

OMEGA Kernel Construct

A Typed, Reproducible Interface for a
One-Arcsecond Planar Euclidean Kernel

Matthew Alan Newman

ORCID: [0009-0007-5468-3125](https://orcid.org/0009-0007-5468-3125)

Release	OMEGA Kernel Construct v1.1.0
Date	6 September 2026
Node contract	OMEGA_KERNEL_NODES/1.0
Document status	Technical release paper; not peer reviewed
Project	omega-naught.grok.me

Release record. The referenced source archive has SHA-256 `bf9a02e1692d5ae46b907ac7b96e44483744824e6aedef12e83abe0e750ac7a39`. The package contains exact-expression records, binary64 realizations, executable validators, a dependency-free browser demonstration, and a deterministic per-file release inventory.

Abstract

This paper specifies the presentation and conformance layer of the OMEGA local Euclidean kernel. The mathematical input is a one-arcsecond angular seed, bilateral extensions of lengths a and $a\varphi$, and an oriented major arc. The resulting six marked points are collinear, while the configuration obtained by adjoining the distinguished center has center-relative Gram rank two. The software release fixes a canonical unit realization, preserves exact-expression strings beside binary64 evaluations, and enforces a three-way claim discipline: **kernel**, **optional**, and **not_implied**. A fifteen-stage screenplay exposes proved properties in order without recasting those stages as fifteen proof steps. The separately adopted scaffold is available only after kernel completion. The paper also proves an exact scalar change of variables between (φ, ε_0) and (τ_Ω, λ) , records its machine certificate, and states the precise boundaries connecting this layer to the floor-map, SRSD, and cubature branches. The contribution is a reproducible, machine-checkable interface around the geometry, not a competing geometric evaluator and not an assertion of unproved outer structure.

Contents

1 Purpose and contribution	1
2 The local Euclidean kernel	1
3 Exact scalar interface	3
4 Canonical machine contract	4
5 Fifteen-stage theorem screenplay	5
6 Typed integration boundaries	6
7 Verification architecture	6
8 Integration rule	7
9 Conclusion	7
A Canonical node table	8
B Release file map	8
C References	9

1 Purpose and contribution

The OMEGA Kernel Construct package addresses a specific reproducibility problem: how to present, export, and integrate the local Euclidean kernel while preventing display choices or optional drawings from becoming mathematical claims. The package sits above the existing Paper VII evaluator. It does not replace the Euclidean proof and does not infer a shell, clam, extrusion, outer circle, or signed depth.

The release makes five contributions.

1. It gives the planar marked kernel a canonical unit realization and a versioned node-log contract.
2. It separates exact expressions from their binary64 evaluations and permanently separates the true one-arcsecond seed from a twelve-degree display exaggeration.
3. It assigns every drawable and every exclusion a claim type, making the kernel/scaffold/nonconsequence boundary executable.
4. It provides a deterministic fifteen-stage presentation state machine whose automated replay ends at the completed kernel.
5. It closes the scalar interface by an exact bijection and includes a zero-dependency symbolic certificate for that layer.

The word *exact* is used in two distinguishable senses. Exact-expression strings such as `pi/648000` specify mathematical values. Binary64 fields are deterministic numerical realizations of those values. A binary64 check is not silently promoted to an exact proof.

2 The local Euclidean kernel

2.1 Primitive data

Let

$$\varphi = \frac{1 + \sqrt{5}}{2}, \quad \varepsilon = \frac{\pi}{648000}, \quad \delta = \frac{\varepsilon}{2}, \quad s = \sin \delta, \quad c = \cos \delta. \quad (1)$$

Thus ε is one arcsecond in radians. For a scale $a > 0$, a distinguished center α , and orthonormal vectors u, v , define

$$x_0 = s, \quad x_1 = 1 + s, \quad x_2 = \varphi^2 + s. \quad (2)$$

The six marked points are

$$p_{j,\sigma} = \alpha + a(cu + \sigma x_j v), \quad j \in \{0, 1, 2\}, \quad \sigma \in \{-1, +1\}. \quad (3)$$

The names are

$$(p_{0,-}, p_{0,+}) = (d, b), \quad (p_{1,-}, p_{1,+}) = (d_1, b_1), \quad (p_{2,-}, p_{2,+}) = (d_2, b_2).$$

Proposition 2.1 (Seed and extensions). *The points d, b lie at distance a from α and subtend angle ε . Moreover,*

$$|d_1 - d| = |b_1 - b| = a, \quad |d_2 - d_1| = |b_2 - b_1| = a\varphi.$$

Proof. Since $c^2 + s^2 = 1$, the two seed radii have length a . Their normalized dot product is $c^2 - s^2 = \cos(2\delta)$, and their oriented separation is $2\delta = \varepsilon$. The first extensions change only the v -coefficient by one. The second change it by $x_2 - x_1 = \varphi^2 - 1 = \varphi$, using $\varphi^2 = \varphi + 1$. $\square$

2.2 Canonical unit realization

The machine profile fixes

$$\alpha = (0, 0, 0), \quad a = 1, \quad u = (0, -1, 0), \quad v = (1, 0, 0), \quad n = u \times v = (0, 0, 1). \quad (4)$$

Normalized coefficients (ξ_u, ξ_v) therefore map to

$$(\xi_u, \xi_v) \mapsto (\xi_v, -\xi_u, 0). \quad (5)$$

Consequently, all six kernel marks have coordinates

$$(\pm x_j, -c, 0), \quad j = 0, 1, 2. \quad (6)$$

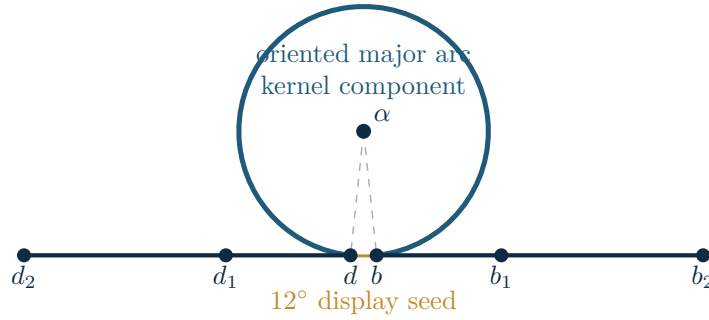


Figure 1: Planar kernel shown with the nonnormative twelve-degree display exaggeration. Every exported coordinate is instead evaluated at the true $\varepsilon = 1''$ seed. The light full circle and radial segments are construction inputs; the major arc and bilateral extension segments are kernel objects.

2.3 Affine dimension and Gram rank

Let the row order be d_2, d_1, d, b, b_1, b_2 , and put

$$z = (-x_2, -x_1, -x_0, x_0, x_1, x_2)^\top, \quad \mathbf{1}_6 = (1, 1, 1, 1, 1, 1)^\top.$$

Theorem 2.2 (Planarity and center-relative factorization). *The six marks in Equation (3) have affine dimension one. After adjoining α , the marked configuration has affine dimension two. Its center-relative Gram matrix is*

$$G_\Omega = a^2 \left(c^2 \mathbf{1}_6 \mathbf{1}_6^\top + z z^\top \right), \quad \text{rank } G_\Omega = 2. \quad (7)$$

Proof. All six marks lie on the affine line $\alpha + acu + \mathbb{R}v$, and at least two have distinct v -coefficients. Since $c > 0$, the center α does not lie on that line; adjoining it produces an affine plane. The center-relative row vectors are $a(cu + z_i v)$. Orthonormality gives $G_{ij} = a^2(c^2 + z_i z_j)$, which yields Equation (7). The vectors $\mathbf{1}_6$ and z are linearly independent and both coefficients are nonzero, so the rank is two. $\square$

This rank-two statement is center-relative. It does not contradict the affine dimension one of the six baseline marks.

3 Exact scalar interface

3.1 Canonical identities

Define

$$\Omega_0 = \varphi + \pi, \quad \tau_\Omega = \Omega_0^{-1}, \quad \lambda = \frac{\varphi}{\Omega_0}, \quad \mu = 1 - \lambda. \quad (8)$$

These are definitions. They immediately imply

$$\mu = \frac{\pi}{\Omega_0}, \quad \lambda + \mu = 1, \quad \frac{\lambda}{\tau_\Omega} = \varphi, \quad \frac{\mu}{\tau_\Omega} = \pi. \quad (9)$$

Using $\varphi^2 = \varphi + 1$ and $\lambda = \varphi\tau_\Omega$ gives

$$\boxed{\lambda^2 - \tau_\Omega \lambda - \tau_\Omega^2 = 0.} \quad (10)$$

The symbol 1 in these equations is the scalar unit under the canonical embedding $\mathbb{Q} \hookrightarrow \mathbb{R}$. The vector $\mathbf{1}_6$ in Equation (7) is built componentwise from that scalar, but it is not the same typed object. No cross-type identity is asserted.

3.2 A two-parameter bijection

Let $K = 648000$,

$$A = (0, \infty) \times (0, \infty), \quad B = (0, \infty) \times (0, 1).$$

Define

$$\Psi(x, e) = \left(\frac{1}{x + Ke}, \frac{x}{x + Ke} \right), \quad (x, e) \in A, \quad (11)$$

$$\Phi(t, \ell) = \left(\frac{\ell}{t}, \frac{1 - \ell}{Kt} \right), \quad (t, \ell) \in B. \quad (12)$$

Theorem 3.1 (Exact scalar reconstruction). *The map $\Psi : A \rightarrow B$ is a bijection with inverse Φ . At the canonical point $e = \varepsilon_0 = \pi/648000$,*

$$\Psi(\varphi, \varepsilon_0) = (\tau_\Omega, \lambda), \quad \Phi(\tau_\Omega, \lambda) = (\varphi, \varepsilon_0).$$

Proof. For $q = x + Ke$, direct substitution gives

$$\Phi(\Psi(x, e)) = \left(\frac{x/q}{1/q}, \frac{1 - x/q}{K/q} \right) = (x, e).$$

Conversely,

$$\frac{\ell}{t} + K \frac{1 - \ell}{Kt} = \frac{1}{t},$$

so substituting Equation (12) into Equation (11) returns (t, ℓ) . Positivity places each map in the stated codomain. At the canonical point, $\varphi + K\varepsilon_0 = \varphi + \pi = \Omega_0$, giving the final identities. $\square$

The inverse denominator is the product $648000t$. This explicit notation removes the comma ambiguity present in an earlier prose draft.

The bundled Python certificate checks both general compositions by exact sparse polynomial cross multiplication over $\mathbb{Q}$. It then checks Equations (9) and (10) in $\mathbb{Q}(\varphi)[\pi]$, reducing products with $\varphi^2 = \varphi + 1$. This certificate is exact for the scalar layer. It does not constitute a symbolic proof of the trigonometric or Euclidean claims in Section 2.

4 Canonical machine contract

4.1 Normative record

The wire contract is named `OMEGA_KERNEL_NODES/1.0`. Package version 1.1.0 adds a strict canonical profile while leaving the wire version stable. Each normalized coordinate has two fields:

- an exact-expression string from a fixed vocabulary; and
- the corresponding finite binary64 evaluation.

The exact string carries the mathematical expression. The numerical field supports immediate browser and integration use. A validator checks their canonical pairing, the unit-frame conversion, finite values, and exact zero in the z -coordinate of every drawable node.

Listing 1: Abbreviated normative record.

```
{
  "schema": "OMEGA_KERNEL_NODES",
  "version": "1.0",
  "normative_seed": {
    "epsilon_exact": "pi/648000",
    "epsilon_arcseconds": 1
  },
  "presentation": {
    "default_display_angle_degrees": 12,
    "display_is_non_normative": true,
    "export_uses_true_epsilon": true
  }
}
```

4.2 Claim typing

Table 1: Claim tags and permitted use.

Tag	Meaning	Machine rule
<code>kernel</code>	Theorem-backed node or display datum	May occur in the kernel chapter; receives an integer stage where applicable.
<code>optional</code>	Separately adopted scaffold datum	May occur only in the scaffold chapter after stage 15; proof stage is null.
<code>not_implied</code>	Explicit exclusion, not a geometric node	May occur only in the exclusion ledger and is never drawable.

The exclusion ledger is closed and exact:

`{outer_circle, clam, shell, extrusion, signed_depth}`.

A missing member, an unknown extra member, or a tag other than `not_implied` is a conformance failure.

4.3 Optional adopted scaffold

Only after completion of the kernel may a second chapter display the adopted scaffold. With $w = -u$ and $m = \alpha + acu$, set

$$S = m + a\varphi^2w, \quad E = d_2 - a\varphi^2w, \quad L = b_2 - a\varphi^2w. \quad (13)$$

At unit scale its normalized coordinates are

$$S = (c - \varphi^2, 0), \quad E = (c + \varphi^2, -x_2), \quad L = (c + \varphi^2, x_2).$$

These nodes are recorded as **optional**. Equation (13) is a separate adoption and is not a consequence of the local seed construction.

5 Fifteen-stage theorem screenplay

The interface reveals the kernel through fifteen ordered presentation stages. These stages refine the display of the Paper VII theorem and its corollaries. They are not a renumbering of the five paragraphs labelled “Step” in that theorem’s proof.

Table 2: Screenplay stages and their mathematical sources.

Stage	Displayed result	Mathematical source
1	One-arcsecond seed	Definition 3.1; equation (4); Theorem 3.4 proof Step 1
2	Bilateral golden extensions	Definition 3.2; equations (5)-(9); proof Step 2
3	Six marks collinear	Theorem 3.4; Proposition 8.1
4	Center-augmented plane	Theorem 3.4; Proposition 8.1
5	Exact center radii	Equations (31)-(37)
6	Pairwise center-angle rule	Equations (22), (23), (29), and (34)
7	Six-ray fan	Equation (23) and Figure 2
8	$\theta_0 = \varepsilon/2$	Theorem 6.1, equation (24)
9	$\theta_1 = \pi/4 + \varepsilon/4$	Theorem 6.1, equation (25)
10	Golden outer increment γ	Lemma 6.2, equations (27)-(28)
11	Fifteen center angles	Equation (29) and Appendix A
12	Center-relative Gram rank two	Equations (39)-(46); Proposition 8.1
13	Labeled Gram and frame recovery	Corollary 5.1; Proposition 9.1; labeled Gram reconstruction
14	Congruence at fixed scale; similarity across scale	Theorem 3.4; equations (53)-(56)
15	Major-arc kernel closure	Definition 3.3; equations (10)-(13); proof Steps 3-4

Proposition 5.1 (Replay boundary). *Automated replay reaches stage 15 in the kernel chapter and stops. It cannot enter the optional scaffold chapter.*

Proof. Replay initializes at stage 0 in the kernel chapter. For each replay tick with stage $k < 15$, the state becomes $k + 1$ and remains in the kernel chapter. The replay flag is retained exactly when $k + 1 < 15$. Hence the transition to stage 15 clears the flag. The separate transition that enters the scaffold is implemented only by a deliberate post-completion step or jump, neither of which is invoked by replay. $\square$

The implemented controls are Step, Back, Replay, speed selection, Jump to Kernel, Jump to Scaffold, and JSON export. Presentation state is computed from the same screenplay metadata used by the TypeScript contract tests and the dependency-free browser module.

6 Typed integration boundaries

The package permits cross-paper interfaces only with an explicit meaning and an explicit nonclaim.

Table 3: Typed interfaces among the OMEGA branches.

Link	Exact meaning	Boundary
VII $\leftrightarrow$ V	The scalar pairs (φ, ε_0) and (τ_Ω, λ) reconstruct one another.	Scalar layer only; geometry also requires realization and construction data.
V $\rightarrow$ I	$\lambda = \lambda_\Omega = \varphi/(\varphi + \pi)$ as real numbers.	Parameter identification, not whole-theorem dependence; Paper I scale h is not τ_Ω .
I $\leftrightarrow_{\mathcal{D}_{15}}$ II	The exact and rational floor maps agree for $0 \leq d \leq 32767$.	Not global; $q \neq \lambda$, with first disagreement $d^* = 42189065$.
II $\rightarrow$ III	$E_S(Qd) = (\lfloor Qd/S \rfloor, Qd - S\lfloor Qd/S \rfloor)$.	An exact use of frozen SRSD, not an SRSD revision.
VII/V and IV/VI	Compatible coefficient representation over $\mathbb{Q}(\sqrt{5})$.	Shared coefficient field only; no geometry-to-cubature proof dependence.

The finite domain above is $\mathcal{D}_{15} = \{0, 1, \dots, 32767\}$, and

$$q = \frac{339949771344778}{10^{15}} \neq \lambda.$$

The package’s domain replay is a binary64 implementation check; the exact finite certificate remains external. Likewise, the independent cubature branch currently has exact positive certificates for $f = 23$ (5292 nodes, degree 71) and $f = 24$ (5762 nodes, degree 75). This status does not imply a maximality theorem or an infinite family.

7 Verification architecture

7.1 Test strata

The default command `npm test` executes four distinct strata.

1. **Contract and mutation tests.** These compare the TypeScript builder, browser builder, and canonical JSON example; deliberately altered records must be rejected.
2. **Mathematical numerical conformance.** These check the true seed, radii, extension lengths, symmetry, collinearity, center span, and Gram factorization at binary64 precision.
3. **Exact scalar certificate.** Sparse polynomial arithmetic verifies both scalar-map compositions and the canonical identities without floating arithmetic or third-party dependencies.
4. **Syntax and release inventory.** JavaScript and TypeScript sources are parsed, JSON documents are loaded, and every payload byte count and SHA-256 digest is compared with `RELEASE.json`.

The observed result for v1.1.0 is:

OMEGA Kernel Construct v1.1.0 full conformance: PASS
 35 payload files; 36 ZIP entries including the generated release manifest.

7.2 Deterministic release

The release builder sorts paths, fixes ZIP timestamps and permissions, records the byte length and SHA-256 of every payload, and computes an aggregate inventory digest over deterministic path-length-hash records. Two independent builds produced byte-identical archives. The source archive hash is printed on the title page.

7.3 What was not promoted to a pass

Two environment limits remain explicit.

- A standalone TypeScript compiler was not present in the verification environment. Node 24 parsed and executed the TypeScript source through its type-stripping runtime, but this release record does not claim a fresh `tsc -noEmit` pass for v1.1.0.
- The available cloud browser could not access the workspace localhost. Browser module syntax, export parity, and UI source were tested, but a rendered-browser pass is not claimed.

These are verification limits, not hidden successes. They do not alter the passing exact scalar certificate or the release-inventory result.

8 Integration rule

The production application already contains the authoritative `paper-vii-kernel.ts` evaluator and `Kernel3DView`. Integration must adapt evaluator output into `OMEGA_KERNEL_NODES/1.0` and call the strict validator before export. The display model may be used for visible separation but never as a source of normative values.

The architectural rule is therefore

Paper VII evaluator $\longrightarrow$ typed adapter and validator $\longrightarrow$ `Kernel3DView` and JSON export.

No second geometry evaluator is introduced. This preserves one proof-backed source for the geometry while allowing the interface, presentation, and export layers to be tested independently.

9 Conclusion

OMEGA Kernel Construct v1.1.0 turns a geometric construction into a precise, reproducible interface. The package fixes the canonical realization, preserves the exact seed independently of display exaggeration, makes optionality and nonimplication machine-checkable, and stops automated presentation exactly at kernel completion. It also closes the scalar chart change by proof and exact symbolic computation.

The resulting artifact establishes conformance of the implementation to the declared kernel interface. It does not replace Paper VII's Euclidean uniqueness argument, broaden the finite floor-map identity, modify frozen SRSD, derive cubature from the kernel, or supply an absent outer-structure axiom. Those boundaries are part of the result: they make the reproducible content strong enough to use without making it larger than the mathematics supports.

A Canonical node table

Table 4: Canonical normalized coordinates and claim types.

ID	Label	u -coefficient	v -coefficient	Claim
alpha	α	0	0	kernel
d	d	c	$-s$	kernel
b	b	c	$+s$	kernel
d1	d_1	c	$-(1+s)$	kernel
b1	b_1	c	$+(1+s)$	kernel
d2	d_2	c	$-(\varphi^2 + s)$	kernel
b2	b_2	c	$+(\varphi^2 + s)$	kernel
S	S	$c - \varphi^2$	0	optional
E	E	$c + \varphi^2$	$-(\varphi^2 + s)$	optional
L	L	$c + \varphi^2$	$+(\varphi^2 + s)$	optional

Under Equation (5), an entry (ξ_u, ξ_v) becomes $(x, y, z) = (\xi_v, -\xi_u, 0)$. This is the only Cartesian realization used by the canonical profile.

B Release file map

The complete release contains the following file groups. Individual byte lengths and SHA-256 values are authoritative in `RELEASE.json`.

Core source	Ten TypeScript modules under <code>src/</code> : constants, node contract and log, display and overlay models, screenplay and state machine, scalar interface, and the package export.
Schemas	Base and strict canonical node schemas plus the scalar interface schema.
Examples	The canonical true-epsilon unit node log and the machine-readable scalar interface record.
Demonstration	A dependency-free HTML, CSS, and JavaScript browser application.
Verification	Contract, mutation, parity, numerical mathematics, syntax, release, and exact scalar-certificate programs.
Documentation	README, verification record, kernel conformance, scalar interface, typed-network boundary, integration patch, and licenses.
Release tools	Package metadata, TypeScript configuration, deterministic builder, and per-file release manifest.

C References

References

- [1] M. A. Newman, *Newman's Quantized Omega Contraction Theorem*, 2026. doi:10.5281/zenodo.21830285.
- [2] M. A. Newman, *Ordered Euclidean Remainders for Exact Orbit Reconstruction and Parameter Intervals in Rational Floor Maps*, 2026. doi:10.5281/zenodo.21843485.
- [3] M. A. Newman, *Scale-Resolved State Dynamics: Foundational Closure Specification*, version 1.0, 2026. doi:10.5281/zenodo.21927248.
- [4] M. A. Newman, *Radial Icosahedral Sampling and Its First Harmonic Obstruction*, 2026. doi:10.5281/zenodo.22033917.
- [5] M. A. Newman, *OMEGA NAUGHT Ω_0* , 2026. doi:10.5281/zenodo.22048540.
- [6] M. A. Newman, *Exact Positive Icosahedral Cubature on Fixed Radial Class-I Grids*, 2026. doi:10.5281/zenodo.22068873.
- [7] M. A. Newman, *OMEGA Paper VII: Local Euclidean Kernel from a One-Arcsecond Seed*, manuscript version 1.6.0, 2026.
- [8] M. A. Newman, *OMEGA Kernel Construct*, software and conformance release version 1.1.0, 6 September 2026. Source archive SHA-256: bf9a02e1692d5ae46b907ac7b96e44483744824e6aedef12e83abe0e750ac7a39.