A Quantum Arithmetic Completion Layer for Field Structure Theory

Machine-Checkable Verification of Loop-Based Structural Physics

Will

In collaboration with Don Briddell (Field Structure Theory)

Quantum Arithmetic Research Group

January 2026

Abstract

Many non-standard physical frameworks propose geometric or structural explanations for particle formation and mass hierarchies but lack formal, machine-checkable verification mechanisms. We present **Quantum Arithmetic (QA)** as a *completion layer*: a framework that formalizes state spaces, generators, invariants, and failure modes without introducing new physical assumptions. We apply this framework to **Field Structure Theory (FST)** and its applied formulation **Structural Physics (SP)**, due to Briddell, as a case study. Key FST assertions—proton stability via geometric superimposability and particle mass-ratio correspondence via loop counting—are encoded as deterministic certificates with explicit failure taxonomies. The proton symmetry witness yields $\delta_{\text{sym}} \approx 0.00327$, well within the declared stability threshold of 0.01. The up/down quark ratio test shows agreement between loop and MeV representations to within 3.5×10^{-4}. Lambda decay bookkeeping reproduces the proton loop count exactly $(2187 - 243 - 81 - 27 = 1836)$, while a 1.95 MeV discrepancy against the PDG proton mass is logged as a structured warning rather than silently adjusted. All validation is reproducible from published artifacts via a single command.

1 Introduction: The Verification Gap in Structural Physics

1.1 Structural theories and the reproducibility problem

A growing body of work in theoretical physics proposes that particles, forces, and spacetime itself emerge from geometric or topological organization rather than point-particle interactions. Loop-based models, field-structure approaches, and fractal hierarchies offer intuitive structural explanations for phenomena that the Standard Model addresses through symmetry groups and renormalization. However, many such frameworks remain *qualitative*—their claims are stated in prose, illustrated with diagrams, and supported by suggestive numerical coincidences, but they are not expressed in a form that permits independent, automated verification.

This gap is not unique to structural physics. Across the sciences, reproducibility depends on claims being stated precisely enough that a third party can check them without recourse to the original author's interpretation. When claims involve specific numerical relationships—mass ratios, symmetry thresholds, decay bookkeeping—the standard of verification should be *deterministic replay*: given the same inputs and declared invariants, any validator must reach the same conclusion.

1.2 Why verification matters more than agreement

Disagreement about physical theories is scientifically healthy. What is not healthy is the inability to determine *what exactly* a theory claims, *which of those claims are numerically testable*, and *what constitutes failure*. A theory can be internally plausible yet externally unverifiable if its claims are embedded in narrative rather than formal structure.

We distinguish three levels of engagement with a structural theory:

1. **Acceptance**: believing the theory is physically true.
2. **Verification**: confirming that stated claims are internally consistent and reproducible.
3. **Falsification**: identifying specific, bounded conditions under which a claim fails.

Quantum Arithmetic operates at levels (2) and (3). It does not address level (1)—physical truth is outside its scope. Its purpose is to make structural theories *auditable*.

1.3 Contribution of this paper

This paper does not propose new physics. It demonstrates how an existing structural theory—Field Structure Theory (FST) and its applied formulation Structural Physics (SP), due to Briddell—can be:

- *formalized* via explicit state spaces, generators, invariants, and obstruction types;
- *validated deterministically* via certificates with recomputable witnesses;
- *audited for numeric drift* via a structured warning taxonomy that logs discrepancies without silently adjusting them.

The relationship is explicit: Briddell's paper presents the theory; this paper completes it computationally.

2 Overview of Field Structure Theory

We provide a brief, non-competitive summary of FST/SP sufficient for the formalization that follows. For the full treatment, we refer the reader to Briddell's companion manuscript *The Plenum-to-Particle Loop Sequence of Field Structure Theory* [1].

2.1 Core elements

FST posits a non-local, space-filling structural potential called the **Plenum**, composed of continuous action loops. Particles form when these loops organize into stable structures through entanglement, rotation, and condensation. Two complementary loop types operate simultaneously: **Rspin** (counter-clockwise, anti-matter) and **Aspin** (clockwise, real-matter). Rspin loops remain deployed and dynamically rotating; Aspin loops may condense to form interaction boundaries.

Cloaking occurs when condensed Aspin loopage fully encloses deployed Rspin loopage such that no Rspin segment intersects the interaction boundary. In this state, Rspin is structurally present but unobservable. The minimal space-defining unit is the **Structor**: a six-loop nucleus formed from three deployed Rspin loops enclosed by three condensed Aspin loops.

Table 1 summarizes the sector and deployment state space. Crucially, *condensation is a deployment state change within a fixed sector*—an Aspin loop that condenses remains Aspin; it does not transform into Rspin. Sector identity (chirality) is preserved; only observability and boundary-forming role change.

Table 1: FST loop state space: sector and deployment are independent.

State	**Sector**	**Deployment**	**Observable?**	**Structor Role**
Rspin (deployed)	Anti-matter	Deployed	Yes	Inner core (3 loops)
Aspin (deployed)	Real-matter	Deployed	Yes	—
Aspin (condensed)	Real-matter	Condensed	No (boundary)	Outer shell (3 loops)

Key rule: Condensation $\neq$ sector transformation. When Aspin condenses, it forms the interaction boundary but remains in the real-matter sector. Cloaking occurs because the condensed Aspin boundary fully encloses the deployed Rspin core, blocking observer intersection—not because of any sector flip.

The framework maps loop counts onto the **Sierpinski Triangle Fractal (STF)**, whose iteration hierarchy $\{3, 9, 27, 81, 243, 729, 2187\}$ produces ratios claimed to correspond to experimentally measured particle mass ratios in electron volts. FST asserts that mass, charge, and particle stability arise from geometry and loop organization.

2.2 Claims selected for completion

We restrict our verification to the following explicit, numerically testable claims from the source texts:

1. **Proton stability**: The proton (1836 loops) is stable because its hexagonal loop architecture satisfies a superimposability condition, quantified by a symmetry defect δ_{sym} below a declared threshold.
2. **Quark ratio preservation**: The ratio of up to down quark masses measured in MeV matches the ratio of their loop counts.
3. **Lambda decay bookkeeping**: The lambda particle (2187 loops) decomposes as $2187 - 243 - 81 - 27 = 1836$, recovering the proton loop count via exact integer arithmetic.

Only these claims are addressed. We make no judgment on the broader physical program of FST/SP.

3 Quantum Arithmetic as a Completion Framework

3.1 Philosophy: verification without reinterpretation

Quantum Arithmetic treats numbers as *geometric objects with intrinsic structure* rather than abstract quantities. Its foundational axiom is *non-reduction*: arithmetic operations preserve scale-bearing information rather than collapsing it. In the context of this paper, QA serves as a *verification layer*—it encodes the concepts of a target theory (here, FST) into a formal structure that can be validated deterministically, without reinterpreting or replacing the theory's own ontology.

The core theorem of the QA certificate framework is:

$$\text{Capability} = \text{Reachability}(S, G, I) \tag{1}$$

where S is a state space, G is a generator set (permitted operations), and I is an invariant set (constraints that must hold). A claim is *verified* if a witness path exists from an initial state to a target state using only generators in G while preserving all invariants in I. A claim *fails* if an explicit obstruction prevents such a path, and the failure is classified by type.

3.2 Formal objects

For any target theory, QA introduces five formal objects:

Definition 1 (Module Spine)**.** *A **QA module spine** is a tuple $\langle \Omega, \Sigma, \mathcal{G}, \mathcal{I}, \mathcal{F} \rangle$ where:*

- *Ω is the **pattern space** (configuration space of the target theory);*
- *Σ is a **state packet** (typed fields describing any configuration);*
- *$\mathcal{G}$ is a **generator set** (permitted transformations);*
- *$\mathcal{I}$ is an **invariant set** (quantities preserved or bounded);*
- *$\mathcal{F}$ is an **obstruction taxonomy** (classified failure modes).*

For FST, these are instantiated as shown in Table 2.

Table 2: QA module spine for Field Structure Theory.

Object	Name	Description
Pattern Space		
	Plenum Ω	Non-local configuration space of loop structures
Generators		
	σ	Add one loop quantum: $\ell \to \ell + 1$
	λ_3	STF scale step: $\ell \to 3\ell$
	μ	Chirality swap: Rspin $\leftrightarrow$ Aspin
	κ	Condensation toggle: deploy $\leftrightarrow$ condense
	χ	Cloaking operator: modify observability
Invariants		
	ℓ_{total}	Total loop count (mass proxy)
	φ_{STF}	STF basis decomposition coefficients
	δ_{sym}	Symmetry defect (superimposability)
	obs	Observer-relative visibility (visible / cloaked)
Obstruction Types		
	`NOT_IN_STF_BASIS`	Loop count not representable in STF basis
	`SYMMETRY_DEFECT`	δ_{sym} exceeds threshold
	`CLOAKED_FROM_OBSERVER`	Boundary intersection blocked
	`SOURCE_NUMERIC_DRIFT`	Source text internally inconsistent

3.3 Deterministic replay and auditability

All QA certificates satisfy a *deterministic replay* contract:

1. All data is serialized as canonical JSON (UTF-8, sorted keys, no whitespace).
2. Hashes are computed as SHA-256 of the canonical representation.
3. A manifest records the hash of every artifact required for replay.
4. Re-execution of the validator on the same artifacts must produce identical output.

This is enforced by draft-07 JSON schemas and a validator whose source code is itself included in the manifest.

4 Formalization of FST in QA

4.1 The module spine as declarative contract

The QA module spine for FST (artifact `qa_fst_module_spine.json`) encodes the pattern space, state packet, generators, invariants, and obstruction types described in the previous section. It also records:

- **Source artifacts**: explicit references to Briddell's draft, paper summary, and cover letter;
- **External references**: the PDG 2024 proton mass (938.272 MeV), used as an independent reference for drift classification;
- **Submission context**: target journal, cover letter claims, and alignment with the companion manuscript.

The spine is *declarative*: it states what must hold, not how to compute it. Validation logic resides in the validator, which is a separate artifact.

4.2 Failure algebra as first-class structure

A distinguishing feature of QA verification is that *failure is structured.* Every validation produces either:

- a success witness (with computed metrics), or
- one or more **fail records**, each of the form:

$$\{\texttt{move},\ \texttt{fail\_type},\ \texttt{invariant\_diff}\} \tag{2}$$

where `move` identifies the validation step, `fail_type` classifies the failure, and `invariant_diff` records the specific numerical discrepancy. This structure is enforced by a JSON schema (`FAIL_RECORD.v1.schema.json`).

The distinction between *hard failures* and *warnings* is critical. A hard failure (e.g., `SYMMETRY_DEFECT`) means a declared invariant is violated. A warning (e.g., `SOURCE_NUMERIC_DRIFT`) means a downstream comparison to an external reference shows drift, but no declared invariant of the source theory is broken. The completion layer *logs* drift rather than *correcting* it.

5 Worked Certificate I: Proton Stability via δ_{sym}

5.1 Structural claim and formal witness

FST asserts that the proton (1836 loops) is stable because its hexagonal loop architecture is quasi-superimposable with the Plenum's tetrahedral isometry template. Specifically, the proton hexagon has two excess triangles on three of its six sides, yielding a small but non-zero asymmetry.

We formalize this as a **symmetry witness**:

$$\text{side excess} = [2, 2, 2, 0, 0, 0] \tag{3}$$

$$L_1 = \sum_i |\text{excess}_i| = 6 \tag{4}$$

$$\delta_{\text{sym}} = \frac{L_1}{\ell_{\text{total}}} = \frac{6}{1836} \approx 0.00327 \tag{5}$$

The declared stability threshold is $\delta_{\text{sym}} \leq 0.01$. Since $0.00327 \leq 0.01$, the certificate result is **PASS**.

5.2 Interpretation

This is not a proof of physical stability. It is verification that:

1. the declared symmetry defect is *recomputable* from the stated witness data;
2. the recomputed value matches the declared value to machine precision;
3. the value satisfies the declared threshold.

The certificate can be replayed independently by any party with access to the published artifacts. If the witness data or threshold were changed, the validator would detect the discrepancy via hash mismatch.

6 Worked Certificate II: Loop-to-MeV Homomorphism

6.1 Ratio preservation (hard check)

FST claims that the ratio of quark masses measured in MeV corresponds to the ratio of their loop counts. For the up and down quarks:

Table 3: Up/down quark ratio test.

System	Up	Down	Ratio
MeV	2.552	4.925	0.51817
Loops	378	729	0.51852

The absolute difference between ratios is $|\Delta| \approx 3.46 \times 10^{-4}$, well within the declared tolerance of 10^{-3}. The certificate result for this test is **PASS**.

6.2 Lambda decay bookkeeping

FST decomposes the lambda particle (2187 loops) via subtraction of STF basis elements:

$$2187 - 243 - 81 - 27 = 1836 \tag{6}$$

This is exact integer arithmetic. The validator recomputes the sum and confirms identity with the declared result. Certificate result: **PASS**.

The corresponding MeV bookkeeping produces:

$$1115.683 - 124.173 - 41.391 - 13.797 = 936.322 \text{ MeV} \tag{7}$$

This recomputation also matches the declared value exactly.

6.3 Numeric drift as logged evidence

The source texts claim that "measuring in electron masses (MeV) and in loops have identical outcomes for proton 1836." However, the MeV bookkeeping result (936.322 MeV) differs from the PDG 2024 proton mass (938.272 MeV) by approximately 1.95 MeV.

The QA validator classifies this as:

```
{
  "move": "ENFORCE_HOMOMORPHISM",
  "fail_type": "SOURCE_NUMERIC_DRIFT",
  "invariant_diff": {
    "bookkeeping_mev": 936.322,
    "proton_mev_reference": 938.272,
    "drift": 1.95,
    "within_tolerance": true
  },
  "severity": "warning"
}
```

This is the central epistemological point of the completion layer. The drift is:

- **not ignored**: it is explicitly recorded with its magnitude;
- **not promoted to failure**: the source theory's own internal arithmetic is consistent (the bookkeeping subtraction is correct);
- **not silently adjusted**: neither the loop count nor the MeV values are modified to produce agreement;
- **bounded**: the drift (1.95 MeV) is within the declared tolerance (5.0 MeV) and is classified as `within_tolerance: true`.

This approach is superior to three common alternatives:

1. *Silent omission*: presenting only the loop bookkeeping and ignoring the MeV comparison.
2. *Post-hoc correction*: adjusting values to force agreement.
3. *Blanket rejection*: dismissing the entire ratio correspondence because one downstream comparison drifts.

By logging drift as structured data, the completion layer allows any reader to assess the discrepancy on its own terms.

7 Reproducibility and Tooling

7.1 Artifact set

The complete set of artifacts required for replay is published at a version- controlled repository and consists of:

The manifest covers nine artifacts (three data files, five schemas, and the validator source), each hashed independently. The manifest itself records a composite hash of all entries.

7.2 One-command verification

The entire validation can be executed with:

Table 4: Published artifact set.

Artifact	Role
`qa_fst_module_spine.json`	Module spine (declarative contract)
`qa_fst_cert_bundle.json`	Worked certificates (witnesses + tests)
`qa_fst_submission_packet_spine.json`	Companion paper posture lock
`qa_fst_manifest.json`	SHA-256 manifest over all artifacts
`qa_fst_validate.py`	Deterministic validator + self-test
`schemas/*.schema.json`	Draft-07 JSON schemas (5 types)

```
python qa_fst_validate.py --validate
```

This produces a JSON report containing all computed metrics, hash values, and any warnings or failures. The validator also supports self-test mode (eight internal checks) and manifest regeneration.

At the repository level, a single command validates the full QA certificate stack—including the tetrad (four certificate directions), conjecture ledger, and the FST module:

```
python qa_meta_validator.py
```

The FST module is invoked as a subprocess, and its JSON output is checked for structural conformance. Any third party with access to the repository can reproduce the result.

8 Why This Is a Completion Layer, Not a Competing Theory

8.1 What QA does not do

It is important to state explicitly what the QA completion layer does *not* claim:

- It does not propose alternative physics. The Plenum, loops, STF hierarchy, Rspin/Aspin sectors, and cloaking mechanism are all Briddell's constructs, used here exactly as stated.
- It does not reinterpret FST concepts. The generators $(\sigma, \lambda_3, \mu, \kappa, \chi)$ are named translations of operations described in the source texts, not new formalisms.
- It does not "correct" the theory. Where numeric drift exists (Section 6.3), it is logged, not adjusted.

8.2 What QA adds

The completion layer adds three capabilities that did not previously exist for FST:

1. **Explicitness**: Claims are encoded as typed fields with declared thresholds, making it unambiguous what constitutes success or failure.
2. **Replayability**: Any third party can recompute every metric from the published artifacts and verify hash integrity.
3. **Structured failure**: When something does not match—whether an internal inconsistency or an external reference mismatch—the discrepancy is classified by type, bounded by magnitude, and recorded for inspection.

This posture converts ambiguity into data. It allows disagreement to be *precise* rather than rhetorical. A critic can point to a specific fail record; a supporter can point to a specific pass. Neither needs to argue about what the theory "really means," because the formal encoding makes the meaning explicit.

9 Generalization and Future Work

The QA completion-layer approach is not specific to FST. Any structural theory that makes numerically testable claims can be formalized via the same pattern:

1. Identify the theory's state space, permitted operations, and conserved quantities.
2. Encode these as a QA module spine.
3. Select explicit, bounded claims and construct certificates with recomputable witnesses.
4. Classify failures by type and log drift against external references.

For FST specifically, natural extensions include:

- Additional particle certificates (electron, neutron, pion) using the same δ_{sym} metric.
- Extended decay-path bookkeeping for particles beyond the lambda.
- Formal cloaking certificates that test the observability obstruction (`CLOAKED_FROM_OBSERVER`) against specific sector/cloak-state configurations.

The broader program is community-driven verification via published manifests: any researcher can fork the artifact set, modify a claim or threshold, and observe whether the validator still passes. This makes scientific engagement with structural theories *constructive* rather than purely critical.

10 Conclusion

Structural theories of physics deserve rigorous verification mechanisms commensurate with the specificity of their claims. Quantum Arithmetic provides a neutral, formal completion layer that encodes claims as typed certificates, validates them deterministically, and classifies failures as structured data.

Applied to Field Structure Theory, this approach demonstrates that:

- the proton stability claim is internally consistent ($\delta_{\text{sym}} \approx 0.00327 \leq 0.01$);
- the loop-to-MeV ratio correspondence holds within declared tolerances ($|\Delta| \approx 3.5 \times 10^{-4} \leq 10^{-3}$);
- lambda decay bookkeeping is exact in loop space ($2187 - 243 - 81 - 27 = 1836$);
- and a 1.95 MeV drift against the PDG proton mass is logged as a bounded warning, not hidden.

The completion layer neither endorses nor disputes FST's physical claims. It makes them *auditable*. We invite independent verification, extension, and—where warranted—precise, structured disagreement.

References

[1] D. Briddell, "The Plenum-to-Particle Loop Sequence of Field Structure Theory (FST)," manuscript submitted to *Frontiers of Physics*, 2026.

[2] R. L. Workman *et al.* (Particle Data Group), "Review of Particle Physics," *Phys. Rev. D* **110**, 030001 (2024).

[3] W. (1r0nw1ll), "Quantum Arithmetic Research: Certificate Spine and Completion Layer," `https://github.com/1r0nw1ll/quantum-arithmetic-research`, 2026.

A Validator Output

The complete JSON output of `qa_fst_validate.py -validate` is reproduced below for reference:


```
{
  "result": "PASS_WITH_WARNINGS",
  "ok": true,
  "hashes": {
    "spine": "cf488a5d...eb67bf7f",
    "cert_bundle": "ea2a3fcd...147f2f22"
  },
  "metrics": {
    "delta_sym_recomputed": 0.0032679738562091504,
    "l1_recomputed": 6,
    "lambda_loops_recomputed": 1836,
    "lambda_mev_recomputed": 936.322,
    "proton_mev_drift": 1.95,
    "u_d_loop_ratio_recomputed": 0.5185185185185185,
    "u_d_mev_ratio_recomputed": 0.5181725888324873,
    "u_d_ratio_abs_delta": 0.0003459296860311989
  },
  "warnings": [
    {
      "move": "ENFORCE_HOMOMORPHISM",
      "fail_type": "SOURCE_NUMERIC_DRIFT",
      "severity": "warning",
      "invariant_diff": {
        "bookkeeping_mev": 936.322,
        "proton_mev_reference": 938.272,
        "drift": 1.95,
        "within_tolerance": true
      }
    }
  ]
}
```


B SHA-256 Manifest

The manifest covers nine artifacts. A representative excerpt:

```
Module spine: cf488a5d8309feaf...
Cert bundle: ea2a3fcd2747b0c6...
Submission packet: e31afb0970baefdd...
```

```
Validator source: 7cae088dcd7daabf...
Manifest hash: 9a3f68f071293b9f...
```

Full hashes and all schema entries are recorded in `qa_fst_manifest.json`.