaFixed.invariants_of_cycle_basis`

A.250 `PaperXI.InertiaQuotient`

Source lines: 98.

SHA-256: `6864c2943a2417a4883fae946d64fcb26273f3e590700e9ae16b296e8d5361bf`.

```
Invariant-space and restricted-operator comparison for a supplied finite
inertia quotient. No finite quotient is embedded into the original group.
```

Direct project imports: `PaperXI.InteractionPolynomial`.

Line	Kind	Name
13	theorem	`PaperXI.InertiaQuotient.invariants_comp_of_surjective`
23	def	`PaperXI.InertiaQuotient.invariantEquiv`
27	theorem	`PaperXI.InertiaQuotient.invariantEquiv_apply`
34	def	`PaperXI.InertiaQuotient.factoredFrobenius`
42	theorem	`PaperXI.InertiaQuotient.factoredFrobenius_apply`
49	theorem	`PaperXI.InertiaQuotient.factored_polynomial_eq`
58	theorem	`PaperXI.InertiaQuotient.factored_local_polynomial`
70	theorem	`PaperXI.InertiaQuotient.factored_character_trivial_iff`
83	theorem	`PaperXI.InertiaQuotient.factored_determinant`

A.251 `PaperXI.IntegerRingComparison`

Source lines: 230.

SHA-256: `382149600512912093535bbc83e0f1dcd500caa3db84e5340ac368157c74b06d`.

```
# Comparison of the algebraic and spectral integer rings

For the unramified residue extension constructed in `LocalFieldResidueExtension`,
the finite-extension spectral norm has unit ball equal to the image of the
algebraically constructed complete DVR.  The proof proceeds through the exact
integrality criterion over the base local-field integers.
```

Direct project imports: `PaperXI.SpectralLocalField`.

Line	Kind	Name
45	abbrev	`PaperXI.IntegerRingComparison.UpperRing`
46	abbrev	`PaperXI.IntegerRingComparison.UpperField`
50	theorem	`PaperXI.IntegerRingComparison.norm_le_one_iff_isIntegral`
95	theorem	`PaperXI.IntegerRingComparison.isIntegral_iff_exists_upper_lift`
108	theorem	`PaperXI.IntegerRingComparison.upper_integerRing_eq_spectral`

Line	Kind	Name
127	def	PaperXI.IntegerRingComparison.upperRingEquivSpectralIntegers
137	instance	PaperXI.IntegerRingComparison.spectralIntegerAlgebra
143	instance	PaperXI.IntegerRingComparison.spectralIntegerAlgebra_isLocal
155	def	PaperXI.IntegerRingComparison.upperRingAlgEquivSpectralIntegers
159	instance	PaperXI.IntegerRingComparison.spectralIntegersEtale
166	theorem	PaperXI.IntegerRingComparison.upperRingEquivSpectralIntegers_coe
173	theorem	PaperXI.IntegerRingComparison.upper_maximalIdeal_map
182	theorem	PaperXI.IntegerRingComparison.spectral_ramificationIdx_one
188	def	PaperXI.IntegerRingComparison.upperResidueFieldEquivSpectral
193	def	PaperXI.IntegerRingComparison.upperResidueFieldAlgEquivSpectral
200	def	PaperXI.IntegerRingComparison.spectralResidueAlgEquiv
207	def	PaperXI.IntegerRingComparison.spectralResidueEquiv
214	theorem	PaperXI.IntegerRingComparison.spectral_residue_finrank
220	theorem	PaperXI.IntegerRingComparison.upper_degree_eq_spectral_residue_finrank
226	theorem	PaperXI.IntegerRingComparison.natCard_spectralResidue

A.252 PaperXI.InteractionPolynomial

Source lines: 77.

SHA-256: e9129207250ae783e05f622a71a16a8654e22680c10c252bdb028abccec69ec2.

Both local-polynomial branches for a cyclic interaction packet.

Direct project imports: PaperXI.InvariantPolynomial, PaperXI.PacketCycle.

Line	Kind	Name
14	theorem	PaperXI.InteractionPolynomial.character_trivial_iff
24	theorem	PaperXI.InteractionPolynomial.local_polynomial
47	theorem	PaperXI.InteractionPolynomial.local_polynomial_character_form
62	theorem	PaperXI.InteractionPolynomial.group_local_polynomial

A.253 PaperXI.IntersectionModel108

Source lines: 62.

SHA-256: 8f2f518c8309162145d80c2ae8d403d2c1eed20ac545a995985c18af108d6343.

The literal conjugate-intersection operator space and its exact finite
quotient model. No identification of distinct Hom spaces is assumed.

Direct project imports: PaperXI.IntertwiningSpace107, PaperXI.QuotientRepresentation, PaperXI.CharacterPairing108.

Line	Kind	Name
16	def	PaperXI.IntersectionModel108.pairs
20	def	PaperXI.IntersectionModel108.left
23	def	PaperXI.IntersectionModel108.right
27	def	PaperXI.IntersectionModel108.spaceEquiv

Line	Kind	Name
39	def	`PaperXI.IntersectionModel108.pullbackEquiv`
53	theorem	`PaperXI.IntersectionModel108.finrank_model`

A.254 `PaperXI.IntertwiningAlong105`

Source lines: 78.

SHA-256: `e1af42743d7ea4d6cbdbb96cb3d355e1f6502d69367e55041090370ccbea5478`.

```
Intertwining on a literal embedded subgroup, with exact carrier maps and
transport through both compact factors.
```

Direct project imports: none.

Line	Kind	Name
13	theorem	`PaperXI.IntertwiningAlong105.act_mul`
16	theorem	`PaperXI.IntertwiningAlong105.act_inv_cancel`
20	theorem	`PaperXI.IntertwiningAlong105.act_mul_inv_cancel`
25	def	`PaperXI.IntertwiningAlong105.Along`
29	def	`PaperXI.IntertwiningAlong105.ExistsAt`
32	theorem	`PaperXI.IntertwiningAlong105.left_right`
46	theorem	`PaperXI.IntertwiningAlong105.left_right_ne_zero`
56	theorem	`PaperXI.IntertwiningAlong105.existsAt_of_left_right`
62	theorem	`PaperXI.IntertwiningAlong105.remove_central`
72	theorem	`PaperXI.IntertwiningAlong105.existsAt_remove_central`

A.255 `PaperXI.IntertwiningSpace107`

Source lines: 81.

SHA-256: `2c3ea887cc1868b2c539bf9d466778b39567a4ea10aa245876ba0dfe9bef8aa7`.

```
A linear equivalence of actual intertwining spaces on the same embedded
compact group. Identification with a smaller integral subgroup is separate.
```

Direct project imports: `PaperXI.TensorIntertwining107`.

Line	Kind	Name
16	def	`PaperXI.IntertwiningSpace107.space`
33	def	`PaperXI.IntertwiningSpace107.tensorMap`
43	def	`PaperXI.IntertwiningSpace107.equivalence`
54	theorem	`PaperXI.IntertwiningSpace107.finrank_eq`
61	theorem	`PaperXI.IntertwiningSpace107.line_of_dimension`

A.256 `PaperXI.InvariantAssembly`

Source lines: 48.

SHA-256: `2fb39cacd47c40f2a53037ffb5b416bb5f6311285265e3f57aed422baa40e5b4`.

Actual invariant spaces under a supplied equivariant finite decomposition.

Direct project imports: `PaperXI.GroupFixed`.

Line	Kind	Name
13	def	`PaperXI.InvariantAssembly.invariantsEquiv`
28	theorem	`PaperXI.InvariantAssembly.invariantsEquiv_apply`
32	theorem	`PaperXI.InvariantAssembly.finrank_invariants`
39	theorem	`PaperXI.InvariantAssembly.count_invariants`

A.257 `PaperXI.InvariantPolynomial`

Source lines: 90.

SHA-256: `cd9db934f06df326729d69a24446a44a9b6d00c9e5dbf48c6ff8c666e2b36990`.

The local polynomial is attached to Frobenius restricted to the actual
invariant subspace. Its determinant meaning is proved before its evaluation.

Direct project imports: `PaperXI.InertiaFixed`, `PaperXI.CyclicPolynomial`.

Line	Kind	Name
13	def	`PaperXI.InvariantPolynomial.operatorPolynomial`
16	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_eq_det`
22	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_of_subsingleton`
29	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_conj`
38	def	`PaperXI.InvariantPolynomial.fixedFrobenius`
49	theorem	`PaperXI.InvariantPolynomial.fixedFrobenius_apply`
55	def	`PaperXI.InvariantPolynomial.localPolynomial`
60	theorem	`PaperXI.InvariantPolynomial.localPolynomial_eq_det`
68	theorem	`PaperXI.InvariantPolynomial.fixed_conj_of_top`
76	theorem	`PaperXI.InvariantPolynomial.localPolynomial_of_top`
83	theorem	`PaperXI.InvariantPolynomial.localPolynomial_of_bot`

A.258 `PaperXI.IsotypicEvaluation`

Source lines: 97.

SHA-256: `9fa381127f5578659025a1972b5e641f6f3000ff3e6a0d17c4f402b6ec08beaa`.

Direct project imports: none.

Line	Kind	Name
19	def	`PaperXI.IsotypicEvaluation.map`
28	theorem	`PaperXI.IsotypicEvaluation.map_tmul`
31	theorem	`PaperXI.IsotypicEvaluation.bijective_of_copies`
80	theorem	`PaperXI.IsotypicEvaluation.bijective`
87	def	`PaperXI.IsotypicEvaluation.equiv`
92	theorem	`PaperXI.IsotypicEvaluation.equiv_tmul`

A.259 PaperXI.LatticeCongruenceTopology60

Source lines: 144.

SHA-256: `2bdb8a0315e6b0015b8c69df7a1ef382ca5455f3976d5d040c507a94e2736fd0`.

```
# Topology of the full periodic-lattice congruence group

The infinite definition is preserved literally. A base-field scalar period
proves openness by finite reduction, while compactness follows from the zero
lattice stabilizer. The ambient group is actual `GL_d(K)`.
```

Direct project imports: `PaperXI.LatticeMatrixTopology60`, `PaperXI.PeriodicLatticeReduction60`.

Line	Kind	Name
21	def	`PaperXI.LatticeCongruenceTopology60.coordinateAut`
28	theorem	`PaperXI.LatticeCongruenceTopology60.coordinateAut_apply`
33	def	`PaperXI.LatticeCongruenceTopology60.congruence`
38	theorem	`PaperXI.LatticeCongruenceTopology60.mem_stabilizer_iff`
60	def	`PaperXI.LatticeCongruenceTopology60.levelSet`
65	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_eq`
72	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_closed`
89	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_open`
105	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_closed`
120	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_compact`
130	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_open`

A.260 PaperXI.LatticeMatrixTopology60

Source lines: 152.

SHA-256: `c924df503533e0221a502da1cdc8e886d4290c90602f3ee33287cdfb9ef98274`.

```
# Compact open lattice stabilizers in actual matrix coordinates

A finite continuous basis transports a literal compact open additive lattice
into the given local field's matrix group. Compactness uses a basis contained
in the lattice. No equality with a simple-character group is asserted.
```

Direct project imports: `PaperXI.ActualELatticeQuotient59`.

Line	Kind	Name
22	def	`PaperXI.LatticeMatrixTopology60.matrixAction`
26	theorem	`PaperXI.LatticeMatrixTopology60.matrixAction_continuous`
36	def	`PaperXI.LatticeMatrixTopology60.preserving`
45	def	`PaperXI.LatticeMatrixTopology60.stabilizer`
50	theorem	`PaperXI.LatticeMatrixTopology60.preserving_closed`
62	theorem	`PaperXI.LatticeMatrixTopology60.preserving_open`
73	theorem	`PaperXI.LatticeMatrixTopology60.preserving_compact`
92	theorem	`PaperXI.LatticeMatrixTopology60.exists_scaled_basis`
110	theorem	`PaperXI.LatticeMatrixTopology60.preserving_compact_of_open`
131	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_compact`
139	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_open`
145	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_compact_of_open`

A.261 PaperXI.LiftedEtale

Source lines: 95.

SHA-256: `2fa34a8ba6c429ab131c13111dc85621648e62f7ef6edf662e0367ebbddce8bf`.

```
Actual finite-etale and ramification-index endpoints for the Hensel-generated
field over a DVR base. All ring/module properties are derived.
```

Direct project imports: `PaperXI.LiftedValuationFinite`, `PaperXI.LocalResidueRank`.

Line	Kind	Name
20	theorem	`PaperXI.LiftedEtale.free`
28	def	`PaperXI.LiftedEtale.canonicalResidueLinearEquiv`
38	theorem	`PaperXI.LiftedEtale.rank_residue`
45	theorem	`PaperXI.LiftedEtale.maximalIdeal_map`
54	theorem	`PaperXI.LiftedEtale.etale`
65	theorem	`PaperXI.LiftedEtale.ramificationIdx_one`
72	theorem	`PaperXI.LiftedEtale.uniformizer_ideal`
80	theorem	`PaperXI.LiftedEtale.monogenic`

A.262 PaperXI.LiftedFullResidue

Source lines: 115.

SHA-256: `88447e2315204103283102ff48fb70a06eb201867cbbeceaaba3f94325b44618`.

```
The whole residue field of the lifted intermediate field, not merely an
embedded copy of the requested finite field.
```

Direct project imports: `PaperXI.LiftedResidueDegree`.

Line	Kind	Name
15	def	`PaperXI.LiftedFullResidue.valuation`
17	def	`PaperXI.LiftedFullResidue.residueMap`
21	theorem	`PaperXI.LiftedFullResidue.residueMap_contains`
35	theorem	`PaperXI.LiftedFullResidue.residueMap_mem`
56	theorem	`PaperXI.LiftedFullResidue.residueMap_range`
63	def	`PaperXI.LiftedFullResidue.toSubfield`
71	def	`PaperXI.LiftedFullResidue.residueEquiv`
77	theorem	`PaperXI.LiftedFullResidue.residueEquiv_val`
81	theorem	`PaperXI.LiftedFullResidue.residue_card`
85	def	`PaperXI.LiftedFullResidue.baseResidueMap`
89	theorem	`PaperXI.LiftedFullResidue.base_composition`
95	instance	`PaperXI.LiftedFullResidue.actualBaseAlgebra`
99	def	`PaperXI.LiftedFullResidue.residueAlgEquiv`
109	theorem	`PaperXI.LiftedFullResidue.residue_degree`

A.263 PaperXI.LiftedResidueDegree

Source lines: 82.

SHA-256: `d49acefdc58604537e7f2afe7c3ab0f46eba16fb8096a2dc28f1f8a655103bf7`.

Exact degree of the Hensel-generated field in a finite Galois ambient extension.
The base is the full residue field of the restricted valuation.

Direct project imports: `PaperXI.HenselianGaloisResidue`.

Line	Kind	Name
13	abbrev	`PaperXI.LiftedResidueDegree.BaseResidue`
15	instance	`PaperXI.LiftedResidueDegree.baseResidueAlgebra`
20	def	`PaperXI.LiftedResidueDegree.residueAut`
26	theorem	`PaperXI.LiftedResidueDegree.residueAut_apply`
32	theorem	`PaperXI.LiftedResidueDegree.residueAut_surjective`
41	def	`PaperXI.LiftedResidueDegree.field`
46	theorem	`PaperXI.LiftedResidueDegree.fixingSubgroup_eq`
67	theorem	`PaperXI.LiftedResidueDegree.degree`
75	theorem	`PaperXI.LiftedResidueDegree.galois`

A.264 `PaperXI.LiftedResidueFrobenius`

Source lines: 67.

SHA-256: `45e5ccfb6f245b6318a6d628168f8eba77834ea08b554b9f74ba407b49a9f45c`.

Frobenius lifting and the exact degree quotient with the actual base residue.
The degree-one lift is derived from finite Galois residue surjectivity.

Direct project imports: `PaperXI.LiftedFullResidue`, `PaperXI.FiniteResidueBaseChange`.

Line	Kind	Name
14	instance	`PaperXI.LiftedResidueFrobenius.baseFinite`
20	def	`PaperXI.LiftedResidueFrobenius.frobenius`
22	theorem	`PaperXI.LiftedResidueFrobenius.frobenius_inverse_power`
30	theorem	`PaperXI.LiftedResidueFrobenius.frobenius_in_range`
38	theorem	`PaperXI.LiftedResidueFrobenius.decomposition_frobenius_lift`
45	theorem	`PaperXI.LiftedResidueFrobenius.full_degree`
50	def	`PaperXI.LiftedResidueFrobenius.cyclicQuotient`
59	theorem	`PaperXI.LiftedResidueFrobenius.quotient_order`

A.265 `PaperXI.LiftedValuationFinite`

Source lines: 78.

SHA-256: `b4d29f529bdede481493a48e735a4998dd0ff3f401796cc652dbc870a82c2b0e`.

The Hensel-generated field has the actual finite integral valuation ring.

Direct project imports: `PaperXI.ValuationIntegralModel`, `PaperXI.LiftedFullResidue`.

Line	Kind	Name
17	instance	`PaperXI.LiftedValuationFinite.baseAlgebra`
20	instance	`PaperXI.LiftedValuationFinite.baseTower`
23	instance	`PaperXI.LiftedValuationFinite.baseFieldTower`

Line	Kind	Name
26	instance	`PaperXI.LiftedValuationFinite.baseLocal`
34	theorem	`PaperXI.LiftedValuationFinite.stable_valuation`
51	theorem	`PaperXI.LiftedValuationFinite.integralClosure`
58	theorem	`PaperXI.LiftedValuationFinite.finite`
68	theorem	`PaperXI.LiftedValuationFinite.rank`

A.266 `PaperXI.LiftedValuationInertia`

Source lines: 51.

SHA-256: `84e9166787ac68126346acfa087e68e5b5ad325b43fe0b761ac2b001f1ba310b`.

```
The actual residue-action inertia of the generated field is trivial.
This is not an assumed ramification-index certificate.
```

Direct project imports: `PaperXI.LiftedFullResidue`.

Line	Kind	Name
16	theorem	`PaperXI.LiftedValuationInertia.inertia_trivial`

A.267 `PaperXI.LineCharacter`

Source lines: 172.

SHA-256: `fe421a06982cd72d4f80a09db6f3ee52a249a6be9ded4bcd4aa927800c4ed007`.

```
The actual scalar character on the stabilizer of a nonzero line.
```

Direct project imports: `PaperXI.PacketBasis`.

Line	Kind	Name
11	theorem	`PaperXI.LineCharacter.scalar_unique`
16	theorem	`PaperXI.LineCharacter.scalar_ne_zero`
22	def	`PaperXI.LineCharacter.lineStabilizer`
39	theorem	`PaperXI.LineCharacter.mem_iff_preserves_line`
70	def	`PaperXI.LineCharacter.scalar`
73	theorem	`PaperXI.LineCharacter.scalar_spec`
77	def	`PaperXI.LineCharacter.character`
94	theorem	`PaperXI.LineCharacter.character_action`
97	theorem	`PaperXI.LineCharacter.power_action`
109	theorem	`PaperXI.LineCharacter.inertia_mem`
116	theorem	`PaperXI.LineCharacter.return_mem`
123	theorem	`PaperXI.LineCharacter.character_on_inertia`
133	theorem	`PaperXI.LineCharacter.character_on_return`
143	theorem	`PaperXI.LineCharacter.packet_stabilizer_iff`
159	theorem	`PaperXI.LineCharacter.expected_stabilizer_le`

A.268 PaperXI.LocalFactorIdentification

Source lines: 150.

SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
Explicit realization and factor-characterization interfaces for the last
arithmetic step. Realizers need not be unique. Their isomorphism class becomes
unique by equality on a genuinely characterizing test family. No reference
comparison map or final equality is stored in either input interface.
```

Direct project imports: `PaperXI.AtomicReconstruction`, `PaperXI.CharacterProjector`.

Line	Kind	Name
14	structure	`PaperXI.LocalFactorIdentification.FactorSystem`
25	structure	`PaperXI.LocalFactorIdentification.Realization`
32	def	`PaperXI.LocalFactorIdentification.chosenAtom`
34	theorem	`PaperXI.LocalFactorIdentification.chosenAtom_realizes`
37	def	`PaperXI.LocalFactorIdentification.candidate`
40	theorem	`PaperXI.LocalFactorIdentification.class_eq_iff`
49	theorem	`PaperXI.LocalFactorIdentification.candidate_eq_reference`
59	theorem	`PaperXI.LocalFactorIdentification.unique_realized_class`
70	theorem	`PaperXI.LocalFactorIdentification.realization_independence`
81	theorem	`PaperXI.LocalFactorIdentification.candidate_congr`
89	def	`PaperXI.LocalFactorIdentification.onClasses`
95	theorem	`PaperXI.LocalFactorIdentification.constructed_comparison`
105	theorem	`PaperXI.LocalFactorIdentification.reconstructed_comparison`
117	def	`PaperXI.LocalFactorIdentification.representationSetoid`
124	theorem	`PaperXI.LocalFactorIdentification.finite_character_separates`
140	def	`PaperXI.LocalFactorIdentification.finiteCharacterSystem`

A.269 PaperXI.LocalFieldResidueExtension

Source lines: 63.

SHA-256: `e7438b3803d39699e4ac020c70f4f4d2377d99e32af405cdea45fa2eb1c9cb9e`.

```
Specialization at the actual ring of integers of a nonarchimedean local field.
The output ring, its residue and degree are constructed for every positive degree.
Absolute Weil groups and their smooth induction remain separate attachments.
```

Direct project imports: `PaperXI.FiniteFreeCompletion`.

Line	Kind	Name
16	abbrev	`PaperXI.LocalFieldResidueExtension.ring`
17	abbrev	`PaperXI.LocalFieldResidueExtension.field`
19	instance	`PaperXI.LocalFieldResidueExtension.baseAlgebra`
22	instance	`PaperXI.LocalFieldResidueExtension.baseTower`
28	theorem	`PaperXI.LocalFieldResidueExtension.degree`
32	instance	`PaperXI.LocalFieldResidueExtension.finiteExtension`
35	theorem	`PaperXI.LocalFieldResidueExtension.source`
50	abbrev	`PaperXI.LocalFieldResidueExtension.residueExtension`
52	theorem	`PaperXI.LocalFieldResidueExtension.every_positive_degree`

A.270 PaperXI.LocalResidueRank

Source lines: 83.

SHA-256: `04aa1ea55d4707f48f22f293dff3a05669d750c0ecaf1dca52bce090c7553fd3`.

```
An equal-rank residue fiber criterion for actual finite etale local algebras.
```

Direct project imports: none.

Line	Kind	Name
19	theorem	`PaperXI.LocalResidueRank.map_maximal_of_rank`
51	theorem	`PaperXI.LocalResidueRank.formallyUnramified`
57	theorem	`PaperXI.LocalResidueRank.etale`
66	theorem	`PaperXI.LocalResidueRank.etale_iff_rank_residue`
77	theorem	`PaperXI.LocalResidueRank.ramificationIdx_one`

A.271 PaperXI.LocallyFiniteSemisimple

Source lines: 108.

SHA-256: `9d51826538fb9c038407e4a3d741ccbb3ad1c1266fc6642f1e29df6b81cefdd8`.

```
Semisimplicity from finite quotient actions on local invariant subspaces.
The representation may have arbitrary dimension; the local pieces supply the
semisimple submodules whose supremum is the whole group-algebra module.
```

Direct project imports: `PaperXI.QuotientRepresentation`.

Line	Kind	Name
21	theorem	`PaperXI.LocallyFiniteSemisimple.module_of_locally_semisimple`
39	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_comp_iff`
45	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_finite_pullback`
52	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_finite_quotient`
59	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_asSubmodule`
81	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_locally_semisimple`
94	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_locally_finite_quotient`

A.272 PaperXI.MonicResidueAlgebra

Source lines: 87.

SHA-256: `61e77b723a6fc1939e003a5cfd834cb152674a48112bee181e683903cdccdca5`.

```
Monic lifts with irreducible residue polynomial give finite free local
algebras with their literal special fiber. No ambient field is supplied.
```

Direct project imports: none.

Line	Kind	Name
17	abbrev	`PaperXI.MonicResidueAlgebra.specialIdeal`
20	instance	`PaperXI.MonicResidueAlgebra.finite`

Line	Kind	Name
21	instance	`PaperXI.MonicResidueAlgebra.free`
24	def	`PaperXI.MonicResidueAlgebra.specialFiberEquiv`
29	theorem	`PaperXI.MonicResidueAlgebra.specialIdeal_maximal`
34	instance	`PaperXI.MonicResidueAlgebra.localRing`
46	theorem	`PaperXI.MonicResidueAlgebra.maximalIdeal_eq`
49	def	`PaperXI.MonicResidueAlgebra.residueEquiv`
53	theorem	`PaperXI.MonicResidueAlgebra.rank`
56	theorem	`PaperXI.MonicResidueAlgebra.degree_pos`
60	instance	`PaperXI.MonicResidueAlgebra.localHom`
63	theorem	`PaperXI.MonicResidueAlgebra.residueEquiv_mk`
72	def	`PaperXI.MonicResidueAlgebra.residueAlgebraEquiv`
81	theorem	`PaperXI.MonicResidueAlgebra.residue_rank`

A.273 `PaperXI.MonodromyAtom`

Source lines: 117.

SHA-256: `f53d79ddc74c43995955bc45c8bce600d821bb6d118e9426d85e8478c0829bf2`.

```
Literal monodromy blocks underlying the formal reconstruction contract.
For ν(Φ)=q the action on the i-th copy is ν(g)^i ρ(g); the identity
R(g) N R(g)⁻¹ = ν(g)⁻¹ N fixes the normalization explicitly.
No assertion of irreducibility or a decomposition classification is assumed here.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
18	structure	`PaperXI.MonodromyAtom.Datum`
25	def	`PaperXI.MonodromyAtom.action`
40	theorem	`PaperXI.MonodromyAtom.action_apply`
44	def	`PaperXI.MonodromyAtom.monodromy`
46	theorem	`PaperXI.MonodromyAtom.nilpotence`
50	theorem	`PaperXI.MonodromyAtom.commutation`
64	theorem	`PaperXI.MonodromyAtom.covariance`
70	def	`PaperXI.MonodromyAtom.sp`
78	theorem	`PaperXI.MonodromyAtom.dimension`
83	def	`PaperXI.MonodromyAtom.mapEquiv`
97	theorem	`PaperXI.MonodromyAtom.mapEquiv_monodromy`
108	theorem	`PaperXI.MonodromyAtom.scalar_twist`

A.274 `PaperXI.MonodromyChain`

Source lines: 122.

SHA-256: `3cf19949976f59a32001fee38e472b4bf8f9fda9d4d7c67e3819ff2e27ab101c`.

```
Paper XI, Proposition 2.3. Actual operators on a finite chain of copies of W.
The transfers A i are arbitrary linear maps. Nilpotence is a conclusion.
`path` lists all composable transfers in their matrix multiplication order.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.MonodromyChain.shift`
21	theorem	`PaperXI.MonodromyChain.shift_apply`
24	def	`PaperXI.MonodromyChain.path`
28	theorem	`PaperXI.MonodromyChain.path_zero`
29	theorem	`PaperXI.MonodromyChain.path_succ`
33	theorem	`PaperXI.MonodromyChain.pow_apply`
51	theorem	`PaperXI.MonodromyChain.nilpotent_bound`
59	def	`PaperXI.MonodromyChain.projector`
64	theorem	`PaperXI.MonodromyChain.projector_apply`
67	theorem	`PaperXI.MonodromyChain.projector_idempotent`
74	theorem	`PaperXI.MonodromyChain.projector_orthogonal`
83	theorem	`PaperXI.MonodromyChain.projector_sum`
90	theorem	`PaperXI.MonodromyChain.block_vanishes`
109	theorem	`PaperXI.MonodromyChain.block_reconstruction`

A.275 `PaperXI.MonomialAction`

Source lines: 92.

SHA-256: `70169fa35998c99c6d9830b345ee188e42778dace1f244f1181413f956a8961b`.

```
Actual monomial actions on a supplied basis. The scalar cocycle laws are
proved from an honest representation, then reused to construct orbit actions.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.MonomialAction.operator`
18	theorem	`PaperXI.MonomialAction.operator_basis`
22	def	`PaperXI.MonomialAction.representation`
37	theorem	`PaperXI.MonomialAction.representation_basis`
42	theorem	`PaperXI.MonomialAction.scalar_injective`
47	theorem	`PaperXI.MonomialAction.weight_one`
54	theorem	`PaperXI.MonomialAction.weight_mul`
63	theorem	`PaperXI.MonomialAction.representation_eq`
76	def	`PaperXI.MonomialAction.tensorPermutation`
82	theorem	`PaperXI.MonomialAction.tensor_basis`

A.276 `PaperXI.MonomialDual`

Source lines: 46.

SHA-256: `1f8a72858be61f32ca3a9daa195e4e3045e31db32c4be366e1a73c604753e8b8`.

```
The contragredient of a genuine monomial representation on the dual basis.
```

Direct project imports: `PaperXI.MonomialAction`.

Line	Kind	Name
18	theorem	`PaperXI.MonomialDual.inverse_weight`
26	theorem	`PaperXI.MonomialDual.dual_basis_action`

A.277 `PaperXI.MonomialInduction`

Source lines: 152.

SHA-256: `9406f172392d78a75931d9a416f9e701e1dc43b326140a53ab238c582b35813b`.

Actual induction from the stabilizer of a nonzero monomial basis line.

Direct project imports: `PaperXI.LineCharacter`.

Line	Kind	Name
12	def	`PaperXI.MonomialInduction.scalarRepresentation`
17	theorem	`PaperXI.MonomialInduction.scalarRepresentation_apply`
22	abbrev	`PaperXI.MonomialInduction.stabilizer`
23	abbrev	`PaperXI.MonomialInduction.character`
24	abbrev	`PaperXI.MonomialInduction.lineRepresentation`
25	abbrev	`PaperXI.MonomialInduction.induced`
26	abbrev	`PaperXI.MonomialInduction.InducedSpace`
29	def	`PaperXI.MonomialInduction.lineMap`
40	def	`PaperXI.MonomialInduction.evaluation`
52	theorem	`PaperXI.MonomialInduction.evaluation_mk`
58	theorem	`PaperXI.MonomialInduction.mk_stabilizer`
71	def	`PaperXI.MonomialInduction.reconstruction`
75	theorem	`PaperXI.MonomialInduction.reconstruction_basis`
83	theorem	`PaperXI.MonomialInduction.evaluation_reconstruction`
92	theorem	`PaperXI.MonomialInduction.representative_inverse`
100	theorem	`PaperXI.MonomialInduction.reconstruction_evaluation`
133	theorem	`PaperXI.MonomialInduction.evaluation_bijective`
144	def	`PaperXI.MonomialInduction.equivalence`
147	theorem	`PaperXI.MonomialInduction.equivalence_mk`

A.278 `PaperXI.MonomialInductionSubgroup`

Source lines: 69.

SHA-256: `392f827d63ce2b3a4c1ede29d1d61d70de0f705e596b32f42c617d03a6a96910`.

Induction from a specified subgroup and its actual scalar character.

Direct project imports: `PaperXI.MonomialInductionTransitive`.

Line	Kind	Name
12	def	`PaperXI.MonomialInduction.equivalenceOfSubgroup`
31	def	`PaperXI.MonomialInduction.transitiveEquivalenceOfSubgroupAction`
50	def	`PaperXI.MonomialInduction.transitiveEquivalenceOfSubgroup`

A.279 PaperXI.MonomialInductionTransitive

Source lines: 69.

SHA-256: 371cd6e7cd417a0f86606695b9bfa11c152a0a424eb0bf642fcdcb4a0ac04140.

The transitive weighted permutation criterion for actual induced representations.

Direct project imports: PaperXI.MonomialInduction.

Line	Kind	Name
15	theorem	PaperXI.MonomialInduction.mem_lineStabilizer_iff
28	theorem	PaperXI.MonomialInduction.character_eq_weight
38	def	PaperXI.MonomialInduction.representative
40	theorem	PaperXI.MonomialInduction.representative_spec
44	theorem	PaperXI.MonomialInduction.representative_action
50	theorem	PaperXI.MonomialInduction.orbit_is_monomial
56	def	PaperXI.MonomialInduction.transitiveEquivalence
63	theorem	PaperXI.MonomialInduction.transitiveEquivalence_mk

A.280 PaperXI.MonomialOrbit

Source lines: 82.

SHA-256: 7552a5e8d7825191e603a7e2b8ad25bcf97b785a4fc4d3dd7e48cb7cdc0c1734.

Decomposition of an actual monomial representation along actual invariant
basis orbits. The direct sum and its representation equivalence are constructed.

Direct project imports: PaperXI.MonomialAction.

Line	Kind	Name
23	def	PaperXI.MonomialOrbit.block
29	theorem	PaperXI.MonomialOrbit.block_basis
34	theorem	PaperXI.MonomialOrbit.block_single
40	def	PaperXI.MonomialOrbit.sumBasis
43	theorem	PaperXI.MonomialOrbit.sumBasis_apply
50	def	PaperXI.MonomialOrbit.linearEquiv
53	theorem	PaperXI.MonomialOrbit.linearEquiv_basis
59	theorem	PaperXI.MonomialOrbit.intertwines
72	def	PaperXI.MonomialOrbit.representationEquiv
79	theorem	PaperXI.MonomialOrbit.block_finrank

A.281 PaperXI.MoritaActualCartan110

Source lines: 266.

SHA-256: dd18d3a60333f0e737ef4ff22c34db295f49460de701a9fddf9fcd3dc22d293a.

Actual centralizer matrices act in the noncommutative endomorphism ring
through explicit ring homomorphisms.

Direct project imports: `PaperXI.MoritaIndex110`, `PaperXI.ActualLocalCartan106`, `PaperXI.MoritaCartanTransport110`, `PaperXI.MoritaScalarUpgrade110`.

Line	Kind	Name
21	def	`PaperXI.MoritaActualCartan110.centralizerEmbedding`
24	def	`PaperXI.MoritaActualCartan110.scalarEmbedding`
27	def	`PaperXI.MoritaActualCartan110.uniformizerEndUnit`
32	theorem	`PaperXI.MoritaActualCartan110.centralizerEmbedding_integral`
39	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_integralUnit`
54	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_uniformizer`
72	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_normalizes`
93	theorem	`PaperXI.MoritaActualCartan110.centralizerEmbedding_cartan`
127	def	`PaperXI.MoritaActualCartan110.RadicalCommutes`
137	theorem	`PaperXI.MoritaActualCartan110.relIndex_of_actual_cartan`
182	theorem	`PaperXI.MoritaActualCartan110.exists_actual_cartan_index`
206	theorem	`PaperXI.MoritaActualCartan110.exists_actual_cartan_index_of_base_lattice`
232	theorem	`PaperXI.MoritaActualCartan110.relIndex_isEmpty`
246	theorem	`PaperXI.MoritaActualCartan110.relIndex_subsingleton`

A.282 `PaperXI.MoritaCartanTransport110`

Source lines: 121.

SHA-256: `61b021d7f9fd0fecaebb1a9dc3c89a7cbe30760ebecf6120c0af691d1e0e0645`.

```
Actual lattice-index transport under normalizing outer Cartan factors.
The intersection is transported by the additive conjugation equivalence;
relative index is preserved by its proved injectivity.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
16	def	`PaperXI.MoritaCartanTransport110.conjugate`
20	theorem	`PaperXI.MoritaCartanTransport110.unitConjugation_mul`
26	theorem	`PaperXI.MoritaCartanTransport110.unitConjugation_inv_apply_apply`
30	theorem	`PaperXI.MoritaCartanTransport110.conjugate_mul`
36	theorem	`PaperXI.MoritaCartanTransport110.conjugate_eq_of_normalizing`
44	theorem	`PaperXI.MoritaCartanTransport110.conjugate_inf`
50	theorem	`PaperXI.MoritaCartanTransport110.intersection_outer_factors`
58	theorem	`PaperXI.MoritaCartanTransport110.intersection_relIndex_outer_factors`
77	def	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit`
80	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_val`
83	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_inv`
89	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_conjugation_mem`
98	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_mem_normalizingUnits`
110	theorem	`PaperXI.MoritaCartanTransport110.matrix_intersection_relIndex_outer_factors`

A.283 `PaperXI.MoritaCornerRank110`

Source lines: 132.

SHA-256: `8b3f281926d04dbaa4b40d47b74138cda2b422b7f8c4fd68785b65d37586aed2`.

```
Genuine finite/full submodules of the endomorphism ring acquire their
rank from the ambient field extension. No numerical index is a hypothesis.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
22	theorem	`PaperXI.MoritaCornerRank110.exists_basis_full_submodule`
63	def	`PaperXI.MoritaCornerRank110.coefficientMap`
67	def	`PaperXI.MoritaCornerRank110.cornerSubmodule`
79	theorem	`PaperXI.MoritaCornerRank110.cornerSubmodule_eq_map`
92	theorem	`PaperXI.MoritaCornerRank110.corner_fg`
100	theorem	`PaperXI.MoritaCornerRank110.corner_full`
120	theorem	`PaperXI.MoritaCornerRank110.exists_corner_basis_degree`

A.284 `PaperXI.MoritaEndomorphismCoordinates111`

Source lines: 176.

SHA-256: `d0023761aee69e81f53222a6e334e6ac57122a15fd64ce9017ee0bc0e27663c0`.

```
Natural block-endomorphism coordinates for the literal Morita carrier.
No basis of E over F is chosen: the block decomposition uses the finite
product's coordinate inclusions and projections.
```

Direct project imports: `PaperXI.MoritaActualCartan110`, `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
20	def	`PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv`
24	theorem	`PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv_apply`
31	theorem	`PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv_mul`
41	def	`PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv`
45	theorem	`PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv_apply`
58	def	`PaperXI.MoritaEndomorphismCoordinates111.coordinateRingEquiv`
65	def	`PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap`
71	theorem	`PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv_centralizerEmbedding`
82	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinateAut_coordinateUnitMap`
96	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap_centralizer`
115	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinateRingEquiv_unitConjugation`
128	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinate_map_conjugate`
142	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinate_intersection_relIndex`
164	theorem	`PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap_integral`

A.285 `PaperXI.MoritaIndex110`

Source lines: 144.

SHA-256: `30251babfb354b36cc6f28c150e892b0e244e9046f24d606dd8921245105a9d2`.

```
Actual corner power quotients and the diagonal matrix-lattice index.
```

Direct project imports: `PaperXI.MoritaCornerRank110`, `PaperXI.MoritaQuotient110`, `PaperXI.FreeLatticeCard110`, `PaperXI.MoritaUniformizer110`.

Line	Kind	Name
20	theorem	`PaperXI.MoritaIndex110.leftPower_addSubgroupOf_eq`
44	theorem	`PaperXI.MoritaIndex110.card_leftPower_quotient`
69	def	`PaperXI.MoritaIndex110.coefficientUnit`
73	theorem	`PaperXI.MoritaIndex110.coefficientUnit_coe`
78	theorem	`PaperXI.MoritaIndex110.card_diagonal_intersection`
122	theorem	`PaperXI.MoritaIndex110.diagonal_relIndex_of_radical_commutation`

A.286 `PaperXI.MoritaLattice110`

Source lines: 354.

SHA-256: `abdbc4e8506fb153a12c66728668a5c6ff97d94d2f2655dd1ebf49d43f4008d0`.

```
The common-corner and diagonal-transport part of the maximal-order
Morita calculation. The coefficient ring acts through an arbitrary ring
homomorphism into a possibly noncommutative endomorphism ring. In particular,
no false centrality assumption on O_E inside End_F(E) is imposed.
```

Direct project imports: none.

Line	Kind	Name
18	def	`PaperXI.MoritaLattice110.IsMatrixBimodule`
23	def	`PaperXI.MoritaLattice110.cornerEmbedding`
29	def	`PaperXI.MoritaLattice110.corner`
33	theorem	`PaperXI.MoritaLattice110.mem_corner`
37	theorem	`PaperXI.MoritaLattice110.entry_mem_corner`
45	theorem	`PaperXI.MoritaLattice110.single_mem_of_corner`
53	theorem	`PaperXI.MoritaLattice110.eq_matrix_corner`
68	theorem	`PaperXI.MoritaLattice110.corner_bimodule`
82	def	`PaperXI.MoritaLattice110.unitConjugation`
89	theorem	`PaperXI.MoritaLattice110.unitConjugation_apply`
93	def	`PaperXI.MoritaLattice110.normalizingUnits`
106	theorem	`PaperXI.MoritaLattice110.mem_normalizingUnits`
110	theorem	`PaperXI.MoritaLattice110.normalizes_of_left_right_map_eq`
137	theorem	`PaperXI.MoritaLattice110.zpow_normalizes`
142	theorem	`PaperXI.MoritaLattice110.two_sided_mem_iff_left`
159	def	`PaperXI.MoritaLattice110.leftPower`
162	theorem	`PaperXI.MoritaLattice110.mem_leftPower`
177	def	`PaperXI.MoritaLattice110.diagonalUnit`
194	theorem	`PaperXI.MoritaLattice110.diagonal_conjugation_apply`
200	theorem	`PaperXI.MoritaLattice110.scalar_conjugation_single`
208	theorem	`PaperXI.MoritaLattice110.corner_normalizes_of_scalar_commutation`
221	theorem	`PaperXI.MoritaLattice110.mem_conjugate`
232	def	`PaperXI.MoritaLattice110.entrywise`
239	theorem	`PaperXI.MoritaLattice110.mem_entrywise`
244	theorem	`PaperXI.MoritaLattice110.conjugate_eq_entrywise_leftPower`
276	theorem	`PaperXI.MoritaLattice110.intersection_eq_entrywise`
297	theorem	`PaperXI.MoritaLattice110.pow_mul_mem`
304	theorem	`PaperXI.MoritaLattice110.leftPower_zero`
310	theorem	`PaperXI.MoritaLattice110.inf_leftPower_eq_toNat`
331	theorem	`PaperXI.MoritaLattice110.corner_scalar_stable`

Line	Kind	Name
341	theorem	`PaperXI.MoritaLattice110.intersection_eq_nonnegative_entrywise`

A.287 `PaperXI.MoritaPairIndex110`

Source lines: 39.

SHA-256: `1a7eeb3145c18b69770ec5d2b83acd8a4b2d23bb966e4cc47a44655a47472727`.

```
The same actual Cartan witnesses compare two actual lattices. This is an
additive assertion; no canonical H1/J1 or multiplicative attachment is assumed.
```

Direct project imports: `PaperXI.MoritaActualCartan110`.

Line	Kind	Name
20	theorem	`PaperXI.MoritaPairIndex110.actual_pair_index`

A.288 `PaperXI.MoritaQuotient110`

Source lines: 170.

SHA-256: `33d3a455e6a8a202c2ce041e07c1c48e5d82d632a4f4dd114458be95809369e9`.

```
Actual product quotients for the entrywise Morita calculation.
The additive equivalences are constructed from coordinate quotient maps,
their proved kernels, and proved surjectivity. The cardinality conclusions
are consequences of those equivalences, with no supplied index formula.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
16	def	`PaperXI.MoritaQuotient110.coordinateQuotientMap`
22	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_apply`
27	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_surjective`
35	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_ker`
43	def	`PaperXI.MoritaQuotient110.piQuotientEquiv`
48	theorem	`PaperXI.MoritaQuotient110.piQuotientEquiv_mk`
54	theorem	`PaperXI.MoritaQuotient110.card_pi_quotient`
60	def	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap`
66	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_apply`
72	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_surjective`
82	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_ker`
93	def	`PaperXI.MoritaQuotient110.subgroupPiQuotientEquiv`
102	theorem	`PaperXI.MoritaQuotient110.subgroupPiQuotientEquiv_mk`
110	theorem	`PaperXI.MoritaQuotient110.card_subgroup_pi_quotient`
121	def	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap`
127	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_apply`
132	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_surjective`
141	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_ker`
149	def	`PaperXI.MoritaQuotient110.matrixQuotientEquiv`
155	theorem	`PaperXI.MoritaQuotient110.matrixQuotientEquiv_mk`

Line	Kind	Name
162	theorem	`PaperXI.MoritaQuotient110.card_matrix_quotient`

A.289 `PaperXI.MoritaRadicalDictionary110`

Source lines: 132.

SHA-256: `caf45f046cafea555ed4871a05c2baece3d153318b1fced87d701d20622f781f`.

```
The literal additive product with the actual mapped maximal-order radical.
Only mapped coefficient matrices commute with its scalar; no centrality in the
whole endomorphism ring is asserted.
```

Direct project imports: `PaperXI.MoritaActualCartan110`.

Line	Kind	Name
17	def	`PaperXI.MoritaRadicalDictionary110.additiveProduct`
20	theorem	`PaperXI.MoritaRadicalDictionary110.mul_mem_additiveProduct`
30	def	`PaperXI.MoritaRadicalDictionary110.maximalOrderRadical`
34	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_mem_maximalOrderRadical`
42	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_commutes_mapped`
49	theorem	`PaperXI.MoritaRadicalDictionary110.radical_product_eq_left_image`
69	theorem	`PaperXI.MoritaRadicalDictionary110.product_radical_eq_right_image`
90	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_images_eq_of_radical_products_eq`
109	def	`PaperXI.MoritaRadicalDictionary110.actualRadical`
115	theorem	`PaperXI.MoritaRadicalDictionary110.RadicalCommutes_of_actual_radical_product s`

A.290 `PaperXI.MoritaScalarUpgrade110`

Source lines: 55.

SHA-256: `0a05325327c85f7a0de2f900d68524b1a168fd2264a2b5d8565ba75c64356291`.

```
Finite generation passes from a genuine base-integer lattice to the same
carrier with extension-integer scalars.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
17	def	`PaperXI.MoritaScalarUpgrade110.upgrade`
24	theorem	`PaperXI.MoritaScalarUpgrade110.upgrade_restrictScalars`
30	theorem	`PaperXI.MoritaScalarUpgrade110.upgrade_fg`
43	theorem	`PaperXI.MoritaScalarUpgrade110.scalar_stable_of_matrix_bimodule`

A.291 `PaperXI.MoritaUniformizer110`

Source lines: 62.

SHA-256: `b8f0b8cf79c2e58d57e03edd7ec29f81f51c91f1a0a837045514ee4ccf1c126e`.

```
Uniformizer arithmetic in the actual integer ring of a local field.
An irreducible integer-ring element generates the actual maximal ideal,
so primality, nonzeroness, and residue size are derived rather than supplied.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.MoritaUniformizer110.uniformizerIdeal`
20	theorem	`PaperXI.MoritaUniformizer110.exists_uniformizer`
24	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_eq_maximalIdeal`
27	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_ne_bot`
32	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_isMaximal`
38	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_isPrime`
43	def	`PaperXI.MoritaUniformizer110.uniformizerResidueEquiv`
47	theorem	`PaperXI.MoritaUniformizer110.uniformizerResidueEquiv_mk`
53	theorem	`PaperXI.MoritaUniformizer110.uniformizer_quotient_finite`
58	theorem	`PaperXI.MoritaUniformizer110.card_uniformizer_quotient`

A.292 `PaperXI.MovingPhase`

Source lines: 87.

SHA-256: `c01ee25e1c3e1f60b027ae5a5f09f85341892753a59e2eae1257490b80f8ec52`.

```
The determinant phase on the actual moving image, with its source
validity, conjugacy invariance and square-class comparison.
```

Direct project imports: `PaperXI.FixedMovingSource`, `PaperXI.OddPhaseTransport`.

Line	Kind	Name
16	def	`PaperXI.MovingPhase.phase`
18	theorem	`PaperXI.MovingPhase.phase_formula`
23	theorem	`PaperXI.MovingPhase.determinant_ne_zero`
28	theorem	`PaperXI.MovingPhase.phase_sq_one`
31	theorem	`PaperXI.MovingPhase.phase_sign`
37	theorem	`PaperXI.MovingPhase.source_valid`
51	theorem	`PaperXI.MovingPhase.conjugacy`
58	theorem	`PaperXI.MovingPhase.square_class`
66	theorem	`PaperXI.MovingPhase.source_square_class`
80	theorem	`PaperXI.MovingPhase.phase_identity`

A.293 `PaperXI.NormalInertia`

Source lines: 85.

SHA-256: `bc52e7a907e034f8583c73eb69d3de4b3f0575f0ce1174605d50cce3864ab80a`.

```
Normal-inertia fixed spaces and abelian quotient descent.
No cyclic finite-inertia generator, Frobenius orbit basis, or tame hypothesis
is required for the fixed-space vanishing theorem.
```

Direct project imports: `PaperXI.QuotientRepresentation`.

Line	Kind	Name
16	def	PaperXI.NormalInertia.fixedSubrepresentation
22	theorem	PaperXI.NormalInertia.fixed_eq_bot_or_top
30	theorem	PaperXI.NormalInertia.fixed_top_iff_le_kernel
46	theorem	PaperXI.NormalInertia.dimension_one_of_inertia_trivial
57	theorem	PaperXI.NormalInertia.fixed_eq_bot_of_dimension_gt_one
68	theorem	PaperXI.NormalInertia.fixed_eq_bot_of_not_le_kernel
76	theorem	PaperXI.NormalInertia.fixed_codimension

A.294 PaperXI.NormalizedFrobeniusQuotient

Source lines: 120.

SHA-256: 917cfa62d1b914d7d47a3c31f682a52d3338dc040a1606450f1cff71ca075b1f.

```
# The field-normalized Frobenius pullback and its inertia quotient

All valuation data below come from the same finite Galois spectral extension.
The degree-one element and surjectivity of degree are derived, not certificate
fields. The quotient remains a bounded pullback model, not an absolute Weil group.
```

Direct project imports: PaperXI.SpectralGeometricFrobenius, PaperXI.RamifiedUnramifiedQuotient.

Line	Kind	Name
22	abbrev	PaperXI.NormalizedFrobeniusQuotient.Decomposition
23	abbrev	PaperXI.NormalizedFrobeniusQuotient.residueHom
25	abbrev	PaperXI.NormalizedFrobeniusQuotient.Weil
26	abbrev	PaperXI.NormalizedFrobeniusQuotient.degree
28	abbrev	PaperXI.NormalizedFrobeniusQuotient.Inertia
31	def	PaperXI.NormalizedFrobeniusQuotient.inertiaEquiv
37	def	PaperXI.NormalizedFrobeniusQuotient.decompositionLift
40	theorem	PaperXI.NormalizedFrobeniusQuotient.residue_decompositionLift
45	theorem	PaperXI.NormalizedFrobeniusQuotient.frobenius_mem_range
49	def	PaperXI.NormalizedFrobeniusQuotient.weilGeometricLift
55	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_geometricLift
59	theorem	PaperXI.NormalizedFrobeniusQuotient.lift_inverse_power
65	def	PaperXI.NormalizedFrobeniusQuotient.weilArithmeticLift
68	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_arithmeticLift
74	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_surjective
79	def	PaperXI.NormalizedFrobeniusQuotient.quotientEquivIntegerDegree
84	theorem	PaperXI.NormalizedFrobeniusQuotient.quotientEquivIntegerDegree_mk
90	theorem	PaperXI.NormalizedFrobeniusQuotient.quotient_geometricLift
102	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_one_action_eq_on_invariants

A.295 PaperXI.NormalizerDictionary110

Source lines: 432.

SHA-256: 19839adda13ba916ac5dbe60982bf13e65ce22a746ae2b2a63ca3cc54e4fe9e4.

```
Conditional character-normalizer dictionary, followed by literal instances.
No canonical simple-stratum group is introduced or identified by assumption.
```

Direct project imports: `PaperXI.ActualCharacterIdentification61`, `PaperXI.ActualCharacterResidue61`, `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
16	theorem	`PaperXI.NormalizerDictionary110.inf_eq_of_support`
37	theorem	`PaperXI.NormalizerDictionary110.compact_eq_sup_of_support`
61	theorem	`PaperXI.NormalizerDictionary110.compact_le_normalizer_inf`
66	theorem	`PaperXI.NormalizerDictionary110.compact_le_normalizer_of_inf_eq`
73	theorem	`PaperXI.NormalizerDictionary110.mem_compact_iff_product`
123	theorem	`PaperXI.NormalizerDictionary110.chainGroup_le_normalizer_congruence`
136	theorem	`PaperXI.NormalizerDictionary110.firstSubgroup_ambient_image`
148	theorem	`PaperXI.NormalizerDictionary110.firstSubgroup_embedding_range`
157	theorem	`PaperXI.NormalizerDictionary110.firstSubgroup_normal`
169	def	`PaperXI.NormalizerDictionary110.integralCentralizer`
174	theorem	`PaperXI.NormalizerDictionary110.matrixType_eq_integral_sup`
182	theorem	`PaperXI.NormalizerDictionary110.integralCentralizer_le_chainGroup`
193	theorem	`PaperXI.NormalizerDictionary110.integralCentralizer_le_character_normalizer`
209	theorem	`PaperXI.NormalizerDictionary110.restrict_mem_chain_iff`
234	theorem	`PaperXI.NormalizerDictionary110.centralizer_inf_chain`
271	theorem	`PaperXI.NormalizerDictionary110.centralizer_inf_congruence`
318	theorem	`PaperXI.NormalizerDictionary110.actual_first_eq`
329	theorem	`PaperXI.NormalizerDictionary110.actual_first_subgroup_eq`
340	theorem	`PaperXI.NormalizerDictionary110.actual_compact_eq_sup`
351	theorem	`PaperXI.NormalizerDictionary110.actual_compact_product`
367	theorem	`PaperXI.NormalizerDictionary110.actual_model_le`
378	theorem	`PaperXI.NormalizerDictionary110.actual_model_from_centralizer`
395	theorem	`PaperXI.NormalizerDictionary110.actual_first_from_residue`
411	theorem	`PaperXI.NormalizerDictionary110.character_support_compact`

A.296 `PaperXI.NormalizerExtensionInducing58`

Source lines: 272.

SHA-256: `e30d543d65b9008eaf4b66affdcd66e745a09db8651881df7678fda5e84c4b79`.

```
# Extension across a noncentral normalizing cyclic direction

A single conjugation intertwiner propagates to all integer powers. Unique
semidirect coordinates then define an action on the literal generated subgroup.
No centrality assumption is made on the adjoined ambient group element.
```

Direct project imports: `PaperXI.CentralExtensionInducing`, `PaperXI.QuotientRepresentation`.

Line	Kind	Name
22	def	`PaperXI.NormalizerExtensionInducing58.compatiblePairs`
40	theorem	`PaperXI.NormalizerExtensionInducing58.covariance_zpow`
53	def	`PaperXI.NormalizerExtensionInducing58.conjugation`
56	def	`PaperXI.NormalizerExtensionInducing58.conjugationAction`
59	theorem	`PaperXI.NormalizerExtensionInducing58.conjugation_coe`
62	theorem	`PaperXI.NormalizerExtensionInducing58.conjugationAction_coe`
69	def	`PaperXI.NormalizerExtensionInducing58.normalFormHom`
75	theorem	`PaperXI.NormalizerExtensionInducing58.normalFormHom_apply`

Line	Kind	Name
80	def	PaperXI.NormalizerExtensionInducing58.extendedSubgroup
82	theorem	PaperXI.NormalizerExtensionInducing58.le_extendedSubgroup
86	theorem	PaperXI.NormalizerExtensionInducing58.generator_mem
89	theorem	PaperXI.NormalizerExtensionInducing58.extendedSubgroup_eq_sup
97	theorem	PaperXI.NormalizerExtensionInducing58.normalFormHom_injective
112	def	PaperXI.NormalizerExtensionInducing58.normalFormEquiv
115	theorem	PaperXI.NormalizerExtensionInducing58.normalFormEquiv_coe
122	def	PaperXI.NormalizerExtensionInducing58.subgroupInclusion
126	def	PaperXI.NormalizerExtensionInducing58.generator
133	def	PaperXI.NormalizerExtensionInducing58.semidirectRepresentation
143	def	PaperXI.NormalizerExtensionInducing58.extendedRepresentation
147	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_normalForm
155	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_restrict
163	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_generator
172	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_irreducible
183	theorem	PaperXI.NormalizerExtensionInducing58.equiv_unit_covariance
200	theorem	PaperXI.NormalizerExtensionInducing58.intertwiner_bijective
209	def	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv
216	theorem	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv_restrict
222	theorem	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv_generator
231	theorem	PaperXI.NormalizerExtensionInducing58.exists_extension_of_nonzero
249	theorem	PaperXI.NormalizerExtensionInducing58.extendedSubgroup_isOpen
254	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_smooth

A.297 PaperXI.NormalizerIntertwining60

Source lines: 111.

SHA-256: 5993f730521e96961b4fed40e9a77796d990e2a0cccd720874c20680873c2c7f.

```
# Actual intersection intertwining at a normalizer

The intertwining locus is defined by nonzero maps on the literal intersection
subgroup. At a normalizing element such a map is a whole-subgroup intertwiner.
An inclusion of the actual character intertwining locus in that of a prescribed
representation supplies the required map, without a determinant or residue-size
condition. The inclusion is an input; no simple-character theorem is asserted.
```

Direct project imports: PaperXI.InductionIrreducibility, PaperXI.NormalizerExtensionInducing58, PaperXI.MonomialInduction.

Line	Kind	Name
23	def	PaperXI.NormalizerIntertwining60.IntertwinedAt
31	def	PaperXI.NormalizerIntertwining60.intersectionInclusion
37	def	PaperXI.NormalizerIntertwining60.normalizerIntertwiner
44	theorem	PaperXI.NormalizerIntertwining60.normalizerIntertwiner_apply
49	theorem	PaperXI.NormalizerIntertwining60.normalizerIntertwiner_ne_zero
60	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerIntertwiner
67	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerEquiv
76	theorem	PaperXI.NormalizerIntertwining60.character_intertwinedAt
101	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerIntertwiner_of_character

A.298 PaperXI.OddEnvelopeHeisenberg

Source lines: 52.

SHA-256: `0ff24e111c1739d834a0e471bac34b82c9da25eb040e12f9cea2f9cf3b1d1465`.

```
Canonical odd-characteristic extension, constructed directly on the given
alternating space. No polarization, symplectic basis or pre-existing group is input.
```

Direct project imports: `PaperXI.BilinearHeisenberg`, `PaperXI.HeisenbergPrime`.

Line	Kind	Name
13	def	`PaperXI.OddEnvelopeHeisenberg.factor`
15	theorem	`PaperXI.OddEnvelopeHeisenberg.factor_apply`
19	theorem	`PaperXI.OddEnvelopeHeisenberg.factor_commutator`
33	def	`PaperXI.OddEnvelopeHeisenberg.isometryEquiv`
46	theorem	`PaperXI.OddEnvelopeHeisenberg.isometryEquiv_center`

A.299 PaperXI.OddPhase

Source lines: 89.

SHA-256: `a78d2e7c2db5c75599339479441e4e0b95181ec11ced910642a239377291249c`.

```
Determinant-defined finite-field phases. These declarations do not identify
the phase with a Weil trace or assume any Weil normalization theorem.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.OddPhase.difference`
17	theorem	`PaperXI.OddPhase.difference_apply`
21	def	`PaperXI.OddPhase.FixedPointFree`
23	theorem	`PaperXI.OddPhase.fixedPointFree_iff_ker`
28	theorem	`PaperXI.OddPhase.fixedPointFree_iff_injective`
35	theorem	`PaperXI.OddPhase.fixedPointFree_iff_bijective`
40	theorem	`PaperXI.OddPhase.fixedPointFree_iff_det_ne_zero`
45	def	`PaperXI.OddPhase.differenceEquiv`
48	theorem	`PaperXI.OddPhase.differenceEquiv_apply`
52	def	`PaperXI.OddPhase.phase`
56	theorem	`PaperXI.OddPhase.phase_sq_one`
61	theorem	`PaperXI.OddPhase.phase_sign`
64	theorem	`PaperXI.OddPhase.phase_ne_zero`
69	theorem	`PaperXI.OddPhase.phase_zero_space`
74	theorem	`PaperXI.OddPhase.fixedPointFree_zero_space`
80	theorem	`PaperXI.OddPhase.phase_eq_zero_iff`

A.300 PaperXI.OddPhaseCertificate

Source lines: 61.

SHA-256: `783d84147c4f09cbca432ad799b1e57d9bd47da13f3841e92d4f071d49709f2e`.

The actual characteristic-polynomial certificate for the determinant phase.
The source specialization derives even dimension from the supplied form.

Direct project imports: `PaperXI.OddPhase`, `PaperXI.OddSymplectic`.

Line	Kind	Name
16	theorem	`PaperXI.OddPhase.charpoly_eval_one`
24	theorem	`PaperXI.OddPhase.determinant_neg_difference`
30	theorem	`PaperXI.OddPhase.characteristic_determinant`
36	theorem	`PaperXI.OddPhase.characteristic_certificate_of_even`
46	theorem	`PaperXI.OddPhase.characteristic_certificate`
55	theorem	`PaperXI.OddPhase.characteristic_eval_one_ne_zero`

A.301 `PaperXI.OddPhaseSum`

Source lines: 92.

SHA-256: `d97a3ea60f4bcb92aedcc1ccd4ba6b44c70ccfdf1ab43161b9c4194ad26867d1`.

Finite-product laws for the actual determinant phases.

Direct project imports: `PaperXI.FiniteProductDeterminant`, `PaperXI.OddPhaseTransport`, `PaperXI.OddSymplectic`.

Line	Kind	Name
16	def	`PaperXI.OddPhaseSum.action`
20	theorem	`PaperXI.OddPhaseSum.action_apply`
24	theorem	`PaperXI.OddPhaseSum.difference`
31	theorem	`PaperXI.OddPhaseSum.determinant`
36	theorem	`PaperXI.OddPhaseSum.fixedPointFree`
42	theorem	`PaperXI.OddPhaseSum.half_dimension`
58	theorem	`PaperXI.OddPhaseSum.phase_product`
65	theorem	`PaperXI.OddPhaseSum.phase_product_symplectic`
75	theorem	`PaperXI.OddPhaseSum.of_decomposition`
86	theorem	`PaperXI.OddPhaseSum.fixedPointFree_of_decomposition`

A.302 `PaperXI.OddPhaseTransport`

Source lines: 72.

SHA-256: `97c339e43d2c59d8b62995b29c986d88df5214c3df3ae472fd5249ad2ef9ba19`.

Actual conjugacy and determinant-square comparisons of the determinant phase.

Direct project imports: `PaperXI.OddPhase`.

Line	Kind	Name
13	theorem	`PaperXI.OddPhase.difference_intertwines`
18	theorem	`PaperXI.OddPhase.difference_conjugacy`
26	theorem	`PaperXI.OddPhase.fixedPointFree_conjugacy`

Line	Kind	Name
42	theorem	`PaperXI.OddPhase.determinant_conjugacy`
50	theorem	`PaperXI.OddPhase.conjugacy`
56	theorem	`PaperXI.OddPhase.square_class`
64	theorem	`PaperXI.OddPhase.square_class_of_exists`

A.303 `PaperXI.OddSymplectic`

Source lines: 70.

SHA-256: `e1b15050070b85893ab0d094b3ffaf7aaed326673ec11418c39d52487354e063`.

```
Even dimension of the actual alternating nondegenerate form in odd
characteristic. The determinant argument uses its matrix in a constructed
finite basis and does not assume symplectic coordinates or evenness.
```

Direct project imports: none.

Line	Kind	Name
17	theorem	`PaperXI.OddSymplectic.transpose_toMatrix`
26	theorem	`PaperXI.OddSymplectic.even_finrank`
44	theorem	`PaperXI.OddSymplectic.finrank_eq_two_mul_half`
51	theorem	`PaperXI.OddSymplectic.two_ne_zero`
59	theorem	`PaperXI.OddSymplectic.even_finrank_zmod`
64	theorem	`PaperXI.OddSymplectic.finrank_eq_two_mul_half_zmod`

A.304 `PaperXI.OnePlusIndex110`

Source lines: 87.

SHA-256: `733e4888e34cb2eec83dd8b0d29ea395d269e9caa209f00d1fb751c61563ef49`.

```
Exact multiplicative/additive coset comparison. The named unit subgroup K
must really consist of all units 1+x for x in L. This does not pretend that
an arbitrary full lattice contained in a radical automatically is a group.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.OnePlusIndex110.congruence`
35	def	`PaperXI.OnePlusIndex110.offsetEquiv`
52	theorem	`PaperXI.OnePlusIndex110.inv_mul_sub_one_mem_iff`
67	def	`PaperXI.OnePlusIndex110.cosetEquiv`
81	theorem	`PaperXI.OnePlusIndex110.index_eq_additive_relIndex`

A.305 `PaperXI.OnePlusIntersection110`

Source lines: 197.

SHA-256: `512e3b7a894d7db7474f8586c4c652dddd45bfd98197f637f7be8e5a8fec3e7e`.

```
Common-ideal quotient towers for exact one-plus groups. The primary
statement stays in the original integral-order ring and only needs nested
unit groups. It does not assume that a conjugate lattice is an ideal in the
full endomorphism algebra, or that a common integral order exists.
```

Direct project imports: `PaperXI.OnePlusIndex110`.

Line	Kind	Name
15	def	`PaperXI.OnePlusIntersection110.idealUnits`
18	theorem	`PaperXI.OnePlusIntersection110.mem_idealUnits`
21	theorem	`PaperXI.OnePlusIntersection110.mem_of_offset`
32	theorem	`PaperXI.OnePlusIntersection110.idealUnits_le`
38	theorem	`PaperXI.OnePlusIntersection110.idealUnits_relIndex`
50	theorem	`PaperXI.OnePlusIntersection110.offset_le_of_group_le`
62	theorem	`PaperXI.OnePlusIntersection110.nested_relIndex_eq`
90	def	`PaperXI.OnePlusIntersection110.innerRing`
99	theorem	`PaperXI.OnePlusIntersection110.mapped_offsets`
111	theorem	`PaperXI.OnePlusIntersection110.mapped_onePlus`
124	def	`PaperXI.OnePlusIntersection110.pullbackGroup`
127	def	`PaperXI.OnePlusIntersection110.pullbackLattice`
130	theorem	`PaperXI.OnePlusIntersection110.pullback_offsets`
143	theorem	`PaperXI.OnePlusIntersection110.pullback_onePlus`
161	theorem	`PaperXI.OnePlusIntersection110.pullback_intersection_relIndex_eq`
179	theorem	`PaperXI.OnePlusIntersection110.conjugate_intersection_relIndex_eq`

A.306 `PaperXI.OpenSubgroupSmoothInduction`

Source lines: 409.

SHA-256: `eec2e1f81ea1ed3ed0161aa0df8518115a4605b351d08127ef4282943d5978f7`.

```
# Smooth induction from an open subgroup, with no finite-index hypothesis

The target is the actual algebraic induced representation `Representation.ind`.
For an open subgroup and a smooth inducing action, its finite-sum vectors are
smooth even when the subgroup has infinite index. This is an existence step
towards compact induction, not an irreducibility or local-factor theorem.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`.

Line	Kind	Name
25	def	`PaperXI.OpenSubgroupSmoothInduction.smoothSubmodule`
48	theorem	`PaperXI.OpenSubgroupSmoothInduction.mem_smoothSubmodule`
57	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_subgroup`
65	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_smooth`
84	theorem	`PaperXI.OpenSubgroupSmoothInduction.ind_isSmooth`
100	def	`PaperXI.OpenSubgroupSmoothInduction.functionModel`
105	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_mk`
114	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_mk_self`
122	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_injective`
129	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_ne_zero`
136	theorem	`PaperXI.OpenSubgroupSmoothInduction.ind_nontrivial`
140	def	`PaperXI.OpenSubgroupSmoothInduction.finiteCosetSupport`
172	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_finite_support`

Line	Kind	Name
194	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_finite_quotient_support`
205	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_compact_quotient_support`
212	def	`PaperXI.OpenSubgroupSmoothInduction.functionIntertwiner`
225	def	`PaperXI.OpenSubgroupSmoothInduction.evaluate`
229	theorem	`PaperXI.OpenSubgroupSmoothInduction.evaluate_mk_self`
233	theorem	`PaperXI.OpenSubgroupSmoothInduction.evaluate_mk_of_not_rel`
243	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_linearIndependent`
264	def	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent`
274	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_mk`
279	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_add`
286	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_smul`
293	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_finite`
309	def	`PaperXI.OpenSubgroupSmoothInduction.reconstruct`
322	def	`PaperXI.OpenSubgroupSmoothInduction.compactModel`
326	theorem	`PaperXI.OpenSubgroupSmoothInduction.reconstruct_compactModel`
342	theorem	`PaperXI.OpenSubgroupSmoothInduction.compactModel_reconstruct`
358	def	`PaperXI.OpenSubgroupSmoothInduction.compactModelEquiv`
364	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_injective`
371	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_range`
382	theorem	`PaperXI.OpenSubgroupSmoothInduction.finite_functions_smooth`

A.307 `PaperXI.OrbitSpan`

Source lines: 86.

SHA-256: `1c23d6d51d10af04e9788052c0601a726639e8d6938f47b653949b20a8bdd0c3`.

```
The span of a Frobenius orbit of actual character spaces. Its stability and
separation from other packets are proved, not assumed.
```

Direct project imports: `PaperXI.FrobeniusOrbit`.

Line	Kind	Name
12	def	`PaperXI.OrbitSpan.orbitSpan`
16	theorem	`PaperXI.OrbitSpan.packet_le`
22	theorem	`PaperXI.OrbitSpan.self_packet_le`
26	theorem	`PaperXI.OrbitSpan.map_mem`
36	theorem	`PaperXI.OrbitSpan.inertia_stable`
42	theorem	`PaperXI.OrbitSpan.forward_stable`
49	theorem	`PaperXI.OrbitSpan.backward_stable`
56	theorem	`PaperXI.OrbitSpan.outside_projector_zero`
66	theorem	`PaperXI.OrbitSpan.span_ne_bot`
71	theorem	`PaperXI.OrbitSpan.outside_packet_eq_bot`
80	theorem	`PaperXI.OrbitSpan.related_of_span_top`

A.308 `PaperXI.OrthogonalPhase`

Source lines: 111.

SHA-256: `f822d2a61741660dd23d656ef3748309c17ca6e982ee0b05bb69f6b06e8edeb0`.

Source 12.4 for an actual finite internal orthogonal decomposition.

Direct project imports: `PaperXI.OddPhaseSum`, `PaperXI.OrthogonalSummands`.

Line	Kind	Name
14	def	`PaperXI.OrthogonalPhase.decomposition`
19	theorem	`PaperXI.OrthogonalPhase.decomposition_sum`
47	def	`PaperXI.OrthogonalPhase.restricted`
52	theorem	`PaperXI.OrthogonalPhase.restricted_coe`
56	theorem	`PaperXI.OrthogonalPhase.intertwines`
68	theorem	`PaperXI.OrthogonalPhase.phase_product`
81	theorem	`PaperXI.OrthogonalPhase.fixedPointFree`
91	theorem	`PaperXI.OrthogonalPhase.source`

A.309 `PaperXI.OrthogonalSummands`

Source lines: 120.

SHA-256: `fa9f0303b80e64cd56907157d0dc0c86d1f3e34d1ebd579bf0614a9c89116898`.

Actual restricted forms and automorphisms on an internal orthogonal family. Nondegeneracy is derived from orthogonality and spanning. Inverse invariance is derived from finite-dimensionality and forward invariance.

Direct project imports: `PaperXI.OddSymplectic`.

Line	Kind	Name
17	theorem	`PaperXI.OrthogonalSummands.restricted_isAlt`
22	theorem	`PaperXI.OrthogonalSummands.restricted_form_nondegenerate`
58	theorem	`PaperXI.OrthogonalSummands.iSup_eq_top_of_equiv_sum`
71	def	`PaperXI.OrthogonalSummands.restrictionEquiv`
78	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_coe`
82	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_symm_coe`
91	theorem	`PaperXI.OrthogonalSummands.inverse_mem`
97	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_form`
104	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_fixed_eq_zero`
112	theorem	`PaperXI.OrthogonalSummands.restricted_even_finrank`

A.310 `PaperXI.PacketBasis`

Source lines: 97.

SHA-256: `c5f5d83a01759c8aa3984c2b11462888159937d7a2fd4c3f407cc7495a5aee45`.

The finite Frobenius orbit is a proved basis, giving the exact ambient dimension and the action of inertia and Frobenius on the basis vectors.

Direct project imports: `PaperXI.PacketLines`.

Line	Kind	Name
16	theorem	`PaperXI.PacketBasis.cycleVector_ne_zero`

Line	Kind	Name
22	theorem	`PaperXI.PacketBasis.cycle_linearIndependent`
39	def	`PaperXI.PacketBasis.orbitBasis`
52	theorem	`PaperXI.PacketBasis.orbitBasis_apply`
63	theorem	`PaperXI.PacketBasis.finrank_eq_period_of_return`
74	theorem	`PaperXI.PacketBasis.finrank_eq_period`
86	theorem	`PaperXI.PacketBasis.finrank_eq_orbit_card`

A.311 `PaperXI.PacketCycle`

Source lines: 76.

SHA-256: `55a96ecf51c156847ccfe2269e6caecfdfb45dbb4792983af8cb809e3a438bc8`.

```
Cycle invariants of the actual irreducible tame algebraic group action.
```

Direct project imports: `PaperXI.CycleCoordinates`, `PaperXI.PacketBasis`.

Line	Kind	Name
13	theorem	`PaperXI.PacketCycle.linear_power`
32	theorem	`PaperXI.PacketCycle.charpoly`
43	theorem	`PaperXI.PacketCycle.det`
54	theorem	`PaperXI.PacketCycle.ambient_polynomial`
66	theorem	`PaperXI.PacketCycle.exists_cycle_invariants`

A.312 `PaperXI.PacketLines`

Source lines: 147.

SHA-256: `a7c34f355ec41735f4fe274cde83c52d9d91c22db3eabb38d6d7de1c7222a4dd`.

```
A return eigenline exhausts its packet in an actual irreducible
representation. No packet dimension or spanning conclusion is assumed.
```

Direct project imports: `PaperXI.CycleSpan`.

Line	Kind	Name
13	def	`PaperXI.PacketLines.orbitLabel`
16	theorem	`PaperXI.PacketLines.orbitLabel_injective`
22	theorem	`PaperXI.PacketLines.cycle_mem_packet`
29	theorem	`PaperXI.PacketLines.inertia_cycle`
38	theorem	`PaperXI.PacketLines.inertia_stable`
49	theorem	`PaperXI.PacketLines.projector_cycle`
66	theorem	`PaperXI.PacketLines.projection_to_line`
83	theorem	`PaperXI.PacketLines.packet_eq_line_of_span_top`
98	theorem	`PaperXI.PacketLines.generators_preserve_cycleSpan`
118	theorem	`PaperXI.PacketLines.cycleSpan_eq_top`
131	theorem	`PaperXI.PacketLines.nonzero_packet_finrank_one`

A.313 PaperXI.PacketReturn

Source lines: 61.

SHA-256: 54669679788cd0095e0d26b16343de68a5196641db0e60dad21ec97090f52942.

```
The actual return operator on a packet and its nonzero eigenline. The
claim that this line is the whole packet is deliberately not assumed.
```

Direct project imports: PaperXI.FrobeniusPeriod.

Line	Kind	Name
13	theorem	PaperXI.PacketReturn.power_mem
23	theorem	PaperXI.PacketReturn.return_mem
30	def	PaperXI.PacketReturn.closingOperator
40	theorem	PaperXI.PacketReturn.exists_closing_eigenvector

A.314 PaperXI.PacketSupport

Source lines: 105.

SHA-256: 3265ba26afef8a86567d0e4ecc83f6448bf3458790ff26d376ae84f7834b78f1.

```
Exact support, and constant packet dimension along the Frobenius orbit.
```

Direct project imports: PaperXI.TameOrbitTransitivity.

Line	Kind	Name
12	theorem	PaperXI.PacketSupport.nonzero_step_iff
19	theorem	PaperXI.PacketSupport.nonzero_back_iff
26	theorem	PaperXI.PacketSupport.nonzero_zpow_iff
39	theorem	PaperXI.PacketSupport.finrank_zpow
68	theorem	PaperXI.PacketSupport.support_iff_orbit
82	theorem	PaperXI.PacketSupport.support_is_one_orbit
92	theorem	PaperXI.PacketSupport.equal_packet_dimensions

A.315 PaperXI.PacketTransport

Source lines: 81.

SHA-256: dc7282e77c935ca61b8807fe87f4fe2fa372d720a28d0a1c8982c6c749e1a910.

```
Actual Frobenius transport between the ranges of the computed projectors.
```

Direct project imports: PaperXI.TameProjectors.

Line	Kind	Name
11	abbrev	PaperXI.PacketTransport.packet
14	abbrev	PaperXI.PacketTransport.projector
17	theorem	PaperXI.PacketTransport.mem_iff_projector
25	theorem	PaperXI.PacketTransport.projector_on_other_packet

Line	Kind	Name
32	theorem	`PaperXI.PacketTransport.inertia_mem`
38	theorem	`PaperXI.PacketTransport.transport_of_conjugation`
46	theorem	`PaperXI.PacketTransport.forward_mem`
53	theorem	`PaperXI.PacketTransport.backward_mem`
62	def	`PaperXI.PacketTransport.packetEquiv`
75	theorem	`PaperXI.PacketTransport.packet_finrank_step`

A.316 `PaperXI.PacketTypeAttachment70`

Source lines: 85.

SHA-256: `ed82a000e951e6dca19a854165a2e9f92be60a3bcc1ed876f0fae8a9978b8a79`.

```
The finite packet action is the restriction of the SAME actual induced
representation on its literal type image. Full operator relations and graph
edge residuals are preserved and reflected. No arithmetic factor theory is used.
```

Direct project imports: `PaperXI.TensorTypeOperators70`.

Line	Kind	Name
17	def	`PaperXI.PacketTypeAttachment70.typeAction`
21	theorem	`PaperXI.PacketTypeAttachment70.ambient_action`
32	theorem	`PaperXI.PacketTypeAttachment70.first_factor_action`
47	theorem	`PaperXI.PacketTypeAttachment70.first_factor_average`
61	theorem	`PaperXI.PacketTypeAttachment70.generator_action`
71	theorem	`PaperXI.PacketTypeAttachment70.edge_compatibility_iff`
79	theorem	`PaperXI.PacketTypeAttachment70.edge_residual_iff`

A.317 `PaperXI.PairIndex110`

Source lines: 122.

SHA-256: `453d18235bd27cd82e48a7cf367e7b3433288906412d7e7154e80b955025408b`.

```
The pair group is identified with the first-coordinate intersection.
The orientation is literal: `ι b = g * ι a * g⁻¹`. The intersection in `K`
therefore contains the `a` whose conjugated image remains in `ι(K)`.
All results are generic group and index results; no arithmetic identity is assumed.
```

Direct project imports: `PaperXI.IntersectionModel108`.

Line	Kind	Name
19	def	`PaperXI.PairIndex110.intersection`
23	def	`PaperXI.PairIndex110.scalarIntersection`
26	theorem	`PaperXI.PairIndex110.scalarIntersection_le`
29	theorem	`PaperXI.PairIndex110.scalarIntersection_le_intersection`
35	theorem	`PaperXI.PairIndex110.intersection_image`
47	theorem	`PaperXI.PairIndex110.scalarIntersection_image`
58	theorem	`PaperXI.PairIndex110.left_mem_intersection`
62	theorem	`PaperXI.PairIndex110.left_range`
71	theorem	`PaperXI.PairIndex110.left_injective`

Line	Kind	Name
82	def	PaperXI.PairIndex110.pairEquivIntersection
91	theorem	PaperXI.PairIndex110.pairSupport_eq_comap
106	theorem	PaperXI.PairIndex110.pairSupport_index
113	theorem	PaperXI.PairIndex110.pairSupport_index_ambient

A.318 PaperXI.PeriodicLatticeReduction60

Source lines: 86.

SHA-256: f70d40f2ca38dfa0635a4232d830d16dbbdc2f58c4472d1c82c4a25950e93a02.

```
# Finite reduction of the literal lattice-chain conditions

A positive period given by a base-field scalar reduces the entire integer
chain to finitely many levels for every base-field linear automorphism.
```

Direct project imports: PaperXI.ActualELatticeNormalizer59.

Line	Kind	Name
17	def	PaperXI.PeriodicLatticeReduction60.levelCondition
21	theorem	PaperXI.PeriodicLatticeReduction60.levelCondition_shift
53	theorem	PaperXI.PeriodicLatticeReduction60.levelCondition_periodic
60	theorem	PaperXI.PeriodicLatticeReduction60.mem_firstCongruence_iff_fin

A.319 PaperXI.PiPolynomial

Source lines: 58.

SHA-256: 3aaecfa5d206cd6d476431f7e9f3f823ec7b57ec488a1edb5d6170a3c973496e.

```
Characteristic and local polynomials of finite families of varying spaces.
```

Direct project imports: PaperXI.InvariantPolynomial.

Line	Kind	Name
13	theorem	PaperXI.PiPolynomial.charpoly_pi
46	theorem	PaperXI.PiPolynomial.operatorPolynomial_pi

A.320 PaperXI.PolynomialAssembly

Source lines: 83.

SHA-256: 538480496e4eb1bceb57fd376c890da010064e5ca4d64edc62ceb4394e77663f.

```
Restricted Frobenius and determinant products under equivariant decompositions.
```

Direct project imports: PaperXI.InvariantAssembly, PaperXI.PiPolynomial.

Line	Kind	Name
14	def	PaperXI.PolynomialAssembly.restricted
20	theorem	PaperXI.PolynomialAssembly.restricted_apply
24	theorem	PaperXI.PolynomialAssembly.stable_of_decomposition
37	theorem	PaperXI.PolynomialAssembly.restricted_conj
53	theorem	PaperXI.PolynomialAssembly.polynomial_product
67	def	PaperXI.PolynomialAssembly.localFactor
71	theorem	PaperXI.PolynomialAssembly.localFactor_product

A.321 PaperXI.PrescribedFieldType70

Source lines: 87.

SHA-256: 6cd430fa9358f4bcd875eaede181bed47500c5d2d71f94e4d937f10231a5d88f.

```
Construction of the full inducing representation from the same compact
first factor and actual finite quotient. The full-H locus is a later condition
on this exact constructed extension, not on an unrelated supplied Lambda.
```

Direct project imports: PaperXI.ActualCompactTypeCompletion58, PaperXI.ActualFieldTypeQuotient69.

Line	Kind	Name
28	def	PaperXI.PrescribedFieldType70.compactInclusion
31	def	PaperXI.PrescribedFieldType70.fieldGenerator
37	def	PaperXI.PrescribedFieldType70.domainEquiv
44	theorem	PaperXI.PrescribedFieldType70.exists_prescribed_extension

A.322 PaperXI.PrimitiveCharacterBlock

Source lines: 128.

SHA-256: 5950f0f65ec50d3dc6a4c36c40237c087ccc4a4c0eeb35b1fe536e897bcdf6e5.

```
Primitive central character blocks in the literal finite group algebra.
Primitivity means absence of nontrivial central sub-idempotents, not rank one.
```

Direct project imports: PaperXI.SemisimpleDetection.

Line	Kind	Name
17	theorem	PaperXI.PrimitiveCharacterBlock.equiv_algebra_action
26	theorem	PaperXI.PrimitiveCharacterBlock.nonzero
35	theorem	PaperXI.PrimitiveCharacterBlock.idempotent
43	theorem	PaperXI.PrimitiveCharacterBlock.supported_eq_zero
60	theorem	PaperXI.PrimitiveCharacterBlock.block_action_injective
69	def	PaperXI.PrimitiveCharacterBlock.IsPrimitiveCentral
74	theorem	PaperXI.PrimitiveCharacterBlock.primitive_central
99	theorem	PaperXI.PrimitiveCharacterBlock.no_nontrivial_central_split
112	theorem	PaperXI.PrimitiveCharacterBlock.orthogonal
123	theorem	PaperXI.PrimitiveCharacterBlock.presentation_invariant

A.323 PaperXI.ProductNormalizerQuotient59

Source lines: 95.

SHA-256: 253347996905c91f7605f14539fa7abaf854e23c86fc22008272369cf3bd30bd.

```
# Product subgroups and the action of a centralizing normalizer

For actual subgroups `H,N` of an ambient group, with `H` normalizing `N`,
the quotient of `H ⊔ N` by `N` is the quotient of `H` by its intersection
with `N`. An ambient element centralizing `H` and normalizing `N` acts
trivially on this quotient. No identification with a simple-character group
is asserted here.
```

Direct project imports: PaperXI.NormalizerExtensionInducing58.

Line	Kind	Name
25	theorem	PaperXI.ProductNormalizerQuotient59.normalizes_sup
32	def	PaperXI.ProductNormalizerQuotient59.conjugation
35	theorem	PaperXI.ProductNormalizerQuotient59.conjugation_coe
40	theorem	PaperXI.ProductNormalizerQuotient59.map_conjugation
63	theorem	PaperXI.ProductNormalizerQuotient59.quotient_conjugation
73	def	PaperXI.ProductNormalizerQuotient59.productQuotientEquiv
83	def	PaperXI.ProductNormalizerQuotient59.quotientEquivOfKernel

A.324 PaperXI.ProjectiveCocycle

Source lines: 171.

SHA-256: d2eb7d728fef4b48932df07f3dc13a003f39995cc477c3728f0677302794dcf2.

```
Scalar cocycles extracted from the multiplication of actual nonzero operators.
Associativity and normalization are conclusions of operator identities.
```

Direct project imports: PaperXI.CocycleCommutator, PaperXI.SchurScalar.

Line	Kind	Name
14	theorem	PaperXI.ProjectiveCocycle.scalar_injective
21	def	PaperXI.ProjectiveCocycle.cocycle
32	theorem	PaperXI.ProjectiveCocycle.normalized
45	theorem	PaperXI.ProjectiveCocycle.coefficient_unique
53	theorem	PaperXI.ProjectiveCocycle.rescale_coefficients
72	theorem	PaperXI.ProjectiveCocycle.commutator_rescale_operators
89	theorem	PaperXI.ProjectiveCocycle.cocycle_rescale_operators
101	theorem	PaperXI.ProjectiveCocycle.cochain_one
109	theorem	PaperXI.ProjectiveCocycle.rescaledFamily_mul
123	theorem	PaperXI.ProjectiveCocycle.conjugate_mul
133	theorem	PaperXI.ProjectiveCocycle.conjugate_rescale_coefficients
149	theorem	PaperXI.ProjectiveCocycle.conjugate_rescale_commutator

A.325 PaperXI.ProjectiveHolonomy

Source lines: 173.

SHA-256: dbc14aa36dc916019628345b811c2d1904c4652d23cddbe71792a93751a8db05.

```
Vertex rescaling of actual graph transport and scalar ratio characters.
```

Direct project imports: `PaperXI.ScalarHolonomy`, `PaperXI.GraphTransport`, `PaperXI.GraphLoopRepresentation`.

Line	Kind	Name
20	def	`PaperXI.ProjectiveHolonomy.rescale`
24	theorem	`PaperXI.ProjectiveHolonomy.rescale_apply`
31	theorem	`PaperXI.ProjectiveHolonomy.rescale_gauge`
39	theorem	`PaperXI.ProjectiveHolonomy.rescale_path`
50	theorem	`PaperXI.ProjectiveHolonomy.rescale_path_apply`
59	theorem	`PaperXI.ProjectiveHolonomy.rescale_loop`
66	theorem	`PaperXI.ProjectiveHolonomy.rescale_holonomy`
75	theorem	`PaperXI.ProjectiveHolonomy.rescale_isScalar`
81	theorem	`PaperXI.ProjectiveHolonomy.isScalar_iff_paths`
93	def	`PaperXI.ProjectiveHolonomy.loopCharacter`
97	theorem	`PaperXI.ProjectiveHolonomy.loopCharacter_spec`
105	theorem	`PaperXI.ProjectiveHolonomy.rescale_character`
116	def	`PaperXI.ProjectiveHolonomy.ratioH1`
119	theorem	`PaperXI.ProjectiveHolonomy.ratioH1_class`
123	theorem	`PaperXI.ProjectiveHolonomy.ratioH1_unique`
129	theorem	`PaperXI.ProjectiveHolonomy.ratioH1_eq_one_iff`
145	def	`PaperXI.ProjectiveHolonomy.transportRatioH1`
152	theorem	`PaperXI.ProjectiveHolonomy.transportRatioH1_class`
162	theorem	`PaperXI.ProjectiveHolonomy.transportRatioH1_rescale`

A.326 `PaperXI.ProjectorTransport`

Source lines: 27.

SHA-256: `c88eba65d49511600eaef67165160b9de7d6f1dcb5c6f56476feb8a5259b3592`.

```
Exact reindexing law underlying the Frobenius covariance in Proposition 3.1.
This does not yet prove Fourier idempotence or the Weil-group realization.
The character weight transforms by the inverse permutation.
```

Direct project imports: none.

Line	Kind	Name
14	theorem	`PaperXI.ProjectorTransport.weighted_sum_conjugation`
21	theorem	`PaperXI.ProjectorTransport.normalized_sum_conjugation`

A.327 `PaperXI.QuotientRepresentation`

Source lines: 72.

SHA-256: `4439d25c13ba1ce6c73f18c0d67c0a8ce5f96caa0c1befb335168c02a01fffe5`.

```
Actual representation descent through a quotient group and preservation
of irreducibility and intertwining equations under a surjective group map.
```

Direct project imports: none.

Line	Kind	Name
14	def	PaperXI.QuotientRepresentation.subrepresentationEquiv
25	theorem	PaperXI.QuotientRepresentation.irreducible_comp_iff
30	theorem	PaperXI.QuotientRepresentation.intertwines_comp_iff
42	def	PaperXI.QuotientRepresentation.descend
46	theorem	PaperXI.QuotientRepresentation.descend_apply
49	theorem	PaperXI.QuotientRepresentation.descend_comp
52	theorem	PaperXI.QuotientRepresentation.descend_irreducible_iff
58	theorem	PaperXI.QuotientRepresentation.descend_irreducible
64	theorem	PaperXI.QuotientRepresentation.descend_unique

A.328 PaperXI.RadicalOnePlus111

Source lines: 58.

SHA-256: 3ebe8185adccd13a0c9d9102e437f22fb910637ef074bfb50331602f7eac41a7.

```
Exact one-plus groups from actual radical ideals in the original order.
The order may be noncommutative. No arbitrary additive lattice is declared
to be a group or an ideal of the full field endomorphism algebra.
```

Direct project imports: PaperXI.OnePlusIntersection110.

Line	Kind	Name
16	theorem	PaperXI.RadicalOnePlus111.isUnit_of_sub_one_mem_jacobson
34	def	PaperXI.RadicalOnePlus111.onePlus
36	theorem	PaperXI.RadicalOnePlus111.onePlus_offsets
40	theorem	PaperXI.RadicalOnePlus111.onePlus_surjective
53	def	PaperXI.RadicalOnePlus111.offsetEquiv

A.329 PaperXI.RamificationPolyphase

Source lines: 326.

SHA-256: 2f34bd694def457221d6f00b874c8b54547dcc95ee249e3afcf34d71b5b64549.

```
# Ramification polyphase projectors

The lower ramification filtration gives a nested family of literal finite-group
averaging projectors.  Successive differences are the depth-detail channels.
This is a representation-theoretic multiresolution theorem; it does not assert
existence of an intrinsic wild-break decomposition beyond the supplied lower
filtration.
```

Direct project imports: PaperXI.ValuationConductor, PaperXI.FiniteAverage.

Line	Kind	Name
29	theorem	PaperXI.RamificationPolyphase.average_comp_operator
40	theorem	PaperXI.RamificationPolyphase.average_comp_restricted_average
53	theorem	PaperXI.RamificationPolyphase.restricted_average_comp_average
69	def	PaperXI.RamificationPolyphase.lowerInclusion
75	theorem	PaperXI.RamificationPolyphase.lowerInclusion_val
80	theorem	PaperXI.RamificationPolyphase.lift_lowerInclusion

Line	Kind	Name
84	def	PaperXI.RamificationPolyphase.depthProjector
88	theorem	PaperXI.RamificationPolyphase.depthProjector_range
93	theorem	PaperXI.RamificationPolyphase.depthProjector_idempotent
100	theorem	PaperXI.RamificationPolyphase.depthProjector_comp_of_le
114	theorem	PaperXI.RamificationPolyphase.depthProjector_comp_of_le_rev
127	def	PaperXI.RamificationPolyphase.detailProjector
130	theorem	PaperXI.RamificationPolyphase.detailProjector_apply
135	theorem	PaperXI.RamificationPolyphase.detailProjector_idempotent
167	theorem	PaperXI.RamificationPolyphase.depthProjector_detailProjector
186	theorem	PaperXI.RamificationPolyphase.depthProjector_succ_detailProjector
206	theorem	PaperXI.RamificationPolyphase.detailProjector_mem_next_fixed
215	theorem	PaperXI.RamificationPolyphase.detailProjector_range
232	theorem	PaperXI.RamificationPolyphase.detailProjector_comp_eq_zero
265	theorem	PaperXI.RamificationPolyphase.detailProjector_comp_eq_zero_rev
299	theorem	PaperXI.RamificationPolyphase.depthProjector_eq_zero_add_sum
311	theorem	PaperXI.RamificationPolyphase.identity_eq_inertia_add_details

A.330 PaperXI.RamificationPolyphaseBaseChange

Source lines: 115.

SHA-256: eb07cacbdaafae8606ca94ef6790ab0e3c7df865ff062943f1757f3ca826daa2.

```
# Unramified transport of ramification polyphase channels

For an inertia-containing subgroup and the corresponding Frobenius power
change, every literal lower-depth averaging projector is preserved. Hence every
successive ramification-detail projector is preserved as an actual endomorphism,
not only at the level of dimensions or conductor sums.
```

Direct project imports: PaperXI.RamificationPolyphase, PaperXI.UnramifiedConductor.

Line	Kind	Name
36	theorem	PaperXI.RamificationPolyphaseBaseChange.depthProjector_eq
59	theorem	PaperXI.RamificationPolyphaseBaseChange.detailProjector_eq
69	theorem	PaperXI.RamificationPolyphaseBaseChange.detailRange_eq
78	theorem	PaperXI.RamificationPolyphaseBaseChange.finiteReconstruction_eq
96	theorem	PaperXI.RamificationPolyphaseBaseChange.conductorSummand_eq
103	theorem	PaperXI.RamificationPolyphaseBaseChange.swan_eq
110	theorem	PaperXI.RamificationPolyphaseBaseChange.artin_eq

A.331 PaperXI.RamificationPolyphaseDimensions

Source lines: 180.

SHA-256: f38214cac70955d55434afad4ca9162cd9145d6a627a88311444bac1906e871d.

```
# Dimension and conductor accounting for ramification details

The literal depth projectors split every successive fixed space into its previous
coarse part and one newly visible detail range.  In finite dimension, the detail
ranks telescope to the whole representation and give a finite tail formula for
every lower-numbering codimension.
```

Direct project imports: `PaperXI.RamificationPolyphase`, `PaperXI.RamificationSeparation`.

Line	Kind	Name
31	def	`PaperXI.RamificationPolyphaseDimensions.detailSpace`
37	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_mono`
47	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_succ_eq_sup_detail`
76	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_disjoint_detail`
97	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_fixed_succ`
108	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_fixed_eq_base_add_details`
119	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_eq_inertia_add_details`
135	theorem	`PaperXI.RamificationPolyphaseDimensions.exists_identity_eq_inertia_add_details`
146	theorem	`PaperXI.RamificationPolyphaseDimensions.exists_finrank_eq_inertia_add_details`
157	theorem	`PaperXI.RamificationPolyphaseDimensions.codimension_eq_detail_tail`
170	theorem	`PaperXI.RamificationPolyphaseDimensions.summand_eq_weight_mul_detail_tail`

A.332 `PaperXI.RamificationSeparation`

Source lines: 78.

SHA-256: `f4b0880d3b76829d7b9274fe0f6817a408aa233aed0a8530687b2ec95f7c6094`.

```
Separated local rings and finite faithful automorphism groups have a terminating
lower congruence filtration. This justifies finite support of the conductor sum;
eventual triviality is a theorem, not an assumed cutoff.
```

Direct project imports: `PaperXI.ValuationRamification`.

Line	Kind	Name
17	theorem	`PaperXI.RamificationSeparation.mem_all_lower_iff`
37	theorem	`PaperXI.RamificationSeparation.eventually_trivial`
59	instance	`PaperXI.RamificationSeparation.valuation_faithful`
72	theorem	`PaperXI.RamificationSeparation.valuation_eventually_trivial`

A.333 `PaperXI.RamifiedUnramifiedQuotient`

Source lines: 322.

SHA-256: `755d457b0f11dfbac94eaee10326fa290bd5b6a331d8883bb8ce26811f00a747`.

```
# Ramified pullbacks and their inertia-free degree quotient

The residue--Frobenius pullback retains a decomposition-group coordinate and an
integer degree.  Its degree kernel is inertia.  This file proves the precise
first-isomorphism statement and compares the quotient with the integer-degree
unramified model only under an explicit degree-one Frobenius-lift hypothesis.

It also proves the exact representation-theoretic information loss: a
representation pulled back only through integer degree kills every lower
ramification subgroup, so every ramification depth projector is the identity
and every positive detail projector is zero.
```

Direct project imports: `PaperXI.RamificationPolyphase`, `PaperXI.FiniteUnramifiedWeil`.

Line	Kind	Name
38	abbrev	PaperXI.RamifiedUnramifiedQuotient.Inertia
42	def	PaperXI.RamifiedUnramifiedQuotient.quotientEquivDegreeRange
47	theorem	PaperXI.RamifiedUnramifiedQuotient.degree_surjective_of_degree_one
59	theorem	PaperXI.RamifiedUnramifiedQuotient.degree_surjective_of_frobenius_mem_range
66	def	PaperXI.RamifiedUnramifiedQuotient.quotientEquivIntegerDegree
72	theorem	PaperXI.RamifiedUnramifiedQuotient.quotientEquivIntegerDegree_mk
79	theorem	PaperXI.RamifiedUnramifiedQuotient.frobenius_lift_ratio_mem_inertia
102	def	PaperXI.RamifiedUnramifiedQuotient.degreeModelMap
107	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_degree
112	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_ker
119	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_surjective
130	def	PaperXI.RamifiedUnramifiedQuotient.quotientEquivUnramified
136	theorem	PaperXI.RamifiedUnramifiedQuotient.quotientEquivUnramified_mk_degree
145	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_eq_geometricLift_of_degree_one
158	def	PaperXI.RamifiedUnramifiedQuotient.pullbackDegreeRepresentation
164	theorem	PaperXI.RamifiedUnramifiedQuotient.inertia_le_pullbackDegreeRepresentation_ker
175	def	PaperXI.RamifiedUnramifiedQuotient.quotientRepresentation
183	def	PaperXI.RamifiedUnramifiedQuotient.descendRepresentation
192	theorem	PaperXI.RamifiedUnramifiedQuotient.descendRepresentation_comp_degreeModelMap
215	theorem	PaperXI.RamifiedUnramifiedQuotient.descendRepresentation_unique
250	abbrev	PaperXI.RamifiedUnramifiedQuotient.degreeOnlyRepresentation
257	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnlyRepresentation_lift_eq_one
271	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_fixed_eq_top
280	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_depthProjector_eq_id
292	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_detailProjector_eq_zero
301	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_summand_eq_zero
308	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_swan_eq_zero
314	theorem	PaperXI.RamifiedUnramifiedQuotient.degreeOnly_artin_eq_zero

A.334 PaperXI.RankOneEulerFactor

Source lines: 113.

SHA-256: a512466f456be1c2a7b5c0f49b6b7cfad6bbda69ec1bc75283bb51b8140b95c3.

```
# Rank-one Euler factors for irreducible unramified representations
```

Direct project imports: PaperXI.UnramifiedConductorZero.

Line	Kind	Name
27	theorem	PaperXI.FiniteUnramifiedWeil.reverse_X_sub_C
38	theorem	PaperXI.FiniteUnramifiedWeil.reverse_charpoly_smul_id
59	def	PaperXI.FiniteUnramifiedWeil.frobeniusEigenvalue
65	theorem	PaperXI.FiniteUnramifiedWeil.geometricFrobenius_action_eq_smul_id
73	theorem	PaperXI.FiniteUnramifiedWeil.frobeniusEigenvalue_ne_zero
92	theorem	PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eq_linear
100	def	PaperXI.FiniteUnramifiedWeil.rankOneEulerFactor

Line	Kind	Name
105	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_eq_rankOneEulerFactor`

A.335 `PaperXI.RankOneSourceCandidate110`

Source lines: 77.

SHA-256: `7f57ede141636d625efea3d9b67e905b1f34bcd5dd6c475f31a7e2ec2a7b2928`.

```
Residue rank one from the lower-level simple-character support dictionary.
The model containment, compact support, integral normalization and finite
residue factor are derived. Canonical character data are not manufactured.
```

Direct project imports: `PaperXI.ArithmeticAttachment110`, `PaperXI.NormalizerDictionary110`.

Line	Kind	Name
29	theorem	`PaperXI.RankOneSourceCandidate110.candidate_exists`

A.336 `PaperXI.RegularCyclic`

Source lines: 90.

SHA-256: `2ed6fa5d0f002e4a087b67badeca9c39702efe1de6d6feade2a7eaf34fc5bccf`.

```
A concrete regular cyclic model. Every character packet is nonzero, and
the erroneous forward-label rule has an actual complex representation counterexample.
```

Direct project imports: `PaperXI.TameProjectors`.

Line	Kind	Name
12	def	`PaperXI.RegularCyclic.translation`
17	def	`PaperXI.RegularCyclic.regular`
30	theorem	`PaperXI.RegularCyclic.operator_apply`
33	def	`PaperXI.RegularCyclic.frobenius`
41	theorem	`PaperXI.RegularCyclic.generator_relation`
50	theorem	`PaperXI.RegularCyclic.character_vector_mem`
59	theorem	`PaperXI.RegularCyclic.projector_nonzero`
67	theorem	`PaperXI.RegularCyclic.projectors_distinct`
76	def	`PaperXI.RegularCyclic.twoModFive`
79	theorem	`PaperXI.RegularCyclic.wrong_forward_label`

A.337 `PaperXI.RepresentationGraph`

Source lines: 106.

SHA-256: `5919599a901246a3def38a1fc9ae5480fabe1dc0a2f1c09be274d25317788d6d`.

```
Gluing within actual representation intertwiners. Edge covariance ensures
that tree transport of a root bridge remains a local bridge at every vertex.
```

Direct project imports: `PaperXI.GraphGauge`.

Line	Kind	Name
19	def	`PaperXI.RepresentationGraph.EdgeEquivariant`
25	def	`PaperXI.RepresentationGraph.Intertwines`
34	theorem	`PaperXI.RepresentationGraph.path_equivariant`
42	theorem	`PaperXI.RepresentationGraph.rootGauge_intertwines`
55	theorem	`PaperXI.RepresentationGraph.based_iff_chords`
71	theorem	`PaperXI.RepresentationGraph.connected_iff_chords`
81	theorem	`PaperXI.RepresentationGraph.based_iff_loops`
91	theorem	`PaperXI.RepresentationGraph.unbased_iff_conjugates`

A.338 `PaperXI.RepresentationStability`

Source lines: 47.

SHA-256: `ea48bcbf4e471e1d436ca33befd6c1aa3dcec5b7e765b15b407087b7c26cd33c`.

```
An explicit invariant-subspace criterion for actual irreducibility. This
interface keeps the nonzero-space condition when comparing group actions and
algebra actions.
```

Direct project imports: `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
13	def	`PaperXI.RepresentationStability.Stable`
17	theorem	`PaperXI.RepresentationStability.eq_bot_or_top`
25	theorem	`PaperXI.RepresentationStability.irreducible_of_stable`
36	theorem	`PaperXI.RepresentationStability.irreducible_iff`

A.339 `PaperXI.ResidueDegreeBaseChange`

Source lines: 148.

SHA-256: `c581b1a747917d5c92d930d8b817fd79fb5bf2c3de951682b786377850705bee`.

```
Integer-degree base change for residue-action pullbacks.
The map (g,m) ↦ (g,n*m) is constructed; neither subgroup equality nor index is
assumed. This does not identify a pullback with an absolute local Weil group.
```

Direct project imports: `PaperXI.ResidueWeil`.

Line	Kind	Name
15	def	`PaperXI.ResidueDegreeBaseChange.degreeMod`
24	def	`PaperXI.ResidueDegreeBaseChange.periodSubgroup`
27	instance	`PaperXI.ResidueDegreeBaseChange.period_normal`
30	theorem	`PaperXI.ResidueDegreeBaseChange.mem_periodSubgroup`
36	def	`PaperXI.ResidueDegreeBaseChange.inclusion`
50	theorem	`PaperXI.ResidueDegreeBaseChange.inclusion_projection`
53	theorem	`PaperXI.ResidueDegreeBaseChange.inclusion_degree`
57	theorem	`PaperXI.ResidueDegreeBaseChange.inclusion_injective`

Line	Kind	Name
69	theorem	`PaperXI.ResidueDegreeBaseChange.inclusion_range`
89	def	`PaperXI.ResidueDegreeBaseChange.periodEquiv`
95	theorem	`PaperXI.ResidueDegreeBaseChange.degree_surjective`
105	theorem	`PaperXI.ResidueDegreeBaseChange.degreeMod_surjective`
116	def	`PaperXI.ResidueDegreeBaseChange.quotientEquiv`
121	theorem	`PaperXI.ResidueDegreeBaseChange.quotient_card`
127	theorem	`PaperXI.ResidueDegreeBaseChange.inertia_preimage`
141	theorem	`PaperXI.ResidueDegreeBaseChange.tower_degree`

A.340 `PaperXI.ResidueExtensionRealization`

Source lines: 97.

SHA-256: `b24752381c6a6353645a53ffe2613d10e783ecf9086a2cdfb5876dba34bc9ccf`.

```
Realization of any finite separable extension of the actual base residue field.
The monic lift and its ring/field are constructed; no ambient Galois extension,
residue embedding into an existing extension, or degree certificate is supplied.
```

Direct project imports: `PaperXI.UnramifiedPolynomial`.

Line	Kind	Name
16	def	`PaperXI.ResidueExtensionRealization.residueBasis`
19	def	`PaperXI.ResidueExtensionRealization.residuePolynomial`
22	theorem	`PaperXI.ResidueExtensionRealization.residuePolynomial_monic`
25	theorem	`PaperXI.ResidueExtensionRealization.residuePolynomial_irreducible`
28	theorem	`PaperXI.ResidueExtensionRealization.residuePolynomial_degree`
33	theorem	`PaperXI.ResidueExtensionRealization.exists_monic_lift`
43	def	`PaperXI.ResidueExtensionRealization.liftedPolynomial`
45	theorem	`PaperXI.ResidueExtensionRealization.liftedPolynomial_map`
49	theorem	`PaperXI.ResidueExtensionRealization.liftedPolynomial_degree`
53	instance	`PaperXI.ResidueExtensionRealization.monic`
56	instance	`PaperXI.ResidueExtensionRealization.residueIrreducible`
60	abbrev	`PaperXI.ResidueExtensionRealization.ring`
61	abbrev	`PaperXI.ResidueExtensionRealization.field`
64	def	`PaperXI.ResidueExtensionRealization.residueEquiv`
72	theorem	`PaperXI.ResidueExtensionRealization.rank`
76	theorem	`PaperXI.ResidueExtensionRealization.field_degree`
82	theorem	`PaperXI.ResidueExtensionRealization.etale`
85	theorem	`PaperXI.ResidueExtensionRealization.ramificationIdx_one`
88	theorem	`PaperXI.ResidueExtensionRealization.source`

A.341 `PaperXI.ResidueFrobeniusAttachment`

Source lines: 144.

SHA-256: `f8bdcb103727ce29ab4fa9647716c21624920c75aff28d23c3a398cee0f59f7c`.

```
# Frobenius attachment to the constructed residue field
```

Direct project imports: `PaperXI.IntegerRingComparison`.

Line	Kind	Name
23	abbrev	PaperXI.FiniteUnramifiedWeil.UpperField
24	abbrev	PaperXI.FiniteUnramifiedWeil.BaseResidue
25	abbrev	PaperXI.FiniteUnramifiedWeil.UpperResidue
26	abbrev	PaperXI.FiniteUnramifiedWeil.ResidueGalois
27	abbrev	PaperXI.FiniteUnramifiedWeil.ResidueDegree
36	def	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius
40	def	PaperXI.FiniteUnramifiedWeil.geometricFrobenius
44	theorem	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_apply
49	theorem	PaperXI.FiniteUnramifiedWeil.geometricFrobenius_inverse_power
55	theorem	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_order
61	theorem	PaperXI.FiniteUnramifiedWeil.geometricFrobenius_order
67	theorem	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_order_eq_fieldDegree
73	theorem	PaperXI.FiniteUnramifiedWeil.geometricFrobenius_order_eq_fieldDegree
79	theorem	PaperXI.FiniteUnramifiedWeil.natCard_residueGalois
85	theorem	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_mem_zpowers
95	theorem	PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_zpowers_eq_top
101	theorem	PaperXI.FiniteUnramifiedWeil.geometricFrobenius_zpowers_eq_top
107	def	PaperXI.FiniteUnramifiedWeil.geometricCycleEquiv
116	theorem	PaperXI.FiniteUnramifiedWeil.geometricCycleEquiv_one
122	def	PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv
127	theorem	PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv_arithmetic
138	theorem	PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv_geometric

A.342 PaperXI.ResidueRestriction107

Source lines: 75.

SHA-256: d1008a38fcb166fa8167e9591321de444cc9d86d3ae176b13511e4639d0ab405.

```
The previously constructed residue map agrees with literal entrywise
reduction on integral centralizer matrices. This identity is proved, not supplied.
```

Direct project imports: PaperXI.ActualCuspidalCandidate105.

Line	Kind	Name
12	theorem	PaperXI.ResidueRestriction107.quotient_inclusion
44	def	PaperXI.ResidueRestriction107.integralInclusion
49	theorem	PaperXI.ResidueRestriction107.integralInclusion_coe
56	theorem	PaperXI.ResidueRestriction107.reduction_integral

A.343 PaperXI.ResidueWeil

Source lines: 114.

SHA-256: 1eee4bb854ff6f99731ba944d6c0540ef7595df6fb8fa862ae030132fcd38166.

```
A pullback of an actual residue-field action along the integer powers of a
chosen residue automorphism. No equivalence with the absolute local Weil group
is postulated. The inertia identification below is with Mathlib's valuation
inertia subgroup, not with an independently named abstract kernel.
```

Direct project imports: `PaperXI.NormalInertia`.

Line	Kind	Name
18	def	`PaperXI.ResidueWeil.subgroup`
30	abbrev	`PaperXI.ResidueWeil.Pullback`
32	def	`PaperXI.ResidueWeil.projection`
35	def	`PaperXI.ResidueWeil.degree`
38	theorem	`PaperXI.ResidueWeil.projection_apply`
41	theorem	`PaperXI.ResidueWeil.degree_apply`
45	def	`PaperXI.ResidueWeil.inertiaEquiv`
65	instance	`PaperXI.ResidueWeil.degreeQuotientCommutative`
73	theorem	`PaperXI.ResidueWeil.exists_degree_one_iff`
90	abbrev	`PaperXI.ResidueWeil.residueAction`
94	abbrev	`PaperXI.ResidueWeil.valuationGroup`
98	def	`PaperXI.ResidueWeil.valuationInertiaEquiv`
106	theorem	`PaperXI.ResidueWeil.valuation_invariants_eq_bot`

A.344 `PaperXI.RestrictedHeisenbergBridge`

Source lines: 84.

SHA-256: `4385767418858294f763e1ef3b8a60700679a6399d58874349e83ce8533e5af6`.

```
Source 11.2: the initial bridge between the supplied Heisenberg envelopes.
The odd branch constructs the lift on arbitrary abstract symplectic spaces.
The dyadic branch retains the supplied isomorphism of actual central
extensions. Neither branch assumes an initial intertwiner, scalar-cocycle
nondegeneracy, or finite-dimensionality of either representation space.
```

Direct project imports: `PaperXI.CentralExtensionComparison`, `PaperXI.AbstractHeisenbergExtension`, `PaperXI.SymplecticCentralExtension`.

Line	Kind	Name
24	theorem	`PaperXI.RestrictedHeisenbergBridge.odd_source`
58	theorem	`PaperXI.RestrictedHeisenbergBridge.dyadic_source`

A.345 `PaperXI.ScalarCocycle112`

Source lines: 195.

SHA-256: `351ac3d1e21afeb0a8897f6eea1ba99f83bb63eaeee390eebefa44c1dfbb0f4e`.

```
Scalar cocycle classification for complex projective comparison.
No identification of the full finite centers or dyadic group lift is assumed.
```

Direct project imports: none.

Line	Kind	Name
16	structure	`PaperXI.ScalarCocycle112.Cocycle`
23	def	`PaperXI.ScalarCocycle112.commutator`
26	structure	`PaperXI.ScalarCocycle112.Extension`
33	theorem	`PaperXI.ScalarCocycle112.Extension.ext`

Line	Kind	Name
42	theorem	`PaperXI.ScalarCocycle112.Extension.mul_scalar`
44	theorem	`PaperXI.ScalarCocycle112.Extension.mul_base`
45	theorem	`PaperXI.ScalarCocycle112.Extension.one_scalar`
46	theorem	`PaperXI.ScalarCocycle112.Extension.one_base`
69	def	`PaperXI.ScalarCocycle112.Extension.commGroup`
78	def	`PaperXI.ScalarCocycle112.Extension.scalarInclusion`
83	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_scalar`
85	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_base`
88	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_injective`
92	def	`PaperXI.ScalarCocycle112.Extension.sectionOf`
94	theorem	`PaperXI.ScalarCocycle112.Extension.sectionOf_one`
96	theorem	`PaperXI.ScalarCocycle112.Extension.sectionOf_mul`
104	theorem	`PaperXI.ScalarCocycle112.symmetric_exists_cochain`
123	def	`PaperXI.ScalarCocycle112.ratio`
131	theorem	`PaperXI.ScalarCocycle112.ratio_symmetric`
141	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain_of_divisible`
150	theorem	`PaperXI.ScalarCocycle112.complex_units_exists_pow`
162	def	`PaperXI.ScalarCocycle112.complexUnitsDivisible`
171	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain`
181	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain_raw`

A.346 `PaperXI.ScalarCovariance`

Source lines: 52.

SHA-256: `4859f25f802820daa9b4d9f45b627eb9f51c9be6756a872134890a6a94da4e06`.

```
Paper XI, Proposition 2.4. F = diag(1,q,...,q^(m-1)) as an actual
linear automorphism. q is a unit. The covariance factor is q⁻¹.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
13	def	`PaperXI.ScalarCovariance.frobenius`
25	theorem	`PaperXI.ScalarCovariance.frobenius_apply`
29	theorem	`PaperXI.ScalarCovariance.covariance`
41	theorem	`PaperXI.ScalarCovariance.conjugation`
48	theorem	`PaperXI.ScalarCovariance.scalar_model`

A.347 `PaperXI.ScalarGluing`

Source lines: 167.

SHA-256: `f32ed302e2af8ac3b71237f1ac455195516c0932bb67a8d9f2c363d241b8321c`.

```
Scalar gluing on actual graph cycles, with the zero-dimensional branch retained.
```

Direct project imports: `PaperXI.GraphGauge`, `PaperXI.ProjectiveHolonomy`.

Line	Kind	Name
25	theorem	`PaperXI.ScalarGluing.cycle_operator_iff`
37	theorem	`PaperXI.ScalarGluing.cycle_independent`
46	theorem	`PaperXI.ScalarGluing.based_iff_scalar_operators`
58	theorem	`PaperXI.ScalarGluing.based_root_independent`
66	theorem	`PaperXI.ScalarGluing.cycle_units_iff`
78	theorem	`PaperXI.ScalarGluing.based_iff_scalar_units`
89	theorem	`PaperXI.ScalarGluing.unbased_iff_scalar_units`
104	theorem	`PaperXI.ScalarGluing.loops_iff_characters`
117	theorem	`PaperXI.ScalarGluing.based_iff_ratioH1_trivial`
127	theorem	`PaperXI.ScalarGluing.zero_root_gluing`
135	theorem	`PaperXI.ScalarGluing.zero_finrank_gluing`
144	theorem	`PaperXI.ScalarGluing.connected_scalar_criterion`
159	theorem	`PaperXI.ScalarGluing.connected_zero_dimensional`

A.348 `PaperXI.ScalarHolonomy`

Source lines: 152.

SHA-256: `f3309937e8a28dc3211c81306280c5e8a7c74c5c2cd944bfc884b0ad5e8cdb8a`.

Scalar loop operators and the faithfulness boundary at rank zero.

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.ScalarHolonomy.scalarHom`
20	theorem	`PaperXI.ScalarHolonomy.scalarHom_apply`
24	theorem	`PaperXI.ScalarHolonomy.scalar_conjugate`
30	theorem	`PaperXI.ScalarHolonomy.scalar_conjugate_independent`
37	theorem	`PaperXI.ScalarHolonomy.scalarHom_injective`
45	def	`PaperXI.ScalarHolonomy.IsScalar`
49	def	`PaperXI.ScalarHolonomy.character`
59	theorem	`PaperXI.ScalarHolonomy.character_spec`
63	theorem	`PaperXI.ScalarHolonomy.character_apply`
69	theorem	`PaperXI.ScalarHolonomy.character_unique`
78	theorem	`PaperXI.ScalarHolonomy.intertwines_iff`
92	theorem	`PaperXI.ScalarHolonomy.character_intertwines_iff`
107	def	`PaperXI.ScalarHolonomy.ratio`
109	theorem	`PaperXI.ScalarHolonomy.ratio_apply`
112	theorem	`PaperXI.ScalarHolonomy.ratio_eq_one_iff`
124	theorem	`PaperXI.ScalarHolonomy.ratio_apply_eq_one_iff`
131	theorem	`PaperXI.ScalarHolonomy.zero_space_operator_eq`
136	theorem	`PaperXI.ScalarHolonomy.zero_space_intertwines`
141	theorem	`PaperXI.ScalarHolonomy.zero_space_scalar_eq`
145	theorem	`PaperXI.ScalarHolonomy.zero_space_complex_not_injective`

A.349 `PaperXI.ScalarRepresentation112`

Source lines: 373.

SHA-256: `f8bad889bfb84cf03242cd2fd13dbd620d15080c126ade4b5f9bbfdb2ae7807c`.

```
Construct an action of the actual group from labelled packet operators
and an explicit scalar cochain. The section multiplication and original theta
restriction are derived from the literal quotient kernel, not supplied as
representation or lambda hypotheses. Existence of a compatible quotient map
and preservation of any required dyadic lift remain separate obligations.
```

Direct project imports: `PaperXI.GlobalHeisenbergBridge112`, `PaperXI.RepresentationStability`, `PaperXI.ScalarCocycle112`.

Line	Kind	Name
23	def	`PaperXI.ScalarRepresentation112.sectionResidue`
30	def	`PaperXI.ScalarRepresentation112.sectionScalar`
33	def	`PaperXI.ScalarRepresentation112.sectionCocycle`
36	theorem	`PaperXI.ScalarRepresentation112.sectionResidue_mul`
47	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_mul`
56	theorem	`PaperXI.ScalarRepresentation112.sectionResidue_section`
61	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_section`
66	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_one`
71	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_one_left`
76	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_one_right`
82	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_assoc`
91	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_on_subgroup`
111	def	`PaperXI.ScalarRepresentation112.packetOperator`
117	theorem	`PaperXI.ScalarRepresentation112.packetOperator_mul`
150	def	`PaperXI.ScalarRepresentation112.scalarTransport`
178	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_on_subgroup`
193	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_section`
204	theorem	`PaperXI.ScalarRepresentation112.packet_normalForm`
219	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_irreducible`
238	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_contains`
254	def	`PaperXI.ScalarRepresentation112.actualScalarCocycle`
266	def	`PaperXI.ScalarRepresentation112.sectionCommutator`
277	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_commutator`
297	theorem	`PaperXI.ScalarRepresentation112.exists_actual_action_of_equal_scalar_commutators`
334	theorem	`PaperXI.ScalarRepresentation112.bridge_to_prescribed_eta_of_equal_commutators`

A.350 `PaperXI.SchrodingerConsequences`

Source lines: 151.

SHA-256: `f0f3bb3edc8a1eeb75ebeb2d70adc089b5e324fb2647841978788425a34c3c19`.

```
Character support and actual isotropic selectors in every irreducible
representation with the prescribed nontrivial central character.
```

Direct project imports: `PaperXI.SchrodingerUniversal`, `PaperXI.HeisenbergSelector`.

Line	Kind	Name
15	def	`PaperXI.SchrodingerConsequences.restriction`

Line	Kind	Name
21	def	`PaperXI.SchrodingerConsequences.projector`
26	theorem	`PaperXI.SchrodingerConsequences.projector_idempotent`
33	theorem	`PaperXI.SchrodingerConsequences.range_eq_invariants`
41	def	`PaperXI.SchrodingerConsequences.subgroupProjector`
47	theorem	`PaperXI.SchrodingerConsequences.projector_eq_subgroupProjector`
68	theorem	`PaperXI.SchrodingerConsequences.conjugates`
79	theorem	`PaperXI.SchrodingerConsequences.trace_eq_model`
85	theorem	`PaperXI.SchrodingerConsequences.character_support`
93	theorem	`PaperXI.SchrodingerConsequences.conjugates_projector`
106	theorem	`PaperXI.SchrodingerConsequences.trace_projector`
114	theorem	`PaperXI.SchrodingerConsequences.rank_projector`
124	theorem	`PaperXI.SchrodingerConsequences.maximal_trace_one`
132	theorem	`PaperXI.SchrodingerConsequences.maximal_rank_one`
144	theorem	`PaperXI.SchrodingerConsequences.maximal_fixed_line`

A.351 `PaperXI.SchrodingerFourier`

Source lines: 77.

SHA-256: `1c9a3b9a2264e75afda601089ef6b0665e2e1f3533f7146575d747946f361125`.

```
Exact coordinate projections from a separating family of diagonal operators.
The finite elimination argument supplies the projector; orthogonality is not a hypothesis.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerFourier.finite_elimination`
49	theorem	`PaperXI.SchrodingerFourier.coordinate_projection_mem`
65	theorem	`PaperXI.SchrodingerFourier.coordinate_unit_mem`

A.352 `PaperXI.SchrodingerIrreducible`

Source lines: 88.

SHA-256: `b45be66566d924a65606b03dc54678b2bb795070e1099de79205c24a0bff555c`.

```
Irreducibility of the finite Schrödinger action from its actual translation
and dual-character multiplication operators. The proof uses coordinate
projections and spans all functions, including the zero-dimensional case.
```

Direct project imports: `PaperXI.SchrodingerFourier`, `PaperXI.SchrodingerSeparation`.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerIrreducible.invariant_eq_bot_or_top`
61	theorem	`PaperXI.SchrodingerIrreducible.irreducible_of_actions`

A.353 `PaperXI.SchrodingerOperator`

Source lines: 93.

SHA-256: `2d1478c4d0de8e6ea366020e41530e8967ad47f25516f5cd4d8cbb544d08616d`.

```
The literal Schrödinger operators in the manuscript's sign convention.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.Schrodinger.operator`
25	theorem	`PaperXI.Schrodinger.operator_apply`
29	def	`PaperXI.Schrodinger.translation`
34	theorem	`PaperXI.Schrodinger.translation_apply`
37	def	`PaperXI.Schrodinger.modulation`
49	theorem	`PaperXI.Schrodinger.modulation_apply`
52	theorem	`PaperXI.Schrodinger.operator_translation`
57	theorem	`PaperXI.Schrodinger.operator_modulation`
62	theorem	`PaperXI.Schrodinger.operator_central`
67	theorem	`PaperXI.Schrodinger.operator_one`
73	theorem	`PaperXI.Schrodinger.phase_add`
83	theorem	`PaperXI.Schrodinger.operator_mul`

A.354 `PaperXI.SchrodingerSeparation`

Source lines: 32.

SHA-256: `ea554103afc7f9badccc857e5421b5b4749f1bea3df4043f0aae400783cb8b8b`.

```
A nontrivial scalar additive character, composed with the full linear dual,
separates points of a vector space. No finiteness assumption is required.
```

Direct project imports: none.

Line	Kind	Name
15	theorem	`PaperXI.SchrodingerSeparation.exists_dual_character_ne`

A.355 `PaperXI.SchrodingerTrace`

Source lines: 88.

SHA-256: `0c523e1824a16155d9072d81a47308e828eef34cdf178595eff2ccbd8dbe7498`.

```
Character support for the literal finite function-space Schrödinger operators.
```

Direct project imports: `PaperXI.SchrodingerOperator`.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerTrace.functional_surjective`
25	theorem	`PaperXI.SchrodingerTrace.functional_character_ne_one`
38	theorem	`PaperXI.SchrodingerTrace.sum_functional_character`
43	theorem	`PaperXI.SchrodingerTrace.operator_diagonal`
58	theorem	`PaperXI.SchrodingerTrace.trace_of_translation_ne_zero`
66	theorem	`PaperXI.SchrodingerTrace.trace_of_translation_zero`
74	theorem	`PaperXI.SchrodingerTrace.character_support`

A.356 PaperXI.SchrodingerUnitary

Source lines: 45.

SHA-256: 90bc68223666d16ea0fd539caf1719dc4d7a73e4774eadd64d5136507b4e40cb.

The literal Schrödinger operators preserve the usual finite sum Hermitian pairing.

Direct project imports: PaperXI.SchrodingerOperator.

Line	Kind	Name
15	theorem	PaperXI.SchrodingerUnitary.conjugate_character
21	theorem	PaperXI.SchrodingerUnitary.conjugate_mul_character
29	theorem	PaperXI.SchrodingerUnitary.pairing_invariant

A.357 PaperXI.SchrodingerUniversal

Source lines: 217.

SHA-256: 3a45fcb361ce2045666670d9fe67cd2b7406dc7840b50f9a9cbe11c00f77c608.

The actual map from the Schrödinger model, built from a nonzero vector
fixed by all modulations, and its representation equivalence.

Direct project imports: PaperXI.HeisenbergNormalForm, PaperXI.HeisenbergFrequency, PaperXI.HeisenbergDimension, PaperXI.HeisenbergRepresentation, PaperXI.SchurScalar.

Line	Kind	Name
18	def	PaperXI.SchrodingerUniversal.basis_map
22	theorem	PaperXI.SchrodingerUniversal.basis_map_basis
28	theorem	PaperXI.SchrodingerUniversal.translation_basis
34	theorem	PaperXI.SchrodingerUniversal.modulation_basis
43	theorem	PaperXI.SchrodingerUniversal.basis_map_translation
56	theorem	PaperXI.SchrodingerUniversal.basis_map_modulation
72	theorem	PaperXI.SchrodingerUniversal.basis_map_central
83	theorem	PaperXI.SchrodingerUniversal.intertwines
103	def	PaperXI.SchrodingerUniversal.intertwiningMap
111	theorem	PaperXI.SchrodingerUniversal.intertwiningMap_ne_zero
122	def	PaperXI.SchrodingerUniversal.equiv_of_fixed
138	theorem	PaperXI.SchrodingerUniversal.exists_equiv
148	def	PaperXI.SchrodingerUniversal.equiv
155	theorem	PaperXI.SchrodingerUniversal.equiv_intertwines
166	theorem	PaperXI.SchrodingerUniversal.finiteDimensional
174	theorem	PaperXI.SchrodingerUniversal.finrank_eq
182	theorem	PaperXI.SchrodingerUniversal.finrank_eq_sqrt
191	theorem	PaperXI.SchrodingerUniversal.unique_scalar
206	theorem	PaperXI.SchrodingerUniversal.exists_equiv_finiteField

A.358 PaperXI.SchurHom

Source lines: 68.

SHA-256: b86b71d721ec121989b4959337263985da723a313f102af82e8b05a8775c35bf.

Exact Hom dimensions for actual irreducible complex representations.

Direct project imports: `PaperXI.DualityLinear`, `PaperXI.SchurScalar`.

Line	Kind	Name
14	def	`PaperXI.SchurHom.postcomposeEquiv`
25	theorem	`PaperXI.SchurHom.finrank_eq_indicator`
40	theorem	`PaperXI.SchurHom.finrank_zero_or_one`
49	theorem	`PaperXI.SchurHom.finrank_one_iff_equiv`
59	theorem	`PaperXI.SchurHom.tensor_invariants_zero_or_one`

A.359 `PaperXI.SchurScalar`

Source lines: 66.

SHA-256: `ab78b7e45ea049accb02eacca96ad12dd74812d23ea41c9186645a9b5700ee92`.

Scalar commutants and uniqueness of actual irreducible representation bridges.

Direct project imports: `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
13	theorem	`PaperXI.SchurScalar.commuting_scalar`
25	theorem	`PaperXI.SchurScalar.scalar_unique`
33	theorem	`PaperXI.SchurScalar.commuting_unit`
46	theorem	`PaperXI.SchurScalar.bridge_unique`

A.360 `PaperXI.SemisimpleDetection`

Source lines: 71.

SHA-256: `b2713fc088d1ea21bffebfd8dd05a2b93962db392825add54d953db52b00c8c8`.

Maschke separation: literal elements of C[G] are detected on simple modules.
No choice of a complete irreducible character table is assumed.

Direct project imports: `PaperXI.CharacterProjector`.

Line	Kind	Name
14	theorem	`PaperXI.SemisimpleDetection.eq_zero_of_simple_actions`
41	theorem	`PaperXI.SemisimpleDetection.eq_zero_of_irreducible_actions`
62	theorem	`PaperXI.SemisimpleDetection.eq_of_irreducible_actions`

A.361 `PaperXI.SemisimpleProjection`

Source lines: 99.

SHA-256: `dab8a3dd87a59b725c83dd9c32b853b2de9db98afc7461311017ed27ec4b21ea`.

A multiplicity-one summand of a semisimple representation is preserved by
every invariant subspace. The splitting used here is obtained from semisimplicity,
not imposed as an additional assumption on the invariant subspace.

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
19	theorem	`PaperXI.SemisimpleProjection.module_range_le_of_nonzero`
43	theorem	`PaperXI.SemisimpleProjection.module_projection_mem`
66	theorem	`PaperXI.SemisimpleProjection.range_le_of_nonzero`
87	theorem	`PaperXI.SemisimpleProjection.projection_mem`

A.362 `PaperXI.SimpleModuleClassification`

Source lines: 106.

SHA-256: `b28f26a821af96bffc99660b774cfedfbafa942365b515c8a346a771ce3b4bf3`.

```
Actual simple modules over a finite-dimensional simple complex algebra:
uniqueness, automatic finite dimensionality, and the square-root dimension.
```

Direct project imports: none.

Line	Kind	Name
17	theorem	`PaperXI.SimpleModuleClassification.equivalent`
27	theorem	`PaperXI.SimpleModuleClassification.regular_decomposition`
37	theorem	`PaperXI.SimpleModuleClassification.exists_simple_ideal`
46	theorem	`PaperXI.SimpleModuleClassification.equivalent_complex`
55	theorem	`PaperXI.SimpleModuleClassification.finiteDimensional`
64	theorem	`PaperXI.SimpleModuleClassification.finrank_pos`
70	def	`PaperXI.SimpleModuleClassification.scalarEndEquiv`
76	theorem	`PaperXI.SimpleModuleClassification.finrank_end`
81	theorem	`PaperXI.SimpleModuleClassification.finrank_sq`
103	theorem	`PaperXI.SimpleModuleClassification.finrank_sqrt`

A.363 `PaperXI.SmithMatrix106`

Source lines: 57.

SHA-256: `9104a6467adaf608eeb4b3700c5d827828b603184ed83c55516445df5f23a948`.

```
Integral left and right basis changes for an injective square matrix over
a PID. The changes of basis are retained as actual general linear matrices.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.SmithMatrix106.basisGL`
23	theorem	`PaperXI.SmithMatrix106.basisGL_apply`
27	theorem	`PaperXI.SmithMatrix106.exists_diagonal`

A.364 `PaperXI.SmoothDegreeQuotient`

Source lines: 192.

SHA-256: `77138b22ce63047224ba2ad5c14fc85e25acce77ad9ea79b7e3404e4fb684a3d`.

```
# A smooth representation quotient which retains every integer degree

The quotient kernel is the intersection of the representation kernel and
integer-degree kernel.  No finiteness of the original inertia group and no
identification with an absolute local-field Weil group is assumed here.
```

Direct project imports: `PaperXI.QuotientRepresentation`, `PaperXI.TameDegree`.

Line	Kind	Name
21	def	`PaperXI.SmoothDegreeQuotient.vectorStabilizer`
33	theorem	`PaperXI.SmoothDegreeQuotient.mem_vectorStabilizer`
40	def	`PaperXI.SmoothDegreeQuotient.IsSmooth`
44	theorem	`PaperXI.SmoothDegreeQuotient.isOpen_kernel_of_smooth`
67	def	`PaperXI.SmoothDegreeQuotient.jointKernel`
69	instance	`PaperXI.SmoothDegreeQuotient.jointKernel_normal`
74	abbrev	`PaperXI.SmoothDegreeQuotient.DegreeQuotient`
77	abbrev	`PaperXI.SmoothDegreeQuotient.quotientMap`
81	def	`PaperXI.SmoothDegreeQuotient.descendedDegree`
85	def	`PaperXI.SmoothDegreeQuotient.descendedRepresentation`
88	theorem	`PaperXI.SmoothDegreeQuotient.descendedDegree_mk`
91	theorem	`PaperXI.SmoothDegreeQuotient.descendedRepresentation_mk`
94	theorem	`PaperXI.SmoothDegreeQuotient.descendedDegree_comp`
97	theorem	`PaperXI.SmoothDegreeQuotient.descendedRepresentation_comp`
101	theorem	`PaperXI.SmoothDegreeQuotient.descended_irreducible`
106	theorem	`PaperXI.SmoothDegreeQuotient.degree_surjective_iff`
118	theorem	`PaperXI.SmoothDegreeQuotient.degreeSubgroup_comap`
126	theorem	`PaperXI.SmoothDegreeQuotient.jointKernel_descended`
135	theorem	`PaperXI.SmoothDegreeQuotient.degree_descends_iff`
150	theorem	`PaperXI.SmoothDegreeQuotient.inertia_image`
165	theorem	`PaperXI.SmoothDegreeQuotient.isOpen_jointKernel`
171	theorem	`PaperXI.SmoothDegreeQuotient.quotient_discrete`
177	theorem	`PaperXI.SmoothDegreeQuotient.quotientMap_continuous`
182	theorem	`PaperXI.SmoothDegreeQuotient.finite_descended_inertia`

A.365 `PaperXI.SmoothHomOrbit68`

Source lines: 95.

SHA-256: `87d24ea9fbb25f0394cace54953a216a8158ddded33b1f0e68f620c6a0e5c2d6`.

```
# Finite orbits on actual linear-map spaces modulo a scalar subgroup

The subgroup and quotient are literal.  Compactness is imposed on G / N, not
on G.  Finiteness of the Hom orbit is derived from smoothness and equal scalar
actions, rather than supplied as a splitting certificate.
```

Direct project imports: `PaperXI.CompactSmoothSemisimple`.

Line	Kind	Name
22	theorem	`PaperXI.SmoothHomOrbit68.linHom_smooth`
48	theorem	`PaperXI.SmoothHomOrbit68.descend_smooth`
56	theorem	`PaperXI.SmoothHomOrbit68.orbit_finite_of_compact_quotient`
69	theorem	`PaperXI.SmoothHomOrbit68.scalar_subgroup_le_linHom_kernel`

Line	Kind	Name
84	theorem	PaperXI.SmoothHomOrbit68.linHom_orbit_finite

A.366 PaperXI.SpectralGeometricFrobenius

Source lines: 184.

SHA-256: e437a81f9af9b976a3c272dbb0dbb4824e47f138852a60adbb5dc5ac9236685e.

```
# Geometric Frobenius for a finite Galois spectral extension

The integer ring is constructed from the spectral norm of the actual extension.
Valuation stability, the Hensel property, and residue finiteness are derived.
The Frobenius-lift premise is then discharged by the inherited Galois/Hensel
surjectivity theorem. No full absolute Weil group or automorphic object is asserted.
```

Direct project imports: PaperXI.HenselianGaloisResidue, PaperXI.HenselianLiftTower, PaperXI.FiniteResidueBaseChange.

Line	Kind	Name
28	def	PaperXI.SpectralGeometricFrobenius.SpectralExtension
30	instance	PaperXI.SpectralGeometricFrobenius.spectralField
31	instance	PaperXI.SpectralGeometricFrobenius.spectralAlgebra
32	instance	PaperXI.SpectralGeometricFrobenius.spectralFinite
34	instance	PaperXI.SpectralGeometricFrobenius.spectralGalois
38	def	PaperXI.SpectralGeometricFrobenius.originalFieldEquiv
73	def	PaperXI.SpectralGeometricFrobenius.integers
76	def	PaperXI.SpectralGeometricFrobenius.baseIntegers
80	theorem	PaperXI.SpectralGeometricFrobenius.mem_integers
92	theorem	PaperXI.SpectralGeometricFrobenius.integers_comap
102	theorem	PaperXI.SpectralGeometricFrobenius.automorphism_stable
116	instance	PaperXI.SpectralGeometricFrobenius.integersDVR
120	instance	PaperXI.SpectralGeometricFrobenius.integersFiniteResidue
124	instance	PaperXI.SpectralGeometricFrobenius.integersHenselian
128	abbrev	PaperXI.SpectralGeometricFrobenius.BaseResidue
129	abbrev	PaperXI.SpectralGeometricFrobenius.ExtensionResidue
140	def	PaperXI.SpectralGeometricFrobenius.geometricFrobenius
144	def	PaperXI.SpectralGeometricFrobenius.arithmeticFrobenius
147	theorem	PaperXI.SpectralGeometricFrobenius.arithmetic_apply
152	theorem	PaperXI.SpectralGeometricFrobenius.geometric_inverse_power
158	def	PaperXI.SpectralGeometricFrobenius.residueAction
164	theorem	PaperXI.SpectralGeometricFrobenius.exists_geometric_lift
172	def	PaperXI.SpectralGeometricFrobenius.geometricLift
174	theorem	PaperXI.SpectralGeometricFrobenius.residue_geometricLift
179	theorem	PaperXI.SpectralGeometricFrobenius.geometricLift_normalized

A.367 PaperXI.SpectralLocalField

Source lines: 175.

SHA-256: 96cc4d62e4fc37f3550c6800e3715c60d191bf454547ea9a67d4aeae6376f9ab.

```
# Spectral local-field structure on the constructed unramified fraction field

The upper field is finite over the given nonarchimedean local field. This file
installs its spectral norm, proves completeness and local compactness, and
packages the norm valuation as an actual nonarchimedean local-field structure.
The canonical DVR valuation remains a separate explicitly named valuation; the
comparison of the two valued structures is not hidden in an instance.
```

Direct project imports: `PaperXI.CanonicalDVRValuation`.

Line	Kind	Name
45	abbrev	`PaperXI.SpectralLocalField.UpperField`
48	instance	`PaperXI.SpectralLocalField.upperNormedField`
52	instance	`PaperXI.SpectralLocalField.upperNontriviallyNormedField`
57	instance	`PaperXI.SpectralLocalField.upperNormedSpace`
61	instance	`PaperXI.SpectralLocalField.upperNormedAlgebra`
65	theorem	`PaperXI.SpectralLocalField.norm_algebraMap`
70	theorem	`PaperXI.SpectralLocalField.norm_eq_spectralNorm`
74	theorem	`PaperXI.SpectralLocalField.algebraMap_isometry`
81	instance	`PaperXI.SpectralLocalField.upperUltrametric`
93	instance	`PaperXI.SpectralLocalField.upperComplete`
97	instance	`PaperXI.SpectralLocalField.upperProper`
101	instance	`PaperXI.SpectralLocalField.upperLocallyCompact`
105	instance	`PaperXI.SpectralLocalField.upperNormValued`
109	instance	`PaperXI.SpectralLocalField.upperValuativeRel`
112	instance	`PaperXI.SpectralLocalField.upperNormValuationCompatible`
117	instance	`PaperXI.SpectralLocalField.upperIsValuativeTopology`
124	instance	`PaperXI.SpectralLocalField.upperValuativeNontrivial`
131	theorem	`PaperXI.SpectralLocalField.relation_integer_eq_norm_integer`
143	theorem	`PaperXI.SpectralLocalField.mem_integer_iff_norm_le_one`
150	instance	`PaperXI.SpectralLocalField.upperIsNonarchimedeanLocalField`
157	instance	`PaperXI.SpectralLocalField.upperFiniteResidue`
161	instance	`PaperXI.SpectralLocalField.upperIntegerIsDVR`
166	def	`PaperXI.SpectralLocalField.upperValueGroupIsoInt`
171	theorem	`PaperXI.SpectralLocalField.finrank_eq_residue_finrank`

A.368 `PaperXI.SplitRadical105`

Source lines: 152.

SHA-256: `bbe4cd3ed2985df636cc326b7fceac83a5c48fec8b393ba3ae4a8da6e7fd5ca7`.

```
Literal square-zero coordinate radicals, entrywise lifting, and contraction
by a normalized Cartan diagonal. Neither a contracting subgroup nor its lifts
are supplied to the matrix theorem.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.SplitRadical105.rect`
20	theorem	`PaperXI.SplitRadical105.rect_mul`
28	theorem	`PaperXI.SplitRadical105.rect_add`
34	def	`PaperXI.SplitRadical105.unip`
40	theorem	`PaperXI.SplitRadical105.unip_add`

Line	Kind	Name
48	theorem	`PaperXI.SplitRadical105.unip_zero`
53	def	`PaperXI.SplitRadical105.radicalMap`
58	theorem	`PaperXI.SplitRadical105.map_unip`
67	def	`PaperXI.SplitRadical105.diagonalGL`
79	def	`PaperXI.SplitRadical105.cartan`
82	def	`PaperXI.SplitRadical105.zeroSet`
85	theorem	`PaperXI.SplitRadical105.mem_zeroSet`
89	theorem	`PaperXI.SplitRadical105.cartan_conjugation`
115	theorem	`PaperXI.SplitRadical105.contracted_reduction`
129	theorem	`PaperXI.SplitRadical105.exists_contracted_lift`
146	theorem	`PaperXI.SplitRadical105.cartan_zero`

A.369 `PaperXI.SymplecticAssembly`

Source lines: 51.

SHA-256: `67819fec0a0969ee2375d7ee93287a9f1e91d60e5da87e7a12aee6013e3e0d94`.

```
Adding one hyperbolic plane to an existing symplectic coordinate system.
```

Direct project imports: `PaperXI.DyadicCoordinates`.

Line	Kind	Name
11	def	`PaperXI.SymplecticAssembly.prepend`
26	theorem	`PaperXI.SymplecticAssembly.prepend_form`
35	def	`PaperXI.SymplecticAssembly.assemble`
40	theorem	`PaperXI.SymplecticAssembly.assemble_form`

A.370 `PaperXI.SymplecticCentralExtension`

Source lines: 75.

SHA-256: `b54d1f3d499676466c42f50e7b1fff895c0dd532d855652a51347f5bff12c61d`.

```
A nontrivial additive character detects every nonzero vector through a
left-nondegenerate field-valued bilinear commutator. This supplies scalar
nondegeneracy for actual symplectic central extensions in every characteristic.
```

Direct project imports: `PaperXI.CentralExtensionStructure`.

Line	Kind	Name
17	def	`PaperXI.SymplecticCentralExtension.scalarCharacter`
20	theorem	`PaperXI.SymplecticCentralExtension.scalarCharacter_apply`
23	theorem	`PaperXI.SymplecticCentralExtension.scalarCharacter_ofAdd`
28	theorem	`PaperXI.SymplecticCentralExtension.character_separating_left`
45	theorem	`PaperXI.SymplecticCentralExtension.radical_eq_bot`
66	theorem	`PaperXI.SymplecticCentralExtension.radical_eq_bot_scalarCharacter`

A.371 PaperXI.SymplecticCoordinates

Source lines: 70.

SHA-256: `b9b5ab88baf7b682acac5698d8cd558de3acf406eaec991f76a8b16e63500ecf`.

```
Symplectic coordinates constructed from alternation and nondegeneracy,
by strong induction on the dimension.
```

Direct project imports: `PaperXI.SymplecticSplit`, `PaperXI.SymplecticAssembly`, `PaperXI.DyadicDimension`.

Line	Kind	Name
17	theorem	`PaperXI.SymplecticCoordinates.exists_coordinates`
45	theorem	`PaperXI.SymplecticCoordinates.exists_coordinates_dimension`
54	theorem	`PaperXI.SymplecticCoordinates.even_finrank`
61	theorem	`PaperXI.SymplecticCoordinates.exists_half_coordinates`

A.372 PaperXI.SymplecticHeisenbergUniqueness

Source lines: 85.

SHA-256: `e54d8758376e52f844dd630abfdd19a6f940709729fabf3449bfa5f755195104`.

```
Finite Stone–von Neumann from an actual field-valued nondegenerate
commutator and a nontrivial additive character. This applies to supplied actual
central extensions in every characteristic, without assuming scalar
nondegeneracy, target dimensions, or a first intertwiner.
```

Direct project imports: `PaperXI.SymplecticCentralExtension`, `PaperXI.CentralExtensionExistence`.

Line	Kind	Name
19	theorem	`PaperXI.SymplecticHeisenbergUniqueness.central_action`
35	theorem	`PaperXI.SymplecticHeisenbergUniqueness.finiteDimensional`
42	theorem	`PaperXI.SymplecticHeisenbergUniqueness.finrank_sqrt`
51	theorem	`PaperXI.SymplecticHeisenbergUniqueness.exists_equiv`
61	theorem	`PaperXI.SymplecticHeisenbergUniqueness.card_eq_square`
69	theorem	`PaperXI.SymplecticHeisenbergUniqueness.exists_irreducible`

A.373 PaperXI.SymplecticSplit

Source lines: 126.

SHA-256: `66521ce6aa9e9ca4cbf9bc05aad9112ad95e36069f2d3d718f214beb6515bc05`.

```
A constructive hyperbolic-plane splitting of a nondegenerate alternating
form over F₂. The complement and both directions of the coordinate map are
built explicitly from one hyperbolic pair.
```

Direct project imports: `PaperXI.DyadicGauss`.

Line	Kind	Name
17	theorem	`PaperXI.SymplecticSplit.alternating_symmetric`
24	theorem	`PaperXI.SymplecticSplit.exists_pair`

Line	Kind	Name
35	theorem	`PaperXI.SymplecticSplit.split_of_pair`
113	theorem	`PaperXI.SymplecticSplit.exists_split`

A.374 `PaperXI.TameConductor`

Source lines: 113.

SHA-256: `808eb443d2a292fc79cd70ce323c369acd62c38e9393ec4c19269dc06d575dcf`.

```
The nontrivial tame conductor calculation, with its finite-quotient
attachment explicit. These definitions encode finite ramification sums. They
do not construct local fields, Weil groups or ramification filtrations.
```

Direct project imports: `PaperXI.PacketBasis`, `PaperXI.InertiaFixed`, `PaperXI.InertiaQuotient`.

Line	Kind	Name
21	theorem	`PaperXI.TameConductor.zero_label_period`
26	theorem	`PaperXI.TameConductor.inertia_invariants_eq_bot`
40	theorem	`PaperXI.TameConductor.inertia_finrank_zero`
51	def	`PaperXI.TameConductor.fixedCodimension`
56	def	`PaperXI.TameConductor.finiteSwan`
61	theorem	`PaperXI.TameConductor.invariants_eq_top_of_trivial_on`
68	theorem	`PaperXI.TameConductor.fixedCodimension_zero_of_trivial_on`
76	theorem	`PaperXI.TameConductor.finiteSwan_zero`
88	def	`PaperXI.TameConductor.finiteArtin`
95	theorem	`PaperXI.TameConductor.conductor_of_nontrivial_tame_atom`

A.375 `PaperXI.TameDegree`

Source lines: 170.

SHA-256: `868e2f79b22a754bc070c23805f852b2c90a71cb802703bea2eaad248f39ad92`.

```
Degree subgroups and their actual cyclic translation actions.
```

Direct project imports: `PaperXI.TameTensorOrbits`.

Line	Kind	Name
13	def	`PaperXI.TameDegree.degree`
15	theorem	`PaperXI.TameDegree.degree_one`
17	theorem	`PaperXI.TameDegree.degree_mul`
20	theorem	`PaperXI.TameDegree.degree_zpow`
23	theorem	`PaperXI.TameDegree.degree_inv`
26	theorem	`PaperXI.TameDegree.degree_conjugate`
30	def	`PaperXI.TameDegree.degreeMod`
35	theorem	`PaperXI.TameDegree.degreeMod_apply`
39	def	`PaperXI.TameDegree.degreeSubgroup`
41	theorem	`PaperXI.TameDegree.mem_degreeSubgroup`
46	theorem	`PaperXI.TameDegree.degreeSubgroup_inter`
54	theorem	`PaperXI.TameDegree.degreeSubgroup_le`

Line	Kind	Name
61	def	PaperXI.TameDegree.conjugationInto
69	theorem	PaperXI.TameDegree.conjugationInto_apply
73	def	PaperXI.TameDegree.translationAction
83	theorem	PaperXI.TameDegree.translationAction_apply
86	theorem	PaperXI.TameDegree.translationAction_fixed_iff
92	def	PaperXI.TameDegree.pairAction
98	theorem	PaperXI.TameDegree.pairAction_apply
102	theorem	PaperXI.TameDegree.pairAction_fixed_iff
107	def	PaperXI.TameDegree.blockAction
113	theorem	PaperXI.TameDegree.blockAction_apply
117	theorem	PaperXI.TameDegree.orbitMap_intertwines
123	theorem	PaperXI.TameDegree.orbitEquiv_intertwines
132	theorem	PaperXI.TameDegree.degree_phi
135	theorem	PaperXI.TameDegree.degree_phi_zpow
139	theorem	PaperXI.TameDegree.degreeMod_phi_zpow
144	theorem	PaperXI.TameDegree.degreeMod_surjective
152	theorem	PaperXI.TameDegree.translationAction_phi_zpow
157	theorem	PaperXI.TameDegree.pairAction_phi_zpow
163	theorem	PaperXI.TameDegree.fiber_transitive

A.376 PaperXI.TameExponent

Source lines: 117.

SHA-256: 738e3f3fe21ccb970422c7dc2b0de7f7780e588ba208153af7e04e70415b767e.

```
Faithful exponent coordinates for genuine finite-field unit characters.
```

Direct project imports: PaperXI.TameNorm, PaperXI.ElementaryEnvelope.

Line	Kind	Name
11	def	PaperXI.TameExponent.rootEmbedding
22	theorem	PaperXI.TameExponent.rootEmbedding_injective
32	def	PaperXI.TameExponent.faithfulCharacter
36	theorem	PaperXI.TameExponent.faithfulCharacter_injective
44	theorem	PaperXI.TameExponent.pow_eq_one_iff
63	theorem	PaperXI.TameExponent.pow_add_card
73	theorem	PaperXI.TameExponent.product
77	theorem	PaperXI.TameExponent.exists_exponent
92	theorem	PaperXI.TameExponent.pow_eq_pow_iff
102	def	PaperXI.TameExponent.coordinate
105	theorem	PaperXI.TameExponent.coordinate_bijective

A.377 PaperXI.TameGaussNaturality

Source lines: 79.

SHA-256: 85bd64090e77888f06f233d6ea04a3cee2a539e3341988dd95273b8cdf5c1f7b.

```
Additive-character normalization is part of the finite comparison data.
These statements concern the literal finite Gauss sum, not a full local epsilon
factor. A nontrivial residue character is required for a unit-norm phase.
```

Direct project imports: `PaperXI.GaussCharacters`.

Line	Kind	Name
14	def	`PaperXI.TameGaussNaturality.pullMul`
18	def	`PaperXI.TameGaussNaturality.pullAdd`
22	theorem	`PaperXI.TameGaussNaturality.sum_transport`
27	theorem	`PaperXI.TameGaussNaturality.phase_transport`
34	theorem	`PaperXI.TameGaussNaturality.phase_of_stable_additive`
40	theorem	`PaperXI.TameGaussNaturality.phase_mulShift`
49	theorem	`PaperXI.TameGaussNaturality.phase_trivial`
59	theorem	`PaperXI.TameGaussNaturality.phase_norm_cases`
71	theorem	`PaperXI.TameGaussNaturality.phase_ne_zero`

A.378 `PaperXI.TameHomCriterion`

Source lines: 66.

SHA-256: `d125abe9165991cb809454306c36d0fa17a486c198612b7ddb05b4dd3cb35d35`.

```
Corollary 5.7: the actual tensor fixed space is the actual Hom space.
The refined packet data are precisely the inherited Proposition 5.6 contract.
The 0/1 bound is derived from irreducibility, not supplied with the packets.
```

Direct project imports: `PaperXI.SchurHom`, `PaperXI.TensorFixed`.

Line	Kind	Name
20	theorem	`PaperXI.TameHomCriterion.hom_finrank_eq_trivial_count`
40	theorem	`PaperXI.TameHomCriterion.tame_hom_test`

A.379 `PaperXI.TameNorm`

Source lines: 102.

SHA-256: `249939314ef4f5eff217c783fa013cb8fe51f659337e9f7db4ec9a22d2a11c05`.

```
Paper XI, 5.3: norm lifting and Frobenius on actual finite-field unit characters.
The characters on subfields are restrictions of one character of the common field.
This is the compatible choice of exponent coordinates used by the source formula.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.TameNorm.embedding`
18	theorem	`PaperXI.TameNorm.embedding_injective`
24	def	`PaperXI.TameNorm.norm`
27	def	`PaperXI.TameNorm.normExponent`
29	theorem	`PaperXI.TameNorm.embedding_norm`
35	theorem	`PaperXI.TameNorm.norm_lift`
44	theorem	`PaperXI.TameNorm.norm_lift_eq_one_iff`
58	def	`PaperXI.TameNorm.frobenius`
61	theorem	`PaperXI.TameNorm.frobenius_apply`

Line	Kind	Name
68	theorem	`PaperXI.TameNorm.frobenius_character`
76	theorem	`PaperXI.TameNorm.shifted_norm_lift`
88	theorem	`PaperXI.TameNorm.norm_frobenius`
93	theorem	`PaperXI.TameNorm.shifted_lower_norm_lift`

A.380 `PaperXI.TameNormPacket`

Source lines: 118.

SHA-256: `d19c32f222a6fd3f565bce086da6baf95a562f7266ec7a0105afefeb54a3a071`.

```
Paper XI, Proposition 5.3. The interaction character is built from actual
finite-field norms, the actual Frobenius, and multiplication of characters.
Exponent coordinates on the two subfields come from one faithful common-field
character. Every pair of characters admits these compatible coordinates.
```

Direct project imports: `PaperXI.TameExponent`.

Line	Kind	Name
17	def	`PaperXI.TameNormPacket.interaction`
23	theorem	`PaperXI.TameNormPacket.refined_exponent`
35	theorem	`PaperXI.TameNormPacket.refined_trivial_iff`
46	theorem	`PaperXI.TameNormPacket.source_formula`
58	theorem	`PaperXI.TameNormPacket.source_trivial_iff`
71	theorem	`PaperXI.TameNormPacket.restricted_faithful`
80	theorem	`PaperXI.TameNormPacket.exists_compatible_exponents`
93	theorem	`PaperXI.TameNormPacket.source`

A.381 `PaperXI.TameOrbitTransitivity`

Source lines: 110.

SHA-256: `e2f06470f27d1f2d54ee8573c71fdb9e4dd3a4f5526cf551c225e087f94f7971`.

```
The orbit-transitivity part of Theorem 3.2 for an actual irreducible group
representation with explicitly generating tame inertia and Frobenius.
```

Direct project imports: `PaperXI.OrbitSpan`, `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
15	abbrev	`PaperXI.TameOrbitTransitivity.tameRestriction`
18	def	`PaperXI.TameOrbitTransitivity.generators`
23	theorem	`PaperXI.TameOrbitTransitivity.geometric_relation`
38	theorem	`PaperXI.TameOrbitTransitivity.generators_preserve_orbitSpan`
58	def	`PaperXI.TameOrbitTransitivity.orbitSubrepresentation`
68	theorem	`PaperXI.TameOrbitTransitivity.orbitSpan_eq_top`
80	theorem	`PaperXI.TameOrbitTransitivity.nonzero_packets_one_orbit`
92	theorem	`PaperXI.TameOrbitTransitivity.exists_nonzero_packet`

A.382 PaperXI.TameProjectors

Source lines: 86.

SHA-256: d5881299706cbe60a5cbefc9dc60bb9bdffbc1365228529c181b9a5120b71d8e.

```
Proposition 3.1 over C, with primitivity and the nonzero denominator proved.
Both Frobenius conventions are stated explicitly, without changing the operators.
```

Direct project imports: PaperXI.FourierPackets.

Line	Kind	Name
14	def	PaperXI.TameProjectors.projector
17	theorem	PaperXI.TameProjectors.idempotent
22	theorem	PaperXI.TameProjectors.orthogonal
27	theorem	PaperXI.TameProjectors.complete
32	theorem	PaperXI.TameProjectors.range_eq_character_space
37	theorem	PaperXI.TameProjectors.forward_generator_covariance
45	theorem	PaperXI.TameProjectors.geometric_covariance
54	theorem	PaperXI.TameProjectors.character_weight
61	theorem	PaperXI.TameProjectors.standard_root
66	theorem	PaperXI.TameProjectors.manuscript_formula
77	theorem	PaperXI.TameProjectors.tame_projector_theorem

A.383 PaperXI.TameTensorInduction

Source lines: 111.

SHA-256: 148b5e2d70afcc38a07c5b9fe8167fd690d2f687def771b465c1a8b81fbb799e.

```
Actual induction of the interaction packets and the tensor decomposition.
```

Direct project imports: PaperXI.TameTensorPackets, PaperXI.MonomialInductionSubgroup.

Line	Kind	Name
20	def	PaperXI.TameTensorInduction.character
39	theorem	PaperXI.TameTensorInduction.character_apply
48	theorem	PaperXI.TameTensorInduction.cycle_transitive
55	def	PaperXI.TameTensorInduction.packetEquivalence
70	def	PaperXI.TameTensorInduction.directSumEquiv
86	def	PaperXI.TameTensorInduction.decomposition
95	def	PaperXI.TameTensorInduction.decompositionWithCharacters

A.384 PaperXI.TameTensorOrbits

Source lines: 210.

SHA-256: 9485733d69ff7a1333197a690fb0ae38421460f342371510df530db8047753ec.

```
Actual simultaneous-translation orbits used in the tame tensor decomposition.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.TameTensorOrbits.diagonal`
16	theorem	`PaperXI.TameTensorOrbits.diagonal_injective`
19	theorem	`PaperXI.TameTensorOrbits.diagonal_intCast`
24	theorem	`PaperXI.TameTensorOrbits.diagonal_natCast`
30	def	`PaperXI.TameTensorOrbits.difference`
34	theorem	`PaperXI.TameTensorOrbits.difference_diagonal`
39	theorem	`PaperXI.TameTensorOrbits.difference_add`
45	def	`PaperXI.TameTensorOrbits.orbitMap`
48	theorem	`PaperXI.TameTensorOrbits.orbitMap_apply`
53	theorem	`PaperXI.TameTensorOrbits.orbitMap_zero`
57	theorem	`PaperXI.TameTensorOrbits.difference_orbitMap`
62	theorem	`PaperXI.TameTensorOrbits.orbitMap_injective`
77	instance	`PaperXI.TameTensorOrbits.lcm_neZero`
80	instance	`PaperXI.TameTensorOrbits.gcd_neZero`
83	theorem	`PaperXI.TameTensorOrbits.orbitMap_bijective`
90	def	`PaperXI.TameTensorOrbits.orbitEquiv`
93	theorem	`PaperXI.TameTensorOrbits.orbitEquiv_apply`
97	theorem	`PaperXI.TameTensorOrbits.orbitMap_translate`
103	def	`PaperXI.TameTensorOrbits.translate`
107	theorem	`PaperXI.TameTensorOrbits.orbitMap_translate_int`
116	theorem	`PaperXI.TameTensorOrbits.translate_eq_iff`
124	theorem	`PaperXI.TameTensorOrbits.interaction_count`
127	theorem	`PaperXI.TameTensorOrbits.interaction_length`
131	theorem	`PaperXI.TameTensorOrbits.total_dimension`
135	theorem	`PaperXI.TameTensorOrbits.difference_translate`
140	theorem	`PaperXI.TameTensorOrbits.sameOrbit_iff`
159	theorem	`PaperXI.TameTensorOrbits.representatives_sameOrbit_iff`
173	def	`PaperXI.TameTensorOrbits.fiberEquiv`
191	theorem	`PaperXI.TameTensorOrbits.fiber_card`
196	theorem	`PaperXI.TameTensorOrbits.fiberEquiv_apply`
200	theorem	`PaperXI.TameTensorOrbits.orbitEquiv_symm_translate`

A.385 `PaperXI.TameTensorPackets`

Source lines: 81.

SHA-256: `507e433f56fdae07efb7824aa8823c182dd07364260ea7b9d594f003e2514334`.

```
Actual gcd/lcm decomposition of tensor representations with cyclic monomial
bases. The bases may belong to nonregular or reducible induced representations.
```

Direct project imports: `PaperXI.MonomialOrbit`, `PaperXI.TameDegree`.

Line	Kind	Name
21	def	`PaperXI.TameTensorPackets.coefficient`
26	theorem	`PaperXI.TameTensorPackets.tensor_action`
33	theorem	`PaperXI.TameTensorPackets.orbit_covariance`
39	def	`PaperXI.TameTensorPackets.packet`
46	def	`PaperXI.TameTensorPackets.decomposition`
54	theorem	`PaperXI.TameTensorPackets.packet_basis`

Line	Kind	Name
61	theorem	`PaperXI.TameTensorPackets.packet_zero`
70	theorem	`PaperXI.TameTensorPackets.packet_count`
73	theorem	`PaperXI.TameTensorPackets.packet_dimension`
76	theorem	`PaperXI.TameTensorPackets.dimension_sum`

A.386 `PaperXI.TameTensorSource`

Source lines: 122.

SHA-256: `b52fef0aac236edae7a22103dcf267e147fa13606783529fa64e3c4c4e954d3e`.

```
Source-facing tame tensor decomposition of genuine induced representations.
The Frobenius convention uses representatives `φ^(-i)` and hence the shifted
character `h ↦ η(φ^(-i) h φ^i)`.
```

Direct project imports: `PaperXI.CyclicInducedCharacter`, `PaperXI.TameTensorInduction`.

Line	Kind	Name
18	def	`PaperXI.TameTensorSource.interactionCharacter`
25	theorem	`PaperXI.TameTensorSource.interactionCharacter_apply`
33	theorem	`PaperXI.TameTensorSource.interactionCharacter_literal`
51	theorem	`PaperXI.TameTensorSource.representative_val`
56	theorem	`PaperXI.TameTensorSource.interactionCharacter_weight`
66	def	`PaperXI.TameTensorSource.decomposition`
81	theorem	`PaperXI.TameTensorSource.decomposition_intertwines`
93	theorem	`PaperXI.TameTensorSource.interaction_dimension`
101	theorem	`PaperXI.TameTensorSource.interaction_count`
106	theorem	`PaperXI.TameTensorSource.interaction_dimension_sum`
116	theorem	`PaperXI.TameTensorSource.tensor_dimension`

A.387 `PaperXI.TensorFixed`

Source lines: 34.

SHA-256: `26c3d415523844b3389e5d7054e76b719d6d67574ac4fda6f67934a3be0c1f41`.

```
Proposition 5.6 for an actual representation with supplied refined packets.
```

Direct project imports: `PaperXI.InvariantAssembly`.

Line	Kind	Name
15	theorem	`PaperXI.TensorFixed.count_trivial_packets`

A.388 `PaperXI.TensorIntertwining107`

Source lines: 106.

SHA-256: `4bb9b42f6f58022b82ae78cc0fe2d83d0aa6a153c7fc03aed66dd2ba7ea6f095`.

```
The operator argument underlying BK 5.3.2. The arithmetic support and
one-dimensional Heisenberg-intertwining results are kept as explicit inputs.
```

Direct project imports: `PaperXI.TensorSlices107`, `PaperXI.IntertwiningAlong105`.

Line	Kind	Name
20	theorem	`PaperXI.TensorIntertwining107.slice_along`
36	theorem	`PaperXI.TensorIntertwining107.restrict_support`
45	theorem	`PaperXI.TensorIntertwining107.second_along`
62	theorem	`PaperXI.TensorIntertwining107.map_along`
76	theorem	`PaperXI.TensorIntertwining107.factor_along`
88	theorem	`PaperXI.TensorIntertwining107.existsAt_iff`

A.389 `PaperXI.TensorIrreducible`

Source lines: 78.

SHA-256: `f75b3acfdfeba7e36d31c8d0085caffec072b5c712c4b76f407d8a0b65d5f721`.

```
A dimension criterion for irreducibility of an actual central-character
representation. Simplicity proves injectivity of its actual algebra action;
the dimension identity then proves surjectivity and irreducibility.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`.

Line	Kind	Name
18	theorem	`PaperXI.TensorIrreducible.finrank_sq_of_irreducible`
32	theorem	`PaperXI.TensorIrreducible.action_injective`
42	theorem	`PaperXI.TensorIrreducible.action_surjective`
58	theorem	`PaperXI.TensorIrreducible.irreducible`

A.390 `PaperXI.TensorPolynomial`

Source lines: 89.

SHA-256: `45345b4d384aa9e69de8e139f5f1cd75cc18a97728cf3ba6d5d6f53413956ca7`.

```
Corollary 5.5: actual inertia determinant and rational local-factor products.
```

Direct project imports: `PaperXI.PolynomialAssembly`.

Line	Kind	Name
13	theorem	`PaperXI.TensorPolynomial.quotient_stable`
22	theorem	`PaperXI.TensorPolynomial.restricted_eq_factored`
33	theorem	`PaperXI.TensorPolynomial.cyclePolynomial_ne_zero`
56	theorem	`PaperXI.TensorPolynomial.total_stable`
63	theorem	`PaperXI.TensorPolynomial.polynomial_packets`
75	theorem	`PaperXI.TensorPolynomial.determinant_packets`
82	theorem	`PaperXI.TensorPolynomial.localFactor_packets`

A.391 `PaperXI.TensorSlices107`

Source lines: 115.

SHA-256: `0176d323e1f5d380a8aac25c784710e8b877b7d4aca3fa718eb098f4f08b954b`.

```
Slice an actual tensor operator and reconstruct it when all its first-factor
slices belong to one nonzero line. No invertibility of that line's generator,
and no finite-dimensionality of either carrier, is assumed.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.TensorSlices107.right`
17	def	`PaperXI.TensorSlices107.left`
20	theorem	`PaperXI.TensorSlices107.right_tmul`
23	theorem	`PaperXI.TensorSlices107.left_tmul`
26	theorem	`PaperXI.TensorSlices107.right_ext`
35	theorem	`PaperXI.TensorSlices107.contractions_commute`
42	theorem	`PaperXI.TensorSlices107.right_map`
49	def	`PaperXI.TensorSlices107.slice`
53	theorem	`PaperXI.TensorSlices107.slice_apply`
57	theorem	`PaperXI.TensorSlices107.exists_nonzero_slice`
71	theorem	`PaperXI.TensorSlices107.factor_of_slice_line`
97	theorem	`PaperXI.TensorSlices107.tmul_left_injective`
104	theorem	`PaperXI.TensorSlices107.map_left_injective`

A.392 `PaperXI.TensorSubspace`

Source lines: 76.

SHA-256: `e65d7a7aea5e82ccccba7c606a3a314c89834d52f16441aa2d64c184362a5038`.

```
Tensoring a multiplicity subspace with a nonzero finite-dimensional
factor preserves whether it is zero or the whole space. The actual image
is transported through a specified, independently proved tensor equivalence.
```

Direct project imports: none.

Line	Kind	Name
18	def	`PaperXI.TensorSubspace.image`
21	theorem	`PaperXI.TensorSubspace.inclusion_injective`
25	theorem	`PaperXI.TensorSubspace.map_injective`
29	theorem	`PaperXI.TensorSubspace.tmul_mem`
34	theorem	`PaperXI.TensorSubspace.image_stable`
48	theorem	`PaperXI.TensorSubspace.finrank_image`
55	theorem	`PaperXI.TensorSubspace.image_eq_bot_iff`
61	theorem	`PaperXI.TensorSubspace.image_eq_top_iff`

A.393 `PaperXI.TensorTypeOperators70`

Source lines: 94.

SHA-256: `e82f0e46bab6ff2d1096ab07053553ddb6218bf13ff00afc296a84e2af2b270c`.

```
Faithful first-factor operator realization in the literal inducing-type image.
The multiplicity space must be nonzero. Neither tensor amplification nor the
ambient induced space is silently identified with the first-factor carrier.
```

Direct project imports: `PaperXI.InducedTypeSpace68`.

Line	Kind	Name
17	def	`PaperXI.TensorTypeOperators70.amplify`
20	theorem	`PaperXI.TensorTypeOperators70.amplify_tmul`
24	theorem	`PaperXI.TensorTypeOperators70.amplify_injective`
37	def	`PaperXI.TensorTypeOperators70.realize`
41	theorem	`PaperXI.TensorTypeOperators70.realize_tmul`
50	theorem	`PaperXI.TensorTypeOperators70.realize_injective`
53	theorem	`PaperXI.TensorTypeOperators70.realize_eq_iff`
56	theorem	`PaperXI.TensorTypeOperators70.realize_zero_iff`
61	theorem	`PaperXI.TensorTypeOperators70.residual_zero_iff`
66	theorem	`PaperXI.TensorTypeOperators70.projector_iff`
71	theorem	`PaperXI.TensorTypeOperators70.intertwining_iff`
77	theorem	`PaperXI.TensorTypeOperators70.word_product`
84	theorem	`PaperXI.TensorTypeOperators70.holonomy_iff`
89	theorem	`PaperXI.TensorTypeOperators70.weighted_sum`

A.394 `PaperXI.TraceDiscrepancy`

Source lines: 84.

SHA-256: `0bdba459861ec52105d2a28d2acdd6d4f0f5ac5d7a55cd93ddff6186ed822e15`.

```
A checked normalization route from actual operator traces to discrepancy.
Nonzero normalizing traces are essential and are retained in every cancellation.
```

Direct project imports: `PaperXI.CharacterGenerators`.

Line	Kind	Name
14	theorem	`PaperXI.TraceDiscrepancy.conjugate_eq`
20	theorem	`PaperXI.TraceDiscrepancy.trace_eq`
26	theorem	`PaperXI.TraceDiscrepancy.scalar_eq_one`
35	theorem	`PaperXI.TraceDiscrepancy.normalized_trace_eq`
41	theorem	`PaperXI.TraceDiscrepancy.scalar_eq_one_of_normalized`
51	theorem	`PaperXI.TraceDiscrepancy.character_trivial`
63	theorem	`PaperXI.TraceDiscrepancy.zero_trace_cannot_detect_scalar`
67	theorem	`PaperXI.TraceDiscrepancy.bridge_from_traces`

A.395 `PaperXI.TranslationAverage`

Source lines: 102.

SHA-256: `7b04c80800e59fc6ced892041eb4fe9306e01e08e18e61e79d27c07acefe9d1e`.

```
The translation representation and its rank-one constant-function projector.
```

Direct project imports: `PaperXI.FiniteAverage`.

Line	Kind	Name
13	def	`PaperXI.TranslationAverage.translation`
22	theorem	`PaperXI.TranslationAverage.translation_apply`
25	def	`PaperXI.TranslationAverage.mean`
27	def	`PaperXI.TranslationAverage.constantSubmodule`
31	theorem	`PaperXI.TranslationAverage.mem_constantSubmodule`
39	theorem	`PaperXI.TranslationAverage.average_apply`
48	theorem	`PaperXI.TranslationAverage.range_eq`
61	theorem	`PaperXI.TranslationAverage.finrank_range`
71	def	`PaperXI.TranslationAverage.pairing`
73	theorem	`PaperXI.TranslationAverage.pairing_translation`
79	theorem	`PaperXI.TranslationAverage.pairing_self_adjoint`
91	theorem	`PaperXI.TranslationAverage.residual_orthogonal`

A.396 `PaperXI.TwistedAction`

Source lines: 79.

SHA-256: `97ed5ee03bec3fc432018d895c823299a7cc73ddf65cb1b712417d9948a5eb71`.

```
Universal extension of actual projective operators to an action of the
constructed twisted group algebra.
```

Direct project imports: `PaperXI.TwistedAlgebra`.

Line	Kind	Name
15	def	`PaperXI.TwistedAction.linearExtension`
19	theorem	`PaperXI.TwistedAction.linearExtension_basis`
24	theorem	`PaperXI.TwistedAction.linearExtension_one`
29	theorem	`PaperXI.TwistedAction.linearExtension_mul`
44	def	`PaperXI.TwistedAction.action`
51	theorem	`PaperXI.TwistedAction.action_basis`
58	theorem	`PaperXI.TwistedAction.action_unique`
70	theorem	`PaperXI.TwistedAction.existsUnique_action`

A.397 `PaperXI.TwistedActionEquivalence`

Source lines: 107.

SHA-256: `6a14cac5426c7fb2c44d091685da2d07440c4b883f7409f7caff5c2c38919e74`.

```
The actual algebra action of an irreducible projective operator family is
an isomorphism when its scalar commutator is nondegenerate.
```

Direct project imports: `PaperXI.TwistedAction`, `PaperXI.TwistedModules`, `PaperXI.AlgebraActionModule`.

Line	Kind	Name
21	theorem	`PaperXI.TwistedActionEquivalence.stable_iff`
34	theorem	`PaperXI.TwistedActionEquivalence.simpleModule`
47	theorem	`PaperXI.TwistedActionEquivalence.finiteDimensional`
58	theorem	`PaperXI.TwistedActionEquivalence.finrank_sq`

Line	Kind	Name
66	theorem	`PaperXI.TwistedActionEquivalence.finrank_sqrt`
71	theorem	`PaperXI.TwistedActionEquivalence.action_injective`
77	theorem	`PaperXI.TwistedActionEquivalence.action_surjective`
92	theorem	`PaperXI.TwistedActionEquivalence.action_bijective`
96	def	`PaperXI.TwistedActionEquivalence.actionEquiv`
100	theorem	`PaperXI.TwistedActionEquivalence.actionEquiv_apply`
103	theorem	`PaperXI.TwistedActionEquivalence.actionEquiv_basis`

A.398 `PaperXI.TwistedAlgebra`

Source lines: 109.

SHA-256: `578da8f15a3a094d50dbc2d501fc66ea1c8ad354c2a29fb8acb5fd3f4c7b2020`.

```
A concrete realization of the complex twisted group algebra, including its
actual basis, multiplication, units, and dimension. The ambient endomorphism
algebra supplies the ring laws; all closure and basis assertions are proved.
```

Direct project imports: `PaperXI.TwistedRegular`.

Line	Kind	Name
16	def	`PaperXI.TwistedAlgebra.operatorSpan`
19	theorem	`PaperXI.TwistedAlgebra.regular_mem`
22	theorem	`PaperXI.TwistedAlgebra.one_mem`
27	theorem	`PaperXI.TwistedAlgebra.mul_mem`
45	def	`PaperXI.TwistedAlgebra.subalgebra`
57	abbrev	`PaperXI.TwistedAlgebra.Algebra`
62	def	`PaperXI.TwistedAlgebra.basis`
65	theorem	`PaperXI.TwistedAlgebra.coe_basis`
71	theorem	`PaperXI.TwistedAlgebra.basis_mul`
78	theorem	`PaperXI.TwistedAlgebra.basis_one`
84	def	`PaperXI.TwistedAlgebra.basisUnit`
96	theorem	`PaperXI.TwistedAlgebra.coe_basisUnit`
99	theorem	`PaperXI.TwistedAlgebra.basis_isUnit`
102	instance	`PaperXI.TwistedAlgebra.finiteDimensional`
105	theorem	`PaperXI.TwistedAlgebra.finrank_eq_card`

A.399 `PaperXI.TwistedMatrix`

Source lines: 54.

SHA-256: `99153ad911c9dab5fd807658f17f2d4d1583b71edd10538aad5c08bfb601e959`.

```
The matrix-algebra part of Paper XI, Proposition 8.3. Simplicity is proved
for the constructed twisted algebra before invoking Wedderburn–Artin. No
finite-image condition is imposed on the cocycle.
```

Direct project imports: `PaperXI.TwistedSimple`.

Line	Kind	Name
18	theorem	`PaperXI.TwistedMatrix.exists_matrix`
27	theorem	`PaperXI.TwistedMatrix.matrix_sqrt`
39	theorem	`PaperXI.TwistedMatrix.order_square`
46	theorem	`PaperXI.TwistedMatrix.classification`

A.400 `PaperXI.TwistedModules`

Source lines: 54.

SHA-256: `75c7bf4253db1b19641a24e2fbeaa7317eac3362bc5c6427038f89481a072edb`.

```
The simple-module conclusions of Paper XI, Proposition 8.3, for the actual
constructed twisted algebra. Existence, uniqueness, finite dimensionality,
and the square-root dimension are proved under the source hypotheses.
```

Direct project imports: `PaperXI.TwistedMatrix`, `PaperXI.SimpleModuleClassification`.

Line	Kind	Name
18	theorem	`PaperXI.TwistedModules.exists_simple_ideal`
26	theorem	`PaperXI.TwistedModules.equivalent`
40	theorem	`PaperXI.TwistedModules.finiteDimensional`
45	theorem	`PaperXI.TwistedModules.finrank_sq`
51	theorem	`PaperXI.TwistedModules.finrank_sqrt`

A.401 `PaperXI.TwistedRegular`

Source lines: 81.

SHA-256: `542e57f3ada86dab99247efc614f2e360c4ed62aa6278ac5d318f7cad41242b0`.

```
The faithful twisted left regular operators for an arbitrary normalized
complex unit-valued cocycle. No finite-image hypothesis is used.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
18	def	`PaperXI.TwistedRegular.regular`
23	theorem	`PaperXI.TwistedRegular.regular_apply`
26	theorem	`PaperXI.TwistedRegular.regular_one`
31	theorem	`PaperXI.TwistedRegular.regular_mul`
46	def	`PaperXI.TwistedRegular.deltaOne`
50	theorem	`PaperXI.TwistedRegular.regular_deltaOne`
66	def	`PaperXI.TwistedRegular.evaluateOne`
71	theorem	`PaperXI.TwistedRegular.regular_linearIndependent`

A.402 `PaperXI.TwistedRescaling`

Source lines: 158.

SHA-256: `1bb12f819e8232e254d6147b5fac209667d032836c4aa02fe72c472015d40e0d`.

```
Actual algebra equivalences under normalized scalar changes of basis.
The equivalence is constructed from the canonical bases of the two concrete
twisted algebras; its multiplicativity follows from their proved tables.
```

Direct project imports: `PaperXI.TwistedAction`, `PaperXI.CocycleBasis`.

Line	Kind	Name
15	def	`PaperXI.TwistedRescaling.congrEquiv`
20	theorem	`PaperXI.TwistedRescaling.congrEquiv_basis`
29	def	`PaperXI.TwistedRescaling.linearEquiv`
36	theorem	`PaperXI.TwistedRescaling.linearEquiv_basis`
46	theorem	`PaperXI.TwistedRescaling.scaled_basis_mul`
56	theorem	`PaperXI.TwistedRescaling.linearEquiv_one`
61	theorem	`PaperXI.TwistedRescaling.linearEquiv_mul`
88	def	`PaperXI.TwistedRescaling.rescaleEquiv`
95	theorem	`PaperXI.TwistedRescaling.rescaleEquiv_basis`
103	theorem	`PaperXI.TwistedRescaling.rescaleEquiv_symm_basis`
116	def	`PaperXI.TwistedRescaling.transport`
122	theorem	`PaperXI.TwistedRescaling.transport_basis`
131	theorem	`PaperXI.TwistedRescaling.transport_unique`
145	def	`PaperXI.TwistedRescaling.actionEquiv`

A.403 `PaperXI.TwistedSimple`

Source lines: 205.

SHA-256: `f420d05bd41765009035ee02338ae4721a2914e0946fb52c572fcee8ca333c60`.

```
Nondegenerate scalar commutators force simplicity of the actual twisted
algebra. Coordinate elimination is applied to two-sided ideals, so no
semisimplicity or matrix-algebra hypothesis enters the argument.
```

Direct project imports: `PaperXI.CocycleBasis`, `PaperXI.TwistedAlgebra`, `PaperXI.SchrodingerFourier`.

Line	Kind	Name
19	theorem	`PaperXI.TwistedSimple.commutator_separates`
34	theorem	`PaperXI.TwistedSimple.basis_commute`
44	theorem	`PaperXI.TwistedSimple.ideal_smul_mem`
52	theorem	`PaperXI.TwistedSimple.simple_of_basis`
117	theorem	`PaperXI.TwistedSimple.isSimpleRing`
129	theorem	`PaperXI.TwistedSimple.radical_eq_bot_iff`
135	theorem	`PaperXI.TwistedSimple.isSimpleRing_of_nondegenerate`
144	theorem	`PaperXI.TwistedSimple.diagonal_mul_basis`
157	theorem	`PaperXI.TwistedSimple.center_scalar_of_basis`
189	theorem	`PaperXI.TwistedSimple.center_eq_scalars`

A.404 `PaperXI.TypeNormalizerAction`

Source lines: 108.

SHA-256: `623d5fefcb3b3c856476bbd194fd5a1101347efa5511dbdd0c6a68ffc6e7c2f2`.

```
The actual ambient conjugation action on N/H and its descent to J/N.
The quotient action is constructed from groups, not from choices of operators.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.TypeNormalizerAction.conjugation`
25	theorem	`PaperXI.TypeNormalizerAction.conjugation_coe`
28	theorem	`PaperXI.TypeNormalizerAction.conjugation_inner`
34	def	`PaperXI.TypeNormalizerAction.Stable`
37	theorem	`PaperXI.TypeNormalizerAction.stable_map`
48	def	`PaperXI.TypeNormalizerAction.action`
65	theorem	`PaperXI.TypeNormalizerAction.action_mk`
72	theorem	`PaperXI.TypeNormalizerAction.subgroup_le_kernel`
85	def	`PaperXI.TypeNormalizerAction.quotientAction`
90	theorem	`PaperXI.TypeNormalizerAction.quotientAction_mk`
96	theorem	`PaperXI.TypeNormalizerAction.quotientAction_unique`

A.405 `PaperXI.TypeNormalizerForm`

Source lines: 85.

SHA-256: `1f05cc7cd75bdb3f15a7b10c3d2153b865baf51da1f73e0a4fe87998c5d6969c`.

```
The source-facing field-valued form: the ambient quotient action is
transported to the declared vector-space coordinates, then preservation is
proved from the actual character commutator. Residue-field linearity is explicit.
```

Direct project imports: `PaperXI.TypeNormalizerPairing`.

Line	Kind	Name
15	def	`PaperXI.TypeNormalizerForm.coordinateAction`
26	theorem	`PaperXI.TypeNormalizerForm.coordinateAction_apply`
33	def	`PaperXI.TypeNormalizerForm.linearAut`
44	def	`PaperXI.TypeNormalizerForm.vectorAction`
66	theorem	`PaperXI.TypeNormalizerForm.source_form_preserved`

A.406 `PaperXI.TypeNormalizerPairing`

Source lines: 154.

SHA-256: `47cff0132a4d65fdb6193671065f95af0c45479300e676253cef05a50957f105`.

```
Character invariance yields preservation of the actual quotient pairing.
No form-preservation hypothesis is inserted into the normalizer theorem.
```

Direct project imports: `PaperXI.TypeNormalizerAction`, `PaperXI.TypeSideExtension`.

Line	Kind	Name
17	def	`PaperXI.TypeNormalizerPairing.subgroupAut`
26	def	`PaperXI.TypeNormalizerPairing.CharacterInvariant`

Line	Kind	Name
32	theorem	`PaperXI.TypeNormalizerPairing.inertiaInvariant`
43	theorem	`PaperXI.TypeNormalizerPairing.characterCommutator_preserved`
61	def	`PaperXI.TypeNormalizerPairing.pairing`
69	theorem	`PaperXI.TypeNormalizerPairing.pairing_on_representatives`
77	theorem	`PaperXI.TypeNormalizerPairing.quotientAction_preserves_pairing`
93	theorem	`PaperXI.TypeNormalizerPairing.pairing_alternating`
98	theorem	`PaperXI.TypeNormalizerPairing.pairing_mul_left`
106	theorem	`PaperXI.TypeNormalizerPairing.pairing_mul_right`
119	theorem	`PaperXI.TypeNormalizerPairing.bilinear_character_separates`
142	theorem	`PaperXI.TypeNormalizerPairing.fieldForm_preserved`

A.407 `PaperXI.TypeProjectiveImplementation`

Source lines: 157.

SHA-256: `3621d11e4ad48c3a7d4986dc55e21778d7bf926a4a008b6de91a38ccc21db617`.

```
Implementation by an actual extension representation. Scalars compare
intertwiners for the same automorphism, not arbitrary quotient representatives.
The section defect is retained as an operator until scalarity is proved.
```

Direct project imports: `PaperXI.TypeNormalizerPairing`, `PaperXI.SchurScalar`, `PaperXI.CliffordDatum`.

Line	Kind	Name
17	def	`PaperXI.TypeProjectiveImplementation.operator`
31	theorem	`PaperXI.TypeProjectiveImplementation.operator_apply`
36	theorem	`PaperXI.TypeProjectiveImplementation.implements_conjugation`
50	theorem	`PaperXI.TypeProjectiveImplementation.implementer_unique`
60	theorem	`PaperXI.TypeProjectiveImplementation.two_implementers`
73	theorem	`PaperXI.TypeProjectiveImplementation.section_operators_mul`
81	theorem	`PaperXI.TypeProjectiveImplementation.representative_change`
86	def	`PaperXI.TypeProjectiveImplementation.ProjectiveDescends`
91	theorem	`PaperXI.TypeProjectiveImplementation.projectiveDescends_iff`
113	theorem	`PaperXI.TypeProjectiveImplementation.scalar_element_commutator`
127	theorem	`PaperXI.TypeProjectiveImplementation.source_operator_commutator`
138	theorem	`PaperXI.TypeProjectiveImplementation.nontrivial_pairing_obstructs_descent`

A.408 `PaperXI.TypeSideDescent`

Source lines: 293.

SHA-256: `e066d77666e2671d02c454f4bd06e3650750cfcb143a6c4635339d8b6a0adb2f`.

```
The actual type-side scalar restriction and quotient action. The source
input that an irreducible representation contains the invariant character
is used through its nonzero character eigenspace; scalar restriction on
the whole representation is a conclusion, not an additional assumption.
```

Direct project imports: `PaperXI.QuotientRepresentation`, `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
18	def	PaperXI.TypeSideDescent.conjugate
22	def	PaperXI.TypeSideDescent.Invariant
26	def	PaperXI.TypeSideDescent.characterSpace
39	def	PaperXI.TypeSideDescent.Contains
43	def	PaperXI.TypeSideDescent.characterRepresentation
53	theorem	PaperXI.TypeSideDescent.contains_iff_nonzero_intertwiner
86	theorem	PaperXI.TypeSideDescent.characterSpace_stable
99	def	PaperXI.TypeSideDescent.characterSubrepresentation
105	theorem	PaperXI.TypeSideDescent.scalar_restriction
121	def	PaperXI.TypeSideDescent.characterKernel
124	theorem	PaperXI.TypeSideDescent.characterKernel_le
129	theorem	PaperXI.TypeSideDescent.mem_characterKernel
139	instance	PaperXI.TypeSideDescent.characterKernel_normal
147	theorem	PaperXI.TypeSideDescent.characterKernel_le_ker
157	def	PaperXI.TypeSideDescent.representation
163	theorem	PaperXI.TypeSideDescent.representation_apply
169	theorem	PaperXI.TypeSideDescent.representation_irreducible
177	theorem	PaperXI.TypeSideDescent.finite_quotient
188	def	PaperXI.TypeSideDescent.quotientProjection
193	theorem	PaperXI.TypeSideDescent.quotientProjection_apply
198	theorem	PaperXI.TypeSideDescent.quotientProjection_surjective
203	theorem	PaperXI.TypeSideDescent.quotientProjection_ker
210	theorem	PaperXI.TypeSideDescent.central_image
229	def	PaperXI.TypeSideDescent.centralCharacter
232	theorem	PaperXI.TypeSideDescent.centralCharacter_apply
235	theorem	PaperXI.TypeSideDescent.centralCharacter_injective
240	def	PaperXI.TypeSideDescent.centerInclusion
248	theorem	PaperXI.TypeSideDescent.centerInclusion_apply
253	theorem	PaperXI.TypeSideDescent.centerInclusion_injective
260	theorem	PaperXI.TypeSideDescent.centerInclusion_range
272	theorem	PaperXI.TypeSideDescent.exact_center_quotient
277	theorem	PaperXI.TypeSideDescent.centerInclusion_central
285	theorem	PaperXI.TypeSideDescent.central_action

A.409 PaperXI.TypeSideExtension

Source lines: 170.

SHA-256: f00297a9c260398048b04cca886e47dbd0ac912e029104cee4892063e5aab11c.

```
The actual type-side central extension, its source commutator formula,
and scalar nondegeneracy transferred from a supplied source pairing. These
interfaces retain the simple-character facts as explicit inputs.
```

Direct project imports: PaperXI.TypeSideDescent, PaperXI.CentralExtensionCocycle.

Line	Kind	Name
20	instance	PaperXI.TypeSideExtension.centralQuotient_isMulCommutative
24	theorem	PaperXI.TypeSideExtension.finite_center
27	theorem	PaperXI.TypeSideExtension.centralCharacter_ne_one

Line	Kind	Name
36	def	`PaperXI.TypeSideExtension.datum`
48	theorem	`PaperXI.TypeSideExtension.commutator_mem`
54	def	`PaperXI.TypeSideExtension.characterCommutator`
60	theorem	`PaperXI.TypeSideExtension.scalar_commutator_source`
81	theorem	`PaperXI.TypeSideExtension.scalar_commutator_eq_pairing`
94	theorem	`PaperXI.TypeSideExtension.nondegenerate_of_pairing`
117	theorem	`PaperXI.TypeSideExtension.bilinear_character_nondegenerate`
138	theorem	`PaperXI.TypeSideExtension.nondegenerate_of_bilinear_pairing`
162	theorem	`PaperXI.TypeSideExtension.representation_central_action`

A.410 `PaperXI.TypeSideUniqueness`

Source lines: 96.

SHA-256: `3a4c2a7589f7cccf49eec7161e463a3c6d21f217693a596505efa5d339330a63`.

```
Uniqueness and dimension for the original actual type-side actions.
Character containment is converted to scalar restriction and quotient
descent by proved constructions. The source simple-character structure and
nondegenerate pairing remain explicit inputs. The first intertwiner and
the representation dimensions are conclusions.
```

Direct project imports: `PaperXI.TypeSideExtension`, `PaperXI.CentralExtensionUniqueness`.

Line	Kind	Name
21	theorem	`PaperXI.TypeSideUniqueness.finiteDimensional`
34	theorem	`PaperXI.TypeSideUniqueness.finrank_sqrt`
50	theorem	`PaperXI.TypeSideUniqueness.exists_equiv`
71	theorem	`PaperXI.TypeSideUniqueness.exists_unique_intertwiner`

A.411 `PaperXI.UnramifiedBaseChangeFactor`

Source lines: 159.

SHA-256: `0f6e188bb24bf02711cdd8d676182d365a8d0fac73456675899fa0f4564a7340`.

```
# Integer-degree base change for the finite unramified Weil model

The finite unramified model is canonically equivalent to `Multiplicative ℤ`.
Multiplication of the degree by `n` therefore gives the restriction map that
models unramified base change of residue degree `n`. This module proves the
resulting Frobenius-power and Euler-factor formulas.

No actual tower of local fields is constructed here; the theorem is scoped to
this integer-degree Weil model.
```

Direct project imports: `PaperXI.RankOneEulerFactor`.

Line	Kind	Name
30	def	`PaperXI.FiniteUnramifiedWeil.integerDegreeDilation`
38	def	`PaperXI.FiniteUnramifiedWeil.degreeDilation`
43	theorem	`PaperXI.FiniteUnramifiedWeil.degreeEquiv_degreeDilation`
49	theorem	`PaperXI.FiniteUnramifiedWeil.degree_degreeDilation`

Line	Kind	Name
61	theorem	`PaperXI.FiniteUnramifiedWeil.degreeDilation_geometricLift`
71	theorem	`PaperXI.FiniteUnramifiedWeil.degreeDilation_comp`
84	def	`PaperXI.FiniteUnramifiedWeil.baseChangeRepresentation`
91	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_geometricFrobenius`
100	theorem	`PaperXI.FiniteUnramifiedWeil.baseChangeRepresentation_comp`
113	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_frobeniusPolynomial`
121	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_localFactor`
131	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_baseChange_frobeniusPolynomial`
151	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_baseChange_localFactor`

A.412 `PaperXI.UnramifiedConductor`

Source lines: 131.

SHA-256: `1a940bc977e7fb0565338370fce87b395fc254c3cbbc39c220d61eef0a6de682`.

```
Restriction across an inertia-containing subgroup, with a power change of
residue Frobenius. The congruence groups, fixed spaces and EVERY lower conductor
summand are preserved, for wild as well as tame representations. This is not the
ramified-quotient/Herbrand change-of-extension theorem.
```

Direct project imports: `PaperXI.ResidueDegreeBaseChange`, `PaperXI.ValuationConductor`.

Line	Kind	Name
18	theorem	`PaperXI.UnramifiedConductor.lower_eq_comap`
21	def	`PaperXI.UnramifiedConductor.lowerEquiv`
28	theorem	`PaperXI.UnramifiedConductor.lower_card`
33	def	`PaperXI.UnramifiedConductor.embedding`
47	theorem	`PaperXI.UnramifiedConductor.embedding_degree`
52	theorem	`PaperXI.UnramifiedConductor.embedding_injective`
65	theorem	`PaperXI.UnramifiedConductor.embedding_lift`
77	theorem	`PaperXI.UnramifiedConductor.fixed_eq`
95	theorem	`PaperXI.UnramifiedConductor.summand_eq`
103	theorem	`PaperXI.UnramifiedConductor.swan_eq`
112	theorem	`PaperXI.UnramifiedConductor.artin_eq`
120	theorem	`PaperXI.UnramifiedConductor.summable_restriction`

A.413 `PaperXI.UnramifiedConductorZero`

Source lines: 109.

SHA-256: `35a3f791e30467fdb145456cea233a2a85f3fc669b8cc9a5216db7b5a705f19d`.

```
# Conductor zero for the finite unramified Weil model

This module does two separate things.

* It proves a generic theorem in the existing `ValuationConductor` interface:
  a representation has Artin conductor zero when the actual degree-kernel inertia
  subgroup is trivial.
* It packages the corresponding codimension/Swan/Artin exponents for the finite
  unramified Weil model constructed in Block 39.

The model-specific exponents are not asserted to be the conductor of an absolute
Weil representation. They record the exact unramified specialization of the
existing fixed-space formula.
```

Direct project imports: `PaperXI.UnramifiedLocalFactor`, `PaperXI.ValuationConductor`.

Line	Kind	Name
34	theorem	`PaperXI.ValuationConductor.artin_zero_of_degree_kernel_bot`
73	def	`PaperXI.FiniteUnramifiedWeil.inertiaCodimension`
80	def	`PaperXI.FiniteUnramifiedWeil.swanConductorExponent`
84	def	`PaperXI.FiniteUnramifiedWeil.artinConductorExponent`
89	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaCodimension_eq_zero`
97	theorem	`PaperXI.FiniteUnramifiedWeil.swanConductorExponent_eq_zero`
103	theorem	`PaperXI.FiniteUnramifiedWeil.artinConductorExponent_eq_zero`

A.414 `PaperXI.UnramifiedFieldTower`

Source lines: 270.

SHA-256: `79699a103ad55a1254dbaa1ea632d59bd77ce2f4ac03367b50d40d1b35f287ee`.

```
# Recursive unramified local-field towers

This module constructs the upper field over the already constructed lower field,
using the checked residue-field equivalence to transport a finite residue tower.
The resulting map on the integer-degree unramified Weil models is defined by the
actual relative residue degree.  No absolute Weil group is asserted.
```

Direct project imports: `PaperXI.UnramifiedBaseChangeFactor`.

Line	Kind	Name
32	abbrev	`PaperXI.UnramifiedFieldTower.LowerField`
36	def	`PaperXI.UnramifiedFieldTower.residueTowerMap`
53	abbrev	`PaperXI.UnramifiedFieldTower.UpperField`
69	def	`PaperXI.UnramifiedFieldTower.lowerToUpper`
75	theorem	`PaperXI.UnramifiedFieldTower.residue_finrank`
83	theorem	`PaperXI.UnramifiedFieldTower.relative_degree`
90	theorem	`PaperXI.UnramifiedFieldTower.total_degree`
98	def	`PaperXI.UnramifiedFieldTower.upperResidueAlgEquiv`
104	theorem	`PaperXI.UnramifiedFieldTower.relative_ramificationIdx_one`
111	def	`PaperXI.UnramifiedFieldTower.restriction`
120	theorem	`PaperXI.UnramifiedFieldTower.restriction_degree_square`
127	theorem	`PaperXI.UnramifiedFieldTower.restriction_degree`
136	theorem	`PaperXI.UnramifiedFieldTower.restriction_geometricLift`
159	theorem	`PaperXI.UnramifiedFieldTower.restriction_injective`
187	theorem	`PaperXI.UnramifiedFieldTower.restriction_range`
217	def	`PaperXI.UnramifiedFieldTower.restrictionEquivPeriod`
228	theorem	`PaperXI.UnramifiedFieldTower.restriction_unique`
246	def	`PaperXI.UnramifiedFieldTower.restrictRepresentation`
253	theorem	`PaperXI.UnramifiedFieldTower.restrictRepresentation_geometricLift`
261	theorem	`PaperXI.UnramifiedFieldTower.restriction_frobeniusPolynomial`

A.415 `PaperXI.UnramifiedLocalFactor`

Source lines: 168.

SHA-256: `ca0d3e6fba2b3caa2549ac65cd50e4fe66b15dc4cf96c9398531f826b265ebbd`.

```
# Unramified local factors for the finite Weil model
```

Direct project imports: `PaperXI.FiniteUnramifiedWeil`.

Line	Kind	Name
28	abbrev	`PaperXI.FiniteUnramifiedWeil.Inertia`
33	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_le_kernel`
39	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_invariants_eq_top`
47	def	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial`
52	def	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor`
59	def	`PaperXI.FiniteUnramifiedWeil.inertiaFixedEquiv`
64	def	`PaperXI.FiniteUnramifiedWeil.fixedGeometricFrobenius`
70	theorem	`PaperXI.FiniteUnramifiedWeil.fixedGeometricFrobenius_charpoly`
77	def	`PaperXI.FiniteUnramifiedWeil.inertiaFrobeniusPolynomial`
83	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaFrobeniusPolynomial_eq`
90	def	`PaperXI.FiniteUnramifiedWeil.inertiaLocalFactor`
95	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaLocalFactor_eq`
103	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eq_det`
114	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_coeff_zero`
121	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eval_zero`
127	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_ne_zero`
136	theorem	`PaperXI.FiniteUnramifiedWeil.mappedFrobeniusPolynomial_ne_zero`
143	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_mul_polynomial`
151	theorem	`PaperXI.FiniteUnramifiedWeil.polynomial_mul_unramifiedLocalFactor`
159	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_ne_zero`

A.416 `PaperXI.UnramifiedPolynomial`

Source lines: 90.

SHA-256: `2dfea600d7050f62c8adde4b0242ca02748ec7427f610c1876bc40b1d7a2f64b`.

```
An irreducible residue polynomial constructs a finite etale DVR and its
actual fraction field, without any supplied ambient Galois extension.
```

Direct project imports: `PaperXI.MonicResidueAlgebra`.

Line	Kind	Name
21	theorem	`PaperXI.UnramifiedPolynomial.irreducible`
24	instance	`PaperXI.UnramifiedPolynomial.domain`
27	instance	`PaperXI.UnramifiedPolynomial.faithful`
30	theorem	`PaperXI.UnramifiedPolynomial.not_field`
37	instance	`PaperXI.UnramifiedPolynomial.dvr`
44	theorem	`PaperXI.UnramifiedPolynomial.uniformizer`
50	abbrev	`PaperXI.UnramifiedPolynomial.field`
52	instance	`PaperXI.UnramifiedPolynomial.baseFieldAlgebra`
55	instance	`PaperXI.UnramifiedPolynomial.baseFieldTower`
58	theorem	`PaperXI.UnramifiedPolynomial.natDegree_pos_of_residue_irreducible`
62	theorem	`PaperXI.UnramifiedPolynomial.field_degree`
66	instance	`PaperXI.UnramifiedPolynomial.finiteFieldExtension`
71	theorem	`PaperXI.UnramifiedPolynomial.integralClosure`

Line	Kind	Name
78	theorem	PaperXI.UnramifiedPolynomial.etale
85	theorem	PaperXI.UnramifiedPolynomial.ramificationIdx_one

A.417 PaperXI.UnramifiedValuation

Source lines: 145.

SHA-256: 001406e3abfac1991c15f4006373c9a8673cd0e8e16df3d80a9ef6c276a50054.

```
Source-facing unramified residue-base attachment to an actual valuation
subring and decomposition-group action. The subgroup fixes the actual embedded
finite residue extension, rather than being assigned an index by a certificate.
This still does not construct its lift to a local-field unramified extension.
```

Direct project imports: PaperXI.FiniteResidueSubgroup, PaperXI.UnramifiedConductor, PaperXI.ValuationTameEndpoint.

Line	Kind	Name
22	abbrev	PaperXI.UnramifiedValuation.fixingGroup
27	theorem	PaperXI.UnramifiedValuation.frobenius_agrees
32	theorem	PaperXI.UnramifiedValuation.contains_inertia
37	def	PaperXI.UnramifiedValuation.embedding
45	theorem	PaperXI.UnramifiedValuation.degree_formula
54	def	PaperXI.UnramifiedValuation.degreeSubgroupEquiv
61	theorem	PaperXI.UnramifiedValuation.embedding_injective
66	theorem	PaperXI.UnramifiedValuation.embedding_range
93	theorem	PaperXI.UnramifiedValuation.degree_index
103	theorem	PaperXI.UnramifiedValuation.norm_compatibility
118	theorem	PaperXI.UnramifiedValuation.source

A.418 PaperXI.ValuationConductor

Source lines: 167.

SHA-256: 3040c48c4558f4796813146cac1147f5d7bf87a20ff05cabecf53a37fce73cb0.

```
Conductor sums from actual maximal-ideal congruence groups. For a finite
Galois extension with discrete valuation these are the lower-numbering expressions.
No change-of-extension theorem, absolute Weil comparison, or Hasse--Arf integrality
is inferred from this definition. Finite support is proved under the stated
Noetherian/finite faithful action conditions.
```

Direct project imports: PaperXI.ResidueWeil, PaperXI.RamificationSeparation.

Line	Kind	Name
19	abbrev	PaperXI.ValuationConductor.residue
23	def	PaperXI.ValuationConductor.lift
32	theorem	PaperXI.ValuationConductor.lift_degree
38	def	PaperXI.ValuationConductor.fixed
43	theorem	PaperXI.ValuationConductor.fixed_zero
61	def	PaperXI.ValuationConductor.codimension

Line	Kind	Name
66	def	`PaperXI.ValuationConductor.summand`
70	def	`PaperXI.ValuationConductor.swan`
74	def	`PaperXI.ValuationConductor.artin`
79	theorem	`PaperXI.ValuationConductor.positive_fixed_top`
88	theorem	`PaperXI.ValuationConductor.tame_summand_zero`
96	theorem	`PaperXI.ValuationConductor.tame_swan_zero`
102	theorem	`PaperXI.ValuationConductor.tame_irreducible_dimension`
116	theorem	`PaperXI.ValuationConductor.tame_artin_dichotomy`
127	theorem	`PaperXI.ValuationConductor.fixed_top_of_lower_bot`
138	theorem	`PaperXI.ValuationConductor.summable_summands`
151	theorem	`PaperXI.ValuationConductor.zeroth_summand`
160	theorem	`PaperXI.ValuationConductor.artin_eq_full_sum`

A.419 `PaperXI.ValuationFixingField`

Source lines: 41.

SHA-256: `fdaeae5757fb111b416bf264d6b76e43cead77d0d338c4bc16110191bedaeb06`.

Restricting the base field in the literal decomposition subgroup.

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
12	def	`PaperXI.ValuationFixingField.decompositionEquiv`
27	theorem	`PaperXI.ValuationFixingField.decompositionEquiv_apply`
31	theorem	`PaperXI.ValuationFixingField.ring_action`
35	theorem	`PaperXI.ValuationFixingField.residue_action`

A.420 `PaperXI.ValuationIntegralModel`

Source lines: 72.

SHA-256: `c126174463d0f9039e8425c76709f0ab71b7babdb8e5ed6b19aa18a6169ffd3c`.

Integral closure and finiteness of an invariant valuation ring.

Direct project imports: `PaperXI.HenselianGaloisResidue`.

Line	Kind	Name
13	abbrev	`PaperXI.ValuationIntegralModel.base`
15	instance	`PaperXI.ValuationIntegralModel.baseAlgebra`
17	instance	`PaperXI.ValuationIntegralModel.baseTower`
19	instance	`PaperXI.ValuationIntegralModel.baseFieldTower`
21	instance	`PaperXI.ValuationIntegralModel.baseLocal`
25	def	`PaperXI.ValuationIntegralModel.ringAction`
31	theorem	`PaperXI.ValuationIntegralModel.invariant`
50	theorem	`PaperXI.ValuationIntegralModel.integral`
56	theorem	`PaperXI.ValuationIntegralModel.integralClosure`

Line	Kind	Name
62	theorem	`PaperXI.ValuationIntegralModel.finite`
67	theorem	`PaperXI.ValuationIntegralModel.rank`

A.421 `PaperXI.ValuationRamification`

Source lines: 96.

SHA-256: `92ddb509dc327389c2168f44ddfd5f7e16f4315af16835a54398077fe81146ca`.

```
Lower congruence subgroups are defined by their action on an actual local ring.
For a discrete valuation ring in a finite Galois extension these are the usual
nonnegative lower ramification groups. The generic local-ring definition is NOT
asserted to be an absolute Weil ramification filtration.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.ValuationRamification.lower`
30	theorem	`PaperXI.ValuationRamification.mem_lower`
34	theorem	`PaperXI.ValuationRamification.smul_mem_power`
48	instance	`PaperXI.ValuationRamification.lower_normal`
55	theorem	`PaperXI.ValuationRamification.antitone`
60	theorem	`PaperXI.ValuationRamification.lower_zero_eq_residue_kernel`
80	theorem	`PaperXI.ValuationRamification.lower_le_inertia`
85	theorem	`PaperXI.ValuationRamification.positive_le_one`
91	theorem	`PaperXI.ValuationRamification.valuation_lower_zero`

A.422 `PaperXI.ValuationResidueEmbedding`

Source lines: 48.

SHA-256: `ee803ae38b207f0084f4fe3d41d3b6348f38ade5e33a9ca430804b2e35a4c98f`.

```
Residue embedding of an actual restricted valuation ring.
```

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
10	def	`PaperXI.ValuationResidueEmbedding.inclusion`
16	theorem	`PaperXI.ValuationResidueEmbedding.inclusion_val`
18	instance	`PaperXI.ValuationResidueEmbedding.inclusion_local`
33	def	`PaperXI.ValuationResidueEmbedding.residueMap`
35	theorem	`PaperXI.ValuationResidueEmbedding.residueMap_residue`
37	theorem	`PaperXI.ValuationResidueEmbedding.residueMap_injective`
39	theorem	`PaperXI.ValuationResidueEmbedding.mem_range_iff`

A.423 `PaperXI.ValuationTameEndpoint`

Source lines: 52.

SHA-256: `613a021b32c397cfc2425e8ee5c9cb796856bc8643069f24fde8b8f634df23d9`.

```
Actual valuation-ring specialization with a constructed geometric residue
Frobenius. The finite residue-base embedding remains declared. There is no
claimed identification with an absolute Weil group or extension-independent
arithmetic conductor.
```

Direct project imports: `PaperXI.ValuationConductor`.

Line	Kind	Name
20	def	`PaperXI.ValuationTameEndpoint.geometricFrobenius`
24	theorem	`PaperXI.ValuationTameEndpoint.geometric_power`
28	abbrev	`PaperXI.ValuationTameEndpoint.WeilPullback`
30	def	`PaperXI.ValuationTameEndpoint.inertiaIdentification`
38	theorem	`PaperXI.ValuationTameEndpoint.conductor`

FORMAL COMPANION XII

Patching Support and Hecke Packets

The formal companion to Chapter XII explains the declared hypotheses, theorem correspondences and evidence boundaries.

IN THIS COMPANION

Formal companion 2618

Return to Chapter XII

The formal record and the reader

This formal companion follows the mathematical text and visual studies. Its source census contains 96 modules and 1,229 public named declarations. Those are source counts, not a count of new mathematical discoveries.

Three contracts remain distinct

Record	What it establishes
Revised internal claims	The formal sources map all 49 amended claims to checked production targets.
Original wording	The historical census remains 41 complete, seven partial, and one unchecked.
Reader preparation	Checks of source, evidence, finite examples, references, layout, and packaging. No fresh Lean/Lake execution.

The full 49-row table in Appendix B now links to the actual main statements. Each mapped target still has its own displayed binders and source location. A link does not establish definitional equivalence between prose and code.

Read the last two constructions carefully

The companion-completion theorem assumes finite occupied classes, complete separated coefficients, continuous representatives, integral polynomial data, and relations valid in every admissible frame. Visual lesson Plate XII.23 makes the frame distinction concrete.

The five-gate theorem requires a compatible Hecke point, actual ordinary or derived carriers, and realization at the selected eigensystem. Visual lesson Plate XII.17 displays the separate premises. Propagation compares two supplied compatible families for the same automorphic object.

Compiler evidence

Compiler records certify the exact sources they checked. Source inventories, figure calculations and reader preparation have separate scopes. None substitutes for executing the kernel on a new formal statement.

Abstract

This guide explains the complete production Lean source accompanying Chapter XII, with a source catalog, declaration census, import audit, exact source excerpts, and a claim crosswalk. It separates constructed algebraic and topological objects from hypotheses supplied by an arithmetic application. The checked revised contract comprises 49 internal claims mapped to 291 distinct production targets. The original wording remains a separate ledger: 41 complete, seven partial, and one unchecked. The guide preserves all 96 production modules and their inherited verification records. Its new checks are source, provenance, and document checks. No fresh Lean compilation was available in the finalization environment.

1 What this guide certifies and how to use it

The complete formal sources accompany this volume. This companion describes their statements and hypotheses alongside the mathematical text's explicit clarifications. The production theorems and their compiler records retain their separate scope.

Read Sections 4–7 for the completed local and global constructions, Sections 8–12 for support, packets, occupancy and conditional realization, and Section 14 for replay. Appendices list every source module and every revised internal claim. The machine-readable catalog gives full SHA-256 hashes, source locations, imports, declaration names and line numbers. The full production source is supplied. Printing every line in this PDF is unnecessary for reproducibility.

Three distinct kinds of evidence must be kept separate. A Lean kernel check verifies the exact formal proposition under its hypotheses and dependencies. An ordinary mathematical proof explains a proposition in mathematical prose, with its own statement and citation obligations. A finite computation verifies the tested cases, numerical identities or model bounds stated by that computation. Neither a finite numerical certificate nor a scan of source tokens is a substitute for kernel elaboration of the production proof.

The compiler evidence is retained with the corresponding source and object identities. Reader preparation does not execute the kernel. The replay procedure is available with the separately supplied runtime.

2 Independently checked counts and their definitions

The registered manifest contains exactly 96 distinct modules, and its source path set equals the set of all production `.lean` files. There are 91 files under `PaperXII/` and five under `PaperX/`. Directory names alone do not identify originality: `PaperXII.ReuseCompactness` and `PaperXII.ReuseFiniteSeparation` retain Paper X namespaces and 18 reused theorem declarations. Together with 63 theorem declarations in the five Paper X files, these account for the 81 reused theorems. Chapter XII's own public theorem count is 748.

The public named lexical census is 1,229: 829 theorems, 319 definitions, 47 abbreviations, 17 structures and 17 named instances. One private theorem helper, `PaperXII.DeterminantalIdeals.option_factor`, is excluded from the public count. The 829 theorem declarations divide into 748 own and 81 reused. They are supporting lemmas and interfaces as well as theorems implementing numbered manuscript claims. This is not a historical novelty count or a twelve-chapter total.

The independent scanner blanks nested block comments, line comments and strings while preserving line numbers. It recognizes anchored named declarations in each module's unique namespace and resolves their qualified names. Its declaration set agrees exactly with the inherited axiom target set. It finds no production `sorry`, `admit`, `axiom`, `unsafe`, or `native_decide` token outside comments/strings. These are lexical findings. Elaboration and transitive axioms remain the role of the compiler records.

The 49 revised claim records comprise 27 theorems, 14 corollaries, seven propositions and one lemma. Three worked examples, definitions, remarks and eight external standard facts are outside that internal census. All 291 distinct mapped targets occur in the 1,229-name inherited audit. A target can support several claims. A claim can require several targets. Target count is therefore neither theorem count nor claim count.

The original wording has 41 complete, seven partial and one unchecked claim. The eight amendments are 6.12, 9.4, 9.5, 9.6, 10.5, 12.2, 12.4 and 12.6. The amended statements have complete mapped coverage for all 49 claims. That does not relabel the original statements as proved.

3 Source organization, dependency order and hypothesis notation

The source audit finds 113 direct internal import edges. Every internal import names a registered module and precedes its consumer in `checks/modules.json`. The external direct imports are Mathlib modules and `Std`. The complete list is in `data/census.json`. This checks dependency order in the sources, not the entire transitive Mathlib import graph. The production tree reuses Chapter X, not Chapter XI.

The 96 modules consist of 71 foundational modules, ten Fitting modules and fifteen completion modules. The completion group has 146 public theorem declarations and 225 public named declarations. These are source counts. Presenting them here supplies no new proof.

In Lean, arguments in square brackets such as `[CommRingR]` and `[CompleteSpaceP]` are typeclass hypotheses. A proposition-valued ordinary argument such as `(hB:IsAdicmB)` is also a hypothesis. A `variable` command may contribute ambient arguments to later declarations. `include` and `omit` control that contribution. Reading only the line beginning `theorem` can therefore hide essential assumptions. The full exact excerpt modules preserve these commands, imports, namespace and section boundaries.

The source excerpts delivered as `excerpts/*.lean` are byte-identical full modules, including their ambient hypotheses. PDF excerpts select line ranges and retain original line numbers. To avoid missing glyphs in portable code typography, the display copies replace eleven symbols by readable forms, for example `iSup` for the indexed supremum and `/` for Lean's ideal-quotient slash. The complete replacement map is in `excerpts/INDEX.json`. This is a documented display convention only. Production source and exact excerpts retain every original Unicode character.

4 Claim 3.1: construct the tensor, then complete it

For a finite index type ι, commutative coefficient ring O, and commutative O-algebras R_i, the raw tensor is Mathlib's actual `PiTensorProduct OR`. It is not a record accepting an already supplied tensor equivalence. The factor inclusion is `PiTensorProduct.singleAlgHom`. The family lift is constructed from a multilinear product and the tensor's universal map.

Given ideals $m_O \subseteq O$, $m_i \subseteq R_i$, and condition ideals I_i, define

$$S = \bigotimes_{i\in\iota,O} R_i, \qquad \mathfrak{a} = m_O S + \sum_i m_i S, \qquad L = \sum_i I_i S.$$

In Lean the finite sums are indexed suprema of mapped ideals. The scalar term is present even if ι is empty. Removing it would change the topology for the empty family to the wrong normalization.

Source: `PaperXII/FiniteTensorQuotient.lean`, lines 11–26.

```
variable {O ι : Type*} [CommRing O] [Fintype ι] [DecidableEq ι]
variable (R : ι → Type*) [∀ i, CommRing (R i)] [∀ i, Algebra O (R i)]

abbrev Raw := PiTensorProduct O R

/-- The canonical inclusion of one factor in the actual finite tensor product. -/
def includeFactor (i : ι) : R i →ₐ[O] Raw (O := O) R := PiTensorProduct.singleAlgHom i

/-- The universal algebra map, constructed by the multilinear product of the factor maps. -/
def liftFamily {A : Type*} [CommRing A] [Algebra O A] (f : ∀ i, R i →ₐ[O] A) :
    Raw (O := O) R →ₐ[O] A :=
  PiTensorProduct.liftAlgHom
    ((MultilinearMap.mkPiAlgebra O ι A).compLinearMap (fun i ↦ (f i).toLinearMap))
    (by simp [MultilinearMap.compLinearMap_apply, MultilinearMap.mkPiAlgebra_apply])
    (by intro x y; simp [MultilinearMap.compLinearMap_apply, MultilinearMap.mkPiAlgebra_apply,
      Pi.mul_apply, Finset.prod_mul_distrib])
```

Source: `PaperXII/FiniteTensorQuotient.lean`, lines 60–75.

```
def conditions (J : ∀ i, Ideal (R i)) : Ideal (Raw (O := O) R) :=
  iSup i, (J i).map (includeFactor R i).toRingHom

/-- The defining ideal also retains the base ideal, necessary for the empty family. -/
def definingIdeal (mO : Ideal O) (m : ∀ i, Ideal (R i)) : Ideal (Raw (O := O) R) :=
  mO.map (algebraMap O (Raw (O := O) R)) ⊔ conditions R m

variable (J : ∀ i, Ideal (R i))

/-- Apply the actual quotient map in every factor. -/
def quotientMap : Raw (O := O) R →ₐ[O] Raw (O := O) (fun i ↦ R i ⧸ J i) :=
  liftFamily R (fun i ↦ (includeFactor (fun i ↦ R i ⧸ J i) i).comp (Ideal.Quotient.mkₐ O (J i)))

@[simp] theorem quotientMap_include (i : ι) (x : R i) :
    quotientMap (O := O) R J (includeFactor R i x) =
      includeFactor (fun i ↦ R i ⧸ J i) i (Ideal.Quotient.mk (J i) x) := by
```

The ordinary quotient map is built by applying the ideal quotient in each factor. Its kernel contains L. Quotient lifts construct both directions between S/L and $\bigotimes_i (R_i/I_i)$. Tensor extensionality and quotient extensionality prove the inverse identities. This produces an ordinary algebra equivalence, with

a proved exact image for the defining ideal. Ordinary tensor right exactness alone would not establish the completed theorem.

`AdicModel.CarrierRI` is a type synonym that installs the specified ideal-adic topology on R. Its `Completed` object is the actual Hausdorff uniform completion of that carrier. The forgetful raw ring equivalence does not assert continuity into an arbitrarily chosen topology. This prevents the topology from being replaced silently.

Source: `PaperXII/AdicModel.lean`, lines 12–42.

```
def Carrier (R : Type*) [CommRing R] (_I : Ideal R) := R

instance carrierCommRing (R : Type*) [CommRing R] (I : Ideal R) : CommRing (Carrier R I) :=
  inferInstanceAs (CommRing R)

/-- Forget the type synonym, without asserting continuity into an arbitrary topology. -/
def rawEquiv (R : Type*) [CommRing R] (I : Ideal R) : Carrier R I ≃+* R := RingEquiv.refl R

instance carrierWithIdeal (R : Type*) [CommRing R] (I : Ideal R) : WithIdeal (Carrier R I) :=
  ⟨I⟩

theorem carrier_isAdic (R : Type*) [CommRing R] (I : Ideal R) :
    IsAdic (WithIdeal.i : Ideal (Carrier R I)) := rfl

instance carrier_countablyGenerated (R : Type*) [CommRing R] (I : Ideal R) :
    Filter.IsCountablyGenerated (uniformity (Carrier R I)) := by
  have := (carrier_isAdic R I).hasBasis_nhds_zero.isCountablyGenerated
  exact IsUniformAddGroup.uniformity_countably_generated

/-- This is an actual completion, not a supplied completion-interface object. -/
abbrev Completed (R : Type*) [CommRing R] (I : Ideal R) := Completion (Carrier R I)

def embed (R : Type*) [CommRing R] (I : Ideal R) : R →+* Completed R I :=
  Completion.coeRingHom.comp (rawEquiv R I).symm.toRingHom

@[simp] theorem embed_apply (R : Type*) [CommRing R] (I : Ideal R) (x : R) :
    embed R I x = ((rawEquiv R I).symm x : Completed R I) := rfl

/-- A ring map carrying the defining ideal into the target ideal is uniformly continuous. -/
theorem map_continuous {R S : Type*} [CommRing R] [CommRing S]
    (I : Ideal R) (J : Ideal S) (f : R →+* S) (h : I.map f ≤ J) :
```

Thus `CompletedTensorRmOm` is the Hausdorff completion of S in the literal $\mathfrak{a}$-adic additive uniformity. Its O-algebra structure comes from the dense embedding of S, and its completed factor maps compose that embedding with the raw factor inclusions. The condition ideal is the closure of the algebraically extended L.

Source: `PaperXII/CompletedFiniteTensor.lean`, lines 16–47.

```
namespace PaperXII.FiniteTensor

variable {O ι : Type*} [CommRing O] [Fintype ι] [DecidableEq ι]
variable (R : ι → Type*) [∀ i, CommRing (R i)] [∀ i, Algebra O (R i)]
variable (mO : Ideal O) (m : ∀ i, Ideal (R i))

/-- The actual finite completed tensor product. -/
abbrev CompletedTensor := AdicModel.Completed (Raw (O := O) R) (definingIdeal R mO m)

/-- The algebra structure is induced by the canonical dense embedding. -/
instance completedTensorAlgebra : Algebra O (CompletedTensor R mO m) :=
```

```
  ((AdicModel.embed (Raw (O := O) R) (definingIdeal R mO m)).comp
    (algebraMap O (Raw (O := O) R))).toAlgebra

/-- Actual dense tensor inclusion, as an algebra homomorphism. -/
def tensorEmbed : Raw (O := O) R →ₐ[O] CompletedTensor R mO m where
  __ := AdicModel.embed (Raw (O := O) R) (definingIdeal R mO m)
  commutes' _ := rfl

/-- Actual completed factor map. -/
def completedInclude (i : ι) : R i →ₐ[O] CompletedTensor R mO m :=
  (tensorEmbed R mO m).comp (includeFactor R i)

/-- The closed ideal generated by all the supplied local condition ideals. -/
def closedConditions (J : ∀ i, Ideal (R i)) : Ideal (CompletedTensor R mO m) :=
  CompletionQuotient.closedExtension
    (AdicModel.carrierIdeal (definingIdeal R mO m) (conditions R J))

omit [Fintype ι] in
/-- This is precisely the literal closed ideal of the completed factor images. -/
theorem closedConditions_eq (J : ∀ i, Ideal (R i)) :
    closedConditions R mO m J =
```

5 Completion and quotient: actual maps, completeness and inverses

The reusable comparison is

$$\widehat{R}\big/\overline{I\widehat{R}} \;\cong\; \widehat{R/I}.$$

Its ambient assumptions are a commutative ring, a uniform space with uniform additive group, and a topological ring. Countable generation of the additive uniformity is used to prove completeness of the quotient of the completion. There is no assumption supplying completeness of that quotient, an inverse comparison map, or comparison surjectivity. The closure makes the quotient Hausdorff.

Source: `PaperXII/CompletionQuotient.lean`, lines 18–61.

```
set_option autoImplicit false
namespace PaperXII.CompletionQuotient

variable {R : Type*} [CommRing R] [UniformSpace R] [IsUniformAddGroup R]
  [IsTopologicalRing R]

/-- Countable uniform bases survive actual Hausdorff completion. -/
theorem completion_countablyGenerated {α : Type*} [UniformSpace α]
    [Filter.IsCountablyGenerated (uniformity α)] :
    Filter.IsCountablyGenerated (uniformity (Completion α)) := by
  have : Filter.IsCountablyGenerated (uniformity (CauchyFilter α)) := by
    obtain ⟨s, hs⟩ := Filter.isCountablyGenerated_iff_exists_antitone_basis.mp
      (inferInstance : Filter.IsCountablyGenerated (uniformity α))
    exact (CauchyFilter.basis_uniformity hs.toHasBasis).isCountablyGenerated
  change Filter.IsCountablyGenerated (uniformity (SeparationQuotient (CauchyFilter α)))
  rw [SeparationQuotient.uniformity_eq]
  infer_instance

/-- The uniformity associated with the actual quotient topology. -/
@[instance_reducible] def quotientUniformSpace {S : Type*} [CommRing S] [TopologicalSpace S]
    [IsTopologicalRing S] (I : Ideal S) : UniformSpace (S ⧸ I) :=
  IsTopologicalAddGroup.rightUniformSpace (S ⧸ I)

attribute [local instance] quotientUniformSpace

instance quotientUniformAddGroup {S : Type*} [CommRing S] [TopologicalSpace S]
```

```
    [IsTopologicalRing S] (I : Ideal S) : IsUniformAddGroup (S / I) :=
  isUniformAddGroup_of_addCommGroup

/-- Close the extended ideal in the actual completion. -/
def closedExtension (I : Ideal R) : Ideal (Completion R) :=
  (I.map (Completion.coeRingHom : R →+* Completion R)).closure

instance closedExtension_isClosed (I : Ideal R) :
    IsClosed (closedExtension I : Set (Completion R)) := isClosed_closure

/-- The quotient of the completion is complete; no finite-generation assumption. -/
theorem quotient_complete (I : Ideal R) [Filter.IsCountablyGenerated (uniformity R)] :
    CompleteSpace (Completion R / closedExtension I) := by
  have := completion_countablyGenerated (α := R)
  exact QuotientAddGroup.completeSpace_right (Completion R) (closedExtension I).toAddSubgroup

/-- The completion of the quotient receives the canonical completed quotient map. -/
def forwardMap (I : Ideal R) : Completion R →+* Completion (R / I) :=
```

The forward map extends the actual quotient map. Its continuity and the target's separation prove that it kills the closed extended ideal. The backward map first descends the dense raw representatives, then extends from R/I to its completion. The source quotient has its actual quotient topology and the associated additive uniformity. `PaperXII.AdicQuotientTopology` separately proves that this quotient topology is the topology of the image defining ideal.

The inverse laws are proved on dense representatives and extended by continuity. The second law uses quotient surjectivity after completion induction. The ring equivalence and its homeomorphism are then constructed from these proved maps.

Source: `PaperXII/CompletionQuotient.lean`, lines 121–180.

```
attribute [local instance] quotient_complete

/-- The reverse comparison extends the quotient map to its actual completion. -/
def backward (I : Ideal R) :
    Completion (R / I) →+* (Completion R / closedExtension I) :=
  Completion.extensionHom (quotientBaseMap I) (quotientBaseMap_continuous I)

theorem backward_continuous (I : Ideal R) : Continuous (backward I) :=
  Completion.continuous_extension

@[simp] theorem backward_coe (I : Ideal R) (x : R / I) :
    backward I (x : Completion (R / I)) = quotientBaseMap I x :=
  Completion.extensionHom_coe _ _ x

/-- First inverse law, established on dense representatives then by continuity. -/
theorem forward_backward (I : Ideal R) (x : Completion (R / I)) :
    forward I (backward I x) = x := by
  refine Completion.induction_on x
    (isClosed_eq ((forward_continuous I).comp (backward_continuous I)) continuous_id) ?_
  intro x
  obtain ⟨r, rfl⟩ := Ideal.Quotient.mk_surjective x
  simp

/-- Second inverse law; the dense representative argument precedes quotient induction. -/
theorem backward_forwardMap (I : Ideal R) (x : Completion R) :
    backward I (forwardMap I x) = Ideal.Quotient.mk (closedExtension I) x := by
  refine Completion.induction_on x
    (isClosed_eq ((backward_continuous I).comp (forwardMap_continuous I))
      (QuotientRing.isOpenQuotientMap_mk _).continuous) ?_
  intro r
```

```
  simp

theorem backward_forward (I : Ideal R) (x : Completion R / closedExtension I) :
    backward I (forward I x) = x := by
  obtain ⟨x, rfl⟩ := Ideal.Quotient.mk_surjective x
  exact backward_forwardMap I x

/-- Completion commutes with quotient by the closed extended ideal. -/
def equiv (I : Ideal R) :
    (Completion R / closedExtension I) ≃+* Completion (R / I) :=
  RingEquiv.ofRingHom (forward I) (backward I)
    (RingHom.ext (forward_backward I)) (RingHom.ext (backward_forward I))

theorem equiv_continuous (I : Ideal R) : Continuous (equiv I) := forward_continuous I

theorem equiv_symm_continuous (I : Ideal R) : Continuous (equiv I).symm :=
  backward_continuous I

@[simp] theorem equiv_mk_coe (I : Ideal R) (r : R) :
    equiv I (Ideal.Quotient.mk (closedExtension I) (r : Completion R)) =
      (Ideal.Quotient.mk I r : Completion (R / I)) := forwardMap_coe I r

/-- A bundled homeomorphism accompanying the ring equivalence. -/
def homeomorph (I : Ideal R) :
    (Completion R / closedExtension I) ≃ₜ Completion (R / I) where
  toEquiv := (equiv I).toEquiv
  continuous_toFun := equiv_continuous I
  continuous_invFun := equiv_symm_continuous I

end Countable
```

Applying this to the raw finite tensor quotient gives the displayed O-algebra equivalence of Claim 3.1. Its forward direction in the manuscript is the inverse of the construction direction. Both maps are continuous, and every factor identity is proved with the actual inclusions.

Source: `PaperXII/CompletedFiniteTensor.lean`, lines 91–133.

```
    CompletedTensor (fun i ↦ R i / J i) mO
      (fun i ↦ (m i).map (Ideal.Quotient.mk (J i))) ≃ₐ[O]
      (CompletedTensor R mO m / closedConditions R mO m J) :=
  (show (CompletedTensor R mO m / closedConditions R mO m J) ≃ₐ[O]
      CompletedTensor (fun i ↦ R i / J i) mO
        (fun i ↦ (m i).map (Ideal.Quotient.mk (J i))) from
    { __ := completedQuotientEquiv R mO m J
      commutes' r := by
        change completedQuotientEquiv R mO m J
          (Ideal.Quotient.mk _ (tensorEmbed R mO m (algebraMap O (Raw (O := O) R) r))) = _
        rw [completedQuotientEquiv_mk_embed, (quotientMap R J).commutes]
        rfl }).symm

/-- Both maps of the completed tensor quotient equivalence are continuous. -/
theorem completed_tensor_quotient_continuous :
    Continuous (completed_tensor_quotient R mO m J) ∧
      Continuous (completed_tensor_quotient R mO m J).symm :=
  ⟨completedQuotientEquiv_symm_continuous R mO m J,
    completedQuotientEquiv_continuous R mO m J⟩

/-- The isomorphism respects every actual canonical factor map. -/
@[simp] theorem completed_tensor_quotient_factor (i : ι) (x : R i) :
    completed_tensor_quotient R mO m J
      (completedInclude (fun i ↦ R i / J i) mO
        (fun i ↦ (m i).map (Ideal.Quotient.mk (J i))) i (Ideal.Quotient.mk (J i) x)) =
      Ideal.Quotient.mk _ (completedInclude R mO m i x) := by
```

```
  apply (completedQuotientEquiv R mO m J).injective
  change completedQuotientEquiv R mO m J
    ((completedQuotientEquiv R mO m J).symm _) = _
  rw [RingEquiv.apply_symm_apply]
  symm
  simpa only [quotientMap_include, completedInclude, AlgHom.comp_apply] using
    completedQuotientEquiv_mk_embed R mO m J (includeFactor R i x)

attribute [local instance] CompletionQuotient.quotientUniformSpace

omit [Fintype ι] in
/-- The right-hand quotient is itself a genuine complete Hausdorff ring. -/
theorem completed_tensor_quotient_complete :
    CompleteSpace (CompletedTensor R mO m / closedConditions R mO m J) ∧
      T2Space (CompletedTensor R mO m / closedConditions R mO m J) :=
  ⟨CompletionQuotient.quotient_complete _, inferInstance⟩

```

The source theorem `closedConditions_eq` identifies the ideal with the literal closure of the ideal generated by the completed factor images. The supplementary consumer `exact_completed_kernel` additionally asserts exact equality of the kernel of the canonical completed map and the stated closed ideal. That consumer is separate from the production theorem census.

The general comparison requires neither locality nor Noetherianity of the raw tensor ring. `PaperXII.AdicCompletionComparison` independently identifies the uniform completion with Mathlib's algebraic inverse-limit adic completion. Finite generation controls ideal powers and identifies the completed topology with the topology of its extended defining ideal. These steps support the subsequent specialization to local rings.

6 Locality, universal maps and the edge cases

For the manuscript specialization, supply a field k, an O-algebra structure on k, residue algebra maps $R_i \to k$, and surjectivity of $O \to k$. The defining ideals are their kernels. `PaperXII.FiniteTensorLocality` constructs $S/\mathfrak{a} \cong k$, proves $\mathfrak{a}$ maximal, and derives finite generation from Noetherianity of O and each R_i. It then proves locality and the maximal-ideal-adic topology of the actual completion. These residue maps are explicit mathematical data, not an implicit assertion that all residue fields somehow agree.

Source: `PaperXII/FiniteTensorLocality.lean`, lines 100–138.

```
    (fgO : mO.FG) (fg : ∀ i, (m i).FG) : (definingIdeal (O := O) R mO m).FG := by
  exact Submodule.FG.sup (fgO.map _) (Submodule.fg_iSup _ (fun i ↦ (fg i).map _))

/-- The completed finite tensor product is local under the actual common-residue data. -/
theorem completedTensor_isLocalRing (hO : Surjective (algebraMap O k))
    (fgO : (RingHom.ker (algebraMap O k)).FG)
    (fg : ∀ i, (RingHom.ker (res i)).FG) :
    IsLocalRing (CompletedTensor (O := O) R (RingHom.ker (algebraMap O k))
      (fun i ↦ RingHom.ker (res i))) := by
  have := residueIdeal_isMaximal R res hO
  exact AdicModel.completed_isLocalRing _ (definingIdeal_fg R _ _ fgO fg)

/-- The manuscript's Noetherian hypotheses discharge finite generation internally. -/
theorem completedTensor_isLocalRing_noetherian [IsNoetherianRing O]
    [∀ i, IsNoetherianRing (R i)] (hO : Surjective (algebraMap O k)) :
    IsLocalRing (CompletedTensor (O := O) R (RingHom.ker (algebraMap O k))
```

```
      (fun i ↦ RingHom.ker (res i))) :=
  completedTensor_isLocalRing R res hO (Ideal.fg_of_isNoetherianRing _)
    (fun _i ↦ Ideal.fg_of_isNoetherianRing _)

/-- Proper quotients retain locality. The zero-ring case remains covered by the ring equivalence. -/
theorem completed_tensor_quotient_isLocalRing [IsNoetherianRing O]
    [∀ i, IsNoetherianRing (R i)] (hO : Surjective (algebraMap O k))
    (J : ∀ i, Ideal (R i))
    [Nontrivial (CompletedTensor (O := O) R (RingHom.ker (algebraMap O k))
      (fun i ↦ RingHom.ker (res i)) / closedConditions R (RingHom.ker (algebraMap O k))
        (fun i ↦ RingHom.ker (res i)) J)] :
    IsLocalRing (CompletedTensor (O := O) R (RingHom.ker (algebraMap O k))
      (fun i ↦ RingHom.ker (res i)) / closedConditions R (RingHom.ker (algebraMap O k))
        (fun i ↦ RingHom.ker (res i)) J) := by
  have := completedTensor_isLocalRing_noetherian R res hO
  exact IsLocalRing.of_surjective' (Ideal.Quotient.mk _) Ideal.Quotient.mk_surjective

/-- The completion topology is the topology of its actual extended defining ideal. -/
theorem completedTensor_isAdic (mO : Ideal O) (m : ∀ i, Ideal (R i))
    (fgO : mO.FG) (fg : ∀ i, (m i).FG) :
    IsAdic ((definingIdeal (O := O) R mO m).map (tensorEmbed R mO m).toRingHom) :=
  AdicModel.completed_isAdic _ (definingIdeal_fg R mO m fgO fg)
```

The completed universal map also has explicit topological control. Its target P is a complete Hausdorff uniform topological ring with O-algebra structure, m_P-adic topology, and factor maps carrying both the scalar and factor defining ideals into m_P. The raw family map is proved continuous, extended to the actual completion, and shown unique by density and finite tensor extensionality. The final theorem on existence and uniqueness is an output of that construction.

Source: `PaperXII/CompletedTensorUniversal.lean`, lines 15–23.

```
  [IsTopologicalRing P] [CompleteSpace P] [T2Space P]
variable (mP : Ideal P) (hP : IsAdic mP) (f : ∀ i, R i →ₐ[O] P)
  (hO : mO.map (algebraMap O P) ≤ mP) (hf : ∀ i, (m i).map (f i).toRingHom ≤ mP)

omit [UniformSpace P] [IsUniformAddGroup P] [IsTopologicalRing P] [CompleteSpace P] [T2Space P] in
include hO hf in
/-- Local ideal control extends from the factors to the actual algebraic tensor. -/
theorem liftFamily_definingIdeal : (definingIdeal R mO m).map (liftFamily R f).toRingHom ≤ mP := by
  rw [definingIdeal, Ideal.map_sup, sup_le_iff, Ideal.map_map]
```

Source: `PaperXII/CompletedTensorUniversal.lean`, lines 83–99.

```
  exact DFunLike.congr_fun hr (AdicModel.rawEquiv _ _ r)

include hP hO hf in
/-- Existence and uniqueness of the continuous algebra map specified by the local factors. -/
theorem exists_unique_completedLift :
    ∃! g : CompletedTensor R mO m →ₐ[O] P,
      Continuous g ∧ ∀ i, g.comp (completedInclude R mO m i) = f i := by
  refine ⟨completedLift R mO m mP hP f hO hf, ⟨completedLift_continuous .., ?_⟩, ?_⟩
  · intro i; ext x; exact completedLift_factor R mO m mP hP f hO hf i x
  · intro g hg
    apply completedTensor_hom_ext R mO m hg.1 (completedLift_continuous ..)
    intro i
    rw [hg.2 i]
    ext x
    exact (completedLift_factor R mO m mP hP f hO hf i x).symm
```

```
98
99 end PaperXII.FiniteTensor
```

The empty family retains m_O's adic topology. Zero conditions give the identity comparison. A unit condition produces the zero ring and still satisfies the ring comparison. Locality of a quotient is claimed only with an explicit nontriviality premise. The zero ring is not assigned the manuscript's nonzero local ring and residue field. Source boundary checks cover these cases separately.

7 Claim 3.3: completed global base change and its points

Let A be the constructed completed local tensor, let J be its closed condition ideal, and let C be the independently completed tensor of R_i/I_i. The A-algebra structure on C is built from the proved Claim 3.1 equivalence $A/J \cong C$. `quotientIdentification_ideal` proves that the transported defining ideal is the one defining the actual condition tensor topology.

Given the supplied global ring B and its structural A-algebra map, the global condition ring is the actual completion of $B \otimes_A C$. The transport code first constructs the tensor with A/J, then transports along the proved quotient identification. It is not an assumed pushout with a supplied point-factorization field. For complete B, the actual ring is bicontinuously identified with $B/\overline{JB}$, and its canonical global map is surjective.

The reusable ring-point theorem assumes finite generation of the scalar and factor defining ideals, B's declared adic topology, and an ideal bound for the structural map. Its target is a complete Hausdorff uniform topological ring. It proves that a unique continuous point factor exists precisely when each original local restriction kills I_i.

Source: `PaperXII/CompletedLocalConditions.lean`, lines 17–38.

```
17 variable (R : ι → Type*) [∀ i, CommRing (R i)] [∀ i, Algebra O (R i)]
18 variable (mO : Ideal O) (m : ∀ i, Ideal (R i)) (I : ∀ i, Ideal (R i))
19
20 abbrev LocalTensor := FiniteTensor.CompletedTensor R mO m
21 abbrev Conditions := FiniteTensor.CompletedTensor (fun i ↦ R i / I i) mO
22   (fun i ↦ (m i).map (Ideal.Quotient.mk (I i)))
23 abbrev closedIdeal := FiniteTensor.closedConditions R mO m I
24 abbrev localIdeal := (FiniteTensor.definingIdeal R mO m).map
25   (FiniteTensor.tensorEmbed R mO m).toRingHom
26
27 /-- The actual condition tensor receives the canonical local quotient action. -/
28 instance conditionAlgebra : Algebra (LocalTensor R mO m) (Conditions R mO m I) :=
29   ((FiniteTensor.completedQuotientEquiv R mO m I).toRingHom.comp
30     (Ideal.Quotient.mk (closedIdeal R mO m I))).toAlgebra
31
32 /-- Claim 3.1 upgraded to the local-tensor coefficient algebra. -/
33 def quotientIdentification : (LocalTensor R mO m / closedIdeal R mO m I) ≃ₐ[LocalTensor R mO m]
34     Conditions R mO m I where
35   __ := FiniteTensor.completedQuotientEquiv R mO m I
36   commutes' _ := rfl
37
38 /-- The transported ideal is exactly the one defining the actual condition tensor topology. -/
```

Source: `PaperXII/CompletedLocalConditions.lean`, lines 66–100.

```
66
67 /-- The defining construction provides genuine completeness and separation. -/
```

```
theorem globalRing_complete : CompleteSpace (GlobalRing R m0 m I mB) ∧
    T2Space (GlobalRing R m0 m I mB) := (inferInstance, inferInstance)

variable [UniformSpace B] [IsUniformAddGroup B] [IsTopologicalRing B]
variable {P : Type*} [CommRing P] [UniformSpace P] [IsUniformAddGroup P]
  [IsTopologicalRing P] [CompleteSpace P] [T2Space P]

omit [IsUniformAddGroup B] in
/-- Full Claim 3.3: the named completed base change imposes exactly all local conditions. -/
theorem global_base_change (fg0 : m0.FG) (fg : ∀ i, (m i).FG) (hB : IsAdic mB)
    (h : (localIdeal R m0 m).map (algebraMap (LocalTensor R m0 m) B) ≤ mB)
    (p : B →+* P) (hp : Continuous p) :
    (∃! g : GlobalRing R m0 m I mB →+* P,
      Continuous g ∧ g.comp (globalMap R m0 m I mB) = p) ↔
        ∀ i, I i ≤ RingHom.ker ((p.comp (algebraMap (LocalTensor R m0 m) B)).comp
          (FiniteTensor.completedInclude R m0 m i).toRingHom) := by
  have hp' : Continuous (p.comp (AdicModel.rawEquiv B mB).toRingHom) :=
    hp.comp (AdicModel.continuous_of_map_le (AdicModel.carrier_isAdic B mB) hB
      (AdicModel.rawEquiv B mB).toRingHom (by
        change mB.map (RingHom.id B) ≤ mB
        simp))
  refine (CompletedBaseChange.transported_exists_unique_factor_iff (localIdeal R m0 m) mB
    (closedIdeal R m0 m I) (quotientIdentification R m0 m I) h p hp').trans ?_
  change FiniteTensor.closedConditions R m0 m I ≤ RingHom.ker
    (p.comp (algebraMap (LocalTensor R m0 m) B)) ↔ _
  rw [FiniteTensor.closedConditions_eq]
  apply LocalQuotients.closed_conditions_iff R
    (fun i ↦ (FiniteTensor.completedInclude R m0 m i).toRingHom) I
  exact hp.comp (AdicModel.continuous_of_map_le
    (FiniteTensor.completedTensor_isAdic R m0 m fg0 fg) hB
      (algebraMap (LocalTensor R m0 m) B) h)

end PaperXII.CompletedLocalConditions
```

The manuscript capstone uses Noetherianity and common residue maps to supply finite generation. It requires B local and the structural algebra map local. The resulting maximal ideal bound is proved internally. Locality is an essential premise here. A bare continuous ring map is not the full printed capstone's input. The final traditional manuscript exposes these inherited conventions directly in Claim 3.3 and spells out the point category rather than relying on conventions elsewhere.

Source: `PaperXII/CompletedPaperCapstone.lean`*, lines 17–69.*

```
variable (R : ι → Type*) [∀ i, CommRing (R i)] [∀ i, Algebra O (R i)]
variable (res : ∀ i, R i →ₐ[O] k) (hO : Function.Surjective (algebraMap O k))
variable [IsNoetherianRing O] [∀ i, IsNoetherianRing (R i)]

include hO in
/-- The ideal defining the actual local tensor topology is its maximal ideal. -/
theorem localIdeal_eq_maximalIdeal :
    haveI := FiniteTensor.completedTensor_isLocalRing_noetherian R res hO
    localIdeal R (RingHom.ker (algebraMap O k)) (fun i ↦ RingHom.ker (res i)) =
      IsLocalRing.maximalIdeal (LocalTensor R (RingHom.ker (algebraMap O k))
        (fun i ↦ RingHom.ker (res i))) := by
  have hm := FiniteTensor.residueIdeal_isMaximal R res hO
  exact (AdicModel.completed_maximalIdeal _ (FiniteTensor.definingIdeal_fg R _ _
    (Ideal.fg_of_isNoetherianRing _) (fun _i ↦ Ideal.fg_of_isNoetherianRing _))).symm

variable (I : ∀ i, Ideal (R i))
variable {B : Type*} [CommRing B]
  [Algebra (LocalTensor R (RingHom.ker (algebraMap O k)) (fun i ↦ RingHom.ker (res i))) B]
  [IsLocalRing B]
  [IsLocalHom (algebraMap (LocalTensor R (RingHom.ker (algebraMap O k))
```

```
    (fun i ↦ RingHom.ker (res i))) B)]

include hO in
/-- A local global coefficient map automatically respects the required adic ideals. -/
theorem local_map_ideal_bound :
    (localIdeal R (RingHom.ker (algebraMap O k)) (fun i ↦ RingHom.ker (res i))).map
      (algebraMap (LocalTensor R (RingHom.ker (algebraMap O k))
        (fun i ↦ RingHom.ker (res i))) B) ≤ IsLocalRing.maximalIdeal B := by
  have := FiniteTensor.completedTensor_isLocalRing_noetherian R res hO
  rw [localIdeal_eq_maximalIdeal R res hO]
  exact IsLocalRing.map_maximalIdeal_le _

variable [UniformSpace B] [IsUniformAddGroup B] [IsTopologicalRing B]
variable {P : Type*} [CommRing P] [UniformSpace P] [IsUniformAddGroup P]
  [IsTopologicalRing P] [CompleteSpace P] [T2Space P]

omit [IsUniformAddGroup B] in
include hO in
/-- Claim 3.3 at the manuscript's Noetherian and local-map boundary. -/
theorem global_base_change_noetherian (hB : IsAdic (IsLocalRing.maximalIdeal B))
    (p : B →+* P) (hp : Continuous p) :
    (∃! g : GlobalRing R (RingHom.ker (algebraMap O k))
      (fun i ↦ RingHom.ker (res i)) I (IsLocalRing.maximalIdeal B) →+* P,
      Continuous g ∧ g.comp (globalMap R (RingHom.ker (algebraMap O k))
        (fun i ↦ RingHom.ker (res i)) I (IsLocalRing.maximalIdeal B)) = p) ↔
      ∀ i, I i ≤ RingHom.ker ((p.comp (algebraMap (LocalTensor R
        (RingHom.ker (algebraMap O k)) (fun i ↦ RingHom.ker (res i))) B)).comp
          (FiniteTensor.completedInclude R (RingHom.ker (algebraMap O k))
            (fun i ↦ RingHom.ker (res i)) i).toRingHom) :=
  global_base_change R _ _ I _ (Ideal.fg_of_isNoetherianRing _)
    (fun _i ↦ Ideal.fg_of_isNoetherianRing _) hB (local_map_ideal_bound R res hO) p hp

end PaperXII.CompletedLocalConditions
```

For continuous local O-algebra points, the production point theorem additionally has B complete, actual O-algebra structures on B and P, and an original point which is an O-algebra map reflecting units. The completed global coefficient action is the actual composite through B. Surjectivity of the global map proves that the unique ring factor reflects units. Equality on coefficients proves it is an O-algebra map. Naturality follows from equality after the surjective global map.

Source: `PaperXII/CompletedPointCategory.lean`, *lines 49–92.*

```
  [IsUniformAddGroup P] [IsTopologicalRing P] [CompleteSpace P] [T2Space P]

/-- Claim 3.3 as an equivalence of continuous local algebra points. -/
theorem local_algebra_point_iff (fgO : mO.FG) (fg : ∀ i, (m i).FG)
    (hB : IsAdic mB)
    (h : (localIdeal R mO m).map (algebraMap (LocalTensor R mO m) B) ≤ mB)
    (p : B →ₐ[O] P) [IsLocalHom p.toRingHom] (hp : Continuous p) :
    (∃! g : GlobalRing R mO m I mB →ₐ[O] P,
      Continuous g ∧ IsLocalHom g.toRingHom ∧ g.comp (globalAlgMap R mO m I mB) = p) ↔
      ∀ i, I i ≤ RingHom.ker ((p.toRingHom.comp (algebraMap (LocalTensor R mO m) B)).comp
        (FiniteTensor.completedInclude R mO m i).toRingHom) := by
  constructor
  · rintro ⟨g, ⟨hg, _, hgp⟩, _⟩
    apply (global_base_change R mO m I mB fgO fg hB h p.toRingHom hp).mp
    refine ⟨g.toRingHom, ⟨hg, ?_⟩, ?_⟩
    · exact congrArg AlgHom.toRingHom hgp
    · intro g' hg'
      ext x
      obtain ⟨b, rfl⟩ := globalMap_surjective R mO m I mB hB h x
      exact (DFunLike.congr_fun hg'.2 b).trans
```

```
        (DFunLike.congr_fun (congrArg AlgHom.toRingHom hgp) b).symm
  · intro hI
    obtain ⟨g, ⟨hg, hgp⟩, hu⟩ :=
      (global_base_change R mO m I mB fgO fg hB h p.toRingHom hp).mpr hI
    let ga : GlobalRing R mO m I mB →ₐ[O] P :=
      { __ := g
        commutes' := fun a ↦ (DFunLike.congr_fun hgp (algebraMap O B a)).trans (p.commutes a) }
    refine ⟨ga, ⟨hg, point_factor_isLocal R mO m I mB hB h p.toRingHom g hgp, ?_⟩, ?_⟩
    · exact AlgHom.ext (fun b ↦ DFunLike.congr_fun hgp b)
    · intro g' hg'
      apply AlgHom.ext
      intro x
      exact DFunLike.congr_fun (hu g'.toRingHom ⟨hg'.1, congrArg AlgHom.toRingHom hg'.2.2⟩) x

omit [UniformSpace P] [IsUniformAddGroup P] [IsTopologicalRing P] [CompleteSpace P] [T2Space P] in
/-- The point factorization is natural under coefficient-preserving target maps. -/
theorem algebra_factor_natural {Q : Type*} [CommRing Q] [Algebra O Q]
    (hB : IsAdic mB)
    (h : (localIdeal R mO m).map (algebraMap (LocalTensor R mO m) B) ≤ mB)
    (p : B →ₐ[O] P) (u : P →ₐ[O] Q)
    (g : GlobalRing R mO m I mB →ₐ[O] P)
    (g' : GlobalRing R mO m I mB →ₐ[O] Q)
    (hg : g.comp (globalAlgMap R mO m I mB) = p)
    (hg' : g'.comp (globalAlgMap R mO m I mB) = u.comp p) :
```

The ring-point result, local algebra-point result and Noetherian/local specialization are distinct declarations and must be read together. A local deformation test point is distinct from a characteristic-zero field specialization, where a coefficient uniformizer can be a unit. Continuity of the specialized representation is separately required by realization. The supplementary `claim33` consumes the local point statements using hypotheses from the manuscript, without an extra finite-generation or ideal-bound hypothesis. It does not prove arithmetic representability of a deformation functor. Representability, universal arithmetic deformation representations, and existence of the local quotient rings remain supplied inputs.

8 Ordinary support, patching and intrinsic Fitting ideals

For a finite module, production support is Mathlib's actual localization support on the prime spectrum. Lean proves

$$\operatorname{Supp}_R M = V(\operatorname{Ann}_R M).$$

Under the Noetherian hypotheses, near faithfulness is equivalent to full support. Quotient support is obtained on corresponding primes using actual quotient modules and localization/Nakayama arguments. Ordinary patching then uses supplied coefficient and module control isomorphisms to descend full support from the patched finite module. Finite generation and finite-level full support are outputs.

An arithmetic patching theorem supplying the patched ring, module, control maps and full patched support remains external. The finite-level R-action on the same carrier must be restriction of its Hecke action through the supplied comparison, as expressed by the ambient scalar-tower hypotheses. Derived carriers require the corresponding transport of the actual complex. Two unrelated actions on a carrier do not supply this compatibility. Freeness over an auxiliary algebra is insufficient for full deformation support. The power-series counterexample demonstrates the missing implication. A ring isomorphism $R \cong \mathbb{T}$ transports existing support. A surjection alone can erase a component, as in the actual node $k[[x,y]]/(xy)$ followed by quotienting by (x).

The Fitting block of ten modules constructs the determinantal ideals of a finite presentation and proves their invariance under changes of presentation, generator and relation counts. Its intrinsic definition chooses a presentation using classical choice and then proves equality with the minors of every actual finite presentation. No theorem on presentation independence is supplied as a hypothesis.

Source: `PaperXII/IntrinsicFitting.lean`*, lines 8–39.*

```
variable (R : Type*) [CommRing R] (M : Type*) [AddCommGroup M] [Module R M]

/-- Actual finite free relations, generators, and a surjective quotient map. -/
structure FiniteMatrixPresentation where
  generators : ℕ
  relations : ℕ
  matrix : Matrix (Fin generators) (Fin relations) R
  quotient : (Fin generators → R) →ₗ[R] M
  presents : Presents matrix quotient

instance finiteMatrixPresentation_nonempty [Module.FinitePresentation R M] : Nonempty (FiniteMatrixPresentation R M) := by
  classical
  obtain (n,m,f,g,hf,hgf) := Module.FinitePresentation.exists_fin' R M
  refine ((n,m,LinearMap.toMatrix' g,f,(hf,?_)))
  simpa only [← Matrix.toLin'_apply',Matrix.toLin'_toMatrix'] using hgf.linearMap_ker_eq.symm

/-- A chosen finite presentation. Its choice is removed by `fittingIdeal_eq_presentation`. -/
def chosenPresentation [Module.FinitePresentation R M] : FiniteMatrixPresentation R M :=
  Classical.choice inferInstance

/-- The intrinsic j-th Fitting ideal of a finitely presented module. -/
def fittingIdeal [Module.FinitePresentation R M] (j : ℕ) : Ideal R :=
  presentationIdeal (chosenPresentation R M).matrix j

variable {R M}

/-- The intrinsic ideal equals the minors computed from any actual finite presentation. -/
theorem fittingIdeal_eq_presentation [Module.FinitePresentation R M]
    {m n : Type*} [Fintype m] [Fintype n]
    (A : Matrix m n R) (π : (m → R) →ₗ[R] M) (h : Presents A π) (j : ℕ) :
    fittingIdeal R M j = presentationIdeal A j :=
  presentationIdeal_independent _ _ _ _ (chosenPresentation R M).presents h j
```

The arbitrary base-change theorem transports each minor under the ring map. The support criterion and all higher strata are stated for finitely presented modules. Noetherianity plus finite generation in the manuscript supplies that premise. For a presentation with n generators,

$$\dim_{\kappa(\mathfrak{p})}(M \otimes \kappa(\mathfrak{p})) = n - \operatorname{rank} A(\mathfrak{p}).$$

For $1 \le r \le n$, the inequality dim $\ge r$ is detected by minors of size $n - r + 1$. If $r > n$, the fiber locus is empty and $\mathrm{Fitt}_{r-1}(M) = R$. The final ordinary proof explicitly separates this boundary. The formal block already includes upper-index normalization and empty/zero-minor checks. Fitting strata count minimal fiber generators, not automorphic multiplicities.

9 Derived support uses actual perfect models

`PaperXII.PerfectDerivedSupport` does not encode a complex by a bare list of supports. Its `PerfectComplex` contains an actual Mathlib object of the derived category, a bounded finite-projective cochain model, and an actual derived isomorphism from that model. Base change is computed degree by degree on the

projective model. The code proves compatibility with actual derived isomorphisms, so support does not depend on the chosen model.

Source: `PaperXII/PerfectDerivedSupport.lean`, *lines 23–53.*

```
23
24 open PerfectModels
25
26 /-- An arbitrary perfect derived object with a bounded finite-projective model. -/
27 structure PerfectComplex (R : Type u) [CommRing R] where
28   object : DerivedCategory (ModuleCat.{u} R)
29   model : PerfectModel R
30   modelIso : model.derivedObject ≅ object
31
32 variable {R S T : Type u} [CommRing R] [CommRing S] [CommRing T]
33
34 /-- A bounded projective model determines a perfect derived object. -/
35 def ofModel (P : PerfectModel R) : PerfectComplex R :=
36   ⟨P.derivedObject, P, Iso.refl _⟩
37
38 /-- Derived scalar extension, computed on the finite-projective model. -/
39 def PerfectComplex.baseChange (P : PerfectComplex R) (f : R →+* S) : PerfectComplex S :=
40   ofModel (P.model.baseChange f)
41
42 /-- Every actual derived isomorphism induces one after arbitrary scalar extension. -/
43 def baseChangeIso (f : R →+* S) {P Q : PerfectComplex R}
44     (e : P.object ≅ Q.object) : (P.baseChange f).object ≅ (Q.baseChange f).object :=
45   baseChange_derivedIso f P.model Q.model (P.modelIso <<>> e <<>> Q.modelIso.symm)
46
47 /-- Successive derived scalar extensions agree with extension by the composite. -/
48 def baseChangeCompIso (P : PerfectComplex R) (f : R →+* S) (g : S →+* T) :
49     ((P.baseChange f).baseChange g).object ≅ (P.baseChange (g.comp f)).object :=
50   ComplexSupport.fiberCompIso f g P.model.complex
51
52 /-- Identity scalar extension recovers the arbitrary represented derived object. -/
53 def baseChangeIdIso (P : PerfectComplex R) :
```

Derived support is the set of primes with nonzero residue-field derived fiber. Scalar extension has spectral inverse-image support. The quotient identity is expressed as an image in the original spectrum. It includes the unit-ideal quotient. Field-valued points are handled at their actual kernel primes and then by faithful flatness of the field extension, without imposing a closed-point restriction.

Source: `PaperXII/PerfectDerivedSupport.lean`, *lines 56–96.*

```
56
57 /-- The residue-field definition of derived support in the original manuscript. -/
58 def support (P : PerfectComplex R) : Set (PrimeSpectrum R) :=
59   ComplexSupport.support P.model.complex
60
61 /-- Full support of a perfect derived object. -/
62 def FullSupport (P : PerfectComplex R) : Prop := support P = Set.univ
63
64 /-- Derived support is independent of both model and representative. -/
65 theorem support_eq_of_iso {P Q : PerfectComplex R} (e : P.object ≅ Q.object) :
66     support P = support Q := by
67   ext p
68   exact not_congr (baseChangeIso (algebraMap R p.asIdeal.ResidueField) e).isZero_iff
69
70 /-- The field-valued criterion uses the exact prime kernel of the point. -/
71 theorem nonzero_fiber_iff_support {K : Type u} [Field K] (P : PerfectComplex R)
72     (x : R →+* K) :
```

```
    ¬IsZero (P.baseChange x).object ↔ FieldFibers.pointPrime x ∈ support P :=
  ComplexSupport.nonzero_fiber_iff_support x P.model.complex

/-- Derived support under every scalar map is the actual spectral inverse image. -/
theorem support_baseChange (P : PerfectComplex R) (f : R →+* S) :
    support (P.baseChange f) = PrimeSpectrum.comap f ⁻¹' support P :=
  ComplexSupport.support_baseChange f P.model.complex

/-- Original Theorem 6.16, in literal ambient-spectrum form for the actual quotient. -/
theorem derived_quotient_support (P : PerfectComplex R) (I : Ideal R) :
    PrimeSpectrum.comap (Ideal.Quotient.mk I) '' support (P.baseChange (Ideal.Quotient.mk I)) =
      support P ∩ PrimeSpectrum.zeroLocus I :=
  ComplexSupport.image_support_quotient I P.model.complex

/-- Support transports along a ring isomorphism by its actual map on spectra. -/
theorem support_ringEquiv (P : PerfectComplex R) (e : R ≃+* S) :
    PrimeSpectrum.comap e.toRingHom '' support (P.baseChange e.toRingHom) = support P := by
  rw [support_baseChange, Set.image_preimage_eq_inter_range]
  have hsurj : Function.Surjective (PrimeSpectrum.comap e.toRingHom) :=
    (PrimeSpectrum.isHomeomorph_comap_of_bijective e.bijective).surjective
  rw [Set.range_eq_univ.mpr hsurj, Set.inter_univ]

/-- Full support survives arbitrary derived scalar extension. -/
theorem fullSupport_baseChange (P : PerfectComplex R) (f : R →+* S)
```

A nonzero field fiber yields some nonzero actual derived cohomology object in an integer degree. The bounded projective model proves vanishing outside its bounds. The induced Hecke action comes from multiplication on the original cochain model, through scalar extension and passage to derived cohomology, and equals the specialized scalar character.

These are formal algebraic conclusions. They neither choose a classical weight nor establish classicality. Revised Claim 9.6 requires the classicality range to contain the model interval. Revised 12.4 permits applicability at the selected eigensystem itself. The original unrestricted versions remain partial in their original ledger. The cited external cohomological description of derived support is not needed by this production residue-field support proof.

10 Artin packets, residual augmentation and finite adequacy

Hecke packet construction starts with a supplied canonical ring decomposition into actual localizations. The input is `ArtinData.DecompositionR`, an R-algebra equivalence whose coordinates are proved to be the canonical localization maps. It supplies no module direct-sum decomposition and no module localization equivalence. Those are constructed subsequently.

Source: `PaperXII/ArtinData.lean`*, lines 17–41.*

```

/-- The canonical ring decomposition supplied by external Fact 8.1. -/
abbrev Decomposition (R : Type*) [CommRing R] :=
  R ≃ₐ[R] (∀ m : MaximalSpectrum R, Localization.AtPrime m.asIdeal)

/-- The actual central idempotent indexed by a maximal ideal. -/
noncomputable def idempotent (D : Decomposition R) (m : MaximalSpectrum R) : R := by
  classical
  exact D.symm (Pi.single m 1)

/-- Every coordinate is the canonical localization map, by R-algebra linearity. -/
theorem coordinate (D : Decomposition R) (r : R) (m : MaximalSpectrum R) :
```

```
    D r m = algebraMap R (Localization.AtPrime m.asIdeal) r := by
  have h := D.commutes r
  exact congrFun h m

/-- The canonical coordinate maps determine the decomposition uniquely. -/
theorem decomposition_unique (D₁ D₂ : Decomposition R) : D₁ = D₂ := by
  ext r m
  rw [coordinate, coordinate]

/-- The coordinate-vector identity determines the idempotent uniquely. -/
theorem map_idempotent [DecidableEq (MaximalSpectrum R)]
    (D : Decomposition R) (m : MaximalSpectrum R) :
    D (idempotent D m) = Pi.single m 1 := by
```

From this input, Lean constructs orthogonal idempotents, their packet submodules, the direct sum, and the equivalence between the packet and the localized module. It verifies the map $v \mapsto v/1$. The selected local factor's nilpotent maximal ideal and an invertible generator on every other factor identify the packet with the simultaneous generalized eigenspace for one positive exponent. This does not assume semisimplicity. In characteristic zero, the projector trace is the dimension of its image cast into the field, and nonzero trace detects occupancy. Trace cancellation in positive characteristic remains a real boundary.

Auxiliary group-algebra transfer is constructed on the actual group basis and commutes with actual augmentation. Coinvariant recovery proves a quotient by all balancing relations and identifies it with the ordinary tensor product when the appropriate algebra is commutative. The right-module treatment handles arbitrary groups, including nonabelian two-groups. A supplied free basis descends to a coefficient basis with the same index set, including rank zero.

For a product of cyclic p-groups in characteristic p, the full multivariable truncated-polynomial algebra equivalence is constructed and its augmentation generators identified. The result excluding nontrivial idempotents covers every finite p-group, including nonabelian ones, by an orbit-span/fixed-vector argument over the prime field. The dyadic code proves that rational sign projectors are not integral and constructs $\mathbb{F}_2[C_2] \cong \mathbb{F}_2[X]/(X^2)$. None of this supplies dyadic arithmetic image, local geometry or classicality hypotheses.

The S_3 example over $\mathbb{F}_7$ is a Lean certificate for one finite representation. The modules construct the standard representation from the permutation module, prove absolute irreducibility, identify its trace-zero adjoint with sign plus standard, and connect averaging to actual first group cohomology. Canonical generalized-eigenspace projectors detect every nonzero adjoint constituent. Independent finite computations can check the printed matrices and trace pairings. Their scope remains separate from the formal certificate and from arithmetic adequacy applications. The adequacy convention remains a cited input.

11 Occupancy, stable cores and adic completion of classes

In one finite-dimensional carrier, nested idempotent ranges stabilize. In that carrier, nonzero ranges at every level yield a nonzero intersection. In changing carriers, the inverse limit is controlled by the actual intersections of transition images, not merely by nonzero individual levels. `PaperXII.StableCores` uses a `LinearSystem` with actual linear maps and identity/composition laws. It proves that transition images are antitone, that finite-dimensionality makes each stable core an eventual image, and that the core transition is surjective. Recursive lifting then produces a compatible nonzero vector above a chosen nonzero core coordinate.

Source: `PaperXII/StableCores.lean`*, lines 6–38.*

```
/-- A countable inverse system, with the ordinary identity and composition laws. -/
structure LinearSystem (K : Type*) [Field K] (V : ℕ → Type*)
    [∀ n, AddCommGroup (V n)] [∀ n, Module K (V n)] where
  map : ∀ n m, m ≤ n → V n →ₗ[K] V m
  map_self : ∀ n, map n n le_rfl = LinearMap.id
  map_comp : ∀ (a b c) (hcb : c ≤ b) (hba : b ≤ a),
    (map b c hcb).comp (map a b hba) = map a c (hcb.trans hba)

variable {K : Type*} [Field K] {V : ℕ → Type*}
    [∀ n, AddCommGroup (V n)] [∀ n, Module K (V n)]

/-- Padded image family: indices before the target level contribute the whole space. -/
def imageAt (D : LinearSystem K V) (m n : ℕ) : Submodule K (V m) :=
  iInf h : m ≤ n, LinearMap.range (D.map n m h)

/-- The actual intersection of the images from every later level. -/
def stableCore (D : LinearSystem K V) (m : ℕ) : Submodule K (V m) :=
  iInf n, imageAt D m n

/-- Membership is precisely membership in every later transition image. -/
theorem mem_stableCore_iff (D : LinearSystem K V) (m : ℕ) (v : V m) :
    v ∈ stableCore D m ↔ ∀ n (h : m ≤ n), ∃ w, D.map n m h w = v := by
  simp only [stableCore, imageAt, Submodule.mem_iInf, LinearMap.mem_range]

/-- A valid later index has its literal transition range as padded image. -/
theorem imageAt_eq_range (D : LinearSystem K V) {m n : ℕ} (h : m ≤ n) :
    imageAt D m n = LinearMap.range (D.map n m h) := by
  simp [imageAt, h]

/-- Composition makes the transition images descend. -/
theorem imageAt_antitone (D : LinearSystem K V) (m : ℕ) : Antitone (imageAt D m) := by
  intro a b hab
  apply le_iInf
```

Finite companion forests give branches when every finite level is nonempty. An empty level obstructs complete realizations whose actual finite reductions belong to those declared levels. It does not rule out representations excluded by an arbitrary admissibility restriction. The production `empty_level_no_realization` explicitly takes reduction maps from the supplied realization type into every forest level. Forests do not alone produce continuous integral representations with every arithmetic condition. Revised Claim 10.5 makes the missing effectivity inputs explicit: complete separated coefficient rings for finitely generated ideals, finite residue rings, finite occupied classes of actual continuous quotient representations, reduction compatibility, continuous coefficient maps, integral Frobenius polynomials with matching scalar changes, and declared polynomial ideal conditions valid in every admissible frame.

Finite frames turn finite admissible conjugacy classes into finite sets of actual representatives. Compactness selects strictly compatible representatives without assuming transition surjectivity or lifting a prescribed conjugating matrix. Entrywise reconstruction, including inverse entries, constructs the actual inverse limit of matrix groups. Continuity is derived from finite quotient continuity and the adic topology. Separatedness upgrades compatible polynomial and subgroup-triviality congruences to exact integral equalities. Evaluation then kills each declared polynomial ideal and factors through its actual quotient.

Source: `PaperXII/ConditionalCompletion.lean`*, lines 8–54.*

```

open ContinuousMonoidHom Matrix
```

```
open AdicMatrixGroups AdicCompanions ContinuousClasses ClassCompanions AdicSpines
open AdicLocalConditions LocalQuotients PaperX.ClassInvariants

variable {Γ ι : Type} [Group Γ] [TopologicalSpace Γ] [Fintype ι] [DecidableEq ι]

/-- Positive finite class forests realize the prescribed common spine and every
declared polynomial quotient condition on the actual representation entries. -/
theorem simultaneous_companion_completion
    {Λ Ω L V : Type} [Field L] (O K : Λ → Type)
    [∀ l, CommRing (O l)] [∀ l, Field (K l)] [∀ l, TopologicalSpace (K l)]
    (J : ∀ l, Ideal (O l)) [∀ l, IsAdicComplete (J l) (O l)]
    (hJ : ∀ l, (J l).FG) [∀ l, Finite (O l ⧸ J l)]
    (C : ∀ l n, LevelClass (Γ := Γ) (ι := ι) (J l) n → Prop)
    (hC : ∀ l n c, C l (n + 1) c → C l n (classStep (J l) n c))
    (finiteC : ∀ l n, 0 < n → Finite {c // C l n c})
    (hc : ∀ l n, 0 < n → ∀ c, C l n c → ClassContinuous c)
    (hne : ∀ l n, 0 < n → ∃ c, C l n c)
    (root : ∀ l, LevelClass (Γ := Γ) (ι := ι) (J l) 1)
    (hroot : ∀ l c, C l 1 c → c = root l)
    (inc : ∀ l, O l →+* K l)
    (hinc : ∀ l, @Continuous (O l) (K l) (J l).adicTopology inferInstance (inc l))
    (emb : ∀ l, L →+* K l) (frob : Ω → Γ) (good : Λ → Ω → Prop)
    (P : Ω → Polynomial L) (localP : ∀ l, Ω → Polynomial (O l))
    (hmatch : ∀ l v, good l v → (localP l v).map (inc l) = (P v).map (emb l))
    (hspec : ∀ l n, 0 < n → ∀ c, C l n c → ∀ v, good l v →
      classCharpoly c (frob v) = (localP l v).map (quotientMap (J l) n))
    (H : Λ → Subgroup Γ)
    (hH : ∀ l n, 0 < n → ∀ c, C l n c → ClassTrivial (H l) c)
    (relations : ∀ l, V → Ideal (MvPolynomial (Coordinates (Γ := Γ) (ι := ι)) (O l)))
    (hrelations : ∀ l v n, 0 < n → ∀ σ, Framed (J l) (C l) n σ →
      relations l v ≤ RingHom.ker (finitePoint (J l) n σ)) :
    ∃ ρ : ∀ l, AdicRealization (J l) (Framed (J l) (C l)),
      (∀ l, classOfRep (reduceRep (J l) 1 (ρ l).val) = root l) ∧
      (∀ l v, good l v →
        (fieldRep (J l) (inc l) (hinc l) (ρ l).val (frob v)).val.charpoly =
          (P v).map (emb l)) ∧
      (∀ l g, g ∈ H l → fieldRep (J l) (inc l) (hinc l) (ρ l).val g = 1) ∧
      ∀ l v, Factors (relations l v) (point (J l) (ρ l).val) := by
  obtain ⟨ρ, hrootρ, hpolyρ, hHρ⟩ := ClassSpines.simultaneous_class_spine
    O K J hJ C hC finiteC hc hne root hroot inc hinc emb frob good P localP
    hmatch hspec H hH
  exact ⟨ρ, hrootρ, hpolyρ, hHρ, fun l v ↦
    class_condition_preserved (J l) (C l) (relations l v) (ρ l) (hrelations l v)⟩

end PaperXII.ConditionalCompletion
```

The conclusion retains a residual conjugacy class, not a prescribed residual matrix frame. Uniqueness up to integral conjugacy is not asserted. No arbitrary Hodge or Weil–Deligne condition is identified with a polynomial ideal without additional arithmetic work. Integral Frobenius lifts and occupancy at every precision are hypotheses, not outcomes of abstract compactness. A missing coefficient-place member is not supplied by propagation inside an existing compatible family.

12 Conditional realization and the five separate gates

The formal realization theorem compares actual matrix representations via an invertible linear intertwiner. It assumes a Hausdorff characteristic-zero topological field with continuous ring operations, continuous compared representations, a supplied semisimplification preserving characteristic polynomials, arithmetic density of the declared Frobenius conjugacy saturation, and a general algebraic Brauer–Nesbitt identification theorem over that coefficient field.

`PaperXII.DenseFrobenius` proves continuity of characteristic-polynomial coefficients and extends equality from the dense Frobenius conjugacy set to all group elements. The supplied Brauer–Nesbitt input then identifies the semisimple representations. Neither density nor the general Brauer–Nesbitt theorem is a newly introduced global Lean axiom: each is an explicit ordinary theorem argument, and the production result is conditional on it.

Source: `PaperXII/GaloisRealization.lean`, lines 13–53.

```
13 variable {E Γ Ω ι : Type*} [Field E] [CharZero E] [Group Γ]
14     [TopologicalSpace E] [IsTopologicalRing E] [T2Space E]
15     [TopologicalSpace Γ] [Fintype ι] [DecidableEq ι]
16
17 /-- The semisimplification of a continuous representation is identified with a
18 semisimple attached representation using only the general standard input. -/
19 theorem pointwise_semisimple_identification (bn : BrauerNesbitt E Γ ι)
20     (ρ σ : Γ →ₜ* GL ι E) (s : Semisimplification ρ.toMonoidHom)
21     (hσ : Representation.IsSemisimpleRepresentation (representation σ.toMonoidHom))
22     (frob : Ω → Γ) (good : Ω → Prop)
23     (hdense : Dense (conjugacySaturation frob good))
24     (hpoly : ∀ v, good v → (ρ (frob v)).val.charpoly = (σ (frob v)).val.charpoly) :
25     Isomorphic s.rep σ.toMonoidHom := by
26   have hall := charpoly_eq_of_dense_frobenius ρ σ frob good hdense hpoly
27   exact bn s.rep σ.toMonoidHom s.semisimple hσ
28     (fun g ↦ (s.charpoly_eq g).trans (hall g))
29
30 /-- The irreducible case gives an isomorphism of the original representations. -/
31 theorem pointwise_irreducible_identification (bn : BrauerNesbitt E Γ ι)
32     (ρ σ : Γ →ₜ* GL ι E)
33     (hρ : Representation.IsIrreducible (representation ρ.toMonoidHom))
34     (hσ : Representation.IsSemisimpleRepresentation (representation σ.toMonoidHom))
35     (frob : Ω → Γ) (good : Ω → Prop)
36     (hdense : Dense (conjugacySaturation frob good))
37     (hpoly : ∀ v, good v → (ρ (frob v)).val.charpoly = (σ (frob v)).val.charpoly) :
38     Isomorphic ρ.toMonoidHom σ.toMonoidHom :=
39   bn ρ.toMonoidHom σ.toMonoidHom (irreducible_semisimple _ hρ) hσ
40     (charpoly_eq_of_dense_frobenius ρ σ frob good hdense hpoly)
41
42 /-- Both printed conclusions for an arbitrary mapped Hecke polynomial spine. -/
43 theorem common_spine_identification {B : Type*} [CommRing B]
44     (bn : BrauerNesbitt E Γ ι) (ρ σ : Γ →ₜ* GL ι E)
45     (s : Semisimplification ρ.toMonoidHom)
46     (hσ : Representation.IsSemisimpleRepresentation (representation σ.toMonoidHom))
47     (frob : Ω → Γ) (good : Ω → Prop)
48     (hdense : Dense (conjugacySaturation frob good)) (x : B →+* E)
49     (P : Ω → Polynomial B)
50     (hρP : ∀ v, good v → (ρ (frob v)).val.charpoly = (P v).map x)
51     (hσP : ∀ v, good v → (σ (frob v)).val.charpoly = (P v).map x) :
52     Isomorphic s.rep σ.toMonoidHom ∧
53       (Representation.IsIrreducible (representation ρ.toMonoidHom) →
```

For a supplied universal representation, `specialize` is literally coefficient change into the chosen field. Its continuity is explicit. The two representations must have the same specialized Hecke polynomials on the same dense good set. The ordinary completion theorem has the universal specialization polynomial identity as a separate explicit premise. Its classicality input supplies the attached representation's matching polynomial identity. The final traditional definition 12.1 spells out these inputs. A nonzero fiber and an attached representation alone would not identify the universal specialization.

Source: `PaperXII/GaloisRealization.lean`, lines 73–97.

```
73
74 /-- Continuity of the specialization under consideration is explicit. -/
75 def continuousSpecialization (ρR : Γ →* GL ι R) (x : R →+* E)
76     (hx : Continuous (specialize ρR x)) : Γ →ₜ* GL ι E :=
77   { __ := specialize ρR x, continuous_toFun := hx }
78
79 /-- Source-shaped pointwise realization: the compared representation is literally
80 x composed with the supplied universal representation. -/
81 theorem universal_pointwise_realization {T : Type*} [CommRing T]
82     (bn : BrauerNesbitt E Γ ι) (ρR : Γ →* GL ι R) (x : R →+* E)
83     (hx : Continuous (specialize ρR x))
84     (s : Semisimplification (specialize ρR x))
85     (σ : Γ →ₜ* GL ι E)
86     (hσ : Representation.IsSemisimpleRepresentation (representation σ.toMonoidHom))
87     (frob : Ω → Γ) (good : Ω → Prop)
88     (hdense : Dense (conjugacySaturation frob good))
89     (y : T →+* E) (P : Ω → Polynomial T)
90     (hρP : ∀ v, good v → (specialize ρR x (frob v)).val.charpoly = (P v).map y)
91     (hσP : ∀ v, good v → (σ (frob v)).val.charpoly = (P v).map y) :
92     Isomorphic s.rep σ.toMonoidHom ∧
93       (Representation.IsIrreducible (representation (specialize ρR x)) →
94         Isomorphic (specialize ρR x) σ.toMonoidHom) :=
95   common_spine_identification bn (continuousSpecialization ρR x hx) σ s hσ
96     frob good hdense y P hρP hσP
97
```

Source: `PaperXII/OrdinaryCompletion.lean`, lines 171–213.

```
171   exact supported_pointwise_realization eT x
172     ((fullSupport_iff_localizations R H).mp hH _) bn ρR (hx x hxU) (s x hxU)
173     frob good hdense P (hρP x hxU) occurs attached (classicality x hxU)
174
175 variable {A Pinf : Type*} [CommRing A] [IsNoetherianRing A] [Algebra A R]
176     [AddCommGroup Pinf] [Module A Pinf] [Module.Finite A Pinf]
177     [Module A H] [IsScalarTower A R H]
178
179 /-- Revised Theorem 12.2, including its actual nonzero tensor fiber, the supplied
180 classical object's occurrence, semisimplified identification, and irreducible
181 identification. Finite-level support and finite generation are derived from
182 the printed patched quotient datum; classicality is used only after this. -/
183 theorem ordinary_conditional_completion (I : Ideal A)
184     (eR : (A ⧸ I) ≃ₐ[A] R)
185     (eM : (Pinf ⧸ (I • ⊤ : Submodule A Pinf)) ≃ₗ[A] H)
186     (eT : R ≃ₐ[R] T) (hP : FullSupport A Pinf) (x : R →+* E)
187     (bn : BrauerNesbitt E Γ ι) (ρR : Γ →* GL ι R)
188     (hx : Continuous (specialize ρR x)) (s : Semisimplification (specialize ρR x))
189     (frob : Ω → Γ) (good : Ω → Prop)
190     (hdense : Dense (conjugacySaturation frob good)) (P : Ω → Polynomial T)
191     (hρP : ∀ v, good v → (specialize ρR x (frob v)).val.charpoly =
192       (P v).map (transportedPoint eT.toRingEquiv x))
193     (occurs : (R →+* E) → Auto → Prop) (attached : Auto → Γ →ₜ* GL ι E)
194     (classicality :
195       letI : Algebra T E := (transportedPoint eT.toRingEquiv x).toAlgebra
196       Nontrivial (E ⊗[T] H) → ∃ π : Auto, occurs x π ∧
197         Representation.IsSemisimpleRepresentation (representation (attached π).toMonoidHom) ∧
198         ∀ v, good v → (attached π (frob v)).val.charpoly =
199           (P v).map (transportedPoint eT.toRingEquiv x)) :
200     letI : Algebra T E := (transportedPoint eT.toRingEquiv x).toAlgebra
201     Nontrivial (H ⊗[T] E) ∧ ∃ π : Auto, occurs x π ∧
202       Isomorphic s.rep (attached π).toMonoidHom ∧
203       (Representation.IsIrreducible (representation (specialize ρR x)) →
204         Isomorphic (specialize ρR x) (attached π).toMonoidHom) := by
205   let : Module.Finite R H := finite_level_finite (R := R) I eM
```

```
  let : Module.Finite T H := Module.Finite.of_restrictScalars_finite R T H
  exact supported_pointwise_realization eT x
    ((fullSupport_iff_localizations R H).mp (finite_level_fullSupport I eR eM hP) _)
    bn ρR hx s frob good hdense P hρP occurs attached classicality

end Realization

end PaperXII.OrdinaryCompletion
```

The five-gate modules derive finite-level support from the actual patched quotient/control data. A general comparison can use an actual inclusion of support subsets under a ring map, with ordinary or derived support as appropriate. The compatible Hecke point y satisfying $y \circ f = x$ is separately supplied. Support preservation alone does not provide such a point over the prescribed coefficient field: $\mathbb{Q} \to \mathbb{Q}(i)$ preserves full support but admits no extension of the identity point to $\mathbb{Q}$.

The chosen nonzero tensor vector or cohomology vector carries all prescribed Hecke eigenvalues. Classicality is applied to this actual eigensystem, and supplies an occurring automorphic object plus its attached representation. The representation comparison follows under the common-polynomial, continuity, density and semisimplification hypotheses. Propagation uses two already existing compatible families, the family attached to the same automorphic object, distinguished member identities, and their common algebraic spine.

Gate	Input and formal consequence
Residual	Arithmetic residual occupancy and representability supply the deformation problem, its ring and universal representation.
Patching	An applicable arithmetic theorem supplies patched carriers, control maps and full patched support. Lean descends support to the actual finite-level carrier.
Comparison	A support-preserving comparison and compatible Hecke point are supplied. Lean transports support and constructs a nonzero specialized eigenvector.
Fiber	Ordinary tensor or derived cohomology is constructed. The actual Hecke action becomes the specialized scalar character.
Realization	Applicability of classicality to the selected eigensystem, occurrence, attachment, common polynomials, density, semisimplification and Brauer–Nesbitt are supplied. Lean deduces the intertwiner.

13 External arithmetic inputs and editorial boundaries

The formal code begins after the relevant arithmetic objects have been supplied. Retained external inputs include representability and local quotient existence; universal arithmetic deformation representations; residual automorphic occupancy; applicability of a named patching package; classicality at the selected eigensystem; an occurring automorphic object and continuous semisimple attached representation; matching Frobenius normalizations; already existing attached compatible families; arithmetic Frobenius density; construction of semisimplification; and general original-field Brauer–Nesbitt. The canonical Artin localization decomposition and the adequacy convention are also supplied inputs.

An application must state a finite ramification set and coherent choices of Frobenius convention and polynomial normalization. The literal abstract Lean group is Γ. Its arithmetic realization as $G_{F,S}$, and density of the corresponding permitted conjugacy classes, are provided by the application. Chebotarev for finite quotients followed by continuity is the relevant density mechanism for an arbitrary declared

Hausdorff coefficient topology. Claims of historical novelty or citation support are not certified by the kernel.

The final traditional manuscript clarifies the statement context without changing production Lean: Claim 3.3 exposes the local structural map and the complete Hausdorff local coefficient-algebra point category; the proof of 6.4 treats $r > n$ explicitly; and Definition 12.1 exposes the same Frobenius polynomials and all realization hypotheses needed by 9.4. Compatible carrier actions and the scope of companion obstruction at the declared levels are also explicit, as are the finite ramification set and Frobenius convention. These are final editorial clarifications, tracked separately from the eight inherited amendments. They are not fresh kernel certification of a newly edited proposition.

The strongest global conclusion is conditional automorphic realization inside an explicitly declared supported and realized class, followed by propagation inside already existing compatible systems. Universal residual companions, existence of arbitrary compatible families, unrestricted reciprocity and an arithmetic theorem at coefficient prime two are not conclusions of these sources.

A Full source catalog

The complete 96-module production catalog follows in registered dependency order. “Thm.” counts each module’s public theorem declarations, including reused declarations where present. SHA prefixes aid visual comparison. The exact 64-character hashes and imports are supplied in `data/source_catalog.json` and `data/source_catalog.csv`. Source paths are determined by replacing dots in the module name with slashes and adding `.lean`, relative to `formal_release/lean/`.

Module	Lines	Thm.	Source SHA-256 (prefix)
`PaperX.Reconstruction`	189	12	`d8de70c175635ea47d8eaed4`
`PaperXII.ArtinData`	97	9	`953b639b0309f883d326d0f8`
`PaperXII.AuxiliaryGroupAlgebra`	72	7	`ef811535231880a3b7aeb984`
`PaperXII.CompletionQuotient`	181	19	`c7b60b3b5a59cf73ce1891b7`
`PaperXII.ComplexBaseChange`	86	3	`6d91bbc8a4849ac4a5bae4f9`
`PaperXII.DenseFrobenius`	113	8	`941465b7bae2df984810d17a`
`PaperXII.DeterminantalIdeals`	131	11	`1091c57b8583977120bbe9d2`
`PaperXII.DyadicProjectors`	195	17	`a2d01c1ddf2152e2b6bd0269`
`PaperXII.FiniteCarrier`	67	7	`ccc9770f4c6cc72b83a94f0a`
`PaperXII.FiniteGroupCohomology`	107	9	`abab8fb127de1bbc804296e2`
`PaperXII.FiniteTensorQuotient`	172	15	`1ede310ac1f69aeeab66b851`
`PaperXII.LocalIdempotents`	18	1	`e34c194a98e51680b23542d6`
`PaperXII.LocalQuotients`	126	11	`738ec11826cea7e7aabccb35`
`PaperXII.ModuleSupport`	63	8	`b724c27d7ea2d3f2a5941ffe`
`PaperXII.PacketTrace`	74	8	`ad4c7192632b5075026bb73c`
`PaperXII.RepresentationData`	86	6	`6f938f551619e48f1f5a94ae`
`PaperXII.ReuseCompactness`	175	12	`0bd9169cb2b0efc4938e70d7`
`PaperXII.ReuseFiniteSeparation`	97	6	`1975b5fb629823eb7f80c17c`
`PaperXII.S3Standard`	221	22	`c483d46bda60362ef0650d78`
`PaperX.RepresentationLimits`	193	15	`81fb37746e7470557deced36`
`PaperXII.AdicModel`	73	6	`61c8a69111fc54671eb20b99`
`PaperXII.ArtinModules`	206	15	`78d81338b8f8f57638051245`
`PaperXII.AugmentationCoinvariants`	120	8	`9bbc93c64a5beccbe8fbc440`
`PaperXII.CompanionForests`	84	10	`db2d885e73099916d65bcaf0`
`PaperXII.ComplexFieldFibers`	108	9	`cf9b3255fcadbb9d1807f957`
`PaperXII.ComponentSupport`	21	1	`d6a2a42f8834ba233d112609`
`PaperXII.DeterminantalStabilization`	130	10	`7ee60aa621c1a1b775918634`
`PaperXII.ExistingCompatibleSystems`	81	3	`ed9748c8de4dc8693af79eb9`
`PaperXII.FieldFibers`	73	6	`36d276ab57e7cc632e15571f`
`PaperXII.FrobeniusSeparation`	76	7	`009a953c292806afb770c631`
`PaperXII.GaloisRealization`	98	5	`19e69eedbf482775e718e525`
`PaperXII.GeneralizedPackets`	137	7	`4a67970524a245b0414d0b7c`
`PaperXII.NonabelianResidual`	142	8	`923412037be8c47c1017e742`
`PaperXII.PerfectModels`	122	2	`fa7dd21c8776d96b8ab917cc`
`PaperXII.PowerSeriesComponent`	246	38	`cd0599c3aee9d2397d505928`
`PaperXII.QuotientSupport`	60	5	`733d40c9b25bbee513955de0`
`PaperXII.ResidualAugmentation`	78	6	`8ceac287f565bf81c0267dae`

Module	Lines	Thm.	Source SHA-256 (prefix)
PaperXII.S3Adjoint	240	16	c5cee6cdd374926c9897e0a6
PaperXII.S3Detection	316	26	365b36245a4c4aa11c4522a4
PaperXII.S3Spectrum	86	8	ca1cdfc3a4797bc71ee146fa
PaperXII.StableCores	92	6	e266cbe63a32d7c4d3b719a3
PaperX.RepresentationClasses	132	16	a35e3a56c160992d4883ced0
PaperX.RepresentationTopology	124	8	9952178e10c02b66f27949cd
PaperXII.AdicQuotientTopology	105	10	c3ffb8523760a84befb64bb0
PaperXII.BalancedAugmentation	128	9	dd128eb806a101cebe777f9a
PaperXII.BaseChangeSupport	47	4	0da87b182a1e79db32ef36c5
PaperXII.ComplexSupport	103	7	186a3d1126dfa9296836b763
PaperXII.DeterminantalBlockStabiliza tion	83	8	f6657438b1c2b5fd31012d09
PaperXII.DyadicProjectorsResidual	124	13	757fa5f1c294698c32648da8
PaperXII.FiniteAdequacy	109	6	aa7fba126405fa95fcb12944
PaperXII.InverseLimitRepresentations	169	9	ba425975d3baba88d65b5316
PaperXII.OrdinaryPatching	98	5	5724b6d1587cb4c657d25380
PaperXII.SupportTransport	59	4	717228feacde958e8dcd7638
PaperXII.TruncatedGroupAlgebra	297	25	676626c8a833d93839711162
PaperXII.VectorInverseLimit	122	10	ae0a5eff607a14a7e93fd523
PaperX.ClassInvariants	103	12	0506bcb396c1d8cfce4dfbed
PaperXII.AdicCompletionComparison	210	14	2dd1070d880040a9990f96b3
PaperXII.AdicMatrixGroups	163	8	21d738f44b4c35e4778528e0
PaperXII.CoinvariantTensorComparison	108	4	e920619c2a26f8b003123021
PaperXII.CompletedFiniteTensor	134	7	9c4e04f241b588d50db1d905
PaperXII.CompletedGlobalBaseChange	263	18	11fc86e509581906275b565e
PaperXII.CorrectedOccupancy	24	1	efbbb72e4f864a073569846e
PaperXII.HeckeFibers	90	7	3a0e3363a03013bd87c2f504
PaperXII.PerfectDerivedSupport	119	9	f0e29718343af1845da6b39b
PaperXII.PresentationComparison	122	4	f64299848f002c18396ce9bd
PaperXII.RightAugmentation	91	3	5fdf4abbf1666a8e03ab4cd9
PaperXII.AdicCompanions	134	7	b1f7e850bd067860825671d1
PaperXII.Coinvariants	47	2	7500fa7b2277651c91f93365
PaperXII.CompletedBaseChangeTransport	113	7	332e73f093c5fefc16ad077d
PaperXII.DerivedHeckeAction	116	7	b1cb976fcc30917465dca404
PaperXII.DerivedPatching	92	8	853d09528413ec978e2ea106
PaperXII.FiniteTensorLocality	149	11	be33c445a4329462b7039912
PaperXII.OrdinaryCompletion	213	9	07087a7da30fd656baed57b3
PaperXII.PerfectCohomology	108	10	2c560e3c14cc3082ffeb9ef4
PaperXII.PresentationIndependence	63	2	88a4c15e4ba774f041eeb364
PaperXII.AdicSpines	112	6	f90d8f75e4106eacda34b776
PaperXII.CompletedGlobalQuotient	173	17	71eede618e5c6aa481d5ec6b
PaperXII.CompletedTensorUniversal	99	7	92a58d870df81ff11bdc3f97
PaperXII.DerivedRealization	177	7	0316cfac39230ec302b3ae8b
PaperXII.HeckeCarriers	237	16	dd3bdac5d81c73a6d28bb62b
PaperXII.IntrinsicFitting	69	4	33c122a7c407120e6bda7a36
PaperXII.CompletedLocalConditions	100	3	7f7f43315592150c1f9b0e48
PaperXII.ContinuousClasses	112	10	45111ecf0dc7bd82f36a468d
PaperXII.FittingBaseChange	60	3	5d693d829097c7810a8c3ef6
PaperXII.PatchingCarriers	109	5	3c8506f8babeddb508dbeb92
PaperXII.ClassCompanions	197	15	9b177812cf70d0fc68998853
PaperXII.CompletedPaperCapstone	69	3	2489bf5ea48d25a39a90ea28

Module	Lines	Thm.	Source SHA-256 (prefix)
`PaperXII.FittingTransport`	60	3	`23d49ac97ae28abf066ebd10`
`PaperXII.FiveGateCompletion`	168	5	`f107470686ed79b1cf37952c`
`PaperXII.ClassSpines`	116	4	`00aed3a592da86f63b02a21a`
`PaperXII.CompletedPointCategory`	101	5	`1e1fe94ece8611b4b6496b82`
`PaperXII.FittingBoundaryChecks`	66	6	`8516575908e9f9351e9f4666`
`PaperXII.AdicLocalConditions`	116	8	`09cce645d5f4f86c36db909a`
`PaperXII.CompletionBoundaryChecks`	45	4	`ed6c4701f20500afe72ba1c8`
`PaperXII.FittingStrata`	64	5	`a5313b7c1341dbf6bd573598`
`PaperXII.ConditionalCompletion`	54	1	`cc5234bb8a3172352e4b8e70`

B All 49 revised internal claims

The crosswalk lists every revised internal claim, its manuscript label, amendment status, original status and number of mapped targets. Every target name and source module is supplied in `data/claim_crosswalk.json`. The condensed module crosswalk is `data/claim_crosswalk.csv`. “Amended” means complete for the revised statement while the original statement retains its own status. Ordinary editorial clarifications described in Section 13 are separately tracked.

Claim	Manuscript label	Revised	Original	Targets
3.1	`thm:tensor_quotient`	complete	complete	17
3.2	`cor:simultaneous_local`	complete	complete	3
3.3	`prop:global_local_basechange`	complete	complete	13
3.4	`prop:local_idempotent`	complete	complete	1
4.2	`thm:s3_adequacy`	complete	complete	20
5.1	`prop:group_algebra_transfer`	complete	complete	5
5.2	`thm:residual_augmentation_algebra`	complete	complete	7
5.3	`cor:no_residual_character_splitting`	complete	complete	7
5.4	`thm:coinvariant_recovery`	complete	complete	6
6.2	`thm:finite_module_support`	complete	complete	1
6.3	`thm:fitting_support`	complete	complete	2
6.4	`thm:higher_fitting_strata`	complete	complete	2
6.5	`prop:fitting_base_change`	complete	complete	2
6.7	`thm:near_faithful_full_support`	complete	complete	4
6.8	`cor:every_component_support`	complete	complete	2
6.9	`thm:quotient_support`	complete	complete	3
6.11	`thm:ordinary_support_descent`	complete	complete	4
6.12	`cor:componentwise_occupancy`	amended	unchecked	1
6.16	`thm:derived_quotient_support`	complete	complete	20
6.18	`thm:derived_support_descent`	complete	complete	15
6.19	`cor:derived_degree_survives`	complete	complete	7
7.1	`thm:rt_support_transport`	complete	complete	2
7.2	`cor:rt_full_support`	complete	complete	1
7.3	`prop:quotient_erases_component`	complete	complete	10
7.4	`thm:field_fiber_criterion`	complete	complete	2
7.5	`cor:supported_hecke_fiber`	complete	complete	1
8.2	`thm:canonical_hecke_packets`	complete	complete	11
8.3	`cor:simultaneous_generalized_eigenspaces`	complete	complete	6

Claim	Manuscript label	Revised	Original	Targets
8.4	cor:rank_trace_detection	complete	complete	6
8.7	thm:nested_finite_carrier	complete	complete	1
8.8	cor:finite_witness_incompatibility	complete	complete	1
8.10	lem:stable_core_surjectivity	complete	complete	1
8.11	thm:nonzero_stable_core	complete	complete	2
8.12	cor:surjective_inverse_system	complete	complete	1
9.3	thm:support_gives_hecke_system	complete	complete	2
9.4	thm:pointwise_galois_realization	amended	partial	7
9.5	cor:full_support_pointwise_automorphy	amended	partial	1
9.6	thm:derived_pointwise_realization	amended	partial	21
10.3	thm:companion_branch_criterion	complete	complete	2
10.4	cor:empty_companion_obstruction	complete	complete	2
10.5	thm:simultaneous_companion_completion	amended	partial	1
10.7	thm:one_member_propagation	complete	complete	2
10.9	prop:finite_frobenius_separator	complete	complete	2
11.1	prop:dyadic_sign_projectors	complete	complete	14
11.2	cor:dyadic_coinvariant_recovery	complete	complete	5
12.2	thm:ordinary_conditional_completion	amended	partial	7
12.4	thm:derived_conditional_completion	amended	partial	14
12.5	thm:system_completion	complete	complete	1
12.6	thm:five_gate_completion	amended	partial	36

C A practical reading and inspection sequence

A reader examining a mathematical claim should first locate its revised number in the crosswalk, then read the exact production declarations, their ambient variables and the relevant proof. Follow internal imports to any structure or construction the theorem consumes. Determine which fields are supplied data and which declarations construct the desired maps, objects or identities. Match the point category, coefficient maps, topologies, completeness, finite-generation and locality assumptions with the intended application. Consult the original ledger when comparing against earlier wording.

For Claims 3.1 and 3.3, the central source chain is the raw tensor quotient, completion/closed quotient comparison, adic quotient topology, completed finite tensor, algebraic adic comparison/locality, global tensor transport, local condition factorization, Noetherian/local capstone and coefficient/local point category. Checking only ordinary tensor quotients or an assumed interface for the universal property would miss the constructed completed objects and their actual topologies.

For the arithmetic conclusion, trace the nonzero fiber or cohomology vector through the actual carrier and point. Verify classicality for that selected eigensystem, then check attachment, matching Frobenius polynomials, density, semisimplification and representation comparison over the original field. Finally ensure any propagated family already exists and belongs to the same attached automorphic object. This sequence explains exactly what the formalization establishes and what an arithmetic application must still provide.

Glossary

Definitions and reading references for the recurring language of the twelve chapters. Local meanings and hypotheses remain those of the cited mathematical treatment.

BROWSE ALPHABETICALLY

A	2648	B	2648	C	2648
D	2650	E	2651	F	2651
G	2652	H	2652	K	2653
L	2654	M	2654	N	2655
O	2655	P	2656	Q	2658
R	2659	S	2661	T	2663
V	2664	W	2664	Z	2664

A

Automorphic Hecke datum

A Hecke algebra with a finite automorphic module or perfect complex, declared unramified Hecke polynomials, and a characteristic zero locus where classicality and an attached Galois representation are supplied. These realization inputs are part of the datum.

Chapter XII, Definition 9.1

See also: Hecke packet, Residual automorphic occupancy

B

Block diagonal plus low rank operator

An operator represented as a block diagonal part plus one thin factor times the adjoint of another. This separates independent packet work from a coupling term of rank at most the factor width. The abbreviation is BDLR.

Chapter VIII, Definition 3.3

See also: Packet operator representation, Output contract

Bridge datum

In the restricted Galois to type comparison, a supplied symplectic identification of two envelopes, compatible stabilizer action, and a common central character. At the prime two, compatible quadratic refinements and a lift to the central extensions are also required.

Chapter XI, Definition 11.1

See also: Symplectic envelope, Quadratic refinement

C

Clifford cocycle class

The cohomology class of the scalar factors by which multiplication of projective multiplicity operators differs from the group law. In the one break setting, its commutator pairing determines the associated symplectic envelope.

Chapter XI, Definition 9.3

See also: Symplectic envelope, Bridge datum

Common arithmetic spine

Algebraic Frobenius characteristic polynomials, coefficient-prime independent weight and determinant data, and prescribed ramified local labels that companion objects must share. Specifying the spine imposes compatibility conditions. It does not ensure that the corresponding residual fibers are occupied.

Chapter X, Definition 8.1

See also: Residual root, Residual forest, Automorphic Hecke datum

Compatibility

Agreement of specified constructions along two routes. For fine and coarse nonlinear models, applying the fine map and then coarsening must equal coarsening first and applying the coarse map. Arithmetic compatibility uses the separately declared common data.

Chapter III, Definition 20.1

See also: Refinement defect, Common arithmetic spine

Complete multiplicative filtration

A decreasing filtration by two-sided ideals whose products add filtration depths, whose intersection is zero, and whose compatible quotient system recovers the algebra. Positive transfer weights then control convergent expansions in the filtration topology.

Chapter II, Definition 5.1

See also: Weighted transfer series, Lift tree

Complete packet pruning

A set of nodes in a finite refinement tree meeting every root-to-terminal path exactly once, with no selected node above another. Parent projectors split into child projectors, so the selected packets still sum to the identity.

Chapter V, Definition 11.1

See also: Refinement, Packet system

Conditional expectation projector

For a fixed finite probability space and partition, the operator replacing a function by its probability-weighted mean on each partition block. It is an orthogonal projector in the corresponding weighted Hilbert space. Nested partitions yield compatible projections.

Chapter VI, Section 6

See also: Projector, Packet system

Congruential Euler coefficients

Coefficients defined by the reciprocal of an exponential section with width N and address j. The original degree-indexed coefficients and normalized packet coefficients differ by the factor j!. Both the normalization and the distinction between degree and row matter in congruences.

Chapter IX, Definition 3.1

See also: Packet row, Hurwitz integral series, p-adic precision

Connection

A chosen family of invertible maps between chart coordinates. The minimal connection is normalized at each chart. The stronger reversible connection also inverts transport on reversed edges. Neither definition alone imposes the cocycle law or identifies the connection with canonical coordinate transport.

Chapter III, Definition 5.1

Chapter III, Definition 5.5

See also: Flatness of a connection, Gauge change, Holonomy

Cyclic vector

A single vector whose orbit under the declared unital algebra generates the whole module by finite linear combinations. A nonzero module has seed rank one precisely when it has such a vector. Cyclicity is always relative to the acting algebra.

Chapter I, Proposition 4.2

See also: Seed, Seed rank, Generated hull

D

Derived support

For a perfect complex over a commutative ring, the prime ideals at which the derived tensor product with the residue field is nonzero. This support detects the complex after specialization, rather than relying on one selected cohomology module.

Chapter XII, Definition 6.14

See also: Module support, Ordinary patching datum

E

Exact lumpability

Compatibility between a fine Markov transition and a coarse transition under a declared aggregation map. The one-step intertwining identity makes fine evolution followed by aggregation agree with coarse evolution at every nonnegative integer time.

Chapter VI, Theorem 5.1

See also: Compatibility, Refinement defect, Transfer semigroup

F

Feshbach reduction

Elimination of hidden packets through their resolvent, producing an effective operator on the retained space. The correction records excursions through hidden packets. The exact resolvent identity requires the stated inverses and generally differs from bare compression.

Chapter VII, Theorem I.9.1

See also: Transfer block, Memory kernel

Flatness of a connection

The cocycle condition: transport through any intermediate chart equals the declared direct transport. In the reversible pair setting, flatness is equivalent to trivial triangle holonomy. Flatness alone does not say that the connection is already canonically calibrated.

Chapter III, Definition 5.6

See also: Connection, Holonomy, Gauge change

Forward difference

The difference between a sequence value after a declared step and its original value. Iterating this operation gives higher differences. In the Euler packet results the step acts on packet rows, so ordinary-degree displacement must be converted using the packet width.

Chapter IX, Definition 3.4

See also: Packet row, Congruential Euler coefficients, p-adic precision

G

Gauge change

A change of coordinate representative at each chart by an automorphism. It acts on a connection by conjugating the source and target representatives separately. Cocycle defects transform covariantly, while loop holonomy is conjugated at its base.

Chapter III, Definition 6.1

See also: Connection, Flatness of a connection, Holonomy

Generated hull

All finite sums of seed vectors acted on by elements of the declared algebra. In a topological realization, the topological hull is its closure. Algebraic generation and dense generation are distinct because closure may add limits with no finite addressed expression.

Chapter I, Definition 2.7

See also: Seed, Seed rank, Realization

Global deformation datum

A residual representation, local deformation rings and represented local-condition quotients, a global deformation ring, and compatible maps imposing those local conditions globally. Existence and representability of these objects require their own hypotheses. The datum does not establish automorphic occupancy.

Chapter XII, Definition 2.1

See also: Residual root, Residual automorphic occupancy, Ordinary patching datum

H

Hecke packet

The part of a finite Hecke module selected by a central idempotent of its finite-dimensional commutative Hecke algebra. It is nonzero precisely at an occupied maximal ideal. Over an algebraically closed field it is a simultaneous generalized eigenspace, retaining nilpotent data.

Chapter XII, Theorem 8.2

See also: Module support, Automorphic Hecke datum, Stable core

Holonomy

The composite transport around a loop for a chosen connection, based at its starting chart. Triangle holonomy measures the failure of a reversible pair connection to satisfy its cocycle law. It is not an invariant of an unspecified change of coordinates.

Chapter III, Definition 5.9

See also: Connection, Flatness of a connection, Gauge change

Hurwitz integral series

An exponential generating series whose factorial-basis coefficients are p-adic integers. Products and differentiation preserve this lattice. A series in it with unit constant coefficient has its reciprocal in the same lattice, supporting the reciprocal-transfer congruence arguments.

Chapter IX, Definition 4.1

See also: Congruential Euler coefficients, p-adic precision

K

Kraus channel

A finite sum of maps sending a matrix to K times that matrix times the adjoint of K, with the adjoint products summing to the identity. This normalization makes the resulting completely positive map trace preserving.

Chapter VII, Classical Input I.7.2

See also: Quantum seed system, Packet system

Krylov space

The span of a vector and its successive images under one operator, through a specified finite order. This records how repeated application enlarges the generated space. Packet decompositions interact with it under the invariance hypotheses stated in the relevant result.

Chapter VIII, Definition 7.1

See also: Cyclic vector, Generated hull, Packet operator representation

L

Lift tree

A rooted tree of approximation classes at increasing filtration depths. A child reduces to its parent and satisfies the next depth condition. Nonempty levels and finite branching provide a compatible branch. Completeness, separatedness and continuity are further needed to recover an exact solution.

Chapter III, Definition 23.5 and Theorem 23.6

See also: Complete multiplicative filtration, Obstruction class, Residual forest

M

Memory kernel

The time-dependent operator describing how eliminated packets return influence to retained packets. In the finite linear system it is a transfer into the hidden block, hidden evolution, and transfer back. Its Laplace transform is the corresponding self-energy under the stated spectral condition.

Chapter VI, Section 13.1 and Theorem 13.2

See also: Feshbach reduction, Transfer semigroup, Transfer block

Module support

The prime ideals of a commutative ring at which localization of the module is nonzero. Full support means every prime is occupied in this algebraic sense. It does not by itself provide a classical automorphic realization.

Chapter XII, Definition 6.1

See also: Near faithfulness, Derived support, Hecke packet

Monodromy packet chain

A finite sequence of packet spaces joined by a nilpotent operator that moves each packet down one level. A compatible Frobenius operator scales this transfer by the declared residue-field factor. Its powers are sums of paths along the chain.

Chapter XI, Definition 2.2 and Proposition 2.3

See also: Transfer block, Transfer graph

N

Near faithfulness

For a finite module over a Noetherian ring, nilpotence of its annihilator ideal. This is equivalent to full module support. If the ring is reduced it is also equivalent to faithfulness. The finiteness and ring hypotheses are part of the equivalence.

Chapter XII, Theorem 6.7

See also: Module support, Ordinary patching datum

O

Obstruction class

The class of a defect in the cokernel of a declared linear correction map. It vanishes exactly when that linear correction can remove the defect. The quotient, not the size of the defect, determines this solvability question.

Chapter III, Definition 21.1 and Theorem 21.2

See also: Refinement defect, Obstruction packet, Lift tree

Obstruction packet

A named packet with a certificate that the residual of one specified candidate is nonzero. It rules out that candidate under the declared solution condition. It neither constructs a solution nor proves that every possible candidate is obstructed.

Chapter III, Definition 7.1 and Theorem 7.2

See also: Obstruction class, Residual, Packet

Ordinary patching datum

A Noetherian patched ring, an ideal, and a finite patched module, together with identifications of the finite-level ring and module as their quotients. Full patched support descends under these hypotheses. Automorphy requires additional arithmetic data.

Chapter XII, Definition 6.10

See also: Module support, Near faithfulness, Global deformation datum

Output contract

The mathematical object an algorithm is required to return, such as an explicit vector, matrix, factors, or an implicit operator with an application rule. Costs are comparable only after the represented object and required output are fixed.

Chapter VIII, Definition 2.7

See also: Packet operator representation, Block diagonal plus low rank operator

P

p-adic precision

A lower bound on the p-adic valuation in the declared integral coefficient ring. The Euler congruence statements assign infinite valuation to zero. Divisibility in the field alone does not express precision, and special finite conventions for a zero step are stated separately.

Chapter IX, Definition 3.4

See also: Congruential Euler coefficients, Forward difference, Hurwitz integral series

Packet

In the linear theory, a component selected by a declared projector, usually identified with its range and index. Analytic, signal and arithmetic realizations attach their own additional structure. The meaning of a packet is therefore read with its specified system.

Chapter I, Definition 7.1

See also: Packet system, Projector, Transfer block

Packet analysis and synthesis

Analysis sends an object to its tuple of projected packet components. Synthesis adds those components in the ambient space. For a finite complete packet system the maps are inverse isomorphisms. Infinite versions require a specified convergence mode and corresponding bounds.

Chapter I, Definition 7.4 and Theorem 7.5

See also: Packet system, Packet support, Refinement

Packet jet

A higher derivative restricted to specified input packet spaces and projected to a specified output packet. Summing over all addresses reconstructs the derivative exactly. The analytic statements require the stated bounded projectors and continuous Fréchet differentiability.

Chapter III, Definition 17.1

See also: Packet, Transfer block

Packet operator representation

A finite packet layout, a declared set of active input-output addresses, and a linear block at each address. Evaluation extends each local block to the ambient direct sum and adds the blocks. Thus the representation has an explicit operator meaning.

Chapter VIII, Definition 2.2

See also: Transfer block, Output contract, Block diagonal plus low rank operator

Packet row

The index n within a fixed coefficient packet of width N. For the reciprocal Euler family, row n corresponds to ordinary degree Nn. A step of u rows therefore changes ordinary degree by Nu, a distinction needed when stating periods and difference estimates.

Chapter IX, Section 4, equations (4.1)-(4.2)

See also: Congruential Euler coefficients, Forward difference

Packet support

The set of packet indices on which an object has a nonzero projected component. A packet-pure vector has at most one such index, including the zero vector. This is different from module support in a prime spectrum.

Chapter I, Definition 7.2

See also: Packet, Seed rank, Module support

Packet system

A finite family of idempotent operators that annihilate one another and sum to the identity, giving an exact direct-sum decomposition into their ranges. Orthogonality is an additional condition when stated. Some chapters remove zero packet spaces by convention.

Chapter I, Definition 7.1

See also: Projector, Packet analysis and synthesis, Refinement

Perfect reconstruction

Recovery of every input by composing a filter bank with its synthesis bank, with any permitted gain and delay declared explicitly. The criterion is an identity of Laurent-polynomial matrices. A delay in decimated channel indices differs from a delay in original sample addresses.

Chapter V, Theorem 4.1

See also: Polyphase channel, Packet analysis and synthesis

Polyphase channel

The decimated sequence obtained by retaining one residue class of sample addresses and reindexing it on a coarse lattice. Its masked counterpart remains on the original lattice with other samples set to zero. Interleaving reconstructs the original sequence from all channels.

Chapter V, Section 2 and Theorem 2.1

See also: Residue selector, Perfect reconstruction

Projector

An idempotent linear operator: applying it twice has the same effect as applying it once. Its range is the selected component. The general packet calculus does not require an orthogonal projector, although Hilbert space and quantum results may impose self-adjointness.

Chapter I, Definition 7.1

See also: Packet system, Packet

Q

Quadratic refinement

Over the field with two elements, a quadratic function whose polar difference recovers a specified symplectic form. The alternating form alone does not determine the refinement. This additional datum controls dyadic phase information in the arithmetic packet constructions.

Chapter XI, Definition 13.1

See also: Symplectic envelope, Bridge datum

Quantum seed system

A finite Hilbert space with a nonzero seed, a declared operator grammar, constraints, packet systems at refinement levels, and the relevant dynamical and topological data. Its generated span is algebraic. Reachability by a restricted physical gate set is a separate question.

Chapter VII, Definition I.1.1

See also: Seed, Cyclic vector, Kraus channel

R

Realization

An assignment of spaces and linear maps to a typed operator grammar that satisfies its relations. More generally, a domain realization interprets specified abstract seed constructions in a chosen mathematical setting. Which operations and universal properties survive must be declared.

Chapter I, Definition 2.1

See also: Typed operator grammar, Realization functor, Seed

Realization functor

A functor from abstract seed systems to a declared domain category, equipped with a preservation profile. The profile states which identities or universal properties it preserves, such as sums, compositions, packet idempotents, quotients, or completions.

Chapter I, Definition 11.1

See also: Realization, Typed operator grammar

Reduced denominator

The one-variable entire function obtained from the signed exponential denominator by using a coordinate w equal to the Nth power of z. A root in the reduced coordinate records a root-of-unity orbit in the original variable, with multiplicities retained.

Chapter IV, Definition 5.7

See also: Residue selector, Transfer tangent, Zero power sum

Refinement

A finer packet system whose projectors sum to each coarse projector according to a surjective parent map. Fine packet ranges lie in their parent ranges. Exact refinement regroups complete component vectors and need not discard information.

Chapter II, Definition 3.2

See also: Packet system, Packet analysis and synthesis, Complete packet pruning

Refinement defect

The difference between applying a fine model and then coarsening, and coarsening first and applying the coarse model. Its vanishing expresses exact compatibility. A correction map must be specified before this vector defines a corresponding obstruction class.

Chapter III, Definition 20.1

See also: Compatibility, Obstruction class, Exact lumpability

Residual

In the candidate-certificate language, the value of a declared test at a candidate and packet. A solution makes every such value zero. This use of residual as a test value is distinct from reduction to a residue field in arithmetic deformation problems.

Chapter III, Section 7.1 and Definition 7.1

See also: Obstruction packet, Residual root

Residual automorphic occupancy

Existence of a nonzero localized automorphic or cohomological Hecke carrier realizing a specified residual representation and local conditions. It is an input at residual depth. Positive-depth deformations over another residual root do not supply it.

Chapter XII, Definition 2.4

See also: Residual root, Automorphic Hecke datum, Global deformation datum

Residual forest

The family of formal deformation problems above the admissible residual roots. Higher-depth objects are related by reduction, so each root has its own lifting branches. Declaring the forest does not establish that every intended branch or characteristic zero realization exists.

Chapter X, Definition 1.1

See also: Residual root, Lift tree, Common arithmetic spine

Residual root

An object over the residue field, fixed as the reduction of all objects in its residual fiber. Positive-depth corrections preserve this root. A universal arithmetic seed system therefore records all admissible roots rather than assuming one root represents every residual class.

Chapter X, Definition 1.1

See also: Residual forest, Residual automorphic occupancy, Visibility weight

Residue selector

The coefficient projector retaining exactly the exponents in one congruence class modulo a positive integer. It is defined directly by a coefficient mask and needs no division. A realization as a roots-of-unity average requires additional orthogonality hypotheses.

Chapter II, Definition 4.1

See also: Roots of unity filter, Polyphase channel, Refinement

Roots of unity filter

A finite character average realizing a residue selector when the required Fourier orthogonality certificate holds. Over a field with a primitive root, the usual normalization divides by the width. Over more general coefficient rings, the coefficient selector remains meaningful even when that average is unavailable.

Chapter II, Theorem 4.3

See also: Residue selector, Separation weight

S

Seed

A distinguished starting element acted on by a declared operator algebra or grammar. A seed family generates its hull through addressed operator expressions and finite sums, with closure added only in a specified topological regime. The acting algebra is fixed independently of the target.

Chapter I, Definition 2.6

See also: Generated hull, Seed rank, Typed operator grammar

Seed rank

The least size of a family generating a module under the declared acting algebra. In the topological version, generation is replaced by dense generation. The zero module has rank zero. Changing the acting algebra can change the seed rank of the same underlying space.

Chapter I, Definition 4.1

See also: Seed, Generated hull, Cyclic vector

Seed zeta function

The formal power series whose coefficient at each positive depth counts the finite objects above one residual root at that depth. Smooth and weighted charts yield explicit formulas under their stated assumptions. It records branching counts, not just the existence of a lift.

Chapter X, Definition 6.1

See also: Residual forest, Visibility weight

Separation weight

The least exponent of a uniformizer that makes a rational projector preserve a specified integral lattice, provided such an exponent exists. It records the precision needed to clear that projector on that lattice and is distinct from the visibility depth of two coordinates.

Chapter II, Definition 7.6

See also: Projector, Visibility weight, Roots of unity filter

Singular packet

In the signal chapter, a step or truncated-power component recording a value jump or a derivative jump at a declared breakpoint. Removing the trace jumps leaves the regular remainder covered by the stated Sobolev assumptions. Representation alone does not solve data-driven recovery.

Chapter V, Section E7

See also: Packet, Perfect reconstruction

Stable core

At one level of an inverse system of finite-dimensional spaces, the intersection of the images of every later level. It retains vectors that remain liftable arbitrarily far. Nonzero stable cores at every level provide compatible nonzero inverse-limit vectors, unlike mere nonvanishing of unrelated levels.

Chapter XII, Definition 8.9 and Theorem 8.11

See also: Hecke packet, Lift tree

Symplectic envelope

For a one break Clifford datum, the quotient of the finite elementary abelian group by the radical of its intrinsic commutator pairing. This removes the degenerate directions of that pairing. Comparison with a type-side envelope requires the additional bridge data.

Chapter XI, Definition 9.5

See also: Clifford cocycle class, Bridge datum, Quadratic refinement

T

Transfer block

The part of an operator carrying one input packet to one output packet: first select the input, apply the operator, and select the output. Summing all blocks reconstructs the operator for a finite complete packet system.

Chapter I, Definition 8.1

See also: Packet system, Transfer graph, Packet operator representation

Transfer graph

A directed graph on packet indices, with an edge where a declared algebra generator has a nonzero transfer block. Reachability limits where generated operators may have nonzero blocks. An allowed path does not prevent cancellation or guarantee a nonzero transfer.

Chapter I, Definition 9.1

See also: Transfer block, Seed rank, Packet operator representation

Transfer semigroup

A family of evolution operators whose composition adds time and whose value at zero is the identity. In the finite setting it is generated by a matrix exponential. The sign convention differs between generators of evolution and nonnegative diffusion operators.

Chapter VI, Definition 1.1 and Theorem 1.2

See also: Memory kernel, Exact lumpability

Transfer tangent

The negative logarithmic derivative of the signed exponential denominator in the cyclic analytic family. It is also the logarithmic derivative of the first reciprocal power. Its expansion connects reciprocal coefficient recurrences to logarithmic coefficients and reduced zero power sums.

Chapter IV, Definition 8.1

See also: Reduced denominator, Zero power sum

Typed operator grammar

A quiver of permitted source and target types together with linear relations among paths having the same endpoints. Its quotient path algebra encodes composition and relations. A realization assigns spaces to vertices and maps to arrows satisfying those relations.

Chapter I, Definition 2.1

See also: Realization, Seed, Transfer graph

V

Visibility weight

The first precision depth at which two distinct integral packet coordinates have different reductions, when such a depth exists. At the prime two, signs may share a residual reduction yet separate at a later depth. This weight measures separation of coordinates, not a projector denominator.

Chapter X, Definition 5.1

See also: Separation weight, Residual root, Seed zeta function

W

Weighted transfer series

A generating series for walks in a finite directed multigraph with positive integer edge weights and scalar multiplicities. It sums multiplicity products and endpoint weights at each total weight, including the empty walk. The positive weights make each coefficient well defined.

Chapter II, Definition 6.1

See also: Transfer graph, Complete multiplicative filtration

Z

Zero power sum

A sum of inverse powers of the reduced zeros of the cyclic analytic denominator, counted with multiplicity. For section order N at least two, these sums converge absolutely and agree with the normalized logarithmic coefficients through the genus-zero product.

Chapter IV, Definition 12.5 and Theorem 12.6

See also: Reduced denominator, Transfer tangent

Sources and references

Sources are grouped by chapter and treatment. Citation labels retain the numbering or short names used in the text. The accompanying files contain the editable material and supporting source records.

BY CHAPTER AND TREATMENT

Chapter I mathematical text 2666

Chapter II mathematical text 2668

Chapter III mathematical text 2670

Chapter IV mathematical text 2672

Chapter V mathematical text 2675

Chapter VI mathematical text 2678

Chapter VII mathematical text 2681

Chapter VIII mathematical text 2684

Chapter IX mathematical text 2687

Chapter X mathematical text 2690

Chapter XI mathematical text 2693

Arithmetic realization studies: direct construction 2695

Arithmetic realization studies: monomial routes and Clifford descent 2698

Arithmetic realization studies: conductors and native markings 2700

Chapter XII mathematical text 2702

Chapter I mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2667

Return to Chapter I

References

[1] I. Assem, D. Simson, and A. Skowroński, *Elements of the Representation Theory of Associative Algebras, Volume 1: Techniques of Representation Theory*, London Mathematical Society Student Texts 65, Cambridge University Press, Cambridge, 2006. doi:10.1017/CBO9780511614309.

[2] A. E. Brouwer and W. H. Haemers, *Spectra of Graphs*, Universitext, Springer, New York, 2012. doi:10.1007/978-1-4614-1939-6.

[3] J. B. Conway, *A Course in Functional Analysis*, second edition, Graduate Texts in Mathematics 96, Springer, New York, 1990. doi:10.1007/978-1-4757-4383-8.

[4] R. A. Horn and C. R. Johnson, *Matrix Analysis*, second edition, Cambridge University Press, Cambridge, 2012. doi:10.1017/CBO9781139020411.

[5] T. Y. Lam, *A First Course in Noncommutative Rings*, second edition, Graduate Texts in Mathematics 131, Springer, New York, 2001. doi:10.1007/978-1-4419-8616-0.

[6] T. Y. Lam, *Lectures on Modules and Rings*, Graduate Texts in Mathematics 189, Springer, New York, 1999. doi:10.1007/978-1-4612-0525-8.

[7] S. Roman, *Advanced Linear Algebra*, third edition, Graduate Texts in Mathematics 135, Springer, New York, 2008. doi:10.1007/978-0-387-72831-5.

[8] H. H. Schaefer and M. P. Wolff, *Topological Vector Spaces*, second edition, Graduate Texts in Mathematics 3, Springer, New York, 1999. doi:10.1007/978-1-4612-1468-7.

[9] J. H. Silverman, *The Arithmetic of Elliptic Curves*, second edition, Graduate Texts in Mathematics 106, Springer, New York, 2009. doi:10.1007/978-0-387-09494-6.

[10] L. de Moura and S. Ullrich, *The Lean 4 Theorem Prover and Programming Language*, in Automated Deduction—CADE 28, Lecture Notes in Computer Science 12699, Springer, Cham, 2021, pp. 625–635. doi:10.1007/978-3-030-79876-5_37.

[11] The mathlib Community, *The Lean Mathematical Library*, in Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, ACM, 2020, pp. 367–381. doi:10.1145/3372885.3373824.

Chapter II mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2669

Return to Chapter II

References

[1] John B. Conway. *A Course in Functional Analysis*. Springer, second edition, 1990.

[2] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[3] Israel Gohberg, Peter Lancaster, and Leiba Rodman. *Matrix Polynomials*, volume 58 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[4] Tosio Kato. *Perturbation Theory for Linear Operators*. Classics in Mathematics. Springer, Berlin, 1995.

[5] Serge Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer, New York, revised third edition, 2002.

[6] Morris Newman. *Integral Matrices*, volume 45 of *Pure and Applied Mathematics*. Academic Press, New York, 1972.

[7] David M. Pozar. *Microwave Engineering*. Wiley, 4th edition, 2012.

[8] Jean Pierre Serre. *Local Fields*, volume 67 of *Graduate Texts in Mathematics*. Springer, New York, 1979. Translated by Marvin Jay Greenberg.

[9] Richard P. Stanley. *Enumerative Combinatorics, Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012.

[10] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

Chapter III mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2671

Return to Chapter III

References

[1] Ralph Abraham, Jerrold E. Marsden, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer, New York, second edition, 1988.

[2] Gregory M. Constantine and Thomas H. Savits. A multivariate Faà di Bruno formula with applications. *Transactions of the American Mathematical Society*, 348(2):503–520, 1996.

[3] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021.

[4] Klaus Deimling. *Nonlinear Functional Analysis*. Springer, Berlin, 1985.

[5] Jonathan L. Gross and Thomas W. Tucker. *Topological Graph Theory*. John Wiley and Sons, New York, 1987.

[6] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations*. Springer, Berlin, second edition, 2006.

[7] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5:435–439, 1950.

[8] Tosio Kato. *Perturbation Theory for Linear Operators*. Springer, Berlin, 1995.

[9] Hansjörg Kielhöfer. *Bifurcation Theory: An Introduction with Applications to Partial Differential Equations*. Springer, New York, second edition, 2012.

[10] Shoshichi Kobayashi and Katsumi Nomizu. *Foundations of Differential Geometry. Volume I*. John Wiley and Sons, New York, 1963.

[11] Jean Louis Loday and Bruno Vallette. *Algebraic Operads*. Springer, Berlin, 2012.

[12] John McCleary. *A User's Guide to Spectral Sequences*. Cambridge University Press, Cambridge, second edition, 2001.

[13] Komorebi P. Projector-Resolved Seed Calculus I: Typed seeds, operator grammars, and exact reconstruction. Chapter I of this volume.

[14] Komorebi P. Projector-Resolved Seed Calculus II: Polyphase refinement, weighted transfer, and dyadic–odd resolution. Chapter II of this volume.

[15] Roger Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

[16] Richard P. Stanley. *Enumerative Combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012.

[17] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 367–381. Association for Computing Machinery, 2020.

[18] Charles A. Weibel. *An Introduction to Homological Algebra*. Cambridge University Press, Cambridge, 1994.

[19] Eberhard Zeidler. *Nonlinear Functional Analysis and Its Applications. I: Fixed Point Theorems*. Springer, New York, 1986.

Chapter IV mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2673

Return to Chapter IV

References

[AAR99] George E. Andrews, Richard Askey, and Ranjan Roy. *Special Functions*, volume 71 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1999.

[Apo76] Tom M. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer, New York, 1976.

[BM06] Pavel Bleher and Robert Mallison, Jr. Zeros of sections of exponential sums. *International Mathematics Research Notices*, 2006:Article ID 38937, 49 pages, 2006. doi: 10.1155/IMRN/2006/38937.

[Boa54] Ralph P. Boas. *Entire Functions*. Academic Press, New York, 1954.

[Com74] Louis Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. D. Reidel Publishing Company, Dordrecht, 1974.

[dMU21] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, Cambridge, 2009.

[Lan31] Rudolph E. Langer. On the zeros of exponential sums and integrals. *Bulletin of the American Mathematical Society*, 37(4):213–239, 1931.

[Lev80] Boris Ya. Levin. *Distribution of Zeros of Entire Functions*, volume 5 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, revised edition, 1980.

[Mor73] Carlos J. Moreno. The zeros of exponential polynomials. I. *Compositio Mathematica*, 26(1):69–78, 1973.

[Mul05] Martin E. Muldoon. Generalized hyperbolic functions, circulant matrices and functional equations. *Linear Algebra and its Applications*, 406:272–284, 2005. doi: 10.1016/j.laa.2005.04.011.

[P.26a] Komorebi P. Projector-Resolved Seed Calculus I: Typed seeds, operator grammars, and exact reconstruction. Chapter I of this volume.

[P.26b] Komorebi P. Projector-Resolved Seed Calculus II: Polyphase refinement, weighted transfer, and dyadic–odd resolution. Chapter II of this volume.

[P.26c] Komorebi P. Projector-resolved seed calculus iii: Typed edge operators, connection defects, and obstruction packets. Chapter III of this volume.

[Pól20] George Pólya. Geometrisches über die verteilung der nullstellen gewisser ganzer transzendenter funktionen. *Sitzungsberichte der Bayerischen Akademie der Wissenschaften, Mathematisch Physikalische Klasse*, 50:285–290, 1920.

[Rai60] Earl D. Rainville. *Special Functions*. The Macmillan Company, New York, 1960.

[Rom84] Steven Roman. *The Umbral Calculus*, volume 111 of *Pure and Applied Mathematics*. Academic Press, New York, 1984.

[Sch92] Jens Schwaiger. On generalized hyperbolic functions and their characterization by functional equations. *Aequationes Mathematicae*, 43:198–210, 1992. doi: 10.1007/BF01835702.

[The20] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 367–381. Association for Computing Machinery, 2020.

[Tit39] Edward C. Titchmarsh. *The Theory of Functions*. Oxford University Press, Oxford, second edition, 1939.

Chapter V mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2676

Return to Chapter V

References

[AFP00] Luigi Ambrosio, Nicola Fusco, and Diego Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press, Oxford, 2000.

[CD04] Emmanuel J. Candès and David L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise smooth singularities. *Communications on Pure and Applied Mathematics*, 57(2):219–266, 2004.

[CDF92] Albert Cohen, Ingrid Daubechies, and Jean Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992.

[CR83] Ronald E. Crochiere and Lawrence R. Rabiner. *Multirate Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[CW92] Ronald R. Coifman and Mladen Victor Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[DeV98] Ronald A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[Don99] David L. Donoho. Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics*, 27(3):859–897, 1999.

[DS98] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998.

[GS97] David Gottlieb and Chi-Wang Shu. On the gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997.

[Mal09] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Burlington, Massachusetts, third edition, 2009.

[MH92] Stéphane Mallat and Wen Liang Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, 1992.

[Nat26] National Institute of Standards and Technology. NIST Digital Library of Mathematical Functions, section 6.16(i): The Gibbs phenomenon. https://dlmf.nist.gov/6.16.i, 2026. Version 1.2.8, released 15 September 2026. Equations 6.16.1–6.16.4. Accessed 20 October 2026.

[Smi69] Phillip H. Smith. *Electronic Applications of the Smith Chart*. McGraw Hill, New York, 1969.

[SN96] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, Wellesley, Massachusetts, 1996.

[Swe96] Wim Sweldens. The lifting scheme: A custom design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

[Vai93] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[VK95] Martin Vetterli and Jelena Kovačević. *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[Zyg02] Antoni Zygmund. *Trigonometric Series*. Cambridge University Press, Cambridge, third edition, 2002.

Chapter VI mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2679

Return to Chapter VI

References

[ANO96] Søren Asmussen, Olle Nerman, and Marita Olsson. Fitting phase type distributions via the expectation maximization algorithm. *Scandinavian Journal of Statistics*, 23(4):419–441, 1996.

[Bal60] A. V. Balakrishnan. Fractional powers of closed operators and the semigroups generated by them. *Pacific Journal of Mathematics*, 10(2):419–437, 1960.

[BS94] José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. John Wiley and Sons, Chichester, 1994.

[CHK00] Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction and the mori zwanzig representation of irreversible processes. *Proceedings of the National Academy of Sciences of the United States of America*, 97(7):2968–2973, 2000.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Hoboken, New Jersey, second edition, 2006.

[DS58] Nelson Dunford and Jacob T. Schwartz. *Linear Operators. Part I: General Theory*, volume 7 of *Pure and Applied Mathematics*. Interscience Publishers, New York, 1958.

[EN00] Klaus Joachim Engel and Rainer Nagel. *One Parameter Semigroups for Linear Evolution Equations*, volume 194 of *Graduate Texts in Mathematics*. Springer, New York, 2000.

[Fes58] Herman Feshbach. Unified theory of nuclear reactions. *Annals of Physics*, 5(4):357–390, 1958.

[Jan97] Svante Janson. *Gaussian Hilbert Spaces*, volume 129 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1997.

[Kac49] Mark Kac. On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):1–13, 1949.

[Kai80] Thomas Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1980.

[KS60] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. D. Van Nostrand, Princeton, New Jersey, 1960.

[Neu81] Marcel F. Neuts. *Matrix Geometric Solutions in Stochastic Models*. The Johns Hopkins University Press, Baltimore, 1981.

[Nor97] J. R. Norris. *Markov Chains*, volume 2 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1997.

[Paz83] Amnon Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*, volume 44 of *Applied Mathematical Sciences*. Springer, New York, 1983.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RP81] L. C. G. Rogers and J. W. Pitman. Markov functions. *The Annals of Probability*, 9(4):573–582, 1981.

[SSV12] René L. Schilling, Renming Song, and Zoran Vondraček. *Bernstein Functions*, volume 37 of *De Gruyter Studies in Mathematics*. De Gruyter, Berlin, second edition, 2012.

[Str04] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics, Philadelphia, second edition, 2004.

[Tay] Michael E. Taylor. Lectures on Banach algebras. Author-hosted lecture notes. Undated; Appendix A. Accessed 6 October 2026.

[Wil91] David Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991.

[Zwa61] Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983–992, 1961.

Chapter VII mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2682

Return to Chapter VII

References

[1] Cédric Bény, Achim Kempf, and David W. Kribs, *Generalization of quantum error correction via the Heisenberg picture*, Physical Review Letters **98** (2007), no. 10, 100502.

[2] Michael V. Berry, *Quantal phase factors accompanying adiabatic changes*, Proceedings of the Royal Society of London. Series A **392** (1984), no. 1802, 45–57.

[3] Rajendra Bhatia, *Matrix analysis*, Graduate Texts in Mathematics, vol. 169, Springer, New York, 1997.

[4] Sergey Bravyi, David P. DiVincenzo, and Daniel Loss, *Schrieffer Wolff transformation for quantum many body systems*, Annals of Physics **326** (2011), no. 10, 2793–2826.

[5] Man Duen Choi, *Completely positive linear maps on complex matrices*, Linear Algebra and its Applications **10** (1975), no. 3, 285–290.

[6] Alain Connes, *Noncommutative geometry*, Academic Press, San Diego, 1994.

[7] Paul A. M. Dirac, *Lectures on quantum mechanics*, Yeshiva University, New York, 1964.

[8] Herman Feshbach, *Unified theory of nuclear reactions*, Annals of Physics **5** (1958), no. 4, 357–390.

[9] William Fulton and Joe Harris, *Representation theory: A first course*, Graduate Texts in Mathematics, vol. 129, Springer, New York, 1991.

[10] James Glimm and Arthur Jaffe, *Quantum physics: A functional integral point of view*, second ed., Springer, New York, 1987.

[11] Marc Henneaux and Claudio Teitelboim, *Quantization of gauge systems*, Princeton University Press, Princeton, New Jersey, 1992.

[12] Tosio Kato, *On the adiabatic theorem of quantum mechanics*, Journal of the Physical Society of Japan **5** (1950), 435–439.

[13] _____, *Perturbation theory for linear operators*, Springer, Berlin, 1995.

[14] Emanuel Knill and Raymond Laflamme, *Theory of quantum error correcting codes*, Physical Review A **55** (1997), no. 2, 900–911.

[15] Karl Kraus, *States, effects, and operations*, Lecture Notes in Physics, vol. 190, Springer, Berlin, 1983.

[16] Goran Lindblad, *On the generators of quantum dynamical semigroups*, Communications in Mathematical Physics **48** (1976), no. 2, 119–130.

[17] Sadao Nakajima, *On quantum theory of transport phenomena: steady diffusion*, Progress of Theoretical Physics **20** (1958), no. 6, 948–959.

[18] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, tenth anniversary ed., Cambridge University Press, Cambridge, 2010.

[19] Konrad Osterwalder and Robert Schrader, *Axioms for Euclidean Green functions*, Communications in Mathematical Physics **31** (1973), no. 2, 83–112.

[20] ———, *Axioms for Euclidean Green functions. II*, Communications in Mathematical Physics **42** (1975), no. 3, 281–305.

[21] Michael Reed and Barry Simon, *Methods of modern mathematical physics. I: Functional analysis*, Academic Press, New York, 1972.

[22] Benjamin Schumacher and Michael A. Nielsen, *Quantum data processing and error correction*, Physical Review A **54** (1996), no. 4, 2629–2635.

[23] Jean Pierre Serre, *Linear representations of finite groups*, Graduate Texts in Mathematics, vol. 42, Springer, New York, 1977.

[24] W. Forrest Stinespring, *Positive functions on C star algebras*, Proceedings of the American Mathematical Society **6** (1955), no. 2, 211–216.

[25] John C. Strikwerda, *Finite difference schemes and partial differential equations*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, 2004.

[26] Dmitri V. Vassilevich, *Heat kernel expansion: user's manual*, Physics Reports **388** (2003), no. 5, 279–360.

[27] John Watrous, *The theory of quantum information*, Cambridge University Press, Cambridge, 2018.

[28] Frank Wilczek and A. Zee, *Appearance of gauge structure in simple dynamical systems*, Physical Review Letters **52** (1984), no. 24, 2111–2114.

[29] Robert Zwanzig, *Ensemble method in the theory of irreversibility*, The Journal of Chemical Physics **33** (1960), no. 5, 1338–1341.

Chapter VIII mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2685

Return to Chapter VIII

References

[1] Sivaram Ambikasaran and Eric Darve, *An o n log n fast direct solver for partial hierarchically semi separable matrices*, Journal of Scientific Computing **57** (2013), no. 3, 477–501.

[2] Grey Ballard, James Demmel, Olga Holtz, and Oded Schwartz, *Minimizing communication in numerical linear algebra*, SIAM Journal on Matrix Analysis and Applications **32** (2011), no. 3, 866–901.

[3] Richard Bellman, *Dynamic programming*, Princeton University Press, Princeton, NJ, 1957.

[4] James R. Bunch and Christopher P. Nielsen, *Updating the singular value decomposition*, Numerische Mathematik **31** (1978), 111–129.

[5] Erin Carson and Nicholas J. Higham, *Accelerating the solution of linear systems by iterative refinement in three precisions*, SIAM Journal on Scientific Computing **40** (2018), no. 2, A817–A847.

[6] Ronald R. Coifman and Mladen Victor Wickerhauser, *Entropy based algorithms for best basis selection*, IEEE Transactions on Information Theory **38** (1992), no. 2, 713–718.

[7] James W. Cooley and John W. Tukey, *An algorithm for the machine calculation of complex fourier series*, Mathematics of Computation **19** (1965), no. 90, 297–301.

[8] Chandler Davis and William M. Kahan, *The rotation of eigenvectors by a perturbation. iii*, SIAM Journal on Numerical Analysis **7** (1970), no. 1, 1–46.

[9] Timothy A. Davis, *Direct methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

[10] James Demmel, Ioana Dumitriu, and Olga Holtz, *Fast linear algebra is stable*, Numerische Mathematik **108** (2007), no. 1, 59–91.

[11] Carl Eckart and Gale Young, *The approximation of one matrix by another of lower rank*, Psychometrika **1** (1936), no. 3, 211–218.

[12] Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran, *Cache oblivious algorithms*, Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1999, pp. 285–297.

[13] Gene H. Golub and William Kahan, *Calculating the singular values and pseudo inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis **2** (1965), no. 2, 205–224.

[14] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, fourth ed., Johns Hopkins University Press, Baltimore, MD, 2013.

[15] William W. Hager, *Updating the inverse of a matrix*, SIAM Review **31** (1989), no. 2, 221–239.

[16] Nathan Halko, Per Gunnar Martinsson, and Joel A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review **53** (2011), no. 2, 217–288.

[17] Nicholas J. Higham, *Accuracy and stability of numerical algorithms*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.

[18] Nicholas J. Higham and Theo Mary, *Mixed precision algorithms in numerical linear algebra*, 2021, MIMS EPrint 2021.20, Sections 3–4; classical real dot-product forward-error model.

[19] Jia Wei Hong and H. T. Kung, *I o complexity: The red blue pebble game*, Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, 1981, pp. 326–333.

[20] IEEE Standards Association, *Ieee standard for floating point arithmetic*, IEEE Std 754 2019, 2019.

[21] Per Gunnar Martinsson and Joel A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numerica **29** (2020), 403–572.

[22] Leon Mirsky, *Symmetric gauge functions and unitarily invariant norms*, The Quarterly Journal of Mathematics **11** (1960), 50–59.

[23] NVIDIA Corporation, CUDA C++ programming guide, NVIDIA Corporation, Archive version 12.8.1, Section 3.2.8.7 (CUDA Graphs), accessed 6 October 2026.

[24] Ivan V. Oseledets, *Tensor train decomposition*, SIAM Journal on Scientific Computing **33** (2011), no. 5, 2295–2317.

[25] Yousef Saad, *Iterative methods for sparse linear systems*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[26] Volker Strassen, *Gaussian elimination is not optimal*, Numerische Mathematik **13** (1969), 354–356.

[27] Samuel Williams, Andrew Waterman, and David Patterson, *Roofline: An insightful visual performance model for multicore architectures*, Communications of the ACM **52** (2009), no. 4, 65–76.

Chapter IX mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2688

Return to Chapter IX

References

[1] Yuta Nishibuchi, p-adic Congruences of Generalized Euler Numbers and Relations to Even Zeta Values, arXiv:2605.10677v2, 1 June 2026. https://arxiv.org/abs/2605.10677v2. Comparison: Conjecture 3.9.

[2] Ira M. Gessel, Applications of the classical umbral calculus, arXiv:math/0108121v3, 22 November 2001, Section 9, in particular the Frobenius Euler congruence and the derivation following Theorem 9.2. https://arxiv.org/abs/math/0108121.

[3] NIST Digital Library of Mathematical Functions, §24.10(i), Eq. 24.10.1 (Bernoulli denominator theorem). https://dlmf.nist.gov/24.10.E1. Used for the positive even-index denominator specialization.

[4] The Mathlib Community, Mathlib, source revision aeb72f1e9981b6d35b426b16dca7759be3e7ba06, used with Lean 4.34.0-rc2. In particular, the polynomial, strict inverse-function and analytic inversefunction developments cited in Section 11. The exact local source paths and dependency records accompany the source files.

[5] Komorebi P. Euler Congruences. Chapter IX and its formal companion in this volume. The accompanying source collection preserves the manuscript, formal supplements and continuation records.

References for Appendix D. These sources support Appendix D's citations. Its mathematical hypotheses and formal coverage remain as stated there.

[6] Yvette Amice, *Interpolation p-adique*, Bulletin de la Société Mathématique de France **92** (1964), 117–180. https://doi.org/10.24033/bsmf.1606.

[7] Stefano Barbero, Umberto Cerruti, and Nadir Murru, *Some Combinatorial Properties of the Hurwitz Series Ring*, Ricerche di Matematica **67** (2018), no. 2, 491–507. https://doi.org/10.1007/s11587-017-0336-x.

[8] Leonard Carlitz and Jack Levine, *Some Problems Concerning Kummer's Congruences for the Euler Numbers and Polynomials*, Transactions of the American Mathematical Society **96** (1960), no. 1, 23–37. https://doi.org/10.1090/s0002-9947-1960-0115971-x.

[9] Kwang-Wu Chen, *Congruences for Euler Numbers*, The Fibonacci Quarterly **42** (2004), no. 2, 128–140. https://doi.org/10.1080/00150517.2004.12428429.

[10] Louis Comtet, *Advanced Combinatorics: The Art of Finite and Infinite Expansions*, D. Reidel Publishing Company, Dordrecht, 1974. https://doi.org/10.1007/978-94-010-2196-8.

[11] Graham Everest, Alf van der Poorten, Igor Shparlinski, and Thomas Ward, *Recurrence Sequences*, Mathematical Surveys and Monographs, vol. 104, American Mathematical Society, Providence, 2003. https://doi.org/10.1090/surv/104.

[12] Fernando Q. Gouvêa, *p-adic Numbers: An Introduction*, second ed., Universitext, Springer, Berlin, 1997. https://doi.org/10.1007/978-3-642-59058-0.

[13] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, second ed., Addison-Wesley, Reading, Massachusetts, 1994.

[14] Kenneth Ireland and Michael Rosen, *A Classical Introduction to Modern Number Theory*, second ed., Graduate Texts in Mathematics, vol. 84, Springer, New York, 1990. https://doi.org/10.1007/978-1-4757-2103-4.

[15] William F. Keigher, *On the Ring of Hurwitz Series*, Communications in Algebra **25** (1997), no. 6, 1845–1859. https://doi.org/10.1080/00927879708825957.

[16] Neal Koblitz, *p-adic Numbers, p-adic Analysis, and Zeta-Functions*, second ed., Graduate Texts in Mathematics, vol. 58, Springer, New York, 1984. https://doi.org/10.1007/978-1-4612-1112-9.

[17] Rudolf Lidl and Harald Niederreiter, *Finite Fields*, second ed., Encyclopedia of Mathematics and its Applications, vol. 20, Cambridge University Press, Cambridge, 1997. https://doi.org/10.1017/CBO9780511525926.

[18] Édouard Lucas, *Théorie des Fonctions Numériques Simplement Périodiques*, American Journal of Mathematics **1** (1878), no. 2, 184–196. https://doi.org/10.2307/2369308.

[19] Kurt Mahler, *An Interpolation Series for Continuous Functions of a p-adic Variable*, Journal für die Reine und Angewandte Mathematik **199** (1958), 23–34. https://doi.org/10.1515/crll.1958.199.23.

[20] Alain M. Robert, *A Course in p-adic Analysis*, Graduate Texts in Mathematics, vol. 198, Springer, New York, 2000. https://doi.org/10.1007/978-1-4757-3254-2.

[21] Steven Roman, *The Umbral Calculus*, Pure and Applied Mathematics, vol. 111, Academic Press, New York, 1984.

[22] Jean-Pierre Serre, *Local Fields*, Graduate Texts in Mathematics, vol. 67, Springer, New York, 1979. https://doi.org/10.1007/978-1-4757-5673-9.

[23] Richard P. Stanley, *Enumerative Combinatorics. Volume 1*, second ed., Cambridge Studies in Advanced Mathematics, vol. 49, Cambridge University Press, Cambridge, 2012. https://doi.org/10.1017/CBO9781139058520.

[24] Zhi-Hong Sun, *Congruences Involving Bernoulli and Euler Numbers*, Journal of Number Theory **128** (2008), no. 2, 280–312. https://doi.org/10.1016/j.jnt.2007.03.003.

[25] Lawrence C. Washington, *Introduction to Cyclotomic Fields*, second ed., Graduate Texts in Mathematics, vol. 83, Springer, New York, 1997. https://doi.org/10.1007/978-1-4612-1934-7.

[26] Paul Thomas Young, *Congruences for Bernoulli, Euler, and Stirling Numbers*, Journal of Number Theory **78** (1999), no. 2, 204–227. https://doi.org/10.1006/jnth.1999.2401.

[27] Paul Thomas Young, *Kummer Congruences for Values of Bernoulli and Euler Polynomials*, Acta Arithmetica **99** (2001), no. 3, 277–288. https://doi.org/10.4064/aa99-3-5.

Chapter X mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2691

Return to Chapter X

References

[1] Michael F. Atiyah and Ian G. Macdonald. *Introduction to Commutative Algebra*. Addison-Wesley, Reading, Massachusetts, 1969.

[2] Joël Bellaïche and Gaëtan Chenevier. *Families of Galois Representations and Selmer Groups*, volume 324 of *Astérisque*. Société Mathématique de France, Paris, 2009.

[3] Nicolas Bourbaki. *Commutative Algebra, Chapters 1 to 7*. Elements of Mathematics. Springer, Berlin, 1989.

[4] Richard Brauer and Cecil Nesbitt. On the modular characters of groups. *Annals of Mathematics*, 42(2):556–590, 1941.

[5] Gaëtan Chenevier. The p-adic analytic space of pseudocharacters of a profinite group and Pseudorepresentations over arbitrary rings. In Fred Diamond, Payman L. Kassaei, and Minhyong Kim, editors, *Automorphic Forms and Galois Representations. Volume 1*, volume 414 of *London Mathematical Society Lecture Note Series*, pages 221–285. Cambridge University Press, Cambridge, 2014.

[6] Charles W. Curtis and Irving Reiner. *Representation Theory of Finite Groups and Associative Algebras*, volume 11 of *Pure and Applied Mathematics*. Interscience, New York, 1962.

[7] Jan Denef. The rationality of the Poincaré series associated to the p-adic points on a variety. *Inventiones Mathematicae*, 77:1–23, 1984.

[8] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, fifth edition, 2017.

[9] David Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry*, volume 150 of *Graduate Texts in Mathematics*. Springer, New York, 1995.

[10] John L. Kelley. *General Topology*. University Series in Higher Mathematics. Van Nostrand, New York, 1955.

[11] Mark Kisin. Deformations of $G_{\mathbb{Q}_p}$ and $\mathrm{GL}_2(\mathbb{Q}_p)$ representations. In Laurent Berger, Christophe Breuil, and Pierre Colmez, editors, *Représentations p-adiques de Groupes p-adiques II: Représentations de* $\mathrm{GL}_2(\mathbb{Q}_p)$ *et* (φ, Γ)*-modules*, volume 330 of *Astérisque*, pages 511–528. Société Mathématique de France, Paris, 2010.

[12] Michael Larsen and Richard Pink. On ℓ-independence of algebraic monodromy groups in compatible systems of representations. *Inventiones Mathematicae*, 107(3):603–636, 1992.

[13] Hideyuki Matsumura. *Commutative Ring Theory*, volume 8 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, corrected paperback edition, 1989. First paperback edition (with corrections), 1989; English translation first published 1986.

[14] Barry Mazur. Deforming Galois representations. In Yasutaka Ihara, Kenneth Ribet, and Jean-Pierre Serre, editors, *Galois Groups over* $\mathbb{Q}$, volume 16 of *Mathematical Sciences Research Institute Publications*, pages 385–437. Springer, New York, 1989.

[15] Jürgen Neukirch. *Algebraic Number Theory*, volume 322 of *Grundlehren der Mathematischen Wissenschaften*. Springer, Berlin, 1999.

[16] Jürgen Neukirch, Alexander Schmidt, and Kay Wingberg. *Cohomology of Number Fields*, volume 323 of *Grundlehren der Mathematischen Wissenschaften*. Springer, Berlin, second edition, 2008.

[17] Ravi Ramakrishna. On a variation of Mazur's deformation functor. *Compositio Mathematica*, 87(3):269–286, 1993.

[18] Luis Ribes and Pavel Zalesskii. *Profinite Groups*, volume 40 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge*. Springer, Berlin, second edition, 2010.

[19] Michael Schlessinger. Functors of Artin rings. *Transactions of the American Mathematical Society*, 130(2):208–222, 1968.

[20] Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics*. Springer, New York, 1977.

[21] Jean-Pierre Serre. *Local Fields*, volume 67 of *Graduate Texts in Mathematics*. Springer, New York, 1979.

[22] Jean-Pierre Serre. *Abelian ℓ-adic Representations and Elliptic Curves*. A K Peters, Wellesley, Massachusetts, 1998.

[23] Charles A. Weibel. *An Introduction to Homological Algebra*, volume 38 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1994.

Chapter XI mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2694

Return to Chapter XI

References

1. Bruce C. Berndt, Ronald J. Evans, and Kenneth S. Williams, *Gauss and Jacobi sums*, Canadian Mathematical Society Series of Monographs and Advanced Texts, Wiley, New York, 1998.
2. Corinne Blondel, *Représentation de Weil et beta extensions*, Annales de l'Institut Fourier **62** (2012), no. 4, 1319–1366.
3. Kenneth S. Brown, *Cohomology of groups*, Graduate Texts in Mathematics, vol. 87, Springer, New York, 1982.
4. Colin J. Bushnell and Guy Henniart, *Local tame lifting for GL(N). I. simple characters*, Publications Mathématiques de l'IHÉS **83** (1996), 105–233.
5. ———, *Local tame lifting for GL(N). II. wildly ramified supercuspidals*, Astérisque, vol. 254, Société Mathématique de France, 1999.
6. ———, *The essentially tame local Langlands correspondence. I*, Journal of the American Mathematical Society **18** (2005), no. 3, 685–710.
7. ———, *The essentially tame local Langlands correspondence. II. totally ramified representations*, Compositio Mathematica **141** (2005), no. 4, 979–1011.
8. ———, *The local Langlands conjecture for GL(2)*, Grundlehren der mathematischen Wissenschaften, vol. 335, Springer, Berlin, 2006.
9. ———, *The essentially tame local Langlands correspondence. III. the general case*, Proceedings of the London Mathematical Society **101** (2010), no. 2, 497–553.
10. ———, *Higher ramification and the local Langlands correspondence*, Annals of Mathematics **185** (2017), no. 3, 919–955.
11. Colin J. Bushnell and Philip C. Kutzko, *The admissible dual of GL(N) via compact open subgroups*, Annals of Mathematics Studies, vol. 129, Princeton University Press, Princeton, 1993.
12. Harold Davenport and Helmut Hasse, *Die nullstellen der kongruenzzetafunktionen in gewissen zyklischen fällen*, Journal für die reine und angewandte Mathematik **172** (1935), 151–182.
13. Pierre Deligne and George Lusztig, *Representations of reductive groups over finite fields*, Annals of Mathematics **103** (1976), no. 1, 103–161.
14. Andrea Dotto, *Canonical beta extensions*, Documenta Mathematica **27** (2022), 295–313.
15. Paul Gérardin, *Weil representations associated to finite fields*, Journal of Algebra **46** (1977), no. 1, 54–101.
16. Shamgar Gurevich and Ronny Hadani, *The geometric Weil representation*, Selecta Mathematica. New Series **13** (2007), no. 3, 465–481.
17. Michael Harris and Richard Taylor, *The geometry and cohomology of some simple Shimura varieties*, Annals of Mathematics Studies, vol. 151, Princeton University Press, Princeton, 2001.
18. Guy Henniart, *Caractérisation de la correspondance de Langlands locale par les facteurs ε de paires*, Inventiones Mathematicae **113** (1993), no. 2, 339–350.
19. ———, *Une preuve simple des conjectures de Langlands pour GL(n) sur un corps p adique*, Inventiones Mathematicae **139** (2000), no. 2, 439–455.
20. Guy Henniart and Bertrand Lemaire, *Changement de base et induction automorphe pour* GL_n *en caractéristique non nulle*, Mémoires de la Société Mathématique de France, no. 124, Société mathématique de France, 2011.
21. I. Martin Isaacs, *Character theory of finite groups*, Academic Press, New York, 1976.
22. Gregory Karpilovsky, *Projective representations of finite groups*, Monographs and Textbooks in Pure and Applied Mathematics, vol. 94, Marcel Dekker, New York, 1985.
23. George W. Mackey, *Induced representations of locally compact groups. I*, Annals of Mathematics **55** (1952), no. 1, 101–139.
24. John Milnor and Dale Husemoller, *Symmetric bilinear forms*, Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 73, Springer, Berlin, 1973.
25. Markus Neuhauser, *An explicit construction of the metaplectic representation over a finite field*, Journal of Lie Theory **12** (2002), no. 1, 15–30.
26. Amritanshu Prasad, *On character values and decomposition of the Weil representation associated to a finite abelian group*, The Journal of Analysis **17** (2009), 73–85.
27. Jean-Pierre Serre, *Local class field theory*, Algebraic Number Theory (J. W. S. Cassels and A. Fröhlich, eds.), Academic Press, London, 1967, pp. 128–161.
28. ———, *Linear representations of finite groups*, Graduate Texts in Mathematics, vol. 42, Springer, New York, 1977.
29. John Tate, *Number theoretic background*, Automorphic Forms, Representations and L-Functions, Part 2 (Armand Borel and William Casselman, eds.), Proceedings of Symposia in Pure Mathematics, vol. 33, American Mathematical Society, 1979, pp. 3–26.

Arithmetic realization studies: direct construction

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Research companion 2696

Return to the arithmetic studies

References

[1] Komorebi P. Tame and Wild Packet Geometry. Chapter XI of this volume, Sections 16 and 19 and construction appendices.

[2] Komorebi P. Formal companion to Chapter XI of this volume, Chapters 1, 13–14 and source catalogue.

[3] *Original Contract162*. Retained source in Closure Attempt162, also preserved among Block 167 dependencies.

[4] *Continuation 110: Arithmetic Bridge*. Retained continuation and its handoff, source contracts and evidence.

[5] *Continuation 111: Canonical Attachment*. Retained README, source contract and conditional proof audit.

[6] *Continuation112: Realization Test*. Retained complex-cochain and whole-carrier obstruction records.

[7] *Continuation113: Arithmetic Descent*. Retained cubic occurrence and corrected-character test records.

[8] *Finite Extension Test*. Extension Diagnosis163. Orthogonal central products, scalar extensions and coprime actions.

[9] *Compact Binder Theorem*. Closure Attempt162, compact branch. Fixed templates, joint finite solvability and exact limits.

[10] *Two Input Witness164*. Status, wild and unramified pair records, and twist-transport notes. Conditional source dependencies retained.

[11] Proof166. Direct trace-zero realization in arbitrary odd-prime length. Ordinary proof synthesis.

[12] *Trace Zero Artin–Schreier Tower166* and *Recursive Tower166*. Parameter, corestriction and defining-sequence arguments.

[13] *Trace Zero Character Induction166*. Full native characters, transfer, correction and shallow-unit extension.

[14] *Leading Layer Capacity166*. Host subspace and scalar-pushout attachment. Its separate classical induction application is not an input to the direct constructor.

[15] Proof167. Weighted dyadic realization and low-rank enlargement.

[16] *Dyadic Artin–Schreier Tower167*. Recursive matrices, native trace one and defining levels.

[17] *Dyadic Full Character167*. Depth-weighted compatibility and full-character extension.

[18] *Dyadic Capacity167*. Graded quotient counts, low-rank obstruction and skipped-degree profile comparison.

[19] *Dyadic Markings167*. Whole finite extensions, transported lifts and independent native quadratic refinements.

[20] *Enlarged Carrier Contract167*. Actual compact carriers, quotient recovery and central-normalized finite projectors.

[21] *Rank2 Enlarged Carrier167*. Actual rank-two compact group and Heisenberg enlargement.

[22] *Rank4 Enlarged Carrier167* and *Rank4 Induction by Stages167*. Actual rank-four carrier, normalization and same-candidate induction.

[23] *Stabilizer Contract166*. Original source hypotheses and the stronger native inner-normalizer obstruction.

[24] *Scalar Native Attachment166*. Genuine whole-carrier inertia phases and finite-cover descent.

[25] Colin J. Bushnell and Philip C. Kutzko. *The Admissible Dual of GL(N) via Compact Open Subgroups*. Annals of Mathematics Studies 129. Princeton University Press, 1993. Exact theorem and page locators are given in Section 13.

Arithmetic realization studies: monomial routes and Clifford descent

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Research companion 2699

Return to the arithmetic studies

Source register

The machine-readable source map gives package-relative paths, SHA-256 hashes and selected locators. The short labels below identify the sources cited in the text. A source title containing “completion” does not override its actual hypotheses or verification status.

M *PSC Langlands Monomial Completion*, recovered manuscript and *Completion Ledger*. In particular: local monomial completion, prime-cyclic subgroup chains, and global monomial induction. Classical rank-one and automorphic-induction inputs remain explicit.

M45 Actual Monomial Local Global. Lean source, source-lineage and claim-classification notes, a preliminary preservation record and two incomplete build logs. The terminal successful receipt was not recovered.

C112 *Continuation112 Realization Test*. The genuine characteristic-five Clifford fixture, symmetric characteristic-three fixture, and source-realizability report. Their arithmetic spectrum restrictions are retained with their original scope.

C113 *Arithmetic Construction and Exact Occurrence Boundary113*. Characteristic-zero source, independent simple character, cyclic automorphic induction, tame descent and central normalization. Rational cyclotomic programs retained.

C113-S
Self Twist Audit113, *Failures113*, and the independent all-extension obstruction check. Original-character exclusion, the changed full-character correction, and the unresolved corrected-character occurrence.

C161 *Clifford Size and Inputs161*. Genuine normal-chain dimensions and the conditional host-capacity comparison. Abstract packet lists are not substituted for actual source multiplicities.

B162 *Compact Binder Theorem*, Closure Attempt162. Fixed compact models, simultaneous congruences and exact limits.

Corner
Structural Corner Note, explicit binder extension, with the full sixteen-coordinate symbolic calculation. Fifth-coordinate repair and surviving leading-coefficient obstruction.

B165 *General Cubic Binder165*. Square-zero support, whole-cycle corestriction and the arbitrary-exponent degree-nine construction, under its arithmetic refinement hypotheses.

BH99 Colin J. Bushnell and Guy Henniart. *Local tame lifting for* $\mathrm{GL}(N)$. *II: Wildly ramified supercuspidals*. Astérisque 254, 1999. Selected statements: Theorem 1.3, Theorem 1.8, Proposition 2.2, §2.3, and §5.4. Printed page references and the source-access record are in the audit file.

Imai Koto Imai. *Ramification groups of some finite Galois extensions of maximal nilpotency class over local fields of positive characteristic*. Manuscripta Mathematica 176, article 66 (2025), DOI 10.1007/s00229-025-01664-5. Example 3.8(1) and its stated ramification hypotheses. The source audit records the exact publication metadata returned by the primary source.

Arithmetic realization studies: conductors and native markings

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Research companion 2701

Return to the arithmetic studies

30.4 Source register

The package's machine-readable source map supplies paths and hashes for inherited records. Page numbers below are printed pages unless a PDF-page locator is explicitly stated. Web-viewed primary sources are recorded as viewed sources, not as locally downloaded byte-pinned copies.

[Q13] *Adjoint Conductor Test161*. Retained dependency of Block166. The source invariant and conditional period-nine actual-order certificate. SHA-256 and exact source path are in the source map.

[Q14] *Principal Order Conductor161*. Principal grades, leading-level normalization and necessary source matching.

[Q15] *Arithmetic Constructor Audit*. The original source-defined induction and tame-descent branch, with full-character and source-input boundaries.

[Q16] *Dyadic Markings167*. Full central extensions, transported finite markings, fixed native refinements and the sharp capacity criterion.

[D167] *Dyadic Capacity167*. Full and skipped defining profiles, floor/ceiling grade counts, and the rank-eight shared-invariant example.

[BK] Colin J. Bushnell and Philip C. Kutzko. The Admissible Dual of GL(N) via Compact Open Subgroups. Annals of Mathematics Studies 129, Princeton University Press, 1993. Relevant inherited inputs: recursive character orders, minimal commutator control, refinement, 6.2.5 and compact-type formal degree. The earlier selected audit remains distinguished from the recorded conductor and marking source audit.

[BH99] Colin J. Bushnell and Guy Henniart. Local tame lifting for GL(n) II: wildly ramified supercuspidals. Astérisque 254, 1999. Selected audit: introduction p. 2, 1.3 p. 15, 1.7 p. 18, 2.3 pp. 23–25, 6.1 pp. 53–54. The PDF page index is the printed page plus four when counted from zero.

[AP] Anne-Marie Aubert and Roger Plymen. *Plancherel measure for GL(n,F) and GL(m,D): explicit formulas and Bernstein decomposition*. arXiv:math/0302169v4, 4 October 2004. Used: formulas (17), (29), Theorems 6.5 and 6.9, Remark 6.7 and the conductor convention in Section 4. The formal-degree pages inspected are pp. 19, 26–28.

[BH13] Colin J. Bushnell and Guy Henniart. *To an effective local Langlands correspondence*. arXiv:1103.5316v2, 9 October 2013. The characteristic note on p. 8 is the selected input. Its effective construction is not used to supply its own totally wild starting correspondence.

[HL11] Guy Henniart and Bertrand Lemaire. Changement de base et induction automorphe pour GL(n) en caractéristique non nulle. Mémoires de la Société Mathématique de France 124, 2011. DOI: 10.24033/msmf.436. The bibliographic record was checked. Detailed statements retain inherited audit status because the full PDF was unavailable during the recorded audit.

Chapter XII mathematical text

The citation labels below belong to this treatment. References in the mathematical text link to the corresponding entries.

READING ROUTES

Mathematical text 2703

Return to Chapter XII

References

1. Patrick B. Allen, Frank Calegari, Ana Caraiani, Toby Gee, David Helm, Bao V. Le Hung, James Newton, Peter Scholze, Richard Taylor, and Jack A. Thorne, *Potential automorphy over CM fields*, Annals of Mathematics **197** (2023), no. 3, 897–1113.
2. Michael F. Atiyah and Ian G. Macdonald, *Introduction to commutative algebra*, Addison Wesley, Reading, Massachusetts, 1969.
3. Thomas Barnet Lamb, Toby Gee, David Geraghty, and Richard Taylor, *Potential automorphy and change of weight*, Annals of Mathematics **179** (2014), no. 2, 501–609.
4. Joël Bellaïche and Gaëtan Chenevier, *Families of Galois representations and Selmer groups*, Astérisque, vol. 324, Société Mathématique de France, Paris, 2009.
5. Richard Brauer and Cecil Nesbitt, *On the modular characters of groups*, Annals of Mathematics **42** (1941), no. 2, 556–590.
6. Frank Calegari and David Geraghty, *Modularity lifting beyond the Taylor Wiles method*, Inventiones Mathematicae **211** (2018), no. 1, 297–433.
7. Ana Caraiani and Peter Scholze, *On the generic part of the cohomology of compact unitary Shimura varieties*, Annals of Mathematics **186** (2017), no. 3, 649–766.
8. Laurent Clozel, Michael Harris, and Richard Taylor, *Automorphy for some l adic lifts of automorphic mod l Galois representations*, Publications Mathématiques de l'IHÉS **108** (2008), 1–181.
9. Fred Diamond, *The Taylor Wiles construction and multiplicity one*, Inventiones Mathematicae **128** (1997), no. 2, 379–391.
10. Reinhard Diestel, *Graph theory*, fifth ed., Graduate Texts in Mathematics, vol. 173, Springer, Berlin, 2017.
11. David Eisenbud, *Commutative algebra with a view toward algebraic geometry*, Graduate Texts in Mathematics, vol. 150, Springer, New York, 1995.
12. Toby Gee, *Modularity lifting theorems*, Essential Number Theory **1** (2022), no. 1, 73–126.
13. Robert Guralnick, Florian Herzig, Richard Taylor, and Jack A. Thorne, *Adequate subgroups*, Journal of the Institute of Mathematics of Jussieu **11** (2012), no. 4, 907–920, Appendix to Jack A. Thorne, On the automorphy of l-adic Galois representations with small residual image.
14. Mark Kisin, *Moduli of finite flat group schemes, and modularity*, Annals of Mathematics **170** (2009), no. 3, 1085–1180.
15. Hideyuki Matsumura, *Commutative ring theory*, second ed., Cambridge Studies in Advanced Mathematics, vol. 8, Cambridge University Press, Cambridge, 1989.
16. Barry Mazur, *Deforming Galois representations*, Galois Groups over Q (Yasutaka Ihara, Kenneth Ribet, and Jean Pierre Serre, eds.), Mathematical Sciences Research Institute Publications, vol. 16, Springer, New York, 1989, p. 385–437.
17. Konstantin Miagkov and Jack A. Thorne, *Automorphy lifting with adequate image*, Forum of Mathematics, Sigma **11** (2023), e8, 31 pages.
18. Jürgen Neukirch, *Algebraic number theory*, Grundlehren der Mathematischen Wissenschaften, vol. 322, Springer, Berlin, 1999.
19. James Newton and Jack A. Thorne, *Torsion Galois representations over CM fields and Hecke algebras in the derived category*, Forum of Mathematics, Sigma **4** (2016), e21, 88 pages.
20. Luis Ribes and Pavel Zalesskii, *Profinite groups*, second ed., Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 40, Springer, Berlin, 2010.
21. Michael Schlessinger, *Functors of Artin rings*, Transactions of the American Mathematical Society **130** (1968), no. 2, 208–222.
22. Peter Scholze, *On torsion in the cohomology of locally symmetric varieties*, Annals of Mathematics **182** (2015), no. 3, 945–1066.

23. Jean Pierre Serre, *Linear representations of finite groups*, Graduate Texts in Mathematics, vol. 42, Springer, New York, 1977.
24. Richard Taylor and Andrew Wiles, *Ring theoretic properties of certain Hecke algebras*, Annals of Mathematics **141** (1995), no. 3, 553–572.
25. Jack A. Thorne, *On the automorphy of l adic Galois representations with small residual image*, Journal of the Institute of Mathematics of Jussieu **11** (2012), no. 4, 855–920.
26. Charles A. Weibel, *An introduction to homological algebra*, Cambridge Studies in Advanced Mathematics, vol. 38, Cambridge University Press, Cambridge, 1994.
27. Andrew Wiles, *Modular elliptic curves and Fermat's last theorem*, Annals of Mathematics **141** (1995), no. 3, 443–551.