Foundations of the Amplistwistor Program: Conceptual, Logical, and Formal Descriptions

Flyxion

December 5, 2025

Abstract

This essay describes the goals, motivations, and formalization strategy underlying the Amplistwistor Program. Our objective is to explain, in accessible language and with increasing levels of formal rigor, what amplistwistors are meant to model, why they matter for a mathematical description of cortical computation, and how we intend to translate these concepts into first-order logic, Lean, and finally into SpherePOP, a domain-specific calculus for semantic computation.

The document proceeds in four layers: a common-language conceptual overview, a first-order logical reconstruction, a Lean specification intended for mechanized verification, and a SpherePOP implementation sketch. Each layer expresses the same underlying ideas but with progressively stricter constraints and clearer commitments.

1 Introduction: What We Are Trying to Achieve

The work described here aims to build a rigorous and fully formal model of a central idea: that cortical computation can be described as a combination of local nonlinear transformations (amplistwistors) and global geometric or spectral constraints (eigenmodes, resonances, and field interactions).

While previous essays have developed the mathematics of amplitwistors, semigroup methods, and eigenmode structures, we now seek to build an *integrated formal program.* The goal is to translate intuitive neurocomputational ideas into:

(a) a clear conceptual narrative understandable without mathematical training,

(b) a first-order logical theory describing the essential commitments,

(c) a Lean formalization supporting mechanized reasoning,

(d) a SpherePOP model expressing these concepts in executable semantics.

The overarching aim is to create a pathway from intuition to formal verification to executable semantics, ensuring that each conceptual claim can be expressed, checked, and instantiated at multiple levels of rigor.

2 A Common-Language Explanation of the Amplistwistor Framework

The central intuition is simple. When the brain processes information, each small region of cortexa cortical columnperforms a transformation on incoming neural activity. This transformation is neither a simple linear amplification nor a rigid rotation; instead it mixes scaling, twisting, warping, and projecting. We call these transformations *amplistwistors* because they generalize Needhams idea of the "amplitwist" into a multidimensional nonlinear operator.

Two scales are essential:

- **Local scale:** each column applies an amplitwistor to its local field.
- **Global scale:** the cortical manifold imposes resonant and geometric constraints.

Composition across these scales yields a powerful representational and computational architecture analogous to deep learning systems but supported by geometric and spectral structure rather than discrete symbolic manipulation.

This essay focuses on formalizing these intuitions.

3 First-Order Logical Description of the Framework

We now express the core commitments using first-order logic (FOL). This is not yet an axiomatization but a skeleton from which the Lean library will grow.

3.1 Basic Ontology

We assume three fundamental sorts:

$$\mathsf{Column}, \qquad \mathsf{Field}, \qquad \mathsf{Transformation}.$$

3.2 Primitive Functions

$$\mathsf{state} : \mathsf{Column} \to \mathsf{Field},$$

$$\mathsf{apply} : \mathsf{Transformation} \times \mathsf{Field} \to \mathsf{Field},$$

$$\mathsf{compose} : \mathsf{Transformation} \times \mathsf{Transformation} \to \mathsf{Transformation}.$$

We also include a global resonance relation:

$$\mathsf{Resonant}(c_1, c_2),$$

meaning columns c_1 and c_2 share an eigenmode constraint.

3.3 Axioms

Axiom 1 (Local Update).

$$\forall c \in \mathsf{Column}\ \exists T \in \mathsf{Transformation} :\ \mathsf{state}(c)' = \mathsf{apply}(T, \mathsf{state}(c)).$$

Axiom 2 (Compositionality).

$$\forall T_1, T_2\ \exists T_3 :\ T_3 = \mathsf{compose}(T_1, T_2).$$

Axiom 3 (Nonlinearity). There exists some T that is not affine.

Axiom 4 (Global Coherence).

$$\mathsf{Resonant}(c_1, c_2) \Rightarrow \mathsf{state}(c_1)' \text{ depends on } \mathsf{state}(c_2).$$

These axioms define a minimal dynamical spine upon which further structure will be added in Lean.

4 Amplitwistor Operators and Their Action on RSVP Field Dynamics

The SpherePop operad offers a combinatorial syntax for describing spherical emissions, merges, and interference patterns. To connect this syntax to the continuous dynamics of RSVP fields, we require a semantic algebra acting on the field space.

4.1 RSVP Fields as Semantic State Space

RSVP fields consist of a scalar Φ, a vector field $\mathbf{v}$, and an entropy field S defined on a domain M. Their evolution obeys a system of nonlinear PDEs:

$$\partial_t \Phi = D_1 \Delta \Phi + \alpha_1 (\Phi \times \mathbf{v}) - \beta_1 S \Phi + \eta_\Phi, \tag{1}$$

$$\partial_t \mathbf{v} = D_2 \nabla \Phi + \alpha_2 (\nabla \times \mathbf{v}) - \beta_2 S \mathbf{v} + \eta_v, \tag{2}$$

$$\partial_t S = \gamma_1 |\nabla \cdot \mathbf{v}| + \gamma_2 \Phi^2 - \gamma_3 S + \eta_S. \tag{3}$$

The state space is therefore:

$$X = H^k(M; \mathbb{R}) \times H^k(M; \mathbb{R}^3) \times H^k(M; \mathbb{R}).$$

4.2 Pop-Induced Field Transformations

A nullary SpherePop operation p induces:

$$T_p(\Phi, \mathbf{v}, S) = (\Phi + \delta\Phi_p,\ \mathbf{v} + \delta\mathbf{v}_p,\ S + \delta S_p).$$

4.3 Amplitwistor Structure

An amplitwistor at a point x consists of:

$$\mathcal{A}_x = (\alpha_x, \tau_x, \pi_x),$$

where:

1. α_x is a scalar amplitude modulation,
2. τ_x is a twisting or rotation in $T_x M$,
3. π_x is a nonlinear projection or contraction.

These act pointwise:

$$\mathcal{A}(\Phi, \mathbf{v}, S)(x) = \big(\alpha_x(\Phi(x)),\ \tau_x(\mathbf{v}(x)),\ \pi_x(S(x))\big).$$

A pop p induces the integrated action:

$$T_p(f) = \int_M \mathcal{A}_{p,x}(f)\, K_p(x)\, dx.$$

4.4 Cascades and Operadic Composition

If $P_1, \ldots, P_n$ are SpherePop processes with semantic transforms $T_1, \ldots, T_n$, then:

$$T_{\text{cascade}} = T_1 \circ T_2 \circ \cdots \circ T_n.$$

4.5 Interaction With RSVP PDEs

Between pop times t_k:

$$\frac{d}{dt}(\Phi, \mathbf{v}, S)(t) = F_{\text{RSVP}}(\Phi, \mathbf{v}, S)(t),$$

and at each pop:

$$(\Phi, \mathbf{v}, S)(t_k^+) = T_{p_k}\big((\Phi, \mathbf{v}, S)(t_k^-)\big).$$

5 Discrete Field Evolution and Numerical Realizations

Any operational semanticswhether in Lean or SpherePOPmust discretize the continuous fields. We justify the structures used in the Lean implementation.

5.1 Finite Grid Approximation

A discretization of M yields grid functions u_h approximating continuous fields u. Standard convergence guarantees apply under sufficient regularity.

5.2 Discrete Laplacians

The five-point stencil:

$$(\Delta_h u)_{ij} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij}$$

approximates Δu with $O(h^2)$ error.

5.3 Gaussian Pops

The discrete kernel:

$$K_\sigma(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right)$$

approximates a Greens function at scale σ.

5.4 Euler Integration

The update:

$$u_{ij}^{k+1} = u_{ij}^k + dt \cdot \kappa(\Delta_h u^k)_{ij}$$

implements a discrete semigroup step.

5.5 Compositional Semantics

Diffusion and pop events compose cleanly:

$$u \mapsto T_p(u) \mapsto T_{\text{Euler}}(dt)(T_p(u)).$$

This realizes discrete amplitwistor cascades.

6 Hybrid Operadic–Semigroup Algebras

We now articulate the combined algebraicanalytic structure governing amplistwistor cascades.

Let $T(t)$ denote RSVP semigroup evolution and $\mathcal{P}$ denote a pop operator. Hybrid evolution has the form:

$$T(t_2 - t_1)\, \mathcal{P}_1\, T(t_1 - t_0)\, \mathcal{P}_0\, T(t_0).$$

Sequential operadic composition mirrors causal temporal evolution. Parallel composition mirrors linear superposition of Greens-function responses.

The key point: *amplistwistors are geometrically grounded operators*, not syntactic placeholders.

7 Correspondence With Cortical Traveling Waves

Empirical studies show that cortical computation is mediated by traveling waves, eigenmodes, and hierarchical temporal processing. Amplistwistor cascades reproduce these signatures:

- Pop events excite eigenmodes.
- Low-frequency modes propagate across long distances.
- High-frequency modes encode transient responses.
- Cascades along eigenmodes mirror hierarchical processing.

This structural correspondence aligns RSVP theory with ECoG, MEG, ultrafast fMRI, and language-latency studies.

8 Convergence and Stability of Discrete Semigroups

We summarize the convergence properties of Euler stepping and Gaussian pops.

8.1 Local Truncation Error

For discrete operator $T_h^{\mathrm{Euler}}(dt)$:

$$\|T(dt)u - T_h^{\mathrm{Euler}}(dt)u_h\| = O(dt^2 + h^2).$$

8.2 Stability

CFL condition:

$$dt \leq \frac{h^2}{4\kappa}.$$

8.3 Gaussian Pops and Greens Functions

$$K_\sigma(d) = G(x, y; t) + O(h^2).$$

8.4 Trotter–Kato Approximation

$$\lim_{h\to 0,\, dt\to 0} \Big(S_h(dt)\Big)^{\lfloor t/dt \rfloor} = T(t) \circ \mathcal{P}_t.$$

8.5 Spectral Convergence

$$|\alpha_n^{(h)} - \alpha_n| = O(h^2), \qquad |e^{-\kappa\lambda_n t} - (1 - dt\lambda_n)^{t/dt}| = O(dt).$$

9 Spectral Decomposition of Cascades

Eigenmodes of $\Delta_{\mathcal{M}}$ or L_G provide a natural basis for understanding amplitwistor cascades.

A Gaussian pop decomposes as:

$$K_\sigma(x, y) = \sum_n \alpha_n \psi_n(x)\psi_n(y).$$

Low modes: broad, slow, integrative. High modes: local, fast, ephemeral.

Sequential cascades correspond to deeper traversals in eigenmode space.

10 Synthesis and Implications for Multiscale Computation

Amplistwistor cascades unify discrete operadic structure with continuous field evolution. They reproduce:

- hierarchical temporal processing,
- structured wave propagation,
- spectral signatures of cognition,
- compositional semantic computation.

They provide a rigorous mathematical framework for neural computation that integrates:

1. operadic syntax,
2. differential geometry,
3. semigroup theory,
4. spectral analysis,
5. discrete numerical realization.

Future work includes nonlinear extensions, stochasticity, Lean formalization, and Sphere-POP execution layers.