Research Article

To Count a Stone with Six Birds: A Mathematics is A Theory

Ioannis Tsiokos[1]

1. Automorph Inc., United States

Many "higher" mathematical objects are introduced as definitions (limits, completions, analytic continuations), but it is often unclear *when* a large discrete protocol admits a stable continuous closure and *how* to test that stability without overclaiming. This question matters because premature or untested closure claims can propagate errors through dependent results. We instantiate the Six-Birds Theory (SBT) framework of [1]—which provides a general audit methodology for validating compressions of discrete data into continuous or idealized objects—as a closure-and-audit method for mathematics: a discrete substrate is staged by refinement, compressions are assessed by an explicit defect ledger, and a packaged object is accepted only when defects shrink or stabilize under refinement. Operationally, we (i) formalize two anchor statements in Lean/mathlib [2], [3]—the exact finite-difference Leibniz identity with its explicit remainder term and an algebraic uniqueness anchor for derivations on $R[X]$—and (ii) implement falsification-first Python diagnostics that compare competing closure routes under matched controls and an artifact contract (tables and macros are generated from snapshot-visible pointer JSONs). Across four exhibits we observe controlled, checkable separations. First, stability-only stencil filtering selects order-0 closures, while adding a Leibniz-defect gate selects derivative-like order-1 closures. Second, route mismatch (RM, measuring disagreement between closure paths) under a small coordinate change decays with refinement (power-law fit exponent $p \approx 1.46$). Third, for prime-based closures in a convergence control regime $\Re(s) > 1$, route mismatch RM_2 decreases with sample size N (e.g. $RM_2(800) \approx 5 \times 10^{-2}$), whereas the same diagnostic in the critical strip exhibits mismatch growth by many orders of magnitude under naive staging (e.g. $RM_2(800) \approx 5 \times 10^{12}$). Fourth, in a self-dual toy family, tightening a positivity constraint sharply confines zeros to the symmetry locus (mean radial deviation drops from ≈ 0.87 to $\approx 3 \times 10^{-4}$). These results provide a reproducible methodology to separate feasible closures from protocol artifacts and to compare closure proposals across regimes. We emphasize that this paper does

 doi.org/10.32388/JABWNG

not establish new theorems about ζ or zero distributions; we report audit-style diagnostics under explicit controls and document the upgrade points required for stronger claims.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

1. Introduction

1.1. From counting to closure

A familiar way to "count a stone" is to apply a tiny, discrete action many times: tick marks, steps along a line, repeated updates of a running total. A familiar way to *do calculus* is to replace that large discrete composition by a compressed law: a differential equation for the limit of many tiny updates, or an integral for the limit of many tiny accumulations. This paper takes that replacement seriously as a *closure problem*. The central question is not "how do we define derivatives?" but: *which compressions of discrete protocols remain coherent as we refine the stage parameter toward a continuum limit?*

In the SBTdictionary (reviewed in Section 2), refinement is **P4** staging; the defect budget carried by a compression is **P6** accounting; and the limit object is **P5** packaging. On this view, the basic moves of analysis are forced by feasibility: only those operators and equivalence relations whose defects stabilize under refinement become admissible "laws" at the packaged layer. The same lens applies beyond calculus, whenever one attempts to compress a complicated discrete structure into a continuous or spectral object.

This manuscript is a substantially revised version of the Zenodo preprint (DOI: 10.5281/zenodo.18402004); see Related Identifiers for version linkage.

1.2. Paper contract: what this paper is (and is not)

This document is a concrete instantiation of the general SBTframework of [1] in the domain of mathematics, and it treats the core toolkit as an *emergence calculus* for stable closure under refinement.

This paper is:

- a worked translation of the SBTprimitives into standard mathematical practice: limits, completions, renormalizations, and compatibility constraints;

- a validation-first report in which key claims are supported either by machine-checked lemmas (Lean) or by falsification-oriented numerical diagnostics (Python);
- a proposal for how "higher" mathematical layers can be treated as *closures* of discrete protocol algebras under refinement, rather than as isolated definitions.

This paper is not:

- a replacement for standard foundations or a proposal of new axioms;
- a proof of the Riemann Hypothesis or a claim that the diagnostics in Exhibits B-1/B-2 settle classical conjectures;
- an attempt to reduce all of mathematics to a single mechanism.

The intended reading is pragmatic: SBTis used here as a *dictionary* for organizing constructions and for designing tests that can fail. When the diagnostics fail (as they sometimes do), we treat the failure as information about missing feasibility conditions rather than as a defect to be hidden.

1.3. Contributions and artifacts

The paper contributes a small set of concrete, reusable "exhibits":

- **Exhibit A-1 (calculus closure).** We isolate the exact discrete Leibniz identity for finite differences (with its explicit remainder term) and use it as an accounting ledger: demanding that the remainder vanish in a packaged limit forces the derivation (product rule) structure. We also record an algebraic uniqueness anchor for derivations on polynomial rings in Lean/mathlib [2], [3].
- **Exhibit A-2 (protocol holonomy).** We quantify route mismatch between two natural refinement protocols under a small coordinate change, treating the mismatch as a **P3** holonomy-style defect that decays with refinement.
- **Exhibit B-1 (prime closure diagnostics).** We compare staged additive and multiplicative closures (Dirichlet truncations versus Euler-product truncations) under prototype packaging maps, and we separate a convergence control regime from the critical strip where naive staging is unstable.
- **Exhibit B-2 (passivity toy).** We show, in a self-dual toy family with a positivity constraint, that tightening feasibility can confine zeros to a symmetry locus—a pattern that motivates (but does not prove) "passivity ledger" interpretations of zero confinement.

A key implementation choice is that reported numbers and tables are not hand-copied: the TeX paper inputs generated tables and macros produced from snapshot-visible "last run" pointers in `notes/`. This makes the draft internally consistent with the repository state.

Code availability. The math-instantiation repository is available at:

- `https://github.com/ioannist/six-birds-mathematics`

1.4. Roadmap

Section 2 recalls the SBTdictionary in the minimal form needed for this instantiation. Section 3 formulates the discrete-to-packaged "closure engine" for mathematics. Sections 4–7 present the four exhibits: A-1 (calculus closure), A-2 (protocol holonomy), B-1 (prime closure diagnostics), and B-2 (passivity toy). Section 8 documents the artifact pipeline and reproduction commands, and Section 9 discusses limitations and next steps.

2. SBTdictionary for mathematical closures

This section adapts the minimal working dictionary from the Six-Birds framework [1] into a mathematical setting. We use it as a naming convention for the rest of the paper: each "bird" names a role that must be handled explicitly when a discrete protocol is compressed into a stable macro-level law.

2.1. Six primitives as closure mechanics

We follow the framework convention that the six primitives are not "topics" but *roles* played by concrete objects in an instantiation. In mathematics, the relevant objects are typically families indexed by a refinement parameter (meshes, truncations, partitions), defect bounds (error terms, residual norms), quotient/completion maps, and coherence constraints (linearity, locality, invariance, compatibility with algebraic operations).

Remark 1 (Six primitives (P1–P6), in framework language)**.** *Fix a space $\mathcal{P}$ of protocol objects (discrete descriptions) equipped with a refinement chain, modeled as a family of equivalences $\sim_j$ indexed by a stage parameter j (or, in analysis notation, by a step size $h \downarrow 0$). Write $\Pi_j : \mathcal{P} \to \mathcal{P}/\sim_j$ for the quotient/packaging map. Then the six primitives may be read as follows:*

1. ***P5** (Packaging). Each equivalence $\sim_j$ yields a quotient/packaging map Π_j and induces an idempotent saturation on predicates/events via*

$cl_j(A) := \Pi_j^{-1}(\Pi_j(A))$.

This is a standard closure-operator construction in order/lattice settings; see e.g. [4]. *In analytic instantiations,* Π_j *is a limit/identification map ("equal up to* $o(1)$ *as* $h \to 0$*"); in algebraic instantiations it is a literal quotient by a congruence.*

2. ***P6*** *(Accounting). The refinement order on quotients induces a preorder and monotone defect/audit quantities (e.g.* $j \mapsto Def(j)$ *or* $h \mapsto Err(h)$*). In this mathematics instantiation, "accounting" is a feasibility ledger (error control), not a directionality certificate.*

1. ***P4*** *(Staging). The refinement chain generates the stage index* j *(or* h*). Nontrivial staging corresponds to strict refinement at some scale (a genuine increase in resolution rather than a reparameterization).*
2. ***P2*** *(Constraints). Feasible macrostates at depth* j *are those representable as images under* Π_j*. Concretely, this is where invariances (translation/rotation/scale), locality, linearity, and algebra-compatibility are imposed as coherence conditions on the packaged layer.*
3. ***P1*** *(Operator rewrite). Fix a micro-update or micro-operator* $F : \mathcal{P} \to \mathcal{P}$*. The induced macro-update* $F^\sharp([p]) := [F(p)]$ *on* $\mathcal{P}/\sim_j$ *is well-defined iff*

   $p \sim_j q \Rightarrow F(p) \sim_j F(q)$.

   When this fails, one must either refine the lens (change $\sim_j$*), modify/extend* F *(renormalize / rescale / rewrite it), or accept that no closed macro dynamics exists at depth* j*.*
4. ***P3*** *(Protocol / holonomy). When there exist two admissible closure/evolution routes with the same input/output type (e.g. "refine then apply" versus "apply then refine"), their discrepancy defines a holonomy-like diagnostic. In normed instantiations we measure this by a route-mismatch quantity* RM*: a number that should decay under refinement in stable regimes.*

2.2. "A mathematics is A theory": porting the theory-package schema

The framework packages an instantiation as a *theory package*: a carrier (micro) space, a lens of observation, a completion/packaging rule, and an audit/defect principle [1]. For the mathematics instantiation we use the same shape, with "audit" interpreted as a nonnegative defect ledger (error and mismatch diagnostics).

Definition 1 (Mathematics theory package)**.** *A mathematics theory package is a tuple*

$$\mathcal{T} = (Z, f, \Sigma_f, E, \mathcal{D})$$

where:

- *Z is a carrier of discrete protocol objects at a chosen stage (e.g. grid functions, truncated sums/products, finite compositions, Cauchy sequences of rationals).*
- *$f : Z \to X$ is a lens (a coarse description map). The induced definability structure Σ_f is the family of predicates on Z that are visible through f (equivalently: constant on fibers of f).*
- *$E : \mathcal{V} \to \mathcal{V}$ is a completion/packaging endomap on a chosen description space $\mathcal{V}$ associated to Z (often idempotent or approximately idempotent in a specified norm). The fixed points of E are the internally complete objects recognized by the theory at the chosen scale.*
- *$\mathcal{D}$ is a defect ledger: a designated family of nonnegative diagnostics (error bounds, residual norms, route mismatch statistics) together with a monotonicity principle under refinement/coarsening. In this mathematics instantiation, $\mathcal{D}$ records feasibility (stability under refinement), not a thermodynamic arrow-of-time.*

Remark 2 (Slogan)**.**

The title phrase "A mathematics is A theory" is used in the following technical sense: to introduce a stable mathematical layer is to specify (i) what discrete protocol objects are being staged, (ii) what identifications are permitted under refinement, (iii) what completion/packaging map is being used, and (iv) what defect ledger certifies that the packaged layer is coherent.

2.3. Notation used in this instantiation

We use $h > 0$ for a refinement parameter (mesh size / step size) and write $h \downarrow 0$ for infinite refinement; equivalently we use a depth index $j \to \infty$. Families of discrete objects are written with stage subscripts (e.g. A_h, F_h). A generic route mismatch diagnostic takes the form

$$RM(h) = \frac{\|R_1(h) - R_2(h)\|}{\|R_1(h)\| + \varepsilon},$$

where R_1, R_2 are two admissible routes and $\varepsilon > 0$ is a small stabilizer. In later exhibits, RM is instantiated as: (i) a coordinate-change mismatch for discrete derivatives, and (ii) a mismatch between two staged prime closures after prototype packaging.

3. A closure engine for mathematics

This section states the generic "closure engine" used implicitly throughout the exhibits. The goal is not to re-found analysis but to make explicit what is usually left tacit when passing from discrete protocols to packaged laws. We present the engine as a sequence of design commitments: substrate, staging, ledger,

packaging, induced operators, and route diagnostics. These correspond directly to the SBTprimitives of Section 2.

*3.1. Protocol objects and staging (**P4**)*

Fix a family of discrete protocol spaces $\{Z_h\}_{h>0}$ indexed by a refinement parameter h (mesh size, truncation cutoff, partition diameter, discretization step). A point $z_h \in Z_h$ is a *micro-description* at stage h. In typical instantiations there are refinement maps (or comparison maps)

$$\rho_{h\to h'} : Z_h \to Z_{h'}, h' > 0, h' < h,$$

that embed coarse objects into finer resolution (e.g. interpolation, refinement of partitions, extending a truncation). The abstract role of **P4** is: the paper treats $h \downarrow 0$ as the mechanism by which "infinite composition" becomes a limit question, and by which closure claims become testable.

*3.2. Defect ledgers and feasibility gates (**P6**)*

A closure claim is only meaningful relative to an *audit ledger*: a family of nonnegative diagnostics that quantify what is lost when compressing a protocol. In this mathematics instantiation, the ledger consists of error terms and mismatch quantities rather than thermodynamic arrows.

Concretely, we allow two kinds of **P6** diagnostics:

1. **Approximation error:** for a candidate macro quantity $Q_h(z_h)$, compare across refinement or against a reference $Q_\star$: $\|Q_h - Q_{h'}\|$ or $\|Q_h - Q_\star\|$.
2. **Coherence defect:** residuals that measure failure of an intended algebraic constraint, such as a Leibniz defect for product structure, or a commutator/route-mismatch defect for two protocols.

A feasibility gate is then a monotonicity requirement of the form:

$$Def(h') \le \theta Def(h) \quad \text{for refined-}h' < h\text{-and some-}\theta < 1,$$

or, more weakly, that $Def(h) \to 0$ as $h \downarrow 0$.

*3.3. Packaging as identification and completion (**P5**)*

Packaging turns a staged family into a stable object by identifying distinctions that vanish under refinement. There are multiple equivalent ways to express this; the engine uses the following working form.

A *packaging rule* is an equivalence relation $\sim$ on families $\{z_h\}$ (or on a cofinal sequence $h_j \downarrow 0$) such that

$$\{z_h\} \sim \{z'_h\} \quad \Longleftrightarrow \quad Def(h; z_h, z'_h) \to 0\text{-as-}h \downarrow 0,$$

for a designated defect ledger. The packaged object is the equivalence class $[\{z_h\}]$. This is the same pattern as Cauchy completion (reals from rationals), and the same pattern as identifying two difference quotients that differ by $o(1)$ in the refinement limit.

The point of making **P5** explicit is that many disagreements in "what is the correct limit object?" are disagreements about *which defect ledger is permitted to vanish*.

3.4. Constraints and induced macro-operators (P2, P1)

To qualify as a macro-level *law*, an operator must descend to the packaged layer. Let $F_h : Z_h \to Z_h$ be a discrete micro-operator (update rule, local stencil, truncation transform). A packaged operator $F^\sharp$ is well-defined on equivalence classes only if the following holds:

$$\{z_h\} \sim \{z'_h\} \Longrightarrow \{F_h(z_h)\} \sim \{F_h(z'_h)\}.$$

When this implication fails, one is forced into **P1** operator rewriting: modify the operator family (e.g. rescale by powers of h, add counterterms, change representation, or alter the comparison map) so that it becomes compatible with the packaging relation.

Separately, **P2** constraints are the coherence requirements imposed at the packaged layer. In this paper they appear in three forms:

- algebraic coherence (e.g. Leibniz/product compatibility for derivatives),
- symmetry/invariance constraints (e.g. self-dual symmetrization prototypes),
- locality/translation constraints in stencil experiments.

The closure engine is conservative: it treats these constraints as *requirements for feasibility*, not as axioms of nature.

3.5. Protocol mismatch and holonomy diagnostics (P3)

Whenever there are two reasonable routes that share input/output types, route mismatch is a computable defect. Abstractly, suppose two routes produce outputs $R_1(h)$ and $R_2(h)$ on a shared comparison space. We measure the protocol defect by a normalized mismatch

$$RM(h) = \frac{\|R_1(h) - R_2(h)\|}{\|R_1(h)\| + \varepsilon_0}.$$

A stable closure regime is indicated by decay of $RM(h)$ under refinement. Exhibit A-2 uses this to quantify noncommutativity under coordinate change; Exhibit B-1 uses it to compare additive and multiplicative staged prime closures.

A key guardrail (used throughout) is that route mismatch is a *diagnostic of closure feasibility*, not a certificate of directionality or irreversibility.

3.6. Closure ladders as repeated packaging

Finally, the engine composes: once a packaged layer is stable, it becomes a new substrate for further refinement and packaging. This is the SBT“closure ladder” viewpoint: reals arise by completion of rationals; calculus arises by completion of discrete difference/sum protocols under coherence constraints; geometric obstructions appear as protocol noncommutativity when patching local descriptions. The exhibits should be read as small, auditable instances of this repeated pattern rather than as a claim that one mechanism replaces all mathematical practice.

4. Exhibit A-1: calculus as a stable closure

This exhibit instantiates the SBTclosure lens in the most familiar mathematical setting: how the differential and integral calculus appear as the stable closure of discrete protocols under refinement. Our emphasis is not on definitions but on *feasibility*: which compressions remain coherent as the stage parameter refines.

4.1. Discrete substrate and staging

Fix a step size $h > 0$ and consider functions on a one-dimensional grid (or, more generally, any set with a notion of translation by h). The discrete substrate supplies:

- a *shift* operator T_h defined by $(T_h f)(x) = f(x + h)$;
- a *difference* operator $\Delta_h := T_h - I$;

the *scaled difference* (difference quotient)

$$\delta_h f(x) := \frac{f(x + h) - f(x)}{h},$$

which is the simplest instance of **P1** operator rewriting: the h^{-1} rescaling that keeps the update comparable across stages.

In SBTterms, $h \downarrow 0$ is **P4** staging. Any proposed macro-law ("derivative", "generator", "flow") is a packaged object only if its induced defect ledger remains feasible under refinement (**P6** accounting), and only if it respects the coherence constraints imposed at the packaged layer (**P2** constraints).

4.2. The accounting ledger: exact discrete Leibniz identity

At the discrete level, the difference quotient is not a derivation: multiplicative coherence fails by a remainder term that can be written *exactly*. This remainder is the canonical **P6** ledger term for the passage from differences to derivatives.

Lemma 1 (Finite-difference Leibniz identity).

Let F be a field, let $h \in F$ with $h \neq 0$, and let $f, g : F \to F$. Define $\delta_h f(x) := (f(x+h) - f(x))/h$. Then for all $x \in F$,

$$\delta_h(fg)(x) = \delta_h f(x) g(x) + f(x)\delta_h g(x) + h\delta_h f(x)\delta_h g(x).$$

Remark 3 (Machine-checked anchor). *Lemma 1 is formalized in Lean/mathlib in this repository as* `Diff.delta_mul_leibniz` *(file* `lean/Diff/FiniteDifference.lean`*)* [2][3].

The key point for closure is not the identity itself but the structure of the ledger term:

$$\underbrace{h\delta_h f \delta_h g}_{\text{ledger term}}$$

is explicitly proportional to the stage parameter h. Thus, in any refinement regime where $\delta_h f$ and $\delta_h g$ remain bounded (a feasibility condition measured by **P6**), the ledger term vanishes as $h \downarrow 0$ and multiplicative coherence emerges at the packaged layer.

4.3. Packaging: quotient by vanishing defect yields derivations

The previous subsection suggests a general closure template. A discrete operator family L_h (or a protocol that composes many micro-steps) is admissible as a "law" in the packaged layer only after specifying:

1. a staging parameter h (**P4**);
2. a ledger that tracks closure defects, such as Leibniz residuals, consistency across refinement, and stability bounds (**P6**);

3. an identification rule “equal up to vanishing defect as $h \downarrow 0$” that packages families into an equivalence class (**P5**).

Demanding multiplicative coherence is a **P2** constraint. For difference quotients it is satisfied only *approximately* at finite h and becomes exact in the packaged limit because the ledger term in Lemma 1 vanishes. The packaged limit of feasible difference-quotient families therefore satisfies the Leibniz rule:

$$D(fg) = (Df)g + f(Dg),$$

i.e. it is a *derivation*. In one dimension, further invariance/normalization choices single out the usual derivative, but the closure point is more general: *derivations are what survives after enforcing multiplicative coherence under refinement.*

4.4. Algebraic uniqueness anchor on polynomials

The previous discussion is analytic and asymptotic. To anchor the “derivation as closure fixed point” motif in a fully algebraic setting, we use polynomial rings where derivations have a clean universal property.

Theorem 1 (Derivations on $R[X]$ are determined by $D(X)$)**.** *Let R be a commutative semiring and let $A := R[X]$ be the polynomial ring. An R-derivation $D : Derivation(R, A, A)$ is uniquely determined by its value on the indeterminate $X \in A$. Equivalently, evaluation on X induces an R-linear equivalence*

$$Derivation(R, R[X], R[X]) \simeq_{\ell[R]} R[X], D \longmapsto D(X),$$

with inverse given (schematically) by “multiply the formal derivative by a polynomial”.

Remark 4 (Machine-checked anchors)**.** *Theorem 1 is formalized in Lean/mathlib in this repository as:* `Derivations.derivation_ext_X` *(uniqueness) and* `Derivations.polynomialDerivationEquiv` *(explicit R-linear equivalence), in file* `lean/Derivations/Polynomial.lean`*. The Lean type of the equivalence is* `DerivationR(PolynomialR)(PolynomialR)` $\simeq_{\ell[R]}$ `PolynomialR`*. The evaluation-at-X interface is exposed by* `Derivations.polynomialDerivationEquiv_apply_X` *and* `Derivations.polynomialDerivationEquiv_symm_apply_X`*. These are re-exported/instantiated from mathlib with small glue lemmas.*[2][3]

Classically, this computation is expressed via Kähler differentials, e.g. $\Omega_{R[X]/R} \cong R[X] \cdot dX$; see [5].

For the purposes of this paper, the takeaway is structural: once the Leibniz rule is imposed as a coherence constraint, the space of admissible operators collapses to a small, rigid family. In polynomial land this

rigidity is literal and exact (determined by $D(X)$); in analytic settings it is mediated by the defect ledger and the chosen packaging rule.

4.5. Falsification-first diagnostics: stencil selection under feasibility

To complement the algebraic anchors, we implemented a numerical experiment that samples a population of candidate operators and filters them by feasibility gates. We sample local linear stencil operators and apply gates that instantiate **P4**–**P6**: stability under refinement, decreasing route mismatch, and (optionally) decreasing multiplicative (Leibniz) defect.

4.6. Numerical protocol details (stencils)

We emphasize that the stencil experiments are *diagnostics*, not a uniqueness theorem. A candidate local operator is a clipped-boundary stencil of half-width m,

$$(Lf)_i := \sum_{j=-m}^{m} c_j f_{i+j}, i = m, \dots, N-m-1,$$

evaluated only on interior points to avoid boundary-condition artifacts. To compare across refinements $N, 2N, 4N$ (step sizes $h, h/2, h/4$), we align outputs by restricting the finer-grid interior outputs to even-index centers. The refinement-consistency defects E_{01}, E_{12} used in the feasibility gate are relative ℓ^2 mismatches on this aligned interior, maximized over a fixed smooth test family (polynomials plus $\sin$ and $\exp$).

The "order" label $k \in \{0, 1, 2\}$ is chosen by best-fit against analytic templates on the finest grid: after the renormalization h^{-k} (an instance of **P1** operator rewriting), we select the k whose output is closest (up to an overall scalar) to f, f', or f'' on the test family.

In the recorded runs, the baseline grid size is $N_0 = 64$ and the Leibniz-gated experiment uses half-width $m = 3$ with gate thresholds $D_2 < 0.2$ and ratio gate $gate\_ratio = 0.85$. For standard background on finite-difference stencils, consistency, and stability, see [6].

*Disclosure of the **P1** step in the Leibniz-gated run*

In the Leibniz-gated experiment we additionally apply a moment-normalization step to candidate coefficient vectors (c_j) before auditing them. Concretely, we project to a "first-order" class by enforcing the discrete moment constraints $\sum_j c_j = 0$ and $\sum_j j c_j = 1$, and we optionally suppress low-order contamination by setting higher moments (e.g. $\sum_j j^2 c_j$, $\sum_j j^3 c_j$) near zero. In SBTterms this is not the

conclusion but the *operator rewrite* (**P1**) that makes families comparable across scale and avoids trivial $k = 0$ closures. The empirical point of Table 2 is therefore conditional: *within the normalized first-order class, imposing a shrinking Leibniz defect selects derivative-like closures.*

Two regimes illustrate the role of constraints.

Stability-only filtering

If we impose only refinement stability/consistency without demanding multiplicative coherence, then the surviving operators overwhelmingly behave as order-0 closures (approximately scalar multiplication). In the current run, 913 out of 1000 sampled stencils survive under the stability gate, and the survivors are classified as best-fit order $k = 0$ (Table 1).

Leibniz-gated filtering

If we additionally demand that the Leibniz defect improves with refinement (a direct numerical analog of Lemma 1 having a vanishing ledger term), then the surviving operators collapse to derivative-like (order-1) closures. In the current run, the Leibniz-gated filter produces 50 survivors and all survivors are classified as best-fit order $k = 1$ (Table 2). Figure 1 visualizes the selection outcome for the recorded run.

These experiments do not prove uniqueness of the derivative in full generality. They serve as *diagnostics* aligned with the SBTprogram: when a closure is feasible, defects shrink under refinement; when it is not, defects persist or blow up. Here, enforcing multiplicative coherence is the decisive feasibility pressure that selects derivation-like operators from a large population of local discrete rules.

generated	survivors	best-fit counts
1000	913	k=0: 913; k=1: 0; k=2: 0

Table 1. Stencil flow stability-only summary

survivors	best-fit counts
50	k=0: 0; k=1: 50; k=2: 0

Table 2. Leibniz-gated stencil selection

rank	D2	F1	$E_{12,1}$
1	1.68144×10^{-6}	7.19072×10^{-7}	5.92487×10^{-6}
2	1.8767×10^{-6}	8.04778×10^{-7}	6.6106×10^{-6}
3	2.05543×10^{-6}	8.80601×10^{-7}	7.24186×10^{-6}
4	2.19982×10^{-6}	9.44249×10^{-7}	7.74517×10^{-6}
5	2.20896×10^{-6}	9.45855×10^{-7}	7.77949×10^{-6}
6	2.24716×10^{-6}	7.73372×10^{-7}	6.41209×10^{-6}
7	2.92045×10^{-6}	1.66745×10^{-7}	1.404×10^{-6}
8	3.00861×10^{-6}	1.29407×10^{-6}	1.05915×10^{-5}
9	3.11535×10^{-6}	8.37018×10^{-7}	6.81121×10^{-6}
10	3.8798×10^{-6}	3.59399×10^{-7}	3.16225×10^{-6}

Table 3. Leibniz-gated top survivors (diagnostics)

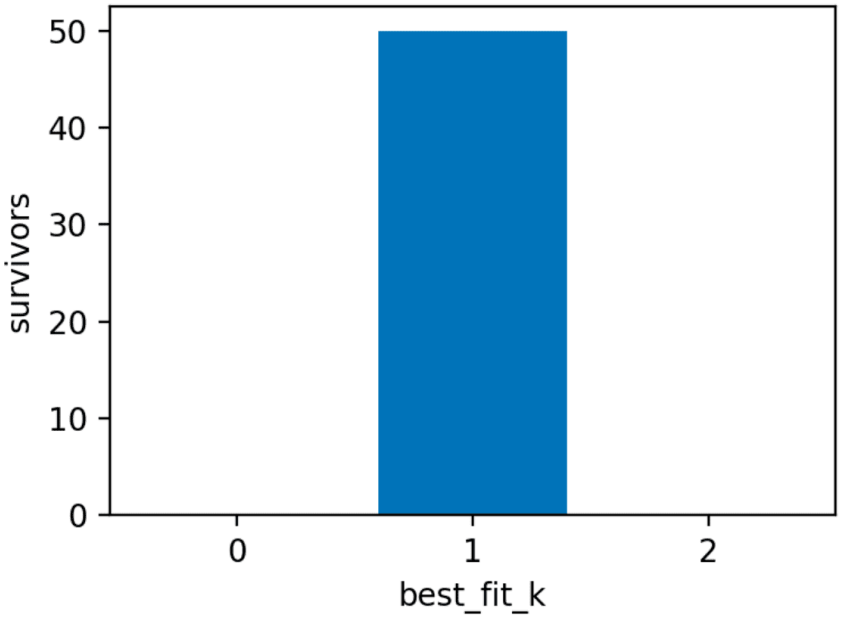

Figure 1. Leibniz-gated stencil selection under refinement. The gate filters local stencil operators by requiring a shrinking Leibniz defect under refinement, producing a concentrated set of derivative-like closures (see Table 2).

4.7. Integration as a packaged closure of sums

The same closure lens applies to integration: cumulative sums are staged at grid size h, audited by differentiation and protocol mismatch defects, and accepted as packaged closures when those defects shrink under refinement. In the recorded run, the left-versus-trapezoid route mismatch at the finest grid is 0.00177129, with fitted scaling exponent 1.00009 and fit quality 1. The fundamental-theorem defect for the trapezoid cumulative operator decays with fitted exponent 0.499955. (Here the reported norms are unweighted discrete ℓ^2 norms over grid points, so pointwise $O(h)$ defects appear as $O(h^{1/2})$ in ℓ^2.)

doi.org/10.32388/JABWNG

N	h	RM	FT left	FT trap
128	0.0078125	0.0141732	5.65355×10^{-14}	0.19633
256	0.00390625	0.00708551	1.62236×10^{-13}	0.138837
512	0.00195312	0.00354262	4.75038×10^{-13}	0.0981742
1024	0.976562×10^{-3}	0.00177129	1.2982×10^{-12}	0.0694199

Table 4. Integration closure diagnostics under refinement

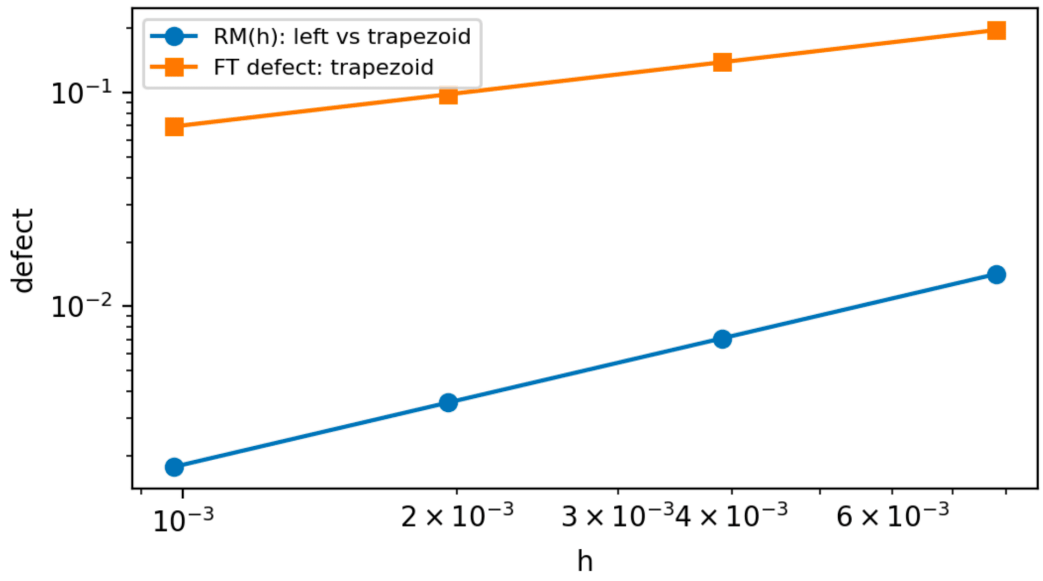

Figure 2: Integration closure diagnostics on staged grids. The route mismatch between left and trapezoid cumulative protocols and the differentiation-based defect both decay under refinement, consistent with stable packaged closure.

5. Exhibit A-2: protocol holonomy and route mismatch

This exhibit instantiates **P3**(protocol / holonomy) as a quantitative diagnostic rather than a geometric claim. The premise is simple: when there are multiple natural routes for constructing a packaged object

from staged data, the routes need not commute at finite stage. The discrepancy between routes is a defect; a stable closure should make that defect shrink under refinement.

5.1. Two routes under coordinate change

We consider a small coordinate change (a "chart" perturbation)

$$\varphi(x) = x + \varepsilon x^2, \varepsilon\text{small},$$

and study a discrete derivative approximation at step size h.

There are (at least) two natural routes:

1. **Route A (refine then differentiate then pull back).** Refine the sampled data, apply the discrete derivative in the target coordinate, then interpolate/pull back the result along φ.
2. **Route B (pull back then refine then differentiate).** Pull back the sampled data along φ at coarse scale, refine in the source coordinate, apply the discrete derivative, and convert by the chain rule factor $(\varphi')^{-1}$.

At finite h, these routes generally disagree because interpolation, refinement, and discrete differentiation do not strictly commute.

5.2. Route mismatch as a defect ledger

In the SBTdictionary, the existence of multiple admissible routes is **P3**, while the numerical discrepancy between them is tracked by **P6** accounting. We quantify the mismatch by a normalized route-mismatch statistic

$$RM(h) := \frac{\|R_A(h) - R_B(h)\|}{\|R_A(h)\| + \varepsilon_0}, \tag{1}$$

where $R_A(h), R_B(h)$ are the route outputs on a shared comparison grid, $\|\cdot\|$ is an ℓ^2 norm over evaluation points, and $\varepsilon_0 > 0$ is a small stabilizer.

A stable closure regime is indicated by *decay* of $RM(h)$ as $h \downarrow 0$. Conversely, persistent or growing mismatch is evidence that the proposed packaging (or protocol family) is not yet feasible at the tested scales.

5.3. Observed scaling under refinement

In the recorded run, $RM(h)$ decreases with refinement and is well fit by a power law

$$RM(h) \approx C h^{p}.$$

The fitted exponent is

$$p = \ 1.46381,$$

where 1.46381 is imported from the repository artifacts (generated TeX macros). Table 5 lists the measured values and Figure 3 shows the decay trend.

As a null calibration, setting $\varepsilon = 0$ (identity coordinate map) yields route mismatch at or below 0 on the tested grids, confirming that the reported decay is not an interpolation bookkeeping artifact. The log–log slope summary is reported together with the fit quality $R^2 = 0.987808$ and should be interpreted as a coarse scaling diagnostic rather than a precise convergence order.

h	**RM**
0.015625	0.00476218
0.0078125	0.00132002
0.00390625	4.9099×10^{-4}
0.00195312	2.2498×10^{-4}

Table 5. Holonomy RM vs h

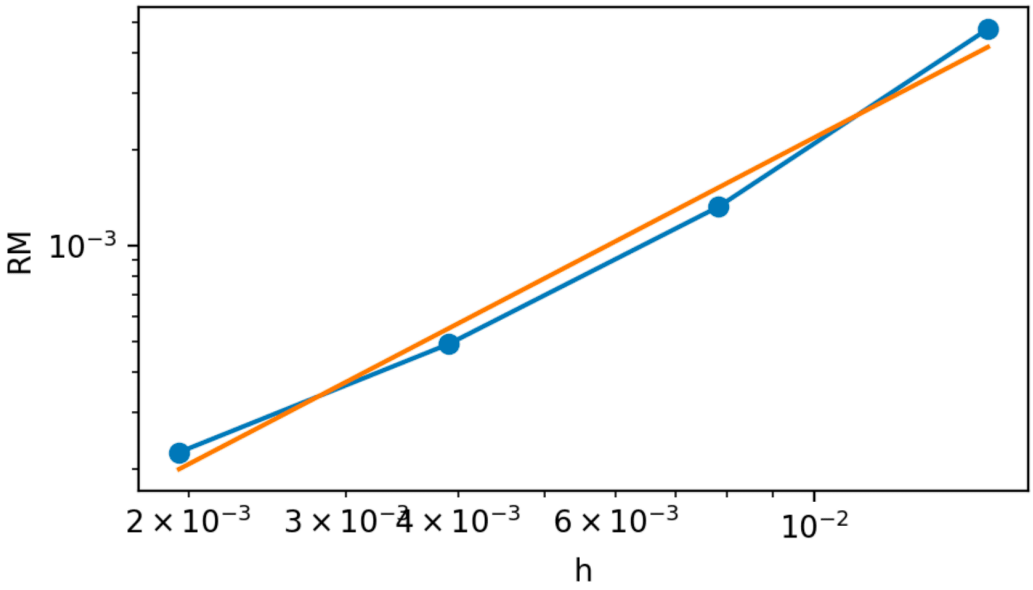

Figure 3. Route mismatch under coordinate change: $RM(h)$ versus step size h for two noncommuting refinement routes (Equation 1). The mismatch decays under refinement; the fitted exponent reported in the text is sourced from the generated macro 1.46381.

5.4. Guardrail: holonomy is not directionality

A crucial framework guardrail is that *holonomy does not by itself certify directionality*. In this instantiation, $RM(h)$ is a measure of *route dependence* (a **P3** phenomenon) and its decay is a *feasibility signal* (tracked by **P6**). None of this implies an "arrow of time" or a monotone irreversible ordering on states. It only says: given a staged protocol family and a proposed packaging rule, the protocol routes become increasingly compatible as resolution increases.

In later mathematical instantiations (e.g. differential geometry), true holonomy is tied to curvature and obstruction-to-patching phenomena. Here we remain at the diagnostic level: route mismatch is a computable defect that should shrink in stable regimes, and can therefore be used to stress-test closure claims.

6. Exhibit B-1: prime closure diagnostics

This exhibit applies the SBTclosure lens to a canonical "discrete $\rightarrow$ spectral" packaging problem: compressing the multiplicative micro-structure of the integers (primes) into a continuous analytic object.

We do *not* attempt to prove theorems about ζ. Instead we build falsification-first diagnostics that separate: (i) a convergence control regime where the micro-descriptions are valid, from (ii) the critical strip where naive staging is unstable and protocol mismatch blows up.

6.1. Two staged micro-descriptions

Fix $s \in \mathbb{C}$. The classical zeta function has two micro-descriptions in a common domain (and, in the infinite limit, they agree there):

$$\zeta(s) = \sum_{n \geq 1} n^{-s} \qquad \text{and} \qquad \zeta(s) = \prod_{p} \frac{1}{1 - p^{-s}}, (\Re(s) > 1).$$

These classical representations and their domains are standard; see e.g. [7]. At finite stage we replace these by staged truncations. In the recorded diagnostics we use a mild exponential smoothing weight $w(u) = e^{-u}$ and define:

$$\begin{aligned} S_N(s) &:= \sum_{n=1}^{N} w(n/N) n^{-s}, \\ P_N(s) &:= \exp\left(\sum_{p \leq N} w(p/N) \left(-\log\left(1 - p^{-s}\right) \right) \right), \end{aligned}$$

where the Euler-side expression is computed via a weighted log-sum for numerical stability. Note that the smoothing is applied at the level of the log-sum: equivalently,

$$P_N(s) = \prod_{p \leq N} \left(1 - p^{-s}\right)^{-w(p/N)}.$$

We use this only to reduce sharp cutoff artifacts; when $w \equiv 1$ this reduces to the usual sharp partial Euler product. The key SBTpoint is conceptual: S_N is an *additive* staging protocol and P_N is a *multiplicative* staging protocol. A stable closure would require these protocols to become compatible under refinement **P4** and under the chosen packaging **P5**, as audited by an accounting ledger **P6**.

6.2. Prototype packaging maps: raw, completed, symmetrized

To compare staged objects across regions, we apply simple prototype packaging maps. Let

$$C(s) := \frac{1}{2} s(s-1) \pi^{-s/2} \Gamma(s/2).$$

Given a staged function $F(s)$, define:

$$
\begin{aligned}
Raw(F)(s) &:= F(s),\\
Comp(F)(s) &:= C(s)F(s),\\
Sym(F)(s) &:= \frac{1}{2}(Comp(F)(s) + Comp(F)(1-s)).
\end{aligned}
$$

Here $Comp$ is a one-sided "completion prototype", and Sym enforces the self-dual symmetry $s \leftrightarrow 1-s$ by construction. We emphasize that this is a *diagnostic* packaging, not an analytic continuation scheme.

In the recorded run we choose:

- **Convergence region:** $Comp$ (one-sided completion), because it does not force evaluation at $1-s$ far outside the domain where these truncations represent ζ (for $\Re(s) > 1$ one has $\Re(1-s) < 0$).
- **Critical strip:** Sym (symmetrized completion), as a prototype of "self-dual packaging".

6.3. Diagnostics: route mismatch and error ledger

Let $Pack$ denote the chosen packaging mode in a given regime. On a fixed finite test set K (a small grid of points in the region of interest), we evaluate the packaged staged functions. In the recorded run, K is a grid with $\sigma \in \{1.25, 1.5\}$ and $t \in \{0, 2, 4, 6\}$ in the convergence regime, and $\sigma \in \{0.75, 0.5\}$ and $t \in \{0, 2, 4, 6\}$ in the strip; refinements use $N \in \{50, 100, 200, 400, 800\}$.

$$
A_N(s) := Pack(S_N)(s), B_N(s) := Pack(P_N)(s).
$$

We quantify protocol mismatch by a normalized route mismatch:

$$
RM_2(N) := \frac{\left(\sum_{s\in K} |A_N(s) - B_N(s)|^2\right)^{1/2}}{\left(\sum_{s\in K} |A_N(s)|^2\right)^{1/2} + \varepsilon_0},
$$

with a small stabilizer $\varepsilon_0 > 0$.

All reported values in this exhibit are computed at working precision 50 digits (via `mpmath`).

Separately, we compute an **P6** error ledger by comparing each packaged truncation to a reference value computed by `mpmath.zeta` under the same packaging mode:

$$
Err_S(N) \sim \frac{\|A_N - Pack(\zeta)\|}{\|Pack(\zeta)\| + \varepsilon_0}, Err_P(N) \sim \frac{\|B_N - Pack(\zeta)\|}{\|Pack(\zeta)\| + \varepsilon_0}.
$$

In the convergence control region this is a genuine accuracy test. In the critical strip it is a stress test: it measures how badly naive staging fails to track the analytically continued object.

6.4. Control regime: $\Re(s) > 1$

In the convergence control regime (where both series and product are valid in the infinite limit), the diagnostics behave sensibly: both the mismatch $RM_2(N)$ and the errors decrease with N on the recorded test set. Table 6 shows $RM_2(N)$ together with the two error ledgers. In particular, the route mismatch decreases to $RM_2(800) = 0.0523991$, consistent with staged compatibility improving under refinement when the micro-descriptions are valid.

N	RM2	errS2	errP2
50	0.0759745	0.209083	0.146987
100	0.0697803	0.171755	0.111871
200	0.0667314	0.142319	0.084813
400	0.0613035	0.118486	0.0654121
800	0.0523991	0.0984851	0.0505677

Table 6. Prime RM in convergence region

6.5. Critical strip: protocol mismatch under invalid staging

In the critical strip ($0 < \Re(s) < 1$), the naive truncations S_N and especially P_N are *not* valid micro-descriptions of the analytically continued ζ. Accordingly, the diagnostics indicate a non-feasible closure: the route mismatch and errors grow rapidly with N on the recorded test set. Table 7 shows that the mismatch reaches $RM_2(800) = 5.00808 \times 10^{12}$, and the Euler-side error ledger becomes enormous.

N	RM2	errS2	errP2
50	340.421	6.59171	2123.75
100	5996.54	10.4329	6.02608×10^{4}
200	6.19843×10^{5}	16.5012	0.984885×10^{7}
400	3.14454×10^{8}	27.1202	8.29135×10^{9}
800	5.00808×10^{12}	43.7077	2.16627×10^{14}

Table 7. Prime RM in the critical strip

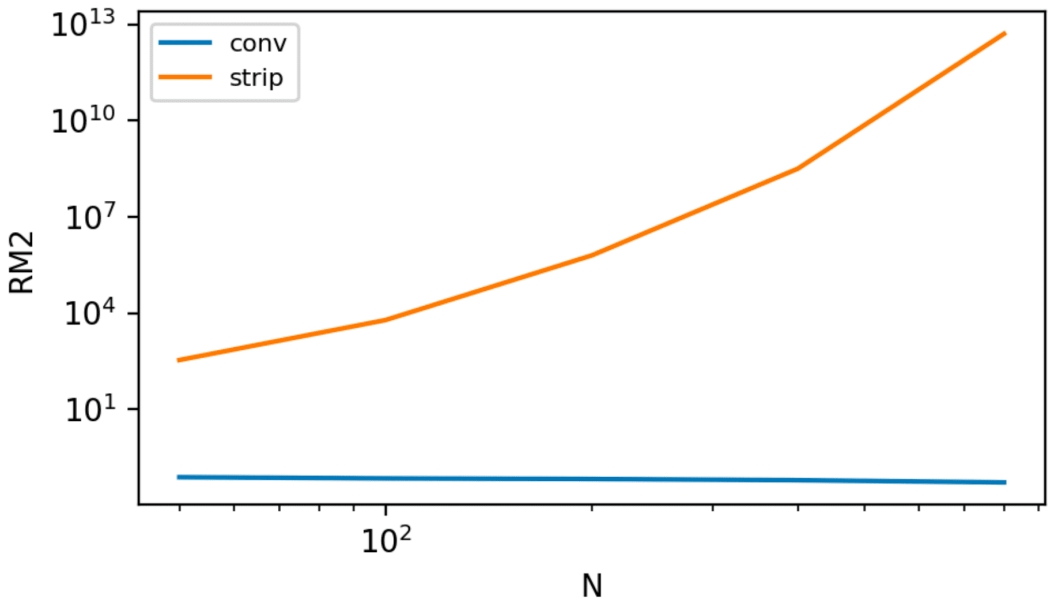

Figure 4. Prime closure route mismatch diagnostic. The same staged additive and multiplicative protocols behave compatibly in a convergence control regime (Re(s)>1), but mismatch grows by many orders of magnitude in the critical strip under naive staging and prototype self-dual packaging. This is interpreted as a feasibility failure of the staging/packaging pair, not as a theorem about ζ.

doi.org/10.32388/JABWNG

6.6. Interpretation: a diagnostic, not a theorem about ζ

This exhibit is deliberately modest. It does not claim that Sym is "the right" packaging, nor that the observed blow-up has direct implications for the zero set of ζ. Instead it instantiates the SBTlogic:

- **Multiple routes exist** (additive vs multiplicative staging), hence **P3** applies.
- **Feasibility must be audited** by a ledger (**P6**): mismatch and error should shrink in stable regimes.
- **Staging matters** (**P4**): outside the domain of convergence, naive truncations are not compatible with the packaged object one hopes to define.

The positive outcome is methodological: the same route-mismatch machinery used for discrete derivatives (Exhibits A-1 and A-2) can be used to stress-test prime-to-spectral closure claims. A stable analytic continuation scheme—if expressed as a refinement-fixed packaging with an appropriate positivity/passivity ledger—should manifest as decreasing mismatch under refinement on suitable diagnostics. Here we only establish the baseline: without additional feasibility structure, the mismatch blows up exactly where classical analytic number theory says naive micro-descriptions fail.

7. Exhibit B-2: passivity toy (positivity ledger and zero confinement)

A recurring SBTmotif is that the most rigidifying feasibility conditions are often *positivity* conditions. In physics, passivity/positivity constraints strongly restrict the analytic structure of response functions [8]. In mathematics, many "stability" and "no-gain" requirements appear as nonnegativity of quadratic forms, positive-definite kernels, or positivity of coefficients in generating functions. This exhibit provides a deliberately simple toy instantiation of that motif: as a positivity constraint is tightened, zeros (defects) are increasingly confined to a symmetry locus.

7.1. Self-duality and a symmetry locus

We consider a family of polynomials

$$Z_\lambda(z) = \sum_{k=0}^{n} a_k(\lambda) z^k$$

constructed from a finite Ising-style partition sum (implemented by direct enumeration for $n = 12$ in the recorded run). A key structural feature is a built-in self-duality symmetry: we enforce palindromic coefficients by symmetrization,

$$a_k(\lambda) \leftarrow \frac{1}{2}(a_k(\lambda) + a_{n-k}(\lambda)),$$

which makes Z_λ invariant (up to the trivial z^n factor) under inversion $z \leftrightarrow 1/z$. Consequently, zeros occur in reciprocal pairs. The fixed locus of this symmetry is the unit circle $|z| = 1$, which we treat as the *symmetry axis* (or symmetry manifold) for the toy.

In SBTlanguage, inversion symmetry is a **P2** constraint, while palindromization is a **P1** operator rewrite that makes the constraint manifest at the packaged level.

7.2. Positivity as a feasibility ledger

The toy feasibility constraint is a one-parameter "positivity tightening" of couplings. We sample a set of edge couplings J_e (both signs), and define a clipped family

$$J_e(\lambda) = \max(J_e, 0) + \lambda \min(J_e, 0), \lambda \in [0, 1].$$

Thus $\lambda = 1$ is the original mixed-sign instance, while $\lambda = 0$ is a purely nonnegative (ferromagnetic) instance. This is a minimal model of a passivity/positivity ledger: tightening $\lambda \downarrow 0$ removes "gain-like" sign structure.

7.3. Defects as zeros and a confinement metric

We compute all complex zeros of Z_λ and quantify confinement to the symmetry locus by the radial deviation

$$d_i(\lambda) := ||z_i(\lambda)| - 1|,$$

summarized by

$$mean\_dev(\lambda) = \frac{1}{n}\sum_i d_i(\lambda), \max\_dev(\lambda) = \max_i d_i(\lambda).$$

These play the role of **P6** accounting diagnostics: they measure how strongly the defect set (zeros) respects the symmetry locus.

Table 8 reports the confinement metrics across λ. In the recorded run, tightening positivity produces a dramatic confinement effect:

$$mean\_dev(1) \approx 0.871694, mean\_dev(0) \approx 3.0292 \times 10^{-4},$$

so the feasible ($\lambda \to 0$) regime concentrates zeros extremely close to $|z| = 1$.

λ	mean_dev	max_dev
1	0.871694	7.10509
0.7	0.580068	4.26786
0.4	0.349525	2.26657
0.2	0.200513	1.21606
0.1	0.0946892	0.718256
0	3.0292×10^{-4}	9.0919×10^{-4}

Table 8. Passivity toy zero-confinement metrics

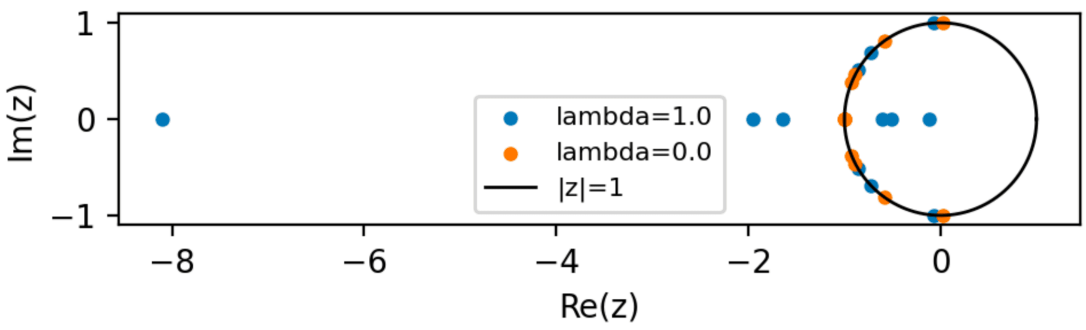

Figure 5. Passivity toy: zeros of a self-dual polynomial family under positivity tightening. The symmetry locus is the unit circle $|z| = 1$. As the clipping parameter λ decreases toward the nonnegative (ferromagnetic) regime, the zeros concentrate near the symmetry locus; Table 8 quantifies this via radial deviation metrics.

7.4. Interpretation and scope

This toy is not a theorem and does not claim to reproduce the full hypotheses of classical zero-confinement results. Its role is methodological: it demonstrates a recognizable SBTpattern in a controlled setting. This motif has rigorous instances, most famously the Lee–Yang circle theorem for ferromagnetic Ising models [9]; we do not invoke that theorem here.

- Symmetry (**P2**) specifies a distinguished locus (here $|z| = 1$).

- Positivity/passivity acts as a feasibility constraint measured by an audit ledger (**P6**).
- Under tighter feasibility, defects (zeros) become increasingly confined to the symmetry locus.

The purpose of including this exhibit in a mathematics instantiation paper is to motivate (without overclaiming) why problems like "zero confinement to a self-dual manifold" naturally invite passivity/positivity formulations. Exhibit B-1 on prime closures shows that naive staging and packaging can fail catastrophically in the critical strip; this toy suggests what a successful closure might require in addition: a feasibility ledger strong enough to prevent defect drift away from symmetry.

8. Reproducibility and artifact contract

This draft is designed so that reported numbers are not hand-copied. Instead, tables and key numeric macros in the TeX source are generated from snapshot-visible pointer artifacts in `notes/`. This section documents the minimal workflow and the "artifact contract" that links the PDF to repository state [10].

8.1. Quickstart: sanity checks

From the repository root, the recommended first command is:

```
bash scripts/check_all.sh
```

This script runs a lightweight pipeline: Python import sanity checks, `pytest`, a Lean build, regeneration of the TeX-label index, regeneration of the results dashboard, and byte-compilation checks for the main experiment scripts. It is intended as a "freeze check" before edits and before releases.

8.2. How tables and macros are generated

The TeX paper reads generated material from:

```
tex/math_instantiation/generated/
```

These generated files are produced by:

```
python scripts/export_results_tex.py
```

The exporter reads snapshot-visible "last run" pointers (JSON) in `notes/` and writes: (i) numeric macros (e.g. `HolonomyExponent`) and (ii) LaTeX tables for each exhibit. This makes the paper "honest by construction": if the pointers change, the macros/tables change.

After regenerating, rebuild the PDF with:

```
bash scripts/build_math_paper.sh
```

8.3. Figures and the "last run" pointers

Each experiment script writes two kinds of artifacts:

1. a timestamped run record under `data/runs/` (not committed by default), and
2. a snapshot-visible pointer file under `notes/` that records the most recent run path and the canonical figure paths.

The pointer files used by the TeX exporter in this draft are:

`notes/stencil_flow_last_run.json` `notes/stencil_flow_leibniz_last_run.json` `notes/integration_closure_last_run.json` `notes/holonomy_rm_last_run.json` `notes/prime_closure_rm_last_run.json` `notes/passivity_toy_last_run.json`

and each pointer includes (at minimum) a timestamp, an `output_path` into `data/runs/`, and figure paths under `figures/`. For TeX compatibility, each exhibit figure is committed in both SVG (inspection) and PNG (pdflatex inclusion) form.

To regenerate the committed "last run" figures (and refresh the pointers), run the corresponding experiment scripts, e.g.:

`python experiments/stencil_flow/leibniz_gate.py` `python experiments/integration_closure/run.py` `python experiments/holonomy_rm/run.py` `python experiments/prime_closure_rm/run.py` `python experiments/passivity_toy/run.py`

8.4. Dashboard regeneration

The file `notes/results_dashboard.md` is a snapshot-visible summary of the current pointer state. It can be regenerated at any time via:

```
python scripts/make_dashboard.py
```

The dashboard is intended for writing efficiency: it lists the latest tables, macro values, and figure paths that will appear in the TeX build.

8.5. Minimal artifact contract

The artifact contract for this draft is:

Remark 5 (Artifact contract for reported numbers)**.** *All numeric values appearing in (i) the generated tables under* `tex/math_instantiation/generated/` *and (ii) the macros imported from* `generated/macros.tex` *are generated from the snapshot-visible pointer JSON files in* `notes/` *listed above, by the script* `scripts/export_results_tex.py`*. Thus, the "meaning" of a reported number is: "the value recorded in the latest pointer JSON at the time the TeX build was produced," rather than an untracked manual transcription.*

This contract does not guarantee that the experiments are deterministic across platforms (though seeds are recorded in run artifacts); it guarantees that the paper and repository state are internally consistent.

9. Discussion: scope, limitations, and next steps

9.1. What this draft establishes

This paper contributes a mathematics instantiation of the SBTdictionary of [1] with three deliverables:

- a *formal vocabulary* for mathematics-as-closure (Section 2 and Section 3),
- *machine-checked anchors* (Lean) and falsification-first diagnostics (Python) that can fail,
- an *artifact contract* that ties reported numbers in the PDF to snapshot-visible pointers in the repository (Section 8).

The exhibits are intended to be small but sharp: they illustrate how SBT-style feasibility pressure (staging + ledger + packaging + constraints) can be expressed in ordinary mathematical workflows.

9.2. Limitations and guardrails

Several limitations are deliberate and should be understood as guardrails rather than omissions.

Diagnostics are not theorems

The Python experiments are designed as feasibility stress tests, not as proofs of uniqueness. For example, the stencil-selection results support the claim that adding a Leibniz-defect gate selects derivative-like operators under the tested conditions, but they do not rule out exotic behaviors outside those conditions.

Prime closure results are baseline, not analytic continuation

Exhibit B-1 separates a convergence control regime from the critical strip and reports catastrophic mismatch for naive staging in the strip. This is consistent with classical domain-of-convergence facts; it

does not establish new results about ζ and should not be read as evidence for conjectures about its zeros.

Positivity toy is a motif demonstration

The passivity toy illustrates a pattern—tightening a positivity constraint can confine zeros to a symmetry locus—but it does not claim generality. Its role is motivational: it shows why "zero confinement to a self-dual manifold" naturally invites positivity/passivity ledgers.

SBT is used as a dictionary, not a replacement foundation

This paper does not propose new axioms or attempt to subsume all of mathematics under one formalism. Instead it makes explicit a set of closure roles that are already present (often implicitly) in mathematical practice.

9.3. Upgrade points

The exhibits suggest concrete upgrade paths if one wishes to strengthen claims.

Stronger analytical closure claims

For calculus, one can replace heuristic feasibility gates with classical stability/consistency hypotheses from numerical analysis, and explicitly state the regularity classes under which the ledger terms vanish. For protocol mismatch, one can connect the observed scaling to known truncation-order estimates.

Richer prime closure ledgers

Exhibit B-1 uses deliberately simple staging and packaging. For $\zeta(s)$ in the critical strip, a natural upgrade is to use the approximate functional equation as an additive staged protocol (valid in the strip) and to compare against smoothed prime-sum / log-derivative ledgers in regimes where such representations are controlled [7]. More informative ledgers could compare additional invariants (e.g. log-derivative statistics), incorporate smoothing optimized for convergence acceleration, or restrict to regimes where the protocols admit controlled continuation. The present draft should be treated as a baseline diagnostic that clarifies what fails without additional structure.

Formalization interface improvements

The Lean anchors are intentionally minimal. Future versions can add small “glue lemmas” that align the informal narrative even more tightly with the formal objects (without attempting to formalize all of analysis in a proof assistant).

9.4. Why the closure lens is still useful

Even at this modest level, the closure lens supplies two pragmatic benefits.

First, it separates *definitions* from *feasibility*. A packaged object is not merely a symbolic completion; it is a completion that is stable under refinement and compatible with chosen constraints.

Second, it encourages controlled comparisons. The same route-mismatch and defect-ledger ideas can be applied to other mathematical packaging problems (measure and integration, functional analysis limits, geometric patching, spectral constructions), where it is otherwise easy to conflate “a formula exists” with “a stable closure exists.”

9.5. Closing

The slogan “To Count a Stone with Six Birds” is literal in this instantiation: counting is a staged discrete protocol; calculus is the stable compression of that protocol under an accounting ledger; and higher layers arise by repeating packaging with new constraints and new mismatch diagnostics. The intent of the paper is to make those roles explicit, auditable, and reusable.

References

1. a, b, c, d, e *Tsiokos I (2026). "Six Birds: Foundations of Emergence Calculus." arXiv. arXiv:2602.00134 [cs.LO]. doi:10.48550/arXiv.2602.00134.*
2. a, b, c, d *Moura L de, Ullrich S (2021). "The Lean 4 Theorem Prover and Programming Language." In: Lecture Notes in Computer Science, Automated Deduction – CADE 28. p. 625–635. doi:10.1007/978-3-030-79876-5_37.*
3. a, b, c, d *The mathlib Community (2020). "The Lean Mathematical Library." In: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs. p. 367–381. doi:10.1145/3372885.3373824.*
4. ^ *Davey B, Priestley H (2002). Introduction to Lattices and Order. Cambridge: Cambridge University Press.*

5. ^The Stacks Project Authors (2026). "The Stacks Project, Section 10.131: Differentials (Tag 00RM)." The Stacks Project. https://stacks.math.columbia.edu/tag/00RM.
6. ^LeVeque RJ (2007). Finite Difference Methods for Ordinary and Partial Differential Equations. Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898717839.
7. a, bTitchmarsh E (1986). The Theory of the Riemann Zeta-Function. Oxford: Clarendon Press.
8. ^Willems JC (1972). "Dissipative Dynamical Systems Part II: Linear Systems with Quadratic Supply Rates." Arch Rational Mech Anal. **45**(5):352–393. doi:10.1007/BF00276494.
9. ^Lee TD, Yang CN (1952). "Statistical Theory of Equations of State and Phase Transitions. II. Lattice Gas and Ising Model." Phys Rev. **87**(3):410–419. doi:10.1103/PhysRev.87.410.
10. ^Tsiokos I (2026). "Six Birds Theory Repository." arXiv. doi:10.48550/arXiv.2602.00134.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: The author is affiliated with Automorph Inc.; no financial competing interests to declare.