

# Projection Algebra as a Reusable Kernel for Mechanized Proofs: Idempotence, Spectral Decomposition, and Cost-Ordered Optimization

Recognition Science Collaboration

January 10, 2026

## Abstract

Projection operators sit at the interface of algebra, geometry, and optimization. This paper distills a “projection algebra” suitable for mechanized mathematics: (i) idempotence ( $\pi^2 = \pi$ ) as the defining axiom of projection, (ii) spectral decomposition as the source of complete families of orthogonal projectors, and (iii) cost-ordering of projectors as an optimization principle that turns projection into dynamics. We explain how this algebraic kernel supports reusable theorems in coarse-graining, renormalization, and decoherence, and why encoding it explicitly in a proof assistant (e.g. Lean) dramatically reduces the marginal cost of future formalizations.

## 1 Introduction

Projectors appear whenever a system must enforce constraints, select a subspace, or discard degrees of freedom. In physics they formalize measurement and decoherence; in applied mathematics they enforce feasibility (e.g. constrained optimization); in information theory they model coarse-graining; and in many mechanized developments they serve as the missing “glue” between analytic statements and algebraic composition.

The core observation is that *projection* is an algebraic idea with a surprisingly small axiom set, but enormous leverage. If a library makes projectors first-class objects, then many subsequent constructions become one-line instantiations rather than bespoke proofs.

We focus on three pillars:

1. **Idempotence** ( $\pi^2 = \pi$ ): projection is characterized by stabilization after one application.
2. **Spectral decomposition**: in inner-product settings, orthogonal projectors arise from eigenspace decompositions and provide complete, mutually exclusive outcomes.
3. **Cost-ordering**: attaching a cost functional to a projector turns it into an optimization primitive; composition laws then yield monotone descent guarantees for multi-stage procedures.

## 2 Abstract projector algebra

### 2.1 Idempotence and fixed points

**Definition 1** (Abstract projector). Let  $\alpha$  be a type (set). A map  $\pi : \alpha \rightarrow \alpha$  is a *projector* if it is *idempotent*:

$$\pi(\pi(x)) = \pi(x) \quad \text{for all } x \in \alpha. \quad (1)$$

The fundamental structure induced by idempotence is the fixed-point set.

**Definition 2** (Fixed points). For a projector  $\pi$ , define  $\text{Fix}(\pi) := \{x \in \alpha \mid \pi(x) = x\}$ .

**Lemma 3** (Image equals fixed points). *If  $\pi$  is idempotent, then  $\text{im}(\pi) \subseteq \text{Fix}(\pi)$ , and  $\pi$  restricts to the identity on  $\text{Fix}(\pi)$ .*

*Proof.* If  $y = \pi(x)$ , then  $\pi(y) = \pi(\pi(x)) = \pi(x) = y$ , so  $y \in \text{Fix}(\pi)$ . If  $x \in \text{Fix}(\pi)$ , then  $\pi(x) = x$  by definition.  $\square$

*Remark 4.* This lemma is one of the most “reused” facts in formal work: any time a proof shows something is a fixed point, the projector disappears.

### 2.2 Commuting composition

Composition of projectors is not always a projector, but commutation is enough.

**Proposition 5** (Commuting composition is a projector). *Let  $\pi_1, \pi_2 : \alpha \rightarrow \alpha$  be idempotent maps such that  $\pi_1 \circ \pi_2 = \pi_2 \circ \pi_1$ . Then  $\pi := \pi_1 \circ \pi_2$  is idempotent.*

*Proof.* Using commutation and idempotence:

$$\pi \circ \pi = (\pi_1 \circ \pi_2) \circ (\pi_1 \circ \pi_2) = \pi_1 \circ (\pi_2 \circ \pi_1) \circ \pi_2 = \pi_1 \circ (\pi_1 \circ \pi_2) \circ \pi_2 = (\pi_1 \circ \pi_1) \circ (\pi_2 \circ \pi_2) = \pi_1 \circ \pi_2 = \pi.$$

$\square$

*Remark 6.* This single proposition underlies multi-stage constraint enforcement, iterative coarse-graining, and many “project then project again” pipelines.

## 3 Orthogonal projectors and spectral decomposition

The preceding section is purely algebraic. In Hilbert spaces, additional structure yields *orthogonal* projectors and complete decompositions.

**Definition 7** (Orthogonal projector). Let  $H$  be a (real or complex) Hilbert space. A bounded linear operator  $P : H \rightarrow H$  is an *orthogonal projector* if it is idempotent and self-adjoint:

$$P^2 = P, \quad P^* = P. \quad (2)$$

Orthogonal projectors correspond to closed subspaces. Their key spectral fact is elementary:

**Lemma 8** (Spectrum of a projector). *If  $P$  is a (linear) projector on a vector space, then every eigenvalue  $\lambda$  of  $P$  satisfies  $\lambda \in \{0, 1\}$ .*

*Proof.* If  $Pv = \lambda v$ , then applying  $P$  again gives  $P^2v = \lambda^2v$ , but  $P^2v = Pv = \lambda v$ . Hence  $\lambda^2 = \lambda$ , so  $\lambda \in \{0, 1\}$ .  $\square$

**Proposition 9** (Decomposition into range and kernel). *Let  $P : H \rightarrow H$  be an orthogonal projector. Then*

$$H = \text{im}(P) \oplus \ker(P), \quad (3)$$

with  $\text{im}(P) \perp \ker(P)$ . Moreover,  $P$  is the identity on  $\text{im}(P)$  and zero on  $\ker(P)$ .

*Proof.* Standard: for any  $x$ , write  $x = Px + (x - Px)$ . Then  $P(x - Px) = Px - P^2x = 0$ , so  $x - Px \in \ker(P)$ . Orthogonality follows from self-adjointness: if  $y = Pu \in \text{im}(P)$  and  $z \in \ker(P)$ , then  $\langle y, z \rangle = \langle Pu, z \rangle = \langle u, P^*z \rangle = \langle u, Pz \rangle = 0$ .  $\square$

### 3.1 Complete families of orthogonal projectors

Measurement and coarse-graining use not one projector, but a complete mutually exclusive family.

**Definition 10** (Complete orthogonal projectors). Let  $H$  be a Hilbert space. A finite family  $\{P_k\}_{k=1}^m$  of operators on  $H$  is a *complete family of orthogonal projectors* if:

$$P_k^2 = P_k, \quad P_k^* = P_k \quad \text{for all } k, \quad (4)$$

$$P_i P_j = 0 \quad \text{for } i \neq j, \quad (5)$$

$$\sum_{k=1}^m P_k = I. \quad (6)$$

*Remark 11.* The spectral theorem provides such a family for any normal operator with finite spectrum; in quantum mechanics, these are precisely the projectors associated to a projective measurement.

## 4 Cost-ordered projectors and optimization

### 4.1 Projectors equipped with a cost

To turn projection into dynamics, we attach a cost functional and demand monotonic descent.

**Definition 12** (Cost-ordered projector). Let  $\alpha$  be a type and  $\pi : \alpha \rightarrow \alpha$  an idempotent map. A *cost-ordered projector* is a pair  $(\pi, \mathcal{C})$  with  $\mathcal{C} : \alpha \rightarrow \mathbb{R}$  such that:

$$\mathcal{C}(x) \geq 0 \quad \text{for all } x, \quad (7)$$

$$\mathcal{C}(\pi(x)) \leq \mathcal{C}(x) \quad \text{for all } x, \quad (8)$$

$$\pi(x) = x \Rightarrow \mathcal{C}(x) = 0. \quad (9)$$

*Remark 13.* This structure is the minimal ‘‘optimization payload’’ needed to prove global statements like: repeated projection never increases cost, and fixed points are certified minima.

### 4.2 Sequencing and commuting descent

If multiple constraints are enforced by commuting projectors, their composition inherits both idempotence and monotone descent.

**Proposition 14** (Monotone descent under commuting composition). *Let  $(\pi_1, \mathcal{C})$  and  $(\pi_2, \mathcal{C})$  be cost-ordered projectors sharing the same cost  $\mathcal{C}$ . If  $\pi_1$  and  $\pi_2$  commute, then  $\pi = \pi_1 \circ \pi_2$  is a projector and satisfies  $\mathcal{C}(\pi(x)) \leq \mathcal{C}(x)$  for all  $x$ .*

*Proof.* Idempotence follows from commuting composition. For monotonicity,

$$\mathcal{C}(\pi(x)) = \mathcal{C}(\pi_1(\pi_2(x))) \leq \mathcal{C}(\pi_2(x)) \leq \mathcal{C}(x).$$

□

## 5 Why this algebra accelerates future formalizations

Once a proof assistant library makes the above structures explicit, large families of theorems become “free”:

1. **Coarse-graining:** A coarse-graining map is typically idempotent (applying it twice adds no further information loss). Many theorems reduce to commuting-composition and fixed-point reasoning.
2. **Renormalization:** Renormalization procedures can be expressed as alternating (i) projection onto an effective subspace and (ii) rescaling. Cost-ordering provides a clean way to prove monotone improvement bounds and convergence-to-fixed-point claims.
3. **Decoherence:** Decoherence and projective measurement are naturally encoded by complete orthogonal families  $\{P_k\}$ . When these are packaged as reusable objects, normalization and orthogonality lemmas are reusable across all measurement-like arguments.

## 6 Mechanization notes (Lean)

In a mechanized setting, the critical design choice is to represent projectors as *structures* (record types) carrying their laws ( $\pi^2 = \pi$ , cost monotonicity, etc.) as fields. This makes theorems depend only on the interface, not on a concrete implementation.

In practice, such a file (e.g. `Support/Projectors.lean`) typically provides:

- an *abstract* projector interface (a map plus idempotence);
- a *cost-ordered* refinement (a cost function plus monotonicity);
- reusable lemmas: commuting composition, fixed-point simplification, and sequencing theorems.

These abstractions allow new domain modules (coarse-graining kernels, decoherence models, bridge maps) to import and reuse the same projection theorems, yielding smaller proofs and fewer ad hoc rewrites.

## 7 Conclusion

Projection algebra is a rare example of a mathematically small interface with outsized downstream impact. Idempotence provides the algebraic core; spectral decomposition supplies complete orthogonal families in inner-product spaces; and cost-ordering turns projectors into optimization primitives that support monotone descent guarantees. Encoding these ideas as reusable interfaces in a proof assistant makes future formalizations (coarse-graining, renormalization, decoherence) substantially easier: proofs become compositions of generic lemmas rather than bespoke arguments.

## References

- [1] P. R. Halmos, *Finite-Dimensional Vector Spaces*, Springer, 1974.
- [2] M. Reed and B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*, Academic Press, 1980.
- [3] J. von Neumann, *Mathematical Foundations of Quantum Mechanics*, Princeton University Press, 1955.
- [4] The Lean Community, *Lean 4 Theorem Prover*, <https://lean-lang.org/>.