FOUNDATIONS OF THE REALIZATION CALCULUS

Komorebi Space

A reference to the formal mathematics

Lean guide

Komorebi P.

Publication and reuse

Komorebi Space: Coherent Realizations, Obstructions, and Descent
Komorebi P.

The original writing, mathematical figures, and editable writing sources in this publication are licensed under the Creative Commons Attribution 4.0 International license. You may share and adapt that material under its terms, with attribution to Komorebi P. and identification of changes. The complete license text accompanies the source package.

`https://creativecommons.org/licenses/by/4.0/`

The author's original Lean proofs, scripts, and other software in the accompanying release use the MIT license. Third-party software, fonts, referenced works, and other external material retain their own rights and notices. The grants above do not replace those terms. Font notices and the software license accompany the source package.

This publication preserves the accepted mathematical statements and proofs. Its source map distinguishes the formal counterparts from explanatory text, illustrations, and written proof routes. The current release report records the checks actually performed and the accepted scope of the formal library.

About this guide

This guide connects the mathematical statements of *Komorebi Space* to their Lean definitions and proofs. It is intended for readers who want to inspect a theorem's hypotheses, use the library in another formal development, or reproduce its verification. The monograph supplies the mathematical narrative. The present volume supplies the route from that narrative to exact declarations.

The library constructs coherent realizations from compatible observations, describes the obstruction to a permitted correction, and studies quantitative repair in normed and Hilbert spaces. Its later analytic results include attained budget profiles, sharp dual certificates, regularization limits, and coordinate invariant stopping certificates. These results are organized below by mathematical subject.

The accompanying edition contains 162 production modules and 4 import entrypoints. Its source contains 1,859 explicitly written theorem and lemma declarations. The integrated declaration inventory contains 5,339 project declarations, a larger count that includes definitions, structure operations, instances, and compiler generated helpers. These figures describe different objects and should not be interchanged.

The formal boundary. The coherent realization, obstruction, descent, repair, and general Hilbert space results described here have checked Lean proofs. The explicit infinite diagonal family, its certified tails, the sharp limit over all real budgets, the rounded noisy and inexact rates, and the concrete range obstruction also have checked counterparts. The source map identifies the exact declarations and their hypotheses. This coverage concerns the stated results, rather than every explanatory sentence or illustration in the monograph.

The ordinary build and the fuller verification audit serve different purposes. A build elaborates the library. The accepted audit additionally checks declaration provenance, permitted axioms, source identity, reproducibility, and diagnostic controls. A separate technical record preserves that evidence without making the reader follow the development history.

Contents

1 Reading a formal statement 4

2 Using and rebuilding the library 5

3 Coherent coordinates and realization 6

4 Finite observation and change of resolution 7

5 Protected correction and obstruction 8

6 View systems, cocycles, and descent 9

7 Quantitative repair and multiscale detail 10

8 Overlap transport and holonomy 11

9 Minimum corrections and bounded realization 12

10 Nested constraints and energy 13

11 Regularized repair and certificates 14

12 Attained budgets, sharp duality, and uncertainty 15

13 Vanishing regularization and noisy stopping 16

14 The infinite diagonal model 17

15 Coordinates without an orthonormal kernel basis 21

16 Verification and formal scope 22

17 Thematic source reference 24

1 Reading a formal statement

1.1 Objects, hypotheses, and conclusions

A Lean theorem is a term of its declared proposition. Definitions determine what the proposition means. Reading only a theorem's English name can therefore hide an essential distinction: a correction in the whole permitted kernel differs from a correction in a chosen subspace, and an attainable residual differs from a residual in the closure of the attainable range.

The library places these distinctions in types. A continuous linear operator has Lean type $H \to_{L[\mathbb{R}]} Y$, conventionally written with the continuous-linear-map arrow. A permitted correction for a protected observation $p : E \to Z$ belongs to the subtype `p.ker`. Its ambient value is accessed by `a.val`. A full coordinate system is a `Module.Basis`; the basis property supplies independence and spanning. A genuine inverse coordinate change is guarded by `IsUnit C`.

The mathematical equations in this guide are readable forms of the stated propositions. For exact binders, universe levels, and typeclass arguments, inspect the source or ask Lean to print the declaration. The following commands can be placed in a file after the relevant import.

```
import KomorebiSpaceCombined

#check KomorebiSpace.CorrectionProblem.flatten_iff
#print KomorebiSpace.CorrectionProblem.flatten_iff
#print axioms KomorebiSpace.CorrectionProblem.flatten_iff
```

The first command displays the type, the second also displays the proof term, and the third displays its axiom dependencies. An identifier appearing in a module need not share the module's name. In particular, both coordinate modules contribute to the namespace `KomorebiSpace.CoordinateCovector`.

1.2 Conventions that matter in applications

For a residual map D and a protected observation p, the repaired object is $f - a$, where $a \in \ker p$. Thus its residual is $Df - Da$. The regularization objective uses $\|Ta - r\|^2$, whose sign is immaterial inside a norm, but the dual witness is naturally $r - Ta$.

The library's `gradient` is the normal equation residual

$$g_\alpha(a) = T^*(Ta - r) + \alpha a.$$

It is one half of the conventional derivative gradient of $\|Ta - r\|^2 + \alpha\|a\|^2$. Every error and stopping estimate below uses this normalization. Substituting the full derivative gradient without adjusting constants changes the statement.

Statements involving minimization assume $\alpha > 0$. Several algebraic identities involving coordinate residuals hold for every real α. The positivity hypotheses must be read from the theorem in use rather than transferred from a neighboring result.

1.3 Existence and computation

Many constructions are marked `noncomputable`. They use classical existence or choice, such as selecting the minimum norm preimage or an inverse limit presentation. Their equations and estimates are kernel checked mathematical results. They do not, by themselves, provide a floating point solver, a complexity bound, or an executable algorithm for constructing the selected object.

2 Using and rebuilding the library

2.1 The four import entrypoints

The source distribution has a `lean/` directory with the package configuration, toolchain specification, dependency pins, and the `KomorebiSpace/` module directory. The following imports are available.

`KomorebiSpace`
: The coherent realization, obstruction, descent, transport, and quantitative repair foundation.

`KomorebiSpaceRegularization`
: Regularized Hilbert repair, primal and dual certificates, and protected regularized repair.

`KomorebiSpaceCombined`
: The general library, including bounded realization, budget duality and robustness, regularization limits, and arbitrary basis coordinate certificates.

`KomorebiSpaceDiagonal`
: The actual infinite diagonal model, its quantitative bounds, its sharp budget asymptotic, its rounded noisy and inexact rates, and its range obstruction.

For a new application, import the narrowest module that contains the required theorem. To make the whole edition available in one file, use both imports:

```
import KomorebiSpaceCombined
import KomorebiSpaceDiagonal
```

The two entrypoints expose the general and diagonal source interfaces.

2.2 Pinned environment

The source targets Lean `4.34.0-rc2`, compiler revision `6a10ac8c22beadecabdbb0919c2b50214762f91d`. Its mathlib revision is `aeb72f1e9981b6d35b426b16dca7759be3e7ba06`. The distributed dependency manifest and pins identify the associated package revisions. Use the supplied `lean-toolchain` and manifest together. Updating to another toolchain or mathlib revision produces a different environment that requires a new build.

With the pinned toolchain and dependencies available, change into the `lean/` directory and run

```
lake build KomorebiSpaceCombined KomorebiSpaceDiagonal
```

This builds both entrypoints without updating dependency revisions. A matching local dependency checkout and compatible cache can be used without downloading the mathematical library again. This publication package contains sources and verification evidence, not the multi-gigabyte compiler and dependency cache.

If the dependencies are not installed, first provision the versions specified by the manifest using the normal Lean and Lake installation workflow. The exact source pins remain the authority. The technical record includes the commands used for the accepted diagonal replay and documents the host compatibility adjustment required in that environment.

2.3 Checking a user file

Once the package and imported modules have been built, save a file such as `Example.lean` alongside the package configuration. For example:

```
import KomorebiSpace.CoordinateCovector

open KomorebiSpace.CoordinateCovector

#check gram_isUnit
#check protected_coordinate_certificate
```

Elaborate the file with

```
lake env lean -j1 -DwarningAsError=true \
  -DautoImplicit=false Example.lean
```

The `-j1` flag limits the Lean process to one thread. Warning rejection and disabled automatic implicit binders help expose accidental omissions in new statements. These flags do not replace a declaration and axiom audit.

3 Coherent coordinates and realization

Module: `KomorebiSpace.Realization`

A resolution gives types X_n and restrictions $r_{ij} : X_j \to X_i$ for $i \le j$, with identity and composition laws. Its point space is constructed explicitly:

$$X.\mathrm{Point} = \{(x_n)_{n\in\mathbb{N}} : r_{ij}(x_j) = x_i \text{ whenever } i \le j\}.$$

The Lean type `Resolution.Point` is a subtype of the product of all coordinate types. No axiom asserting the existence of a global realization is added. The type may be empty when the compatibility conditions cannot be met.

An exact finite level map consists of maps $f_n : X_n \to Y_n$ that commute with every restriction. The construction `Resolution.Map.onLimit` applies the maps coordinatewise. Its output is a point because naturality converts coherence in X into coherence in Y.

Lean: `KomorebiSpace.Resolution.Map.onLimit_unique`

The uniqueness theorem says that a function $g : X.\mathrm{Point} \to Y.\mathrm{Point}$ with the specified coordinate values equals the constructed function. Coordinate equality is the entire uniqueness condition. This does not assert that every function between point spaces is induced by a map at the same resolution levels.

A cone from a type S is a compatible family $S \to X_n$. The universal property is expressed as a unique function into the explicitly constructed point type:

Lean: `KomorebiSpace.Resolution.Cone.exists_unique_lift`

$$\exists!\, F : S \to X.\mathrm{Point}, \qquad (F(s))_n = \mathrm{leg}_n(s) \quad \text{for every } s, n.$$

3.1 External realizations require explicit identifications

To interpret a coordinate object as an independently given space S, the formal interface asks for an equivalence $S \simeq X.\mathrm{Point}$. Given corresponding equivalences for source and target, `Resolution.realize` transports the induced map between point types to a map between the external spaces.

Lean: `KomorebiSpace.Resolution.realize_coordinate`

This requirement records the difference between constructing a coherent coordinate object and identifying it with a previously named mathematical object. The identification is part of the data, with a checked coordinate equation.

3.2 Adjacent restrictions

Module: `KomorebiSpace.AdjacentResolution`

Applications often begin with only adjacent maps $X_{n+1} \to X_n$. The adjacent resolution construction supplies restrictions for every ordered pair by composition. It proves that adjacent coherence suffices for all level pairs and identifies the resulting point type with the adjacent compatible sequences.

Lean: `KomorebiSpace.Resolution.Adjacent.pointEquiv`

This interface is useful when a concrete system is naturally recursive. The user supplies adjacent data once; the all-pairs coherence laws are then derived.

4 Finite observation and change of resolution

Module: `KomorebiSpace.FiniteObservation`

Finite observation means that each output coordinate factors through one input coordinate. The input level may depend on the output level:

$$\forall n\ \exists k\ \exists a : X_k \to Y_n\ \forall x \in X.\mathrm{Point}, \qquad (F(x))_n = a(x_k).$$

The definition does not contain a topology, a uniform bound on k, or a computational complexity condition. It describes exactly the finite coordinate dependence needed by the realization theorem.

The function `levelEnvelope` replaces a chosen schedule of input levels by a monotone schedule. Under projection surjectivity of the source, the library derives naturality of the finite presentation from its behavior on realized points.

Lean: `KomorebiSpace.Resolution.exists_presentation_of_finitelyObserved`

Projection surjectivity means that every element of each X_n occurs as the nth coordinate of some coherent point. It is the hypothesis that lets equality on actual points control the whole finite level type.

Lean: `KomorebiSpace.Resolution.realization_image_iff`

The resulting characterization is

$$\big(\exists f : \mathrm{EventualHom}(X, Y),\ f.\mathrm{onLimit} = F\big) \quad \Longleftrightarrow \quad F \text{ is finitely observed.}$$

The morphism type uses eventual equivalence of presentations. This is the appropriate equality when different schedules represent the same global map.

Lean: `KomorebiSpace.Resolution.finiteObservationEquiv`

The equivalence identifies these eventual morphisms with the finitely observed functions on point spaces, retaining the projection surjectivity assumption.

4.1 Qualitative transport and quantitative norms

Cofinal and reindexed transport modules formalize changes of resolution and their action on coherent objects. Their preservation statements concern the declared structures. A cofinal sampling map does not automatically preserve the supremum norm placed on a family of finite level operators.

Module: `KomorebiSpace.ScheduleNormObstruction`

Lean: `KomorebiSpace.ScheduleNormObstruction.cofinal_sampling_changes_norm`

The explicit example has an exact family of norm 1 and a sampled family of norm $1/2$. Thus a valid qualitative change of presentation can change a numerical norm. When transporting a stability estimate, use a theorem with the required quantitative hypotheses in addition to the realization equivalence.

5 Protected correction and obstruction

Module: `KomorebiSpace.Correction`

Let E, R, S be additive commutative groups. A correction problem consists of a residual homomorphism $D : E \to R$ and a protected semantics homomorphism $p : E \to S$. An admissible correction lies in $\ker p$. Its possible residual changes form the subgroup $D(\ker p)$.

The obstruction of $f \in E$ is the quotient class

$$[Df] \in R/D(\ker p).$$

The quotient is part of the Lean definition `CorrectionProblem.Obstruction`. Its vanishing is equivalent to the existence of an admissible correction with residual Df.

Lean: `KomorebiSpace.CorrectionProblem.obstruction_zero_iff`

Subtracting that correction gives the central equivalence:

Lean: `KomorebiSpace.CorrectionProblem.flatten_iff`

$$[Df] = 0 \quad \Longleftrightarrow \quad \exists y \in E, \quad Dy = 0 \text{ and } py = pf.$$

Only additive group structure is needed. The formal statement therefore separates the algebraic obstruction from later metric or Hilbert space refinements.

5.1 Invariance and witnesses

The obstruction is unchanged by admissible correction. If a homomorphism $L : R \to A$ annihilates every $D(a)$ with $pa = 0$, but $L(Df) \neq 0$, it gives a witness that the obstruction is nonzero.

Lean: `KomorebiSpace.CorrectionProblem.obstruction_invariant`

Lean: `KomorebiSpace.CorrectionProblem.nonzero_certificate`

The witness theorem excludes every permitted exact correction. It has no finite sample qualifier. Conversely, removing the protected constraint makes the elementary residual correction $a = f$ available. The semantics condition is consequently substantive mathematical input.

5.2 The full set of exact answers

Module: `KomorebiSpace.ExtensionFibers`

Once a protected exact answer exists, all such answers form a fiber. Its difference group is

$$H_0 = \ker D \cap \ker p.$$

The library proves that two answers in the same exact fiber differ by a unique element of H_0. Choosing one base answer identifies the entire fiber with this group.

Lean: `KomorebiSpace.CorrectionProblem.exists_unique_translation`

Lean: `KomorebiSpace.CorrectionProblem.fiberParameters`

This formulation distinguishes existence from uniqueness. A nonempty fiber is a singleton exactly when its homogeneous ambiguity is trivial. The parametrization depends on the chosen base answer, and a separate theorem describes changing that base.

6 View systems, cocycles, and descent

Module: `KomorebiSpace.ViewSections`

A view system associates an object type to each object of a category and restriction maps to its arrows. Its ambient type consists of sections compatible with all the restrictions. This generalizes a linearly indexed resolution to a category of observations.

Lean: `KomorebiSpace.ViewSystem.evaluations_jointly_injective`

Joint injectivity of evaluation means that two ambient sections are equal when all their view values agree. The universal cone property supplies the corresponding construction of a compatible section from any compatible family of maps.

Lean: `KomorebiSpace.ViewSystem.Cone.exists_unique_lift`

Restriction to a set of views produces a partial section. Its extension type retains the specified partial values. Nonemptiness of this type is an extension problem, not an automatic property of arbitrary local data.

6.1 The cover must support the intended compatibility

Module: `KomorebiSpace.CocycleDescent`

Group-valued transition data carry cocycle equations. To glue them, the cover must detect the equations involving composable arrows. The definition `GroupViewSystem.CompositionCover` states the needed coverage. It implies the corresponding transition coverage and supports a two-sided gluing equivalence for cocycles.

Lean: `KomorebiSpace.GroupViewSystem.CocycleMatching.descentEquiv`

The result retains transition and composition information. Object coverage alone does not supply all the data required to check a cocycle law.

6.2 Gauges and effective descent

Module: `KomorebiSpace.EffectiveCocycleDescent`

Local cocycles may agree after a change of frame. Descent data record both the local cocycles and the changes relating them. The effectuation construction uses a composition cover to assemble a global object with compatible local identifications.

Lean: `KomorebiSpace.GroupViewSystem.CocycleDescentData.effective`

Module: `KomorebiSpace.DescentEquivalence`

Full faithfulness of restriction and effectuation of objects combine into an equivalence of groupoids.

Lean: `KomorebiSpace.GroupViewSystem.CocycleDescentData.descentEquivalence`

An equivalence of groupoids records objects together with their gauge morphisms. It is stronger and more informative than a bijection between unstructured collections of representatives.

6.3 The site-level formulation

Module: `KomorebiSpace.CompositionStack`

The formal development also supplies the standard descent formulation and establishes a stack property for the patch pseudofunctor on the composition topology.

Lean: `KomorebiSpace.GroupViewSystem.patch_isStack`

The theorem refers to that explicitly defined site and pseudofunctor. It should be used with their declared cover and restriction structures.

7 Quantitative repair and multiscale detail

Module: `KomorebiSpace.SplitTowerRepair`

For a Banach tower with a bounded target splitting, the library constructs a correction operator for defects of finite level maps. The correction is bounded in the declared supremum norm. If the source restriction bound is b and the target splitting bound is ℓ, the geometric repair construction assumes

$$\ell b < 1, \qquad C = \frac{\ell}{1 - \ell b}.$$

The denominator is controlled by the explicit gap hypothesis. It is not a formal estimate valid when the gap closes.

Lean: `KomorebiSpace.BanachTower.Splitting.correction_bound`

Lean: `KomorebiSpace.BanachTower.Splitting.repair_exact`

The constructed repair removes the defect, with an error estimate proportional to the original residual. The related exactification results connect the repaired compatible family to the coherent realization.

Module: `KomorebiSpace.BanachTowerExactification`

Lean: `KomorebiSpace.BanachTower.repair_and_realize`

7.1 Separate operator identities from convergence

A finite telescoping identity can hold before a limiting reconstruction has been justified. The dyadic modules retain this separation. They construct residual details, prove finite reconstruction identities, and identify conditions for ordered or absolutely convergent infinite reconstruction.

Module: `KomorebiSpace.DyadicInfiniteReconstruction`

Lean: `KomorebiSpace.DyadicResidual.System.infinite_details_injective`

All details together determine the residual. This injectivity theorem does not assert that every arbitrary sequence of candidate details is summable or arises from a residual.

Lean: `KomorebiSpace.DyadicResidual.System.absolute_residual_reconstruction`

The absolute reconstruction theorem supplies a sufficient summability condition. A separate route uses decay of the coarse remainder under the stated bound on the target splitting.

Lean: `KomorebiSpace.DyadicResidual.System.residual_reconstruction_of_coarse_decay`

7.2 Coordinatewise and norm reconstruction

Lean: `KomorebiSpace.DyadicResidual.System.coordinatewise_reconstruction`

Coordinatewise stabilization or convergence and convergence in a global norm are distinct assertions. The source names distinguish them, and the monograph explains the examples that separate them. When applying a reconstruction theorem to an infinite object, keep the convergence mode and its hypothesis together.

8 Overlap transport and holonomy

Module: `KomorebiSpace.OverlapPathRepair`

Overlap transport describes compatibility along the edges of a graph. A routing chooses a path from a root to each vertex. Integrating a residual along those paths constructs a correction. In the finite graph setting, the routing radius bounds the number of accumulated edge errors.

Lean: `KomorebiSpace.OverlapTransport.Routing.correction_bound`

For actual residuals of vertex data, the path integral telescopes. This yields an exactness characterization and a uniqueness theorem for the repaired data with the specified root value.

Lean: `KomorebiSpace.OverlapTransport.Routing.unique_exact_repair`

Different routings can agree on an actual differential even when their actions on arbitrary edge assignments differ.

Lean: `KomorebiSpace.OverlapTransport.Routing.routes_agree_on_actual_residual`

The qualification about actual residuals is essential. It prevents an identity on the differential image from being used as an identity on all edge data.

8.1 Hilbert holonomy repair

Module: `KomorebiSpace.HilbertHolonomyRepair`

In the Hilbert setting, the rooted holonomy residual map determines a closed fixed subspace. Orthogonal projection selects the corrected anchor, and transport propagates it to compatible vertex data. The resulting operator is an orthogonal repair relative to the declared rooted transport structure.

Lean: `KomorebiSpace.AmplifiedTransport.Rooted.Hilbert.orthogonalRepair_parallel`

Lean: `KomorebiSpace.AmplifiedTransport.Rooted.Hilbert.orthogonalRepair_error`

The quantitative estimate uses the route gain together with the Hilbert holonomy gain. This makes the cost of propagation and the cost of resolving cyclic inconsistency separately visible.

Lean: `KomorebiSpace.AmplifiedTransport.Rooted.Hilbert.closed_range_iff_global_bound`

The closed-range theorem characterizes the existence of a global stability bound for the relevant residual operator. It is not an assumption that every bounded operator has closed range.

Lean: `KomorebiSpace.AmplifiedTransport.Rooted.Hilbert.orthogonalRepair_route_independent`

The exact route-independence result belongs to the constructed Hilbert repair. The hypotheses identify the same rooted transport problem. The theorem does not claim that every chosen numerical path procedure has identical intermediate values or costs.

9 Minimum corrections and bounded realization

Module: `KomorebiSpace.ProtectedHilbertRepair`

Let E and R be real Hilbert spaces, let Z be a real normed space, and let $D : E \to R$ and $p : E \to Z$ be continuous linear maps. The permitted correction space is the closed kernel of p. Define

$$T = D|_{\ker p} : \ker p \longrightarrow R.$$

The predicate `Feasible` asks whether Df belongs to the range of this restricted operator. Under feasibility, the minimum norm correction is selected in the orthogonal complement of $\ker T$.

Lean: `KomorebiSpace.ProtectedHilbertRepair.exists_unique_nearest`

The repaired point $f - a_{\min}$ is the unique nearest point satisfying both $Dy = 0$ and $py = pf$. Its Pythagorean identity compares it with every exact protected answer.

Lean: `KomorebiSpace.ProtectedHilbertRepair.repair_pythagoras`

Lean: `KomorebiSpace.ProtectedHilbertRepair.closed_range_iff_protected_bound`

Closed range is equivalent to a uniform protected repair bound. Existence of the nearest answer for one feasible datum and existence of a uniform bound over all feasible data are separate results.

9.1 A bounded approximate solution yields an exact solution

Module: `KomorebiSpace.BoundedApproximateRealization`

For a continuous linear map $T : H \to Y$ between real Hilbert spaces, suppose

$$\|u_n\| \le M \quad \text{for all } n, \qquad Tu_n \longrightarrow r.$$

Then there exists a with $Ta = r$ and $\|a\| \le M$.

Lean: `KomorebiSpace.BoundedApproximateRealization.exists_solution_of_bounded_approximation`

No finite-dimensionality, separability, or closed-range assumption is imposed. The argument preserves the same norm budget. It gives existence of a solution and does not claim that the supplied sequence u_n converges strongly.

The stronger filtered version only needs a nontrivial filter and a frequently recurring bound $\|u_i\| \le M$. This is enough to exclude every bounded recurrent part of an approximation to unattained data.

Lean: `KomorebiSpace.BoundedApproximateRealization.norm_tendsto_atTop_of_not_mem_range`

If $r \notin \operatorname{ran} T$ and $Tu_n \to r$, then $\|u_n\| \to +\infty$. The conclusion is divergence of the entire sequence of norms, not just lack of a uniform bound.

10 Nested constraints and energy

Module: `KomorebiSpace.NestedMinimumCorrections`

Let A_n be a decreasing sequence of nonempty closed affine subspaces of a real Hilbert space. Let a_n be the point of least norm in A_n. For $n \le m$, affine orthogonality yields

$$\|a_m - a_n\|^2 = \|a_m\|^2 - \|a_n\|^2.$$

Lean: `KomorebiSpace.NestedMinimumCorrections.nested_affine_energy`

The equality provides a direct link between the cost of a further constraint and the change in the selected correction. In particular, the norms are nondecreasing.

Lean: `KomorebiSpace.NestedMinimumCorrections.nested_affine_intersection_iff_bounded`

The common intersection is nonempty exactly when the minimum norm corrections have a finite uniform bound:

$$\bigcap_n A_n \neq \varnothing \quad \Longleftrightarrow \quad \exists M\, \forall n,\ \|a_n\| \le M.$$

The nonemptiness of each individual A_n is not sufficient on its own. The bound is what allows the infinitely many constraints to be realized simultaneously.

Lean: `KomorebiSpace.NestedMinimumCorrections.nested_affine_strong_realization`

When the bound holds, a_n converges strongly to the unique minimum norm point a_* of the intersection. The theorem also supplies the exact remaining energy identity

$$\|a_* - a_n\|^2 = \|a_*\|^2 - \|a_n\|^2.$$

Thus this particular canonical sequence has a strong convergence conclusion. That additional conclusion comes from the nested affine geometry and should not be inferred for arbitrary bounded approximate solutions.

Lean: `KomorebiSpace.NestedMinimumCorrections.nested_affine_escape`

If the intersection is empty, then $\|a_n\| \to +\infty$. This is a precise boundary for a sequence of individually solvable finite constraint problems.

10.1 Applying the result to protected correction

Module: `KomorebiSpace.ProtectedBoundedRealization`

Lean: `KomorebiSpace.ProtectedBoundedRealization.exists_exact_repair_of_bounded_approximation`

The protected version combines a residual approximation with the semantics constraint. Its conclusion is an actual exact repair with the protected value unchanged. The theorem should be applied using its typed admissible corrections, rather than treating a small observed residual as automatic exact feasibility.

11 Regularized repair and certificates

Module: `KomorebiSpace.RegularizedHilbertRepair`

For real Hilbert spaces H, Y, a continuous linear map $T : H \to Y$, a datum $r \in Y$, and $\alpha > 0$, define

$$J_\alpha(a) = \|Ta - r\|^2 + \alpha\|a\|^2.$$

The library constructs the inverse of the coercive normal operator $T^*T + \alpha I$ and defines the regularized correction a_α. It proves the normal equation and unique minimization.

Lean: `KomorebiSpace.RegularizedHilbertRepair.exists_unique_minimizer`

The exact energy identity is

$$J_\alpha(a) - J_\alpha(a_\alpha) = \|T(a - a_\alpha)\|^2 + \alpha\|a - a_\alpha\|^2.$$

Lean: `KomorebiSpace.RegularizedHilbertRepair.energy_identity`

It immediately exhibits strict uniqueness and explains why a positive parameter controls errors in the correction norm.

11.1 An inexact proposal is allowed

For $g_\alpha(a) = T^*(Ta - r) + \alpha a$, the formal error estimates are

$$\|a - a_\alpha\| \le \frac{\|g_\alpha(a)\|}{\alpha}, \qquad J_\alpha(a) - J_\alpha(a_\alpha) \le \frac{\|g_\alpha(a)\|^2}{\alpha}.$$

Lean: `KomorebiSpace.RegularizedHilbertRepair.correction_error_bound`

Lean: `KomorebiSpace.RegularizedHilbertRepair.objective_gap_bound`

The proposal a need not satisfy the normal equation. The measured quantity is the actual Hilbert norm of the normal residual.

11.2 A dual witness certifies the objective

Module: `KomorebiSpace.RegularizationCertificates`

For any $z \in Y$, define

$$L_\alpha(z) = 2\langle r, z\rangle - \|z\|^2 - \frac{\|T^*z\|^2}{\alpha}.$$

The primal and dual difference has an exact sum-of-squares form:

$$J_\alpha(a) - L_\alpha(z) = \|Ta - r + z\|^2 + \frac{\|T^*z - \alpha a\|^2}{\alpha}.$$

Lean: `KomorebiSpace.RegularizationCertificates.primal_dual_gap`

Consequently every dual value is a lower bound for every primal value. With $z = r - Ta$, the gap becomes $\|g_\alpha(a)\|^2/\alpha$.

Lean: `KomorebiSpace.RegularizationCertificates.residual_witness_gap`

The protected regularized modules apply these results to $T = D|_{\ker p}$ and prove that subtracting the selected correction preserves p exactly.

The source distribution's `examples/WORKED_EXAMPLE.md` follows one exact-rational protected proposal from the monograph to its executable certificate and these Lean declarations. It gives the input, commands, expected output, and rejection checks. The finite computation checks that recorded example; it is not a verified extraction of the general Lean proofs.

12 Attained budgets, sharp duality, and uncertainty

Module: `KomorebiSpace.BudgetProfile`

For $M \geq 0$, the budget profile is

$$\beta_{T,r}(M) = \inf_{\|a\| \leq M} \|r - Ta\|.$$

The definition is made using the distance to the image of the closed correction ball. In real Hilbert spaces this image is norm closed, even when the full operator range is not closed. The profile is attained for every nonnegative budget.

Lean: `KomorebiSpace.BudgetProfile.exists_budget_minimizer`

Let b be the orthogonal projection of r onto $\overline{\operatorname{ran} T}$. The floor is $\|r - b\|$. Reaching that floor at budget M is equivalent to b having a preimage of norm at most M.

Lean: `KomorebiSpace.BudgetProfile.beta_eq_floor_iff`

This is a criterion for exact attainment of the least residual. Merely having $b \in \overline{\operatorname{ran} T}$ is not enough to produce a finite budget preimage.

12.1 The geometry of the profile

Module: `KomorebiSpace.BudgetProfileGeometry`

The profile is nonincreasing and convex on nonnegative budgets. For $0 \leq M_1 \leq M_2$,

$$0 \leq \beta(M_1) - \beta(M_2) \leq \|T\|(M_2 - M_1), \qquad \beta(0) = \|r\|.$$

Lean: `KomorebiSpace.BudgetProfileGeometry.beta_profile_bounds`

Lean: `KomorebiSpace.BudgetProfileGeometry.beta_convex_combination`

The endpoint $M = 0$ is included. No closed-range, injectivity, or finite-dimensionality assumption is needed for these statements.

12.2 An attained dual maximum

Module: `KomorebiSpace.BudgetDuality`

The sharp formula is

$$\beta_{T,r}(M) = \max_{\|z\| \leq 1} \big(\langle r, z \rangle - M \|T^* z\| \big).$$

Lean: `KomorebiSpace.BudgetDuality.sharp_budget_duality`

The Lean statement uses `IsGreatest` on the image of the closed unit ball. Thus it proves an attained maximum, including the zero residual case, rather than only equality with a supremum. A witness exceeding a tolerance ε excludes every correction of budget M with residual at most ε.

Lean: `KomorebiSpace.BudgetDuality.strict_exclusion_certificate`

12.3 Perturbed operators and data

Module: `KomorebiSpace.BudgetRobustness`

If $\|T - T_0\| \le \tau$ and $\|r - r_0\| \le \delta$, then, for $M \ge 0$,

$$|\beta_{T,r}(M) - \beta_{T_0,r_0}(M)| \le \delta + M\tau.$$

Lean: `KomorebiSpace.BudgetRobustness.beta_abs_sub_le`

The $M\tau$ term is the amplification of operator uncertainty over the entire correction ball. The library also transports a nominal dual certificate with this explicit uncertainty deduction.

Lean: `KomorebiSpace.BudgetRobustness.nominal_dual_lower_certificate`

13 Vanishing regularization and noisy stopping

Module: `KomorebiSpace.RegularizationLimits`

The regularized path is defined for every real parameter by a total function. The limiting statements concern the right-hand filter $\alpha \to 0^+$, where it agrees with the positive-parameter minimizer. Let $b = P_{\overline{\operatorname{ran} T}} r$. Then

$$Ta_\alpha \longrightarrow b, \quad \|Ta_\alpha - r\| \longrightarrow \|r - b\|, \quad \alpha\|a_\alpha\|^2 \longrightarrow 0.$$

Lean: `KomorebiSpace.RegularizationLimits.fitted_tendsto_projectedDatum`

Lean: `KomorebiSpace.RegularizationLimits.weighted_penalty_tendsto_zero`

These conclusions do not require b to be attained. If $b \notin \operatorname{ran} T$, the correction norm tends to infinity along the entire positive-parameter filter.

Lean: `KomorebiSpace.RegularizationLimits.norm_tendsto_atTop_of_not_mem_range`

13.1 Strong convergence when the projection is attained

Module: `KomorebiSpace.RegularizationStrongLimit`

If $b = Ta_*$ for some a_*, there is a unique preimage of minimum norm. The regularized corrections converge strongly to that minimum preimage.

Lean: `KomorebiSpace.RegularizationStrongLimit.exists_unique_minimum_and_limit`

The attainment hypothesis distinguishes this conclusion from convergence of the fitted data. No closed-range assumption is added merely to ensure the strong limit for this individual attained datum.

13.2 Noise and an inexact solver

Module: `KomorebiSpace.RegularizationNoiseLimits`

For data s with $\|s - r\| \le \delta$ and a proposal a whose normal residual at the measured data satisfies $\|g_{\alpha,s}(a)\| \le \varepsilon$, the error to the minimum preimage obeys

$$\|a - a_*\| \le \frac{\varepsilon}{\alpha} + \frac{\delta}{2\sqrt{\alpha}} + \|a_\alpha(r) - a_*\|.$$

Lean: `KomorebiSpace.RegularizationNoiseLimits.stopping_three_term_bound`

The three terms respectively measure solver error, propagated data noise, and exact-data bias. The coefficient $1/(2\sqrt{\alpha})$ is sharp in a scalar example formalized in the same module.

Lean: `KomorebiSpace.RegularizationNoiseLimits.data_noise_coefficient_attained`

For a positive parameter schedule tending to zero, the sufficient scaling conditions are $\delta/\sqrt{\alpha} \to 0$ and $\varepsilon/\alpha \to 0$. The sequence theorem returns both strong convergence and the pointwise three-term certificate at every index.

Lean: `KomorebiSpace.RegularizationNoiseLimits.stopping_rule_sequence`

The more general result admits arbitrary filtered schedules. Noise tending to zero without the displayed scaling is not the theorem's hypothesis.

14 The infinite diagonal model

Module: `KomorebiSpace.DiagonalDefinitions`

Module: `KomorebiSpace.InfiniteDiagonalModel`

The explicit model uses the full real Hilbert space of square-summable sequences. Its Lean type is `lp (fun _ : Nat => Real) 2`, abbreviated by `H` in the namespace `KomorebiSpace.DiagonalRegularization`. The diagonal declarations referenced below belong to that namespace. A natural coordinate j represents the printed positive coordinate $n = j + 1$, so no denominator is evaluated at a zero coordinate.

The continuous linear map `T` and the two data vectors satisfy

$$(Tu)_n = \frac{u_n}{n}, \qquad w_n = \frac{1}{n}, \qquad b_n = \frac{1}{n^2}, \qquad b = Tw.$$

The named operator results prove the bound $\|T\| \le 1$, self-adjointness, and injectivity. The vector definitions include proofs of membership in the infinite Hilbert space. The identity `norm_sq_eq_tsum` identifies its squared norm with the full sum of squared coordinates.

Lean: `KomorebiSpace.DiagonalRegularization.T_opNorm_le_one`

Lean: `KomorebiSpace.DiagonalRegularization.T_adjoint`

Lean: `KomorebiSpace.DiagonalRegularization.T_injective`

14.1 An explicit vector and the universal minimizer

For an integer $m \ge 1$, the definition `alpha m` is m^{-2}. The actual vectors `a m` and `v m` have coordinates

$$a_n^{(m)} = \frac{m^2}{m^2 + n^2}, \qquad v_n^{(m)} = \frac{n}{m^2 + n^2}.$$

The coordinate identities are `a_apply` and `v_apply`. The normal equation proves that `a m` equals the universal regularized correction for w, and its residual is exactly `v m`.

Lean: `KomorebiSpace.DiagonalRegularization.a_eq_correction`

Lean: `KomorebiSpace.DiagonalRegularization.residual_eq_v`

Module: `KomorebiSpace.DiagonalTheorems`

Lean: `KomorebiSpace.DiagonalRegularization.a_unique_minimizer`

This last declaration quantifies over every $x \in H$. It states both that $a^{(m)}$ minimizes

$$J_m(x) = \|Tx - w\|^2 + m^{-2}\|x\|^2$$

and that every global minimizer equals $a^{(m)}$. Its hypothesis `hm : 1 <= m` supplies the strictly positive regularization parameter.

14.2 Infinite energies and certified finite sums

Module: `KomorebiSpace.DiagonalSeriesBounds`

The definitions `A m` and `B m` are infinite sums. The model proves

$$A_m = \|a^{(m)}\|^2, \qquad B_m = \|v^{(m)}\|^2.$$

The finite sums `partialA m K` and `partialB m K` use `Finset.range K`. Thus they include exactly the printed coordinates $1, \dots, K$, and the omitted tail begins at $K+1$. For $m \geq 1$ and $K \geq m$, the checked estimates are

$$\frac{m}{4} \leq A_m \leq \frac{4m}{3}, \qquad \frac{1}{4(m+1)} \leq B_m \leq \frac{2}{m},$$

$$0 \leq A_m - A_{m,K} \leq \frac{m^4}{3K^3}, \qquad 0 \leq B_m - B_{m,K} \leq \frac{1}{K}.$$

Lean: `KomorebiSpace.DiagonalRegularization.proposition_30_1_series`

Lean: `KomorebiSpace.DiagonalRegularization.proposition_30_1`

The first declaration supplies the series inequalities alone. The second also supplies equality with the universal correction, the actual residual identity, and the two Hilbert norm identities. Together with `a_unique_minimizer` and the coordinate definitions, it represents Proposition 30.1 in the monograph. An application requiring only one bound can use the nearby declarations `A_lower`, `A_upper`, `B_lower`, `B_upper`, `A_tail_upper`, and `B_tail_upper`.

14.3 The sharp limit over all real budgets

Module: `KomorebiSpace.DiagonalAsymptotic`

The two scalar limits are proved for the actual infinite sums:

$$\frac{A_m}{m} \longrightarrow \frac{\pi}{4}, \qquad mB_m \longrightarrow \frac{\pi}{4}.$$

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_A_div`

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_mul_B`

The Lean proof controls sums of the two decreasing kernels $(1+x^2)^{-1}$ and $(1+x^2)^{-2}$ by their half-line integrals. Their difference gives the residual kernel $x^2/(1+x^2)^2$. This route proves the same limits as the monograph's compact-interval and tail argument.

The profile is the existing `BudgetProfile.beta T w`, defined using the image of the actual closed Hilbert ball. Its exact value at a regularized correction budget is

$$\beta_{T,w}(\sqrt{A_m}) = \sqrt{B_m} \qquad (m \geq 1).$$

Lean: `KomorebiSpace.DiagonalRegularization.budget_at_sqrt_A`

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_actual_budget_product`

The latter theorem has the full real-budget conclusion

$$\lim_{M\to\infty} M\beta_{T,w}(M) = \frac{\pi}{4}, \qquad M \in \mathbb{R}.$$

Its source explicitly uses `fun M : Real => M * BudgetProfile.beta T w M`. The budget variable ranges over the reals, rather than only the selected sequence $\sqrt{A_m}$.

Module: `KomorebiSpace.MonotoneBudgetAsymptotic`

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_budget_product_of_mesh`

This reusable transfer theorem applies to a positive, monotone sequence $u_n \to \infty$ with $u_{n+1}/u_n \to 1$, and a nonnegative antitone profile β on $[0,\infty)$. If $u_n\beta(u_n) \to c$, then $M\beta(M) \to c$ over all real budgets. In the diagonal application, $u_n = \sqrt{A_{n+1}}$. The shift makes every mesh point positive. The proof brackets sufficiently large M between adjacent mesh points and proves that the bracket index tends to infinity.

The hard-prefix correction is also an actual element of H, defined as a finite sum of unit coordinate vectors. Its norm and residual satisfy

$$\|\mathbf{1}_N\|^2 = N, \qquad \|w - T\mathbf{1}_N\|^2 = \sum_{n>N} \frac{1}{n^2}.$$

Lean: `KomorebiSpace.DiagonalRegularization.norm_hardPrefix_sq`

Lean: `KomorebiSpace.DiagonalRegularization.norm_hardPrefix_residual_sq`

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_sqrt_mul_hardPrefix_residual`

The last declaration gives $\sqrt{N}\,\|w - T\mathbf{1}_N\| \to 1$. The bundled theorem below contains both scalar limits, the full real-budget limit, and the hard-prefix comparator, matching all four conclusions of Theorem 30.2.

Lean: `KomorebiSpace.DiagonalRegularization.sharp_diagonal_budget_asymptotic`

14.4 A concrete range obstruction

Module: `KomorebiSpace.DiagonalRangeObstruction`

The definition `unitCoordinate` constructs actual unit vectors in H. For natural coordinate j, the checked identities give norm one and image norm $1/(j+1)$. The images tend to zero in norm, and each vector belongs to the orthogonal complement of the kernel.

Lean: `KomorebiSpace.DiagonalRegularization.norm_T_unitCoordinate`

Lean: `KomorebiSpace.DiagonalRegularization.tendsto_norm_T_unitCoordinate`

The theorem `unitCoordinate_witness` supplies a coordinate that violates any proposed finite nonnegative inverse gain. Applying the general unit-witness obstruction to this operator proves

$$\neg\,\mathrm{IsClosed}(\mathrm{ran}\,T).$$

Lean: `KomorebiSpace.DiagonalRegularization.T_not_closed_range`

The images of `hardPrefix` converge in Hilbert norm to w. An exact preimage of w would have all coordinates equal to one, which is incompatible with square summability. Thus

$$w \in \overline{\mathrm{ran}\,T} \setminus \mathrm{ran}\,T.$$

Lean: `KomorebiSpace.DiagonalRegularization.w_mem_closure_not_range`

These are statements about the same infinite operator used in the regularization model.

14.5 The attainable datum and exact rounded noise schedule

For $b = Tw$, injectivity identifies w as the unique preimage and therefore the preimage of minimum norm. The vector `c m` is the universal regularized correction for b, with

$$c_n^{(m)} = \frac{m^2}{n(m^2+n^2)}, \qquad \|c^{(m)} - w\|^2 = B_m.$$

Lean: `KomorebiSpace.DiagonalRegularization.b_preimage_unique`

Lean: `KomorebiSpace.DiagonalRegularization.c_eq_correction`

Lean: `KomorebiSpace.DiagonalRegularization.norm_bias_sq`

Module: `KomorebiSpace.DiagonalNoisyRate`

For arbitrary $s \in H$ with $\|s - b\| \le \delta$ and $0 < \delta \le 1$, set

$$m = \left\lceil \delta^{-2/3} \right\rceil, \qquad \alpha = m^{-2}.$$

The definition `noiseSchedule` uses the literal natural ceiling `Nat.ceil`; the exponents are real powers. The final corollary proves both inequalities

$$\|a_\alpha(s) - w\| \le \sqrt{\frac{2}{m}} + \frac{\delta m}{2} \le (\sqrt{2}+1)\delta^{1/3}.$$

Lean: `KomorebiSpace.DiagonalRegularization.corollary_30_3`

Here $a_\alpha(s)$ denotes `RegularizedHilbertRepair.correction` applied to the measured datum s. The result admits arbitrary noise in the full Hilbert space subject to the norm bound. The exact-data bias estimate is proved for this datum inside the development. It is not a hypothesis of the final corollary. The general transfer lemmas whose names end in `of_bias_bound` retain such a hypothesis for use with other models.

14.6 The rounded rate for an inexact solver

Module: `KomorebiSpace.DiagonalInexactRate`

Keep the same s, δ, m, and α. For a proposal $a \in H$, assume $\kappa \ge 0$ and

$$\|T^*(Ta - s) + \alpha a\| \le \varepsilon \le \kappa\alpha\delta^{1/3}.$$

Then the checked conclusion is

$$\|a - w\| \le (\sqrt{2}+1+\kappa)\delta^{1/3}.$$

Lean: `KomorebiSpace.DiagonalRegularization.diagonal_inexact_rounded_rate`

The theorem type retains `Nat.ceil` and evaluates the normal residual against the measured datum s. It assumes $0 < \delta \le 1$ and $\|s - b\| \le \delta$. It derives the exact-data bias bound inside the diagonal model, then combines the noisy rate with the general solver certificate. Its tolerance hypothesis does not assert that a is an exact minimizer.

14.7 Inspecting the final interfaces

After importing the diagonal entrypoint, the principal declarations can be inspected directly:

```
import KomorebiSpaceDiagonal

open KomorebiSpace.DiagonalRegularization

#check a_unique_minimizer
#check proposition_30_1
#check sharp_diagonal_budget_asymptotic
#check corollary_30_3
#check diagonal_inexact_rounded_rate
#check T_not_closed_range
#check w_mem_closure_not_range
#print axioms sharp_diagonal_budget_asymptotic
```

The scalar series, Hilbert model, mesh transfer, rate estimates, and range obstruction remain separately importable modules. This organization lets a user reuse a specific component while retaining its exact hypotheses.

15 Coordinates without an orthonormal kernel basis

Module: `KomorebiSpace.CoordinateCovector`

Suppose the permitted correction space $\ker p$ has a finite full basis, and choose finite orthonormal coordinate bases for the real Hilbert spaces E and Y. Write the kernel basis columns in the ambient coordinates as B. Let d be the matrix of the ambient residual operator in orthonormal domain and residual coordinates. Put

$$G = B^{\mathsf{T}}B, \qquad S = dB, \qquad q = (S^{\mathsf{T}}S + \alpha G)c - S^{\mathsf{T}}r.$$

The basis in the permitted kernel is arbitrary. Its columns need not be orthonormal. The ambient and residual orthonormal coordinates are used to identify the matrices with the corresponding inner products.

The actual normal residual is a vector $g \in \ker p$, while q records its pairings with the chosen basis. Consequently the checked reconstruction and norm formulas are

$$g = BG^{-1}q, \qquad \|g\|^2 = q^{\mathsf{T}}G^{-1}q.$$

Lean: `KomorebiSpace.CoordinateCovector.protected_coordinate_certificate`

The consolidated theorem proves that G is invertible, identifies g with the projected ambient normal residual, reconstructs its ambient coordinates, and identifies the squared ambient Hilbert norm with the inverse Gram expression. Invertibility is derived from the full basis rather than postulated independently.

15.1 A covector changes by the transpose

Module: `KomorebiSpace.CoordinateBasisChange`

For any invertible real matrix C, change the basis to $B' = BC$ and the coefficients to $c' = C^{-1}c$. The same correction is represented, and the changed equation satisfies

$$G' = C^{\mathsf{T}}GC, \qquad S' = SC, \qquad q' = C^{\mathsf{T}}q.$$

Lean: `KomorebiSpace.CoordinateCovector.normalResidual_basis_change`

The transformed covector formula is derived from the changed normal equation. It is not an extra covariance assumption supplied by the user.

Lean: `KomorebiSpace.CoordinateCovector.normalResidual_certificate_basis_change`

The certificate is unchanged:

$$(q')^{\mathsf{T}}(G')^{-1}q' = q^{\mathsf{T}}G^{-1}q.$$

The associated `changedBasis` construction provides an actual full basis of the same space, with a theorem showing that the inverse coefficients synthesize the same vector.

Lean: `KomorebiSpace.CoordinateCovector.changedBasis_same_vector`

15.2 Boundary cases

The finite index type is allowed to be empty. In a zero-dimensional permitted kernel the normal residual and its certificate are zero. A proper subspace of the kernel is a different correction problem and does not certify the full projected gradient. The raw coordinate sum $q^{\mathsf{T}}q$ is basis dependent. The inverse Gram factor is necessary to recover the invariant Hilbert norm.

16 Verification and formal scope

16.1 What the accepted audit checks

The integrated audit compiles all 166 project targets from a clean source snapshot with warnings treated as errors and automatic implicit parameters disabled. It exports and checks all 5,339 project declarations together, using their defining modules to determine the audit scope. Permitted axiom dependencies are

$$\{\texttt{propext}, \texttt{Classical.choice}, \texttt{Quot.sound}\}.$$

These are standard logical principles used by Lean and mathlib. No project-specific axiom or admitted proof is accepted. The audit also checks that production declarations are not unsafe and binds its declaration inventory to the verified source files.

The integrated suite runs all 56 diagnostic controls: 18 expected successes, 35 expected elaboration rejections, and three axiom-policy rejections. It includes the 52 previously accepted controls and four additional controls for the inexact rate and concrete range obstruction. These exercise positive parameters, noise and solver hypotheses, unit dual witnesses, permitted correction spaces, nonorthogonal coordinates, empty index types, and the rejection of admitted proofs and project axioms. The current record contains fresh outcomes for every control.

An expected rejection succeeds as a diagnostic only when it fails for its intended reason. A missing import or another unrelated compiler error is not accepted as the intended mathematical rejection.

These controls complement proofs. They do not turn a finite collection of numerical examples into a universal theorem. Universal claims in the library are supported by the corresponding Lean proof terms.

16.2 Reproducibility and trust

The technical record contains compiler and dependency identities, fresh command records, source hashes, declaration exports, control outcomes, axiom reports, and independent statement reviews. It also preserves the accepted earlier evidence with its original paths. The integrated replay binds the current

results to a source snapshot and checks the installed compiler, dependency caches, and generated project objects for unchanged bytes at the recorded endpoints.

After preparing the pinned environment described in `lean/USAGE.md`, run the following command from the publication root, choosing a new output directory outside the publication:

```
python3 verification/runner/runner.py --output ../audit-run
```

The runner rebuilds the project targets, audits their declarations, and replays the controls. A normal `lake build` supplies a separate local compilation check. The proof boundary includes Lean's kernel, the specified compiler, installed external dependency caches, and the software and hardware used to run them. The external caches are recorded but are not rebuilt or independently authenticated by this procedure. Endpoint identity checks do not detect a transient modification restored before the final snapshot.

16.3 The diagonal statement correspondence

The independent reviews compare the infinite model and theorem types directly with Proposition 30.1, Theorem 30.2, and Corollary 30.3. They check the positive coordinate shift, actual Hilbert norms, complete infinite sums, global minimizer uniqueness, all-real-budget quantification, and literal rounded noise schedule. All three targets are covered with their stated hypotheses. Further independent reviews check the inexact rate against its measured-data normal residual and solver tolerance, and the range obstruction against the actual unit coordinate vectors and hard-prefix images. Both connections retain the printed hypotheses and conclusions.

Some Lean arguments take a different route from their written counterparts. Exact telescoping supplies the certified series tails, and decreasing-kernel comparisons supply the scalar asymptotics. These are proofs of the same stated conclusions. The formal scope is determined by the theorem types and the source map, rather than by requiring each written proof step to have an identically organized Lean proof.

16.4 Finding further results

The thematic reference below lists stable starting points in the source. From each one, nearby declarations provide the supporting identities, equivalences, examples, and boundary cases. A machine-readable companion map records the same entries with exact module and declaration names. For the full environment, use the declaration inventory in the verification material.

17 Thematic source reference

All modules in this table have the prefix `KomorebiSpace.`. The declaration references in the preceding chapters identify specific proved statements within them. Module names are exact, including their capitalization.

Mathematical subject	Module
Coherent realization	`Realization`
Coherent realization	`AdjacentResolution`
Finite observation	`FiniteObservation`
Finite observation	`ScheduleNormObstruction`
Protected correction	`Correction`
Protected correction	`ExtensionFibers`
Views and descent	`ViewSections`
Views and descent	`CocycleDescent`
Views and descent	`EffectiveCocycleDescent`
Views and descent	`DescentEquivalence`
Views and descent	`CompositionStack`
Multiscale repair	`SplitTowerRepair`
Multiscale repair	`BanachTowerExactification`
Multiscale repair	`DyadicInfiniteReconstruction`
Overlap and holonomy	`OverlapPathRepair`
Overlap and holonomy	`HilbertHolonomyRepair`
Bounded realization	`ProtectedHilbertRepair`
Bounded realization	`BoundedApproximateRealization`
Nested constraints	`NestedMinimumCorrections`
Nested constraints	`ProtectedBoundedRealization`
Regularized repair	`RegularizedHilbertRepair`
Regularized repair	`RegularizationCertificates`
Budgets and duality	`BudgetProfile`
Budgets and duality	`BudgetProfileGeometry`
Budgets and duality	`BudgetDuality`
Budgets and duality	`BudgetRobustness`
Limits and stopping	`RegularizationLimits`
Limits and stopping	`RegularizationStrongLimit`
Limits and stopping	`RegularizationNoiseLimits`
Infinite diagonal model	`DiagonalDefinitions`
Infinite diagonal model	`InfiniteDiagonalModel`
Infinite series and certified tails	`DiagonalSeriesBounds`
Explicit optimizer and estimates	`DiagonalTheorems`
Limits over real budgets	`MonotoneBudgetAsymptotic`
Sharp diagonal asymptotics	`DiagonalAsymptotic`
Rounded noisy parameter rate	`DiagonalNoisyRate`
Inexact diagonal rate	`DiagonalInexactRate`
Concrete range obstruction	`DiagonalRangeObstruction`
Coordinate certificates	`CoordinateCovector`
Coordinate certificates	`CoordinateBasisChange`

The companion `verification/LEAN_THEOREM_MAP.json` supplies exact source paths for all 114 selected declarations.