

CPM Method Closure:

A Domain-Agnostic Certificate for Coercivity and Aggregation

Jonathan Washburn
Recognition Physics Institute

January 23, 2026

Abstract

The Coercive Projection Method (CPM) is a reusable proof kernel that converts three quantitative hypotheses into global, domain-independent consequences. In this paper we isolate a *method-closure* theorem for CPM: given nonnegative constants $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}})$ and four abstract functionals on a state space—defect D , orthogonal mass O , energy gap G , and test supremum T —we assume (A) a projection–defect inequality $D \leq K_{\text{net}} C_{\text{proj}} O$, (B) an energy control inequality $O \leq C_{\text{eng}} G$, and (C) a dispersion/interface inequality $O \leq C_{\text{disp}} T$. We prove the universal closure consequences:

$$D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G, \quad D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T,$$

and, when $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$, the reverse coercivity form

$$G \geq c_{\min} D, \quad c_{\min} := (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}.$$

To prevent vacuity, we additionally provide an explicit toy witness model satisfying the assumptions, yielding a self-contained certificate that CPM closure is consistent. The entire development is machine-verified in Lean 4 as a small, stable API intended for downstream domain instantiations.

Keywords. coercivity; projection methods; defect-energy inequality; aggregation; certificates; Lean formalization

Contents

1	Introduction	2
1.1	Motivation: separate the method from the instance	2
1.2	What “method closure” means in this paper	3
1.3	Why a certificate includes a witness (non-vacuity)	3
1.4	Scope and non-goals	4
1.5	Roadmap	4
2	Executive Summary of Results	4
2.1	CPM data and assumptions	4
2.2	Derived closure inequalities	4
3	Formal Setup	5
3.1	Constants and the coercivity constant c_{\min}	5
3.2	State space and functionals	6

4 CPM Assumptions (A/B/C)	6
4.1 Assumption A: Projection–Defect inequality (distance reduction)	6
4.2 Assumption B: Energy control (coercive link)	7
4.3 Assumption C: Dispersion/interface tests (aggregation link)	7
4.4 How the assumptions split: AB vs. AC	7
5 Method Closure Theorems	7
5.1 AB coercivity (forward)	7
5.2 AB coercivity (reverse, via c_{\min})	8
5.3 AC aggregation bound	8
5.4 Convenience corollaries	9
6 Non-Vacuity: A Concrete Witness Model	9
6.1 A minimal toy witness	9
7 How to Instantiate CPM in a New Domain	10
7.1 Minimal instantiation checklist	10
7.2 What method closure gives you “for free”	11
7.3 How to package CPM in a domain paper	11
7.4 A few standard downstream uses	11
8 Machine Verification (Lean 4)	12
8.1 Core method layer: abstract constants, model, and closure theorems	12
8.2 Dictionary from this paper to Lean	12
8.3 Certificates and eval-friendly reports	13
9 Discussion and Limits	13
9.1 What method closure does guarantee	13
9.2 What method closure does not guarantee	13
9.3 Common pitfalls in CPM instantiations	14
9.4 Relationship to other closure certificates (optional)	14
9.5 Recommended citation usage	14
A Algebraic Lemmas	14
B Alternative Normalizations	14

1 Introduction

The Coercive Projection Method (CPM) is best understood as a *reusable closure kernel*: it turns a small set of quantitative inequalities into universal consequences that can be transported across domains. In many settings one wants to argue that “being far from a structured set” is expensive, or that failing local checks forces a global deficit. CPM isolates the part of such arguments that is *purely algebraic* from the part that is *domain-specific*.

1.1 Motivation: separate the method from the instance

Across analysis, geometry, PDE, and number theory, a typical CPM-style proof decomposes into two layers:

- **Domain layer (hard):** define the concrete objects and supply the estimates that connect them to a structured set (projection bounds, dispersion bounds, energy bounds).

- **Method layer (routine but essential):** once those estimates are available with explicit constants, propagate them into coercivity and aggregation inequalities that drive the rest of the argument (existence, stability, positivity, local-to-global implications).

The domain layer changes from paper to paper; the method layer does not. This paper formalizes the method layer as a *standalone closure theorem* that downstream work can cite and reuse without re-proving the same algebra each time.

1.2 What “method closure” means in this paper

We work in a deliberately minimal setting. Fix a state space (type) and four real-valued functionals:

$$D, O, G, T : (\text{state}) \rightarrow \mathbb{R},$$

interpreted abstractly as:

- D — a nonnegative “distance-to-structure” proxy;
- O — a nonnegative “off-structure mass” proxy (often a squared projection norm);
- G — a nonnegative “energy gap” proxy;
- T — a nonnegative “local test supremum” proxy (dispersion/interface checks).

We assume three inequalities (A/B/C) relating these quantities via nonnegative constants ($K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}}$). Informally:

$$(A) D \lesssim O, \quad (B) O \lesssim G, \quad (C) O \lesssim T,$$

with explicit multiplicative factors.

Method closure is the claim that these three hypotheses are **closed** under a small set of universal deductions, producing the two key consequences:

$$\begin{aligned} D(x) &\leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G(x), \\ D(x) &\leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T(x), \end{aligned}$$

for every state x . When the product $K_{\text{net}} C_{\text{proj}} C_{\text{eng}}$ is strictly positive, we also derive the “reverse” coercivity form

$$G(x) \geq c_{\min} D(x), \quad c_{\min} := (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}.$$

We emphasize two design choices that matter for reuse:

- **No hidden divisions.** The forward inequalities require only nonnegativity. The reverse form explicitly tracks the additional strict-positivity hypothesis needed to invert the product.
- **Intermediate functional O .** CPM often proceeds through a tractable intermediate quantity (a projection or residual). Keeping O explicit makes the interface match how proofs are actually structured.

1.3 Why a certificate includes a witness (non-vacuity)

In a purely formal presentation, it is easy to write theorems of the form “for every model satisfying A/B/C, conclusion holds”: the conclusions can be derived by straightforward algebra. However, this has a subtle failure mode in mechanized or certificate-style settings: if the assumptions are inconsistent, then quantifying over “all models” becomes vacuous.

For this reason, our closure statement is packaged as a *certificate* that includes a concrete witness model satisfying the assumptions. The witness is intentionally simple; it is not meant to model any physical or mathematical system. Its sole purpose is to certify that the method layer is not empty: CPM’s abstract assumptions are mutually consistent and can be instantiated.

1.4 Scope and non-goals

This paper is intentionally not domain-specific. In particular:

- We do **not** prove that any particular domain satisfies A/B/C; that work belongs to domain papers.
- We do **not** define semantics or “meaning”; CPM method closure is a quantitative inequality engine, not a theory of reference.
- We do **not** commit to a particular analytic framework (Hilbert spaces, measures, PDE regularity). Those choices live in the domain layer.

1.5 Roadmap

Section 2 states the closure consequences in a compact, citation-friendly form. Section 3 fixes the constants and abstract functionals. Section 4 records the A/B/C assumptions. Section 5 proves the method-closure theorems (AB coercivity and AC aggregation). Section 6 gives the non-vacuity witness model. Section 7 provides an instantiation checklist for domain authors. Section 8 documents the Lean 4 formalization as a stable API and certificate wrapper, and Section 9 records limitations and clarifies what “closure” does and does not imply.

2 Executive Summary of Results

This section states the CPM method-closure theorem in a citation-ready form. All objects here are *domain-agnostic*: the conclusions depend only on the inequalities below.

2.1 CPM data and assumptions

Data. Fix a state space X (a type), four functionals

$$D, O, G, T : X \rightarrow \mathbb{R},$$

and nonnegative constants $K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}} \in \mathbb{R}$.

Assumptions (A/B/C). For every state $x \in X$, assume:

$$(A) \quad D(x) \leq K_{\text{net}} C_{\text{proj}} O(x), \tag{1}$$

$$(B) \quad O(x) \leq C_{\text{eng}} G(x), \tag{2}$$

$$(C) \quad O(x) \leq C_{\text{disp}} T(x). \tag{3}$$

2.2 Derived closure inequalities

Theorem 2.1 (CPM method closure: $A/B/C \Rightarrow$ coercivity and aggregation). *Assume (1)–(3) hold for all $x \in X$ with nonnegative constants $K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}} \geq 0$. Then for all $x \in X$ the following closure inequalities hold:*

$$D(x) \leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G(x), \tag{4}$$

$$D(x) \leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T(x). \tag{5}$$

Moreover, if $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$ (e.g. if $K_{\text{net}} > 0$, $C_{\text{proj}} > 0$, and $C_{\text{eng}} > 0$), define

$$c_{\min} := (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}. \tag{6}$$

Then the reverse coercivity inequality holds for all $x \in X$:

$$G(x) \geq c_{\min} D(x). \tag{7}$$

Remark 2.2 (Certificate statement (non-vacuous closure)). The phrase “CPM method closure” is used in this paper in the following precise sense:

1. **Closure:** any model (i.e. any choice of X, D, O, G, T) satisfying Assumptions (1)–(3) automatically satisfies the closure inequalities (4), (5), and (under (6)) (7).
2. **Non-vacuity:** there exists at least one concrete witness model satisfying A/B/C, so the closure theorem is not an empty implication arising from inconsistent hypotheses.

Section 6 provides an explicit witness model, and Section 8 records the Lean 4 formalization and certificate wrappers used to mechanize these statements.

3 Formal Setup

3.1 Constants and the coercivity constant c_{\min}

The CPM method layer is parameterized by four real constants:

$$K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}} \in \mathbb{R}.$$

Throughout, these constants are assumed *nonnegative*. In many applications one has $K_{\text{net}} \geq 1$ and $C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}} \geq 1$, but the method-closure algebra only requires $K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}} \geq 0$.

Two derived products. We will repeatedly use the combined constants

$$K_{\text{net}}C_{\text{proj}}C_{\text{eng}} \quad \text{and} \quad K_{\text{net}}C_{\text{proj}}C_{\text{disp}}. \quad (8)$$

The first product is the constant that governs coercivity (AB), and the second is the constant that governs the aggregation/test bound (AC).

The coercivity constant c_{\min} . The forward coercivity inequality (AB) has the form

$$D \leq (K_{\text{net}}C_{\text{proj}}C_{\text{eng}})G.$$

To rewrite this as a reverse inequality

$$G \geq c_{\min} D,$$

we must invert the product $K_{\text{net}}C_{\text{proj}}C_{\text{eng}}$. This is possible only under a *strict positivity* assumption, which we keep explicit:

$$K_{\text{net}}C_{\text{proj}}C_{\text{eng}} > 0. \quad (9)$$

When (9) holds, we define

$$c_{\min} := (K_{\text{net}}C_{\text{proj}}C_{\text{eng}})^{-1}. \quad (10)$$

We stress that (10) is a *definition*, not an extra postulate: the only additional hypothesis needed for the reverse form is the positivity condition (9).

Remark 3.1 (Why we separate forward vs. reverse). In domain work it is common to “divide by the constant” without comment. Since this paper is meant to be reusable as a certificate component (including in mechanized settings), we track precisely where invertibility is required. The forward inequalities require only nonnegativity; the reverse form requires (9).

3.2 State space and functionals

Fix a state space X . No structure on X is assumed: X may be a set, a type, a space of functions, or a space of configurations. We consider four real-valued functionals on X :

$$D, O, G, T : X \rightarrow \mathbb{R}.$$

The CPM closure result is purely *relational*: it depends only on inequalities between these functionals. It does not require norms, inner products, measures, or topology.

Intended interpretation (optional). In typical CPM instantiations, these functionals arise from a structured set S in a normed/Hilbert space and a projection map:

- $D(x)$ is a proxy for distance-to-structure, often $\text{dist}(x, S)^2$;
- $O(x)$ is a proxy for off-structure mass, often $\|\text{proj}_{S^\perp} x\|^2$;
- $G(x)$ is a proxy for an energy gap, often $E(x) - E(x_0)$ for a structured minimizer $x_0 \in S$;
- $T(x)$ is a proxy for a supremum of local test violations (dispersion/interface checks).

This paper does not assume these interpretations; they are provided only to clarify the role of the symbols.

Definition 3.2 (CPM data). A *CPM datum* is the choice of a state space X together with functionals $D, O, G, T : X \rightarrow \mathbb{R}$ and constants $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}}) \in \mathbb{R}^4$.

Remark 3.3 (What a domain must supply). To apply CPM in a new domain, the domain work supplies: (i) the concrete space X and the functionals (D, O, G, T) , and (ii) proofs of Assumptions A/B/C with explicit constants. Once those are in place, the method layer contributes the closure inequalities mechanically.

4 CPM Assumptions (A/B/C)

This section records the *only* hypotheses used by the CPM method layer. All domain-specific work enters through proofs that these inequalities hold for the chosen functionals and constants. The closure theorems in Section 5 will use no further structure.

4.1 Assumption A: Projection–Defect inequality (distance reduction)

Assumption 4.1 reduces a potentially complicated defect functional D to a more tractable intermediate quantity O (often an “orthogonal” or “off-structure” residual). The constants K_{net} and C_{proj} should be read as follows:

- K_{net} (*net/covering constant*) accounts for discretization or model reduction steps (e.g. passing from a continuum of structured directions to a finite ε -net, or from a global structured set to a finite family of local structured templates).
- C_{proj} (*projection constant*) accounts for the geometry of the reduction: how well O controls the distance-to-structure proxy encoded by D .

The method closure never inspects how K_{net} or C_{proj} are obtained; it only propagates them as factors.

Assumption 4.1 (A: Projection–Defect inequality). There exists $K_{\text{net}} \geq 0$ and $C_{\text{proj}} \geq 0$ such that for all states x ,

$$D(x) \leq K_{\text{net}} C_{\text{proj}} O(x).$$

4.2 Assumption B: Energy control (coercive link)

Assumption 4.2 is the mechanism that turns “off-structure mass” into an *energy gap*. In applications, G is usually defined relative to a structured baseline (a minimizer or a reference state), and C_{eng} is the constant that witnesses that the gap is strong enough to control deviations. At the method level, C_{eng} is just another nonnegative scalar carried through algebra.

Assumption 4.2 (B: Energy control). There exists $C_{\text{eng}} \geq 0$ such that for all states x ,

$$O(x) \leq C_{\text{eng}} G(x).$$

4.3 Assumption C: Dispersion/interface tests (aggregation link)

Assumption 4.3 is the input required for an aggregation or “local-to-global” step. The functional $T(x)$ is intended to represent a supremum over local test violations or interface checks that are easier to estimate than global defect. The constant C_{disp} quantifies how strongly these tests control the off-structure mass O .

Assumption 4.3 (C: Dispersion/interface tests). There exists $C_{\text{disp}} \geq 0$ such that for all states x ,

$$O(x) \leq C_{\text{disp}} T(x).$$

4.4 How the assumptions split: AB vs. AC

It is useful to view CPM closure as two coupled but logically distinct deductions:

- **AB (coercivity):** Assumptions 4.1 and 4.2 imply that the defect is controlled by the energy gap, yielding the coercivity inequalities (4) and (under positivity) (7).
- **AC (aggregation):** Assumptions 4.1 and 4.3 imply that the defect is controlled by the test functional, yielding (5). In applications this is the entry point for “local tests \Rightarrow global structure”.

The role of the intermediate functional O is precisely to make both deductions uniform across domains: many proofs naturally obtain bounds on a residual/projection quantity first, and only later relate it to energy or tests.

Remark 4.4 (Nonnegativity of the functionals). Many domain instantiations arrange, by definition, that for all $x \in X$,

$$D(x) \geq 0, \quad O(x) \geq 0, \quad G(x) \geq 0, \quad T(x) \geq 0.$$

Method closure does not require these inequalities explicitly: it uses only Assumptions 4.1–4.3 and basic order-preserving algebra. We therefore keep nonnegativity as an *optional* semantic convention rather than an extra hypothesis.

5 Method Closure Theorems

5.1 AB coercivity (forward)

In this subsection we prove that Assumptions 4.1 and 4.2 imply that defect is controlled by the energy gap, with an explicit multiplicative constant.

Proposition 5.1 (AB coercivity (forward)). *Assume $K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}} \geq 0$ and that Assumptions 4.1 and 4.2 hold. Then for every state $x \in X$,*

$$D(x) \leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G(x).$$

Proof. Fix $x \in X$. By Assumption 4.1,

$$D(x) \leq K_{\text{net}} C_{\text{proj}} O(x).$$

By Assumption 4.2, $O(x) \leq C_{\text{eng}} G(x)$. Since $K_{\text{net}} C_{\text{proj}} \geq 0$, multiplying preserves the inequality:

$$K_{\text{net}} C_{\text{proj}} O(x) \leq K_{\text{net}} C_{\text{proj}} (C_{\text{eng}} G(x)).$$

Combining these two inequalities yields

$$D(x) \leq K_{\text{net}} C_{\text{proj}} (C_{\text{eng}} G(x)) = (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G(x),$$

as claimed. \square

5.2 AB coercivity (reverse, via c_{\min})

The reverse form is the usual statement of coercivity: the energy gap is bounded below by a positive multiple of defect. It is equivalent to the forward form when the constant product is strictly positive, but we keep the strict-positivity condition explicit because it is exactly where division becomes valid.

Proposition 5.2 (AB coercivity (reverse)). *Assume $K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}} \geq 0$ and Assumptions 4.1 and 4.2. Assume in addition that*

$$K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0,$$

and define $c_{\min} := (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}$. Then for every $x \in X$,

$$G(x) \geq c_{\min} D(x).$$

Proof. Fix $x \in X$ and set $p := K_{\text{net}} C_{\text{proj}} C_{\text{eng}}$. By Proposition 5.1, we have $D(x) \leq p G(x)$. Since $p > 0$, multiplying both sides by p^{-1} preserves the inequality:

$$p^{-1} D(x) \leq p^{-1} (p G(x)) = G(x).$$

Recalling that $p^{-1} = c_{\min}$ proves the claim. \square

Remark 5.3 (Equivalence of forward and reverse forms). When $p := K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$, the forward inequality $D \leq p G$ and the reverse inequality $G \geq p^{-1} D$ are logically equivalent. In CPM papers one often uses whichever form is most convenient for the downstream argument (e.g. existence of minimizers, monotone descent, or stability bounds).

5.3 AC aggregation bound

The aggregation bound replaces the energy gap by a “test” functional. In applications T is typically computable or locally checkable; controlling T then yields a global defect bound.

Proposition 5.4 (AC aggregation bound). *Assume $K_{\text{net}}, C_{\text{proj}}, C_{\text{disp}} \geq 0$ and that Assumptions 4.1 and 4.3 hold. Then for every state $x \in X$,*

$$D(x) \leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T(x).$$

Proof. Fix $x \in X$. By Assumption 4.1,

$$D(x) \leq K_{\text{net}} C_{\text{proj}} O(x).$$

By Assumption 4.3, $O(x) \leq C_{\text{disp}} T(x)$. Since $K_{\text{net}} C_{\text{proj}} \geq 0$, multiplying preserves the inequality:

$$K_{\text{net}} C_{\text{proj}} O(x) \leq K_{\text{net}} C_{\text{proj}} (C_{\text{disp}} T(x)).$$

Combining yields

$$D(x) \leq K_{\text{net}} C_{\text{proj}} (C_{\text{disp}} T(x)) = (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T(x),$$

as claimed. \square

5.4 Convenience corollaries

The following corollaries are often convenient when presenting a domain instantiation, especially when the domain work has already normalized constants to 1 or has chosen O to coincide with D .

Corollary 5.5 (Normalized constants). *Assume Assumptions 4.1–4.3 hold.*

1. *If $K_{\text{net}} = C_{\text{proj}} = C_{\text{eng}} = 1$, then $D(x) \leq G(x)$ for all x .*
2. *If $K_{\text{net}} = C_{\text{proj}} = C_{\text{disp}} = 1$, then $D(x) \leq T(x)$ for all x .*

Proof. Immediate from Propositions 5.1 and 5.4 by substituting the specified constants. \square

Corollary 5.6 (Subspace case). *Assume Assumption 4.1 holds with $K_{\text{net}} = C_{\text{proj}} = 1$ and moreover that $O(x) = D(x)$ for all x . Then $D(x) = O(x)$ for all x , and the AB and AC conclusions reduce to*

$$D(x) \leq C_{\text{eng}} G(x), \quad D(x) \leq C_{\text{disp}} T(x).$$

Proof. The equality $D = O$ is by hypothesis. Substituting $K_{\text{net}} = C_{\text{proj}} = 1$ and $O = D$ into Assumptions 4.2 and 4.3 yields the displayed inequalities. \square

6 Non-Vacuity: A Concrete Witness Model

The closure implications proved in Section 5 have the schematic form

$$(\text{A/B/C assumptions}) \implies (\text{closure inequalities}).$$

As a purely logical statement, such an implication can be true even if there are *no* instances of the assumptions. In that case the implication holds vacuously. For ordinary mathematical exposition this is rarely a concern because the assumptions typically arise from a concrete construction in the same paper. In a *method closure* paper—whose purpose is to package the method layer as a reusable component—it is important to certify that the assumptions are mutually consistent, so that downstream users are not building on an empty interface.

This motivates the present section: we give an explicit toy CPM datum satisfying Assumptions 4.1–4.3. The witness is intentionally minimal. It is *not* intended to model any real domain; it exists solely to establish non-vacuity of the CPM method layer and to support certificate-style uses (see Section 8).

6.1 A minimal toy witness

Let X_{toy} be a singleton state space, i.e. $X_{\text{toy}} = \{\star\}$. Define constant functionals

$$D(\star) = 1, \quad O(\star) = 1, \quad G(\star) = 1, \quad T(\star) = 1,$$

and choose constants

$$K_{\text{net}} := 1, \quad C_{\text{proj}} := 2, \quad C_{\text{eng}} := 1, \quad C_{\text{disp}} := 1.$$

With these choices, Assumptions 4.1–4.3 reduce to numerical inequalities.

Proposition 6.1 (Existence of a non-vacuous CPM instance). *There exists a CPM datum $(X, D, O, G, T; K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}})$ satisfying Assumptions 4.1–4.3. Moreover, the coercivity product is strictly positive:*

$$K_{\text{net}} C_{\text{proj}} C_{\text{eng}} = 2 > 0,$$

so the reverse coercivity constant $c_{\min} = (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}$ is well-defined and equals $1/2$.

Proof. Take $X = X_{\text{toy}} = \{\star\}$ with the constant functionals and constants defined above. Then:

- **Assumption A:** $D(\star) = 1 \leq 1 \cdot 2 \cdot 1 = 2 = K_{\text{net}} C_{\text{proj}} O(\star)$.
- **Assumption B:** $O(\star) = 1 \leq 1 \cdot 1 = C_{\text{eng}} G(\star)$.
- **Assumption C:** $O(\star) = 1 \leq 1 \cdot 1 = C_{\text{disp}} T(\star)$.

Finally, $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} = 1 \cdot 2 \cdot 1 = 2 > 0$, so $c_{\min} = 2^{-1} = 1/2$. \square

Remark 6.2 (Why the witness is deliberately simple). The role of Proposition 6.1 is not to validate CPM in any domain, but to certify that the method assumptions are not internally contradictory. Domain papers provide substantive instantiations by defining meaningful functionals (D, O, G, T) and proving A/B/C from domain structure. In contrast, the toy witness is a consistency check for the *interface*.

7 How to Instantiate CPM in a New Domain

This section is a practical interface guide for domain authors. It explains what must be provided in a domain instantiation, what can be cited from the present paper, and how to keep constants and hypotheses organized so that CPM arguments remain auditable.

7.1 Minimal instantiation checklist

To instantiate CPM in a new domain, provide the following data and proofs. We write X for the domain's state space and keep the notation of Sections 3–4.

Step 1 (state space). Specify the class of objects you want to control and choose a corresponding state space X . Typical choices include:

- a space of fields/configurations on a manifold,
- a family of functions (e.g. boundary data, kernels, or test functions),
- a parameterized class of arithmetic objects (e.g. weights, characters, truncated Euler products),
- a space of programs/traces in a discrete system.

Step 2 (structured set, optional but clarifying). Although the method layer does not require it, it is often clarifying to identify an intended “structured” subset S of an ambient space, together with a baseline element $x_0 \in S$. In many proofs S is a subspace or cone of “admissible” or “ideal” configurations.

Step 3 (define the four functionals). Define functionals $D, O, G, T : X \rightarrow \mathbb{R}$. The most common pattern is:

- **Defect:** $D(x) = \text{dist}(x, S)^2$ or a proxy for it.
- **Orthogonal mass:** $O(x) = \|\text{proj}_{S^\perp} x\|^2$ or a proxy that is easier to estimate.
- **Energy gap:** $G(x) = E(x) - E(x_0)$ for a suitable energy functional E .
- **Tests:** $T(x)$ is a supremum (or sum) of *local* checks (dispersion/interface tests) whose control is feasible.

The method closure will treat these as black-box real-valued functions.

Step 4 (choose constants). Exhibit explicit constants $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}})$, typically derived from a combination of:

- a discretization/covering argument (contributing to K_{net}),
- a projection or rank-one estimate (contributing to C_{proj}),
- an energy comparison/coercivity estimate (contributing to C_{eng}),
- a dispersion/interface inequality (contributing to C_{disp}).

Keep these constants explicit and avoid “absorbing” them into big-O notation if the goal is a reusable certificate.

Step 5 (prove A/B/C). Prove Assumptions 4.1–4.3 for your chosen functionals and constants. In many domains, the proofs naturally factor as:

$$D \lesssim (\text{net loss}) \times (\text{projection loss}) \times O, \quad O \lesssim (\text{energy gap}), \quad O \lesssim (\text{tests}).$$

7.2 What method closure gives you “for free”

Once A/B/C are established, the following consequences require no additional domain input:

- **AB coercivity (forward):** apply Proposition 5.1 to obtain $D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G$.
- **AB coercivity (reverse):** if $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$, define $c_{\min} = (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}$ and apply Proposition 5.2 to obtain $G \geq c_{\min} D$.
- **AC aggregation:** apply Proposition 5.4 to obtain $D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T$.

In domain exposition, it is usually best to cite these three results rather than re-proving the same multiplicative chaining argument. This is precisely the purpose of a method-closure paper.

7.3 How to package CPM in a domain paper

A domain paper can present CPM usage cleanly by organizing the argument into three layers:

1. **Domain definitions:** define X and the functionals (D, O, G, T) , and record the constants.
2. **Domain estimates:** prove Assumptions 4.1–4.3 as named lemmas (often the hardest part).
3. **Closure invocation:** cite Propositions 5.1, 5.2, 5.4 to obtain the global inequalities, and then use those inequalities to finish the domain-specific theorem.

This structure has two advantages: it keeps the method layer auditable (constants are visible), and it makes it clear to readers which steps are genuinely domain-dependent.

7.4 A few standard downstream uses

While CPM is not tied to any one conclusion, the closure inequalities commonly support:

- **Stability from margins:** if you can show $G(x)$ (or $T(x)$) is small, then $D(x)$ is small, hence x is close to S in the relevant metric/proxy.
- **Convergence of descent procedures:** if an algorithm produces a sequence (x_n) with $G(x_n) \rightarrow 0$, the reverse coercivity form implies $D(x_n) \rightarrow 0$ (under $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$).
- **Local-to-global structure:** if local tests enforce $T(x) = 0$ (or $T(x)$ below a threshold), AC gives a global defect bound, which is then converted into membership or positivity by a domain lemma.

These are templates: the final step (“defect small implies property”) is always domain-specific.

8 Machine Verification (Lean 4)

The results of this paper are mechanized in the Lean 4 theorem prover in a way that deliberately mirrors the mathematical interface presented above. The goal of the mechanization is not to add new assumptions or domain content, but to provide a small, stable, audit-friendly kernel that downstream projects can import without re-proving the method algebra.

8.1 Core method layer: abstract constants, model, and closure theorems

The core formalization lives in `IndisputableMonolith/CPM/LawOfExistence.lean`. It defines:

- a constants bundle `Constants` containing $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}})$ and nonnegativity fields,
- the derived coercivity constant `cmin` (our c_{\min}),
- an abstract CPM model `Model` parameterized by a state type β ,
- and the three method-layer theorems corresponding to AB/AC closure.

8.2 Dictionary from this paper to Lean

The following correspondence is intended to be exact:

- **State space.** Paper X corresponds to Lean's type parameter β .
- **Functionals.** Paper functionals correspond to fields of a Lean model $M : \text{Model } \beta$:

$$\begin{aligned} D &\leftrightarrow M.\text{defectMass}, \\ O &\leftrightarrow M.\text{orthoMass}, \\ G &\leftrightarrow M.\text{energyGap}, \\ T &\leftrightarrow M.\text{tests}. \end{aligned}$$

- **Constants.** Paper constants correspond to fields of $M.C : \text{Constants}$:

$$(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}}) \leftrightarrow (M.C.K_{\text{net}}, M.C.C_{\text{proj}}, M.C.C_{\text{eng}}, M.C.C_{\text{disp}}).$$

- **Assumptions.** The assumptions A/B/C are precisely the structural fields of `Model`:

$$\begin{aligned} (\text{A}) &\leftrightarrow M.\text{projection_defect}, \\ (\text{B}) &\leftrightarrow M.\text{energy_control}, \\ (\text{C}) &\leftrightarrow M.\text{dispersion}. \end{aligned}$$

- **Closure theorems.** The key results of Section 5 correspond to the following Lean theorems:

- Proposition 5.1 corresponds to
`Model.defect_le_constants_mul_energyGap`.
- Proposition 5.2 corresponds to
`Model.energyGap_ge_cmin_mul_defect`, whose hypotheses include strict positivity of $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}})$ (matching our $K_{\text{net}}C_{\text{proj}}C_{\text{eng}} > 0$ requirement).
- Proposition 5.4 corresponds to
`Model.defect_le_constants_mul_tests`.

8.3 Certificates and eval-friendly reports

On top of the core method kernel, the repository provides certificate wrappers and lightweight reports. In particular, several adapters expose #eval-friendly confirmation strings:

- **Method certificate.** `IndisputableMonolith/URCGenerators/CPMMethodCert.lean` defines a certificate predicate `CPMMethodCert.verified` which (i) asserts the generic AB/AC consequences for arbitrary models and (ii) bundles a concrete witness `toyModel` to prevent vacuity.
- **Closure wrapper.** `IndisputableMonolith/URCGenerators/CPMClosureCert.lean` packages the method certificate into a minimal top-level `CPMClosureCert`.
- **Report adapter.** `IndisputableMonolith/URCAdapters/CPMClosureReport.lean` provides #eval-friendly strings (e.g. `cpm_closure_ok`) so that interactive audits can confirm the certificate without exposing implementation details.

Remark 8.1 (Relationship to the witness in this paper). The witness model used in Lean (`toyModel`) plays the same logical role as Proposition 6.1: it certifies that the interface assumptions are satisfiable. Domain work remains responsible for producing nontrivial, problem-specific models.

9 Discussion and Limits

This paper isolates the CPM *method layer* as a standalone, reusable unit. It is therefore important to be explicit about what is proved here, what is intentionally left to domain papers, and what kinds of additional structure are needed for downstream interpretations.

9.1 What method closure does guarantee

Assuming A/B/C (Section 4) with explicit constants, the method layer guarantees:

- **A uniform coercivity inequality.** AB implies $D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{eng}}) G$ (Proposition 5.1). When $K_{\text{net}} C_{\text{proj}} C_{\text{eng}} > 0$, this is equivalent to the reverse form $G \geq c_{\min} D$ with $c_{\min} = (K_{\text{net}} C_{\text{proj}} C_{\text{eng}})^{-1}$ (Proposition 5.2).
- **A uniform aggregation/test bound.** AC implies $D \leq (K_{\text{net}} C_{\text{proj}} C_{\text{disp}}) T$ (Proposition 5.4).
- **Non-vacuity of the interface.** The assumptions are satisfiable (Section 6), so “closure” is not merely a vacuous implication.

These consequences are intentionally elementary: CPM’s value is that *many* difficult domain problems can be reduced to verifying A/B/C, after which the coercivity/aggregation propagation is immediate and auditable.

9.2 What method closure does not guarantee

Method closure does *not* by itself establish any of the following:

- **That A/B/C hold in a particular domain.** The analytic or combinatorial heart of a CPM instantiation is proving Assumptions 4.1–4.3 with explicit constants; this is domain work.

- **That small defect implies the final domain theorem.** CPM delivers defect bounds; the implication “ D small \Rightarrow property” is always domain-specific and must be proved separately.
- **A theory of semantics or meaning.** CPM method closure is a quantitative inequality engine. Even if a domain chooses to interpret “structure” as “admissibility” or “normal form,” the method layer does not by itself define an aboutness relation, an object space, or an argmin selection rule. (In other words: CPM can help certify canonical *representations*; it does not, on its own, imply reference or meaning.)
- **Uniqueness of constants.** The present paper treats $(K_{\text{net}}, C_{\text{proj}}, C_{\text{eng}}, C_{\text{disp}})$ as inputs. Deriving them from deeper principles (when desired) is a separate layer.

9.3 Common pitfalls in CPM instantiations

The following issues recur in CPM-style arguments and are worth flagging explicitly:

- **Hidden non-uniformity.** If the constants depend on scale, dimension, or parameters, then the closure inequalities remain true but may lose their intended force unless one proves uniform bounds.
- **Untracked division.** The reverse coercivity form requires strict positivity of $K_{\text{net}}C_{\text{proj}}C_{\text{eng}}$. When constants are obtained by inequalities rather than exact identities, it is easy to forget this step.
- **Over-absorbing constants into asymptotic notation.** For certificate-style reuse, it is preferable to keep constants explicit and to isolate the exact lemmas where each constant is introduced.
- **Choice of intermediate functional O .** The method layer is agnostic to O , but the domain layer benefits from selecting an O that (i) is provably controlled by energy/tests and (ii) is strong enough to control defect.

9.4 Relationship to other closure certificates (optional)

In larger “bundle” developments, CPM method closure is often packaged together with additional closure statements (e.g. uniqueness or selection of a preferred scale, or consistency bundles for a broader framework). From an engineering and audit standpoint, it is advantageous to keep CPM method closure separate: it has a small dependency surface, a simple interface, and a mechanically checkable proof. Other certificates may depend on additional subprojects and should be audited as separate targets.

9.5 Recommended citation usage

When using CPM as a lemma in a domain paper, it is usually sufficient to cite Propositions 5.1, 5.2 (when applicable), and 5.4 after stating A/B/C with explicit constants. This makes clear which parts of the argument are domain estimates and which parts are method closure.

A Algebraic Lemmas

B Alternative Normalizations

References