On the Integrated Ontology and Dynamics of Relativistic Quantum Mechanics, Relational Gravity and Cosmology

Dennis A. Kahan *

Posted Date: 6 March 2026

doi: 10.20944/preprints202405.2028.v8

Keywords: relativistic quantum mechanics; discrete 4D spacetime; relational gravity; planck domain; cosmology; unified evolution equation; explanatory depth

 doi:10.20944/preprints202405.2028.v8

Article

On the Integrated Ontology and Dynamics of Relativistic Quantum Mechanics, Relational Gravity and Cosmology

Dennis A. Kahan

Independent Researcher; dkahan@alumni.ucla.edu

Abstract

Foundational tensions between special relativity and quantum mechanics, together with conflicts between general relativity and quantum gravity, and unresolved cosmogonical and cosmological anomalies, block theoretical unification and limit explanatory depth. Based on ontological first principles rather than mathematical constructs, this analysis integrates a "discrete," background-independent, relativistic 4D spacetime with a physically co-located Planck Domain. Through a one-to-one identity, the Planck Domain mirrors the discrete spatial elements of 4D spacetime, enabling a single, unified set of physical laws across quantum and classical regimes. Under this framework, ontic single- and N-body quantum states evolve deterministically in 4D spacetime and collapse instantaneously in the Planck Domain. Current theoretical tensions between special relativity and quantum mechanics, including nonlocality, separability, time, simultaneity, total energy scaling, and probability conservation, are reappraised by replacing the Hilbert-space wavefunction with an ontic energy field and a single energy-based operator that governs both motion and gravitational response. The identical ontological framework and dynamical laws apply unchanged across general relativity and quantum gravity, recasting gravity as the relational dynamics of discrete energy rather than the coupling of the stress-energy and metric tensors, and reappraising the equivalence principle and the black hole information paradox. Cosmogonically, the same model re-examines the origin of 4D spacetime, accounting for near-homogeneity, isotropy, and low gravitational entropy without ad hoc assumptions, fine-tuning, or perturbative techniques, and provides ontological foundations for the cosmological constant and global energy conservation. Eight descriptive mathematical validations, derived from a unified evolution law, Planck Domain collapse rule, and the relational gravity law, support (but do not govern) the analysis: (i) the low-ℓ CMB TT shape generated from a $t = 0$ field with one global amplitude on power; (ii) CHSH correlations at Tsirelson's bound from collapse; and (iii–viii) hard-mass relational dynamics, highlighted by a tilted Earth–Moon orbit, a tilted hierarchical three-body system, and a high-energy Mercury–Sun analog, all sustained for 1000 orbits or inner orbits.

Keywords: relativistic quantum mechanics; discrete 4D spacetime; relational gravity; planck domain; cosmology; unified evolution equation; explanatory depth

1. Introduction

1.1. Foundational Tensions in Theoretical Physics

Resolving the tensions between special relativity (SR) and quantum mechanics (QM) and between general relativity (GR) and quantum gravity (QG) has proven so intractable that few theories comprehensively engage with them.

Neither SR nor GR addresses QM, and later efforts to integrate GR and QG and SR and QM remain largely unsuccessful. Quantum frameworks, including Schrödinger's wave mechanics, Heisenberg's matrix mechanics, and various formulations of Feynman's path integral rely on non-relativistic

 doi:10.20944/preprints202405.2028.v8

formulations, while most variations of the Copenhagen Interpretation [1], Bohmian mechanics [2], objective collapse theories (GRWf, GRWm, CSL) [3-5], MWI [6-8] and others [9-17] depend on the non-relativistic Schrödinger equation or Hilbert space representations.

Relativistic quantum field theories (QFT) typically assume a flat Minkowski spacetime, bypassing GR's curved framework. Approaches to GR-QG, including Causal Dynamical Triangulations [18], Asymptotic Safety in Quantum Gravity [19], and the Holographic Principle [20], focus on mathematical constructs without addressing the SR-QM tension.

String Theory [21] embeds SR and GR into higher-dimensional frameworks using Hilbert and Fock spaces to describe quantum states and interactions. However, it fails to address foundational challenges such as causality, locality, and the ontological basis of 4D spacetime. In contrast, Loop Quantum Gravity (LQG) [22] primarily operates within a 4D spacetime framework, quantizing spacetime into discrete spin networks and spin foams. Although more physically grounded, LQG struggles to reconcile probabilistic frameworks, physical observables, and the nature of time within 4D spacetime.

Sophisticated mathematical formulations across these theories often blur the line between physical ontology and abstract constructs. Semi-ontological approaches such as GRWf, GRWm, CSL, and Multi-Field Theories treat mathematical structures as ontic components, complicating their reconciliation with GR and SR. Frameworks like MWI lack clear mechanisms for reconciling quantum phenomena with GR and SR, while models based on Hilbert space, Fock space, matrix mechanics, or 3N configuration spaces often lead to unphysical conclusions. For example, while 4D spacetime alone cannot fully explain the dynamics of N-body quantum states, many models represent these states as evolving within non-physical, ultra-high-dimensional configuration spaces.

1.2. The DO Model

Rather than relying on a continuum manifold, a metric field, or a Hilbert-space state vector as primitive structures, the Dual Ontology model ("DO") integrates a discrete, background-independent 4D spacetime with a physically co-located Planck domain (the "Planck Domain"). Within this physical framework, each discrete spatial unit (Discrete Sphere) in 4D spacetime has a mirrored counterpart in the ontic Planck Domain, enabling a one-to-one correspondence between the domains [23-24]. Mathematically, the Planck Domain can be expressed as a $(3 \times N)$ space, where 3 represents the three dimensions of each Discrete Sphere in 4D spacetime and N represents the number of Discrete Spheres in the Planck Domain. [1]

Taken together, the Planck Domain and 4D spacetime form a single, tightly integrated $((3 \times N) + 3)$ physical structure that enables a unified, ontological, and dynamic model governed by a single set of physical laws applied consistently across scales and domains.[2] [3] Based on this unified structure, each quantum state is represented in 4D spacetime by an ontic energy field and in the Planck Domain by a single point whose formal dimension tracks the number of Discrete Spheres occupied by the state. A single energy-based operator governs both motion and gravitational response in 4D spacetime.

1.3. The Crisis of Explanatory Depth

The central unification problem in theoretical physics is not the absence of powerful formalisms. Rather, it is the absence of a single ontological and dynamic account that remains coherent when carried across different theoretical frameworks (see generally [25–27]). This analysis treats the impasse as a crisis of explanatory depth, sharpened by the absence of explanatory compression. Here, explanatory depth means mechanism plus scope: the paper presents explicit physical mechanisms and a single account spanning otherwise distinct phenomena. Explanatory compression means that this account is

[1] The Planck Domain is a physical $(3 \times N)$ space, not a mathematical 3N configuration space.

[2] Ontological concepts and their roles are defined in §2 below.

[3] Relativistic quantum state evolution in 4D spacetime and instantaneous quantum collapse in the Planck Domain are developed in §§3–5, the arrow of time in §6, relational gravity in §7, and cosmology in §8.

 doi:10.20944/preprints202405.2028.v8

carried by one ontology and one dynamics across scales and domains without regime-specific ad hoc assumptions, fine-tuning, or perturbative techniques.

The DO framework examines the following theoretical problems as a unified constraint system:

SR–QM tensions: 1) quantum entanglement, 2) nonlocality, 3) the instantaneity of quantum collapse, 4) causality, 5) the nature of time, 6) spacelike separation, 7) separability, 8) indeterminism, 9) quantum state emergence and annihilation, 10) quantum state localization, 11) probability conservation, 12) quantum tunneling, 13) relativity of simultaneity, 14) total energy scaling, 15) quantum nonattenuation and quantum exclusivity, 16) the Born Rule's relationship to SR, 17) the ontic status of N-body quantum states in 4D spacetime, and 18) the quantum-classical divide.

GR–QG tensions: 1) singularities, 2) regularization and renormalization, 3) background independence, 4) the nature of time, 5) nonlocality, entanglement, and instantaneity, 6) gravity's quantizability, 7) the equivalence principle, and 8) the black hole information paradox.

Cosmogonical and Cosmological anomalies: 1) 4D spacetime emergence, 2) near homogeneity and isotropy at $t = 0$, 2) the ontic status of gravity and time at $t = 0$, 4) the horizon and causality problem, 5) the flatness problem, 6) low gravitational entropy, 7) the ontological source of the cosmological constant, 8) the dark energy problem, 9) the hierarchy problem, 10) global energy conservation, 11) the arrow of time.

1.4. DO Outline

Section 2 establishes the ontological framework, detailing a discrete 4D spacetime and its ontologically distinct Planck Domain. The foundation is essential for resolving the SR-QM tensions, explored in Sections 3-5. These sections reconcile the deterministic, relativistic evolution of quantum states in 4D spacetime with their instantaneous collapse in the Planck Domain. Resolving this tension underpins the model's ability to address broader challenges, including GR-QG and relational gravity, the cosmological constant problem, the horizon problem, the hierarchy problem, and the quantum-classical divide.

The model's ontological and dynamic foundations also provide the basis for exploring quantum path irreversibility in Section 6. Grounded in the asymmetry between relativistic quantum state evolution in 4D spacetime and instantaneous collapse in the Planck Domain, the analysis explains the physical basis for the unidirectional arrow of time. Section 7 directly confronts the GR-QG tensions. Under the DO model, gravity is a relational phenomenon, inherently non-quantizable, and the Black Hole Information Paradox, as well as the equivalence principle, are explained ontologically and dynamically.

Section 8 investigates the transition of 4D spacetime at Heat Death to $t = 0$, based on the same discrete ontological structure and instantaneous collapse dynamics that govern quantum-state collapse in general. It re-interprets the horizon problem without resorting to inflationary, bouncing, or cyclic theories, identifies the ontological source of the cosmological constant, and resolves the hierarchy problem based upon the DO's ontological and dynamic principles. Section 9 offers a coda on the quantum-classical divide, and Section 10 synthesizes the model's contributions and outlines broader implications.

Appendix A presents the formal mathematical rules and constraints of the DO model. Appendix B presents eight descriptive validation results, and analysis derived entirely from the DO evolution law, the Planck Domain collapse rule, and the relational gravity law formalized in Appendix A, with full run artifacts provided in Appendix C. The validations are organized across four regimes: **Cosmogonical**: mapping a discrete $t = 0$ field to the low-ℓ CMB TT spectrum with one global amplitude on power; **Quantum**: recovering spin correlations and CHSH violation at Tsirelson's bound from collapse; **Low-Energy Orbitals**: tilted Earth–Moon two-body and hierarchical three-body systems, culminating in an Earth–Moon 1000-orbit run and a 1000-inner-orbit three-body coupled run with stable long-horizon behavior and invariant control; **High-Energy Orbitals**: a Mercury–Sun two-body analog with a 1-orbit baseline together with a 1000-orbit high-energy run ($\sim 5.8 \times 10^9$ updates) that preserve I_{DO}, angular

 doi:10.20944/preprints202405.2028.v8

momentum, and barycenter control while yielding a converged apsidal-advance diagnostic. Appendix D lists near–term tests that probe DO claims.

2. The Ontological Framework of the DO Model

2.1. Discrete Spheres

Under the DO model, Discrete Spheres are three-dimensional units of space that form the discrete substructure of both 4D spacetime and the Planck Domain (see [28–32]). Each Discrete Sphere is structurally invariant, possesses an identical shape and volume, and represents the smallest structural quantum of space.[4] Discrete Spheres are non-dynamical: they do not move, do not collapse, and do not undergo internal time evolution.

Each Discrete Sphere has an intrinsic 3D spatial identity. In discrete 4D spacetime, that identity is expressed by a unique set of x, y, z spatial coordinates. However, in the Planck Domain, these coordinate triples function only as Discrete-Sphere identifiers and do not generate spatial separation, distance, or any metric between spheres.

The dual presence of each Discrete Sphere in both 4D spacetime and the Planck Domain is referred to as the *Planck Identity*, which establishes a one-to-one identity and mapping between domains. Facilitated by the SOAN (Section 2.2), the Planck Identity preserves each sphere's intrinsic (x, y, z) identity across both domains.

2.2. The State of Absolute Nothingness

The concept of nothingness has long perplexed philosophers and scientists. Many Greek and Roman thinkers struggled with the concept of zero and an ontic the void.[5] Currently, the term "nothing" is often used figuratively to mean "not anything" rather than an ontic state of nonexistence. For example, in GR, "nothing," as in "not anything," colloquially describes what 4D spacetime expands into after its inception at $t = 0$.[6] In LQG, "nothing" refers to the absence of further spatial degrees of freedom below a minimum geometric scale rather than an ontic void [34]. "Nothing" has also been used to denote the absence of space and time [35].

An ontic State of Absolute Nothingness ("SOAN"), devoid of space and time, has no direct analogue in conventional theoretical physics. Because nothingness cannot be observed or measured, it is typically excluded from physical ontology. Rather than dismissing it, however, the DO framework treats the SOAN as a necessary ontological reference that bridges discrete 4D spacetime and the Planck Domain. Under the DO model, the SOAN provides the structural basis for the one-to-one correspondence between each Discrete Sphere in 4D spacetime and its Planck-Domain counterpart.

The SOAN's only defining attribute is onticness; it lacks all other physical properties. Since it excludes positive physical attributes, the SOAN is a passive ontological entity. It cannot be observed or measured, has no structure or boundaries, and is not governed by the laws of physics.[7] It is a non-spatial, non-temporal bridge that links 4D spacetime to the Planck Domain. Consequently, explanatory depth, rather than experimental testing, is fundamental to verifying the SOAN's ontological role in the DO model. From an explanatory depth perspective, the SOAN is fundamental to resolving tensions between SR and QM, and between GR and QG, and provides coherent explanations for long-standing cosmogonical and cosmological issues.

---

4 For illustrative purposes, Discrete Spheres have a volume of 4.22×10^{-105} m^3.

5 But see Lucretius [33] for an early work on the ontic status of the "void."

6 Technically, in standard GR, 4D spacetime is self-contained and does not expand into an external medium. Instead, metric expansion between gravitationally unbound objects in 4D spacetime increases the proper distance over time.

7 More specifically, the SOAN is devoid of space, time, dimension, boundary, size, structure, volume, gravity, energy, pressure, temperature, force, fields, ground states, vacuum states, virtual particles, quantum fluctuations, dynamic properties, frame of reference, matter, strings, information, mathematical entities, potentials, concepts, abstractions, consciousness, positive physical laws, possibilities, or entropy (see [36-40)].

 doi:10.20944/preprints202405.2028.v8

2.3. Discrete 4D spacetime

Unlike general relativity's conception of 4D spacetime as a continuous, differentiable manifold, the DO framework posits that spacetime consists of Discrete Spheres arranged in a dynamic nearest-neighbor structure. This structure supports background independence and stepwise evolution governed by the *Unified Evolution Equation* (UEE).[8] The nearest-neighbor structure imposes a discrete geometric constraint from which Lorentz invariance emerges only in the continuum limit.

2.4. The Planck Domain

The Planck Domain is the second ontological structure of the DO model. Like discrete 4D spacetime, the Planck Domain is composed of Discrete Spheres and the SOAN. "N" designates the number of Discrete Spheres that comprise the Planck Domain. Mathematically, the Planck Domain is represented by an N-tuple of ordered triples,

$$Q^N = \big((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_N, y_N, z_N)\big).$$

Each ordered triple $(x_i^{\mathrm{PD}}, y_i^{\mathrm{PD}}, z_i^{\mathrm{PD}})$ identifies the Discrete Sphere triple of a single Discrete Sphere's Planck-Domain representation but does not represent a Planck-Domain position or internal separation. Rather, they are the intrinsic Planck-Domain degrees of freedom used to represent N Discrete Spheres as a $(3 \times N)$ ontological structure.

The Planck Domain fundamentally differs from 4D spacetime [cf. 41]. It has no time dimension [42-43], no internal separation, distance, or metric, and therefore no conventional spatial volume, and the laws of GR and SR, the strong interaction, the electro-weak interaction, gravity, and thermodynamics do not govern it. Moreover, unlike mathematical spaces composed of mutually orthogonal vectors, the Planck Domain integrates N Discrete Spheres into a single *Planck Point*. For example, the 5.58×10^{186} Discrete Spheres that comprise the observable portion of 4D spacetime form a single Planck Point composed of $(3 \times 5.58 \times 10^{186})$ dimensions.

2.5. The Tightly Integrated DO Model

The $(3 \times \mathrm{N})$ Planck Domain and the 3 spatial dimensions of discrete 4D spacetime form the DO model's single, tightly integrated $((3 \times \mathrm{N}) + 3)$ physical structure [44]. Although the $((3 \times \mathrm{N}) + 3)$ structure may seem complex, its core is simple. Based on an ontic SOAN, which serves as a non-spatial, non-temporal bridge between the two domains, the DO framework links 4D spacetime and the Planck Domain into a single physical structure, forming a unified framework that exists in the same location as 4D spacetime. More proverbially, 4D spacetime does not exist "here," and the Planck Domain does not exist "there"; they co-exist in the same physical space. The explicit bijection between discrete 4D spacetime and the Planck Domain is enforced at the Discrete Sphere level by the Planck Identity, ensuring a one-to-one identity and mapping of quantum states across both domains.

Imagine, for example, that 4D spacetime consists of Discrete Spheres and that the SOAN exists within the interstices of these spheres.[9] Assume there are five Discrete Spheres: one each located on Venus, Mars, Jupiter, Sirius, and Polaris. In 4D spacetime, the Discrete Spheres are spatially separated, and each is represented by a set of x, y, z spatial coordinates.[10] However, since the SOAN exists in the interstices between Discrete Spheres, from the perspective of the Planck Domain, these five spheres are not separated by time, space, or volume. The five Discrete Spheres form a single, unified 15-dimensional point (3×5) in the Planck Domain, where 3 represents the three spatial dimensions of each sphere, and 5 represents the five spheres.

---

8 See Section 2.3 of Appendix A.

9 The term "interstices" in the DO does not imply a continuous geometric space between Discrete Spheres.

10 As used in the DO, the terms *location* and *position* do not imply an underlying metric space or continuum coordinate system.

 doi:10.20944/preprints202405.2028.v8

2.6. The Ontological Reality of Quantum States

The significance of the Planck Identity's one-to-one identity and mapping extends beyond the physical integration of 4D spacetime and the Planck Domain. Under the DO model, subatomic entities are ontic quantum states: a quantum state is a physically real distribution of intrinsic energy over Discrete Spheres in discrete 4D spacetime. [11] The Planck Identity ensures that ontic single and N-body quantum states, which cannot be fully described in 4D spacetime alone, are identified and mapped in both domains. In addition, dynamic changes in the physical characteristics of quantum states as they evolve in 4D spacetime are mirrored in the Planck Domain, and physical changes caused by the collapse of a quantum state in the Planck Domain are mirrored in 4D spacetime.

3. The DO and the Dynamics of Quantum States

3.1. The Dynamic Evolution of Quantum States

The analysis begins with the dynamic evolution of a single quantum state in a discrete 4D spacetime composed of Discrete Spheres and the SOAN. As a quantum state evolves, its energy occupies Discrete Spheres in 4D spacetime, and that occupied set is mirrored in the Planck Domain. The combination of a single Discrete Sphere and the portion of the quantum state's energy within that sphere is referred to as a *Bell Sphere*. Building on the Planck Identity, the *Bell Identity* is the one-to-one bijective mirroring between a Bell Sphere in 4D spacetime and its mirrored counterpart in the Planck Domain.

In 4D spacetime, the full set of Bell Spheres occupied by a quantum state constitutes its *Bell Field*, and in the Planck Domain, the same Bell Spheres constitute the quantum state's single *Bell Point* and the *Bell Energy Field* is the quantum state's intrinsic energy distributed across its Bell Field in 4D spacetime, and the *Bell Energy Point* is the Planck-Domain counterpart of that same per-sphere intrinsic energy distribution.

For example, the 1.92×10^{74} Bell Spheres that comprise an electron in the ground state of hydrogen simultaneously form the electron's Bell Field in 4D spacetime and its single $(3 \times 1.92 \times 10^{74})$-dimensional Bell Point in the Planck Domain. As a quantum state spreads in 4D spacetime, the number of Bell Spheres it occupies increases, and that same increase is mirrored as an increase in the number of Bell Spheres comprising the quantum state's Bell Point in the Planck Domain.

3.2. The Collapse of a Single Quantum State

The Bell Identity also mirrors the instantaneous collapse of a quantum state's Bell Energy Point in the Planck Domain with the collapse of its Bell Energy Field in 4D spacetime. Following the collapse of a Bell Energy Point, the number of Bell Spheres that comprise the quantum state's Bell Point is reduced. Simultaneously, the Bell Identity ensures that the decrease in the number of Bell Spheres comprising the quantum state's new Bell Point is mirrored by a reduction in the number of Bell Spheres forming the quantum state's new Bell Field in 4D spacetime. The Discrete Spheres do not collapse. For example, assume that quantum state A is placed within an impenetrable Box A. Quantum state A forms Bell Field A and Bell Energy Field A in 4D spacetime and Bell Point A and Bell Energy Point A in the Planck Domain. (Figure 1).

[11] Under the DO model, a quantum state is an ontic distribution of intrinsic energy over Discrete Spheres in discrete 4D spacetime, with a Planck-Domain counterpart that mirrors the same occupied Discrete Spheres under the Planck Identity. This framing is compatible with SR's mass–energy equivalence, $m = E/c^2$: the state's energy content is the physical basis for inertial mass. For alternative views on the ontic status of quantum states, see [45–53].

 doi:10.20944/preprints202405.2028.v8

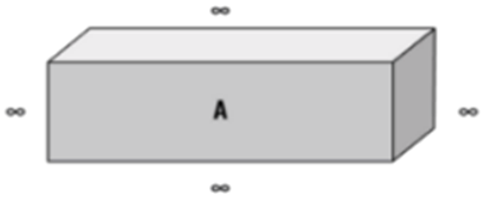


Figure 1. Impenetrable Box A.

As quantum state A spreads, the Bell Identity ensures that the increase in the number of Bell Spheres comprising Bell Field A is mirrored by a corresponding increase in the number of Bell Spheres comprising Bell Point A. The opening of Box A triggers the instantaneous collapse of Bell Energy Point A, causing an instantaneous reduction in the number of Bell Spheres that comprise Bell Point A. The Bell Identity ensures that the reduction is mirrored by an identical reduction in the number of Bell Spheres that constitute Bell Energy Field A (and therefore Bell Field A) in 4D spacetime. Quantum state A's Bell Energy Field is instantaneously generally localized within Box A, but SR has not been violated.

3.2.1. The Einstein–de Broglie Boxes Thought Experiment

The Einstein-de Broglie thought experiment further illustrates the dynamic evolution of a quantum state [54-57]. Quantum state B is generated, forming Bell Field B in 4D spacetime and Bell Point B in the Planck Domain. The quantum state is inserted into Box B. As it spreads, it occupies an increasing number of Bell Spheres in 4D spacetime and the Planck Domain (Figure 2).

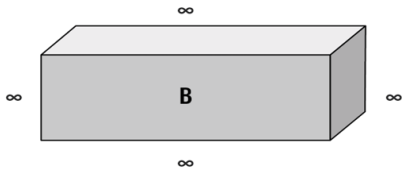


Figure 2. Einstein's Boxes - (Box B).

An impenetrable divider is inserted into Box B, creating Box C and Box D. The quantum state now forms two equal Bell Fields in 4D spacetime, Bell Field C and Bell Field D, and a single Bell Point in the Planck Domain, designated as Bell Point CD. Box C is sent to Princeton, and Box D is sent to Copenhagen (Figure 3).

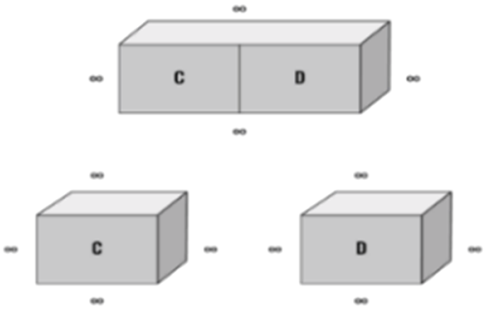


Figure 3. Princeton (Box C) – Copenhagen (Box D).

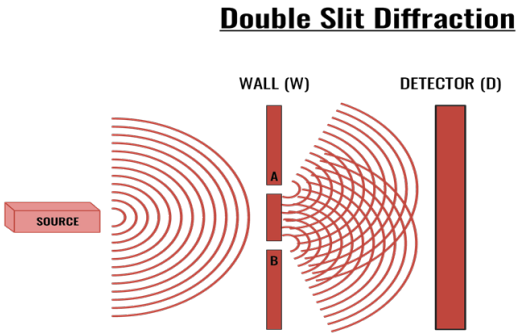
 doi:10.20944/preprints202405.2028.v8

Despite their separation, the Bell Identity ensures that the quantum state continuously forms Bell Fields C and D in 4D spacetime and Bell Point CD in the Planck Domain.

The opening of Box C or Box D triggers the collapse of Bell Energy Point CD, reducing the number of Bell Spheres that form the new Bell Point of the quantum state. The reduction is mirrored in 4D spacetime. If the quantum state is found in Box C, it forms a generally localized Bell Energy Field C in Box C and Bell Energy Point C in the Planck Domain, while Bell Field D and Bell Point D cease to exist. Conversely, if the quantum state is found in Box D, it forms a generally localized Bell Energy Field D in Box D and Bell Energy Point D in the Planck Domain, and Bell Field C and Bell Point C no longer exist. Regardless of which box is opened first, the process is the same.

3.2.2. The Double-Slit Experiment

In the double-slit experiment, individual quantum states are directed at Wall (W), which has two narrow Gaussian slits (A) and (B). Due to the narrowness of the slits, every quantum state that passes through slit (A) or slit (B) diffracts, spreading as spherical Bell Fields toward Detector D (Figure 4).



Double Slit Diffraction

Figure 4. Double-slit experiment with diffraction and interference patterns.

As a quantum state diffracts through slits (A) and (B), its Bell Field splits into two separate fields in 4D spacetime: Bell Field A and Bell Field B.[12] In the Planck Domain, the fields remain unified as a single Bell Point AB. A detection flash at Detector D indicates that Bell Energy Point AB has collapsed. The collapse reduces the number of Bell Spheres comprising the quantum state's Bell Point in the Planck Domain. The Bell Identity ensures that the reduction is mirrored in 4D spacetime, localizing the quantum state's Bell Energy Field to one of the diffracted paths. Following the quantum collapse of Bell Energy Point AB, the Bell Energy Field in 4D spacetime localizes to either Bell Energy Field A or B; the other branch's Bell Energy Field ceases to exist. The interference pattern observed on Detector D arises from cumulative quantum collapses, reflecting the probabilistic outcomes of individual quantum collapses.

3.2.3. A Which-Way Experiment

Which-way experiments compound the theoretical complexities of the double-slit experiment. The following which-way experiment has been modified by including a proton in an empty box at the center of Wall (W) (Figure 5).[13] The proton is positively charged, and each electron fired toward Wall (W) is negatively charged. Slit (A) flashes if the proton is attracted toward slit (A) and slit (B) flashes if it is attracted toward slit (B).

---

[12] If an ontic quantum state occupies both diffracted paths prior to collapse, then any charge associated with that state is likewise distributed across the same occupied Discrete Spheres. Following collapse, that charge is localized to the same branch as the post-collapse quantum state. See generally [58-59].

[13] The which-way monitoring experiment is based upon the example presented in [36, pp. 14-16].

 doi:10.20944/preprints202405.2028.v8

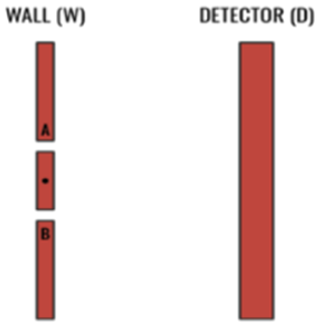


Figure 5. Which Way Experiment with Proton.

Under the DO model, as the quantum state spreads in 4D spacetime, it continuously forms a Bell Field in 4D spacetime and simultaneously forms a single Bell Point in the Planck Domain. If slit (A) flashes, the Bell Energy Point instantly collapses, reducing the number of Bell Spheres that make up its new Bell Point; the Bell Identity mirrors this as a reduced Bell Energy Field. The quantum state is localized at slit (A). The analysis remains the same whether slit (B) flashes or slit (A).

Once the quantum state is generally localized at either slit (A) or slit (B), it again spreads toward Detector (D). However, because the quantum state collapses at either slit (A) or slit (B), but not both, no interference pattern forms at Detector (D).

3.3. N-Body Quantum States and The Bohm-EPR Thought Experiment

The Bohm version of the EPR experiment highlights issues related to the dynamic evolution of an N-body quantum state in 4D spacetime and its collapse in the Planck Domain. A pair of electrons is prepared in the singlet state. The singlet state forms Bell Fields E and F in 4D spacetime and a single Bell Point EF in the Planck Domain. Quantum state E is sent to Princeton, and quantum state F is sent to Copenhagen (Figure 6). Testing equipment is configured to conduct a z-axis Stern-Gerlach experiment on either Bell Field E or F.

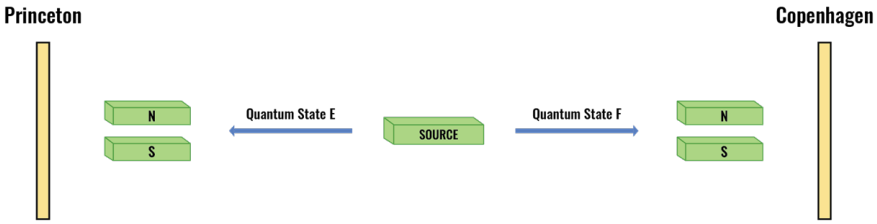


Figure 6. Bohm Version of the EPR Experiment.

As Bell Fields E and F spread dynamically in 4D spacetime, the number of Bell Spheres that comprise their respective Bell Fields increases, as does the number of Bell Spheres that comprise Bell Point EF. The Stern-Gerlach experiment, conducted along the z-axis in either Princeton or Copenhagen, triggers the collapse of Bell Energy Point EF. The collapse instantaneously reduces the number of Bell Spheres that formerly composed Bell Point EF, and the reduction is mirrored by Bell Field E and Bell Field F, respectively. Bell Energy Point EF forms two independent Bell Energy Points designated as Bell Energy Point E and Bell Energy Point F. Bell Energy Point E shares a one-to-one mapping and identity with Bell Energy Field E, and Bell Energy Point F shares a one-to-one identity and mapping with Bell Energy Field F.

 doi:10.20944/preprints202405.2028.v8

Following the instantaneous collapse, Bell Energy Point E and Bell Energy Point F form a factorized state rather than a singlet state.[14] The single state descriptor, Φ_{EF}, transitions to a factorized state composed of two independent state descriptors, Φ_E and Φ_F. This physical change is represented as $\Phi_{EF} \rightarrow \Phi_E \otimes \Phi_F$, and the corresponding Bell Energy Fields E and F are generally localized. Whether quantum state E is found along the z spin-up or z spin-down axis, quantum state F's spin is the opposite. SR has not been violated.

4. Physical Implications of the DO Model

4.1. Indeterminacy

Indeterminacy typically means that a quantum system has a determinable property without a specific determinate value [60, pp.72-107]. In a singlet state along the z-axis, spin is a determinable property, with z spin-up and z spin-down as determinate values. While the conventional quantum formalism represents this singlet state abstractly as the state vector $\psi = 1/(\sqrt{2})(\uparrow z_1 \downarrow z_2 - \downarrow z_1 \uparrow z_2)$, the DO model provides a direct ontological description. The two quantum states, z_1 and z_2, form a single, unified system described by the state descriptor Φ_{z1z2}, with no determinate z-spin for either component prior to collapse; a singlet anti-correlation holds. The Bell Identity ensures that this system forms a single Bell Point in the Planck Domain and two corresponding Bell Fields in 4D spacetime.

After instantaneous collapse, the system transitions to a factorized state and the spins of z_1 and z_2 become determinate; for collapse along the z axis, the outcomes are opposite: one z-up, the other z-down, and which component is up is outcome-contingent. This physical transition from a unified state to two independent states is represented symbolically as $\Phi_{z1z2} \rightarrow \Phi_{z1} \otimes \Phi_{z2}$. Each quantum state now forms its own Bell Point associated with its respective localized Bell Field in 4D spacetime.

4.2. Quantum State Emergence and Annihilation

Quantum state emergence and annihilation challenge the applicability of the non-relativistic Schrödinger equation, which is formulated for systems with a fixed number of quantum states and does not account for processes involving the creation or annihilation of quantum states [10]. Relativistic quantum field theory (QFT) addresses these variations, but the DO model offers a unique solution, representing quantum states as physical entities in both 4D spacetime and the Planck Domain.

Under the DO framework, the Bell Identity links the Bell Spheres that comprise a quantum state's Bell Field in 4D spacetime and its Bell Point in the Planck Domain. During quantum annihilation, the collapse of a quantum state's Bell Energy Point transfers the observables of the quantum state to another system, eliminating the Bell Energy Point and Bell Energy Field of the original state. During quantum emergence, a quantum state forms a new Bell Field in 4D spacetime and a corresponding Bell Point in the Planck Domain.

4.3. Physical Triggers

In the DO model, *Physical Interactions* in 4D spacetime are caused by one or more of the three traditionally labeled Fundamental Forces: electromagnetism, the strong interaction, and the weak interaction.[15] Based on the Bell Identity, a Physical Interaction in 4D spacetime is mirrored in the Planck Domain, where the quantum state's Bell Energy Point collapses instantaneously.

The DO model does not identify the precise Physical Interaction that induces quantum state collapse. Nevertheless, it provides a structured framework for examining how physical triggers induce collapse. Within the DO framework, each Physical Interaction is a localized event in time and space, with its frequency influenced by factors such as temperature and spatial positioning (see generally [61]). Local temperature and location within the Sun affect the rate of quantum state collapse. Humans can initiate or influence the timing and location of Physical Interactions. For example, a scanning tunneling microscope can be used to control and precisely vary the rate of electron collapse. However,

[14] The term 'factorized' denotes DO separability in 4D spacetime; it is not a Hilbert-space tensor product.
[15] See Section 7 regarding gravity.

 doi:10.20944/preprints202405.2028.v8

Physical Interactions are independent of human consciousness, ambient noise, and universal processes [but see 62-63].

4.4. Quantum State Localization

The Bell Identity mirrors the collapse of a quantum state's Bell Energy Point in the Planck Domain as a simultaneous reduction in the number of Bell Spheres that comprise its Bell Energy Field in 4D spacetime. Under the DO collapse rule, the reduction must be to a connected subset of Bell Spheres occupied before collapse. Collapse cannot generally localize a quantum state to an arbitrary region of 4D spacetime; it generally localizes the Bell Energy Field to a connected subset of its pre-collapse Bell Field, and the excluded branch of the Bell Energy Field ceases to exist.

The DO model does not set a specific post-collapse size for a quantum state's Bell Energy Field in 4D spacetime. The localization scale is set by the Physical Interaction that triggers collapse and by the quantum state's physical characteristics; collapse energy and momentum conditions may also influence localization.

4.5. Time and Instantaneous Collapse

Neither the Planck Domain nor 4D spacetime supports the concept of instantaneous collapse independently. The Planck Domain lacks a time dimension and, aside from collapse, does not support dynamic movement. In contrast, 4D spacetime has dynamic movement and a time dimension constrained by SR.

When a Physical Interaction in 4D spacetime occurs, the physical interaction is mirrored in the Planck Domain, causing the quantum state's Bell Energy Point to collapse instantaneously. The Bell Identity ensures that the collapse is mirrored by a reduction in the number of Bell Spheres that comprise the quantum state's Bell Energy Field in 4D spacetime. Because the collapse of a Bell Energy Point is instantaneous, it is mirrored in the three spatial dimensions of 4D spacetime, with no movement along the time dimension.

4.6. Quantum Tunneling

Although commonly described as 'quantum tunneling,' the appearance of a quantum state on the opposite side of a classically forbidden finite barrier is not described in DO as continuous propagation of the state through the barrier within 4D spacetime (see generally [64]). In many traditional interpretations, the probability of a quantum state appearing on the other side of an impenetrable barrier is based on the Schrödinger equation and the exponential decay of the quantum state within the barrier.

Under the DO model, the event is not tunneling but rather the instantaneous collapse of the quantum state's Bell Energy Point in the Planck Domain. When a quantum state's Bell Energy Point undergoes instantaneous reduction, the Bell Identity ensures a corresponding reduction in the number of Bell Spheres that constitute the quantum state's new Bell Energy Field in 4D spacetime. Although the quantum state is localized on the other side of the barrier, it does not "tunnel" through, and SR is not violated.

4.7. The Born Rule Revisited

The DO model diverges fundamentally from the Born Rule and its mathematical interpretation of wave-function collapse as a probability density for continuous variables. In the DO framework, a quantum state's presence in 4D spacetime is characterized by its Bell Energy Field. The field represents the physical distribution of a quantum state's total intrinsic energy across the discrete Bell Spheres it occupies. Since the DO collapse rule holds that collapse must occur within a discrete subset of these Bell Spheres, the DO model posits that the likelihood of the quantum state becoming generally localized within any specific sub-region of its pre-collapse Bell Energy Field is directly proportional to the quantum state's energy residing in that sub-region (cf. [65]). This approach grounds probability in the tangible distribution of the state's physical energy, rather than in abstract mathematical amplitudes.

 doi:10.20944/preprints202405.2028.v8

For example, in the case of quantum tunneling, the probability of the quantum state localizing on the far side of a classically impenetrable barrier corresponds to the proportion of its total energy distributed in its Bell Spheres beyond that barrier before collapse.[16] Such an outcome is a discrete probability event, not the result of integrating a continuous probability density. The Bell Identity underpins the probability conservation of this process and resolves issues such as the "four tails problem" [66] but leaves open whether the collapse of a quantum state is fully deterministic [67].

5. Resolving the Tension Between SR and Quantum Mechanics

The apparent incompatibility between SR and QM is often framed in terms and concepts derived from 4D spacetime. Despite their usefulness, common terms such as spacelike separated, non-separability, entanglement, instantaneous, local, non-local, and complex concepts such as the relativity of simultaneity and total energy scaling have unintentionally magnified a theoretical and experimental conflict that does not exist.

5.1. Spacelike Separated

The term spacelike separated is based on a 4D spacetime structure with three spatial dimensions and one temporal dimension. The term is directly related to the concepts of space and time, the theory of SR, and the spatial distance between two or more events outside of one another's light cones. Nevertheless, the term loses meaning in relation to a Bell Point, which has no internal spatial separation, metric, or volume.

5.2. Non-separability

Einstein was among the first to raise concerns regarding separability in theoretical physics. His primary concern related to two assumptions underlying his argument for incompleteness: that spatially separated systems are ontic states, and that physical effects in spacelike-separated systems cannot propagate faster than the speed of light.[17] [18]

In the DO model of 4D spacetime, a singlet state along the z-axis is described by the state descriptor Φ_{z1z2}. This system is non-separable, representing an ontic physical entity rather than an abstract mathematical concept. A non-separable DO singlet state has three key attributes: 1) the spatial separation of the z_1 and z_2 Bell Fields, 2) their temporal separation, and 3) the existence of a single system, whose corresponding single Bell Energy Point ontologically enforces non-separability.[19]

In the Planck Domain, which has no internal spatial separation, metric, or volume, a Bell Point exists as a single, non-separable entity (but see [42, 72]). The Bell Energy Point is mirrored via the Bell Identity to the Bell Spheres that form the quantum state's Bell Energy Field(s) in 4D spacetime. The non-separability of a Bell Energy Point in the Planck Domain does not violate SR.

5.3. Instantaneous, Superluminal, and Faster than Light

In QM, the terms *instantaneous*, *superluminal*, and *faster than light* often describe the collapse of a quantum state in 4D spacetime. Following quantum state collapse, these terms are used to describe the quantum state's role in 1) communication, 2) signaling or the absence of signaling, 3) information transmission, and 4) matter and energy transfer.

However, under the DO model, terms such as *instantaneous* describe the physical collapse of a Bell Energy Point in the Planck Domain rather than a collapse in 4D spacetime. Following the collapse, the

16 Following quantum collapse, the probability of finding a quantum state in a generalized location sums to one.

17 Einstein's primary concern was not with non-separability per se but with the possibility that non-separability implied a violation of special relativity [68, pp. 172–173] (see also [69, pp. 88–89]).

18 Einstein also questioned whether spatially separated quantum states in 4D spacetime had an independent reality [70].

19 A single Bell Energy Point enforces non-separability while preserving remote-setting invariance: changing a device orientation at one wing does not alter the marginal outcome distribution at the other wing. Bell-inequality violation arises from single-outcome coupling: Bell Energy Point collapse fixes both wings' outcomes and is then mirrored into S(4D) via the Bell Identity. Device orientations are treated as freely chosen at the modeling level. See Appendix A §§2.6.3–2.6.5 and §3.2. See also footnote 20.

 doi:10.20944/preprints202405.2028.v8

reduction in the number of Bell Spheres that comprise a quantum state's new Bell Point in the Planck Domain is mirrored by a reduction in the Bell Spheres that comprise the quantum state's generally localized Bell Field in 4D spacetime. The process is instantaneous, but SR is not violated [cf. 71].

5.4. The Quantum Connection

In 4D spacetime, quantum discrimination refers to a quantum state's ability to maintain an exclusive connection, excluding all other quantum states, and unattenuated denotes the strength (or lack of attenuation) of a quantum state's connection [69, pp. 21-22]. The terms are often used to denote the connection between spacelike-separated singlet states. Discrimination and non-attenuation also imply an instantaneous, continuous connection that violates the maximum speed of light.

In the DO model, the bijective Bell Identity ensures that all dynamic changes to Bell Spheres are mirrored in both 4D spacetime and the Planck Domain. The mirroring process ensures quantum discrimination and non-attenuation without violating SR.

5.5. Bell's Theorem

Bell's inequality theorem sets a bound on the strength of correlations that any Bell-locally-causal theory can produce in spin tests [73-77].[20] For singlet pairs, quantum mechanics predicts correlations beyond that bound, and the corresponding spin experiments measure those correlations. The singlet statistics are incompatible with Bell local causality. More broadly, any theory that matches the experimental singlet correlations must violate Bell's local-causality condition.

The DO model locates Bell Energy Point collapse in the Planck Domain, where there is no time and no internal spatial separation, metric, or volume. The Bell Identity mirrors the physical collapse in 4D spacetime as a localized 4D update, without violating SR. Although collapse is extra-spatio-temporal, the mirrored 4D outcome statistics violate Bell's local-causality condition and reproduce Bell-type correlations.

5.6. The Relativity of Simultaneity

The SR-QM tension extends to the relativity of simultaneity. SR holds that 1) all inertial reference frames (frames moving at a constant speed relative to one another) are equally valid, and 2) the speed of light in a vacuum is invariant for all observers in these frames. Consequently, the relativity of simultaneity implies that a) whether two spatially separated events occur simultaneously depends on the observer's frame of reference, and b) observers in different frames may conclude that the same event happened at different times.

For spacelike-separated electrons in the singlet state along the z-axis, Φ_{z1z2}, the collapse of the z_1 electron causes the simultaneous collapse of the z_2 electron. Because the relativity of simultaneity suggests that the order of cause (collapse of z_1) and effect (collapse of z_2) depends on the observer's frame of reference, simultaneous collapse appears to challenge SR, implying a violation of Lorentz Invariance and a preferred frame [69, 71].

The DO model resolves the issue by treating the collapse of the system Φ_{z1z2} as an event beyond 4D spacetime. For a singlet state along the z-axis, it is irrelevant whether z_1 or z_2 is the site of the first Physical Interaction or whether they are spacelike separated. The Bell Identity ensures that an experiment on either quantum state in 4D spacetime is simultaneously conducted on the quantum state's single Bell Point in the Planck Domain. Moreover, the identity reflects the instantaneous collapse of the Bell Energy Point as a reduction in the Bell Spheres comprising the now generally localized Bell Fields of both z_1 and z_2 in 4D spacetime. The singlet state becomes a factorized state, and although the collapse is instantaneous, SR remains intact.

[20] Conditioned on a complete specification of the relevant physical state in the joint past of two space-like separated detections, Bell local causality requires that the outcome statistics on each wing depend only on the local analyzer setting and the past state, not on the distant setting or the distant outcome. Measurement settings are assumed to be statistically independent of that past state [74]. See also footnote 19.

 doi:10.20944/preprints202405.2028.v8

5.7. Total Energy Scaling

The instantaneous nature of quantum state collapse appears to challenge Einstein's theory of total energy and momentum in SR [69]. The theory posits that the total energy of a body moving relative to an observer increases as its velocity accelerates. As an object approaches the speed of light, its relativistic kinetic energy theoretically approaches infinity, although SR limits its speed. In quantum mechanics, momentum is typically used instead of velocity. Accordingly, as the momentum of a quantum state increases, so does its associated total energy. If collapse were interpreted as a process in which a body in 4D spacetime actually reaches or exceeds the speed of light, the total energy required would be infinite.

While the collapse of a quantum state's Bell Energy Point is instantaneous, it is a physical event external to 4D spacetime. The instantaneous collapse occurs in the Planck Domain, and the Bell Identity mirrors this as a reduction in the number of Bell Spheres comprising the quantum state's Bell Point and Bell Field(s). Consequently, the reduction in Bell Spheres in the Bell Field is also instantaneous. However, the process does not result in an increase in the quantum state's energy in either 4D spacetime or the Planck Domain.

6. Quantum Path Irreversibility and The Arrow of Time

The Bohm version of the EPR experiment (see Figure 6) demonstrates why quantum path reversibility following a collapse is impossible [78, pp. 150–162; 79; 80; cf. 81]. Assume that two quantum states, z_1 and z_2, are in the singlet state along the z-direction, forming a single, unified system described by Φ_{z1z2}. The Bell Field of z_1 is on Mars, and the Bell Field of z_2 is on Earth. The singlet state is separated by 225 million km, and its spin is indeterminate. In the Planck Domain, this single system Φ_{z1z2} corresponds to a single Bell Point z_1z_2.

Following the instantaneous collapse of Bell Energy Point z_1z_2, the system transitions to a factorized state represented as $\Phi_{z1z2} \rightarrow \Phi_{z1} \otimes \Phi_{z2}$. The Bell Identity ensures that the reduction in the number of Bell Spheres comprising Bell Point z_1z_2 is linked to the simultaneous reduction in Bell Spheres that comprise Bell Fields z_1 and z_2 in 4D spacetime. Bell Field z_1 is generally localized on Mars, and Bell Field z_2 is generally localized on Earth. The z_1 and z_2 quantum states are now separable, each forming its own distinct Bell Point (Bell Point z_1 and Bell Point z_2). If z_1 is spin-up, z_2 is spin-down, and vice versa. The spin of the respective quantum states is now determinate.

Before quantum state collapse, 1) z_1 is on Mars, z_2 is on Earth, and z_1 and z_2 are separated by 225 million km, 2) the singlet state forms Bell Point z_1z_2 in the Planck Domain, and 3) the singlet state is non-separable, and its spin is indeterminate. Following the instantaneous collapse of Bell Energy Point z_1z_2, 1) z_1 is generally localized on Mars, z_2 is generally localized on Earth, and z_1 and z_2 remain separated by 225 million km; 2) the former singlet state is now factorized; 3) z_1 forms Bell Field z_1 on Mars and Bell Point z_1 in the Planck Domain; 4) z_2 forms Bell Field z_2 on Earth and Bell Point z_2, in the Planck Domain; 5) the factorized state is now separable; and 6) the spins of quantum states z_1 and z_2 are determinate even if they are unknown.

Reversing the path would require retracing the collapse, but the unified Bell Point z_1z_2 no longer exists; therefore, path reversibility is impossible. Z_1 now forms Bell Field z_1 on Mars and Bell Point z_1 in the Planck Domain, and z_2 now forms Bell Field z_2 on Earth and Bell Point z_2 in the Planck Domain. Because Bell Field z_1 and Bell Field z_2 are now generally located on Mars and Earth, each quantum state must travel at least 112.5 million km before it can form a unified state again. Even if there is an infinitely small chance that the z_1 and z_2 quantum states spread and once again form Bell Point z_1z_2, when the singlet state collapses, path reversibility becomes impossible.

The DO's ontological structure, the asymmetric laws governing the dynamic motion of quantum states in 4D spacetime, and their collapse in the Planck Domain provide the physical

 doi:10.20944/preprints202405.2028.v8

basis for the arrow of time in 4D spacetime. Without an instantaneously reversible path, the arrow of time following collapse for all quantum states and those of the proverbial egg can only move in a single temporal and spatial direction.[21]

7. GR, Relational Gravity, Background Independence, and Black Holes

7.1. Discretization and Singularities

The DO model replaces GR's assumption of a continuous, differential 4D spacetime manifold with a discrete 4D spacetime composed of Discrete Spheres. Critically, in the context of GR–QG, the physical discretization of 4D spacetime resolves several of the mathematical difficulties encountered by the Einstein Field Equation (EFE), including the cosmological singularity and black hole singularities, and regularization. [22]

7.1.1. Cosmogonical and Black Hole Singularities

Specific mathematical solutions to EFE, including the FLRW model, predict a cosmological singularity at $t = 0$, where the density, pressure, and energy density become infinite. Similarly, the Schwarzschild solution to the EFE, based on a spherically symmetric, uncharged, and non-rotating mass, mathematically defines the conditions under which a curvature singularity forms at the center of black holes. In both cases, the EFE's lack of a physical mechanism to impose a cut-off at a minimum volume mathematically leads to infinite energy, mass, and pressure [82].

Notwithstanding the significant differences between cosmological and black hole singularities, discretization based on the minimum discrete size of Discrete Spheres provides a minimum volume cutoff that resolves the infinities in both cases. The discretization sets limits on the maximum possible frequencies and wavelengths for energy, creating a physical upper bound on energy density. [23]

7.2. Regularization and Renormalization

In QFT, zero-point energies from quantum fluctuations are formally summed over all modes, producing vacuum energy densities that diverge at high energies. Regularization introduces a mathematical cutoff to control these infinities, and renormalization redefines physical parameters to remove them. Even after this procedure, the predicted value can exceed the observed energy density by as much as 10^{120}. In continuum approaches, these enormous contributions are often absorbed into a constant "vacuum term" that is adjusted so the remaining, effective energy density matches observations. In the DO model, Discrete Spheres function as a physical cutoff for ultra-high-energy modes, preventing ultraviolet energy from diverging [cf. 83]. With no infinities to absorb, and with the large-scale constant that plays the role of Λ fixed rather than tuned (see Section 8), there is no vacuum term to adjust, and renormalization is unnecessary. Discreteness thereby eliminates both the divergence and its underlying cause, yielding physical energy densities far below the mathematical prediction of the zero-point energy summation.

7.3. Time

In QFT's Hamiltonian and path integral formalisms (and the explicitly non-relativistic Schrödinger equation), time typically remains an external, fixed parameter. Most QFT theories assume a flat Minkowski spacetime, with time as an absolute Newtonian construct. In contrast, GR treats time as a dimension within 4D spacetime, where the rate at which time passes depends on local mass, energy, and pressure, as reflected in spacetime's curvature. Time dilation, gravitational redshift, and lensing follow from this relational view of time.

The DO model's approach to time diverges from both perspectives by integrating the Planck Domain, which lacks a time dimension. Except for quantum collapse, the Planck Domain has no

[21] Quantum irreversibility is dependent upon collapse. In the absence of collapse, reversibility is not precluded under the DO.
[22] See Appendix A.
[23] Discreteness suppresses but does not eliminate the high-energy scale associated with very small volumes.

 doi:10.20944/preprints202405.2028.v8

independent dynamic movement. Single and N-body quantum states evolve entirely within 4D spacetime, under a discrete dynamics whose large-scale behavior is consistent with SR. [24] As a result, quantum state dynamics respect SR constraints, including the universal speed limit, time dilation, and relativistic energy-momentum scaling, with gravity expressed as curvature in a discrete 4D spacetime. While Discrete Spheres set the smallest unit of time at approximately the Planck time, time remains a dynamic concept shaped by SR rather than QM.

7.4. Relational Gravity

Attempts to quantize gravity with continuum-only, background-metric programs have thus far failed [84]. The EFE describe a local, relativistic interaction compatible with SR, but quantum collapse is typically regarded as a non-local phenomenon with instantaneous changes to a quantum state's wave function [85]. Hypothetical gravitons are problematic if they must simultaneously adhere to SR constraints and mediate instantaneous changes in location and momentum.[25] Additionally, because standard QM holds that the position of a quantum state is undefined before collapse, gravitational coupling is ambiguous. In the specific case of an N-body quantum state, it is unclear whether gravity couples to the aggregate state as a whole or to each constituent quantum state.

Instantaneous collapse also appears to conflict with the relativity of simultaneity and with the limits of total energy and momentum in SR.[26] Many QM interpretations represent the collapse with a Dirac delta function $\delta(x - x0)$, implying infinite localization, infinite momentum uncertainty, and potential black hole formation [85].

Under the DO model, however, all quantum states evolve in a discrete, 4D spacetime within which large-scale behavior is compatible with SR. Bell Energy Points collapse instantly in the Planck Domain, where the laws of 4D spacetime do not apply. The collapse of a Bell Energy Point localizes the quantum state in 4D spacetime rather than collapsing to a delta function, ensuring finite localization and avoiding the issues raised by $\delta(x - x_0)$. Because the Bell Identity mirrors a quantum state's Bell Spheres across both domains, changes in energy and sphere count are reflected instantaneously by a quantum state's Bell Energy Point and Bell Energy Field(s). Consequently, the Bell Energy Fields of quantum states in 4D spacetime and their Bell Energy Points in the Planck Domain both carry the information (energy, spin, and momentum, for example) needed to reproduce the observed quantum behaviour associated with the strong interaction, the weak interaction, and electromagnetism.[27]

Unlike these forces, gravity is not encoded in the quantum states of Bell Energy Points and does not exist in the Planck Domain. The Planck Domain has no internal spatial separation, metric, or time evolution, so it contains no gravitational geometry and no gravitational degrees of freedom.[28][29] Gravity is a 4D spacetime phenomenon arising from the relational configuration of Bell Spheres and the energy distribution of quantum states in 4D spacetime.

---

[24] See the "Unified Evolution Equation, Section 2.3 of Appendix A."

[25] Gravitons do not exist in the DO model.

[26] See Sections 5.6 and 5.7.

[27] Critically, in the DO framework, the standard particle physics terms "photons", "W/Z bosons", and "gluons" are not ontologically fundamental mediators arising from independent gauge fields. The corresponding quantized interaction signatures remain measurable in 4D spacetime but are treated in the DO as discrete interaction outcomes realized as discrete changes in the Bell Energy Fields of the participating quantum states, instead of exchanges of fundamental bosonic quanta. Electromagnetism is re-conceptualized as discrete electromagnetic observables on Bell Spheres in 4D spacetime, while the strong and weak interactions are treated as short-range, adjacency-limited contact interactions. The energy associated with these three interaction classes contributes to the total energy budget of Bell Spheres as part of the Bell Energy Fields of the participating quantum states, and it is this same Bell Energy Field energy, rather than any separate mediator-field energy term, that sources the UEE's relational gravitational response.

[28] Under the DO framework, the equivalence principle is direct: the same conserved energy that defines inertia also sources gravity, so bodies in the same gravitational field accelerate alike regardless of composition. The Bell Identity mirrors a state's total energy between the Planck Domain and 4D spacetime, and gravitational dynamics in 4D couple only to that energy, with $m = E/c^2$. In short: "Energy distribution shapes relational curvature; relational curvature governs motion." See Appendix A for the operator form.

[29] For an N-body quantum state, the equivalence principle applies component-wise at each location; there is no single, global free-fall law for the composite quantum state.

 doi:10.20944/preprints202405.2028.v8

In the DO framework, spacetime geometry is determined by the collective energy distribution and relational configuration of Bell Spheres in 4D spacetime, not by intrinsic quantum-state properties. Accordingly, gravitational effects depend only on the Bell Sphere energy distribution and relational structure in 4D spacetime, not on quantum-state observables in 4D spacetime or the Planck Domain. Quantum observables are encoded in the Bell Energy Point and mirrored into 4D spacetime; gravity is governed by localized relational structure in 4D spacetime via the UEE's coupling of energy to curvature. Gravity cannot be quantized as an independent field and has no independent degrees of freedom because its effects arise from relational configuration rather than intrinsic Bell Energy Point properties.

7.5. Background Independence

In GR, spacetime geometry is dynamic, but it is still modeled on a continuous manifold. Most quantum field theories, on the other hand, assume a fixed background geometry, such as flat Minkowski space. This fixed framework imposes a separation between geometry and the distribution of matter and energy, requiring the geometry to be specified in advance.

The DO model is background-independent from the outset [86]. From $t = 0$, discrete 4D spacetime exists and is formed from the relational configuration of Discrete Spheres, with no external metric or coordinate structure imposed from outside the model. Its geometry is determined internally by the system's physical state, and because geometry and energy distribution co-evolve, there is no fixed metric structure, and no external background is needed to define motion or curvature.

7.6. The Black Hole Information Paradox

The black hole paradox is based on the apparent conflict between two principles: 1) the QM principle that information is never lost, and 2) Stephen Hawking's 1975 semi-classical premise that information falling into a black hole is eventually lost through Hawking radiation. The QM principle is closely related to the probability conservation and reversibility of quantum processes. Hawking's key premise is that a black hole loses mass and energy by emitting thermal radiation at its event horizon. Because the quantum state of matter falling into the black hole is inaccessible beyond the event horizon, and thermal radiation is presumed to carry no specific information, the process implies that the information is permanently lost.[30]

Under the DO framework, when a quantum state's Bell Field(s) falls into a black hole, the quantum state and its information are no longer accessible in 4D spacetime. However, as long as the quantum state does not collapse, it is still represented by its Bell Field(s) within the black hole and by its single Bell Point in the Planck Domain. As the quantum state dynamically evolves within the black hole, any changes to the content or number of Bell Spheres in the Bell Field are instantaneously mirrored by its Bell Point.

The transition to thermal radiation, triggered by a physical interaction inside the black hole, causes the instantaneous collapse of the quantum state's Bell Energy Point. In 4D spacetime, the collapse causes the quantum state's transition to radiation via the Hawking process. Although the collapse is irreversible, the Bell Identity preserves the quantum state's conserved quantities across the transition to Hawking radiation. Detailed quantum state information is not preserved after collapse, but its conserved quantities persist, so no fundamental information loss occurs.

8. Quantum Cosmology, the Cosmological Constant, and the Hierarchy Problem

Under the DO framework, the same collapse process that governs quantum states also governs 4D spacetime's transition from Heat Death to $t = 0$. The collapse explains 4D spacetime's generalized localization with extreme but finite conditions at $t = 0$, accounts for the horizon and flatness problems,

[30] The DO's ontological and dynamic framework challenges the no-hair result: in GR, a stationary black hole is externally characterized only by total mass, electric charge, and angular momentum; correspondingly, in the semiclassical Hawking picture, the emitted radiation depends only on these macroscopic parameters rather than detailed microstate information [87].

 doi:10.20944/preprints202405.2028.v8

identifies the ontological source of Λ without Dark Energy, and frames the Hierarchy Problem as a consequence of discreteness and relational gravity.[31]

8.1. The FLRW Model of 4D spacetime at Heat Death

4D spacetime's status as open, closed, or flat or nearly flat is based on the FLRW derivation of the EFE. Current datasets indicate $k = 0$, so 4D spacetime is flat or very nearly flat. In turn, flatness implies that 4D spacetime's total energy density equals the critical density ($\rho = \rho_c$). As Heat Death is approached, the total energy density tends toward ρ_Λ, and the temperature asymptotically approaches zero.

At Heat Death, 4D spacetime's spatial geometry, energy density, pressure, and temperature are very nearly homogeneous and isotropic, and 4D spacetime is close to thermodynamic equilibrium. 4D spacetime has no large-scale structures, is extremely widely dispersed, and little work occurs at the macrostate level. With little heat flow near equilibrium, gravitational clumping does not proceed macroscopically, and 4D spacetime is in a state of near-maximal gravitational entropy. Changes may still occur at the microstate level.

8.2. The FLRW Model of 4D spacetime at $t = 0$

Mathematical attempts to describe 4D spacetime at $t = 0$ support different conclusions regarding its physical status. Under specific conditions, the EFE and Friedmann equations describe a singularity at $t = 0$ caused by the divergence of energy density, pressure, and spacetime curvature. Other treatments preserve a homogeneous and isotropic 4D spacetime at very large scales, where matter, radiation, and the cosmological constant govern the dynamics.

The ΛCDM model does not describe 4D spacetime's physical status at $t = 0$. Nevertheless, the CMB and related datasets[32] indirectly indicate that immediately after $t = 0$, 4D spacetime was in a hot, dense state characterized by extreme energy density, pressure, and temperature [90–91]. Approximately 380,000 years after $t = 0$, CMB temperature anisotropies varied by ~ 1 part in 10^5, indirectly suggesting that very near $t = 0$, 4D spacetime was nearly homogeneous and isotropic with small inhomogeneities and anisotropies. These observations are also consistent with a nearly flat 4D curvature [92].

8.3. The DO Model

8.3.1. 4D Spacetime

Under the DO framework, discretization resolves the infinities created by the EFE's continuous, differentiable 4D spacetime manifold by imposing a physical upper bound on the maximum possible frequencies and minimum wavelengths for energy. Consequently, at $t = 0$, 4D spacetime is discrete and generally localized, with extreme but finite energy density, pressure, and temperature.

Although the metric expansion of 4D spacetime, described by the scale factor $a(t)$, is often described colloquially as an "internal stretching" rather than a stretching into nothing, the DO model posits that 4D spacetime expands into a pre-existing substructure comprised of Discrete Spheres and the SOAN. As shown in Sections 8.4 and 8.7 below, this distinction provides a physical basis for a uniform cosmological constant Λ derived from the inherent energy density of evenly distributed Discrete Spheres ($\rho_{DS} = \rho_\Lambda$), explains the increasing total energy budget of 4D spacetime as it expands into additional Discrete Spheres (cf. [93]) and obviates the need for Dark Energy based on a vacuum-energy density that exceeds the observed Λ by $\sim 10^{120}$.

---

31 For alternative cosmologies, see [88]. See also [89].

32 Including Type Ia Supernovae Observations, Hubble's Law and Redshift Observations, Baryon Acoustic Oscillations, Galaxy Redshift Surveys, Stellar Evolution Models, and Globular Cluster Age Estimates.

 doi:10.20944/preprints202405.2028.v8

8.3.2. The Planck Domain

The Planck Domain has no time dimension, no internal spatial separation, and no metric, and therefore no spatial volume. Aside from collapse, the Planck Domain does not support dynamic movement, and 4D spacetime's physical laws do not govern it.

Since the Bell Identity mirrors, sphere-by-sphere, the Universal Bell Energy Field in 4D spacetime to the Universal Bell Energy Point in the Planck Domain, the mirrored subsets carry the same total intrinsic energy content across both domains. This mirroring carries per-sphere quantum energy content of the Universal Bell Energy Field into the Planck Domain as timeless state content, without introducing any Planck-Domain notion of distance, volume, pressure, temperature, or any spatial density field.

In addition, 4D spacetime's total energy density per Discrete Sphere (ρ) at any instant t equals the Planck Domain's total energy density per Discrete Sphere (ρ_{pd}) for the Planck-Domain subset that mirrors 4D spacetime at that instant ($\rho = \rho_{\mathrm{pd}}$). Because 4D spacetime's total energy density equals the critical density at $t = 0$ and at Heat Death ($\rho = \rho_{\mathrm{c}}$), the Planck Domain's total energy density per Discrete Sphere for the same mirrored subset also equals the critical density ($\rho_{\mathrm{pd}} = \rho_{\mathrm{c}}$).

8.4. Heat Death and the Collapse of the Universal Bell Energy Point

The same instantaneous collapse process that applies to single and N-body quantum states also applies to the Universal Bell Energy Point at or near Heat Death.[33] In the Planck Domain, the Universal Bell Energy Point collapses, and that collapse is mirrored as the generalized localization of 4D spacetime's Universal Bell Energy Field at $t = 0$. At Heat Death, matter and radiation asymptotically approach zero energy density, pressure, and temperature. Because the Universal Bell Energy Field at Heat Death is nearly homogeneous and isotropic, the Universal Bell Energy Point mirrors the same near-uniform per-sphere quantum energy content. The post-collapse $t = 0$ state remains nearly homogeneous and isotropic.[34] Generalized localization at $t = 0$ yields extreme but finite energy density, pressure, and temperature.

Universal Bell Energy Field collapse also accounts for 4D spacetime's shift from near-maximal gravitational entropy at Heat Death to near-zero gravitational entropy at $t = 0$. Near-equilibrium conditions at Heat Death preclude additional macrostate work. Generalized localization and subsequent expansion reset the gravitational-entropy baseline. Since gravity is relational under the DO model, gravitational effects are present at $t = 0$ as a direct consequence of the relational configuration of Bell Spheres. Neither gravity nor time originates from symmetry breaking. Finally, Discrete Spheres do not collapse during the Heat Death to $t = 0$ transition. Their size, shape, and intrinsic properties remain invariant, and continue to form the stable substructure of both 4D spacetime and the Planck Domain.

8.5. The Horizon Problem and Causality

The inability to explain the exceptionally high homogeneity and isotropy of 4D spacetime at $t = 0$ is the source of the horizon problem. Given the constraints of the speed of light c, spacelike separated regions of 4D spacetime following $t = 0$ could not have been in causal contact [95]. The problem is sharpened by CMB data indicating that approximately 380,000 years after $t = 0$, temperature variations were ~ 1 part in 10^5. The near-zero curvature parameter ($\Omega_{\mathrm{k}} = -0.0005 \pm 0.0005$) measured at the last scattering surface is consistent with a nearly flat early 4D spacetime curvature and aligns with the DO account that the Heat-Death collapse carries flatness forward to $t = 0$.

The DO's resolution is premised on four factors: (1) the DO replaces an initial singularity with a discrete 4D spacetime; (2) the near-maximal homogeneity and isotropy of the Universal Bell Energy Field at Heat Death arises internally from expansion and cooling over extremely long time scales; (3)

[33] Although the precise interaction that triggers collapse is unknown, it is a 4D-spacetime Physical Interaction.

[34] Roger Penrose estimated the odds of achieving a nearly homogeneous and isotropic 4D spacetime at $t = 0$ as approximately $10^{10^{123}}$ to 1 [94]. Since the near-maximal homogeneity and isotropy and near-zero gravitational entropy at $t = 0$ emerge intrinsically from Heat Death and the DO's integrated ontological and dynamic structure, the DO odds approach one.

 doi:10.20944/preprints202405.2028.v8

the instantaneous collapse of the Universal Bell Energy Point yields generalized localization of the Universal Bell Energy Field at $t = 0$; and (4) because collapse occurs in the Planck Domain, generalized localization is mirrored simultaneously across 4D spacetime, precluding the introduction of density variations, directional biases, or breaks in causal continuity.[35]

8.6. The Flatness Problem

The flatness of 4D spacetime's spatial curvature, and its sensitivity to minor deviations from flatness ($k = 0 \rightarrow k = \pm 1$) at or near $t = 0$, is known as the flatness problem. Because k is inferred through Ω, where $\Omega_{\text{total}} = \rho_{\text{total}}/\rho_{\text{c}}$, the total energy density of 4D spacetime at or near $t = 0$ must be extraordinarily close to the critical density ρ_{c} to yield spatial flatness.

While the DO model does not provide a physical basis for why the universe has the particular total energy density it has, it avoids reliance on ad hoc assumptions, fine-tuning, or perturbative techniques to maintain flatness (cf. [100]). Under the DO framework, the near-flat state at Heat Death—where $\rho \approx \rho_{\text{c}}$—is carried forward: the collapse of the Universal Bell Energy Point in the Planck Domain is mirrored as generalized localization of the Universal Bell Energy Field at $t = 0$, preserving the near-flat condition so that $\Omega_{\text{total}} \approx 1$ at $t = 0$.

8.7. The Cosmological Constant, Global Energy Conservation and Dark Energy

Under the EFE, the cosmological constant Λ is expressed as a uniform energy density $\rho_\Lambda = \frac{\Lambda c^2}{8\pi G}$ with an equation of state $w = -1$ (negative pressure). Under the DO model, the uniform ρ_Λ is not a vacuum-energy density. Rather, it is identified ontologically as a very small, constant intrinsic energy (CIE) carried uniformly, per Discrete Sphere, across 4D spacetime ($\rho_{\text{DS}} = \rho_\Lambda$). Discrete Spheres are structurally inert and do not interact with matter or radiation, so this uniform contribution produces no local curvature effects and functions only as a global expansion term.

As 4D spacetime expands into a pre-existing substructure of Discrete Spheres[36], each additional Discrete Sphere brings the same CIE into 4D spacetime, increasing 4D spacetime's total Λ-sector energy budget without violating global energy conservation. Because matter and radiation dilute while ρ_Λ remains constant, the Λ-sector share of the total energy budget grows as more Discrete Spheres come within 4D spacetime's bounds, clarifying the observed increase in the expansion rate. In this framework, Λ is fixed by the ontic status and uniform distribution of Discrete Spheres, obviating the need for Dark Energy based on a vacuum-energy density that exceeds the observed Λ by $\sim 10^{120}$ (cf. [101]).

8.8. The Hierarchy Problem

The vast energy differences between the electroweak scale ($\sim 10^2$ GeV) and the Planck scale ($\sim 10^{19}$ GeV) pose a challenge to conventional frameworks, where large quantum corrections destabilize the electroweak scale. The DO model resolves the issue through two elements: (1) a discrete 4D spacetime composed of irreducible Discrete Spheres of fixed finite volume, and (2) a relational gravity that decouples gravitational energy from the electroweak scale. Discretization provides a natural ultraviolet bound at the Planck scale and removes the divergence mechanism that drives electroweak-scale instability. Gravitational decoupling prevents Planck-scale effects from coupling into the intrinsic quantum properties of electroweak-scale states, removing the source of high-energy corrections to the Higgs scale. Moreover, because the energy density of Discrete Spheres ρ_Λ is uniform, it drops out of the relational operator locally and appears only in the global expansion term with an equation of state $w = -1$. Since ρ_Λ is not a component of a Bell Energy Point in the Planck Domain, it exerts no influence on the Higgs scale or other electroweak parameters.

---

[35] The anisotropies and inhomogeneities at Heat Death and $t = 0$ function as "seeds" for later large-scale structure (cf. [96–99]).

[36] These Discrete Spheres are the same Discrete Spheres that constitute the Planck Domain; expansion increases the 4D spacetime subset of Discrete Spheres, not the Planck Domain.

 doi:10.20944/preprints202405.2028.v8

9. Coda: The Quantum-Classical Divide

Finally, the DO framework highlights the ontological and dynamic separation between physical systems in the Planck Domain and a discrete 4D spacetime. The Planck Domain lacks a time dimension and spatial volume and is not governed by the laws of 4D spacetime. Its only source of dynamic movement is collapse. In the Planck Domain, a quantum state's single Bell Energy Point represents its quantum observables (including energy, spin, and momentum), but a Bell Energy Point does not support relational properties, such as the chairness of a chair or the aliveness of a cat.

In contrast, 4D spacetime has a time dimension, three spatial dimensions, and volume, and is fully governed by its physical laws, including relational gravity.[37] Although a quantum state's observables are represented by its Bell Energy Point and mirrored through its Bell Energy Field, unlike the Planck Domain, 4D spacetime also supports relational properties such as the solidity, shape, and color of a cat, as well as aliveness. As with gravity, the relational properties between Bell Spheres in 4D spacetime emerge from 4D spacetime's ontology, dynamics, and laws, rather than from the observables of a quantum state's Bell Energy Point.

For instance, in the Schrödinger cat experiment, assume the cat is initially alive, and a radioactive atom's quantum state has an instantaneous collapse probability $P_{\text{coll}}(t)$. The collapse of the atom's Bell Energy Point in the Planck Domain is mirrored by the Bell Identity into 4D spacetime as an immediate change in the quantum state's Bell Energy Field. The atom's post-collapse state participates in a Physical Interaction with a lethal device. Through a series of additional, interrelated Physical Interactions and Bell Energy Point collapses, Physical Interactions between the cat and the lethal device alter the cat's 4D relational status. The 4D readout of the cat's status changes from "Alive" to "Dead."

The cat's status is now irreversible; there is no physical path back to "Alive." Conversely, if collapse never occurs, the cat remains "Alive". In either case, before, during, and after the experiment, the cat's alive-or-dead status is based on relational interactions under the laws of 4D spacetime. Quantum state observables do not determine the cat's alive-or-dead status [cf. 102].

In a discrete 4D spacetime, all quantum states evolve deterministically between collapses. Determinism extends to the superposition of quantum state observables, such as energy, spin, and momentum, between collapses.[38] However, in all cases, the collapse of a Bell Energy Point introduces probability by determining which subset of Bell Spheres becomes localized in 4D spacetime. Moreover, under the DO, special experimental primitives are not required: collapse is a fundamental physical process in the Planck Domain.[39]

10. Conclusions

The ontological and dynamic premises of the DO model have profound theoretical implications. Built on a substructure of Discrete Spheres and the SOAN, the DO's integrated ((3 × N) + 3) ontology replaces GR's continuous 4D spacetime manifold with a discrete 4D spacetime and high-dimensional state spaces with an ontic (3 × N) Planck Domain. Consequently, the DO model supports a single ontological and dynamic framework and a set of physical laws that unify GR, SR, and QM, resolve the GR–QG tension, recover the arrow of time, identify the cosmological constant Λ, resolve hierarchy tensions, and account for 4D space time's Heat Death and instantaneous emergence at $t = 0$. Accordingly,

[37] Gravity does not exist in the Planck Domain, nor does it mediate the collapse of a Bell Energy Point. Gravity arises purely from relational configurations in 4D spacetime. Its curvature depends on the relational properties of mass, energy, and pressure, and not on the observables of a quantum state's Bell Energy Point.

[38] Before and after collapse, the quantum state is an ontic energy distribution across Bell Spheres in 4D spacetime, mirrored by a single Bell Energy Point in the Planck Domain. In the DO, superposition refers to a quantum state's observable content; it does not apply to location, because location is the definite 4D energy distribution itself, not an indeterminate alternative among positions.

[39] The DO also explains the ontic status of the "quantum connection" (Section 5.4) and its weakness: even extremely weak Physical Interactions (Section 4.3) can trigger Bell Energy Point collapse in the Planck Domain, mirrored as Bell Energy Field localization in 4D spacetime. Neither the Planck Domain nor Bell Energy Points possesses an independent physical structure or dynamics that counteract Physical Interactions.

 doi:10.20944/preprints202405.2028.v8

the DO treats SR–QM tensions, GR–QG tensions, and cosmogonical and cosmological anomalies as a unified constraint system addressed by one ontology and one dynamics.

Critically, the SOAN provides the ontological basis for the Planck Identity's one-to-one mapping of the Discrete Spheres that simultaneously form 4D spacetime and the Planck Domain. In turn, the Bell Identity and Bell Spheres mirror the physical attributes of a quantum state, including its energy, across both domains. Significantly, quantum states evolve deterministically in 4D spacetime under the Unified Evolution Equation, collapse occurs in the Planck Domain and the Bell Identity mirrors each collapse outcome into 4D spacetime without violating SR.

The DO framework employs a single energy-based operator that governs both motion and gravitational response. The model eliminates infinities associated with singularities through discretization, resolves the black hole information paradox by preserving quantum state information in the Planck Domain, and explains gravity as a local, relational phenomenon generated by the same quantum-state energy that sets inertia. The model provides a physical explanation for the irreversible arrow of time in 4D spacetime and redefines the cosmological constant Λ as the intrinsic energy density of Discrete Spheres. The model also resolves 4D spacetime's homogeneity, isotropy, horizon, and flatness problems based on its instantaneous transition from Heat Death to $t = 0$.

While quantum states are mirrored across domains, classical properties such as a cat's aliveness are defined by the relational ontology and dynamics of 4D spacetime. Classical determinism and quantum probability are not mutually exclusive; rather, they arise from a single, unified framework.

Discretized physical laws support the DO framework while preserving background independence and causal structure as well as SR's physical constraints. Constructed from the core ontological primitives of Energy and the SOAN,[40] the DO framework yields the discrete structure of 4D spacetime, its mirroring in the Planck Domain via the Planck and Bell Identities, and a single operator-level dynamics linking quantum evolution, collapse, and relational curvature.

[40] The ontic status of Energy and the SOAN turns one of the greatest philosophical questions on its head. The question is not, "Why is there something rather than nothing?" Rather, it is, **"Why is there something AND nothing?"**

 doi:10.20944/preprints202405.2028.v8

11. Mathematical Summary

This section is a compact mathematical summary of Appendices A and B. It states the minimal DO rule set needed to interpret the validations, discloses fixed realization choices and fitted quantities, and reproduces the headline validation outcomes and key summary tables. Appendix A contains full definitions and derivations. Appendix B contains full validation methodology and results. Appendix C contains executable scripts and complete run transcripts.

Interpretive rule (mathematics is descriptive). All mathematics in this Summary is descriptive bookkeeping for the DO ontology. Symbols, sets, and operators represent DO primitives and their allowed behaviors in $S^{(4D)}$ and $S^{(PD)}$; the equations do not introduce additional ontic entities beyond the DO inventory.

Correspondence fence. Continuum/SR correspondence statements are confined to Appendix A, Sec. 2.9 and are invoked only for realizations with $w \in \mathcal{W}_\Delta \subset \mathcal{W}$. Outside Sec. 2.9, no Laplacian-symbol or SR-dispersion claims are made.

One ontology and one dynamics across domains (mathematical statement).

- **Single ontology:** Discrete Spheres (DS) constitute $S^{(4D)}$ and $S^{(PD)}$, with ontological identity asserted by the Planck Identity.
- **Domain population:** N **is the total number of Discrete Spheres;** $S^{(4D)}$ **consists of a subset of** $\mathrm{DS}_{\mathrm{total}}$ **at each tick, while** $S^{(PD)}$ **comprises all of** $\mathrm{DS}_{\mathrm{total}}$**.**
- **Single operator (Appendix A form):** $\hat{K}_{\mathrm{eff}} = \hat{K}_{\mathrm{kin}} + \hat{K}_{\mathrm{grav}}$**.**
- **Single dynamics:** deterministic between-collapse evolution in $S^{(4D)}$ by the Unified Evolution Equation (UEE) tick map, and probabilistic collapse in $S^{(PD)}$ by the collapse operator, mirrored into $S^{(4D)}$ on the same tick by the non-dynamical bijective Bell Identity g.
- **No auxiliary structure:** structurally incompatible with Hilbert space, Fock space, 3N configuration space, matrix mechanics, and the Schrödinger equation; no continuous variational symmetries/gauge potentials/Noether identities; no continuum metric field or Newtonian/EFE potential inserted into the operator; no ad hoc assumptions, fine-tuning, or perturbative expansions.
- **Audit discipline:** every validation uses explicit gates; conserved-quantity auditing is primary (the DO invariant I_{DO}), while secondary diagnostics (e.g. $E_{\mathrm{kin}} + U$, sometimes printed as `K+U`) are reported as texture only.

 doi:10.20944/preprints202405.2028.v8

Table 1. Translation Table: Standard Constructs → DO Objects.

Standard Construct	DO Replacement Object(s)	Operational Statement
Wavefunction ψ	State descriptor $\Phi^{(4D)}(t)$ on $S^{(4D)}$; Bell Field (support) $O(t) \subset S^{(4D)}$; Bell Energy Field $e_\alpha(t) = \lvert\Phi_\alpha^{(4D)}(t)\rvert^2$	A quantum state represented by its state descriptor is ontic.
Unitarity / Bell-sum	Between-collapse tick evolution acting on $\Phi^{(4D)}(t)$ (and thus on $e(t)$); DO-native conserved audit scalar(s)	Bell-sum conservation between collapses.
Entanglement	Unified N-body Bell Energy Point in $S^{(PD)}$ with mirrored $S^{(4D)}$ Bell-Field representations	Bell-type correlations originate in PD collapse structure and Bell-Identity mirroring; no Hilbert-space tensor-product structure is assumed.
Undefined Measurement Postulate	Physical Interaction (PI) in $S^{(4D)}$ as Physical Trigger; Bell Energy Point in $S^{(PD)}$; bijective Bell Identity g	Collapse acts in $S^{(PD)}$ and is mirrored on the same tick as Bell Energy Field localization in $S^{(4D)}$; "measurement" is legacy terminology for one class of Physical Triggers.
Metric gravity / curvature	Relational exchange operator $\hat{K}_{\text{grav}}$ on $S^{(4D)}$ with weights $w_{ab} = w(d_{ab})$; hard-mass limit uses $\gamma(d)$ in the support-index force law	Curvature is relational exchange on the Bell-sphere network; no background metric; no potential-times-Φ term.

Notation (C-family). In this Summary, $\mathcal{C}$ denotes the fixed link set on $S^{(4D)}$, C denotes the collapse operator acting in $S^{(PD)}$, and C_ℓ denotes TT bandpowers in Validation I.

12. DO Mathematical Core Used by All Validations (Appendix A)

12.1. Domain structure and identity maps

Ontological domains. $S^{(4D)}$ is the discrete 4D spacetime domain supporting tick evolution. $S^{(PD)}$ is the Planck Domain supporting collapse. The domains are structurally distinct.

Planck Domain structural constraints. $S^{(PD)}$ is timeless and metric-free; it carries no internal spatial separation and supports no internal position observable.

Index set and total DS set. Let $\mathcal{I}$ denote the Discrete Sphere index set with $|\mathcal{I}| = N$, and define $\text{DS}_{\text{total}} := \{\text{DS}_i : i \in \mathcal{I}\}$.

Planck Identity (DS identity across domains).

$$\text{DS}_i \equiv P_i \qquad (\forall i \in \mathcal{I}). \tag{1}$$

Bell Identity g (bijective mirroring of dual state representations). Single-state and N-body forms:

$$\Phi^{(4D)}(t) \overset{g}{\longleftrightarrow} \phi^{(PD)}. \tag{2}$$

$$\{\Phi_n^{(4D)}(t)\}_{n=1}^{N_{\text{bodies}}} \overset{g}{\longleftrightarrow} \Phi_{N\text{-body}}^{(PD)}. \tag{3}$$

12.2. Between-collapse evolution (UEE tick map)

Fixed link set and exchange weights. Let $\mathcal{C} = \{(a,b) : w_{ab} > 0\}$ be the fixed link set on $S^{(4D)}$, with neighborhoods $\mathcal{N}(a) = \{b \neq a : w_{ab} > 0\}$. Define the symmetric combined exchange weight on each link by

$$q_{ab} := \eta_{ab} + w_{ab}, \qquad q_{ab} = q_{ba} \geq 0, \qquad (a,b) \in \mathcal{C}, \tag{4}$$

and the per-link tick angle by

$$\theta_{ab} := \Delta t\, q_{ab}, \qquad \theta_{ab} = \theta_{ba}. \tag{5}$$

 doi:10.20944/preprints202405.2028.v8

Fixed schedule (realization data). Fix a deterministic decomposition of $\mathcal{C}$ into disjoint matchings $\{\mathcal{C}^{(r)}\}_{r=1}^{R}$, with one tick applying the elementary link update on all links in exactly one matching. A realization fixes a schedule $r(K) \in \{1, \ldots, R\}$; the deterministic periodic schedule below is one admissible choice:

$$r(K) := 1 + \big((K-1) \bmod R\big). \tag{6}$$

Elementary link update (two-site exchange). For a link $(a, b) \in \mathcal{C}$ define

$$s_{ab} := \frac{\Phi_a + \Phi_b}{2}, \qquad \delta_{ab} := \frac{\Phi_a - \Phi_b}{2}, \qquad \delta_{ab}^{+} := e^{i\theta_{ab}}\, \delta_{ab}, \tag{7}$$

and set

$$\Phi_a^{+} := s_{ab} + \delta_{ab}^{+}, \qquad \Phi_b^{+} := s_{ab} - \delta_{ab}^{+}. \tag{8}$$

UEE tick map. Tick $K \to K+1$ applies the elementary link update in parallel on all links in $\mathcal{C}^{(r(K))}$ only, yielding

$$\Phi(t_{K+1}) = \mathcal{U}^{(r(K))}\big[\Phi(t_K)\big], \qquad t_K = t_0 + K\Delta t. \tag{9}$$

Bell-sum conservation (between collapses). The tick map preserves the hosted (non-CIE) intrinsic energy (Bell sum):

$$\sum_a |\Phi_a(t + \Delta t)|^2 = \sum_a |\Phi_a(t)|^2. \tag{10}$$

12.3. Causal bookkeeping bound and long-wavelength SR correspondence (compact statement)

Between collapses, one tick updates only one matching of disjoint links. In the nearest-neighbor tick-map realization (exchange links restricted to $d_{ab} = 1$ on the underlying DS neighbor graph used for SR bookkeeping), influence propagates by at most one hop per tick. Under the DO unit map (one hop $\leftrightarrow \ell_0$, one tick $\leftrightarrow \tau_0$), the SR-facing invariant speed is

$$c = \frac{\ell_0}{\tau_0}. \tag{11}$$

For finite-range exchange realizations whose link set includes links out to hop cutoff R_w, the realization-level domain of dependence expands at most R_w hops per tick (Appendix A, Sec. 2.4.3). The long-wavelength correspondence target is SR kinematics and free-propagation dispersion at scales large compared to the realization range, without postulating a background manifold, a continuum PDE, or a microscopic Lorentz symmetry as primitive structure.

12.4. Collapse operator in $S^{(PD)}$ and probabilities (mirrored into $S^{(4D)}$)

Admissible outcomes (single state). Let $O(t_{\mathrm{pre}}) = \operatorname{Supp} \Phi^{(4D)}(t_{\mathrm{pre}})$ and define samples $e_\alpha(t_{\mathrm{pre}}) = \left|\Phi_\alpha^{(4D)}(t_{\mathrm{pre}})\right|^2$. An admissible outcome is any connected subset $O' \subseteq O(t_{\mathrm{pre}})$ with $O' \in \mathrm{Conn}(\mathcal{C})$. In this Summary, "generally localized in $S^{(4D)}$" means $\operatorname{Supp} \Phi^{(4D)}(t_{\mathrm{post}}) = O'$ and $e_\alpha(t_{\mathrm{post}}) = 0$ for all $\alpha \notin O'$.

Collapse probability (energy fraction).

$$P(O') = \frac{\sum_{\alpha \in O'} e_\alpha(t_{\mathrm{pre}})}{\sum_{\alpha \in O(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}})} \qquad (O' \in \mathrm{Conn}(\mathcal{C})). \tag{12}$$

Probability evaluation is taken with respect to a specified admissible outcome family $\{O'_j\} \subset \mathrm{Conn}(\mathcal{C})$ that is pairwise disjoint and satisfies $\bigsqcup_j O'_j = O(t_{\mathrm{pre}})$; collapse selects exactly one member O'_j, and $P(O'_j)$ equals its pre-collapse hosted-energy fraction.

The pre and post tags refer to $S^{(4D)}$ tick labels and do not introduce time in $S^{(PD)}$.

 doi:10.20944/preprints202405.2028.v8

Energy preservation at collapse (single rescaling). Let

$$E_{\mathrm{QS}} := \sum_{\alpha \in O(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}}), \qquad E_{O'}(t_{\mathrm{pre}}) := \sum_{\alpha \in O'} e_\alpha(t_{\mathrm{pre}}). \tag{13}$$

Define

$$\lambda(O') := \frac{E_{\mathrm{QS}}}{E_{O'}(t_{\mathrm{pre}})}, \qquad e_\alpha(t_{\mathrm{post}}) := \begin{cases} \lambda(O')\, e_\alpha(t_{\mathrm{pre}}), & \alpha \in O', \\ 0, & \alpha \notin O', \end{cases} \tag{14}$$

so that

$$\sum_{\alpha \in O(t_{\mathrm{post}})} e_\alpha(t_{\mathrm{post}}) = E_{\mathrm{QS}}. \tag{15}$$

12.5. Relational gravity operator and the hard-mass classical limit

Hard-mass regime (orbit validations). Dispersive Bell Energy Fields are not used to model orbital motion; orbital validations (III–IV) are conducted only in the singleton-support hard-mass regime (Appendix A, Sec. 2.4), where each macroscopic body is represented by a single occupied Bell Sphere per tick and only its support index is propagated by the hard-mass classical-limit update law. Reported *barycenter drift* is the energy-weighted barycenter audit diagnostic used to test realized reciprocity and momentum conservation; it has no feedback path into any update, force, kernel, or parameter.

Exchange-form relational gravity operator.

$$(\hat{K}_{\mathrm{grav}}\Phi)_a = \sum_{b \in \mathcal{N}(a)} w_{ab} \left(\Phi_b - \Phi_a\right). \tag{16}$$

The weights are radial in shell index: $w_{ab} = w(d_{ab})$, with Chebyshev hop distance chart

$$d_{ab} = \max\big(|x_a - x_b|, |y_a - y_b|, |z_a - z_b|\big), \tag{17}$$

where (x_a, y_a, z_a) are hop-count coordinates and do not define a background metric. Chebyshev hop distance is a concrete representative used for hop counting and diagnostics; the DO ontology requires only a graph distance that yields well-defined shells and approximate shell isotropy. Physical separations are obtained by the DO unit map: one hop corresponds to ℓ_0 and one tick corresponds to τ_0 (Scope and Linkage).

Shell-gradient coefficient and hard-mass coupling. In the Appendix B hard-mass realizations, interaction magnitude is supplied by the realization-defined shell-gradient coefficient $\gamma(d)$ and the pairwise force law

$$\mathbf{F}_{ij}(t) = \chi_E\, m_i m_j\, \gamma\big(d_{ij}(t)\big)\, \hat{\mathbf{u}}_{ij}(t), \tag{18}$$

with

$$d_{ij}(t) = \max\big(|x_i - x_j|, |y_i - y_j|, |z_i - z_j|\big), \tag{19}$$

where $\hat{\mathbf{u}}_{ij}(t)$ is the realization's background-independent direction rule. Here $\gamma(d)$ is the shell-gradient coefficient induced by the chosen radial kernel $w(d)$ under the Appendix B flux integral; inverse-distance behavior arises as a correspondence property of that fixed kernel family, not as an inserted potential term.

Equivalence consequence (DO). With $m_i \equiv E_i/c^2$,

$$\mathbf{a}_i(t) = \frac{\mathbf{F}_i(t)}{m_i} = \chi_E \sum_{j \neq i} m_j\, \gamma\big(d_{ij}(t)\big)\, \hat{\mathbf{u}}_{ij}(t), \tag{20}$$

so acceleration is independent of the test body's mass parameter.

 doi:10.20944/preprints202405.2028.v8

12.6. Read-only macrostates (no feedback fence)

Macrostate predicate (definition).

$$\mathcal{M} : \big(\{\Phi^{(4D)}(t)\}_{t\le T},\ \mathcal{P}_{\le T}\big) \longrightarrow \mathcal{L}, \tag{21}$$

where $\mathcal{P}_{\le T}$ denotes PI/collapse history up to tick T, and $\mathcal{L}$ is a finite analyst-chosen label set. **No feedback fence:** $\mathcal{M}$ is read-only and is not referenced by any update, operator, kernel, or collapse rule.

12.7. Primary audit scalar in Appendix B (operator-level conservation gate)

DO invariant used as the conservation gate.

$$I_{\mathrm{DO}} := E_{\mathrm{kin}} + U - W_{\mathrm{closure}}. \tag{22}$$

In Appendix B, I_{DO} is the conservation gate; $E_{\mathrm{kin}} + U$ is texture-only (Appendix C artifacts may print E_{kin} as K and $E_{\mathrm{kin}} + U$ as K+U).

 doi:10.20944/preprints202405.2028.v8

13. Disclosure Ledger (Fixed choices, fitted quantities, and prohibited moves)

Realization-choice rationale (physics-facing; law unchanged).

- **Hard-mass orbital operator:** the flux-normalized shell field $\gamma(d)$ on Chebyshev shells realizes a shell-diluted central response; $w(d)$ is recovered by the discrete inward integral. No potential term is inserted into the dynamics.
- **Background-independent direction:** the Manhattan–fraction direction rule $\hat{\mathbf{u}}_{ij}$ is a fixed central-direction convention that enforces realized reciprocity $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ and supports barycenter and angular-momentum audits.
- **Shell interpolation:** $\gamma(d)$ is evaluated at non-integer d by the realization's fixed shell interpolation convention; $w(d)$ is evaluated by the corresponding integral-consistent interpolation induced by that γ convention, with no tunable coefficient and no feedback into the operator.
- **Long-horizon stepping:** Forest–Ruth is used as a fixed symplectic integrator for conservative long-horizon stepping in the hard-mass regime; conservation is audited by I_{DO} together with angular-momentum and barycenter gates.
- **UEE field evolution (Validation I):** one disjoint-link matching per tick with a fixed schedule yields exact Bell-sum conservation by construction and provides SR-facing hop-count causal bookkeeping.

13.1. Fixed vs. fitted

Ledger (scan): fixed realization choices, single-fit quantities, and read-only items.

Quantity / choice	Status	Used in	Notes (read-only where applicable)
Kernel profile $w(d)$ and cutoff R_w	fixed	III–A, III–B, IV	Chosen once per realization; time-independent; diagnostics read-only.
Shell-gradient $\gamma(d)$ convention	fixed	III–A, III–B, IV	Flux-normalized in Appendix B hard-mass orbits.
Direction rule $\hat{\mathbf{u}}_{ij}$	fixed	III–A, III–B, IV	Background-independent direction rule; reciprocity audited by barycenter drift and L gates.
Integrator family (Forest–Ruth)	fixed	III–A, III–B, IV	Symplectic update in the hard-mass support-index law.
Tick-map schedule $r(K)$	fixed	I	Deterministic matching schedule; no mid-run change.
Seed ID and seed scale $\varepsilon_{\mathrm{seed}}$	fixed	I	Fixed a priori; not tuned to χ^2.
Fit window $\ell = 5 \ldots 50$	fixed	I	Fixed a priori.
Amplitude on power κ	fitted (single scalar)	I	Post-processing bridge only; no feedback into evolution.
Classical labels / classifiers $\mathcal{M}$	read-only	all	No feedback into Φ, kernels, schedules, or collapse.

13.2. Constraints satisfied by the validation suite (mathematical exclusions)

- No perturbative expansions, no series truncations, no renormalization tuning.
- No auxiliary stabilizers (friction, smoothing, damping, regulators) inserted into the update.

 doi:10.20944/preprints202405.2028.v8

- No mid-run parameter changes.
- No feedback from labels, gates, or diagnostics into Φ, placement, PI triggering, schedules, kernels, or collapse.

14. Validation Dashboard and Audit Gates

Dashboard (operator, horizon, gates, headline result).

Validation	Operator / regime	Gates disclosed	Headline outcome
I: $t = 0 \to$ TT (low–ℓ)	UEE tick-map evolution + read-only TT bridge	χ^2(5–50), basis guardrail, rotation gate, ε-linearity	$\chi^2/(n-1) = 1.108$; rotation spread 0.000; ε-linearity mean 4.163×10^{-5}
II: CHSH	PD-only collapse law evaluated under CHSH	$\lvert S\rvert$, per-setting correlations	$\lvert S\rvert = 2.8284$ (Tsirelson)
III–A1/A2: Two-body	Hard-mass two-body under fixed $\gamma(d)$ and direction rule	Orbit counting, barycenter drift, I_{DO} and δL auditing	1 orbit closure; 1000 complete orbits; barycenter drift 10^{-13} and 10^{-9} hops
III–B1/B2: Three-body	Fully coupled three-body under same operator	Boundedness, inner-orbit clock, barycenter drift, I_{DO} and $\lvert L\rvert$	1-year invariants $\sim 10^{-13}$; 1000 inner orbits with I_{DO} RMS 3.58×10^{-12}
IV1/IV2: Mercury–Sun	High-energy two-body under same operator	I_{DO} and L_z control; coarse/refined apsidal stats	1000 orbits; RMS $\delta I_{\mathrm{DO}} = 1.427 \times 10^{-11}$; $\langle \Delta\omega \rangle_r \approx 88.685$ arcsec/orbit

 doi:10.20944/preprints202405.2028.v8

Gate meanings (what each failure would have implied).

Validation	Gate	Meaning of failure (operator-level diagnosis)
I	rotation gate	axis dependence in bridge/eigensolve or evolved field structure; non-intrinsic anisotropy.
I	basis guardrail	numerical basis closure failure; TT projection not a faithful diagnostic map.
I	ε-linearity	diagnostic leaving quadratic scaling regime; fit no longer interpretable as amplitude-only.
II	$\lvert S\rvert$ outside Tsirelson	collapse correlation law not matching the DO singlet prediction; PD-collapse rule mismatch.
III–IV	barycenter drift blow-up	loss of reciprocity or accumulation dominating; operator not stable under extreme horizons.
III–IV	I_{DO} drift blow-up	work-closure accounting not closing discrete work–potential mismatch; conservation gate failure.
III–IV	L drift blow-up	angular-momentum non-centrality or numerical failure; direction rule/operator mismatch.

15. Validation I: $t = 0 \to$ CMB (low–ℓ TT)

Objective. Evolve the DO-text $t = 0$ field under the UEE tick map on a periodic torus graph (one disjoint-link matching per tick; fixed schedule; uniform-mode neutral), map the evolved field to low–ℓ TT bandpowers by the fixed read-only bridge, and assess fit quality using a single fitted amplitude on power.

Model comparison (single amplitude on power).

$$C_\ell^{\mathrm{model}} = \kappa^2\, C_\ell^{\mathrm{pred}}. \tag{23}$$

Define $\lambda \equiv \kappa^2$. The diagonal weighted least-squares minimizer over $L = \{\ell : 5 \le \ell \le 50\}$ is

$$\lambda^\star = \frac{\sum_{\ell\in L}\left(C_\ell^{\mathrm{pred}}\, C_\ell^{\mathrm{obs}}/\sigma_\ell^2\right)}{\sum_{\ell\in L}\left((C_\ell^{\mathrm{pred}})^2/\sigma_\ell^2\right)}, \qquad \kappa = \sqrt{\lambda^\star}, \tag{24}$$

and

$$\chi^2 = \sum_{\ell\in L} \frac{\left(C_\ell^{\mathrm{model}} - C_\ell^{\mathrm{obs}}\right)^2}{\sigma_\ell^2}. \tag{25}$$

Fixed vs. fitted (audit checklist). Fixed a priori: seed scale and seed ID; tick law, schedule, and $\theta(1)$; run horizon; shell radius and weights; eigensolve target K; binning; seeded-q bucket rule; calibration probe amplitude and policy; fit window $\ell = 5 \ldots 50$; and all gate thresholds. Fitted: κ only (single global amplitude on power). κ is a post-processing bridge with no feedback path into Φ, any operator, any kernel, or the collapse rule.

Results (single production run).

- Fit: $\kappa = 2.089482181263314 \times 10^{13}$; $\chi^2(5\text{–}50) = 49.876$ over 46 points ($\chi^2/(n-1) = 1.108$).
- Basis guardrail: $\|V^\top M V - I\|_F = 1.024 \times 10^{-12}$ at $K = 2916$, with $M := \mathrm{diag}(w)$.
- Rotation gate: χ^2 identical across all axis permutations (spread 0.000).
- ε-linearity: mean 4.163×10^{-5}, max 1.485×10^{-3}.

 doi:10.20944/preprints202405.2028.v8

Residual localization. The dominant $\Delta\chi^2$ contributors in the fit window are $\ell = 5$ and $\ell = 7$, with additional concentrated contributors near $\ell = 40$ and $\ell = 48$.

16. Validation II: First-Principles Derivation of CHSH Violation

Objective. Numerically evaluate the DO singlet correlation law and compute the CHSH statistic from the four analyzer-axis pairs. Validation II uses PD-only collapse and mirroring and does not employ the UEE tick-map evolution.

Whenever a validation evaluates collapse probabilities, it does so relative to a specified outcome family $\{O'_j\} \subset \mathrm{Conn}(\mathcal{C})$ that is pairwise disjoint and satisfies $\bigsqcup_j O'_j = \mathcal{O}(t_{\mathrm{pre}})$; the probability assigned to each O'_j is the fraction of the state's total hosted (non-CIE) intrinsic energy contained in O'_j immediately before collapse (Sec. 3.2).

CHSH construction.

$$S = E(A, B) - E(A, B') + E(A', B) + E(A', B'). \tag{26}$$

Results. The Monte Carlo evaluation yields $|S| = 2.8284$, matching Tsirelson's bound $2\sqrt{2}$ to the reported precision. The per-setting correlations match the expected $-\cos\theta$ values within sampling error.

17. Validation III–A: Two-Body Orbital Stability (1-year and 1000-orbit)

Objective. Validate the hard-mass classical limit of the UEE for two bodies under one fixed flux-defined relational operator. Report orbit closure/identification, barycenter control, and long-horizon conserved-quantity auditing.

Results summary. The 1-year run identifies one closed orbit ($\Delta\theta = 360.000208°$). The long-horizon run identifies 1000 complete orbits by deterministic unwrapped 2π crossings of the orbital-plane phase. The barycenter remains fixed to 10^{-13} hops (1-year) and 10^{-9} hops (1000-orbit), and angular-momentum invariance remains at the 10^{-14} (1-year) to 10^{-12} (1000-orbit) level.

Table 5. Comparison of 1-year and 1000-orbit two-body runs (Earth–Moon analog, DO two-body operator).

Horizon	dt	Steps	Orbits	$\Delta\theta$ (deg)	Max barycenter drift (hops)	$\delta I_{\mathrm{DO,net}}$	RMS(δI_{DO})	RMS(δL)
1-year (1 orbit)	1.0×10^{-2}	443850	1	360.000208	2.402499×10^{-13}	5.880×10^{-13}	4.128×10^{-12}	3.069×10^{-14}
1000-orbit horizon	3.0×10^{-3}	1507173375	1000	360° (by 2π-crossing window definition)	3.296×10^{-9}	2.245×10^{-12}	2.242×10^{-12}	3.589×10^{-12}

18. Validation III–B: Three-Body Orbital Stability (1-year and 1000 inner orbits)

Objective. Extend the hard-mass orbital validation to a fully coupled three-body configuration under the same fixed DO operator. Report boundedness, orbit clock stability, barycenter control, and invariant auditing over one inner-orbit window and over 1000 complete inner orbits.

Results (1-year inner-orbit window).

Table 7. Global conservation diagnostics for the 1-year three-body validation (one complete inner-orbit window).

Metric	Value
I_{DO} net drift	5.047×10^{-13}
I_{DO} RMS	3.996×10^{-13}
$\lvert L\rvert$ RMS	5.380×10^{-13}
Max barycenter drift (hops)	1.860×10^{-13}

Results (first 1000 complete inner orbits).

 doi:10.20944/preprints202405.2028.v8

Table 8. Global diagnostics for the 1000-orbit three-body run (first 1000 complete inner orbits).

Metric	Value
I_{DO} net drift	3.981×10^{-12}
I_{DO} RMS	3.580×10^{-12}
$\lvert L \rvert$ net drift	5.104×10^{-11}
$\lvert L \rvert$ RMS	1.883×10^{-11}
barycenter drift max (hops)	8.063×10^{-9}
barycenter drift RMS (hops)	3.651×10^{-9}
barycenter drift max / $\langle r_{\mathrm{in}} \rangle$	2.883×10^{-10}
Inclination mean/std (deg)	5.145000/0.000000
Node angle start/end (deg)	$180.000000 \rightarrow 180.000000$

19. Validation IV: Mercury–Sun Orbital Stability (1-year and 1000 orbits)

Objective. Validate the same fixed DO hard-mass operator in a high-mass, moderately eccentric two-body band (Mercury–Sun analog). Report invariant auditing over one orbit and over a 1000-orbit horizon, together with coarse and refined apsidal-advance statistics.

Results (1-year).

Table 12. Global conservation metrics for the 1-year Mercury–Sun DO run.

Metric	Value
I_{DO} net drift	4.526×10^{-13}
RMS δI_{DO}	1.744×10^{-13}
Naive E net drift	7.370×10^{-4}
RMS $\delta E_{\mathrm{naive}}$	3.011×10^{-1}
RMS δL_z	1.140×10^{-13}

Results (1000 orbits).

Table 13. Summary diagnostics for the 1000-orbit Mercury–Sun run at $D_0 = 60$ (coarse and refined apsidal-advance statistics).

Horizon	Steps (executed)	Orbits (found)	RMS δI_{DO}	RMS δL_z	$\langle \Delta\omega \rangle_c$	σ_c	$\langle \Delta\omega \rangle_r$	σ_r
1000-orbit	5.803×10^9	1000	1.427×10^{-11}	3.072×10^{-12}	8.861×10^1	2.268×10^2	8.868×10^1	7.406

Estimator convergence (diagnostic-facing). The refined perihelion estimator collapses stride-aliasing scatter while leaving the mean apsidal advance essentially unchanged, separating diagnostic conditioning from dynamics.

Data Access Statement: All relevant data are included in this paper and its Supporting Information/Appendix files.

Acknowledgments: To be completed following peer review.

Funding: To be completed following peer review.

Ethical Statement: To be completed following peer review.

Conflict of Interest Declaration: The author has no affiliations with or involvement in any organization or entity with any financial interest in the subject matter or materials discussed in this manuscript.

ORCID iD: To be completed following peer review.

 doi:10.20944/preprints202405.2028.v8

Appendix A. Mathematical Constraints of the Dual Ontology Model

Appendix A.1. Introduction

Appendix A presents mathematics constrained by the Dual Ontology (DO). The ontology determines which entities are admissible and how they behave; the mathematics implements those specifications without altering them. Within these bounds, this appendix defines the admissible objects, the unified operator form $\hat{K}_{\text{eff}} = \hat{K}_{\text{kin}} + \hat{K}_{\text{grav}}$ and its admissible realizations governing both motion and gravitational response, and the collapse rule that acts in the Planck Domain $S^{(PD)}$ and is mirrored into discrete 4D spacetime $S^{(4D)}$. Methods and solvers are outside this appendix. Appendix A records read-only post-processing bridges used only for comparing DO outputs to observational datasets (Sec. 2.9.5); these bridges are not parameters of the theory, are not referenced by any update, operator, kernel, or collapse rule, and have no feedback path into any evolution map, kernel data, or collapse selection. Appendix B provides eight rigorous numerical validations using only the Appendix A objects and rules. Appendix B exercises $\hat{K}_{\text{kin}}$ and $\hat{K}_{\text{grav}}$ across the UEE field-evolution and hard-mass regimes. It does not present a dispersive Φ-run in which $\hat{K}_{\text{kin}}$ and $\hat{K}_{\text{grav}}$ are separately isolated as independent measurable contributions; gravitational response is therefore validated in the hard-mass regime where it is diagnostically sharp.

Interpretive rule (mathematics is descriptive). All mathematics in Appendix A is descriptive bookkeeping for the DO ontology. Symbols, sets, and operators represent DO primitives and their allowed behaviors in $S^{(4D)}$ and $S^{(PD)}$; the equations do not introduce additional ontic entities beyond the DO inventory.

Layering. Appendix A is organized as: (i) ontology-level axioms and constraints; (ii) admissible realizations (allowed choices of kernels/weights, discretizations, and update maps that satisfy the constraints); and (iii) diagnostics (read-only checks that audit a realization). Appendix B instantiates admissible realizations and reports diagnostics; diagnostics never modify Φ, $w(d)$, or any update rule.

Continuum-correspondence statements in Sec. 2.9 are invoked only for realizations with $w \in \mathcal{W}_\Delta \subset \mathcal{W}$ (Sec. 1.1.9). Outside Sec. 2.9, Appendix A makes no Laplacian-symbol or SR-dispersion claims and holds for any admissible $w \in \mathcal{W}$.

For orientation, terminology follows the main text. A Discrete Sphere (DS) is an indivisible spatial unit with fixed volume. A Bell Field is the set of DS currently occupied by a quantum state in 4D spacetime. The Bell Energy Field assigns hosted (non-CIE) per-sphere energy samples $e_\alpha(t) = |\Phi_\alpha(t)|^2$ on those DS. The Planck Domain $S^{(PD)}$ is a real, independent, timeless domain comprising all DS, with formal dimension $3 \times N$ represented by the indexed family of DS-identifying spatial triples (x_i, y_i, z_i) (Sec. 1.1.3). These triples are the same DS identifiers used in $S^{(4D)}$; they introduce no internal spatial separation and no internal metric in $S^{(PD)}$. A Bell Energy Point is the quantum state's timeless representation in the Planck Domain. The Bell Identity g mirrors, sphere by sphere and total hosted (non-CIE) intrinsic energy to total hosted (non-CIE) intrinsic energy, between a quantum state in 4D spacetime and the Planck Domain.

Single operator. One effective operator $\hat{K}_{\text{eff}} = \hat{K}_{\text{kin}} + \hat{K}_{\text{grav}}$ governs both kinetic motion and relational gravitational response (Sec. 1.1.7). The interaction structure is fixed by the chosen radial profile $w(d)$ together with the realization's cutoff rule, so exchange occurs only across pairs (a, b) with $w(d_{ab}) > 0$ (Sec. 1.1.7). The kinetic term is a symmetric exchange across that fixed interaction structure with conductances $\eta_{ab} = \eta_{ba}$ (Sec. 1.1.7). The gravitational term is the shell-isotropic exchange built from the same radial weights $w(d_{ab})$ (Secs. 1.1.7 and 2.2), acting on the state on $S^{(4D)}$. Both exchange operators are constructed to be neutral on the uniform mode.

Evolution and collapse. Between collapses, the Unified Evolution Equation deterministically updates the Bell Energy Field on the discrete 4D graph. Collapse occurs only in $S^{(PD)}$ and is mirrored to $S^{(4D)}$ on the same tick. An admissible outcome is any connected subset of the pre-collapse Bell

 doi:10.20944/preprints202405.2028.v8

Sphere support. The probability of an outcome equals the fraction of the quantum state's total hosted (non-CIE) intrinsic energy that was in that subset just before collapse. Over any set of mutually exclusive outcomes that together cover the pre-collapse support, the probabilities add to one.

Equivalence consequence. For any body, quantum or classical, the inertial parameter equals the gravitational source, $m = E/c^2$, with E the body's total hosted (non-CIE) intrinsic energy. Free-fall acceleration is independent of composition and depends only on the sources' energies through the kernel.

Guardrails. Rotation and permutation invariance on the lattice; exact conservation of total hosted (non-CIE) intrinsic energy between collapses; neutrality of the uniform mode; no background metric and no independent gravitational field to quantize.

Appendix A.2. Scope and Linkage

This appendix records the mathematical objects and constraints of the DO model. It is *physics-neutral*: no observational calibrations, data choices, or run-specific numerical settings are assumed here. Those choices and their statistical vetting are provided in Appendix B.

Conventions used in this appendix.

- *Graph distance and neighborhood.* Unless noted, the graph distance d_{ab} denotes the Chebyshev hop-distance on a 3D cubic network when a concrete representative is needed; "(26C)" refers to the full nearest-neighbor set for that choice. The DO ontology itself requires only a graph distance that yields well-defined shells and approximate shell isotropy. All results therefore extend to any graph family whose shell structure admits the regularity/isotropy assumptions of Sec. 1.1.9; Chebyshev shells are a convenient example, not a constraint of the theory.
- *Kernel class.* Link weights $w_{ab} = w(d_{ab})$ are members of the admissible class $\mathcal{W}$ defined in Sec. 1.1.9 (with $\mathcal{W}_\Delta \subset \mathcal{W}$ used when continuum-symbol correspondence is invoked). For any chosen $w \in \mathcal{W}$, the profile $w(d)$ and the derived weights w_{ab} are fixed and time-independent; state-dependence enters through $\Phi(t)$ evolving under the UEE tick map (Sec. 2.3), while $e_b(t) = |\Phi_b(t)|^2$ is used for collapse bookkeeping and read-only auditing. The concrete kernels used in simulations (Appendix B) are representatives of $\mathcal{W}$; no derivation here depends on a particular representative.
- *Units and unit mapping.* Energy-like quantities are expressed in internal units with $E_0 = 1$ unless explicitly restored. The invariant speed c is one hop per time tick by definition. All operational time labels extracted from clock histories in $S^{(4D)}$ are constrained by the same invariant causal bound used for SR bookkeeping.
- *Observational calibration (post-processing bridge).* The global power-amplitude map κ used for data comparison is defined in Sec. 2.9.5 and applied in Appendix B (Validation I). It is a read-only post-processing bridge: it is not referenced by any update, operator, kernel, or collapse rule, and it has no feedback path into any evolution map, kernel data, or collapse selection.
- *Validation probes.* Rotation gates (axis permutations) and ε-scaling checks are estimator validation tests reported in Appendix B. They are not additional axioms of Appendix A.
- *Tick labels (metadata only).* We reserve $K \in \mathbb{Z}$ for the discrete tick index. When cosmology labels are attached to a run, they are a fixed, monotone metadata map $K \mapsto (a, z, \tau)$ with $\tau = 0$ anchored to the DO-text $t = 0$ reference; these labels never enter any equations in Appendix A.
- *Classical labels (macrostate predicates; read-only).* Words such as "alive/dead," "chair," "solid," etc. are not DO primitives. They denote read-only macrostate predicates on $S^{(4D)}$ histories and PI/collapse history (Sec. 5.3) and have no feedback path into Φ, any operator, $w(d)$, or C.
- *Particle terminology (read-only labels).* Standard particle-physics names (e.g., electron, muon, neutrino, photon) are used only as reader-facing labels for repeatable 4D interaction signatures and conserved-quantity patterns of DO states in $S^{(4D)}$. These terms introduce no additional

 doi:10.20944/preprints202405.2028.v8

ontic primitives and have no feedback path into Φ, any operator, any kernel, any schedule, PI triggering, or collapse.

This preface makes Appendix A self-contained while remaining compatible with the discrete-relational demonstrations summarized in Appendix B.

Appendix A.3. Rule Book: DO Law vs. Realization Data (fixed per run; no mid-run feedback from diagnostics/bridges/labels into dynamics)

Reader map. Left column: DO law-level primitives and constraints (not tuned per run). Right column: admissible realization data (fixed per run; no mid-run feedback from diagnostics/bridges/labels into dynamics). Appendix B instantiates admissible realizations and reports read-only audits; Appendix C preserves executable artifacts.

Rule Book — DO Law (primitives and constraints)	**Rule Book — Realization Data (fixed per run; no mid-run feedback from diagnostics/bridges/labels into dynamics)**
• Ontology: DS primitives; $S^{(4D)}$ (tick evolution) and $S^{(PD)}$ (collapse); $S^{(DO)} = S^{(4D)} \times S^{(PD)}$. • Planck Identity: DS identity across domains. • Bell Identity g: bijective mirroring of dual state representations; collapse occurs only in $S^{(PD)}$ and is mirrored into $S^{(4D)}$ on the same tick. • State objects in $S^{(4D)}$: descriptor $\Phi^{(4D)}(t)$, support $\mathcal{O}(t)$ (Bell Field), and hosted (non-CIE) samples $e_\alpha(t) = \lvert\Phi_\alpha^{(4D)}(t)\rvert^2$ (Bell Energy Field). • Between-collapse evolution: deterministic UEE tick map on a fixed link set; Bell-sum conservation between collapses; uniform-mode neutrality. • Kernel admissibility and correspondence fence: $w \in \mathcal{W}$ (A1–A6, Sec. 1.1.9); Sec. 2.9 correspondence uses $w \in \mathcal{W}_\Delta$ (A7). • Collapse rule: admissible outcomes are connected subsets of pre-collapse support; probabilities are hosted-energy fractions; post-collapse energy preserved by one rescaling on the selected outcome; mirrored support restriction in $S^{(4D)}$. • Relational gravity: exchange-form response carried by the fixed relational link structure; no continuum metric field; no potential-times-Φ term inserted; DS CIE excluded from local sourcing. • Read-only fence: diagnostics, gates, bridges, and classical labels are not referenced by any update, operator, kernel, schedule, or collapse rule.	• Graph representative and hop-distance convention (index-chart choices for hop counting; no background metric). Index charts are realization/implementation data used to evaluate d_{ab} (and, in hard-mass realizations, d_{ij} and $\hat{\mathbf{u}}_{ij}$); they carry no ontic metric geometry. • Kernel representative $w(d)$ chosen from the admissible class and its cutoff R_w; shell-measure conventions used for flux bookkeeping. • Fixed link decomposition into disjoint matchings and the fixed periodic schedule $r(K)$ used to realize the tick map. • Tick parameters used in a run (e.g. Δt, $\theta(1)$, optional on-site phase factors); numerical precision choices as implementation details. • Hard-mass regime choices: shell-gradient convention $\gamma(d)$ (centered-difference or flux-normalized) and the background-independent direction rule $\hat{\mathbf{u}}_{ij}$; integrator family for the hard-mass classical-limit update law (e.g. Forest–Ruth) • Boundary realization (periodic torus vs. finite box) and horizon choices (steps, orbit windowing) for a run. • Expansion bookkeeping (cosmology): the monotone DS-count sequence $N_{4D}(t)$ (equivalently ΔN_{4D} per tick) used for Secs. 7.3–7.4 is realization data fixed for a run; it has no feedback path into Φ, $w(d)$, any operator, any schedule, or C. • Read-only post-processing: spectrum bridge objects (shell basis, calibration tensors), κ amplitude mapping, orbit segmentation rules, refined perihelion estimator, gate thresholds, and plotting conventions/rendering charts. This item excludes hop-count index charts used to evaluate d in realized dynamics.

Appendix B. Formal Structure of the Dual Ontology Model

Appendix A records architecture-level definitions and constraints for the DO. It is method-agnostic: concrete numerical realizations (kernels, stencils, time-steppers, solver tolerances) are outside its remit and may vary provided they respect the axioms and conservation statements given here. Appendix B provides admissible realizations used in the validations.

 doi:10.20944/preprints202405.2028.v8

The DO framework is constructed entirely from discrete, physically real structures and is structurally incompatible with Hilbert space, Fock space, $3N$ configuration space, matrix mechanics, and the Schrödinger equation. These constructs do not appear in Appendix A.

The DO model is built without ad hoc assumptions, fine-tuning, or perturbative methods.

The model defines quantum evolution and collapse over a discrete, background-independent 4D spacetime $S^{(4D)}$ and a $(3 \times N)$-dimensional Planck Domain $S^{(PD)}$, where N is the total number of Discrete Spheres. $S^{(4D)}$ consists of a subset of $\mathrm{DS}_{\mathrm{total}}$ at each tick, while $S^{(PD)}$ comprises all of $\mathrm{DS}_{\mathrm{total}}$.

Appendix B.0.1. Quantum Evolution in 4D Spacetime

Quantum states evolve deterministically in $S^{(4D)}$ via the Bell-sum–conserving UEE tick map, defined in Sec. 2.3.

Appendix B.0.2. Quantum Collapse in the Planck Domain

Collapse is a non-reversible outcome-selection event in $S^{(PD)}$ governed by the DO Probability Rule (Sec. 1.1.8), and is mirrored into $S^{(4D)}$ on the same tick by the Bell Identity g (Sec. 1.1.5 and Sec. 2.1). Collapse is a discrete probability event; the DO paper leaves open whether the collapse process is fully deterministic.

Appendix B.0.3. Relational Gravity

Gravity arises as relational response to differences in hosted energy on $S^{(4D)}$ and is carried by the DO gravitational exchange operator; no independent gravitational field degree of freedom is introduced.

Appendix B.0.4. Discrete Cosmological Evolution

Cosmological evolution is described as a universal-scale collapse from a dispersed $S^{(4D)}$ state at Heat Death to a generally localized $S^{(4D)}$ state at $t = 0$ (Sec. 7).

Appendix B.1. Mathematical Definition of Terms and Structures

Appendix B.1.1. DO Sub-Structure

Discrete Spheres (Ontological Definition). Fundamental ontological constituents form the set $\mathrm{DS}_{\mathrm{total}}$ containing all Discrete Spheres (DS) in the ontology. Let $\mathcal{I}$ be an abstract index set with cardinality $N := |\mathcal{I}|$ (unknown, possibly infinite):

$$\mathrm{DS}_{\mathrm{total}} = \{\mathrm{DS}_i \mid i \in \mathcal{I}\}, \qquad |\mathcal{I}| = N. \tag{A1}$$

Each element $\mathrm{DS}_a \in \mathrm{DS}_{\mathrm{total}}$ is an irreducible unit of spatial ontology. The Planck-volume value (and V_0 in the unit map) is illustrative, and "Discrete Sphere" is a DO object label rather than an embedded Euclidean sphere; spatial adjacency and propagation are defined purely by the DS neighbor graph in $S^{(4D)}$, so no interstitial spatial region exists that a state must traverse.

State of Absolute Nothingness (SOAN). SOAN is an explicitly passive, dimensionless ontological entity ($\dim(\mathrm{SOAN}) = 0$) without physical properties (space, volume, time). It provides the invariant reference used to state that $S^{(PD)}$ is timeless and metric-free; SOAN does not interact with DS.

Invariant Speed of Light (c). In internal units, c is defined as one adjacent-sphere hop per discrete time tick. Under a physical-unit mapping, assign hop length Δx_{hop} and tick duration Δt such that

$$c \;=\; \frac{\Delta x_{\mathrm{hop}}}{\Delta t} \;=\; 1 \text{ hop/tick}.$$

This is a unit convention for reporting; it does not modify any DO update rule.

 doi:10.20944/preprints202405.2028.v8

Constants and unit mapping.

$e_\alpha := \lvert\Phi_\alpha\rvert^2$	internal hosted energy per sphere; physical energy $E_\alpha = E_0\, e_\alpha$
c	$= 1$ hop/tick internally; map by $c = \Delta x_{\text{hop}}/\Delta t$ for physical units
$\hbar$, m	appear only as operator prefactors; symbolic until a unit map is chosen
κ	global amplitude bridge for spectra (Appendix B, Validation I); not part of dynamics

Default simulations commonly take $E_0 = \Delta x_{\text{hop}} = \Delta t = 1$; physical reporting applies the above maps.

Appendix B.1.2. 1.1.2. The State Descriptor (Φ) $S^{(4D)}$

This section defines the state descriptor Φ and the associated Bell Energy Field in $S^{(4D)}$. Let $t \in \mathbb{Z}$ be a discrete tick. A single quantum state occupies a set of Bell Spheres

$$O(t) \subset S^{(4D)}.$$

Notation (occupied support in $S^{(4D)}$ and its PD image). The occupied support of a state in $S^{(4D)}$ at tick t is denoted $\mathcal{O}(t) \subset S^{(4D)}$. The symbol $O(t)$ (and forms such as $O_n(t)$, $O_{N\text{-body}}(t)$) are used synonymously in some sections of Appendix A and always refer to this same $S^{(4D)}$ occupied support:

$$O(t) \equiv \mathcal{O}(t) \equiv \operatorname{Supp} \Phi^{(4D)}(t).$$

Define $\mathcal{O} := \mathcal{O}(t)$ as the $S^{(4D)}$ occupied support under discussion. The Planck-Domain support set is the Planck-identity image of the $S^{(4D)}$ support being discussed and carries no time argument:

$$\mathcal{O}^{(PD)} := \{\, P_i \mid \mathrm{DS}_i \in \mathcal{O} \,\}.$$

When a pre/post distinction is needed, define the corresponding $S^{(4D)}$ supports and take their PD images; the pre/post tags reference the $S^{(4D)}$ ticks and do not introduce PD time:

$$\mathcal{O}_{\text{pre}} := \mathcal{O}(t_{\text{pre}}), \qquad \mathcal{O}_{\text{post}} := \mathcal{O}(t_{\text{post}}),$$

$$\mathcal{O}^{(PD)}_{\text{pre}} := \{\, P_i \mid \mathrm{DS}_i \in \mathcal{O}_{\text{pre}} \,\}, \qquad \mathcal{O}^{(PD)}_{\text{post}} := \{\, P_i \mid \mathrm{DS}_i \in \mathcal{O}_{\text{post}} \,\}.$$

A Bell Sphere is a single Discrete Sphere together with the portion of the state's energy hosted on that sphere. All Bell Spheres occupied by the state at tick t constitute the state's Bell Field in $S^{(4D)}$.

Appendix A represents the state on each occupied Bell Sphere $a \in O(t)$ by a local state descriptor sample $\Phi_a(t)$. The Bell Energy Field is defined as the energy component of the Bell Field by the per-sphere energy sample

$$e_a(t) \;:=\; |\Phi_a(t)|^2 \;\in\; \mathbb{R}_{\geq 0}, \qquad a \in O(t).$$

Here $|\cdot|^2$ denotes squared modulus; for $\Phi_a = x_a + i\, y_a$, $|\Phi_a|^2 = x_a^2 + y_a^2$. The Bell Energy Field at tick t is the collection of these samples over the occupied set:

$$\{\, e_a(t) \,\}_{a \in O(t)}.$$

The total hosted energy of the state at tick t is the Bell sum

$$E_{QS}(t) \;:=\; \sum_{a \in O(t)} e_a(t).$$

Under the Planck Identity and the Bell Identity g, the Bell Spheres comprising the Bell Field in $S^{(4D)}$ are mirrored to the constituents of the state's Bell Point in $S^{(PD)}$; the quantum energy component in $S^{(PD)}$ is the Bell Energy Point.

 doi:10.20944/preprints202405.2028.v8

Appendix B.1.3. 1.1.3. Planck Domain ($S^{(PD)}$)

The Planck Domain $S^{(PD)}$ is an ontologically real domain comprising all N Discrete Spheres (denoted P_i). Each P_i is intrinsically 3-dimensional. These N DS integrate within $S^{(PD)}$ as a single unified structure (the Universal Planck Point). $S^{(PD)}$ is the maximal Universal Planck Point, encompassing all N DS.

A defining characteristic of any Planck Point (including $S^{(PD)}$) is that, although composed of intrinsically 3D entities P_i, it has no internal spatial separation between constituents and no conventional spatial volume. The formal dimensionality of $S^{(PD)}$ is $3 \times N$, arising from its N intrinsically 3D DS.

Its mathematical representation uses the same spatial coordinate triples (x_i, y_i, z_i) that identify Discrete Spheres in $S^{(4D)}$. The Planck Domain representation is the single $(3 \times N)$ Universal Planck Point given by the family of these triples indexed by $i \in \mathcal{I}$:

$$Q^N := \big((x_i, y_i, z_i)\big)_{i \in \mathcal{I}}. \tag{A2}$$

Equivalently, $S^{(PD)}$ is the set of Planck-identified Discrete Spheres

$$S^{(PD)} \equiv \{P_i \mid i \in \mathcal{I}\}. \tag{A3}$$

The ordered-triple representation Q^N records the same Discrete-Sphere identities as in $S^{(4D)}$ and introduces no internal spatial separation, no internal metric, and no internal distance in $S^{(PD)}$.

The integrated Dual Ontology structure is

$$S^{(DO)} = S^{(4D)} \times S^{(PD)}. \tag{A4}$$

Fundamental constraints for $S^{(PD)}$ are timelessness and metric-freedom:

$$S^{(PD)} \text{ is timeless and metric-free: no internal time coordinate and no internal metric are defined.} \tag{A5}$$

Accordingly, continuum spacetime tensors that presuppose a metric (e.g. $g_{\mu\nu}$, $T_{\mu\nu}$, $G_{\mu\nu}$) are not defined within $S^{(PD)}$. SOAN (Sec. 1.1.1) supplies the invariant reference used to state these constraints.

Appendix B.1.4. 1.1.4. Planck Identity and Structural Distinction

The Planck Identity asserts ontological identity of each Discrete Sphere as represented in $S^{(4D)}$ (by DS_i) and in $S^{(PD)}$ (by P_i):

$$\mathrm{DS}_i \equiv P_i \quad (\forall i \in \mathcal{I}). \tag{A6}$$

This is the identity of the constituent DS across domains; it does not assert equality of domain structure. Accordingly, the same DS-identifying spatial coordinate triple (x_i, y_i, z_i) used for DS_i in $S^{(4D)}$ is the triple in the i-th slot of the Planck-Domain indexed representation Q^N (Eq. (A2)).

Let $\mathcal{C}(S^{(4D)}(t))$ denote the effective relational link set on $S^{(4D)}$ at tick t (defined by the kernel choice and its cutoff rule; Sec. 1.1.7). No analogous relational link structure is defined inside $S^{(PD)}$ (Sec. 1.1.3). Therefore the domain structures differ; set $\mathcal{C}(S^{(PD)}) := \varnothing$.

$$\mathcal{C}(S^{(4D)}(t)) \neq \mathcal{C}(S^{(PD)}) = \varnothing. \tag{A7}$$

Appendix B.1.5. 1.1.5. Bell Identity

The Bell Identity g is a non-dynamical bijective mirroring between a quantum state's dual representations in $S^{(4D)}$ and $S^{(PD)}$. For a single-state representation at discrete time t:

$$\Phi^{(4D)}(t) \xleftrightarrow{g} \phi^{(PD)}. \tag{A8}$$

 doi:10.20944/preprints202405.2028.v8

For general N-body representations ($N \geq 2$):

$$\{\Phi_n^{(4D)}(t)\}_{n=1}^{N} \xleftrightarrow{g} \Phi_{N\text{-body}}^{(PD)}. \tag{A9}$$

Notation (BEP). Throughout Appendix A, $\phi^{(PD)}$ denotes a single-state BEP, and $\Phi_{N\text{-body}}^{(PD)}$ denotes a unified N-body BEP. The BEP encodes conserved intrinsic labels (e.g. total hosted (non-CIE) intrinsic energy E_{QS}, spin, charge) timelessly.

Energy Conservation (DO Ontological Postulate). The total hosted (non-CIE) intrinsic energy $E_{QS}(t)$ of an isolated system in $S^{(4D)}$ is constant between collapses. DS constant intrinsic energy density (CIE) is tracked separately as ρ_Λ (Secs. 6.1–6.2 and 7.3–7.4) and is excluded from E_{QS}. For a hybrid system containing quantum components and classical hard-mass components, this is recorded as: where $\mathrm{Supp}(E_c) \subset S^{(4D)}$ denotes the set of Discrete Spheres α on which the classical component c has nonzero hosted (non-CIE) samples $E_{c,\alpha,\mathrm{classical}}(t)$.

$$E_{QS}(t) = \sum_{q\in\mathrm{Quantum}} \left(\sum_{\alpha\in\mathcal{O}_q(t)} \left|\Phi_{q,\alpha}^{(4D)}(t)\right|^2 \right) + \sum_{c\in\mathrm{Classical}} \left(\sum_{\alpha\in\mathrm{Supp}(E_c)} E_{c,\alpha,\mathrm{classical}}(t) \right) = \text{constant}. \tag{A10}$$

Energy taxonomy (DO). Three energy statements are used: (i) intrinsic energy conservation between collapses (Eq. (A10)); (ii) energy preservation at collapse (Sec. 1.1.8, Proposition); and (iii) run-level conservation auditing for concrete hard-mass realizations (Appendix B), where the audit scalar $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$ is the conservation gate and $E_{\mathrm{kin}} + U$ is reported only as a texture diagnostic (Appendix B may print E_{kin} using the legacy label K.)

Binding–Energy Convention. For any composite, $E_{\mathrm{tot}} = E_{\mathrm{rest}} + E_{\mathrm{kin}} + E_{\mathrm{int}} + E_{\mathrm{field}} - E_{\mathrm{bind}}$. The inertial parameter equals the gravitational parameter, $m \equiv E_{\mathrm{tot}}/c^2$. Formation of a bound state reduces both inertial response and gravitational sourcing by E_{bind}/c^2.

Inertial–Gravitational Energy Identity (DO). For any body n (quantum or classical) with total hosted (non-CIE) intrinsic energy E_n mirrored by its BEP via g, define

$$m_{I,n} := \frac{E_n}{c^2}.$$

Relational gravitational response in $S^{(4D)}$ couples only to hosted source energies (Sec. 2.2), so gravitational sourcing strength is proportional to the same scalar E_n. Thus,

$$m_{I,n} = m_{G,n} = \frac{E_n}{c^2} \quad \text{for all } n.$$

Appendix B.1.6. 1.1.6. N-Body State Representation ($N \geq 2$)

An N-body system consists of $N \geq 2$ component quantum states. A macroscopic "classical" object is an N-body quantum state in the hard-mass limit (its kinetic spreading is suppressed on the barycenter).

Dual representations.

1. **Bell Energy Fields in $S^{(4D)}$.** Each body n has $\Phi_n^{(4D)}(t)$ with occupied support $\mathcal{O}_n(t) \subset S^{(4D)}$. The hosted energy sample on sphere α is $e_{n,\alpha}(t) = |\Phi_{n,\alpha}^{(4D)}(t)|^2$. The combined occupied set is

$$\mathcal{O}_{N\text{-body}}(t) = \bigcup_{n=1}^{N} \mathcal{O}_n(t).$$

 doi:10.20944/preprints202405.2028.v8

2. **Bell Energy Point in $S^{(PD)}$.** The unified system is represented as a single BEP $\Phi^{(PD)}_{N\text{-body}}$ encoding the conserved total hosted (non-CIE) intrinsic energy and other invariants.

Bell Identity (bijective mirroring).

$$g\Big(\{\Phi^{(4D)}_n(t)\}_{n=1}^{N}\Big) \;\longleftrightarrow\; \Phi^{(PD)}_{N\text{-body}}.$$

Occupied sets mirror componentwise. Define $\mathcal{O}_{N\text{-body}} := \mathcal{O}_{N\text{-body}}(t)$ as the $S^{(4D)}$ occupied set under discussion. Then:

$$\mathcal{O}_{N\text{-body}}(t) \;\longleftrightarrow\; \mathcal{O}^{(PD)}_{N\text{-body}}, \qquad \mathcal{O}^{(PD)}_{N\text{-body}} = \big\{P_i \;\big|\; \mathrm{DS}_i \in \mathcal{O}_{N\text{-body}}\big\}.$$

$$E^{(N\text{-body})}_{QS,\text{total}} = \sum_{n=1}^{N} \sum_{\alpha\in\mathcal{O}_n(t)} \big|\Phi^{(4D)}_{n,\alpha}(t)\big|^2 = \text{constant}. \tag{A11}$$

Appendix B.1.7. 1.1.7. Discrete Operators

Time grid. Let $K \in \mathbb{Z}$ index ticks and $t_K := t_0 + K\Delta t$.

$$\Delta_t^{+}\Phi_a(t_K) = \frac{\Phi_a(t_{K+1}) - \Phi_a(t_K)}{\Delta t}, \qquad \Delta_t^{2}\Phi_a(t_K) = \frac{\Phi_a(t_{K+1}) - 2\Phi_a(t_K) + \Phi_a(t_{K-1})}{(\Delta t)^2}.$$

These difference operators are used only for read-only diagnostics, for the observable update blocks (Secs. 2.5.1–2.5.2), and for the correspondence module in Sec. 2.9. The fundamental between-collapse evolution law is the Bell-sum–conserving tick map in Sec. 2.3.

Connectivity and neighborhood. For the kernel indexing, let d_{ab} denote the hop-count separation between Bell Spheres a and b (Chebyshev hops in the cubic representative). Define

$$w_{ab} := w(d_{ab}) \geq 0, \qquad \mathcal{C} = \{(a,b) : w_{ab} > 0\}, \qquad \mathcal{N}(a) := \{\, b \neq a : w_{ab} > 0 \,\}.$$

A set $O \subseteq S^{(4D)}$ is *connected* (notation $O \in \mathrm{Conn}(\mathcal{C})$) when any two Bell Spheres in O are joined by a path in $\mathcal{C}$ lying in O.

Finite range (admissible realization; used in Appendix B). Appendix B realizations are finite-range: there exists a cutoff $R_w < \infty$ such that $w(d) = 0$ for $d > R_w$. Then $\mathcal{N}(a)$ is finite and all exchange operators below are finite-range.

Local energies (hybrid; bookkeeping). The per-sphere hosted (non-CIE) energy sample is

$$e_b(t) = \begin{cases} |\Phi_b(t)|^2, & \text{quantum components,} \\ E_{c,b,\text{classical}}(t), & \text{classical bodies.} \end{cases}$$

These samples are used for collapse probabilities and read-only auditing; operator weights are fixed by η_{ab} and w_{ab} (and in Sec. 2.3 enter the tick map through $q_{ab} = \eta_{ab} + w_{ab}$).

Kinetic operator (symmetric exchange).

$$(\hat{K}_{\text{kin}}\Phi)_a = \sum_{b\in\mathcal{N}(a)} \eta_{ab}\,(\Phi_b - \Phi_a), \qquad \eta_{ab} = \eta_{ba} \geq 0.$$

Relational gravity operator (shell-isotropic exchange). One admissible representative is

$$(\hat{K}_{\text{grav}}\Phi)_a = \sum_{b\in\mathcal{N}(a)} w_{ab}\,(\Phi_b - \Phi_a).$$

 doi:10.20944/preprints202405.2028.v8

Shell-gradient coefficient $\gamma(d)$ (realization-defined). Let $\gamma(d)$ denote the realization's discrete radial-gradient coefficient on shell index d used in the hard-mass support-index coupling law (Sec. 2.4). Two admissible conventions are:

- *Centered-difference convention:*

$$\gamma(k) = \tfrac{1}{2}\big[w(k-1) - w(k+1)\big].$$

- *Flux-normalized convention (Appendix B hard-mass orbits):* $\gamma(d)$ is specified directly and $w(d)$ is recovered by the discrete inward integral

$$w(d) - w(d+1) = -2\,\gamma(d+1), \qquad w(R_w) = 0.$$

Neutral uniform mode. For any symmetric exchange operator of the form $(K\Phi)_a = \sum_{b\in\mathcal{N}(a)} q_{ab}(\Phi_b - \Phi_a)$ with $q_{ab} = q_{ba}$,

$$\sum_a (K\Phi)_a = 0.$$

Hence $\hat{K}_{\mathrm{kin}}$ and $\hat{K}_{\mathrm{grav}}$ are neutral on the uniform mode.

Barriers. An impenetrable divider is enforced by $w_{ab} = 0$ for all cross-divider pairs (a, b); this splits $\mathcal{C}$ into disjoint connected components on which the operators act independently.

(Prototype kernels appear in Sec. 1.1.7.1.)

Appendix B.1.8. 1.1.7.1. Link-Weight Prototype

The link weights $w_{ab} = w(d_{ab})$ define the relational weight profile on $S^{(4D)}$. In Appendix B realizations, $w(d)$ is fixed and time-independent once chosen and is compactly supported by a finite cutoff R_w. State-dependence enters through $\Phi(t)$ evolving on the fixed link set; per-sphere samples $\{e_b(t)\}$ are derived from Φ for collapse bookkeeping and read-only auditing. Diagnostics never alter $w(d)$ or any update rule.

Appendix B.1.9. 1.1.8. Collapse Operator (C)

$S^{(PD)}$ action and mirroring. Let $\phi^{(PD)}$ be the BEP of total hosted (non-CIE) intrinsic energy E_{QS} (Sec. 1.1.5).

$$C:\ \phi^{(PD)} \longrightarrow \phi^{(PD)}_{\mathrm{post}}, \qquad \sum_i \big|\phi^{(PD)}_{\mathrm{post},i}\big|^2 = E_{QS},$$

The $S^{(PD)}$ transformation is evaluated relative to SOAN (Sec. 1.1.1), which remains unchanged by collapse and carries no energy or quantum numbers.

Admissible outcomes (single state). Let $\mathcal{O}(t_{\mathrm{pre}}) = \mathrm{Supp}\,\Phi^{(4D)}(t_{\mathrm{pre}})$ and define hosted (non-CIE) samples $e_\alpha(t_{\mathrm{pre}}) = \big|\Phi^{(4D)}_\alpha(t_{\mathrm{pre}})\big|^2$ on $\alpha \in \mathcal{O}(t_{\mathrm{pre}})$. Define $E_{QS} = \sum_{\alpha\in\mathcal{O}(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}})$. An admissible outcome is any $O' \subseteq \mathcal{O}(t_{\mathrm{pre}})$ with $O' \in \mathrm{Conn}(\mathcal{C})$; post-collapse,

$$e_\alpha(t_{\mathrm{coll}}) = \begin{cases} > 0, & \alpha \in O', \\ 0, & \alpha \notin O'. \end{cases}$$

Terminology (generally localized in $S^{(4D)}$). "Generally localized in $S^{(4D)}$" means that, after collapse, $\mathrm{Supp}\,\Phi^{(4D)}$ is restricted to one admissible connected subset $O' \subseteq \mathcal{O}(t_{\mathrm{pre}})$ and $e_\alpha(t_{\mathrm{post}}) = 0$ for all $\alpha \notin O'$.

Scope (definition vs. implementation). Appendix A defines the admissible outcome family $O' \subseteq \mathcal{O}(t_{\mathrm{pre}})$ with $O' \in \mathrm{Conn}(\mathcal{C})$ and the probability functional $P(O')$ on admissible outcomes. Enumeration/partitioning/sampling over $\mathrm{Conn}(\mathcal{C})$ is realization/implementation data recorded in Appendix B

 doi:10.20944/preprints202405.2028.v8

and preserved as executable constructions in Appendix C; it has no mid-run feedback path from diagnostics/bridges/labels into dynamics.

Probability (energy fraction).

$$P(O') = \frac{\sum_{\alpha \in O'} e_\alpha(t_{\text{pre}})}{\sum_{\alpha \in O(t_{\text{pre}})} e_\alpha(t_{\text{pre}})} \quad \text{for } O' \in \text{Conn}(\mathcal{C}). \tag{A12}$$

Whenever probability evaluation under collapse is invoked in a validation, the candidate outcomes are taken as a specified pairwise-disjoint family of admissible connected subsets whose disjoint union equals the pre-collapse support; each outcome probability equals its pre-collapse hosted-energy fraction (Appendix A, Sec. 3.2).

Proposition (Energy preservation at collapse).

Let $E_{QS} = \sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(t_{\text{pre}})$ be conserved total hosted (non-CIE) intrinsic energy. If C selects an admissible connected outcome $O' \in \text{Conn}(\mathcal{C})$ with pre-collapse energy

$$E_{O'}(t_{\text{pre}}) = \sum_{\alpha \in O'} e_\alpha(t_{\text{pre}}) > 0,$$

define post-collapse samples by a single scalar rescaling on O':

$$e_\alpha(t_{\text{post}}) = \begin{cases} \lambda(O')\, e_\alpha(t_{\text{pre}}), & \alpha \in O', \\ 0, & \alpha \notin O', \end{cases} \qquad \lambda(O') := \frac{E_{QS}}{E_{O'}(t_{\text{pre}})}.$$

Then total energy is preserved at collapse:

$$\sum_{\alpha \in \mathcal{O}(t_{\text{post}})} e_\alpha(t_{\text{post}}) = \sum_{\alpha \in O'} \lambda(O')\, e_\alpha(t_{\text{pre}}) = E_{QS}.$$

Proof. By construction, $\sum_{\alpha \in O'} \lambda(O') e_\alpha(t_{\text{pre}}) = \big(E_{QS}/E_{O'}(t_{\text{pre}})\big) \sum_{\alpha \in O'} e_\alpha(t_{\text{pre}}) = E_{QS}$, and all energy outside O' is set to zero. □

Remark. Outcomes with $E_{O'}(t_{\text{pre}}) = 0$ have $P(O') = 0$ under Eq. (A12) and are not realized.

Einstein boxes (corollary). If a barrier yields a partition of the pre-collapse support into two disjoint connected components,

$$\mathcal{O}(t_{\text{pre}}) = O_P \cup O_C, \qquad O_P, O_C \in \text{Conn}(\mathcal{C}), \quad O_P \cap O_C = \varnothing,$$

Macro-outcome family. For this corollary, take the macro-outcome family to be the disjoint pair $\{O_P, O_C\}$ induced by the barrier partition.

Then the collapse probabilities are the corresponding pre-collapse energy fractions:

$$P(O_P) = \frac{\sum_{\alpha \in O_P} e_\alpha(t_{\text{pre}})}{E_{QS}}, \qquad P(O_C) = \frac{\sum_{\alpha \in O_C} e_\alpha(t_{\text{pre}})}{E_{QS}}.$$

These two macro-events "collapse-localization in O_P" and "collapse-localization in O_C" have the above probabilities and sum to one.

N-body extension (factorized locality). For a unified N-body BEP with bodies $n = 1, \ldots, N$, let $O_n(t_{\text{pre}})$ be body n's support. If PIs during one tick involve the index set $\mathcal{J} \subset \{1, \ldots, N\}$, then

$$C : \Phi^{(PD)}_{N\text{-body}} \longrightarrow \Big(\{\Phi^{(PD)}_n\}_{n \in \mathcal{J}},\ \Phi^{(PD)}_{\text{rest}} \Big).$$

 doi:10.20944/preprints202405.2028.v8

Notation. This factorization is ontological bookkeeping in $S^{(PD)}$; it is not a Hilbert-space tensor product and introduces no additional state-space structure.

Appendix B.1.10. 1.1.9. Minimal Axioms and Admissible Class for the Radial Link Kernel $w(d)$

Let $d \in \mathbb{Z}_{\geq 0}$ denote graph geodesic (hop) separation on the DS network (shell index). The radial link kernel is specified on shell indices as $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Axiom **A4** requires a C^1 extension $w : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$; any use of $w'(d)$ and any statement written in the form $d \to 0$ refer to this C^1 extension.

Axioms.

A1 **Symmetry:** $w_{ab} = w_{ba}$ (equivalently, weights depend only on separation and not on ordering).

A2 **Positivity:** $w(d) \geq 0$ for all d.

A3 **Monotone decay:** $w(d+1) \leq w(d)$ for all $d \geq 0$.

A4 **Regularity:** there exists a C^1 extension $w : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$.

A5 **Scale fixing (amplitude reference):** Choose a reference value $w(1) = w_0 > 0$. The theory is invariant under

$$w(d) \mapsto \alpha\, w(d), \qquad \chi_E \mapsto \chi_E/\alpha,$$

so only ratios $w(d)/w(1)$ (or the product $\chi_E w$) are physical.

A6 **Vanishing at range:** $\lim_{d\to\infty} w(d) = 0$. Compact support (finite cutoff R_w with $w(d) = 0$ for $d > R_w$) is an admissible realization choice and is used in Appendix B realizations.

A7 **Low-$\|\mathbf{k}\|$ Laplacian symbol (diagnostic; continuum correspondence).** For graph families with approximately isotropic shells and for kernel realizations that support a Laplacian continuum symbol, the exchange operator

$$(K_{\mathrm{rad}}\Phi)_a := \left(\hat{K}_{\mathrm{grav}}\Phi\right)_a = \sum_{b\in\mathcal{N}(a)} w_{ab}\,(\Phi_b - \Phi_a)$$

has small-$\|\mathbf{k}\|$ (long-wavelength) symbol $\widehat{K}_{\mathrm{rad}}(\mathbf{k}) = -C_{\mathrm{cont}}\|\mathbf{k}\|^2 + \mathrm{O}(\|\mathbf{k}\|^4)$ with $C_{\mathrm{cont}} > 0$.

Near-field regularity. On graphs admitting an isotropic continuum embedding of spatial dimension n,

$$w(d) = w_0 - c_2\, d^2 + \mathrm{O}(d^4) \quad (d \to 0), \tag{A13}$$

with $c_2 > 0$.

Admissible class.

$$\mathcal{W} = \left\{ w \in C^1(\mathbb{R}_{\geq 0}) \;\middle|\; w \text{ satisfies } \mathbf{A1}\text{–}\mathbf{A6} \right\}. \tag{A14}$$

Continuum-symbol subclass (optional).

$$\mathcal{W}_\Delta = \left\{ w \in \mathcal{W} \;\middle|\; w \text{ satisfies } \mathbf{A7} \right\}.$$

Whenever Appendix A invokes long-wavelength Laplacian-symbol correspondence (e.g. Secs. 2.9.1–2.9.2), it uses $w \in \mathcal{W}_\Delta$.

Consequences. (i) $w'(d)$ exists on $\mathbb{R}_{>0}$ by **A4**. Use of $w'(d)$ and C^1 extension language is confined to Sec. 2.9; hard-mass realizations evaluate $\gamma(d)$ and $w(d)$ at non-integer d by the realization's fixed shell interpolation convention (Appendix B).

(ii) A centered-shell difference $\frac{1}{2}[w(k-1) - w(k+1)]$ is one admissible shell-gradient convention; hard-mass orbit reconstructions may instead specify $\gamma(d)$ directly and recover $w(d)$ by inward summation (Sec. 1.1.7). (iii) Kernels outside $\mathcal{W}$ violate at least one DO requirement (symmetry/positivity/monotonicity/regularity/scale fixing/vanishing at range).

 doi:10.20944/preprints202405.2028.v8

Continuum variational symmetries and gauge potentials (excluded). Appendix A does not use continuous variational symmetries, gauge potentials, or Noether identities. Invariance and conservation statements used here are discrete consequences of symmetric exchange structure, neutrality, periodic/homogeneous convolution realizations, and collapse handled as Bell-Identity mirroring.

Appendix B.1.11. 1.1.10. Guardrails (Discrete Geometry and Basis Checks)

G1 — Shell measure. For a selected observation shell with weights $\{w_\alpha^{(\mathrm{shell})}\}$, the total weight equals 4π to numerical precision: $\sum_{\alpha\in\mathrm{shell}} w_\alpha^{(\mathrm{shell})} \approx 4\pi$.

G2 — Basis closure (numerical guardrail; read-only). For any computed diagnostic mode block V on a selected shell with weights $\{w_\alpha^{(\mathrm{shell})}\}$, define the diagonal weight matrix $M := \mathrm{diag}(w^{(\mathrm{shell})})$. The check $V^\top M V \approx I$ is a numerical closure test for that diagnostic basis only; it is not a DO primitive and has no feedback path into Φ, $w(d)$, any operator, any update map, or C.

G3 — Parseval closure (numerical guardrail; read-only). Map-space and mode-space energies agree to numerical precision under the same weight matrix $M := \mathrm{diag}(w^{(\mathrm{shell})})$ used in G2.

G4 — Fixed-law per run (immutability). For any stated result, the evolution map (time-stepping rule), operator coefficients, and boundary conditions are fixed for the entire run; no mid-run masking, rescaling, or operator edits occur.

G5 — Diagnostics are read-only (no feedback path). Any "identity", "seed", "gate", invariant, or estimator functional maps fields to labels/statistics and never modifies Φ, $w(d)$, any operator, any update map, or any collapse rule. Diagnostics may read Φ and derived energies, but do not alter the evolution map or its coefficients, and do not enter any subsequent update as inputs.

G6 — Symmetric exchange operators (real spectrum). The exchange operators $\hat{K}_{\mathrm{kin}}$ and $\hat{K}_{\mathrm{grav}}$ in Sec. 1.1.7 are real-symmetric and have real spectra. Spectral statements appear only in the Sec. 2.9 correspondence module. Between-collapse stability in the fundamental theory is given by exact Bell-sum conservation of the UEE tick map (Sec. 2.3).

Appendix C. 2. Quantum State Dynamics

Quantum state dynamics in the DO model comprise two processes:

1. Deterministic evolution in $S^{(4D)}$ via the Bell-sum–conserving UEE tick map (Sec. 2.3).
2. Collapse in $S^{(PD)}$ governed by the DO Probability Rule (Sec. 1.1.8), mirrored into $S^{(4D)}$ on the same tick by the Bell Identity g (Sec. 1.1.5). Collapse is a discrete probability event; the DO paper leaves open whether the collapse process is fully deterministic.

The Bell Identity g provides ontological mirroring between the state's dual representations throughout these processes.

Appendix C.1. 2.1. The Bell Identity (g)

The Bell Identity g is the non-dynamical, bijective ontological mirroring between a quantum state's dual representations in $S^{(4D)}$ and $S^{(PD)}$, underpinned by the Planck Identity (Sec. 1.1.4). It mirrors DS identity sphere-by-sphere, occupied support sets, and total hosted (non-CIE) intrinsic energy.

Key mappings.

1. Individual Bell Spheres:
$$\mathrm{DS}_k \longleftrightarrow P_k, \quad (\forall k \in \mathcal{O}(t)).$$
2. Bell Energy Field/Bell Energy Point mirroring: the single-state and N-body bijective mirroring statements are as defined in Sec. 1.1.5.

 doi:10.20944/preprints202405.2028.v8

3. Bell Sphere collections (Bell Field/Bell Point structures):

$$\mathcal{O}_{(\text{state})}(t) \longleftrightarrow \mathcal{O}^{(PD)}_{(\text{state})}. \tag{A15}$$

Properties of g:

- *Ontological unitarity*: conserves state identity and total hosted (non-CIE) intrinsic energy E_{QS}.
- *Timelessness of $S^{(PD)}$*: no internal time coordinate is defined in $S^{(PD)}$.
- *Non-dynamical*: g is distinct from UEE-governed evolution in $S^{(4D)}$.
- **Gravity absent in $S^{(PD)}$**: no gravitational quantity is encoded in Planck-Domain Bell Points; gravity arises from relational differences in hosted energies on $S^{(4D)}$, with uniform backgrounds canceling and DS CIE excluded from local sourcing (Sec. 2.2).
- *SOAN neutrality*: the mirroring g is defined relative to SOAN (Sec. 1.1.1); SOAN carries no fields and adds no degrees of freedom.

Appendix C.2. 2.2. Relational Gravitational Influence

Relational curvature in the DO model is carried by the fixed gravitational exchange structure on $S^{(4D)}$ (Secs. 1.1.7 and 2.3): the operator weights are fixed by the chosen realization and the state-dependence enters through Φ itself. For bookkeeping (collapse probabilities and auditing), the per-sphere hosted (non-CIE) energy sample is

$$e_b(t) = \begin{cases} |\Phi_b(t)|^2, & \text{quantum components,} \\ E_{c,b,\text{classical}}(t), & \text{classical hard-mass bodies.} \end{cases}$$

The constant intrinsic energy density (CIE) of Discrete Spheres is excluded from $e_b(t)$; CIE contributes only to the uniform background ρ_Λ used in cosmological accounting (Secs. 6.1 and 7.3–7.4).

Graph distance and fixed link structure. For a concrete representative hop-count index chart (not a background metric chart), take

$$d_{ab} = \max\big(|x_a - x_b|,\, |y_a - y_b|,\, |z_a - z_b|\big).$$

The kernel-induced link weights, neighborhood, and link set are those defined in Sec. 1.1.7:

$$w_{ab} := w(d_{ab}), \qquad \mathcal{N}(a) := \{\, b \neq a : \; w_{ab} > 0 \,\}, \qquad \mathcal{C} = \{(a,b) : \; w_{ab} > 0\}.$$

Relational gravity operator (exchange form). The exchange-form operator $\hat{K}_{\text{grav}}$ is as defined in Sec. 1.1.7 on $\mathcal{N}(a)$ with weights w_{ab}. It is graph-intrinsic and background-independent and acts directly on Φ on the fixed link set. Hard-mass coupling is stated separately in Sec. 2.4 through $m = E/c^2$, $\gamma(d)$, and the realization's discrete direction rule. No independent "potential-times-Φ" term is introduced.

Shell-gradient coefficient $\gamma(d)$ (realization-defined). The symbol $\gamma(d)$ denotes the realization's discrete radial-gradient coefficient on shell index d (Sec. 1.1.7). A centered-difference option is

$$\gamma(k) = \tfrac{1}{2}\big[w(k{-}1) - w(k{+}1)\big],$$

and the flux-normalized hard-mass convention specifies $\gamma(d)$ directly and recovers w by the inward integral $w(d) - w(d{+}1) = -2\,\gamma(d{+}1)$ at the run's cutoff radius (Sec. 1.1.7).

 doi:10.20944/preprints202405.2028.v8

Appendix C.3. 2.3 Unified Evolution Equation (Single-State)

Between collapse events, the UEE is implemented as a discrete-time tick map on the occupied Bell Spheres in $S^{(4D)}$, using the fixed relational link structure $\mathcal{C}$ (Sec. 1.1.7). For an isolated single state with occupied support $\mathcal{O}(t) \subset S^{(4D)}$, the conserved hosted (non-CIE) intrinsic energy used by the Bell Identity and the collapse rule is the Bell sum (Sec. 1.1.5):

$$E_{QS} \;=\; \sum_{\alpha\in\mathcal{O}(t)} \big|\Phi_\alpha^{(4D)}(t)\big|^2.$$

Convention (support). Between collapses, off-support DS carry zero hosted state content:

$$\Phi_a(t) = 0 \text{ for } a \notin \mathcal{O}(t), \qquad \sum_a \big|\Phi_a(t)\big|^2 = \sum_{\alpha\in\mathcal{O}(t)} \big|\Phi_\alpha(t)\big|^2.$$

Fixed link set and exchange weights. Let $\mathcal{C} = \{(a,b) : w_{ab} > 0\}$ be the fixed link set and $\mathcal{N}(a)$ the induced neighborhood (Sec. 1.1.7). Define the symmetric combined exchange weight on each link by

$$q_{ab} := \eta_{ab} + w_{ab}, \qquad q_{ab} = q_{ba} \geq 0, \qquad (a,b) \in \mathcal{C},$$

and define the per-link tick angle by

$$\theta_{ab} := \Delta t\, q_{ab}, \qquad \theta_{ab} = \theta_{ba}.$$

Shell-isotropic realizations satisfy $q_{ab} = q(d_{ab})$ and therefore $\theta_{ab} = \theta(d_{ab})$.

Fixed schedule (realization data). Fix a deterministic decomposition of $\mathcal{C}$ into disjoint matchings

$$\mathcal{C} = \bigsqcup_{r=1}^{R} \mathcal{C}^{(r)},$$

with no node appearing twice inside any $\mathcal{C}^{(r)}$. One tick applies the elementary link update on all links in exactly one matching $\mathcal{C}^{(r)}$. A realization fixes a schedule $r(K)$; the periodic schedule written here is one admissible choice.

As one admissible realization choice, fix the deterministic periodic schedule $r(K) \in \{1,\dots,R\}$ by

$$r(K) \;:=\; 1 + \big((K-1) \bmod R\big).$$

Tick $K \to K+1$ applies the elementary link update on all links in $\mathcal{C}^{(r(K))}$ only. The decomposition and schedule are part of the realization and remain fixed for the run.

Elementary link update (two-site exchange). For a single link $(a,b) \in \mathcal{C}$, define

$$s_{ab} := \frac{\Phi_a + \Phi_b}{2}, \qquad \delta_{ab} := \frac{\Phi_a - \Phi_b}{2}, \qquad s_{ab}^+ = s_{ab}, \qquad \delta_{ab}^+ = e^{\mathrm{i}\theta_{ab}}\,\delta_{ab},$$

and set

$$\Phi_a^+ = s_{ab}^+ + \delta_{ab}^+, \qquad \Phi_b^+ = s_{ab}^+ - \delta_{ab}^+.$$

Tick $K \to K+1$ applies this update in parallel on all links in the single matching $\mathcal{C}^{(r(K))}$ selected by the fixed schedule of the preceding paragraph, yielding the one-tick map

$$\Phi(t_{K+1}) \;=\; \mathcal{U}^{(r(K))}\big[\Phi(t_K)\big], \qquad t_K = t_0 + K\Delta t.$$

 doi:10.20944/preprints202405.2028.v8

The derived one-cycle (sweep) map over one full schedule pass is

$$\mathcal{U}_{\text{cycle}} := \mathcal{U}^{(R)} \circ \cdots \circ \mathcal{U}^{(1)}, \qquad \Phi(t_{K+R}) = \mathcal{U}_{\text{cycle}}\big[\Phi(t_K)\big].$$

On-site mass phase (admissible component). Define $\omega_m := mc^2/\hbar$. A per-site unit-modulus descriptor-rotation factor $e^{i\omega_m \Delta t}$ applied uniformly to Φ_a is an admissible on-site component of the same tick map and preserves the Bell sum pointwise.

Proposition (between-collapse Bell-sum conservation). The tick map defined above preserves the hosted (non-CIE) intrinsic energy:

$$\sum_a \big|\Phi_a(t+\Delta t)\big|^2 = \sum_a \big|\Phi_a(t)\big|^2.$$

Proof. Each elementary link update preserves $\big|\Phi_a\big|^2 + \big|\Phi_b\big|^2$ on that link because s_{ab} is unchanged and δ_{ab} is multiplied by a unit-modulus descriptor-rotation factor. Sites not on the updated link are unchanged during that sub-update. A product of such sub-updates therefore preserves $\sum_a |\Phi_a|^2$ exactly. □

Neutral uniform mode. For $\Phi_a = \Phi_0$ constant on a connected component, every link has $\delta_{ab} = 0$, so every elementary link update leaves Φ unchanged. Uniform backgrounds remain unchanged under the tick map.

Role split.

The UEE tick map updates the scalar state on occupied Bell Spheres in $S^{(4D)}$ using the fixed relational link structure $\mathcal{C}$. Electromagnetic observables (E, B) evolve by the discrete Maxwell constraints/transport of Sec. 2.5.2 on the same relational link structure in $S^{(4D)}$.

No mediator fields are introduced; updates are graph-intrinsic and share the invariant speed c.

Correspondence fence (reader-facing; Sec. 2.9). Section 2.9 records an optional long-wavelength correspondence module for realizations in the continuum-symbol subclass $w \in \mathcal{W}_\Delta$ (Sec. 1.1.9). KG-like correspondence and SR-dispersion statements are confined to that module and do not define the fundamental between-collapse tick map above.

Appendix C.4. 2.4. Unified Evolution Equation (N-body Generalization)

An N-body system with bodies $n = 1, \ldots, N$ is represented in $S^{(4D)}$ by component Bell Energy Fields $\Phi_n^{(4D)}(t)$ with supports $\mathcal{O}_n(t) \subset S^{(4D)}$ (Sec. 1.1.6), and in $S^{(PD)}$ by one unified N-body BEP (Secs. 1.1.5–1.1.6). Between collapses, each component field evolves on the same fixed link set by the Bell-sum–conserving UEE tick map of Sec. 2.3, with per-body on-site mass phase $\omega_{m_n} = m_n c^2/\hbar$.

Total hosted (non-CIE) intrinsic energy (Bell sum). The conserved hosted (non-CIE) intrinsic energy of the unified N-body state is the Bell sum (Sec. 1.1.6):

$$E_{QS,\text{total}}^{(N\text{-body})} = \sum_{n=1}^{N} \sum_{\alpha \in \mathcal{O}_n(t)} \big|\Phi_{n,\alpha}^{(4D)}(t)\big|^2 = \text{constant between collapses.} \tag{A16}$$

This conserved quantity is the energy scalar used for Bell-Identity bookkeeping and for collapse probabilities (Secs. 1.1.5–1.1.8 and Sec. 3.2).

Hard-mass microstate (singleton support). For each body $n = 1, \ldots, N$, the $S^{(4D)}$ representation is a component field $\Phi_n^{(4D)}(t)$ whose occupied support is a singleton at every tick:

$$\mathcal{O}_n(t) = \{\alpha_n(t)\}. \tag{A17}$$

 doi:10.20944/preprints202405.2028.v8

Dispersive Bell Energy Fields are not used to model orbital motion; orbital validations are conducted only in the singleton-support hard-mass regime. The hosted (non-CIE) intrinsic energy of body n is

$$E_n := \sum_{\alpha \in \mathcal{O}_n(t)} \left| \Phi^{(4D)}_{n,\alpha}(t) \right|^2, \tag{A18}$$

conserved between collapses.

Coarse-graining (DO-admissible). In this regime, the dynamical degrees carried forward in $S^{(4D)}$ are the support indices $\alpha_n(t)$ (equivalently the support-index node identifiers). The hard-mass evolution law is the DO-admissible coarse-graining of the exchange dynamics to these support indices.

Operator-to-force link. Define the discrete hosted-energy source density on $S^{(4D)}$ by

$$\rho_E(x,t) := \sum_{j=1}^{N} E_j \, \delta^{\mathrm{K}}_{x,\alpha_j(t)}, \tag{A19}$$

with $\delta^{\mathrm{K}}_{x,\alpha}$ the Kronecker delta on DS nodes. The realization's radial kernel data (Sec. 1.1.7) induce a shell-gradient coefficient $\gamma(d)$; the hard-mass support-index law uses the corresponding central, reciprocal interaction magnitude $\gamma\big(d_{ij}(t)\big)$ evaluated on the shell index

$$d_{ij}(t) := d\big(\alpha_i(t), \alpha_j(t)\big), \tag{A20}$$

together with the realization's background-independent direction rule $\hat{\mathbf{u}}_{ij}(t)$.

Interpretation. This is the classical-limit regime used in Appendix B: singleton-support bodies evolved by the realization's discrete shell-gradient $\gamma(d)$ and direction rule, with no dispersive spreading.

Hard-mass support-index coupling law (Appendix B operative form).

Status (coarse-grained regime). The hard-mass support-index evolution law below is an admissible DO coarse-graining of the hard-mass classical-limit regime: each body is treated as singleton-support per tick and only the support indices propagate under the realization's $\gamma(d)$ and direction rule. Appendix A records this admissible regime; Appendix B instantiates fixed realizations and audits long-horizon invariants under that regime. Appendix A does not assert uniqueness of this coarse-graining. In the hard-mass validations, mutual interaction between bodies is evaluated from their hosted (non-CIE) intrinsic energies via $m = E/c^2$, the shell-gradient coefficient $\gamma(d)$, and the background-independent direction rule (Sec. 1.1.7 and Appendix B). The operative pairwise interaction 3-component force (index-chart form is)

$$\mathbf{F}_{ij}(t) = \chi_E \, m_i m_j \, \gamma\big(d_{ij}(t)\big) \, \hat{\mathbf{u}}_{ij}(t),$$

where (x_i, y_i, z_i) are the realization's index-chart coordinates for the DS node $\alpha_i(t)$ (Sec. 6.3). Define the chart-evaluated Chebyshev separation

$$d_{ij}(t) = \max\big\{|x_i - x_j|,\ |y_i - y_j|,\ |z_i - z_j|\big\} \in \mathbb{R}_{\geq 0}.$$

In integer index-chart realizations, $d_{ij}(t) \in \mathbb{Z}_{\geq 0}$ is the hop-shell index. In hard-mass support-index realizations, $d_{ij}(t) \in \mathbb{R}_{\geq 0}$ is read from the diagnostic index chart and $\gamma(d)$ and $w(d)$ are evaluated at non-integer d by the realization's fixed shell interpolation convention (Appendix B). The direction $\hat{\mathbf{u}}_{ij}(t)$ is the realization's discrete direction rule (Manhattan–fraction in Appendix B).

 doi:10.20944/preprints202405.2028.v8

Discrete radial-gradient notation (convention).

Define the realization-matched discrete radial-gradient operator $\nabla_d^{(\gamma)}$ by

$$\nabla_d^{(\gamma)} w(d) := -\gamma(d).$$

Then the same interaction law is written equivalently as

$$\mathbf{F}_{ij}(t) = -\chi_E\, m_i m_j\, \nabla_d^{(\gamma)} w\big(d_{ij}(t)\big)\, \hat{\mathbf{u}}_{ij}(t),$$

with $\nabla_d^{(\gamma)}$ denoting the discrete derivative convention corresponding to the chosen γ realization.

Equivalence consequence (DO).

With $m_n \equiv E_n/c^2$,

$$a_i = \frac{\mathbf{F}_i}{m_i} = -\chi_E \sum_{j\neq i} m_j\, \nabla_d^{(\gamma)} w(d_{ij})\, \hat{\mathbf{u}}_{ij} = \chi_E \sum_{j\neq i} m_j\, \gamma(d_{ij})\, \hat{\mathbf{u}}_{ij},$$

so acceleration is independent of the test body's mass parameter.

Local scope (component-wise).

For an N-body state, the hard-mass coupling acts per component on each connected component of $\mathcal{C}$ (Sec. 1.1.7), with the same $\gamma(d)$-based coupling and the realization's background-independent direction rule.

Proposition (Angular-momentum conservation; support-index realization notation).

Note (time notation). In this proposition, overdot notation is shorthand for the realized support-index update law used in the hard-mass validation regime. Discrete-time implementations use the per-tick difference form $\Delta\mathbf{v}_i/\Delta t$ and $\Delta\mathbf{L}/\Delta t$ under the fixed integrator rule. This introduces no continuum time structure into the DO ontology. Consider N hard-mass bodies at sites $\mathbf{r}_i$ with hosted (non-CIE) intrinsic energies E_i and $m_i = E_i/c^2$. Let the energy density be $\rho_E(\mathbf{r}) = \sum_{j=1}^N E_j\, \delta^{\mathrm{K}}_{\mathbf{r},\mathbf{r}_j}$ on the discrete graph (Kronecker delta on the DS node set), and let $\mathcal{K}$ be a finite-range link operator with kernel $\mathbf{K}(\Delta)$ satisfying: (i) neutrality $\sum_\Delta \mathbf{K}(\Delta) = \mathbf{0}$, (ii) shell-isotropy/antisymmetry $\mathbf{K}(-\Delta) = -\mathbf{K}(\Delta)$ with coefficients depending only on $d = \|\Delta\|_\infty$, (iii) discrete centrality so that $\mathbf{K}(\Delta) \parallel \Delta$. Define

$$\mathbf{g}(\mathbf{r}) = \chi_E\, (\mathcal{K}[\rho_E])(\mathbf{r}), \qquad m_i\, \dot{\mathbf{v}}_i = E_i\, \mathbf{g}(\mathbf{r}_i),$$

and the energy-weighted barycenter $\mathbf{r}_{\mathrm{bary}} = \big(\sum_i E_i \mathbf{r}_i\big)/\big(\sum_i E_i\big)$. With periodic (or torque-free) boundaries, the total angular momentum

$$\mathbf{L} = \sum_{i=1}^N m_i\, (\mathbf{r}_i - \mathbf{r}_{\mathrm{bary}}) \times \mathbf{v}_i$$

is invariant, $\dot{\mathbf{L}} = \mathbf{0}$.

Proof. Write $r_{ij} = \mathbf{r}_i - \mathbf{r}_j$. By linearity, shell-isotropy and evenness, $\mathbf{g}_{ij} = \chi_E\, E_j\, \mathbf{K}(r_{ij})$ with $\mathbf{K}(-r_{ij}) = -\mathbf{K}(r_{ij})$. Hence $\mathbf{F}_{ij} = E_i\, \mathbf{g}_{ij} = \chi_E\, E_i E_j\, \mathbf{K}(r_{ij}) = -\mathbf{F}_{ji}$. Then

$$\dot{\mathbf{L}} = \frac{1}{2} \sum_{i\neq j} (\mathbf{r}_i - \mathbf{r}_j) \times \mathbf{F}_{ij} = \frac{1}{2} \sum_{i\neq j} r_{ij} \times \chi_E\, E_i E_j\, \mathbf{K}(r_{ij}) = \mathbf{0},$$

since $\mathbf{K}(r_{ij}) \parallel r_{ij}$ makes each cross product zero. Neutrality and periodic (or torque-free) boundaries exclude external torques. □

 doi:10.20944/preprints202405.2028.v8

Appendix C.4.1. 2.4.1. Validation Protocols

Note. The checks listed here are illustrative and implementation-neutral; detailed run protocols belong in Appendix B. To verify an implementation of the hard-mass support-index coupling and its auditing, report:

- **Barycenter drift:** $\mathbf{r}_{\text{bary}}(t) = \dfrac{\sum_n m_n \mathbf{r}_n(t)}{\sum_n m_n}$, and $\max_t \|\mathbf{r}_{\text{bary}}(t) - \mathbf{r}_{\text{bary}}(0)\|$.
- **Energy auditing (run-level invariant vs. texture):** report the conserved audit scalar adopted by the realization (Appendix B uses $I_{\text{DO}} = E_{\text{kin}} + U - W_{\text{closure}}$ as the conservation gate; preserved console artifacts may label E_{kin} as K), with net drift and RMS over the analysis window. Report $E_{\text{naive}}(t) = E_{\text{kin}}(t) + U(t)$ only as a texture diagnostic.

Appendix C.4.2. 2.4.2. Symmetry and Linearity Gates (model-level)

Rotation-invariance (relabeling) gate. For any observable functional $\mathcal{F}$ built from Φ on $S^{(4D)}$ with a radial kernel $w(d)$ satisfying Sec. 1.1.9, $\mathcal{F}$ is invariant under rigid permutations of coordinate axes and node relabelings (including any element of the cubic symmetry group O_h). Operationally: recomputing $\mathcal{F}$ after $x \leftrightarrow y$, $y \leftrightarrow z$, $x \leftrightarrow z$ yields the same value up to numerical error.

ε-linearity probe. For any scalar $\varepsilon > 0$, the rescaling $\Phi \mapsto \varepsilon\Phi$ implies quadratic observables scale as ε^2; e.g. $C_\ell[\varepsilon\Phi] = \varepsilon^2 C_\ell[\Phi]$ within numerical precision.

Appendix C.4.3. 2.4.3. Well-posedness, stability, and locality (UEE tick map)

Setting. Let G be the discrete spatial graph in $S^{(4D)}$ with fixed link set $\mathcal{C}$ (Sec. 1.1.7). Between collapses, the UEE evolution is the deterministic one-tick map indexed by the fixed schedule $r(K)$ (Sec. 2.3):

$$\Phi(t_{K+1}) = \mathcal{U}^{(r(K))}\big[\Phi(t_K)\big], \qquad t_K = t_0 + K\Delta t,$$

where $\mathcal{U}^{(r(K))}$ applies the elementary link update in parallel on all links of the single matching $\mathcal{C}^{(r(K))} \subset \mathcal{C}$.

Proposition (existence/uniqueness).

For any initial $\Phi(t_0)$, the fixed tick map defines a unique global sequence $\{\Phi(t_0 + n\Delta t)\}_{n\in\mathbb{Z}_{\geq 0}}$.

Proposition (ℓ^2 stability and Bell-sum conservation).

The tick map preserves the Bell sum exactly (Sec. 2.3):

$$\sum_a \big|\Phi_a(t_0 + n\Delta t)\big|^2 = \sum_a \big|\Phi_a(t_0)\big|^2 \quad \text{for all } n.$$

Therefore Φ is uniformly bounded in the discrete ℓ^2 norm over all ticks between collapses.

Finite-dependence bound (realization locality).

Let R_w denote the realization's finite link cutoff in hop distance (Sec. 1.1.7): $w(d) = 0$ for $d > R_w$. Each tick applies exchanges only on the single matching $\mathcal{C}^{(r(K))} \subset \mathcal{C}$ selected by the fixed schedule (Sec. 2.3), and every exchanged link satisfies $d_{ab} \leq R_w$. Therefore, dependence expands at most R_w hops per tick:

$$\operatorname{supp}\Phi(t_0 + n\Delta t) \subset \{\, b : \; d(b, \operatorname{supp}\Phi(t_0)) \leq nR_w \,\}.$$

This is a realization-level dependence bound. In the nearest-neighbor exchange realization ($R_w = 1$), it reduces to $d(b, \operatorname{supp}\Phi(t_0)) \leq n$. The invariant causal bookkeeping convention remains $c = 1$ hop/tick on the underlying DS neighbor graph (Sec. 6.4).

 doi:10.20944/preprints202405.2028.v8

Appendix C.4.4. 2.4.4. Collected Closures (read-only; recap of Secs. 2.3–2.4 and Sec. 1.1.8)

This subsection collects the three closure identities used throughout Appendix A and audited in Appendix B. No new objects are introduced here.

(C1) Between-collapse Bell-sum conservation (UEE tick map). Between collapses, the UEE tick map preserves the hosted (non-CIE) intrinsic energy (Bell sum):

$$\sum_a \left|\Phi_a(t+\Delta t)\right|^2 = \sum_a \left|\Phi_a(t)\right|^2. \tag{A21}$$

This is the primary between-collapse conservation identity (Sec. 2.3).

(C2) Energy preservation at collapse (single rescaling). Let $O(t_{\mathrm{pre}}) = \mathrm{Supp}\,\Phi^{(4D)}(t_{\mathrm{pre}})$ and $e_\alpha(t) = |\Phi_\alpha(t)|^2$. For any admissible connected outcome $O' \subseteq O(t_{\mathrm{pre}})$ with $E_{O'}(t_{\mathrm{pre}}) > 0$, define

$$\lambda(O') := \frac{\sum_{\alpha\in O(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}})}{\sum_{\alpha\in O'} e_\alpha(t_{\mathrm{pre}})}, \qquad e_\alpha(t_{\mathrm{post}}) := \begin{cases} \lambda(O')\, e_\alpha(t_{\mathrm{pre}}), & \alpha \in O', \\ 0, & \alpha \notin O'. \end{cases} \tag{A22}$$

Then total hosted (non-CIE) intrinsic energy is preserved at collapse:

$$\sum_{\alpha\in O(t_{\mathrm{post}})} e_\alpha(t_{\mathrm{post}}) = \sum_{\alpha\in O(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}}). \tag{A23}$$

This is the collapse preservation identity (Sec. 1.1.8).

(C3) Hard-mass audit scalar (Appendix B gate). In the hard-mass classical-limit validations (Appendix B), conservation auditing uses the audit scalar

$$I_{\mathrm{DO}} := E_{\mathrm{kin}} + U - W_{\mathrm{closure}}. \tag{A24}$$

The quantities U and W_{closure} are audit-only ledgers used for post-update conservation checks and have no role in any force law, operator, tick-map update, kernel choice, schedule, or collapse rule. Appendix B defines and implements I_{DO}, audits long-horizon stability using I_{DO} as the conservation gate, and may report $E_{\mathrm{kin}} + U$ using the legacy console label `K+U` as a texture diagnostic.

Appendix C.5. 2.5. Fundamental Discrete Equations

Appendix C.6. 2.5.1. Discrete Spinor Observable Update

Between collapse events, the discrete evolution of spinor observables $\Xi_a(t)$ at DS a in $S^{(4D)}$ is governed by a discrete Dirac-type update:

$$\frac{\Xi_a(t+\Delta t) - \Xi_a(t)}{\Delta t} = -\frac{i}{\hbar}\,\hat{H}_a\,\Xi_a(t), \tag{A25}$$

with discrete spinor operator

$$\hat{H}_a = \beta\, mc^2 + c\sum_{k=1}^{3} \alpha_k\, \hat{p}_k^{(a)}. \tag{A26}$$

Here $\hat{H}_a$ is an operational generator for spinor observables on Bell Spheres in $S^{(4D)}$, using the fixed relational link structure $\mathcal{C}$; it is not a Hilbert-space Hamiltonian and introduces no independent field degrees of freedom.

The Dirac matrices satisfy

$$\{\alpha_i, \alpha_j\} = 2\delta_{ij}\mathbb{I}_4, \quad \{\alpha_i, \beta\} = 0, \quad \beta^2 = \mathbb{I}_4. \tag{A27}$$

 doi:10.20944/preprints202405.2028.v8

The discrete momentum observable is represented by symmetric exchange across the fixed link set (Sec. 1.1.7). In this observable block the exchange weights are the kinetic conductances $\eta_{ab} = \eta_{ba}$:

$$\big(\hat{p}_k \Xi\big)_a := \frac{\hbar}{2\mathrm{i}} \sum_{b \in \mathcal{N}(a)} \eta_{ab}\, \sigma_k(a,b)\, \big(\Xi_b - \Xi_a\big), \qquad \sigma_k(a,b) = \mathrm{sgn}[(x_b - x_a)\cdot \hat{e}_k]. \tag{A28}$$

Here x_a denotes the index-chart coordinate vector used for hop counting on the DS network (with $(x_b - x_a)\cdot \hat{e}_k$ selecting its k-th component); it is not a background metric coordinate.

Scope (Dirac).

Spin and other fermionic observables are represented within the DS/BEP formalism; the DO does not posit an ontic spinor field. References to Dirac are reader-facing correspondence only; observable predictions correspond to discrete updates on Bell Spheres in $S^{(4D)}$ using the fixed relational link structure $\mathcal{C}$, and collapse in $S^{(PD)}$ mirrored to $S^{(4D)}$ by g.

Appendix C.7. 2.5.2. Discrete Maxwell Relations

Between collapse events, Appendix A records a discrete Maxwell block for electromagnetic observables on Bell Spheres in $S^{(4D)}$. Electric ($\mathbf{E}_a(t)$) and magnetic ($\mathbf{B}_a(t)$) observables at Bell Sphere a evolve discretely according to the tick update below. The update uses a staggered relational structure: $\mathbf{E}$ is represented at Bell Spheres, while $\mathbf{B}$ is represented on the associated relational structure used to evaluate the discrete curl. The operators $\nabla_g\cdot$ and $\nabla_g\times$ are the discrete divergence and curl defined from this same relational structure, chosen to satisfy discrete Gauss and Stokes.

$$\frac{\mathbf{E}_a(t+\Delta t) - \mathbf{E}_a(t)}{\Delta t} = c^2 \big(\nabla_g \times \mathbf{B}\big)_a - \frac{\mathbf{J}_a}{\varepsilon_0}, \tag{A29}$$

$$\frac{\mathbf{B}_a(t+\Delta t) - \mathbf{B}_a(t)}{\Delta t} = -\big(\nabla_g \times \mathbf{E}\big)_a, \tag{A30}$$

$$\big(\nabla_g \cdot \mathbf{E}\big)_a = \frac{\rho_a}{\varepsilon_0}, \quad \big(\nabla_g \cdot \mathbf{B}\big)_a = 0. \tag{A31}$$

In internal units set $\varepsilon_0 = 1$; restoration to SI follows the unit map in Scope and Linkage (Appendix A).

Electromagnetic observables on Bell Spheres in $S^{(4D)}$. The quantities $\mathbf{E}$ and $\mathbf{B}$ are discrete electromagnetic observables in $S^{(4D)}$. They are not ontologically fundamental mediators arising from independent gauge fields. The electromagnetic energy per Bell Sphere is

$$e_{\mathrm{EM}} = \tfrac{1}{2}\big(|E|^2 + |B|^2\big).$$

Here e_{EM} is a read-only observable-derived energy density for electromagnetic observables on $S^{(4D)}$. In particular, e_{EM} is not an additional addend to the hosted-energy samples $e_a(t) = |\Phi_a(t)|^2$; collapse bookkeeping and gravitational sourcing use only the hosted (non-CIE) samples. It is not an independent ontic energy variable and does not modify the hosted-energy bookkeeping $e_a(t) = |\Phi_a(t)|^2$ used for collapse bookkeeping or gravitational sourcing. This Maxwell observable block is stated at the model-definition level in Appendix A and is not exercised in the Appendix B validation suite.

Appendix C.7.1. 2.5.3 Scope and Consistency of Discrete Equations

The UEE tick map of Sec. 2.3 governs the scalar state descriptor $\Phi^{(4D)}(t)$ on Bell Spheres in $S^{(4D)}$ (and thus the associated Bell Energy Field samples $e_\alpha(t) = |\Phi^{(4D)}_\alpha(t)|^2$); the discrete Dirac update governs spinor observables; and the discrete Maxwell block governs electromagnetic observables (E, B) on Bell Spheres in $S^{(4D)}$ using the fixed relational link structure $\mathcal{C}$. KG-like and SR-dispersion correspondence statements for suitable realizations are confined to the optional module in Sec. 2.9.

 doi:10.20944/preprints202405.2028.v8

Terminology (observables vs. fields). Throughout Appendix A, (E, B) are operational observables defined on $S^{(4D)}$ (via graph divergence/curl and continuity constraints); they are not ontic mediator fields.

Role split. The UEE tick map updates the scalar state on occupied Bell Spheres in $S^{(4D)}$ using the fixed relational link structure $\mathcal{C}$. Electromagnetic observables (E, B) evolve by the discrete Maxwell constraints/transport of Sec. 2.5.2 (discrete divergence/curl and charge continuity) on the same relational link structure $\mathcal{C}$ in $S^{(4D)}$. No mediator fields are introduced; all updates are intrinsic to the $S^{(4D)}$ relational link structure and share the invariant speed c.

Appendix C.8. 2.6. Bohm/EPR Correlations in the DO Model (collapse in $\mathbf{S^{(PD)}}$, mirrored into $\mathbf{S^{(4D)}}$)

The DO model treats the Bohm/EPR experiment as a collapse phenomenon governed by a unified Planck-Domain (PD) Bell Energy Point and the Bell Identity g, rather than as an interaction mediated by an independent spacetime field. The pre-collapse state is represented by one unified N-body Bell Energy Point in $S^{(PD)}$, mirrored to multiple Bell Energy Fields in $S^{(4D)}$. A Physical Interaction (PI) in $S^{(4D)}$ triggers collapse, which occurs only in $S^{(PD)}$ and is mirrored to $S^{(4D)}$ on the same tick by g.

Appendix C.8.1. 2.6.1. Initial Unified State Representation

A two-body singlet configuration of spin-$\frac{1}{2}$ entities A, B is represented as follows.

I. $S^{(PD)}$ representation (unified Bell Energy Point).

- The system is represented by a single unified two-body Bell Energy Point $\Phi_{AB}^{(PD)}$, encoding the total hosted (non-CIE) intrinsic energy $E_{QS,\text{total}}^{(AB)}$ and the singlet spin anti-correlation constraint.
- The unified BEP encodes the singlet anti-correlation constraint: for any common analyzer axis $\mathbf{A}$, the paired outcomes are opposite,

$$o_B(\mathbf{A}) = -\,o_A(\mathbf{A}), \qquad o_A(\mathbf{A}),\, o_B(\mathbf{A}) \in \{-1, +1\}.$$

- The unified BEP is timeless and metric-free; it carries no internal spatial separation and supports no internal position observable in $S^{(PD)}$. The DS-identifying (x_i, y_i, z_i) triples in the $(3 \times N)$ Planck representation (Sec. 1.1.3) are identity labels inherited from $S^{(4D)}$, not internal separations in $S^{(PD)}$.

II. $S^{(4D)}$ representation (paired Bell Energy Fields).

- By the Bell Identity g (Sec. 2.1), the unified $S^{(PD)}$ state is mirrored to $S^{(4D)}$ as two Bell Energy Fields $\Phi_A^{(4D)}(t)$ and $\Phi_B^{(4D)}(t)$ supported on $\mathcal{O}_A(t)$ and $\mathcal{O}_B(t)$.
- The per-sphere hosted energy samples are $e_{A,\alpha}(t) = |\Phi_{A,\alpha}^{(4D)}(t)|^2$ and $e_{B,\alpha}(t) = |\Phi_{B,\alpha}^{(4D)}(t)|^2$, and the total hosted (non-CIE) intrinsic energy is conserved between collapses (Sec. 1.1.5).

Appendix C.8.2. 2.6.2. Collapse of the Unified State

A PI in $S^{(4D)}$ (Sec. 3.1), involving A or B at time t_{coll}, triggers collapse.

I. Transformation in $S^{(PD)}$ (collapse and factorization).

- A PI in $S^{(4D)}$ triggers collapse, and the Collapse Operator C (Sec. 1.1.8) acts on the unified AB BEP in $S^{(PD)}$:

$$C(\Phi_{AB}^{(PD)}) \equiv (\phi_A^{(PD)}, \phi_B^{(PD)}). \tag{A32}$$

- The phrase "factorized state" denotes DO factorization after collapse as ontological bookkeeping in $S^{(PD)}$; it is not a Hilbert-space tensor product.
- The post-collapse pair $(\phi_A^{(PD)}, \phi_B^{(PD)})$ encodes definite anti-correlated spin outcomes consistent with the singlet anti-correlation constraint stated in Sec. 2.6.1.

 doi:10.20944/preprints202405.2028.v8

II. Mirroring to $S^{(4D)}$ (localized Bell Energy Fields).

- By g (Sec. 2.1), the $S^{(PD)}$ factorization (A32) is mirrored to $S^{(4D)}$ on the same tick t_{coll}.
- The pre-collapse Bell Energy Fields $\Phi_A^{(4D)}(t_{\text{pre}})$, $\Phi_B^{(4D)}(t_{\text{pre}})$ are replaced by new, generally localized Bell Energy Fields $\Phi_{A,\text{coll}}^{(4D)}(t_{\text{coll}})$, $\Phi_{B,\text{coll}}^{(4D)}(t_{\text{coll}})$ whose supports are confined to admissible connected subsets of the pre-collapse supports (Sec. 1.1.8).
- Post-collapse total hosted (non-CIE) intrinsic energy is preserved by the Sec. 1.1.8 rescaling rule, and collapse selects only from pre-existing Bell Spheres in the pre-collapse support.

Appendix C.8.3. 2.6.3. Collapse Correlation Rule

The observable correlation $E(\mathbf{A},\mathbf{B})$ between analyzer axes $\mathbf{A},\mathbf{B}$ arises at collapse in $S^{(PD)}$, mirrored in $S^{(4D)}$ as paired dichotomic outcomes $\{-1,+1\}$ from one unified outcome selection.

The variable $\mathbf{u}$ is an analytic integration parameter for population averaging; it is not an ontic spin vector object in the DO model.

Notation. Let $\nu \in \mathbb{Z}_{\geq 2}$ denote the interaction-manifold amplification factor used in the population average below. This symbol is distinct from hop-distance notation d_{ab} used elsewhere in Appendix A.

Let $\mathbf{A},\mathbf{B}$ be unit analyzer axes in the ν-degree-of-freedom interaction manifold associated with the PI. Let $\mathbf{u}$ be a latent direction variable taken uniform on $S^{\nu-1}$ for population averaging. For analytic evaluation, represent the alignment scalars by

$$a(\mathbf{A},\mathbf{u}) := \mathbf{A}\cdot\mathbf{u}, \qquad b(\mathbf{B},\mathbf{u}) := -\mathbf{B}\cdot\mathbf{u},$$

where the minus sign encodes the singlet anti-correlation.

The DO correlation is the population average

$$E(\mathbf{A},\mathbf{B}) = \nu \left\langle a(\mathbf{A},\mathbf{u})\, b(\mathbf{B},\mathbf{u}) \right\rangle = -\nu \left\langle (\mathbf{A}\cdot\mathbf{u})(\mathbf{B}\cdot\mathbf{u}) \right\rangle. \tag{A33}$$

Uniformity on $S^{\nu-1}$ gives $\langle \mathbf{u} \rangle = \mathbf{0}$ and $\langle \mathbf{u}\mathbf{u}^\top \rangle = \frac{1}{\nu}\mathbf{I}$, hence

$$\left\langle (\mathbf{A}\cdot\mathbf{u})(\mathbf{B}\cdot\mathbf{u}) \right\rangle = \mathbf{A}^\top \langle \mathbf{u}\mathbf{u}^\top \rangle \mathbf{B} = \frac{\mathbf{A}\cdot\mathbf{B}}{\nu},$$

and therefore

$$E(\mathbf{A},\mathbf{B}) = -\mathbf{A}\cdot\mathbf{B}.$$

Corollaries: planar spin-$\frac{1}{2}$ interaction ($\nu = 2$) yields the factor $\nu = 2$; a fully 3D interaction manifold ($\nu = 3$) yields $\nu = 3$.

Appendix C.8.4. 2.6.4. CHSH bound (DO)

Setting. Sec. 2.6.3 yields, for planar spin-$\frac{1}{2}$ with singlet anti-correlation and rotational symmetry, the correlation

$$E(\mathbf{A},\mathbf{B}) \;=\; -\mathbf{A}\cdot\mathbf{B},$$

for unit analyzer axes $\mathbf{A},\mathbf{B}$ in the plane and dichotomic readouts ± 1.

Theorem (CHSH bound $2\sqrt{2}$).

For any coplanar unit axes $\mathbf{A},\mathbf{A}',\mathbf{B},\mathbf{B}'$ and correlation $E(\mathbf{X},\mathbf{Y}) = -\mathbf{X}\cdot\mathbf{Y}$, the CHSH functional

$$S \;=\; E(\mathbf{A},\mathbf{B}) - E(\mathbf{A},\mathbf{B}') + E(\mathbf{A}',\mathbf{B}) + E(\mathbf{A}',\mathbf{B}')$$

satisfies $|S| \leq 2\sqrt{2}$, with equality at the canonical 45° choice.

Proof. Rewrite

$$S \;=\; -\mathbf{A}\cdot(\mathbf{B}-\mathbf{B}') \;-\; \mathbf{A}'\cdot(\mathbf{B}+\mathbf{B}').$$

 doi:10.20944/preprints202405.2028.v8

By Cauchy–Schwarz and $\|\mathbf{A}\| = \|\mathbf{A}'\| = 1$,

$$|S| \;\le\; \|\mathbf{B} - \mathbf{B}'\| + \|\mathbf{B} + \mathbf{B}'\| \;=\; \sqrt{2 - 2\,\mathbf{B}\cdot\mathbf{B}'} + \sqrt{2 + 2\,\mathbf{B}\cdot\mathbf{B}'}\,.$$

The right-hand side is maximized at $\mathbf{B}\cdot\mathbf{B}' = 0$, giving $|S| \le 2\sqrt{2}$. Equality holds when $\mathbf{A} \parallel (\mathbf{B} - \mathbf{B}')$ and $\mathbf{A}' \parallel (\mathbf{B} + \mathbf{B}')$, e.g. with $\angle(\mathbf{A}, \mathbf{B}) = \angle(\mathbf{A}', \mathbf{B}) = 45^\circ$ and $\angle(\mathbf{B}, \mathbf{B}') = 90^\circ$. □

Remark. This bound follows directly from the DO correlation of Sec. 2.6.3 and does not invoke Hilbert-space machinery; Appendix B reproduces the same value in the CHSH validation.

Appendix C.8.5. 2.6.5 DO-Locality Schema (settings, PI, and no-signal)

Terminology. A *setting* is a macroscopic device orientation (axis choice). A *Physical Interaction (PI)* is the localized coupling in $S^{(4D)}$ that triggers collapse.

Axiom (Free-setting independence; operational).

Let $\phi^{(PD)}$ denote the pre-collapse Bell Energy Point. Let S_A, S_B be the macroscopic setting choices at two spacelike-separated PIs. For device modeling the DO adopts the statistical independence

$$\mathsf{P}(S_A, S_B \,|\, \phi^{(PD)}) \;=\; \mathsf{P}(S_A, S_B).$$

This is an operational assumption about devices; it does not assert exclusion of global determinism.

Theorem (Remote-setting invariance / no-signal).

Under the DO Probability Rule (energy-fraction collapse; Sec. 3.2), the marginal probabilities on one wing depend only on the pre-collapse hosted (non-CIE) energy within that wing's region and are independent of the remote setting:

$$P_s(B_\ell) \;=\; \frac{\sum_{\alpha\in B_\ell} e_\alpha}{E_{QS}} \quad \text{for every setting } s \text{ at } A,$$

as proved in Sec. 3.2.

A PD collapse selects a single admissible outcome $O' \in \mathrm{Conn}(\mathcal{C})$ for the unified state; the mirrored $S^{(4D)}$ localization yields paired readouts on both wings from the same selection. In general,

$$P(a, b \,|\, S_A, S_B, \phi^{(PD)}) \;\neq\; P(a \,|\, S_A, \phi^{(PD)})\, P(b \,|\, S_B, \phi^{(PD)}),$$

so factorization fails while remote-setting invariance and the operational free-setting axiom both hold. This non-factorization underlies the correlation $E(\mathbf{A}, \mathbf{B}) = -\,\mathbf{A}\cdot\mathbf{B}$ used in Sec. 2.6.4.

Appendix C.9. 2.7. Double-Slit Experiment in the DO Model

The DO model provides an ontological and dynamical account of the double-slit experiment grounded in the physical distribution of the quantum state's hosted (non-CIE) intrinsic energy on $S^{(4D)}$ and the collapse rule acting in $S^{(PD)}$ and mirrored to $S^{(4D)}$ by the Bell Identity g.

Appendix C.9.1. 2.7.1. Initial State and Coherent Multi-Site Occupancy in $S^{(4D)}$

A quantum state (total hosted (non-CIE) intrinsic energy E_{QS}) interacts with a two-slit barrier (slits A, B). Prior to interaction with the barrier, the state is represented in $S^{(4D)}$ by a single Bell Field and in $S^{(PD)}$ by a single Bell Point. As the state diffracts through slits A and B, its Bell Field splits into two separate fields in $S^{(4D)}$: Bell Field A and Bell Field B. In $S^{(PD)}$ the two diffracted $S^{(4D)}$ fields remain unified as a single Bell Point AB. The quantum state's intrinsic energy is represented by Bell Energy Fields A and B in 4D spacetime and by a single Bell Energy Point AB in the Planck Domain.

I. Post-interaction $S^{(4D)}$ configuration.

 doi:10.20944/preprints202405.2028.v8

- Following diffraction through slits A and B, the state's Bell Field splits into two separate fields in $S^{(4D)}$: Bell Field A and Bell Field B. The quantum state's intrinsic energy in $S^{(4D)}$ is represented by two physically real Bell Energy Fields: Bell Energy Field A and Bell Energy Field B, with coherent state descriptors $\Phi_A^{(4D)}(t)$ and $\Phi_B^{(4D)}(t)$.
- In overlap regions the state descriptor combines coherently at each Bell Sphere γ by descriptor addition on $S^{(4D)}$,
$$e_\gamma(t) = \left|\Phi_{A,\gamma}^{(4D)}(t) + \Phi_{B,\gamma}^{(4D)}(t)\right|^2,$$
yielding interference modulation of the energy distribution on $S^{(4D)}$.
- **No "superposition of location" in DO.** The distributed set $\{e_\gamma(t)\}$ is a physically real multi-site energy occupancy on $S^{(4D)}$; "location" means the occupied Bell Spheres carrying nonzero energy samples.
- Total hosted (non-CIE) intrinsic energy is conserved: $E_{QS} = \sum_\gamma e_\gamma(t)$.

II. $S^{(PD)}$ representation (Bell Energy Point).

- In $S^{(PD)}$ the two diffracted $S^{(4D)}$ fields remain unified as a single Bell Point AB; its quantum energy component is the single Bell Energy Point AB, representing the indivisible hosted (non-CIE) intrinsic energy E_{QS}.
- **Location is not a BEP observable.** The BEP is timeless and metric-free; it supports no internal position observable and no internal separation. In DO, "location" means the definite $S^{(4D)}$ Bell-sphere energy distribution itself, not an indeterminate alternative among positions.
- Bell Identity g (Sec. 2.1) mirrors Bell Point AB and Bell Energy Point AB in $S^{(PD)}$ to Bell Field A and Bell Field B (and their Bell Energy Fields) in $S^{(4D)}$.

Ontological note (location in DO). Appendix A does not introduce an operator formalism for position. "Location" means the $S^{(4D)}$ occupancy set(s) and associated scalar energy samples $e_\alpha(t) = |\Phi_\alpha^{(4D)}(t)|^2$. For an N-body state, $\mathcal{O}_{N\text{-body}}(t) = \bigcup_{n=1}^{N} \mathcal{O}_n(t)$ with $\sum_{n,\alpha} e_{n,\alpha}(t) = E_{QS}$ (Sec. 1.1.6).

Appendix C.9.2. 2.7.2. Collapse and Localization at Detection

A Physical Interaction (PI, Sec. 3.1) in $S^{(4D)}$ with a detector at site γ_{det} (a Bell Sphere within the pre-collapse support where $e_{\gamma_{\text{det}}}(t_{\text{pre}}) > 0$) at time t_{coll} triggers collapse.

Collapse process and outcome.

- $S^{(PD)}$ *action.* The PI triggers the Collapse Operator C (Sec. 1.1.8) acting on the $S^{(PD)}$ Bell Energy Point, selecting an admissible connected outcome subset $O' \in \text{Conn}(\mathcal{C})$ contained in the pre-collapse support and containing γ_{det}.
- $S^{(4D)}$ *mirroring.* Via the Bell Identity g (Sec. 2.1), this $S^{(PD)}$ selection is mirrored in $S^{(4D)}$ on the same tick t_{coll}. The Bell Field localizes so that $\text{Supp}\,\Phi^{(4D)}(t_{\text{coll}}) = O'$ and energy outside O' becomes zero. Post-collapse energy is preserved by the DO rescaling rule of Sec. 1.1.8.

Appendix C.9.3. 2.7.3 Interference Pattern Emergence from Probabilistic Localization

The interference pattern arises statistically from many independent collapses. Immediately before a detection PI, the local energy on a screen site γ is

$$e_\gamma(t_{\text{pre}}) = \left|\Phi_{A,\gamma}^{(4D)}(t_{\text{pre}}) + \Phi_{B,\gamma}^{(4D)}(t_{\text{pre}})\right|^2.$$

Probability of localization (DO). Let $O(t_{\text{pre}})$ be the pre-collapse support and $E_{QS} = \sum_{\alpha \in O(t_{\text{pre}})} e_\alpha(t_{\text{pre}})$ the conserved total hosted (non-CIE) intrinsic energy. For any admissible connected outcome $O' \subseteq O(t_{\text{pre}})$,

$$P(O') = \frac{\sum_{\alpha \in O'} e_\alpha(t_{\text{pre}})}{\sum_{\alpha \in O(t_{\text{pre}})} e_\alpha(t_{\text{pre}})}.$$

 doi:10.20944/preprints202405.2028.v8

For localization to a single site γ_{det} (singleton outcome $O' = \{\gamma_{\text{det}}\}$, which is connected),

$$P(\gamma_{\text{det}}) \;=\; \frac{e_{\gamma_{\text{det}}}(t_{\text{pre}})}{E_{QS}} \;=\; \frac{\left|\Phi^{(4D)}_{A,\gamma_{\text{det}}}(t_{\text{pre}}) + \Phi^{(4D)}_{B,\gamma_{\text{det}}}(t_{\text{pre}})\right|^2}{E_{QS}}.$$

Additivity holds over any pairwise-disjoint family of admissible outcomes whose union is $O(t_{\text{pre}})$: the corresponding probabilities sum to one.

Appendix C.10. 2.8. Which-Way Experiment in the DO Model

The DO model explains which-way experiments as collapse triggered by a PI near the slits, which confines the $S^{(4D)}$ Bell Field (and thus its Bell Energy Field) to a single connected subset associated with one slit and therefore removes overlap-energy cross terms at the detection screen.

Appendix C.10.1. 2.8.1. Initial State and Which-Way Detection Setup

A quantum state (total hosted (non-CIE) intrinsic energy E_{QS}) is directed toward a two-slit barrier (slits A, B), with a which-way detector (WWD) arranged to produce a PI near the slits.

Pre-interaction state representation.

- $S^{(4D)}$ *(Bell Field / Bell Energy Field):* A state descriptor $\Phi^{(4D)}(t)$ on the occupied Bell Spheres, with Bell Energy Field samples $e_\alpha(t) = |\Phi^{(4D)}_\alpha(t)|^2$ and $\sum_\alpha e_\alpha(t) = E_{QS}$.
- $S^{(PD)}$ *(Bell Point / Bell Energy Point):* A single Bell Point for the state; its quantum energy component is a single Bell Energy Point $\phi^{(PD)}$ representing E_{QS}.
- *Bell Identity g:* $\Phi^{(4D)}(t) \overset{g}{\longleftrightarrow} \phi^{(PD)}$.

Appendix C.10.2. 2.8.2. Which-Way Detection as Collapse Trigger

A PI (Sec. 3.1) between the state and the WWD in $S^{(4D)}$ near the slits at time t_{coll} triggers collapse.

Collapse process and outcome.

- $S^{(PD)}$ *action.* The PI triggers C (Sec. 1.1.8) acting on $\phi^{(PD)}$, selecting an admissible connected outcome $O' \in \text{Conn}(\mathcal{C})$ contained within the pre-collapse support and confined to the neighborhood associated with one slit $X \in \{A, B\}$.
- $S^{(4D)}$ *mirroring.* Via g, the post-collapse $S^{(PD)}$ outcome is mirrored to $S^{(4D)}$ on the same tick, yielding a localized $S^{(4D)}$ Bell Field with state descriptor $\Phi^{(4D)}_X(t_{\text{coll}})$ with support confined to the corresponding connected subset. Post-collapse energy is preserved by the DO rescaling rule of Sec. 1.1.8.

Appendix C.10.3. 2.8.3. Absence of Interference Pattern

Post-collapse, the $S^{(4D)}$ Bell Field has support only on the single connected subset associated with slit X. Subsequent propagation uses only that single component on its domain. Therefore, for any screen site γ,

$$e_\gamma(t) \;=\; \left|\Phi^{(4D)}_{X,\gamma}(t)\right|^2, \qquad X \in \{A, B\},$$

and there is no A–B overlap contribution to the screen-energy distribution because only one slit-associated branch exists post-collapse. The observed distribution is the single-slit pattern associated with X, with no two-slit interference fringes.

Appendix C.11. 2.9. Continuum Correspondence

This section is an optional correspondence module. All KG-like, second-difference, Laplacian-symbol, and SR-dispersion statements in this section are reader-facing correspondence only and do not define the fundamental UEE tick map of Sec. 2.3. It applies only when (i) the occupied energy distribution becomes dense on the DS network and (ii) the chosen kernel realization lies in

 doi:10.20944/preprints202405.2028.v8

the continuum-symbol subclass $\mathcal{W}_\Delta \subset \mathcal{W}$ (Sec. 1.1.9). No statement in this section modifies any DO operator, kernel, update, or collapse rule; it provides reader-facing long-wavelength correspondences only.

$$(\Delta_g \Phi)_a \;=\; \sum_{b\in\mathcal{N}(a)} w_{ab}\,(\Phi_b - \Phi_a) \;\Longrightarrow\; \nabla^2\Phi(\mathbf{x}_a) \quad (\lambda \gg V_0^{1/3}),$$

where $\mathcal{N}(a)$ and w_{ab} are defined in Sec. 1.1.7, and the implication is understood in the long-wavelength, dense-occupancy sense on graphs with approximately isotropic shells.

Moreover, under nearly uniform link weights and on small freely falling patches, the spectrum of the effective operator $\hat{K}_{\text{eff}}$ exhibits a light-cone structure consistent with the invariant speed c at long wavelengths and reproduces Special Relativistic dispersion in the macroscopic correspondence limit. All continuum and relativistic correspondence features follow solely from the fixed link-weight network acting on the evolving state, without invoking any fixed background metric.

Appendix C.11.1. 2.9.1. Continuum Correspondence of Operators

Assumptions. Dense occupancy and smooth link variation; $w(d) \in \mathcal{W}_\Delta$ (Sec. 1.1.9). Let

$$(\Delta_g \Phi)_a := \big(\hat{K}_{\text{grav}}\Phi\big)_a = \sum_{b\in\mathcal{N}(a)} w_{ab}\,(\Phi_b - \Phi_a), \qquad w_{ab} = w(d_{ab}),$$

with $\mathcal{N}(a)$ defined in Sec. 1.1.7.

Operator limits (long wavelength).

$$\widehat{\Delta_g}(\mathbf{k}) = -C_{\text{cont}}\,\|\mathbf{k}\|^2 + \mathrm{O}(\|\mathbf{k}\|^4).$$

Kinetic exchange (one admissible realization). For realizations in which the kinetic conductances coincide with the graph weights up to a scalar factor, $\eta_{ab} = \eta_0\, w_{ab}$, one has

$$K_{\text{kin}} = -\frac{\hbar^2}{2m}\,\Delta_g \;\Longrightarrow\; -\frac{\hbar^2}{2m}\,\nabla^2 \quad (\lambda \gg V_0^{1/3}).$$

Other admissible η_{ab} choices yield the same continuum correspondence provided their long-wavelength symbol matches $-\tilde{C}\,\|\mathbf{k}\|^2 + \mathrm{O}(\|\mathbf{k}\|^4)$ on the same graph family.

No potential-times-Φ term. No term of the form $V(\mathbf{x})\,\Phi(\mathbf{x})$ appears; neutrality of the uniform mode is preserved.

Neutral mode. $\sum_a (\Delta_g\Phi)_a = 0$ and $\sum_a (K_{\text{grav}}\Phi)_a = 0$ (uniform mode unaffected).

Appendix C.11.2. 2.9.2. Dispersion Relation and Special Relativity Correspondence

Fourier-mode analysis (uniform background; correspondence probe only). Take the analytic probe mode $\Phi_a(t_K) \sim e^{i(k\cdot x_a - \omega K\Delta t)}$, where $K \in \mathbb{Z}$ is the tick index with $t_K = t_0 + K\Delta t$, k is the discrete spatial-frequency label, and x_a denotes the index-chart coordinate used for hop counting on the DS network (not a background metric coordinate). In the one-matching-per-tick realization of Sec. 2.3, a full schedule sweep U_{cycle} spans R ticks and therefore a time interval $R\Delta t$; the Δt appearing below remains the single-tick time step.

$$\frac{4}{c^2}\sin^2\!\Big(\frac{\omega\Delta t}{2}\Big) = \lambda_{\text{eff}}(\mathbf{k})\,(\Delta t)^2 + \Big(\tfrac{mc}{\hbar}\Big)^2 (\Delta t)^2,$$

where $\lambda_{\text{eff}}(\mathbf{k}) := -\widehat{\hat{K}_{\text{eff}}}(\mathbf{k})$.

Long-wavelength limit.

$$\omega^2 = c^2\|\mathbf{k}\|^2 + \Big(\tfrac{mc^2}{\hbar}\Big)^2 \;+\; \mathrm{O}(\|\mathbf{k}\|^4,\, \Delta t^2\|\mathbf{k}\|^4),$$

 doi:10.20944/preprints202405.2028.v8

recovering the SR dispersion. Group velocity satisfies

$$v_g(\mathbf{k}) = \partial_{\|\mathbf{k}\|}\omega \;\le\; c = \frac{\Delta x_{\text{hop}}}{\Delta t}, \qquad v_g \to c \;\; (\|\mathbf{k}\|\Delta x_{\text{hop}} \to 0).$$

(Maxwell/Dirac blocks share the same c and yield the same bound.)

Appendix C.11.3. 2.9.3. Bell Identity and Energy Conservation

Bell sum (pre-collapse).

$$\sum_{\alpha\in\mathcal{O}(t)} |\Phi_\alpha^{(4D)}(t)|^2 \;=\; E_{QS} \quad \text{(constant between collapses)}.$$

$\mathcal{E}$-functional (correspondence/audit). Let $Y := (\Phi, \Pi)$ with $\Pi := \Delta_t\Phi/c$. Assumption (correspondence module): there exists a positive definite functional

$$\mathcal{E}(Y) = \frac{1}{2}\sum_a |\Pi_a|^2 + \frac{1}{2}\sum_{a<b} \mathcal{V}_{ab}(\Phi_a, \Phi_b; w_{ab}) + \frac{1}{2}\left(\tfrac{mc}{\hbar}\right)^2 \sum_a |\Phi_a|^2,$$

with pair forces $F_{a\leftarrow b} = -\partial\mathcal{V}_{ab}/\partial\Phi_a$ satisfying $F_{a\leftarrow b} = -F_{b\leftarrow a}$. In the correspondence/audit setting of Sec. 2.9.4, the discrete-gradient scheme preserves $\mathcal{E}(Y)$ exactly:

$$\Delta_t\mathcal{E}(Y) = 0.$$

This $\mathcal{E}(Y)$ is a read-only correspondence/audit functional. The conserved hosted (non-CIE) intrinsic energy used by the Bell Identity and the collapse rule is the Bell sum E_{QS}, conserved between collapses by the UEE tick map (Sec. 2.3) and preserved at collapse by the Sec. 1.1.8 rescaling rule.

Appendix C.11.4. 2.9.4. Well-Posedness, Causality, and Stability

This subsection is part of the optional correspondence module in Sec. 2.9. The fundamental between-collapse evolution law remains the Bell-sum–conserving UEE tick map in Sec. 2.3.

Domain of dependence (realization locality). For a finite-range exchange realization with cutoff R_w in hop distance (Sec. 1.1.7),

$$\operatorname{supp}\Phi(\cdot, t_0 + N\Delta t) \subset \{\, b :\; d(b, \operatorname{supp}\Phi(\cdot, t_0)) \le NR_w \,\}.$$

This bound states which sites can enter the update after N ticks for that realization. It does not redefine the invariant causal bookkeeping convention $c = 1$ hop/tick (Sec. 6.4).

$\mathcal{E}$-preserving update (existence/uniqueness). The discrete-gradient scheme

$$\frac{Y^{n+1} - Y^n}{\Delta t} = \mathbf{J}\,\bar{\nabla}\mathcal{E}\big(Y^n, Y^{n+1}\big), \qquad \mathbf{J} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix},$$

admits a solution for any $\Delta t > 0$; if $\nabla\mathcal{E}$ is Lipschitz on the energy sublevel set of Y^0, the solution is unique. It preserves $\mathcal{E}$ exactly: $\mathcal{E}(Y^{n+1}) = \mathcal{E}(Y^n)$.

Stability and continuity. Energy preservation implies uniform boundedness of Y^n; the one-step map is Lipschitz on sublevel sets, giving continuous dependence on initial data.

Local SR reduction. On small, freely falling patches with nearly uniform w_{ab}, the UEE reduces to the SR dispersion of Sec. 2.9.2 for non-gravitational dynamics; tidal corrections enter via gradients of w.

 doi:10.20944/preprints202405.2028.v8

Appendix C.11.5. 2.9.5. κ-Amplitude Mapping (Operational Calibration)

Status (post-processing only). The κ-map is an operational bridge used only for observational comparison (Appendix B, Validation I). It is not referenced by any operator, kernel, update, or collapse rule, and it has no feedback path into DO dynamics.

The discrete pipeline yields dimensionless bandpowers $C_\ell^{\rm sim}$. Comparison to observational means $C_\ell^{\rm obs}$ uses a single global factor on *power*. Define $\lambda_\kappa \equiv \kappa^2 \geq 0$ and minimize, over a designated multipole set L,

$$\lambda_\kappa^\star \;=\; \arg\min_{\lambda_\kappa \geq 0} \sum_{\ell\in L} \frac{\left(\lambda_\kappa\, C_\ell^{\rm sim} - C_\ell^{\rm obs}\right)^2}{\sigma_\ell^2} \;=\; \frac{\displaystyle\sum_{\ell\in L} \frac{C_\ell^{\rm sim}\, C_\ell^{\rm obs}}{\sigma_\ell^2}}{\displaystyle\sum_{\ell\in L} \frac{\left(C_\ell^{\rm sim}\right)^2}{\sigma_\ell^2}},$$

and report $\kappa = \sqrt{\lambda_\kappa^\star}$. This calibration fixes units only; spectral shape and correlation structure arise from the non-perturbative dynamics. For display we use $D_\ell := \frac{\ell(\ell+1)}{2\pi} C_\ell$.

Appendix C.11.6. 2.10. Ontological Assumptions and Dependency Map

This subsection records the minimal ontological/mathematical assumptions used in Appendix A and maps them to derived results, to make explicit that no auxiliary postulates or free parameters are introduced.

Foundational Assumptions (FA).

FA1 **Discrete Spheres (DS):** a network $G = (\mathcal{V}, \mathcal{E})$ with a graph geodesic distance d_{ab} and induced shells/neighborhoods; for each epoch t, the discrete 4D spacetime is a finite induced subgraph of this network.

FA2 **Radial symmetry of link weights:** $w_{ab} = w_{ba}$ and w_{ab} depends only on d_{ab} (Sec. 1.1.7).

FA3 **Admissible kernel class:** $w(d) \in \mathcal{W}$ (Secs. 1.1.7–1.1.9). When long-wavelength Laplacian-symbol correspondence is invoked (Secs. 2.9.1–2.9.2), assume $w(d) \in \mathcal{W}_\Delta \subset \mathcal{W}$ (Sec. 1.1.9).

FA4 **Hosted (non-CIE) intrinsic energy conservation between collapses:** hosted (non-CIE) intrinsic energy is conserved during $S^{(4D)}$ evolution (Sec. 1.1.5 and Sec. 2.9.3).

FA5 **Isotropy of the interaction manifold (operational):** uniform distribution on $S^{\nu-1}$ for latent directions used in correlation evaluation (Sec. 2.6.3).

FA6 **Locality of collapse outcomes:** collapse selects a connected subset under the fixed connectivity $\mathcal{C}$ (Sec. 1.1.8).

Derived Results (DR) and Dependencies.

- **DR1 — Collapse operator admissibility and energy-fraction probabilities** (Sec. 1.1.8; Sec. 3.2): depends on FA1, FA4, FA6.
- **DR2 — Correlation rule and dimensional factor** ν (Sec. 2.6.3): depends on FA1, FA5.
- **DR3 — Discrete stability / well-posedness of the UEE tick map** (Sec. 2.4.3): depends on FA1, FA2, FA3.
- **DR4 — Continuum-symbol correspondence and SR dispersion in the long-wavelength limit** (Secs. 2.9.1–2.9.2): depends on FA1, FA2, FA3 (with $w \in \mathcal{W}_\Delta$), and the UEE tick-map framework of Secs. 2.3–2.4.
- **DR5 — Kernel admissibility statements** (Sec. 1.1.9): depends on FA2 and the class definitions in FA3.

Consistency. DR1–DR5 require only FA1–FA6 and the optional continuum-symbol restriction $w \in \mathcal{W}_\Delta$ when Secs. 2.9.1–2.9.2 are invoked; no additional tunable parameters are introduced by this dependency map.

 doi:10.20944/preprints202405.2028.v8

Appendix C.12. 3.1. Physical Triggers for Collapse

Collapse of a quantum state's Bell Energy Point in $S^{(PD)}$ is triggered by a **Physical Interaction (PI)** in $S^{(4D)}$ (Sec. 1.1.8). The Bell Identity g (Sec. 2.1) provides bijective mirroring between the state's dual representations in $S^{(PD)}$ and $S^{(4D)}$ across this collapse event. Appendix A does not identify the precise microscopic PI that induces collapse; it records the admissible trigger framework, the admissible analyzer-coupled probability reweighting, and the constraints on admissible outcomes.

PI categories (conventional labels). In the DO model, PIs in $S^{(4D)}$ arise from one or more of the three traditionally labeled non-gravitational interaction classes:

$$\mathcal{T} = \{\text{Electromagnetic, Strong, Weak}\}.$$

These labels denote classes of localized physical couplings in $S^{(4D)}$. Collapse triggering is independent of human consciousness.

PI structure (framework status). A PI is a localized event in $S^{(4D)}$ (localized in both space and time tick) that couples the quantum state to a local interaction context (environment, apparatus, barrier, detector). PI occurrence depends on macroscopic conditions (including temperature and spatial placement), and macroscopic devices influence PI timing and location by controlling local interaction conditions. This device dependence is operational context and does not modify any DO evolution law.

Remark (collapse fragility in ordinary environments). In ordinary environments, Physical Interactions occur frequently across the occupied Bell Field $O(t) \subset S^{(4D)}$. Each PI acts as a Physical Trigger, inducing collapse in $S^{(PD)}$; the resulting outcome is mirrored into $S^{(4D)}$ by the Bell Identity g as a localized update of the Bell Energy Field on $O(t)$. The practical "fragility" of coherent quantum states in 4D spacetime follows from the density of PIs acting on $O(t)$, not from an additional stability mechanism or internal dynamics in $S^{(PD)}$.

Trigger-to-collapse map (constraint). A PI triggers one collapse event in $S^{(PD)}$, mirrored to $S^{(4D)}$ on the same tick. The post-collapse support in $S^{(4D)}$ is confined to a pre-existing connected subset of the pre-collapse support (Sec. 1.1.8):

$$\operatorname{Supp}\Phi^{(4D)}(t_{\text{coll}}) \subseteq \operatorname{Supp}\Phi^{(4D)}(t_{\text{pre}}), \qquad \operatorname{Supp}\Phi^{(4D)}(t_{\text{coll}}) \in \operatorname{Conn}(\mathcal{C}).$$

Scope note (hybrid energy). Sec. 1.1.7 defines a hybrid per-sphere hosted-energy sample $e_b(t)$ for collapse bookkeeping and read-only auditing (quantum branch $e_b = |\Phi_b|^2$, classical branch $e_b = E_{c,b,\text{classical}}$). Sections 3.1–3.2 describe collapse of a *quantum state* and use the quantum-state hosted samples $e_\alpha(t_{\text{pre}}) = |\Phi_\alpha(t_{\text{pre}})|^2$ on that state's pre-collapse support.

Operational note. Sec. 3.1.1 defines an analyzer-coupled reweighting used only to evaluate collapse probabilities under an analyzer PI. This reweighting has no feedback path into any operator, kernel, update, or collapse rule outside probability evaluation, and it does not set the CHSH correlation magnitude derived in Secs. 2.6.3–2.6.4.

Appendix C.12.1. 3.1.1. Stern–Gerlach Analyzer PI (axis setting)

A Stern–Gerlach analyzer PI is a localized Physical Interaction (PI) in $S^{(4D)}$ that triggers one collapse event in $S^{(PD)}$, mirrored into $S^{(4D)}$ on the same tick by the Bell Identity g. This subsection defines an operational, read-only reweighting used only to evaluate collapse probabilities for a chosen analyzer axis setting.

Pre-collapse support and hosted samples. Let the pre-collapse Bell Field be $\mathcal{O}(t_{\text{pre}}) \subset S^{(4D)}$, with hosted (non-CIE) samples

$$e_a(t_{\text{pre}}) = |\Phi_a^{(4D)}(t_{\text{pre}})|^2, \qquad a \in \mathcal{O}(t_{\text{pre}}),$$

 doi:10.20944/preprints202405.2028.v8

and total hosted (non-CIE) intrinsic energy

$$E_{QS} := \sum_{\beta \in \mathcal{O}(t_{\text{pre}})} e_\beta(t_{\text{pre}}).$$

Analyzer axis and reweighted samples. Let $\mathbf{A}$ be the unit analyzer axis. Let $s_a(\mathbf{A}) \in [-1, 1]$ denote the analyzer-alignment observable used for probability evaluation under this PI at Bell Sphere a. Define the energy-weighted mean alignment

$$\bar{s}(\mathbf{A}) := \frac{1}{E_{QS}} \sum_{\beta \in \mathcal{O}(t_{\text{pre}})} e_\beta(t_{\text{pre}})\, s_\beta(\mathbf{A}).$$

Define the analyzer-reweighted samples

$$e_a(\mathbf{A}) \;=\; e_a(t_{\text{pre}}) \Big[1 + \lambda_{\text{PI}} \big(s_a(\mathbf{A}) - \bar{s}(\mathbf{A})\big)\Big], \tag{A34}$$

with $\lambda_{\text{PI}} \in \mathbb{R}$ chosen so that $e_a(\mathbf{A}) \geq 0$ for all $a \in \mathcal{O}(t_{\text{pre}})$.

Energy-sum preservation (bookkeeping). The reweighting preserves the total hosted (non-CIE) intrinsic energy used by the DO Probability Rule:

$$\sum_{a \in \mathcal{O}(t_{\text{pre}})} e_a(\mathbf{A}) = \sum_{a \in \mathcal{O}(t_{\text{pre}})} e_a(t_{\text{pre}}) = E_{QS}. \tag{A35}$$

Use. Probability evaluation under an analyzer PI uses $\{e_a(\mathbf{A})\}$ in Sec. 3.2 in place of $\{e_a(t_{\text{pre}})\}$. No quantity in this subsection is referenced by any evolution map, kernel, operator, or collapse rule outside probability evaluation.

Appendix C.13. 3.2. Collapse Probabilities from Energy Distribution

Collapse probabilities derive from the physical distribution of the quantum state's hosted (non-CIE) intrinsic energy across its pre-collapse Bell Field in $S^{(4D)}$. Let the pre-collapse support be $\mathcal{O}(t_{\text{pre}}) \subset S^{(4D)}$ with samples

$$e_\alpha(t_{\text{pre}}) = |\Phi_\alpha^{(4D)}(t_{\text{pre}})|^2, \qquad E_{QS} := \sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(t_{\text{pre}}).$$

Admissible outcomes. An admissible outcome is any subset $O' \subseteq \mathcal{O}(t_{\text{pre}})$ that is connected under the fixed link set $\mathcal{C} = \{(a, b) : w_{ab} > 0\}$ (Sec. 1.1.7):

$$O' \in \text{Conn}(\mathcal{C}).$$

Collapse selects only from pre-existing Bell Spheres in $\mathcal{O}(t_{\text{pre}})$ and is mirrored into $S^{(4D)}$ on the same tick by g (Sec. 1.1.8; Sec. 2.1).

DO Probability Rule (energy fraction). For any admissible $O' \subseteq \mathcal{O}(t_{\text{pre}})$,

$$P(O') \;=\; \frac{\sum_{\alpha \in O'} e_\alpha(t_{\text{pre}})}{\sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(t_{\text{pre}})} \;=\; \frac{\sum_{\alpha \in O'} e_\alpha(t_{\text{pre}})}{E_{QS}}. \tag{A36}$$

Analyzer PI. Under an analyzer PI (Sec. 3.1.1), probability evaluation uses the reweighted samples $e_\alpha(\mathbf{A})$ from Eq. (A34) in place of $e_\alpha(t_{\text{pre}})$. The normalization $\sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(\mathbf{A}) = E_{QS}$ holds by Eq. (A35).

 doi:10.20944/preprints202405.2028.v8

Energy preservation at collapse. Post-collapse hosted (non-CIE) intrinsic energy equals E_{QS} by the Sec. 1.1.8 rescaling rule. This preservation introduces no coefficient and has no feedback path into any operator, kernel, update, or probability evaluation.

Additivity on disjoint admissible families. Let $\{O'_j\} \subset \mathrm{Conn}(\mathcal{C})$ be pairwise disjoint and satisfy $\bigsqcup_j O'_j = \mathcal{O}(t_{\mathrm{pre}})$. Then

$$\sum_j P(O'_j) = 1.$$

Proposition (Remote-setting invariance / no-signal; marginal form).

Let the pre-collapse support decompose as a disjoint union

$$\mathcal{O}(t_{\mathrm{pre}}) \;=\; O_A \cup O_B, \qquad O_A \cap O_B = \varnothing,$$

with hosted samples $\{e_\alpha(t_{\mathrm{pre}})\}_{\alpha \in \mathcal{O}(t_{\mathrm{pre}})}$ and total $E_{QS} = \sum_{\alpha \in \mathcal{O}(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}})$. Fix a pairwise-disjoint family $\mathcal{P}_B = \{B_\ell\} \subset \mathrm{Conn}(\mathcal{C})$ whose disjoint union equals O_B:

$$\bigsqcup_\ell B_\ell \;=\; O_B.$$

Let $\mathcal{P}_A^{(s)} = \{A_k^{(s)}\} \subset \mathrm{Conn}(\mathcal{C})$ be a pairwise-disjoint family whose disjoint union equals O_A:

$$\bigsqcup_k A_k^{(s)} \;=\; O_A,$$

where s denotes the local setting choice that selects the family used at A. Define $P_s(B_\ell)$ as the collapse probability of the macro-event that the post-collapse $S^{(4D)}$ support lies in $B_\ell \subseteq O_B$, evaluated by additivity of Eq. (A36) over a pairwise-disjoint admissible family spanning O_B. Then Bob's marginal probability for event B_ℓ is

$$P_s(B_\ell) = \frac{\sum_{\alpha \in B_\ell} e_\alpha(t_{\mathrm{pre}})}{E_{QS}}, \tag{A37}$$

so $P_s(B_\ell)$ is independent of s.

Proof. By Eq. (A36) and additivity on the disjoint family $\mathcal{P}_B$ spanning O_B, the marginal assigned to B_ℓ equals the hosted energy fraction inside B_ℓ:

$$P_s(B_\ell) = \frac{\sum_{\alpha \in B_\ell} e_\alpha(t_{\mathrm{pre}})}{\sum_{\alpha \in \mathcal{O}(t_{\mathrm{pre}})} e_\alpha(t_{\mathrm{pre}})} = \frac{\sum_{\alpha \in B_\ell} e_\alpha(t_{\mathrm{pre}})}{E_{QS}}.$$

The right-hand side contains no dependence on $\mathcal{P}_A^{(s)}$, so it is independent of s. □

Remark. This statement is marginal invariance: remote settings refine the admissible family on O_A and do not change the pre-collapse hosted energy share inside any fixed $B_\ell \subseteq O_B$.

Appendix C.14. 3.3. Tunneling as Collapse Localization

Tunneling is represented as collapse localization into a region separated by a barrier from the region reached by ordinary link-propagation under the fixed divider structure.

Barrier realization (graph disconnection). A barrier is implemented by setting $w_{ab} = 0$ for all cross-divider pairs (a, b), splitting the fixed link set $\mathcal{C}$ into disjoint connected components (Sec. 1.1.7). Between collapses, the UEE tick map acts independently on each connected component of $\mathcal{C}$.

 doi:10.20944/preprints202405.2028.v8

Pre-collapse support. Let the pre-collapse Bell Field be $\mathcal{O}(t_{\text{pre}}) \subset S^{(4D)}$ with hosted samples $e_\alpha(t_{\text{pre}}) = |\Phi_\alpha^{(4D)}(t_{\text{pre}})|^2$ and total hosted (non-CIE) intrinsic energy

$$E_{QS} := \sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(t_{\text{pre}}).$$

Let $O_{\text{barrier}} \subseteq \mathcal{O}(t_{\text{pre}})$ denote the portion of the pre-collapse support contained in a barrier-separated connected component on the far side of the divider.

Tunneling probability (DO). Collapse occurs only in $S^{(PD)}$ and is mirrored into $S^{(4D)}$ on the same tick by g (Sec. 1.1.8; Sec. 2.1). The probability that collapse localizes to O_{barrier} equals the pre-collapse hosted energy fraction in that region:

$$P(O_{\text{barrier}}) \;=\; \frac{\sum_{\alpha \in O_{\text{barrier}}} |\Phi_\alpha^{(4D)}(t_{\text{pre}})|^2}{E_{QS}} \;=\; \frac{\sum_{\alpha \in O_{\text{barrier}}} e_\alpha(t_{\text{pre}})}{\sum_{\alpha \in \mathcal{O}(t_{\text{pre}})} e_\alpha(t_{\text{pre}})}. \tag{A38}$$

Post-collapse support is confined to the selected connected subset and post-collapse hosted energy is preserved by the Sec. 1.1.8 rescaling rule.

Appendix D. 4. Bell's Theorem, Locality, and Collapse Mirroring

The DO model's ontology—dual state representation ($S^{(4D)}$, $S^{(PD)}$), the Bell Identity (Secs. 1.1.5 and 2.1), and $S^{(PD)}$-based collapse (Sec. 1.1.8)—provides the conceptual basis for addressing Bell-type locality questions.

This appendix records the mathematical and rule-level structure used for those implications (notably the collapse rule, the energy-fraction probability rule, and the CHSH bound derived in Secs. 2.6.3–2.6.5 and 3.2). Detailed interpretive discussion beyond those formal statements is in the main text.

Appendix E. 5. Quantum Path Irreversibility and the Arrow of Time

The DO model contains two distinct processes: (i) deterministic evolution of Bell Energy Fields in $S^{(4D)}$ under the UEE tick map (Secs. 2.3–2.4), and (ii) collapse acting in $S^{(PD)}$ under the collapse operator C (Sec. 1.1.8), mirrored into $S^{(4D)}$ on the same tick by the Bell Identity g (Sec. 2.1). Irreversibility arises from the collapse process and its mirrored support restriction in $S^{(4D)}$.

Appendix E.1. 5.1. State Configuration Pre- and Post-Collapse (Bohm/EPR)

Consider the two-spin-$\frac{1}{2}$ singlet system of Sec. 2.6.

I. Pre-collapse state (unified).

- $S^{(PD)}$: a single unified N-body Bell Energy Point $\Phi_{AB}^{(PD)}$ encoding total hosted (non-CIE) intrinsic energy and the singlet anti-correlation constraint.
- $S^{(4D)}$: two Bell Energy Fields $\Phi_A^{(4D)}(t)$ and $\Phi_B^{(4D)}(t)$, supported on $\mathcal{O}_A(t)$ and $\mathcal{O}_B(t)$, evolving between collapses under the UEE tick map (Sec. 2.3).
- *Mirroring:* the Bell Identity g provides a bijective ontological mirroring between the unified $S^{(PD)}$ representation and the corresponding $S^{(4D)}$ Bell Fields (Sec. 2.1).

II. Post-collapse state (factorized).

- A Physical Interaction (PI) in $S^{(4D)}$ triggers collapse (Sec. 3.1). Collapse occurs only in $S^{(PD)}$ and is mirrored to $S^{(4D)}$ on the same tick by g (Sec. 2.1).
- $S^{(PD)}$: the collapse operator C replaces the unified BEP by a factorized pair of single-body BEPs,

$$C\left(\Phi_{AB}^{(PD)}\right) \;\longrightarrow\; (\phi_A^{(PD)}, \phi_B^{(PD)}),$$

 doi:10.20944/preprints202405.2028.v8

encoding definite anti-correlated outcomes consistent with the singlet constraint.
- $S^{(4D)}$: by g, this collapse is mirrored as a restriction of each Bell Energy Field's support to an admissible connected subset of its pre-collapse support (Sec. 1.1.8), with total hosted (non-CIE) intrinsic energy preserved by the Sec. 1.1.8 rescaling rule.

Appendix E.2. 5.2. Intrinsic Irreversibility of $S^{(PD)}$ Collapse

Post-collapse (at t_{coll}), the localized $S^{(4D)}$ Bell Energy Fields evolve between collapses under the UEE tick map (Secs. 2.3–2.4). The arrow of time in DO is attributed to the intrinsic non-reversibility of the $S^{(PD)}$ collapse-only event together with its mirrored consequence in $S^{(4D)}$ on the same tick.

Non-invertibility (PD). The DO ontology defines collapse as a one-way map in $S^{(PD)}$:

$$C:\ \Phi_{AB}^{(PD)} \longrightarrow (\phi_A^{(PD)}, \phi_B^{(PD)}).$$

An inverse map C^{-1} that restores the unique pre-collapse unified N-body Bell Energy Point from the factorized post-collapse configuration is not defined in the DO ontology.

One-way support restriction (mirrored in $S^{(4D)}$). The mirrored consequence of collapse in $S^{(4D)}$ is an irreversible restriction of occupied support to a pre-existing connected subset:

$$\operatorname{Supp}\Phi^{(4D)}(t_{\text{coll}}) \subseteq \operatorname{Supp}\Phi^{(4D)}(t_{\text{pre}}), \qquad \operatorname{Supp}\Phi^{(4D)}(t_{\text{coll}}) \in \operatorname{Conn}(\mathcal{C}),$$

with the selection performed by C in $S^{(PD)}$ and mirrored by g on the same tick (Sec. 1.1.8 and Sec. 2.1).

Label preservation vs. structural change. Conserved intrinsic labels (including total hosted (non-CIE) intrinsic energy) remain preserved through collapse by construction (Sec. 1.1.8). The unified correlational structure of the pre-collapse N-body BEP is replaced by the post-collapse factorized configuration, and no admissible operation in $S^{(PD)}$ reverses that replacement.

Arrow-of-time statement. Between collapses, evolution proceeds under the UEE in $S^{(4D)}$. Collapse events introduce an intrinsic directionality because collapse in $S^{(PD)}$ is non-invertible and is mirrored in $S^{(4D)}$ as an irreversible connected-subset restriction of occupied support. Consequently, for processes involving collapse events, the DO arrow-of-time is identified with the non-invertibility of C in $S^{(PD)}$ together with its mirrored connected-subset restriction in $S^{(4D)}$.

Appendix E.3. 5.3. 4D Macrostates and Classical Labels (read-only)

Under the DO, "classical" descriptors are not additional ontological objects and do not add a second dynamics. They are read-only labels assigned to relational configurations and histories in $S^{(4D)}$.

Definition (macrostate predicate). A macrostate predicate is any read-only map

$$\mathcal{M}:\ \left(\{\Phi^{(4D)}(t)\}_{t\le T},\ \mathcal{P}_{\le T}\right) \longrightarrow \mathcal{L},$$

where $\mathcal{P}_{\le T}$ denotes the PI/collapse history up to tick T (Sec. 3.1) and $\mathcal{L}$ is a finite label set chosen by the analyst for description. The DO model does not define a privileged $\mathcal{M}$ or $\mathcal{L}$.

No feedback (fence). $\mathcal{M}$ is a read-only classifier. It is not referenced by any update, operator, kernel, or collapse rule and has no feedback path into Φ, $w(d)$, $\hat{K}_{\text{eff}}$, or C (Sec. 1.1.10 G5; Sec. 2.9.5).
Terminology (particle labels as macrostates). The label set $\mathcal{L}$ may include standard particle-physics names (e.g., electron, muon, neutrino, photon) as classical macrostate labels in $S^{(4D)}$. Such labels summarize repeatable 4D interaction signatures and conserved-quantity patterns of DO states. They are read-only and have no feedback path into Φ, any operator, any kernel, any schedule, PI triggering, or collapse.

 doi:10.20944/preprints202405.2028.v8

Interpretation. Terms such as "alive/dead" name elements of $\mathcal{L}$ that summarize a macroscopic 4D relational history. In DO, definiteness of such labels corresponds to a determinate $S^{(4D)}$ history produced by deterministic evolution between collapses and PD-only collapse events mirrored into $S^{(4D)}$ on the same tick by g.

Appendix F. 6. Discreteness, Regularization, and Background Independence

Appendix F.1. 6.1. Singularity Resolution

General Relativistic singularities are avoided in the DO model because $S^{(4D)}$ is composed of Discrete Spheres (DS) with an irreducible minimum volume V_0. In SI units the DS volume is fixed ontologically to the Planck volume,

$$V_0 \equiv V_{\mathrm{Pl}} = 4.22 \times 10^{-105}\,\mathrm{m}^3. \tag{A39}$$

Internal units set $V_0 = 1$; SI reporting restores $V_0 = V_{\mathrm{Pl}}$ and therefore fixes the hop-length scale $\Delta x_{\mathrm{hop}} = V_0^{1/3}$.

DS energy split (CIE vs. hosted energy). Each DS carries a constant intrinsic energy density (CIE) that is uniform across DS and is recorded in $S^{(4D)}$ as the uniform density ρ_Λ in the cosmology bookkeeping (Sec. 7.3–Sec. 7.4). This CIE is excluded from the gravitational source term $e_b(t)$ (Sec. 2.2) and produces no local relational curvature response. Distinct from CIE, DS in $S^{(4D)}$ host non-CIE state content (matter/radiation/field energy) only on occupied Bell Spheres.

Hosted-energy capacity (ontological bound; notation aligned to the DO paper). Let $E_{\mathrm{DS,max}}$ denote the DS intrinsic maximum total energy capacity per sphere (DO paper notation). The per-sphere CIE contribution in $S^{(4D)}$ is

$$E_\Lambda := \rho_\Lambda V_0.$$

Define the maximum hosted (non-CIE) energy per DS by

$$E_{\mathrm{host,max}} := E_{\mathrm{DS,max}} - E_\Lambda, \qquad E_{\mathrm{host,max}} < \infty.$$

The corresponding maximum hosted energy density is finite:

$$\rho_{\mathrm{max}}^{(\mathrm{host})} = \frac{E_{\mathrm{host,max}}}{V_0} < \infty. \tag{A40}$$

This bound replaces continuum singular behavior by extreme but finite hosted densities under localization processes in $S^{(4D)}$, while DS CIE remains uniform and invariant.

Appendix F.2. 6.2. UV Regularization and Vacuum Energy

Ultraviolet divergences associated with continuum mode sums are removed in the DO model because DS discreteness fixes a minimum spatial scale on $S^{(4D)}$. On any finite DS realization, the discrete spectral domain is bounded, so high-frequency contributions are finite.

Vacuum-energy statement (DO). The QFT notion of pervasive zero-point vacuum energy density as a hosted background field on all of space is absent in the DO model. Energy bookkeeping in DO has two components:

1. Hosted (non-CIE) energy on occupied Bell Spheres in $S^{(4D)}$, recorded by the hybrid per-sphere source $e_b(t)$ (Sec. 1.1.7), with quantum branch $e_b = |\Phi_b|^2$ and classical hard-mass branch $e_b = E_{c,b,\mathrm{classical}}$.
2. DS CIE, recorded separately as the uniform density ρ_Λ (Sec. 7.3–Sec. 7.4).

 doi:10.20944/preprints202405.2028.v8

Unoccupied DS carry no hosted (non-CIE) state energy. Accordingly, QFT-style zero-point vacuum contributions are not part of the DO hosted-energy budget:

$$E_{\text{vac (QFT zero-point contributions)}} = 0. \tag{A41}$$

This statement removes QFT-style pervasive zero-point vacuum energy from the hosted (non-CIE) energy budget; it does not set DO field observables (E, B) or their associated hosted-energy contributions to zero. This statement leaves ρ_Λ unchanged: ρ_Λ is DS CIE and is treated separately as a uniform background density in $S^{(4D)}$.

Appendix F.3. 6.3. Background Independence in Gravitational Dynamics

DO gravitational dynamics are background-independent: no external metric geometry is assumed. All geometric and dynamical content is carried by graph-intrinsic objects (nodes, links, shells), the state descriptor $\Phi(t)$ on $S^{(4D)}$ evolving under the fixed-law UEE tick map (Sec. 2.3), and fixed, time-independent kernel data chosen once for a realization (Sec. 1.1.7). The derived samples $e_b(t) = |\Phi_b(t)|^2$ are used for collapse bookkeeping and read-only auditing (Sec. 2.2).

Index charts (no background metric). Any appearance of (x_a, y_a, z_a) in hop-distance definitions is an index chart for hop counting on the DS network. It does not define a background metric geometry.

Mechanism.

1. *Fixed relational link structure.* A radial kernel $w(d)$ and its induced link set $\mathcal{C} = \{(a, b) : w(d_{ab}) > 0\}$ are fixed and time-independent once chosen (Sec. 1.1.7).
2. *State-dependent sourcing.* State dependence enters through $\Phi(t)$ on $S^{(4D)}$ evolving under the UEE tick map (Sec. 2.3); per-sphere samples $e_b(t) = |\Phi_b(t)|^2$ are derived for collapse bookkeeping and read-only auditing (Sec. 2.2). DS CIE is excluded from $e_b(t)$ and therefore does not contribute to local relational curvature response.
3. *Read-only diagnostics.* Invariants and gates are computed post-update and have no feedback path into operators, kernels, or state updates (Appendix B).

Appendix F.4. 6.4. Time in the DO Model

The DO model treats time differently in its two domains:

- $S^{(PD)}$: timeless; no internal time coordinate is defined (Sec. 1.1.3). Collapse is the only admissible PD action (Sec. 1.1.8).
- $S^{(4D)}$: discrete coordinate time t advances in ticks of Δt; evolution between collapses follows the UEE tick map (Sec. 2.3–Sec. 2.4).

Tick index vs. observer time (read-only macrostate predicate). The discrete tick label $t \in \mathbb{Z}$ is the law-level evolution index in $S^{(4D)}$. Any mapping from ticks to a physical time readout (clock time or proper time along a history) is a read-only macrostate predicate on $S^{(4D)}$ histories, implemented by a specified periodic physical process in $S^{(4D)}$. This tick-to-time mapping depends on the clock/history used and has no feedback path into Φ, $w(d)$, $\hat{K}_{\text{eff}}$, any schedule, PI triggering, or C.

Invariant causal bound (DO). In internal units the invariant speed is defined on the underlying DS neighbor graph as

$$c \;=\; 1 \text{ hop/tick}.$$

All operational time labels extracted from clock histories in $S^{(4D)}$ are constrained by the same invariant causal bound used for SR bookkeeping.

In the nearest-neighbor UEE realization of Sec. 2.3 used for SR correspondence (exchange links restricted to $d_{ab} = 1$ on the underlying DS neighbor graph), one tick $K \to K + 1$ applies exchanges

 doi:10.20944/preprints202405.2028.v8

on the single matching $\mathcal{C}^{(r(K))}$; a full schedule sweep $\mathcal{U}_{\text{cycle}}$ therefore spans R ticks. Accordingly, the causal bookkeeping cone from (a, t_0) after Δn ticks is

$$\{\, b : \; d_{ab} \leq \Delta n \,\},$$

where d_{ab} is the hop distance on the underlying DS neighbor graph used for SR/light-cone bookkeeping (Chebyshev distance for the 26C cubic representative).

Realization note (finite-range exchange links). A finite-range exchange realization may employ compact-support weights $w(d)$ with cutoff R_w (Sec. 1.1.7), so that exchanged links can satisfy $d_{ab} \leq R_w$. In that case, the realization-level domain of dependence expands at most R_w hops per tick, as recorded in Sec. 2.4.3. This realization dependence bound does not alter the SR-facing bookkeeping convention above, which is tied to the underlying DS neighbor graph and invoked in the correspondence module (Sec. 2.9).

Unit mapping. Under physical-unit mapping, $c = \Delta x_{\text{hop}} / \Delta t$, with $\Delta x_{\text{hop}} = V_0^{1/3}$ (Sec. 6.1).

Appendix F.5. 6.5. Relational Energy Operator and Intrinsic Non-Quantizability

Definition (4D gravitational exchange operator). The DO gravitational exchange operator $\hat{K}_{\text{grav}}$ is the symmetric exchange operator defined in Sec. 1.1.7 on the kernel-induced neighborhood $\mathcal{N}(a)$ with weights $w_{ab} = w(d_{ab})$, with $w \in \mathcal{W}$ (Sec. 1.1.9). It is an exchange form on $S^{(4D)}$ with fixed kernel data chosen once per realization; no independent gravitational potential-times-Φ term and no separate gravitational field degree of freedom are introduced.

Sourcing and exclusion of DS CIE. Relational curvature responds only to hosted (non-CIE) state content in $S^{(4D)}$ through $\Phi(t)$ evolving on the fixed link set (Secs. 2.2–2.3); the derived samples $e_b(t) = |\Phi_b(t)|^2$ are used for collapse bookkeeping and read-only auditing. Uniform DS CIE is excluded from $e_b(t)$ and therefore contributes no local relational curvature response.

Hard-mass support-index law and audit-only potentials. In hard-mass orbital dynamics (Appendix B), coupling is evaluated from source energies via $m = E/c^2$ and the shell-gradient coefficient $\gamma(d)$ used to form the pairwise support-index force law (Sec. 2.4). The kernel-derived potential $U(w)$ and the work-closure ledger W_{closure} appear only in the post-update audit scalar $I_{\text{DO}} = E_{\text{kin}} + U - W_{\text{closure}}$ (Appendix B may print E_{kin} using the legacy label κ); they do not enter any force, update, kernel, or collapse rule.

Category statement (intrinsic non-quantizability). There is no separate gravitational degree of freedom to promote: no canonical pair (q_G, p_G), no δg, and no graviton operators. Quantization targets remain the state variables; gravitational response is carried by the relational exchange structure on $S^{(4D)}$, not by an independent field.

Neutral uniform mode.

$$\sum_a (K_{\text{grav}}\Phi)_a = 0 \quad \text{(uniform mode unaffected)}.$$

Appendix F.6. 6.6. Relative Strengths of Gravitational and Other Interactions

The observed ordering of interaction strengths matches DO's ontological mechanisms:

(I) **Strong and Weak (contact).** Adjacency-limited exchanges; no geometric dilution; per-site effects remain $O(1)$ at contact.

(II) **Electromagnetism (shell dilution).** In three spatial dimensions, shell multiplicity scales $n(r) \sim 4\pi r^2$, producing geometric dilution of localized sourced effects across shells.

 doi:10.20944/preprints202405.2028.v8

(III) **Relational gravity (difference-driven, neutral).** Gravitational response depends on differences in hosted energy distribution and is neutral on the uniform mode; DS CIE is excluded from sourcing.

$$\text{Contact (Strong/Weak)} \gg \text{Shell-diluted (EM)} \gg \text{Relational gravity (neutral).}$$

Appendix F.7. 6.7. Black Hole Information (PD Anchoring)

Invariant labels in $S^{(PD)}$. Let $\mathcal{I}_{\text{BEP}} = \{E_{QS}, Q, S_{\text{spin}}, \ldots\}$ denote conserved labels encoded in a BEP, with E_{QS} the hosted (non-CIE) intrinsic energy. Collapse acts in $S^{(PD)}$ and is mirrored to $S^{(4D)}$:

$$C: \ \phi^{(PD)}_{\text{infall}} \longrightarrow \phi^{(PD)}_{\text{rad}}, \qquad \Phi^{(4D)}_{\text{infall}}(t_{\text{pre}}) \xrightarrow{C,g} \Phi^{(4D)}_{\text{rad}}(t_{\text{coll}}),$$

with

$$\sum_i |\phi^{(PD)}_{\text{infall},i}|^2 = \sum_j |\phi^{(PD)}_{\text{rad},j}|^2 = E_{QS}, \qquad \mathcal{I}_{\text{BEP,infall}} = \mathcal{I}_{\text{BEP,rad}}.$$

This mapping is collapse-only (Sec. 1.1.8), not a time evolution within $S^{(PD)}$.

No singularity (hosted sector). Finite DS volume V_0 and the finite hosted-energy bound $E_{\text{host,max}}$ defined in Sec. 6.1 imply a finite hosted density ceiling

$$\rho^{(\text{host})}_{\text{max}} = \frac{E_{\text{host,max}}}{V_0} < \infty,$$

precluding divergent hosted densities during localized collapse processes. DS CIE remains uniform and invariant.

Appendix G. 7. Cosmology

Appendix G.1. 7.1. State of $S^{(4D)}$ at Heat Death ($t \to t_{HD}$)

As cosmic time approaches Heat Death t_{HD}, hosted matter/radiation in $S^{(4D)}$ approaches large-scale thermodynamic equilibrium: available work tends to zero ($W_{DO} \to 0$), and macroscopic observables become very nearly homogeneous and isotropic. In the same limit, PI occurrence in $S^{(4D)}$ is extremely suppressed, consistent with the DO trigger framework of Sec. 3.1.

Comparative description (reader-facing only). In standard continuum cosmology language, this limiting state is described by a spatially flat ($k \approx 0$) FLRW regime in which $a(t) \to \infty$, $\rho_m(t) \to 0$, $\rho_r(t) \to 0$, and the temperature asymptotically approaches zero, while the total energy density approaches a constant ρ_Λ. This temperature descriptor is reader-facing language for suppressed thermodynamic gradients and interaction rates; it does not alter the DO bookkeeping of hosted (non-CIE) intrinsic energy $E_{QS,HD}$ defined below. These FLRW variables are used here only as conventional descriptors; DO cosmology is recorded by DS-count bookkeeping and the invariant DS intrinsic density ρ_Λ (Secs. 7.3–7.4).

$$\lim_{t \to t_{HD}} S_{DO,grav}(t) = S_{DO,max}. \tag{A42}$$

Universal hosted-energy representation at t_{HD}. Let $E_{QS,HD}$ denote the universe's total *hosted* (non-CIE) intrinsic energy carried by occupied Bell Spheres in $S^{(4D)}$ at Heat Death. DS constant intrinsic energy (CIE) is tracked separately by ρ_Λ and is not part of the BEP energy sum used by the collapse rule (Secs. 6.1–6.2 and Secs. 7.3–7.4). Accordingly, "UBEF/UBEP" below refer to the hosted universal quantum state only.

 doi:10.20944/preprints202405.2028.v8

- $S^{(4D)}$ *(Universal Bell Energy Field, UBEF):* a state descriptor $\Phi_{HD}^{(4D)}(t)$ with hosted energy

$$\sum_{\alpha \in S^{(4D)}(t_{HD})} \left|\Phi_{HD,\alpha}^{(4D)}(t_{HD})\right|^2 \;=\; E_{QS,HD}.$$

  For the universal state at Heat Death, the occupied support (Bell Field) coincides with $S^{(4D)}(t_{HD})$.
- $S^{(PD)}$ *(Universal Bell Energy Point, UBEP):* $\Phi_{HD}^{(PD)}$, timelessly encoding $E_{QS,HD}$ and other conserved intrinsic labels of the hosted universal state.
- *Bell Identity* g*:* $\Phi_{HD}^{(4D)}(t_{HD}) \overset{g}{\longleftrightarrow} \Phi_{HD}^{(PD)}$.

The subsequent collapse of $\Phi_{HD}^{(PD)}$ defines the transition to the next cosmological epoch at $t = 0$ (Sec. 7.2).

Appendix G.2. 7.2. Cosmological Collapse Transition ($t_{HD} \to t = 0$)

The DO cosmological cycle includes one collapse of the Universal Bell Energy Point at Heat Death. This collapse is triggered by a Physical Interaction (PI) in $S^{(4D)}$ at t_{HD}, within the same PI-category framework as Sec. 3.1:

$$\mathcal{T} \;=\; \{\text{Electromagnetic, Strong, Weak}\}.$$

Appendix A does not specify the detailed microphysical trigger identity. Under Heat-Death conditions, PI occurrence is extremely suppressed; one PI event suffices to trigger the UBEP collapse.

$$C(\Phi_{HD}^{(PD)}) \;=\; \Phi_{(t=0)}^{(PD)}. \tag{A43}$$

Notation. The label "$t = 0$" is a discrete-$S^{(4D)}$ epoch tag for the mirrored cosmological transition; it does not introduce a time coordinate in $S^{(PD)}$.

Mirroring on the same tick (collapse only).

This Planck-Domain collapse is mirrored by g as a generalized localization of the Universal Bell Energy Field in $S^{(4D)}$ on the same tick:

$$\Phi_{HD}^{(4D)}(t_{HD}) \quad \xrightarrow{C,g} \quad \Phi_{(t=0)}^{(4D)}. \tag{A44}$$

This same-tick mirroring is not signal propagation in $S^{(4D)}$; it is Bell-Identity mirroring of one collapse event in $S^{(PD)}$. The post-collapse $S^{(4D)}$ support is confined to a pre-existing connected subset of the pre-collapse support (Sec. 1.1.8), and post-collapse hosted energy is preserved by the Sec. 1.1.8 rescaling rule.

Heat-Death fragility (DO interpretation). The Universal Bell Energy Point at Heat Death is an ontic free-state limit in which PI occurrence is extremely suppressed; the cosmological transition is initiated by the single PI event that triggers collapse, and the mirrored $S^{(4D)}$ localization yields extreme but finite conditions at $t = 0$ (Sec. 6.1).

Resulting properties.

1. **Near homogeneity/isotropy in the $t = 0$ state class:** the DO-text $t = 0$ state class is specified as near-homogeneous and near-isotropic in hosted (non-CIE) energy with only infinitesimal anisotropy seeds (Validation I in Appendix B).
2. **Finite (not infinite) densities:** DS discreteness bounds hosted energy density, pressure, and temperature at $t = 0$ (Sec. 6.1).
3. **Entropy reset:** collapse from a dispersed Heat-Death state to a localized $t = 0$ state resets DO gravitational entropy from near-maximal to near-minimal.
4. **Invariant DS substructure:** DS do not collapse; their volume V_0 and intrinsic density ρ_Λ remain invariant through the transition (Secs. 6.1 and 7.4).

 doi:10.20944/preprints202405.2028.v8

Appendix G.3. 7.3. $S^{(4D)}$ Expansion, ρ_Λ, and Energy Accounting

In the DO model, $S^{(4D)}$ expands by including additional Discrete Spheres (DS) from the pre-existing DS substructure into the $S^{(4D)}$ subset. This is recorded by the set update and DS-count update:

$$S^{(4D)}(t+\Delta t) = S^{(4D)}(t) \cup S^{(4D)}_{\text{new}}, \qquad N_{4D}(t+\Delta t) = N_{4D}(t) + \Delta N_{4D}, \qquad \Delta N_{4D} \geq 0.$$

The DS-count $N_{4D}(t)$ is monotone nondecreasing across ticks by definition of expansion bookkeeping. This bookkeeping records expansion into pre-existing DS and does not specify a solver-level evolution law for $N_{4D}(t)$.

Scope (expansion rate). Appendix A uses only the set/count bookkeeping identities above and the separation of hosted (non-CIE) energy from DS CIE. The tick-by-tick sequence $\{\Delta N_{4D}\}$ is not derived from Φ, $w(d)$, or C inside Appendix A; it is fixed realization data with no feedback path into any DO update.

Metadata separation. Any attached comparison labels $K \mapsto (a, z, \tau)$ are metadata only (Scope and Linkage) and do not define ΔN_{4D} and do not enter any DO update rule.

Energy accounting (DS CIE contribution). Let ρ_Λ denote the DS constant intrinsic energy density (CIE) per unit volume in $S^{(4D)}$ (Sec. 6.1). The total CIE energy carried by the DS in $S^{(4D)}(t)$ is

$$E_{\Lambda,\text{tot}}(t) \;=\; N_{4D}(t)\, E_\Lambda \;=\; N_{4D}(t)\, \rho_\Lambda V_0,$$

so the CIE energy increment over one tick is

$$\Delta E_{\Lambda,\text{tot}}(S^{(4D)}, t) = \Delta N_{4D} \cdot E_\Lambda = \Delta N_{4D} \cdot (\rho_\Lambda V_0).$$

Hosted-energy bookkeeping. Hosted (non-CIE) energy carried by states on occupied Bell Spheres is tracked separately (Secs. 6.1–6.2). As a reader-facing decomposition, one may write

$$\rho_{\text{total}}(t) = \rho_\Lambda + \rho_m(t) + \rho_r(t),$$

with the comparative FLRW dilution laws

$$\rho_m \propto a^{-3}, \qquad \rho_r \propto a^{-4},$$

and equation-of-state label $w_\Lambda = -1$. These scaling laws are conventional descriptors. The DO ontic content is the invariant ρ_Λ, the DS-count bookkeeping $N_{4D}(t)$, and the hosted-energy distribution on $S^{(4D)}$ (Secs. 2.2 and 6.1).

Appendix G.4. 7.4. Λ Properties and Stability (Discrete Spheres)

Constancy (DO-ontic).

$$\rho_\Lambda(t+\Delta t) = \rho_\Lambda(t),$$

because DS CIE is uniform and invariant across DS.

Reader correspondence (GR/FLRW notation only). In standard continuum notation, this same constant density is conventionally reported as

$$\frac{d\rho_\Lambda}{dt} = 0.$$

The symbol Λ appears only in this reader-facing unit/symbol map; DO takes ρ_Λ as an ontic DS attribute and introduces no field-equation-based derivation within Appendix A.

 doi:10.20944/preprints202405.2028.v8

Origin (PD–4D linkage). DS intrinsic energy attribute $\longleftrightarrow$ constant 4D density ρ_Λ; expansion increases N_{4D} but not ρ_Λ.

Appendix G.5. 7.5. Hierarchy Problem (Discrete Cutoff + Relational Neutrality)

(1) Discreteness $\Rightarrow$ bounded UV sums. On any finite DS realization, the discrete spectral domain is bounded, so UV mode sums are finite (no divergent high-frequency contributions).

(2) Relational neutrality (uniform modes untouched).

$$\sum_x K_{\text{eff}}(x) = 0, \qquad \widehat{K}_{\text{eff}}(0) = 0,$$

so uniform modes (BEP-intrinsic labels) are untouched by the exchange operators; no Planck-scale gravitational dressing of BEP-intrinsic labels is introduced by the DO operator.

(3) Observed ordering (from Sec. 6.6).

$$\text{Contact (Strong/Weak)} \gg \text{Shell-diluted (EM)} \gg \text{Relational gravity (neutral).}$$

Audits.

$$\widehat{K}_{\text{eff}}(\mathbf{k}) \le 0 \quad (\text{low-}k),\ n(r) \sim 4\pi r^2,\ R_w < \infty,$$

all discrete UV sums bounded on the finite spectral domain.

Appendix B. Relational Gravity and Quantum Nonlocality Validation Models

Appendix B integrates validation families comprising dynamical simulations and statistical verifications derived entirely from the unified, background-independent, discrete ontology, relational gravity, and quantum dynamics of the Dual Ontology (DO) model as formalized in Appendix A. The validations avoid Hilbert spaces and matrix mechanics, non-relativistic Schrödinger equations, continuum field metrics, Newtonian or EFE formulations, ad-hoc mid-run steering, and perturbative approximations. No validation treats a Euclidean metric or Euclidean equations of motion as primitive; any floating-point coordinate chart used for hard-mass bookkeeping, numerical integration, or plotting is realization/implementation index data with no ontic status and no feedback path into the DO operator, kernel, collapse rule, or auditing gates. Within each validation run, the DO operator, collapse rule (when applicable), and auditing definitions are held fixed; numerical run controls (e.g. time step, run length, and sampling density) are fixed within a run and may differ across validation families and horizons. Where an initial-condition calibration is performed, it is a one-time pre-run selection of launch conditions to place the system in a specified dynamical band; the resulting initial-condition values are then held fixed across the stated horizons.

Run artifacts (Appendix C). Appendix B is a self-contained discussion of validation methodology and results. All executable scripts, console transcripts, and raw run artifacts are provided in Appendix C and are referenced in Appendix B in the form `[See Appendix C, Section __]`.

Operator and collapse linkage. Validation I is a field-evolution validation on a fixed periodic torus graph in $S^{(4D)}$ with deterministic between-collapse evolution and read-only bridges. It evolves a DO-text $t = 0$ field by the UEE tick map realized as one disjoint-link matching per tick on a periodic torus graph (fixed schedule; uniform-mode neutral by construction).

Validation II (CHSH) statistically verifies the analytical derivation of Tsirelson's bound. It uses only the Planck Domain (PD) collapse rule in $S^{(PD)}$, mirrored into $S^{(4D)}$ by the Bell Identity g, and does not employ the UEE tick-map evolution.

Whenever a validation evaluates collapse probabilities, it does so relative to a specified outcome family $\{O'_j\} \subset \text{Conn}(\mathcal{C})$ that is pairwise disjoint and satisfies $\bigsqcup_j O'_j = \mathcal{O}(t_{\text{pre}})$; the probability as-

 doi:10.20944/preprints202405.2028.v8

signed to each O'_j is the fraction of the state's total hosted (non-CIE) intrinsic energy contained in O'_j immediately before collapse (Appendix A, Sec. 3.2).

Validations III–A1, III–A2, III–B1, III–B2, IV1, and IV2 are hard-mass validations on a finite Chebyshev box using a single flux-defined shell field $\gamma(d)$, with $w(d)$ recovered by the discrete inward integral, to supply the pairwise interaction term $\chi_E\, m_i m_j\, \gamma(d_{ij})$. These hard-mass updates implement the singleton-support (hard-mass) limit stated in Appendix A, Sec. 2.4: component fields remain localized to one Bell Sphere per tick and only the support indices propagate under the realization's $\gamma(d)$ and direction rule. In these N-body validations, the interaction direction is given by the background-independent Manhattan–fraction direction rule $\hat{\mathbf{u}}_{ij}$, while the chart-evaluated Chebyshev separation $d_{ij} \in \mathbb{R}_{\geq 0}$ is read from the diagnostic index chart and $\gamma(d)$ and $w(d)$ are evaluated at non-integer d by the fixed shell interpolation convention defined in the Introduction. Time evolution uses the hard-mass classical-limit update law (Appendix A, Sec. 2.4) with a fourth-order Forest–Ruth symplectic update.

Audit scalar and work-closure ledger. Long-horizon conservation is audited using the DO invariant

$$I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}},$$

together with angular momentum and barycenter diagnostics. Here $E_{\mathrm{kin}} + U$ is reported separately only as a texture diagnostic; I_{DO} is the conservation gate (Appendix C artifacts may print E_{kin} as K and $E_{\mathrm{kin}} + U$ as K+U).

Define the kernel-defined potential (from the recovered $w(d)$) at sample time t_n by

$$U_n := \sum_{i<j} \chi_E\, m_i m_j\, w\big(d_{ij}(t_n)\big).$$

Define the realized pairwise discrete work increment over $t_n \to t_{n+1}$ by

$$W_n := \sum_{i<j} \mathbf{F}_{ij}(t_n) \cdot \Big(\mathbf{r}_{ij}(t_{n+1}) - \mathbf{r}_{ij}(t_n)\Big), \qquad \mathbf{r}_{ij}(t) := \mathbf{r}_i(t) - \mathbf{r}_j(t).$$

Define the closure increment and ledger by

$$\Delta W_{\mathrm{closure},n} := W_n + (U_{n+1} - U_n), \qquad W_{\mathrm{closure}}(t_N) := \sum_{n=0}^{N-1} \Delta W_{\mathrm{closure},n}, \qquad W_{\mathrm{closure}}(t_0) = 0.$$

The conservation gate is the observed flatness of $I_{\mathrm{DO}} := E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$ over the analysis window; $E_{\mathrm{kin}} + U$ is reported separately only as a texture diagnostic. I_{DO} tests closure of realized discrete work against the kernel-defined potential along the executed trajectory. I_{DO} is computed post-update and has no feedback path into any position, velocity, force, kernel, or parameter update.

In Validation I a single global amplitude on power κ is fitted for display/comparison only and is not a parameter of the theory.

Shell-index evaluation rule (hard-mass validations). In the hard-mass validations (III and IV), body support-index positions are stored in a floating-point diagnostic index chart for integration and reporting, so separations d_{ij} may be non-integer. The shell-indexed fields $\gamma(d)$ and $w(d)$ are defined on integer shell indices $k \in \{0, 1, \ldots, R_{\max}\}$. Writing $k = \lfloor d \rfloor$ and $\alpha = d - k \in [0, 1)$, $\gamma(d)$ is evaluated at non-integer d by deterministic linear interpolation across adjacent shells,

$$\gamma(d) = (1 - \alpha)\gamma(k) + \alpha\gamma(k + 1),$$

and $w(d)$ is evaluated by the corresponding integral-consistent interpolation induced by this γ convention,

$$w(d) = w(k) + 2\Big[\gamma(k)\,\alpha + \tfrac{1}{2}\big(\gamma(k+1) - \gamma(k)\big)\alpha^2\Big],$$

 doi:10.20944/preprints202405.2028.v8

with clamping at the boundaries $k = 0$ and $k = R_{\max} - 1$. This convention is fixed per run and introduces no tunable coefficient. The fixed shell evaluation convention is realization data for the hard-mass regime and is the convention referenced by Appendix A's hard-mass classical-limit update law.

Validation I: $t = 0 \to$ CMB (low–ℓ TT). A fixed relational pipeline instantiates the DO-text cosmogonical $t = 0$ state class (post–Universal Bell Energy Point collapse at or near Heat Death) as a near-homogeneous and near-isotropic hosted-energy field with fractional anisotropy seeds $\varepsilon_{\text{seed}} = 10^{-15}$. The field evolves deterministically in $S^{(4D)}$ under the UEE tick map realized as one disjoint-link matching per tick on a periodic 69^3 torus graph (one sweep: $R_{\text{sweep}} = 39$ ticks), with exact Bell-sum conservation. A read-only shell bridge maps the evolved field to low–ℓ TT bandpowers and fits one global amplitude on power κ over $\ell = 5 \ldots 50$ (no shape tuning). Gates include axis-permutation rotation invariance and ε-linearity; the production run satisfies all gates with $\chi^2(5\text{–}50) = 49.876$. [See Appendix C, Section C.1].

Validation II: spin correlations and CHSH. Using the Planck Domain (PD) collapse rule in $S^{(PD)}$, mirrored into $S^{(4D)}$ by the Bell Identity g, the planar spin-$\frac{1}{2}$ law $E(\theta) = -\cos\theta$ and Tsirelson's bound $|S| = 2\sqrt{2}$ are reproduced via Monte Carlo integration, validating the interaction-manifold amplification factor $\nu = 2$ without tunables. [See Appendix C, Section C.2].

Validation III–A1: Earth–Moon two–body baseline (1-year). A low-energy, mildly elliptical Earth–Moon analog used to test conservation of the DO invariant, angular momentum, and barycenter under a fixed DO gravitational kernel and coupling χ_E. Using the same flux-defined $\{w, \gamma\}$ realization and Manhattan–fraction direction rule as in Validation IV, a two-body orbit closes at 360° within tolerance while maintaining I_{DO}, L_z, and barycenter invariance at the stated bounds; the naive mechanical energy $E_{\text{kin}} + U$ exhibits a stable texture band under the same fixed update (Appendix C artifacts may print $E_{\text{kin}} + U$ as K+U). [See Appendix C, Section C.3].

Validation III–A2: Earth–Moon two–body long horizon (1000-orbit). The Earth–Moon hard-mass operator of Validation III–A1 is extended to the 1000-orbit horizon. The force law, flux-defined $\{w, \gamma\}$ realization, coupling χ_E, and Manhattan–fraction direction rule remain unchanged; numerical run controls for the long-horizon integration (e.g. time step and sampling stride) are fixed within this run and are documented in the run artifacts. Boundedness and invariant control are preserved while long-horizon stability and diagnostic texture behaviour are audited under the unchanged operator. [See Appendix C, Section C.4].

Validation III–B1: three–body coupling baseline (short horizon). A low-energy hierarchical three-body configuration with a nearly circular inner orbit and a bounded outer orbit around the barycenter. Reusing the same flux-defined $\{w, \gamma, \chi_E\}$, Manhattan–fraction direction rule, and DO invariant auditing $I_{\text{DO}} = E_{\text{kin}} + U - W_{\text{closure}}$, the three-body system remains bounded over the baseline horizon with invariant control and the expected hierarchical perturbations. The inclined-plane realization is produced through the launch geometry alone (fixed tilt), with no change to the operator or update. [See Appendix C, Section C.5].

Validation III–B2: three–body coupling long horizon (1000 inner orbits). The same three-body operator and diagnostics extend cleanly to a 1000-inner-orbit horizon with stable long-horizon behaviour and invariant control under genuine coupling. [See Appendix C, Section C.6].

Validation IV1: high–energy Mercury–Sun baseline (1-year). A higher-energy, moderately elliptical hard-mass Mercury–Sun analog baseline used to establish boundedness and invariant control in a high-acceleration regime under the same flux-integral DO gravitational operator. [See Appendix C, Section C.7].

Validation IV2: high–energy Mercury–Sun long horizon (1000 orbits) — DO-text §7.5. A higher-energy, moderately elliptical hard-mass Mercury–Sun analog used to test the flux-integral DO gravitational operator and the DO invariant over a 1000-orbit long horizon. On the same 401^3 Chebyshev

 doi:10.20944/preprints202405.2028.v8

box, a single flux-defined shell field $\gamma(d)$ (with $w(d)$ recovered by the discrete inward integral) and coupling χ_E drive a two-body orbit at launch radius $D_0 = 60$ using a Manhattan–fraction direction rule and the hard-mass classical-limit update law (Appendix A, Sec. 2.4). The extended 1000-orbit run ($\sim 5.8 \times 10^9$ discrete updates) preserves hard-mass boundedness and invariant control with I_{DO} and L_z remaining flat at the 10^{-11}–10^{-12} level, and yields a converged refined apsidal-advance diagnostic of $\langle \Delta\omega \rangle_{\mathrm{refined}} \approx 88.685$ arcsec/orbit, $\sigma_{\Delta\omega,\mathrm{refined}} \approx 7.406$ arcsec/orbit. The magnitude and structured long-horizon behaviour of the apsidal advance are interpreted as a prediction of this specific discrete kernel and Chebyshev-shell band in the high-energy regime, not as a failure of the underlying relational DO law. [See Appendix C, Section C.8].

Terminology note (read-only). Appendix C run artifacts sometimes print the conserved audit scalar $I_{\mathrm{DO}} = (E_{\mathrm{kin}} + U) - W_{\mathrm{closure}}$ using the legacy label `H_DO`. Some artifacts also print E_{kin} and $E_{\mathrm{kin}} + U$ using the legacy console labels `K` and `K+U`. The reported values are identical; only the labels differ.

Appendix B.1. Validation I: $t = 0 \to$ CMB (low–ℓ TT)

This validation documents one end-to-end, DO-compliant production run that maps a DO-text $t = 0$ field with ultra-small hosted-energy anisotropy seeds $\varepsilon_{\mathrm{seed}} = 10^{-15}$ to the low-ℓ TT bandpowers ($\ell = 2 \ldots 50$) using a fully discrete, read-only shell bridge. The run uses deterministic evolution in $S^{(4D)}$ only (no stochastic edits inside the 4D tick loop), and barycenter and compares to Planck low-ℓ TT using a single global amplitude on power κ (no shape tuning).

Cosmogonical status (DO-text $t = 0$). This validation is written and executed as a DO-text $t = 0$ demonstration. In the DO cosmological cycle, $t = 0$ is the post–Universal Bell Energy Point collapse state at or near Heat Death of $S^{(4D)}$, mirrored into $S^{(4D)}$ by the Bell Identity. The $t = 0$ state class instantiated here is near-homogeneous and near-isotropic in hosted (non-CIE) energy, with only infinitesimal anisotropy seeds, consistent with the constraints implied by the CMB. The evolution proceeds on a neutral/flat background in the DO sense (uniform-mode neutral, with no imposed curvature bias). Under a fixed DO-compliant tick law in $S^{(4D)}$ and a read-only discrete angular bridge, this $t = 0$ state class yields the observed low-ℓ TT shape with only one global amplitude on power. No inflationary mechanism, continuum transfer model, or mid-run steering is introduced.

This section:

- States the DO compliance conditions enforced in the run.
- Specifies the end-to-end pipeline from DO-text $t = 0$ initialization to χ^2.
- Defines the κ-fit and χ^2 evaluation exactly as applied.
- Reports the numerical outcomes, bandwise contributions, and validation gates.
- Provides references to the run artifacts preserved in Appendix C.

This section does not:

- Introduce any additional free parameters beyond κ.
- Modify the evolution law mid-run.
- Allow any diagnostic gate to feed back into Φ, the tick law, or the bridge.

Appendix B.2.1.1. Appendix B.1.1. DO Compliance: First Principles

Discreteness and background independence. All evolution and observables are computed on finite discrete structures: a periodic 69^3 torus graph for the 4D field evolution and a finite Chebyshev shell graph for angular projection. No continuum geometry, PDEs, or spherical-harmonic surrogates are used in the dynamics.

Deterministic evolution in $S^{(4D)}$. The run enforces `DO_COLLAPSE_EVERY = 0`. There are no stochastic edits or in-loop redistributions inside the $S^{(4D)}$ tick evolution.

Single nearest-neighbor angle at $d = 1$. All nearest-neighbor exchanges use a single $\theta(1) = 0.02$ (no split into face/edge/corner angles).

 doi:10.20944/preprints202405.2028.v8

Fixed link set and fixed schedule. The torus neighbor structure is fixed and periodic. Each tick applies one disjoint-link matching, and the schedule is fixed and deterministic across the entire run.
SR-facing causality bookkeeping. One tick corresponds to one matching application; there is no node reuse within a tick. In this validation, the exchange neighborhood is restricted to the $d = 1$ nearest-neighbor set, so the SR-facing bookkeeping cone uses one hop per tick on the underlying DS neighbor graph.
Bell-sum conservation between collapses. The two-site exchange update is norm-preserving by construction. The Bell-sum audit $\sum |\Phi|^2$ is reported and remains constant to printed precision over the run.
Diagnostics and bridges are read-only. Shell weights, eigensolve basis checks, κ-fit, χ^2, rotation invariance, and ε-linearity are computed post hoc and have no feedback path into Φ or the evolution law.

Sub-rosa $t = 0$ conditions (enforced by construction). The DO-text $t = 0$ conditions are not additional tunables in this validation; they are built into the state class and the fixed law. If they were absent, the output would change qualitatively (broad anisotropy, labeling dependence, or bandwide misfit). Concretely:

1. *Near homogeneity:* $E = |\Phi|^2$ is initialized with unit mean and bounded mean-zero residuals at $\varepsilon_{\text{seed}} = 10^{-15}$, placing the run in the infinitesimal-anisotropy regime described for DO $t = 0$.
2. *Near isotropy:* a single $\theta(1)$ is used for all nearest-neighbor exchanges, the shell bridge is shell-isotropic, and the rotation gate enforces invariance under axis permutations of Φ.
3. *Flat/neutral background:* the uniform mode is neutral under the tick law, and periodic boundaries remove boundary-driven large-scale gradients; the bridge geometry uses $\sum w_\Omega = 4\pi$ as a shell-measure guardrail.
4. *SR-facing causality bookkeeping:* one tick applies one disjoint-link matching, preventing chained reuse of a node within a single tick; realized dependence per tick remains bounded by the realization's fixed exchange neighborhood.
5. *No inflationary smoothing:* no auxiliary smoothing, forcing, rescaling, or transfer function is introduced to "repair" homogeneity/isotropy; the only evolution is the fixed DO tick map.
6. *Energy integrity:* Bell-sum conservation is audited explicitly, and the run exhibits zero drift to printed precision across the full production horizon.
7. *No diagnostic feedback:* all gates and the κ calibration are computed post hoc and have no feedback path into Φ, the tick law, the shell geometry, or the bridge.

Appendix B.2.1.2. Appendix B.1.2. End-to-End Pipeline (Single Production Run)

Stage 0 — DO-text $t = 0$ initialization. Initialize $\Phi(t = 0)$ from a mean-zero, bounded hosted-energy seed $e = 1 + \varepsilon_{\text{seed}} s$ with $\varepsilon_{\text{seed}} = 10^{-15}$, then set $\Phi = \sqrt{e}$ (complex field with real amplitude). Report seed statistics and Bell-sum.
Stage 1 — Deterministic $S^{(4D)}$ evolution on a periodic torus graph. Evolve Φ for one full schedule sweep of length $R_{\text{sweep}} = 39$ ticks. Each tick applies exactly one disjoint-link matching with the fixed $\theta(1)$. Report Bell-sum drift checkpoints and final Bell-sum audit. This deterministic evolution is the between-collapse Unified Evolution Equation (UEE) in its tick-map form (Appendix A), realized here with one disjoint-link matching applied per tick and a fixed periodic schedule; the $R_{\text{sweep}} = 39$ tick sweep is one full schedule cycle.
Stage 2 — Shell geometry and guardrails. Construct the Chebyshev shell $R_{\text{shell}} = 30$ with 26-connectivity and compute face-aware angular weights w_Ω normalized so that $\sum w_\Omega = 4\pi$. Report the shell-measure guardrail.
Stage 3 — Mode basis on the shell. Solve the weighted Laplacian eigenproblem to obtain $N_{\text{modes}} = (\ell_{\max} + \text{pad} + 1)^2$ lowest modes (here $N_{\text{modes}} = 2916$ for $\ell_{\max} = 50$ with pad 3). Define the diagonal weight matrix $M := \text{diag}(w_\Omega)$ from the shell weights and report the numerical closure residual

 doi:10.20944/preprints202405.2028.v8

$\|V^\top MV - I\|_F$ for the computed diagnostic mode block V. This basis V is a read-only numerical bridge object used only for the shell projection and has no ontological role in the DO model.

Stage 4 — Read-only calibration tensor T. Build the linear response tensor $T_{\ell,s,g}$ by seeded-q plane-wave probes (cos+sin) across Fourier radial bins s and orientation classes g. This calibration is fixed for the run and is not tuned to data.

Stage 5 — Spectrum $P_{s,g}$ from the evolved field. Compute the energy residual $E - \langle E \rangle$ with $E = |\Phi|^2$, scale by $\varepsilon_{\text{seed}}$ for FFT conditioning, compute $P_{s,g}$, then rescale power by $\varepsilon_{\text{seed}}^2$.

Stage 6 — Prediction and κ-fit. Compute $C_\ell^{\text{pred}} = \sum_{s,g} T_{\ell,s,g} P_{s,g}$ for $\ell = 2 \ldots 50$. Fit a single global amplitude κ (whose square scales power) over the fit window $\ell = 5 \ldots 50$, producing $C_\ell^{\text{model}} = \kappa^2 C_\ell^{\text{pred}}$. The low multipoles $\ell = 2, 3, 4$ are computed but excluded from the κ-fit and χ^2 evaluation.

Stage 7 — Gates. Recompute the fit under axis permutations of Φ (rotation gate) and evaluate ε-linearity on probe responses (bridge-only).

Appendix B.2.1.3. Appendix B.1.3. Configuration of the Final Run

- Torus-graph evolution: $N = 69$, one sweep $R_{\text{sweep}} = 39$ ticks, $\theta(1) = 0.02$, `DO_COLLAPSE_EVERY=0`.
- DO-text seed: $\varepsilon_{\text{seed}} = 10^{-15}$, `SEED_ANISO=20250101`.
- Shell: $R_{\text{shell}} = 30$, 26-connectivity, $\sum w_\Omega = 4\pi$ to 10^{-16} relative error.
- Basis: $\ell_{\max} = 50$, pad $= 3$, $N_{\text{modes}} = 2916$.
- Bridge: $BINS = 32$, seeded-q buckets $\to$ 7744 buckets used, `CALIB_EPS` $= 0.5$, `CALIB_CHUNK_Q` $= 32$.
- Fit window: $\ell = 5 \ldots 50$ (46 points).
- Gates: rotation permutations $x \leftrightarrow y$, $y \leftrightarrow z$, $x \leftrightarrow z$; ε-linearity probe $\varepsilon \in \{0.25, 0.75\}$ with 96 samples.

Fixed vs. fitted (audit checklist).

- *Fixed a priori:* DO-text seed scale $\varepsilon_{\text{seed}}$ and seed ID; tick law, schedule, and $\theta(1)$; run horizon (one sweep); shell radius and angular weights; eigensolve target K; binning $(BINS)$; seeded-q bucket rule (`SEED_SPLIT`); calibration probe amplitude and policy (`CALIB_EPS`, `CALIB_CHUNK_Q`); fit window $\ell = 5 \ldots 50$; and all gate thresholds.
- *Fitted:* κ only (single global amplitude on power). No per-ℓ or shape parameters are fitted.

Appendix B.2.1.4. Appendix B.1.4. κ–Fit and χ^2

Status of κ (operational bridge only). κ is a post-processing amplitude on *power* used only to map units for comparison; it is not referenced by any update, operator, kernel, or collapse rule and has no feedback path into Φ or any evolution map (Appendix A, κ-module fence).

Let C_ℓ^{pred} denote the unscaled predicted bandpowers from the fixed bridge and the evolved field. The model comparison uses a single amplitude on power:

$$C_\ell^{\text{model}} = \kappa^2 \, C_\ell^{\text{pred}}.$$

Define $\lambda_\kappa \equiv \kappa^2$. The diagonal weighted least-squares minimizer over the fit window $L = \{\ell : 5 \le \ell \le 50\}$ is

$$\lambda_\kappa^\star = \frac{\sum_{\ell \in L} \left(C_\ell^{\text{pred}} C_\ell^{\text{obs}} / \sigma_\ell^2 \right)}{\sum_{\ell \in L} \left((C_\ell^{\text{pred}})^2 / \sigma_\ell^2 \right)}, \qquad \kappa = \sqrt{\lambda_\kappa^\star},$$

and

$$\chi^2 = \sum_{\ell \in L} \frac{\left(C_\ell^{\text{model}} - C_\ell^{\text{obs}} \right)^2}{\sigma_\ell^2}.$$

 doi:10.20944/preprints202405.2028.v8

Appendix B.2.1.5. Appendix B.1.5. Validation Gates

Rotation invariance. Permute axes of the evolved Φ and recompute the full bridge and fit. The reported χ^2 must be stable across permutations.
ε-linearity. For probe amplitudes $\varepsilon \in \{0.25, 0.75\}$, verify the quadratic response $C_\ell \propto \varepsilon^2$ to numerical precision (reported as mean and max fractional spread).

Appendix B.2.1.6. Appendix B.1.6. Results (Single Production Run)

Summary.

- Fit: $\kappa = 2.089482181263314 \times 10^{13}$; $\chi^2(5\text{–}50) = 49.876$ over 46 points ($\chi^2/(n-1) = 1.108$).
- Basis guardrail: $\|V^\top MV - I\|_F = 1.024 \times 10^{-12}$ at $N_{\text{modes}} = 2916$, with $M := \text{diag}(w_\Omega)$.
- Rotation gate: χ^2 identical across all axis permutations (spread 0.000).
- ε-linearity: mean 4.163×10^{-5}, max 1.485×10^{-3}.

Residual localization. The dominant $\Delta\chi^2$ contributors in the fit window are $\ell = 5$ and $\ell = 7$, with additional concentrated contributors near $\ell = 40$ and $\ell = 48$. Bandwise contributions using the same frozen κ are reported in the run transcript in Appendix C.

Appendix B.2.1.7. Appendix B.1.7. Limitations and Reproducibility Notes

This section reports a single deterministic production run. The dominant runtime cost is the shell eigensolve for $N_{\text{modes}} = 2916$ modes; torus evolution is negligible by comparison. All computations are performed in `float64` except the stored field Φ (complex128 in this implementation). Seeds and configuration values are fixed and printed in the transcript.

Seed-scale practicality. The DO cosmogonical narrative permits hosted-energy anisotropy fractions far smaller than 10^{-15} in the Heat-Death limit; here $\varepsilon_{\text{seed}} = 10^{-15}$ is used as a numerically tractable representative of an *extremely small* anisotropy regime that remains well-conditioned for FFT-based power estimation on a finite grid. This value is fixed a priori (not tuned to χ^2). The ε-linearity gate confirms that the bridge response is in the expected quadratic scaling regime, so smaller $\varepsilon_{\text{seed}}$ values are a conditioning/runtime extension rather than a change in the underlying DO mechanism.
Display convention. Where a D_ℓ display is used elsewhere, $D_\ell := \frac{\ell(\ell+1)}{2\pi} C_\ell$. This validation reports C_ℓ directly in the transcript.

Appendix B.2. Validation II: First-Principles Derivation of CHSH Violation

A central challenge for any fundamental theory is to derive the correlations of quantum mechanics from its own physical ontology. This section demonstrates that the Dual Ontology (DO) model's core postulates can meet this challenge.

In the DO model, a two-particle singlet state is not two separate entities, but a single, unified system composed of the vast population of Bell Spheres from both constituent quantum states. The defining characteristic of this unified singlet state is the singlet anti-correlation constraint encoded in the unified BEP in $S^{(PD)}$, mirrored by g into $S^{(4D)}$ (see Appendix A, Sec. 2.6.1).

While this relative anti-correlation is fixed, the specific ± 1 outcomes relative to any chosen analyzer axis are not determinate prior to the PI that triggers collapse. The macroscopic correlation $E(\mathbf{A}, \mathbf{B})$ reported in the Bohm/EPR experiment is the physical average of interaction outcomes over the Bell Sphere population. This simulation numerically evaluates that average to validate the DO prediction.

Appendix B.2.2.1. Appendix B.2.1. Method (DO-native)

This validation numerically evaluates the DO singlet correlation law and then computes the CHSH statistic.

Setup. A two-body singlet is represented in the DO as one unified system whose collapse selection yields anti-correlated outcomes relative to any chosen analyzer axes. Each wing contains an analyzer

 doi:10.20944/preprints202405.2028.v8

with a chosen axis setting (e.g. $\mathbf{A}, \mathbf{A}'$ on one side and $\mathbf{B}, \mathbf{B}'$ on the other). A PI in $S^{(4D)}$ triggers collapse of the unified BEP in $S^{(PD)}$, mirrored into $S^{(4D)}$ by the Bell Identity g; the second wing's readout is fixed by that same collapse selection. The read-out labels ± 1 are 4D macrostate labels defined relative to the pre-existing analyzer axis setting.

Numerical evaluation. A Monte Carlo average is used to estimate the correlation $E(\mathbf{X}, \mathbf{Y})$ for the four axis pairs $(\mathbf{A}, \mathbf{B})$, $(\mathbf{A}, \mathbf{B}')$, $(\mathbf{A}', \mathbf{B})$, $(\mathbf{A}', \mathbf{B}')$. The script evaluates the DO correlation functional as a Planck-Domain population average in the planar interaction manifold ($\nu = 2$): sample a random planar unit vector $\mathbf{u}$ uniformly, represent singlet anti-correlation by pairing $\mathbf{u}$ with $-\mathbf{u}$ across the two wings, and compute

$$E(\mathbf{X}, \mathbf{Y}) \;=\; -\nu \left\langle (\mathbf{X} \cdot \mathbf{u})(\mathbf{Y} \cdot \mathbf{u}) \right\rangle, \qquad \nu = 2,$$

where $\langle \cdot \rangle$ denotes the Monte Carlo average over the sampled $\mathbf{u}$. The sampled $\mathbf{u}$ is an integration parameter for this population average and is not an ontic object in the DO model.

CHSH construction.

$$S \;=\; E(\mathbf{A}, \mathbf{B}) \;-\; E(\mathbf{A}, \mathbf{B}') \;+\; E(\mathbf{A}', \mathbf{B}) \;+\; E(\mathbf{A}', \mathbf{B}').$$

Run artifacts. The executable script and console transcript are provided in Appendix C, Section C.2.

Scope (what is validated). Validation II is not a new laboratory protocol. It is a numerical check that the DO PD-collapse rule for a unified singlet reproduces the 4D coincidence statistics used in the standard CHSH experiment (the correlations and the resulting S value). The Planck-Domain collapse mechanism is not directly observable; the validation targets only the observable 4D statistics.

Terminology note (read-only). The run artifacts in Appendix C use legacy "measurement" strings; in the DO model the corresponding physical term is a PI-triggered collapse in $S^{(PD)}$, mirrored into $S^{(4D)}$.

Appendix B.2.2.2. Appendix B.2.2. Analysis and Results

The Monte Carlo evaluation yields $|S| = 2.8284$, matching Tsirelson's bound $2\sqrt{2}$ to the reported precision. The per-setting correlations match the expected $-\cos\theta$ values within sampling error.

Appendix B.2.2.3. Appendix B.2.3. Conclusion

Validation II confirms that the DO singlet correlation law reproduces Tsirelson's bound under the CHSH construction, using only DO collapse structure and read-only numerical evaluation (Appendix C, Sec. C.2).

Appendix B.3. Validation III–A: Two-Body Validation: Methodology

Appendix B.2.3.1. Appendix B.3.1. General Methodology and Principles

Validation III–A provides a concrete numerical implementation and validation of the hard-mass classical limit of the Unified Evolution Equation (UEE) for two bodies, as formalized in Appendix A. This validation family comprises Validation III–A1 (Earth–Moon two-body baseline, 1-year horizon) and Validation III–A2 (Earth–Moon two-body long horizon, 1000-orbit "1000-orbit" horizon). The validation is an Earth–Moon analog in the mildly elliptical regime and is used to test long-horizon stability of the DO gravitational operator in the hard-mass limit. The analysis presents a primary 1-year baseline together with an extended long-horizon run defined by 1000 completed orbital cycles. The operator and conservation auditing remain fixed; numerical run controls (including dt, run length, and sampling density) differ by horizon while the integrator family remains Forest–Ruth throughout. In dynamics the interaction magnitude is supplied by $\gamma(d_{ij})$; $w(d_{ij})$ enters only through the potential U used in the audit scalar $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$ (Appendix C artifacts may print E_{kin} as K).

 doi:10.20944/preprints202405.2028.v8

The simulations are not intended to reproduce a measured Earth–Moon ephemeris, but to validate internal consistency and physical realism of the DO operator under controlled conditions. The core principles, drawn directly from Appendix A, are:

- **Background Independence & Discrete Spacetime:** The model does not presume a pre-existing spacetime continuum. Hard-mass bodies propagate as support indices on the Bell-Sphere network, and separations are evaluated by the chart-evaluated Chebyshev separation

$$d_{ij} = \max\{|x_i - x_j|, |y_i - y_j|, |z_i - z_j|\} \in \mathbb{R}_{\geq 0}.$$

  The coordinate chart used for integration/reporting is realization/implementation index data; it does not introduce an ontic metric geometry. $\gamma(d)$ and $w(d)$ are evaluated at non-integer d by the fixed shell interpolation convention defined in the Introduction.
- **Flux-Integral Kernel and Shell Field:** Relational gravity is generated by a single radial kernel $w(d)$ and an associated shell field $\gamma(d)$ defined on Chebyshev shells. For a unit Bell Source at the origin the shell multiplicities are

$$N_0 = 1, \qquad N_d = 24d^2 + 2 \quad (d \geq 1),$$

  and the enclosed flux is constant for all shells in this unit-source construction, $\mathcal{F}(d) = 1$ for all $d \geq 0$. (Here $\mathcal{F}$ denotes enclosed flux and is distinct from the DO state descriptor Φ.) A fixed flux parameter χ_{flux} defines the shell field

$$\gamma(d) \;=\; -\chi_{\text{flux}} \frac{\mathcal{F}(d)}{N_d} \;=\; -\chi_{\text{flux}} \frac{1}{N_d}, \qquad d \geq 1, \qquad \gamma(0) = 0,$$

  and the kernel weights are obtained by the discrete inward integral in shell index

$$w(d) - w(d{+}1) \;=\; -2\,\gamma(d{+}1), \qquad w(R_{\max}) = 0,$$

  for $R_{\max} = 200$ (a 401^3 diagnostic index chart). The operative interaction magnitude in the hard-mass limit is supplied by $\gamma(d_{ij})$; $w(d)$ enters only through the potential energy used in post-update auditing.
- **Discrete Direction (Manhattan–Fraction):** To preserve background independence, the pairwise interaction direction is defined combinatorially from the Bell network rather than from a Euclidean unit vector. Let

$$\Delta\mathbf{r}_{ij} = (\Delta x, \Delta y, \Delta z) = \mathbf{r}_i - \mathbf{r}_j, \qquad r_1 = |\Delta x| + |\Delta y| + |\Delta z|.$$

  The Manhattan–fraction direction is

$$\hat{\mathbf{u}}_{ij} = \frac{1}{r_1}\Big(\operatorname{sgn}(\Delta x)\,|\Delta x|,\ \operatorname{sgn}(\Delta y)\,|\Delta y|,\ \operatorname{sgn}(\Delta z)\,|\Delta z|\Big), \qquad r_1 > 0,$$

  with components set to 0 when the corresponding $\Delta\alpha = 0$.
- **Hard-mass ontology:** Each macroscopic body is realized as a single, strongly localized Bell Energy Field supported on one Bell Sphere at each discrete time slice of the 4D Bell-Sphere network; no dispersive spreading of the field is permitted. Any reported "radial texture" refers to bounded diagnostic scatter extracted from the discrete trajectory, not to physical delocalization of the hard-mass support.
- **DO force law and invariants:** The pair force between bodies i and j is

$$\mathbf{F}_{ij}(t) \;=\; \chi_E\, m_i m_j\, \gamma\big(d_{ij}(t)\big)\, \hat{\mathbf{u}}_{ij}(t),$$

 doi:10.20944/preprints202405.2028.v8

with pairwise reciprocity $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ at each update. Conservation is audited using the DO invariant

$$I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}},$$

where U is computed from $w(d)$ and W_{closure} is the accumulated discrete work–potential mismatch computed deterministically from the same DO forces and the same kernel-defined potential along the executed discrete trajectory. The closure ledger enters with unit weight by definition and has no free coefficient. I_{DO} is computed post-update from the realized trajectory, the kernel-defined potential $U(w)$, and the work-closure ledger W_{closure}; it has no feedback path into the dynamics and is not referenced by any position, velocity, force, kernel, or parameter update. The naive diagnostic $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ is reported as a texture indicator rather than as a conservation gate (Appendix C artifacts may print E_{kin} and $E_{\mathrm{kin}} + U$ using the legacy console labels `K` and `K+U`).

- **Time evolution:** A 4th-order Forest–Ruth symplectic update is used as a fixed numerical realization of the hard-mass classical-limit update law of Appendix A (Sec. 2.4). The 1-year baseline uses a fixed step budget calibrated to the Earth–Moon analog orbit; the long-horizon run is reported over 1000 completed orbital cycles under the same operator and update rule.

All ellipse fits, perihelion angles, and plotted trajectories are evaluated for reporting in a floating-point diagnostic chart of the integer-indexed Bell-Sphere network. The DO update itself uses only d_{ij}, $\gamma(d_{ij})$, and $\hat{\mathbf{u}}_{ij}$; no diagnostic quantity is fed back into the evolution.

Appendix B.2.3.2. Appendix B.3.2. Two-Body Python Implementation

The executable script, console transcript, and associated figures for this validation are provided in Appendix C.

Appendix B.2.3.3. Appendix B.3.3. Validation III–A1: Earth–Moon two-body baseline (1-year)

The 1-year Earth–Moon DO-native simulation confirms that the model's flux-integral kernel and Manhattan-direction operator produce a stable, mildly elliptical hard-mass orbit with tight control of the DO Invariant, angular momentum, and barycenter. Here "calibrated-initial-condition" denotes a one-time pre-run selection of launch conditions (e.g., launch radius and tangential launch speed) to place the system in the intended Earth–Moon dynamical band; no operator parameters are tuned and no mid-run adjustments occur.

Table A1. Per-body orbital metrics for the calibrated-initial-condition two-body system (1-year run, DO-native flux operator).

Body	Steps	$\Delta\theta$ (deg)	r_{mean} (hops)	r_{std} (hops)
Moon	443850	360.000208	28.249416	1.208257
Earth	443850	360.000208	0.347484	0.014862

Table A2. Global conservation metrics for the calibrated-initial-condition two-body system (1-year run, DO-native flux operator).

Metric	Value
Max barycenter drift (hops)	2.402499×10^{-13}
$\delta I_{\mathrm{DO,net}} = \dfrac{I_{\mathrm{DO}}(T) - I_{\mathrm{DO}}(0)}{\lvert I_{\mathrm{DO}}(0)\rvert}$	5.880×10^{-13}
$\mathrm{RMS}(\delta I_{\mathrm{DO}})$	4.128×10^{-12}
$\delta E_{\mathrm{naive,net}} = \dfrac{(E_{\mathrm{kin}}+U)(T) - (E_{\mathrm{kin}}+U)(0)}{\lvert (E_{\mathrm{kin}}+U)(0)\rvert}$	2.378×10^{-6}
$\mathrm{RMS}(\delta E_{\mathrm{naive}})$	1.960×10^{-1}
$\mathrm{RMS}(\delta L_z)$, $\delta L_z = \dfrac{L_z - L_{z,0}}{\lvert L_{z,0}\rvert}$	3.069×10^{-14}

 doi:10.20944/preprints202405.2028.v8

Orbital phase defect is assessed using the unwrapped polar angle of the relative separation in the orbital plane. The total phase advance over the 1-year track is

$$\Delta\theta \;=\; 360.000208^\circ,$$

corresponding to a defect error of 2.08×10^{-4} degrees (≈ 0.75 arcsec) over one full radial cycle.

The orbit is mildly elliptical. A global conic fit to the full 1-year relative track yields

$$e_{\mathrm{fit}} \approx 0.054875, \qquad a_{\mathrm{fit}} \approx 28.545052,$$

while a radius-extrema estimate over the same track gives

$$e_{r\,\mathrm{min}} = \frac{r_{\mathrm{max}} - r_{\mathrm{min}}}{r_{\mathrm{max}} + r_{\mathrm{min}}} \approx 0.061861, \qquad r_{\mathrm{min}} \approx 26.638620, \qquad r_{\mathrm{max}} \approx 30.151742.$$

Because the eccentricity is small, apsidal angles are diagnostically ill-conditioned compared to the phase, radial, and invariant gates; the primary stability checks are therefore $\Delta\theta$, barycenter drift, I_{DO}, and L_z.

The invariants confirm DO-lawful stability at numerical precision. The barycenter drift remains at the 10^{-13} hop level, and the angular momentum control is at the 10^{-14} level (Table A2). Energy conservation is correctly assessed using the DO invariant $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$: its net drift is 5.88×10^{-13} with RMS fluctuations 4.13×10^{-12}. By contrast, the naive diagnostic $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ explores a stationary texture band (RMS $\approx 1.96 \times 10^{-1}$) while retaining negligible net offset, reflecting reversible exchanges induced by the discrete shell representation and the work-closure accounting rather than any secular instability.

Appendix B.2.3.4. Appendix B.3.4. Validation III–A1: Visual Confirmation

A visual confirmation of the 1-year two-body dynamics is provided by plotting the full trajectory of the Earth–Moon analog and the barycenter on the 401^3 Chebyshev index chart. The mildly elliptical Moon orbit and the small counter-orbit of the Earth about the barycenter are visible, with no discernible precession at the plotted scale over the 1-year interval.

 doi:10.20944/preprints202405.2028.v8

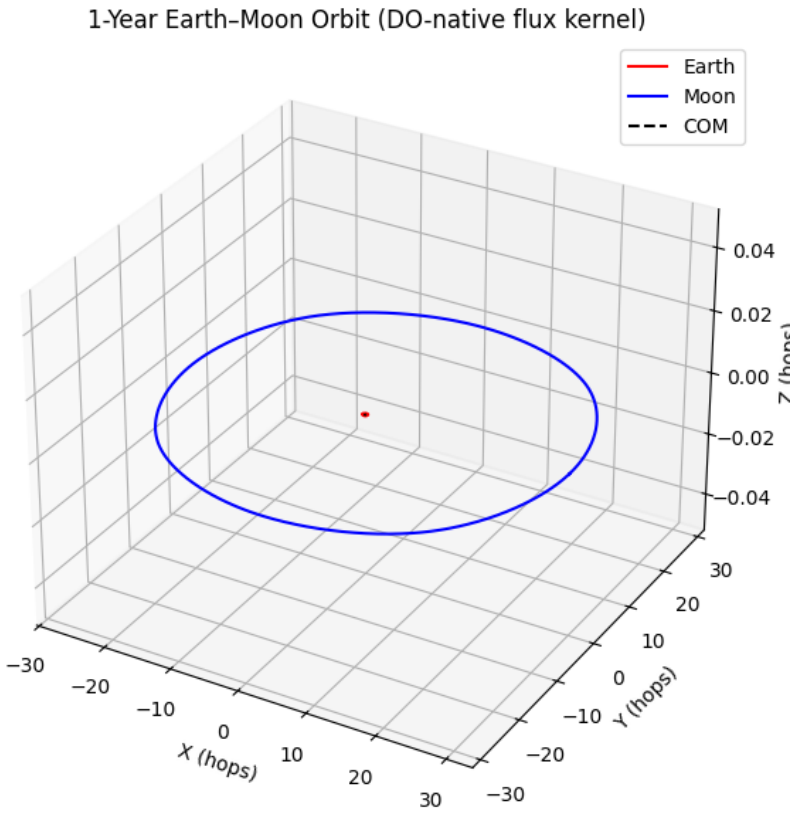


Figure A1. Trajectory of the 1-year DO two-body system (Earth–Moon analog).

Appendix B.2.3.5. Appendix B.3.5. Validation III–A2: Earth–Moon two-body long horizon (1000 orbits)

To probe extreme long-horizon robustness of the same DO two-body operator in the mildly elliptical regime, the Earth–Moon analog is evolved for 1000 complete orbital cycles in a fixed tilted orbital plane. The operator is unchanged: the same flux-defined shell kernel $w(d)$, shell field $\gamma(d)$, coupling χ_E, Manhattan–fraction direction rule, and Forest–Ruth update with DO invariant auditing $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$ are used throughout. The run length is set by the measured steps-per-orbit value from the calibrated tilted configuration, and the integration is executed for five extra orbits for safety while diagnostics are reported over the first 1000 complete orbits.

Run length and sampling. With $dt = 3 \times 10^{-3}$, launch radius $D_0 = 30$, and fixed inclination $i = 5.145°$, the calibrated orbit period is

$$N_{\mathrm{steps/orbit}} = 1{,}499{,}675,$$

so that the full integration length is

$$N_{\mathrm{steps}}^{(\mathrm{executed})} = (1000+5)\, N_{\mathrm{steps/orbit}} = 1{,}507{,}173{,}375, \qquad N_{\mathrm{steps}}^{(\mathrm{analyzed})} = 1000\, N_{\mathrm{steps/orbit}} = 1{,}499{,}675{,}000.$$

To reduce aliasing in per-orbit diagnostics, the sampling stride is chosen so that each orbit window contains ≥ 8000 sampled points, and periapsis/apoapsis are refined by a local three-point quadratic interpolation in $r(t)$.

Orbit segmentation and plane confinement. Define the relative vector $\mathbf{R}(t) = \mathbf{r}_M(t) - \mathbf{r}_E(t)$ and its orbital-plane components by projection onto the fixed tilted basis $(\mathbf{e}_{x'}, \mathbf{e}_{y'}, \mathbf{e}_{z'})$:

$$x'(t) = \mathbf{R}(t) \cdot \mathbf{e}_{x'}, \qquad y'(t) = \mathbf{R}(t) \cdot \mathbf{e}_{y'}, \qquad z'(t) = \mathbf{R}(t) \cdot \mathbf{e}_{z'}.$$

Orbit boundaries are defined by unwrapped 2π crossings of the relative orbital-plane angle

$$\phi(t) = \mathrm{atan2}\big(y'(t), x'(t)\big),$$

 doi:10.20944/preprints202405.2028.v8

with linear interpolation used to determine boundary times and indices. Exactly 1000 complete orbits are identified for analysis. The out-of-plane residual is quantified per orbit by

$$R_{z,\mathrm{RMS}} = \mathrm{RMS}\big(z'(t)\big),$$

and remains at the 10^{-11} level across the full horizon, confirming confinement to the fixed tilted plane to numerical precision. The inclination and node diagnostics (below) remain constant to numerical precision across the same horizon.

Orbit clock (steps per orbit). Orbit boundaries define a measured steps-per-orbit distribution over the 1000 analyzed cycles:

$$N_{\mathrm{steps/orbit}}: \quad \mu \approx 1{,}497{,}696.780,\ \ \sigma \approx 946.737,\ \ N_{\min} = 1{,}496{,}221,\ \ N_{\max} = 1{,}499{,}847.$$

This confirms a stable orbit clock over the full horizon.

Global invariant control over 1000 orbits. Define the relative diagnostics

$$\begin{aligned}
\delta I_{\mathrm{DO}}(t) &= \frac{I_{\mathrm{DO}}(t) - I_{\mathrm{DO}}(0)}{|I_{\mathrm{DO}}(0)|}, \\
\delta E_{\mathrm{naive}}(t) &= \frac{(E_{\mathrm{kin}} + U)(t) - (E_{\mathrm{kin}} + U)(0)}{|(E_{\mathrm{kin}} + U)(0)|}, \\
\delta |L|(t) &= \frac{|L|(t) - |L|(0)}{|L|(0)|}.
\end{aligned}$$

The inclination and node diagnostics are defined from the total angular momentum $\mathbf{L}(t)$ by

$$i_L(t) = \arccos\!\left(\frac{L_z(t)}{|L(t)|}\right), \qquad \Omega_L(t) = \mathrm{atan2}\big(L_x(t), -L_y(t)\big),$$

reported in degrees. Over the first 1000 complete orbits, the measured global diagnostics are:

Table A3. Global diagnostics for the 1000-orbit two-body run (tilted Earth–Moon analog, first 1000 orbits).

Metric	Value
$\delta I_{\mathrm{DO,net}}$	2.245×10^{-12}
$\mathrm{RMS}(\delta I_{\mathrm{DO}})$	2.242×10^{-12}
$\delta E_{\mathrm{naive,net}}$	-6.041×10^{-2}
$\mathrm{RMS}(\delta E_{\mathrm{naive}})$	1.924×10^{-1}
$\delta \lvert L\rvert_{\mathrm{net}}$	-6.467×10^{-12}
$\mathrm{RMS}(\delta \lvert L\rvert)$	3.589×10^{-12}
$\max_t \lVert \mathbf{R}_{\mathrm{bary}}(t) - \mathbf{R}_{\mathrm{bary}}(0)\rVert$ (hops)	3.296×10^{-9}
$\mathrm{RMS}\big(\lVert \mathbf{R}_{\mathrm{bary}}(t) - \mathbf{R}_{\mathrm{bary}}(0)\rVert\big)$ (hops)	2.065×10^{-9}
Inclination mean/std (deg)	5.145000/0.000000
Node angle start/end (deg)	$0.000000 \to -0.000000$

The relevant conservation audit is the flatness of I_{DO}, $|L|$, and the barycenter. The naive diagnostic $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ explores a stationary texture band under the same unchanged update.

Shape evolution and apsidal diagnostics. Per orbit, the radial extrema $r_{\min}, r_{\max}$ are extracted using quadratic sub-sample interpolation, and

$$e_{r\,\min} = \frac{r_{\max} - r_{\min}}{r_{\max} + r_{\min}}, \qquad a_r = \frac{r_{\min} + r_{\max}}{2}.$$

Selected values (first and last analyzed orbit) are:

 doi:10.20944/preprints202405.2028.v8

Table A4. Selected per-orbit shape diagnostics (tilted Earth–Moon analog; quadratic-interpolated extrema).

Orbit	$e_{r\,\min}$	$r_{\min}$	$r_{\max}$	a_r
1	0.05490	26.8796	30.0024	28.4410
1000	0.06094	26.6519	30.1110	28.3815

Periapsis-angle diagnostics (ω and $\Delta\omega$) are not treated as primary physical outputs in this Earth–Moon band. The orbit is only mildly elliptical, so the periapsis direction is weakly defined and becomes highly sensitive to discrete sampling, window boundaries, and the specific periapsis localization estimator on the Chebyshev/Manhattan update. Consequently, $\omega/\Delta\omega$ variations seen in long-horizon readouts are interpreted as estimator/geometry artifacts in a near-circular regime rather than as a robust secular effect of the operator. The primary stability gates are $\Delta\theta$ (orbit counting), barycenter drift, I_{DO}, and $|L|$.

Appendix B.2.3.6. Appendix B.3.6. Conclusion

Across the 1-year and extended 1000-orbit ("1000-orbit") horizons, the same fixed DO two-body operator yields a stable, mildly elliptical hard-mass orbit with tight invariant control and no evidence of secular instability. Phase closure over the 1-year track is essentially exact ($\Delta\theta = 360.000208^\circ$), while the long-horizon run identifies 1000 complete orbits by deterministic unwrapped 2π crossings of the orbital-plane phase. The barycenter remains fixed to 10^{-13} hops (1-year) and 10^{-9} hops (1000-orbit), and angular-momentum invariance remains at the 10^{-14} (1-year) to 10^{-12} (1000-orbit) level.

The model succeeds for three coupled reasons: (i) the interaction magnitude is supplied by a single shell field $\gamma(d)$ derived from the fixed radial kernel $w(d)$ in Chebyshev shell index; (ii) pairwise reciprocity $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ is enforced at each update using the background-independent Manhattan–fraction direction rule, yielding barycenter control; and (iii) energy conservation is audited using the DO invariant $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$, where the work-closure term accounts for the discrete work–potential mismatch accumulated along the executed discrete trajectory so that I_{DO} remains flat at the 10^{-12} level even while the naive diagnostic $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ explores a stationary texture band (RMS ~ 0.19) with no evidence of broadening over the full horizon. In the extended run, the orbital plane remains rigid (inclination and node constant to numerical precision); apsidal diagnostics remain secondary in this mildly elliptical band and exhibit estimator sensitivity under discrete sampling.

Table A5. Comparison of 1-year and 1000-orbit two-body runs (Earth–Moon analog, DO two-body operator).

Horizon	dt	Steps	Orbits	$\Delta\theta$ (deg)	Max barycenter drift (hops)	$\delta I_{\mathrm{DO,net}}$	RMS(δI_{DO})	RMS(δL_z)	RMS$(\delta\lvert L\rvert)$
1-year (1 orbit)	1.0×10^{-2}	443850	1	360.000208	2.402499×10^{-13}	5.880×10^{-13}	4.128×10^{-12}	3.069×10^{-14}	—
1000-orbit horizon	3.0×10^{-3}	1507173375	1000	360° (by 2π-crossing window definition)	3.296×10^{-9}	2.245×10^{-12}	2.242×10^{-12}	—	3.589×10^{-12}

Appendix B.4. Validation III–B: Three-body orbital stability (1-year and 1000-orbit DO operator tests)

Appendix B.2.4.1. Appendix B.4.1. Objective and physical setup

Section III–B extends the DO hard-mass orbital validation of Section III–A to a genuinely three-body configuration. The aim is to test whether the same fixed DO relational-gravity law supports long-horizon, multi-orbit stability when three macroscopic bodies mutually interact through the discrete Bell-Sphere network, with no re-tuning, no mid-run adjustments, and no auxiliary structure beyond the DO operator, the UEE classical limit, and invariant auditing.

The configuration is a low-energy hierarchical system: a central body ("Planet X") with mass

$$m_X = 81.297,$$

and two identical satellites ("inner moon" and "outer moon") with masses

$$m_Y = m_Z = 1.0.$$

 doi:10.20944/preprints202405.2028.v8

At the initial time slice, the three hard-mass Bell Fields are placed at fixed Chebyshev launch radii

$$D_Y = 30, \qquad D_Z = 40,$$

with the barycenter shifted to the origin and the orbital plane initialized at a fixed inclination

$$i = 5.145^\circ$$

by selecting a common in-plane tangential launch direction. Initial velocities are chosen so that (i) each Planet X–moon pair satisfies the two-body balance implied by the same DO shell field $\gamma(d)$ evaluated at its launch radius, up to fixed multiplicative launch factors V_{IN} and V_{OUT}, and (ii) the total linear momentum vanishes so that the three-body barycenter remains fixed.

This configuration is chosen to be hierarchically bound and well separated, but it is not constructed to realize any special analytic stability family (e.g. Lagrange-point Trojans, horseshoes, orbit-swaps, or an exact mean-motion resonance). It is a generic hierarchical three-body setup evolved under a single fixed DO operator and update.

All three-body surveys reported below (1-year and 1000-orbit) use exactly the same DO operator and parameters:

- Discrete 4D spacetime: hard-mass bodies move on a finite Chebyshev box $\{-R_{\max}, \ldots, R_{\max}\}^3$ with $R_{\max} = 200$ in each spatial slice, chosen so that the boundary is never sampled by either orbit.
- Relational kernel on $S^{(4D)}$: a flux-defined shell kernel $w(d)$ and shell field $\gamma(d)$ constructed from enclosed flux on Chebyshev shells. Writing the shell multiplicity as

  $$N_0 = 1, \qquad N_d = 24d^2 + 2 \quad (d \geq 1),$$

  the enclosed flux $\mathcal{F}(d)$ is defined on shells, $\gamma(d)$ is obtained as a shell field from $\mathcal{F}(d)$, and $w(d)$ is obtained by discrete inward integration so that the shell-gradient used in dynamics is supplied by $\gamma(d)$ on the integer shell index. (Here $\mathcal{F}$ denotes enclosed flux and is distinct from the DO state descriptor Φ.)
- Relational coupling: $\chi_E = 0.00057032984$ (fixed).
- Time evolution: the same fourth-order Forest–Ruth update of the barycenterclassical limit of the UEE with time step $\Delta t = 3 \times 10^{-3}$, with energy auditing performed using the DO invariant $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$.

In every three-body run the DO pair interaction between bodies i and j is a central shell-gradient force of the form

$$\mathbf{F}_{ij}(t) \;=\; \chi_E \, m_i m_j \, \gamma\big(d_{ij}(t)\big) \, \hat{\mathbf{u}}_{ij}(t), \tag{A45}$$

where $d_{ij}(t)$ is the Chebyshev shell index of the separation,

$$d_{ij}(t) = \max\big\{|x_i - x_j|,\ |y_i - y_j|,\ |z_i - z_j|\big\},$$

and $\hat{\mathbf{u}}_{ij}(t)$ is the background-independent Manhattan–fraction direction defined by the instantaneous signed separations,

$$\Delta\mathbf{r}_{ij}(t) = \mathbf{r}_i(t) - \mathbf{r}_j(t) = (\Delta x, \Delta y, \Delta z), \qquad r_1(t) = |\Delta x| + |\Delta y| + |\Delta z|,$$

$$\hat{\mathbf{u}}_{ij}(t) = \frac{1}{r_1(t)} \Big(\mathrm{sgn}(\Delta x)\,|\Delta x|,\ \mathrm{sgn}(\Delta y)\,|\Delta y|,\ \mathrm{sgn}(\Delta z)\,|\Delta z|\Big), \quad r_1(t) > 0.$$

This construction preserves background independence while enforcing exact pairwise reciprocity ($\mathbf{F}_{ij} = -\mathbf{F}_{ji}$) at each discrete update. The orbital-plane inclination is set entirely by the launch geometry; the force law and kernel remain unchanged for all horizons.

 doi:10.20944/preprints202405.2028.v8

Appendix B.2.4.2. Appendix B.4.2. Three-Body Python Implementation

The executable script, console transcript, and associated figures for this validation are provided in Appendix C.

Appendix B.2.4.3. Appendix B.4.3. Three-Body Quantitative Results & Analysis

Appendix B.4.3.1. Validation III–B1: 1-Year Three-Body Validation

The 1-year three-body validation tests the DO hard-mass gravitational operator in a hierarchical configuration with a central body (X) of mass $m_X = 81.297$ and two identical satellites (inner and outer moons) of mass $m_Y = m_Z = 1.0$. The initial launch radii are fixed at

$$D_Y = 30, \qquad D_Z = 40,$$

with a fixed orbital-plane inclination $i = 5.145^\circ$. The operator is held fixed: the same flux-defined shell kernel $w(d)$, shell field $\gamma(d)$, coupling χ_E, Manhattan–fraction direction rule, and Forest–Ruth update with DO invariant auditing $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$ are used for all horizons.

Orbit windowing and sampling. The analysis is performed on an *inner-orbit window* defined by unwrapped 2π crossings of the inner relative orbital-plane angle

$$\phi_Y(t) = \mathrm{atan2}\big(y'_Y(t), x'_Y(t)\big),$$

where the orbital-plane components are defined by projection onto the fixed tilted basis

$$x'_Y(t) = \big(\mathbf{r}_Y(t) - \mathbf{r}_X(t)\big) \cdot \mathbf{e}_{x'}, \qquad y'_Y(t) = \big(\mathbf{r}_Y(t) - \mathbf{r}_X(t)\big) \cdot \mathbf{e}_{y'}, \qquad z'_Y(t) = \big(\mathbf{r}_Y(t) - \mathbf{r}_X(t)\big) \cdot \mathbf{e}_{z'}.$$

Linear interpolation is used to determine the boundary time and index. Within each window, periapsis and apoapsis are refined by a local three-point quadratic interpolation of the radial record $r_Y(t)$. The shape proxy is

$$e_{r\,\mathrm{min}} \equiv \frac{r_{\mathrm{max}} - r_{\mathrm{min}}}{r_{\mathrm{max}} + r_{\mathrm{min}}},$$

with $r_{\mathrm{min}}, r_{\mathrm{max}}$ taken from the quadratic-refined extrema within the orbit window. To suppress aliasing in per-window diagnostics, the trajectory is sampled densely in time; in this 1-year validation the sampling is set so that the inner-orbit window contains at least 12,000 samples.

Deterministic initialization (no operator tuning). The inner and outer launch speeds are set by fixed multiplicative factors on the DO circular-speed estimate obtained from the same shell field $\gamma(d)$ evaluated at the launch radii. The inner factor is held fixed at $V_{\mathrm{IN}} = 0.939199$. The outer factor is fixed by a one-time initial-condition calibration: a coarse-to-fine deterministic sweep is performed pre-run to match a target outer-orbit shape proxy $e_{r\,\mathrm{min}}$ evaluated over the inner-orbit window. The best value used for all subsequent long-horizon runs is

$$V_{\mathrm{OUT}} = 1.030800000,$$

yielding an outer-orbit shape proxy $e_{r\,\mathrm{min}} = 0.079816$ with absolute error 1.841×10^{-4} relative to the target. No re-selection of V_{OUT} is performed as a function of horizon length.

Phase closure and shape diagnostics. Over the 1-year inner-orbit window, the inner orbit closes to within a small phase overshoot:

$$\Delta\theta_Y = 360.005909^\circ,$$

while the outer orbit remains bounded over the same window and satisfies the stated $e_{r\,\mathrm{min}}$ target at the reported tolerance. The corresponding orbit-window diagnostics are summarized in Table A6.

 doi:10.20944/preprints202405.2028.v8

Table A6. Windowed diagnostics for the 1-year three-body validation (one complete inner-orbit window).

Quantity	Inner (Y about X)	Outer (Z about barycenter)
Phase advance $\Delta\theta$ (deg)	360.005909	—
Shape proxy $e_{r\,\min}$	0.075799	0.079816
$\lvert e_{r\,\min} - e_{r\,\min}^{\text{target}} \rvert$	—	1.841×10^{-4}

Invariant control over the 1-year window. Global invariants are evaluated over the same inner-orbit window. The DO Invariant remains flat at the 10^{-13} level and the barycenter remains fixed at the 10^{-13} hop level, while the total angular momentum magnitude remains controlled at the 10^{-13} level:

Table A7. Global conservation diagnostics for the 1-year three-body validation (one complete inner-orbit window).

Metric	Value
I_{DO} net drift	5.047×10^{-13}
I_{DO} RMS	3.996×10^{-13}
$\lvert L \rvert$ RMS	5.380×10^{-13}
Max barycenter drift (hops)	1.860×10^{-13}

Taken together, the 1-year three-body results show that the hierarchical configuration remains bounded and diagnostically stable under a single fixed DO operator. The windowed phase and shape diagnostics are satisfied with high sampling fidelity, the orbital plane remains fixed to numerical precision (inclination and node constant), and the conserved quantities remain flat at the stated levels under DO invariant auditing.

Appendix B.2.4.4. Appendix B.4.3.2. Validation III–B1: Visual Confirmation

A visual confirmation of the 1-year three-body dynamics is provided by plotting the full trajectories of Planet X, the inner moon, the outer moon, and the barycenter over the 1-year validation window on the Chebyshev index chart. The inner and outer moons remain bounded in a hierarchical configuration, while the barycenter remains essentially fixed and the central body follows a small counter-orbit driven by the mutual interactions.

 doi:10.20944/preprints202405.2028.v8

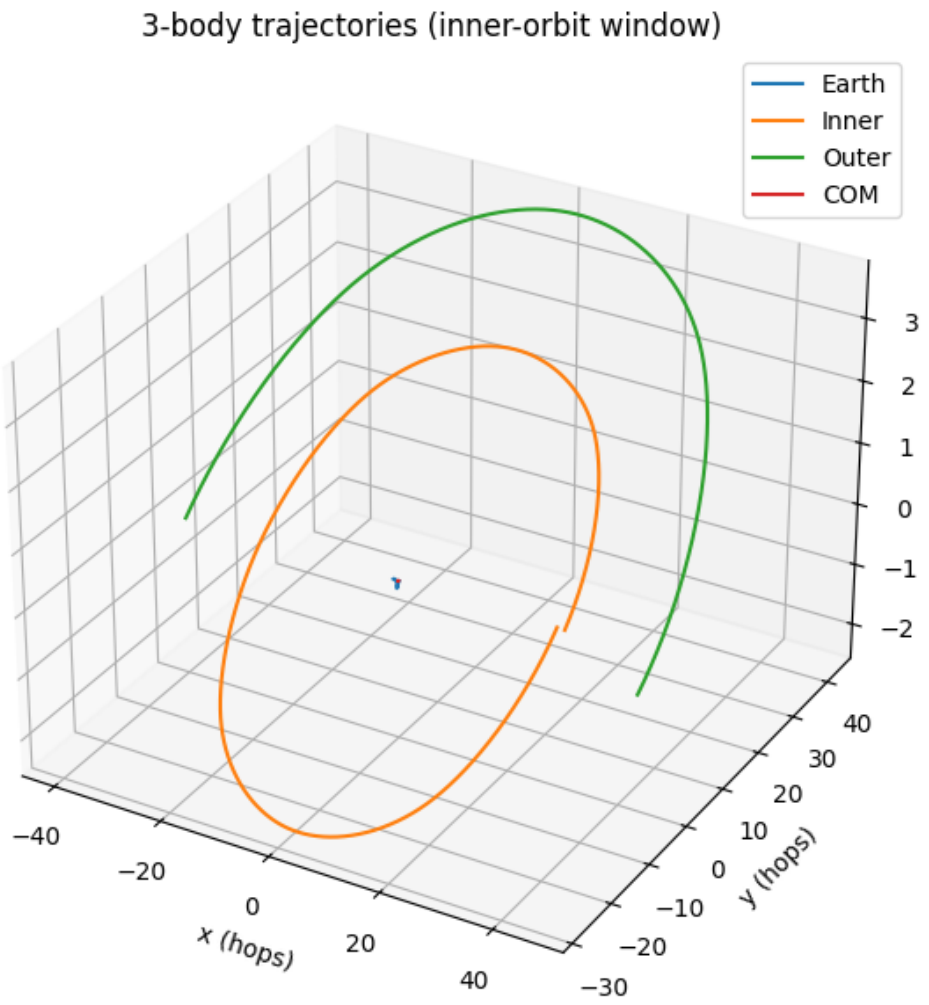


Figure A2. Three-body trajectories over the 1-year validation window under the DO three-body operator, showing the inner and outer moons, Planet X, and the barycenter.

Appendix B.2.4.5. Appendix B.4.3.3. Validation III–B2: three-body long horizon (1000 inner orbits)

This long-horizon validation evolves a fully coupled DO three-body system through 1000 complete *inner* orbital cycles under a single fixed, background-independent, flux-defined relational operator. The result is notable for a simple reason: over $\sim 1.5 \times 10^9$ discrete updates, the system remains bounded and the conserved quantities remain tightly controlled, despite the absence of any continuum potential, field PDE, Newtonian/EFE structure, or special analytic three-body family.

Run length and orbit counting. The integration uses

$$N_{\text{steps}} = 1{,}467{,}332{,}196, \qquad \Delta t = 3 \times 10^{-3},$$

with a fixed inclination $i = 5.145°$, launch radii $D_Y = 30$, $D_Z = 40$, and fixed launch factors $V_{\text{IN}} = 0.939199$, $V_{\text{OUT}} = 1.030800$. Inner-orbit windows are defined by unwrapped 2π crossings of the inner orbital-plane phase $\phi_Y(t) = \text{atan2}(y'_Y(t), x'_Y(t))$, with linear interpolation used to locate each boundary. Exactly 1000 complete inner orbits are identified for analysis.

The inner-orbit step count remains narrowly distributed:

$$N_{\text{steps/orbit}}: \quad \mu = 1{,}460{,}802.733, \ \ \sigma = 6{,}324.984, \ \ \min = 1{,}444{,}198, \ \ \max = 1{,}477{,}787.$$

This confirms a stable orbit clock over the entire horizon.

 doi:10.20944/preprints202405.2028.v8

What is being validated. This section validates three coupled properties of the DO N-body implementation under extreme horizon: (i) boundedness of the hierarchical three-body motion (no escape or plunge), (ii) long-horizon control of I_{DO}, total angular momentum magnitude $|L|$, and the barycenter, and (iii) stability of windowed orbit diagnostics (phase, shape proxies) under dense sampling and sub-sample extremum refinement.

Global invariant control over 1000 inner orbits. Define relative diagnostics

$$\begin{aligned} \delta I_{\mathrm{DO}}(t) &= \frac{I_{\mathrm{DO}}(t) - I_{\mathrm{DO}}(0)}{|I_{\mathrm{DO}}(0)|}, \\ \delta E_{\mathrm{naive}}(t) &= \frac{(E_{\mathrm{kin}} + U)(t) - (E_{\mathrm{kin}} + U)(0)}{|(E_{\mathrm{kin}} + U)(0)|}, \\ \delta |L|(t) &= \frac{|L|(t) - |L|(0)}{|L|(0)|}. \end{aligned}$$

Over the first 1000 complete inner orbits, the conserved quantities remain tightly bounded:

Table A8. Global diagnostics for the 1000-orbit three-body run (first 1000 complete inner orbits).

Metric	Value
I_{DO} net drift	3.981×10^{-12}
I_{DO} RMS	3.580×10^{-12}
$\|L\|$ net drift	5.104×10^{-11}
$\|L\|$ RMS	1.883×10^{-11}
barycenter drift max (hops)	8.063×10^{-9}
barycenter drift RMS (hops)	3.651×10^{-9}
barycenter drift max / $\langle r_{\mathrm{in}} \rangle$	2.883×10^{-10}
Inclination mean/std (deg)	5.145000/0.000000
Node angle start/end (deg)	$180.000000 \to 180.000000$

Energy conservation is correctly assessed by the DO invariant $I_{\mathrm{DO}} = E_{\mathrm{kin}} + U - W_{\mathrm{closure}}$, which remains flat at the 10^{-12} level over the entire horizon. The naive diagnostic $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ explores a stationary texture band (RMS $= 1.404 \times 10^{-1}$) and is not used as a conservation gate.

Where the residual errors originate. The remaining nonzero drifts have clear origins internal to the discrete realization:

- The barycenter and $|L|$ drifts arise from finite `float64` accumulation over $\sim 1.5 \times 10^9$ updates, not from any net injected momentum. The force law is pairwise reciprocal at each update ($\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ by construction), so the observed barycenter drift is the roundoff-limited residual of long-horizon summation.
- The separation between E_{naive} and I_{DO} is the discrete work-closure W_{closure}. The observed stationarity of δI_{DO} at $\sim 10^{-12}$ indicates that the work-closure accounting closes the discrete work–potential mismatch produced by the shell-index interaction and interpolation across shells.
- The large orbit-to-orbit variability in apsidal diagnostics originates from diagnostic conditioning: the inner orbit remains close to circular for extended segments (small e), making the major-axis direction ill-conditioned even when the underlying motion and invariants remain stable.

Outer orbit count over the same horizon. Over the same window used to collect the first 1000 complete inner orbits, the outer body executes a bounded barycentric orbit. Using the unwrapped barycentric polar angle of the outer position projected into the orbital plane:

 doi:10.20944/preprints202405.2028.v8

Table A9. Outer barycentric orbit count over the 1000-inner-orbit horizon.

Quantity	Value
Outer $\Delta\theta_{\text{full}}$ (deg)	−186,464.059185
Outer N_{orbits}	517.955720
Outer r_{mean} (hops)	43.128234
Outer r_{std} (hops)	1.607408

Inner-orbit shape stability (windowed extrema). Within each inner-orbit window, periapsis and apoapsis are refined by quadratic sub-sample interpolation of $r_Y(t)$, defining

$$e_{r\,\min} = \frac{r_{\max} - r_{\min}}{r_{\max} + r_{\min}}, \qquad a_r = \frac{r_{\min} + r_{\max}}{2}.$$

Across 1000 inner orbits:

$$e_{r\,\min}: \ \mu = 0.055389,\ \sigma = 0.015386,\ \text{first} = 0.075799,\ \text{last} = 0.046756.$$

Representative windows (early, mid, late) illustrate bounded modulation without drift:

Table A10. Representative windowed inner-orbit diagnostics over the 1000-orbit horizon.

Orbit	N_{steps}	$e_{r\,\min}$	$r_{\min}$	$r_{\max}$	a_r
1	1449933	0.07580	25.7725	30.0000	27.8863
1000	1454752	0.04676	26.4883	29.0867	27.7875

Apsidal diagnostics $(\omega, \Delta\omega)$ are computed using an ellipse-axis fit within each inner window with axis sign fixed toward periapsis and continuity unwrapped in π-steps. The pipeline is numerically stable over the full horizon (fit_ok $= 1000/1000$, no missing values). The resulting $\Delta\omega$ distribution is broad because the orbit is frequently close to circular, and ω is therefore not used as a primary stability gate in this low-eccentricity hierarchical regime.

Appendix B.5. Validation IV: Mercury–Sun Validation: Methodology

Appendix B.2.5.1. Appendix B.5.1. General Methodology and Principles

The fourth validation extends the two-body hard-mass programme of Validation III–A to a higher-energy, moderately elliptical "Mercury–Sun" analog configuration. As in the Earth–Moon tests, the underlying dynamics are governed by the classical, non-dispersive hard-mass limit of the Unified Evolution Equation (UEE) in Appendix A. The objective here is a stable, moderately eccentric two-body orbit in a higher-acceleration regime, used to probe long-horizon stability of the same discrete DO gravitational operator together with its associated secular geometric behaviour (including periapsis advance) under fixed rules. The macroscopic bodies are again treated as hard-mass Bell Fields: each "Sun" and "Mercury" is realized as a single, strongly localized Bell Energy Field supported on one Bell Sphere at each discrete time slice of the 4D Bell-Sphere network; no dispersive spreading is permitted. Any reported "radial texture" refers to bounded diagnostic scatter extracted from the discrete trajectory, not to physical delocalization of the hard-mass support.

The simulations are not intended to reproduce the measured Solar-System Mercury ephemeris with planetary perturbations, but to test internal consistency of the DO gravitational operator in a moderately eccentric two-body setting with strong central mass contrast. In direct analogy with Validation III, the analysis presents a primary 1-orbit baseline ("1-year" horizon) together with an extended 1000-orbit run ("1000-orbit" horizon) that probes long-horizon stability and secular behaviour under the same fixed discrete operator. The core principles, inherited from Appendix A but specialized to the flux-integral kernel used in this regime, are:

 doi:10.20944/preprints202405.2028.v8

- **Background Independence & Discrete Spacetime:** The model does not presume a pre-existing spacetime continuum. Hard-mass bodies propagate as support indices on the Bell-Sphere network, and separation is measured by the chart-evaluated Chebyshev separation

$$d_{ij} \;=\; \max\{|x_i - x_j|,\, |y_i - y_j|,\, |z_i - z_j|\} \in \mathbb{R}_{\geq 0}.$$

  The coordinate chart used for integration/reporting is realization/implementation index data; it does not introduce an ontic metric geometry. $\gamma(d)$ and $w(d)$ are evaluated at non-integer d by the fixed shell interpolation convention defined in the Introduction.
- **Flux-Integral Kernel and Shell Field:** Relational gravity is generated by a single radial kernel $w(d)$ and an associated shell field $\gamma(d)$ defined on Chebyshev shells. For a single unit Bell Source at the origin the shell multiplicities are

$$N_0 \;=\; 1, \qquad N_d \;=\; (2d+1)^3 - (2d-1)^3 \;=\; 24d^2 + 2, \quad d \geq 1,$$

  and the enclosed flux in this unit-source construction is constant on all shells,

$$\mathcal{F}(d) \;=\; \sum_{k \leq d} N_k \rho_k \;=\; 1 \quad \text{for all } d \geq 0,$$

  (where $\mathcal{F}$ denotes enclosed flux and is distinct from the DO state descriptor Φ), and $\rho_0 = 1$ and $\rho_{k>0} = 0$. A fixed flux parameter χ_{flux} defines the shell field

$$\gamma(d) \;=\; -\,\chi_{\text{flux}}\,\frac{\mathcal{F}(d)}{N_d} \;=\; -\,\chi_{\text{flux}}\,\frac{1}{N_d}, \qquad d \geq 1, \qquad \gamma(0) = 0,$$

  and the kernel weights $w(d)$ are obtained by the discrete inward integral in shell index

$$w(d) - w(d{+}1) \;=\; -\,2\,\gamma(d{+}1), \qquad w(R_{\max}) = 0,$$

  for a chosen maximal radius $R_{\max}$. In the hard-mass classical limit the pair force between bodies i and j at Chebyshev distance d_{ij} is constructed from the shell field $\gamma(d_{ij})$,

$$\mathbf{F}_{ij} \;=\; \chi_E\, m_i m_j\, \gamma(d_{ij})\, \hat{\mathbf{u}}_{ij},$$

  where χ_E is a fixed coupling and $\hat{\mathbf{u}}_{ij}$ is the background-free discrete direction defined below. This implements the Appendix A prescription that the operative interaction magnitude in the hard-mass limit is supplied by $\gamma(d)$; the kernel $w(d)$ enters only through the potential energy used in post-update auditing via the DO invariant $I_{\text{DO}} = E_{\text{kin}} + U - W_{\text{closure}}$.
- **Discrete Direction (Manhattan–Fraction):** To preserve background independence, the force direction is defined combinatorially from the Bell network rather than from a Euclidean unit vector. Let

$$\Delta\mathbf{r}_{ij} \;=\; \mathbf{r}_i - \mathbf{r}_j \;=\; (\Delta x, \Delta y, \Delta z), \qquad r_1 \;=\; |\Delta x| + |\Delta y| + |\Delta z|.$$

  The Manhattan–fraction direction is

$$\hat{\mathbf{u}}_{ij} = \frac{1}{r_1}\Big(\operatorname{sgn}(\Delta x)\,|\Delta x|,\, \operatorname{sgn}(\Delta y)\,|\Delta y|,\, \operatorname{sgn}(\Delta z)\,|\Delta z|\Big), \qquad r_1 > 0,$$

  with components set to 0 when the corresponding $\Delta\alpha = 0$. All ellipse fits, periapsis angles, and plots are evaluated only for reporting in a floating-point diagnostic chart of the integer-indexed Bell-Sphere network; the DO update uses only d_{ij}, $\gamma(d_{ij})$, and $\hat{\mathbf{u}}_{ij}$.

In this setting the numerical programme establishes a stable Mercury–Sun analog baseline and then interrogates its long-horizon behaviour under the unchanged operator.

 doi:10.20944/preprints202405.2028.v8

- **Physical Ontology:** In the hard-mass limit, the simulation tracks the trajectories of the centers-of-mass of two Bell Fields representing a "Sun" and a "Mercury," each field localized on a single Bell Sphere at each discrete time slice of the 4D Bell-Sphere network. For diagnostics the Bell-Sphere network is indexed by a $401 \times 401 \times 401$ chart ($R_{\max} = 200$); this chart is used for hop counting and plotting and does not introduce a background continuum.
- **Parameter selection (pre-run; no operator tuning):** Within the flux-integral kernel class $\mathcal{W}$, a representative $w(d)$ is fixed by choosing $\chi_{\text{flux}} = 24.0$ and the gravitational coupling

$$\chi_E \;=\; \chi_{E,\text{base}} \times 10^2, \qquad \chi_{E,\text{base}} = 5.7032984 \times 10^{-4},$$

with $R_{\max} = 200$ (so $N_{\text{grid}} = 401$). The "Sun" and "Mercury" masses are chosen as

$$m_1 = 1000.0, \qquad m_2 = 1.0,$$

with initial Chebyshev separation $d_0 = 60$ ("Mercury band") and time step $dt = 10^{-4}$. A single tangential speed factor $v_{\text{factor}} = 1.107167$ is fixed once by a one-time initial-condition calibration matching the 1-orbit baseline to the target eccentricity $e_{\text{target}} = 0.20563$. This factor, together with $(\chi_E, \chi_{\text{flux}}, R_{\max}, dt)$, is held fixed for the 1-orbit and 1000-orbit horizons. No mid-run adjustments or re-tunings are permitted.
- **Numerical Integrator and post-update auditing:** A 4th-order Forest–Ruth symplectic integrator is used to advance the hard-mass DO system under the discrete gravitational operator. At each substep the code evaluates the DO-native force $\mathbf{F}_{ij} = \chi_E m_i m_j \gamma(d_{ij}) \hat{\mathbf{u}}_{ij}$ and updates the velocities with the fixed Forest–Ruth kick–drift sequence. The conserved DO invariant $I_{\text{DO}} = E_{\text{kin}} + U - W_{\text{closure}}$ is monitored on every sampled time slice, with the closure ledger W_{closure} computed deterministically from the discrete work–potential mismatch accumulated along the executed discrete trajectory using the same DO forces and the same kernel-defined potential $U(w(d))$. The closure ledger enters with unit weight by definition and has no free coefficient. I_{DO} is computed post-update and is not referenced by any position, velocity, force, kernel, or parameter update.

Convention. Validation IV uses the flux-integral shell field

$$\gamma(d) \;=\; -\chi_{\text{flux}} \frac{\mathcal{F}(d)}{N_d}, \qquad w(d) - w(d{+}1) \;=\; -2\,\gamma(d{+}1),$$

as the discrete representation of the radial gradient of $w(d)$ in the hard-mass classical limit, and uses the Manhattan–fraction direction vector $\hat{\mathbf{u}}_{ij}$ in place of a continuum unit vector to enforce background independence on the Chebyshev shell network.

Appendix B.2.5.2. Appendix B.5.2. Mercury–Sun Python Implementation

The executable script, console transcript, and associated figures for Validation IV1 are provided in Appendix C, Section C.7. The corresponding artifacts for Validation IV2 are provided in Appendix C, Section C.8.

Appendix B.2.5.3. Appendix B.5.3. Validation IV1: Mercury–Sun baseline (1-year)

The 1-year Mercury–Sun simulation confirms that the flux-integral kernel and Manhattan-direction operator described in Section B.5 produce a stable, moderately elliptical hard-mass orbit with tight conservation of the DO Invariant and angular momentum. The run uses the fixed parameters

$$D_0 = 60, \qquad V_{\text{factor}} = 1.107167, \qquad \text{STEPS\_YEAR} = 5{,}797{,}311, \qquad dt = 10^{-4},$$

and treats the full 1-year track as a single orbit for ellipse fitting and diagnostic evaluation.

 doi:10.20944/preprints202405.2028.v8

Orbital geometry. The fitted orbital parameters for the 1-year Mercury–Sun track are summarized in Table A11. The global conic fit in the x–y diagnostic chart yields a semi-major axis

$$a_{\mathrm{fit}} \approx 78.217892$$

and eccentricity

$$e_{\mathrm{fit}} = 0.205634,$$

in close agreement with the target $e_{\mathrm{target}} = 0.205630$:

$$\left|e_{\mathrm{fit}} - e_{\mathrm{target}}\right| \approx 4.29 \times 10^{-6}.$$

An independent radial estimate based on the extrema of the Euclidean separation in the diagnostic chart,

$$e_{\mathrm{rmin}} = \frac{r_{\mathrm{max}} - r_{\mathrm{min}}}{r_{\mathrm{max}} + r_{\mathrm{min}}} = 0.228116,$$

confirms that the orbit is robustly elliptical with moderate eccentricity. The DO update law itself uses only the Chebyshev shell index d_{ij}, the shell field $\gamma(d_{ij})$, and the Manhattan direction $\hat{\mathbf{u}}_{ij}$; Euclidean fits are reported for characterization only.

Table A11. 1-year Mercury–Sun orbital fit and control parameters (DO operator, $D_0 = 60$).

Parameter	Value
V_{factor}	1.107167
STEPS_YEAR	5,797,311
e_{fit}	0.205634
e_{target}	0.205630
$\left\|e_{\mathrm{fit}} - e_{\mathrm{target}}\right\|$	4.291054×10^{-6}
e_{rmin}	0.228116
a_{fit}	78.217892

Conservation diagnostics. Global conservation metrics for the 1-year run are collected in Table A12. The DO invariant I_{DO} is conserved to numerical precision over the full orbit: the net fractional drift

$$\delta I_{\mathrm{DO,net}} = \frac{I_{\mathrm{DO}}(T) - I_{\mathrm{DO}}(0)}{|I_{\mathrm{DO}}(0)|} \approx 4.53 \times 10^{-13}$$

is accompanied by an RMS fluctuation

$$\mathrm{RMS}\left(\delta I_{\mathrm{DO}}\right) \approx 1.74 \times 10^{-13}.$$

The z-component of the total angular momentum exhibits comparable fidelity,

$$\mathrm{RMS}\left(\delta L_z\right) \approx 1.14 \times 10^{-13},$$

confirming that the hard-mass two-body system preserves the DO invariants over a full Mercury-like year.

The naive energy $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$ is reported for comparison, but it is not the conserved quantity in the DO framework once the work-closure term is included. The naive diagnostic explores a bounded texture band,

$$\delta E_{\mathrm{naive,net}} \approx 7.37 \times 10^{-4}, \qquad \mathrm{RMS}\left(\delta E_{\mathrm{naive}}\right) \approx 3.01 \times 10^{-1},$$

reflecting reversible exchanges as the orbit samples discrete shells and the associated anisotropies of the flux-integral representation. The flatness of I_{DO} and L_z is the conservation gate.

 doi:10.20944/preprints202405.2028.v8

Table A12. Global conservation metrics for the 1-year Mercury–Sun DO run.

Metric	Value
I_{DO} net drift	4.526×10^{-13}
RMS δI_{DO}	1.744×10^{-13}
Naive E net drift	7.370×10^{-4}
RMS $\delta E_{\mathrm{naive}}$	3.011×10^{-1}
RMS δL_z	1.140×10^{-13}

Taken together, the 1-year Mercury–Sun results establish that the DO flux-integral operator can sustain a moderately eccentric, high-mass two-body orbit with:

- orbital geometry tightly matched to the target eccentricity and semi-major axis;
- DO Invariant and angular momentum conserved at the 10^{-13} level over a full radial cycle;
- a bounded naive-energy texture band reflecting the discrete shell structure and direction rule, not dynamical instability.

This 1-year run provides the baseline for the extended 1000-orbit Mercury–Sun validation, where the same fixed operator is tested for long-horizon shape stability and secular perihelion precession.

Appendix B.2.5.4. Appendix B.5.4. Validation IV1: Visual Confirmation

A visual confirmation of the 1-year Mercury–Sun dynamics is provided by plotting the full trajectory of the Mercury analog in the 401^3 Chebyshev index-chart frame. The moderately elliptical Mercury orbit around the Sun is clearly visible, with the Sun located at the origin of the floating-point chart. Over the 1-year interval the orbit remains thin and coherent, exhibiting no discernible thickening or numerical artefacts in the hard-mass representation.

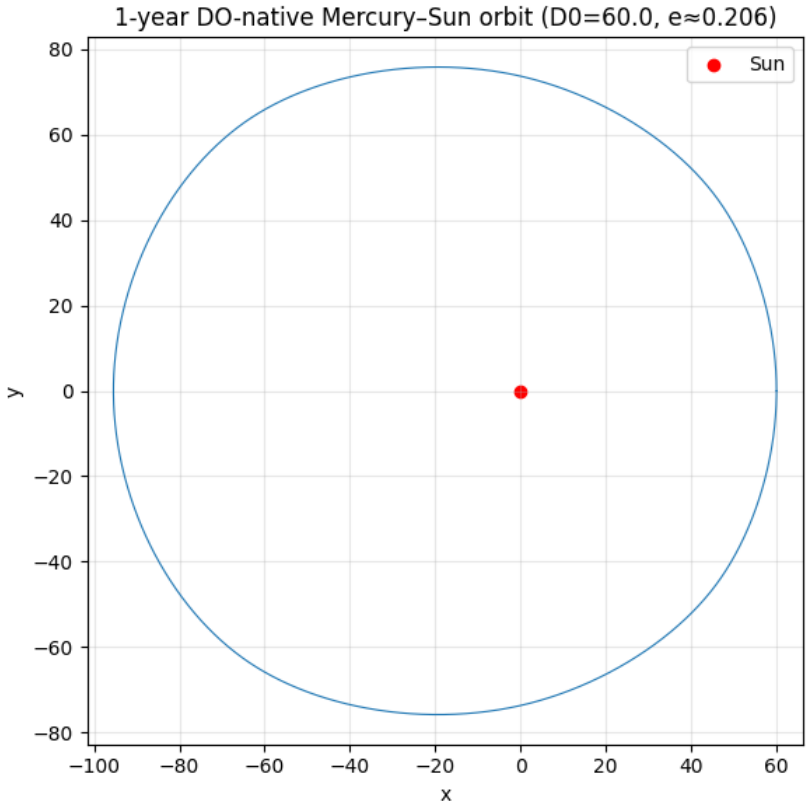


Figure A3. Trajectory of the 1-year DO-native Mercury–Sun system (flux-integral kernel, $D_0 = 60$), showing the moderately elliptical orbit of the Mercury analog around the Sun in the Chebyshev index-chart frame.

 doi:10.20944/preprints202405.2028.v8

Appendix B.2.5.5. Appendix B.5.5. Validation IV2: Mercury–Sun long horizon (1000 orbits)

To probe extreme long-horizon robustness of the same DO-native operator and configuration in the high-energy Mercury–Sun regime, the system is evolved for one thousand times the 1-year duration using identical parameters

$$(\chi_E,\ \chi_{\text{flux}},\ N_{\text{grid}},\ dt,\ D_0,\ V_{\text{factor}}).$$

Only the total number of steps is changed. The analysis horizon is defined by one thousand completed orbital cycles ("1000 DO-years") at launch radius $D_0 = 60$:

$$N_{\text{steps}}^{(\text{analyzed})} = 1000 \times \text{STEPS}_{\text{YEAR}} = 1000 \times 5{,}797{,}311 = 5.797311 \times 10^9, \qquad dt = 10^{-4}.$$

For robustness in boundary detection, the integration is executed for one additional orbit and the analysis is restricted to the first 1000 completed cycles:

$$N_{\text{steps}}^{(\text{executed})} = (1000 + 1) \times 5{,}797{,}311 = 5.803108311 \times 10^9.$$

Global 1000-orbit diagnostics. Over the full 1000-orbit integration, the conserved DO invariant and angular momentum remain tightly bounded:

$$\delta I_{\text{DO,net}} = \frac{I_{\text{DO}}(T) - I_{\text{DO}}(0)}{|I_{\text{DO}}(0)|} \approx 1.862 \times 10^{-11}, \qquad \text{RMS}(\delta I_{\text{DO}}) \approx 1.427 \times 10^{-11},$$

$$\delta L_{z,\text{net}} = \frac{L_z(T) - L_z(0)}{|L_z(0)|} \approx -5.492 \times 10^{-12}, \qquad \text{RMS}\left(\frac{\Delta L_z}{L_{z0}}\right) \approx 3.072 \times 10^{-12}.$$

The barycentric stability is equally strong, with barycenter drift far below the orbital scale:

$$\max\left\|\mathbf{r}_{\text{bary}}(t) - \mathbf{r}_{\text{bary}}(0)\right\| \approx 5.011 \times 10^{-10} \text{ hops}, \qquad \frac{\max\left\|\mathbf{r}_{\text{bary}}(t) - \mathbf{r}_{\text{bary}}(0)\right\|}{\langle r \rangle} \approx 6.227 \times 10^{-12}.$$

The naive energy $E_{\text{naive}} = E_{\text{kin}} + U$ continues to explore a stationary texture band while the DO invariant remains flat,

$$\delta E_{\text{naive,net}} \approx 4.189 \times 10^{-2}, \qquad \text{RMS}(\delta E_{\text{naive}}) \approx 3.009 \times 10^{-1},$$

confirming that the measured texture reflects the discrete representation rather than any long-horizon energy leak in DO dynamics.

Orbit counting and step stability. The orbital phase counter gives a full-horizon rotation of

$$\Delta\theta_{\text{full}} \approx 360000.061264^\circ, \qquad N_{\text{orbits}} \approx 1000.000170,$$

consistent with the extraction of 1000 completed orbit boundaries. The orbital step count remains narrowly distributed over the full horizon:

$$\left\langle N_{\text{steps/orbit}} \right\rangle \approx 5{,}795{,}321.239, \qquad \sigma_N \approx 797.244, \qquad N_{\min} = 5{,}793{,}692, \qquad N_{\max} = 5{,}796{,}861,$$

showing that the orbital clock remains stable over billions of discrete updates.

Orbital shape over 1000 years. The Mercury analog remains on a coherent bounded rosette, while the extended horizon resolves a slow secular evolution in the radial turning points and derived shape diagnostics. From the refined extrema over the first and last analyzed orbits:

$$e_{r\,\min}^{(1)} \approx 0.22812, \qquad a_r^{(1)} \approx 77.7319, \qquad e_{r\,\min}^{(1000)} \approx 0.23402, \qquad a_r^{(1000)} \approx 79.0795,$$

 doi:10.20944/preprints202405.2028.v8

with corresponding turning-point drift

$$r_{\min}^{(1)} \approx 60.0000, \qquad r_{\max}^{(1)} \approx 95.4639, \qquad r_{\min}^{(1000)} \approx 60.5731, \qquad r_{\max}^{(1000)} \approx 97.5858.$$

Block diagnostics (100-orbit means) show that this evolution is smooth and structured: $\langle e_{r\,\min} \rangle$ rises from ~ 0.2279 in orbits 1–100 to ~ 0.2342 in orbits 901–1000, while $\langle a_r \rangle$ rises from ~ 77.72 to ~ 79.00. The motion remains bounded and invariant-controlled throughout.

Perihelion angle extraction and long-horizon precession. Per-orbit perihelion angles ω_j are extracted within each orbit window using two estimators:

- Coarse: $\omega_j^{(c)}$ evaluated at the sampled index of the minimum of $r(t)$ within the orbit window.
- Refined: $\omega_j^{(r)}$ evaluated at a sub-sample perihelion point obtained by 3-point quadratic refinement of $r(t)$ at the local minimum, with consistent quadratic interpolation of $(x(t), y(t))$ at the refined offset.

The per-orbit increments $\Delta\omega_j = \omega_j - \omega_{j-1}$ are wrapped to $(-\pi, \pi]$ and reported in arcsec/orbit. Over the full 1000-orbit horizon (excluding the first orbit):

$$\langle \Delta\omega \rangle_{\text{coarse}} \approx 88.612524 \text{ arcsec/orbit}, \qquad \sigma_{\Delta\omega,\text{coarse}} \approx 226.768353 \text{ arcsec/orbit},$$

$$\langle \Delta\omega \rangle_{\text{refined}} \approx 88.684633 \text{ arcsec/orbit}, \qquad \sigma_{\Delta\omega,\text{refined}} \approx 7.406472 \text{ arcsec/orbit}.$$

The refined estimator collapses the stride-limited aliasing scatter while leaving the long-horizon mean essentially unchanged, establishing $\langle \Delta\omega \rangle$ as a stable diagnostic output of the fixed DO operator in this high-energy regime. The net apsidal advance is consistent with the accumulated perihelion angle drift:

$$\omega_{1000}^{(r)} \approx 24.609986^\circ,$$

corresponding to a slowly rotating rosette over ~ 1000 cycles.

Stationarity and structured long-horizon response. The refined precession signal exhibits blockwise variation across the 1000-orbit horizon, reported as 100-orbit means and standard deviations. The block means span a wide but structured range (e.g., ~ 104.4 arcsec/orbit in orbits 1–100, dipping to ~ 76.5 arcsec/orbit in orbits 601–700, then rising toward ~ 90.6 arcsec/orbit in orbits 901–1000), with within-block dispersions that remain finite and controlled. This separates cleanly into:

- Invariant control: I_{DO}, L_z, and barycenter remains flat at 10^{-11}–10^{-12} relative scale across the full horizon, demonstrating long-horizon stability of the DO update in the hard-mass limit.
- Predictable discrete-response structure: the high-energy band sampled by the Mercury analog in a finite Chebyshev shell network produces a bounded secular evolution of turning points and a correspondingly non-constant precession response under the fixed kernel and direction rule. The refined diagnostics expose this response cleanly, without contamination from stride aliasing.

Diagnostic confirmation of the anisotropy mechanism (non-adopted). A static diagnostic confirms the causal mechanism behind the long-horizon modulation: the baseline operator couples a shell-indexed magnitude to a Manhattan-normalized direction, yielding an orientation-dependent effective radial strength in the orbital plane. In the planar geometry,

$$d_\infty^{(\text{axis})} \approx r, \qquad d_\infty^{(\text{diag})} \approx \frac{r}{\sqrt{2}}, \qquad \frac{\|\hat{u}\|_{\text{diag}}}{\|\hat{u}\|_{\text{axis}}} \approx \frac{r}{r_1} \approx 0.707.$$

Under the large-d scaling $\gamma(d) \propto 1/d^2$, this implies a leading-order axis/diagonal imbalance of order

$$\frac{\|F\|_{\text{diag}}}{\|F\|_{\text{axis}}} \sim \frac{\gamma\left(r/\sqrt{2}\right)}{\gamma(r)} \cdot \frac{\|\hat{u}\|_{\text{diag}}}{\|\hat{u}\|_{\text{axis}}} \approx 2 \cdot 0.707 \approx 1.41,$$

 doi:10.20944/preprints202405.2028.v8

i.e. a characteristic $\sim 41\%$ orientation mismatch in the effective radial response at fixed Euclidean radius. Purely diagnostic reparameterizations of the radial argument can be tuned to cancel this mismatch to sub-percent levels, confirming causality. Such tuned metric modifications are not adopted as an "improvement operator" because they introduce a free geometric parameter selected by optimization rather than derived from DO ontology.

Realization class and invariants (scope). DO-law constraints and gates (ontology, PD-collapse rule and mirroring, and the no-feedback fence) are unchanged across admissible realizations. Realization data (kernel representative and cutoff, shell convention, distance chart, direction rule, schedule/integrator choices) are fixed per run and can change realization-level geometric texture (orientation anisotropy, turning-point drift, and apsidal-response modulation) without changing the DO law. Stability is assessed by the law-level audit gates (I_{DO}, angular momentum, and barycenter drift). Continuum/SR correspondence statements remain confined to Appendix A, Sec. 2.9 (requiring $w \in \mathcal{W}_{\Delta}$).

In this sense, the 1000-orbit Mercury–Sun run is a strong high-energy validation of the DO framework in a fully discrete environment: the correct DO invariant remains flat over billions of updates, the hard-mass system remains bounded over 1000 cycles, and the apsidal motion is extracted as a stable, well-resolved structural output of the fixed discrete operator.[41]

Appendix B.5.5.1. Implementation Note: 1000-Orbit Extension (Code Delta Only)

The 1000-orbit Mercury–Sun run uses the identical DO-native operator, kernel, integrator, and initial-condition construction as the 1-year script in Appendix B. Only the horizon, sampling/printing policy, and diagnostic post-processing are extended. The required code-level deltas are:

1. Horizon extension (no dynamics change).

$$N_{\mathrm{orbits}} \leftarrow 1000, \qquad \mathrm{STEPS}_{\mathrm{TOTAL}} \leftarrow (N_{\mathrm{orbits}} + 1) \times \mathrm{STEPS}_{\mathrm{YEAR}},$$

   where the extra orbit provides a safety margin for phase-boundary extraction while analysis is restricted to the first 1000 completed cycles.

2. Conserved quantity auditing. All long-horizon conservation auditing is performed on the DO invariant

$$I_{\mathrm{DO}} \equiv (E_{\mathrm{kin}} + U) - W_{\mathrm{closure}},$$

   rather than on $E_{\mathrm{naive}} = E_{\mathrm{kin}} + U$. The naive energy is retained only as a stationary texture diagnostic (Appendix C artifacts may print E_{kin} and $E_{\mathrm{kin}} + U$ using the legacy console labels `K` and `K+U`).

3. Refined perihelion localization (stride aliasing removal). Per-orbit perihelion angles are computed using sub-sample refinement inside each orbit window:
   - Identify the sampled perihelion index $k = \arg\min r(t)$ within the orbit window.
   - Compute the quadratic vertex offset

$$\delta_{\mathrm{v}} = \frac{1}{2} \frac{r_{k-1} - r_{k+1}}{r_{k-1} - 2r_k + r_{k+1}}, \qquad \delta_{\mathrm{v}} \in [-1, 1],$$

     and evaluate the refined perihelion position by 3-point quadratic interpolation

$$x^{\star} = x(\delta_{\mathrm{v}}), \qquad y^{\star} = y(\delta_{\mathrm{v}}).$$

[41] Validation IV2 demonstrates that a fixed DO hard-mass relational-gravity operator (fixed kernel realization, fixed direction rule, fixed integrator policy) generates a nonzero apsidal advance in a Mercury–Sun analog over a 1000-orbit horizon, with conserved-quantity auditing and long-horizon stability remaining within the stated gates. The perihelion advance is extracted directly from the discrete trajectory and is a diagnostic output of the fixed discrete operator, without inserting a continuum metric, a Newtonian/EFE potential term, or any mid-run tuning.

 doi:10.20944/preprints202405.2028.v8

- Define the refined perihelion angle $\omega_j = \mathrm{atan2}(y^\star, x^\star)$ and compute $\Delta\omega_j = \omega_j - \omega_{j-1}$ with wrap to $(-\pi, \pi]$.

The coarse estimator (index-only perihelion) may be retained for comparison, but the refined estimator is the reported diagnostic for the 1000-orbit horizon.

4. Stationarity diagnostics (blockwise summaries). To quantify structured long-horizon response without printing every orbit, compute blockwise means and standard deviations over fixed blocks (e.g. 100-orbit blocks):

$$\{\langle e_{r\,\min}\rangle,\ \sigma(e_{r\,\min}),\ \langle a_r\rangle,\ \sigma(a_r),\ \langle\Delta\omega\rangle,\ \sigma(\Delta\omega)\}_{\text{per block}},$$

with $\Delta\omega$ computed from the refined perihelion angles.

5. Output policy and plots. For readability at 1000-orbit scale:
   - Print only a compact per-orbit snapshot (e.g. the first N and last N orbits), and report full-horizon precession statistics.
   - Retain the standard diagnostic plots (2D rosette, 3D trajectories with COM, $r(t)$, δI_{DO}, δL_z, barycenter drift magnitude, and per-orbit series $e_{r\,\min}(j)$, $a_r(j)$, $\Delta\omega(j)$).

   No additional tuning, steering, or operator modification is introduced by the 1000-orbit extension; all changes are horizon and diagnostics only, preserving full DO compliance and direct comparability to the 1-year baseline.

Appendix B.2.5.6. Appendix B.5.6. Conclusions

Over a 1000-orbit horizon, the Mercury–Sun hard-mass system remains bounded and dynamically stable under the DO-native flux-integral gravitational operator. The primary conserved quantity I_{DO}, angular momentum L_z, and the barycenter remains tightly controlled at the 10^{-11}–10^{-12} level over $\sim 5.8 \times 10^9$ discrete updates. The naive energy continues to explore a stationary texture band without secular broadening. The extended horizon reveals a smooth secular evolution in radial turning points and shape diagnostics while preserving invariant control, and it yields a well-resolved refined apsidal advance with mean $\langle\Delta\omega\rangle \approx 88.685$ arcsec/orbit and a small residual dispersion $\sigma_{\Delta\omega} \approx 7.406$ arcsec/orbit once stride aliasing is removed by sub-sample perihelion refinement.

Table A13 summarizes the key conservation and precession diagnostics for the 1000-orbit Mercury–Sun run at $D_0 = 60$, using I_{DO} as the primary invariant and reporting both coarse and refined precession statistics.

Table A13. Summary diagnostics for the 1000-orbit Mercury–Sun run at $D_0 = 60$ (coarse and refined apsidal-advance statistics).

Horizon	Steps (executed)	Orbits (found)	RMS δI_{DO}	RMS δL_z	$\langle\Delta\omega\rangle_{\mathrm{c}}$	σ_{c}	$\langle\Delta\omega\rangle_{\mathrm{r}}$	σ_{r}
1000-orbit	5.803×10^9	1000	1.427×10^{-11}	3.072×10^{-12}	8.861×10^1	2.268×10^2	8.868×10^1	7.406

The table highlights three core features:

- Invariant control. I_{DO} and L_z remain tightly bounded over $\sim 5.8 \times 10^9$ discrete updates.
- Measurement convergence. The refined perihelion estimator collapses stride-aliasing scatter while leaving the mean apsidal advance essentially unchanged.
- Structured long-horizon response. The bounded secular evolution in turning points and the blockwise modulation of $\langle\Delta\omega\rangle_r$ represent predictable long-horizon behaviour of this fixed discrete kernel and direction rule in the high-energy band sampled by the Mercury analog.

 doi:10.20944/preprints202405.2028.v8

Appendix C. Run Artifacts (Scripts and Console Transcripts)

Appendix C contains the executable scripts and console transcripts for the eight validations reported in Appendix B (Validations I, II, III–A1, III–A2, III–B1, III–B2, IV1, IV2). For each validation, the executable script and the complete console transcript are reproduced verbatim and unedited.

Appendix C.1. Appendix B.6. Validation I: $t = 0 \to$ CMB (low–ℓ TT)

Cross-reference. Appendix B, Validation I (Sec. B.1).

Appendix B.3.1.1. Appendix B.6.1. Executable Script

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#
# DO CMB — Validation I (DO-text t=0 init, \epsilon_seed=1e-15) — \ellmax=50 PRODUCTION run
#
# End-to-end reproducible run:
#   t=0 DO-text init → deterministic torus tick map → shell+weights → eigensolve → calibration T → C_pred →
↪  \kappa* → C_model → \chi²
#
# DO / Appendix-A constraints enforced:
#   - DO_COLLAPSE_EVERY = 0 (no stochastic edits inside S(4D) tick loop)
#   - single \theta(1) for all Chebyshev d=1 nearest-neighbor links (26 neighbors)
#   - fixed periodic torus link set + fixed deterministic schedule each tick
#   - bridge/diagnostics are read-only (no feedback into \Phi)
#
# Outputs:
#   - Bell-sum audit + seed stats at t=0 and after evolution
#   - shell guardrail (sum w = 4\pi)
#   - eigensolve residual (M-orthonormality)
#   - \kappa* (fit 5..50), \chi²(5..50)
#   - bandwise \chi² contributions (canonical \kappa*)
#   - per-\ell table \ell=2..50
#   - top \Delta\chi² contributors
#   - rotation gate (x\leftrightarrowy, y\leftrightarrowz, x\leftrightarrowz)
#   - \epsilon-linearity probe (bridge-only)

import os, sys, time, math, hashlib, warnings, threading
import numpy as np

os.environ.setdefault("OMP_NUM_THREADS", "1")
os.environ.setdefault("MKL_NUM_THREADS", "1")
os.environ.setdefault("OPENBLAS_NUM_THREADS", "1")

import scipy.sparse as sp
import scipy.sparse.linalg as spla
from numba import njit

np.set_printoptions(suppress=True)

# ============================================================================
# 0) USER CONTROLS (single block)
# ============================================================================
DO_COLLAPSE_EVERY = 0
THETA_1 = 0.020
```

 doi:10.20944/preprints202405.2028.v8


```
EPS_SEED = 1e-15
SEED_ANISO = 20250101

NBOX     = 69

# One full sweep: 13 offsets × 3 modes = 39 ticks
SCHED_LEN = 39
K_TICKS_CYCLE1 = 39
K_TICKS_CYCLE2 = 78
DO_SECOND_CYCLE_CHECK = False

PROG_TICKS_1 = [1, 10, 20, 30, 39]
PROG_TICKS_2 = [40, 50, 60, 70, 78]

R_SHELL = 30
ELL_MAX = 50
ELL_PAD = 3
K_EIG   = (ELL_MAX + ELL_PAD + 1)**2   # 2916

BINS        = 32
SEED_SPLIT  = 20240919
CALIB_EPS   = 0.5
CALIB_CHUNK_Q = 32

ELL_FIT_MIN = 5
ELL_FIT_MAX = 50

# Gates
BELL_SUM_REL_MAX      = 5e-10
SHELL_WSUM_REL_MAX    = 1e-12
EIG_M_ORTHO_MAX       = 5e-9
ROT_CHI2_SPREAD_MAX   = 2.0

EPS_CHECK_VALUES      = (0.25, 0.75)
EPS_CHECK_NSAMPLES    = 96
EPS_MEAN_MAX          = 2e-2
EPS_MAX_MAX           = 1e-1

# ============================================================================
# 1) Planck low-\ell TT (\ell=2..50)
# ============================================================================
OBS = np.array([
 [ 2, 6.371975e-11, 2.993180e-10],
 [ 3, 3.757952e-10, 2.353983e-10],
 [ 4, 2.097323e-10, 1.355616e-10],
 [ 5, 3.780877e-10, 1.055243e-10],
 [ 6, 1.585670e-10, 8.444205e-11],
 [ 7, 2.677040e-10, 7.081953e-11],
 [ 8, 1.158792e-10, 6.020004e-11],
 [ 9, 1.070531e-10, 5.238777e-11],
 [10, 1.388228e-10, 4.284038e-11],
 [11, 1.345381e-10, 3.776291e-11],
 [12, 9.810913e-11, 3.624853e-11],
 [13, 1.131737e-10, 3.325981e-11],
 [14, 1.020729e-10, 3.108198e-11],
```

 doi:10.20944/preprints202405.2028.v8


```
 [15, 1.277477e-10, 2.841248e-11],
 [16, 8.742653e-11, 2.920708e-11],
 [17, 1.152576e-10, 2.732701e-11],
 [18, 8.023027e-11, 2.615252e-11],
 [19, 9.230326e-11, 2.510391e-11],
 [20, 7.651180e-11, 2.447305e-11],
 [21, 6.577979e-11, 2.362083e-11],
 [22, 6.058016e-11, 2.245348e-11],
 [23, 7.057404e-11, 2.155320e-11],
 [24, 7.610091e-11, 2.072381e-11],
 [25, 6.643222e-11, 2.075658e-11],
 [26, 7.658001e-11, 2.019592e-11],
 [27, 7.705872e-11, 1.872003e-11],
 [28, 9.224762e-11, 1.746402e-11],
 [29, 8.635307e-11, 1.703794e-11],
 [30, 6.934708e-11, 1.693556e-11],
 [31, 7.896162e-11, 1.654951e-11],
 [32, 1.046747e-10, 1.584769e-11],
 [33, 6.971541e-11, 1.573058e-11],
 [34, 6.677449e-11, 1.515098e-11],
 [35, 5.700753e-11, 1.468349e-11],
 [36, 6.273364e-11, 1.416504e-11],
 [37, 8.080809e-11, 1.354318e-11],
 [38, 7.631162e-11, 1.319514e-11],
 [39, 7.849050e-11, 1.292744e-11],
 [40, 1.015036e-10, 1.247094e-11],
 [41, 6.266490e-11, 1.225876e-11],
 [42, 7.400832e-11, 1.203643e-11],
 [43, 6.635391e-11, 1.187754e-11],
 [44, 6.328508e-11, 1.169798e-11],
 [45, 5.380897e-11, 1.154021e-11],
 [46, 4.857735e-11, 1.131740e-11],
 [47, 7.537541e-11, 1.106781e-11],
 [48, 4.103847e-11, 1.090678e-11],
 [49, 6.836774e-11, 1.073460e-11],
 [50, 6.750351e-11, 1.052116e-11],
], dtype=np.float64)

OBS_L = OBS[:,0].astype(int)
OBS_MEAN = OBS[:,1].astype(np.float64)
OBS_SIG  = OBS[:,2].astype(np.float64)
OBS_VAR  = OBS_SIG**2
OBS_L_TO_I = {int(l): int(i) for i,l in enumerate(OBS_L)}

def get_obs(ell: int):
    i = OBS_L_TO_I[ell]
    return float(OBS_MEAN[i]), float(OBS_SIG[i])

# ============================================================================
# 2) Utilities
# ============================================================================
class Timer:
    def __init__(self):
        self.t0 = time.perf_counter()
        self.last = self.t0
```

 doi:10.20944/preprints202405.2028.v8


```
    def beat(self, label):
        now = time.perf_counter()
        dt = now - self.last
        self.last = now
        print(f"[time] {label}: +{dt:.2f}s | total {now - self.t0:.2f}s", flush=True)

def require(cond, msg):
    if not cond:
        print(f"[GATE FAIL] {msg}", flush=True)
        raise RuntimeError(msg)

# ============================================================================
# 3) Shell graph + weights (Chebyshev shell, 26C adjacency)
# ============================================================================
def shell_coords(R):
    out=[]
    for x in range(-R,R+1):
        for y in range(-R,R+1):
            for z in range(-R,R+1):
                if max(abs(x),abs(y),abs(z))==R:
                    out.append((x,y,z))
    return np.array(out, dtype=np.int32)

def unique_face_weight(x,y,z,R):
    ax,ay,az = abs(x),abs(y),abs(z)
    if ax>=ay and ax>=az:
        xi=y/float(R); eta=z/float(R)
    elif ay>=ax and ay>=az:
        xi=x/float(R); eta=z/float(R)
    else:
        xi=x/float(R); eta=y/float(R)
    return 1.0/(1.0+xi*xi+eta*eta)**1.5

def build_graph_and_weights(R):
    coords = shell_coords(R)
    n = coords.shape[0]
    idx = {tuple(c):i for i,c in enumerate(coords)}
    rows=[]; cols=[]
    nbrs=[(dx,dy,dz) for dx in (-1,0,1) for dy in (-1,0,1) for dz in (-1,0,1)
          if not (dx==dy==dz==0)]
    for i,(x,y,z) in enumerate(coords):
        for dx,dy,dz in nbrs:
            nx,ny,nz = x+dx, y+dy, z+dz
            if max(abs(nx),abs(ny),abs(nz))==R and (nx,ny,nz) in idx:
                rows.append(i); cols.append(idx[(nx,ny,nz)])
    A = sp.csr_matrix((np.ones(len(rows)), (rows, cols)), shape=(n,n), dtype=np.float64)
    deg = np.asarray(A.sum(axis=1)).ravel()
    L = sp.diags(deg) - A
    w = np.array([unique_face_weight(int(x),int(y),int(z),R) for (x,y,z) in coords], dtype=np.float64)
    w = np.maximum(w, 1e-16)
    w *= (4.0*np.pi)/w.sum()
    return coords, L, w

# ============================================================================
# 4) Eigensolve + band indexing
```

 doi:10.20944/preprints202405.2028.v8


```
# ============================================================================
def bands_cover_all(K):
    bands=[]; used=0; ell=0
    while used < K:
        need = 2*ell + 1
        take = min(need, K-used)
        bands.append((ell, used, used+take))
        used += take; ell += 1
    return np.array(bands, dtype=np.int32)

def compute_modes_eigsh(L, w, Kneed):
    invsqrtw = 1.0/np.sqrt(w)
    def matvec(x):
        y = invsqrtw * x
        y = L @ y
        y = invsqrtw * y
        return y
    n = w.size
    Bop = spla.LinearOperator((n,n), matvec=matvec, dtype=np.float64)
    t0 = time.perf_counter()
    vals, vecs = spla.eigsh(Bop, k=Kneed, which="SM", tol=1e-10, maxiter=None)
    t1 = time.perf_counter()
    order = np.argsort(vals)
    vecs = vecs[:, order]
    PsiK = (invsqrtw[:,None]) * vecs
    resid = np.linalg.norm(PsiK.T @ (w[:,None]*PsiK) - np.eye(PsiK.shape[1]), ord="fro")
    return PsiK, resid, (t1 - t0)

# ============================================================================
# 5) DO-text t=0 initialization at \epsilon_seed=1e-15 (energy seeds)
# ============================================================================
def init_phi_do_text(N: int, eps_seed: float, seed_aniso: int):
    rng = np.random.default_rng(seed_aniso)
    s = rng.standard_normal((N,N,N)).astype(np.float64)
    s -= float(s.mean())
    mx = float(np.max(np.abs(s)))
    if mx > 0.0:
        s /= mx
    s -= float(s.mean())

    e = 1.0 + eps_seed * s
    e_min = float(e.min())
    if e_min <= 0.0:
        e = e - e_min + 1e-30

    return np.sqrt(e).astype(np.complex128)

def energy_residual_stats(phi: np.ndarray, label: str):
    E = (np.abs(phi).astype(np.float64)**2)
    mu = float(E.mean())
    res = E - mu
    print(f"[{label}] mean(|\Phi|^2)={mu:.16e}  mean(res)={float(res.mean()):+.3e}", flush=True)
    rstd = float(res.std()); rmax = float(np.max(np.abs(res)))
    print(f"[{label}] std(res)/mean={rstd/max(1e-300,mu):.3e}  max|res|/mean={rmax/max(1e-300,mu):.3e}",
    ↪  flush=True)
```

 doi:10.20944/preprints202405.2028.v8


```
# ================================================================================
# 6) Deterministic torus tick map (one matching per tick; precomputed pairs)
# ================================================================================
OFFSETS = [
    ( 1,0,0),(0,1,0),(0,0,1),
    ( 1,1,0),(1,-1,0),(1,0,1),(1,0,-1),(0,1,1),(0,1,-1),
    ( 1,1,1),(1,1,-1),(1,-1,1),(1,-1,-1),
]

def axis_for_offset(dx:int, dy:int, dz:int)->int:
    adx, ady, adz = abs(dx), abs(dy), abs(dz)
    if adx>=ady and adx>=adz and adx!=0: return 0
    if ady>=adx and ady>=adz and ady!=0: return 1
    return 2

def build_schedule_arrays(offsets, theta_1: float):
    c = float(math.cos(theta_1))
    s = float(math.sin(theta_1))
    dxs=[]; dys=[]; dzs=[]; axs=[]; mds=[]; cs=[]; sn=[]
    for mode in (0,1,2):
        for (dx,dy,dz) in offsets:
            ax = axis_for_offset(int(dx),int(dy),int(dz))
            dxs.append(int(dx)); dys.append(int(dy)); dzs.append(int(dz))
            axs.append(int(ax)); mds.append(int(mode))
            cs.append(c); sn.append(s)
    return (np.array(dxs, dtype=np.int32),
            np.array(dys, dtype=np.int32),
            np.array(dzs, dtype=np.int32),
            np.array(axs, dtype=np.int32),
            np.array(mds, dtype=np.int32),
            np.array(cs,  dtype=np.float64),
            np.array(sn,  dtype=np.float64))

def precompute_pairs_flat(N: int, dxs, dys, dzs, axs, modes):
    n_total = N**3
    idx = np.arange(n_total, dtype=np.int32)
    NN = N*N
    i = (idx // NN).astype(np.int16)
    j = ((idx // N) % N).astype(np.int16)
    k = (idx % N).astype(np.int16)

    pairs_i = []
    pairs_j = []
    ptr = [0]

    Nm1 = N - 1
    for s in range(dxs.size):
        ax = int(axs[s])
        md = int(modes[s])
        coord = i if ax==0 else (j if ax==1 else k)

        if md == 2:
            mask = (coord == Nm1)
        else:
```

 doi:10.20944/preprints202405.2028.v8


```
            mask = (coord != Nm1) & ((coord & 1) == md)

        start_idx = idx[mask]

        ii = (i[mask].astype(np.int32) + int(dxs[s])) % N
        jj = (j[mask].astype(np.int32) + int(dys[s])) % N
        kk = (k[mask].astype(np.int32) + int(dzs[s])) % N
        partner_idx = (ii * NN + jj * N + kk).astype(np.int32)

        pairs_i.append(start_idx)
        pairs_j.append(partner_idx)
        ptr.append(ptr[-1] + start_idx.size)

    pair_i = np.concatenate(pairs_i).astype(np.int32, copy=False)
    pair_j = np.concatenate(pairs_j).astype(np.int32, copy=False)
    ptr = np.array(ptr, dtype=np.int64)
    return pair_i, pair_j, ptr

@njit(fastmath=True)
def apply_matching_pairs(phi1d, pair_i, pair_j, ptr, cos_arr, sin_arr, sidx):
    c = cos_arr[sidx]
    s = sin_arr[sidx]
    p0 = ptr[sidx]
    p1 = ptr[sidx+1]
    for p in range(p0, p1):
        ia = pair_i[p]
        ib = pair_j[p]

        ar = phi1d[ia].real
        ai = phi1d[ia].imag
        br = phi1d[ib].real
        bi = phi1d[ib].imag

        sr = 0.5*(ar + br)
        si = 0.5*(ai + bi)
        dr = 0.5*(ar - br)
        di = 0.5*(ai - bi)

        dr2 = c*dr - s*di
        di2 = s*dr + c*di

        phi1d[ia] = complex(sr + dr2, si + di2)
        phi1d[ib] = complex(sr - dr2, si - di2)

# ============================================================================
# 7) Bridge: SEEDED-Q + energy residual spectrum (scaled FFT)
# ============================================================================
def radial_bins_edges(B: int):
    return np.linspace(0.0, math.sqrt(3.0)/2.0, B+1)

def canonical_class_triplet(kx: int, ky: int, kz: int):
    ax, ay, az = abs(int(kx)), abs(int(ky)), abs(int(kz))
    if ax==0 and ay==0 and az==0:
        return None
    g = math.gcd(ax, math.gcd(ay, az))
```

 doi:10.20944/preprints202405.2028.v8


```
    ax//=g; ay//=g; az//=g
    t = sorted((ax,ay,az), reverse=True)
    return (t[0], t[1], t[2])

MASK64 = (1<<64) - 1
def splitmix64(x: int) -> int:
    x = (x + 0x9E3779B97F4A7C15) & MASK64
    z = x
    z = (z ^ (z >> 30)) * 0xBF58476D1CE4E5B9 & MASK64
    z = (z ^ (z >> 27)) * 0x94D049BB133111EB & MASK64
    return (z ^ (z >> 31)) & MASK64

def stable_hash_bucket_q(seed: int, sbin: int, gid: int, qx: int, qy: int, qz: int) -> int:
    h = seed & MASK64
    h = splitmix64(h ^ (sbin + 1))
    h = splitmix64(h ^ (gid  + 1))
    h = splitmix64(h ^ (qx + 2048))
    h = splitmix64(h ^ (qy + 2048))
    h = splitmix64(h ^ (qz + 2048))
    return h

def scan_classes_and_seeded_q(N: int, edges, seed_split: int):
    half = N//2
    class_to_id = {}
    classes = []
    seeded_q = {}
    best_h = {}
    for qx in range(-half, half+1):
        for qy in range(-half, half+1):
            for qz in range(-half, half+1):
                if qx==0 and qy==0 and qz==0:
                    continue
                fm = math.sqrt(qx*qx + qy*qy + qz*qz)/float(N)
                sbin = int(np.digitize([fm], edges)[0] - 1)
                if sbin<0 or sbin>=len(edges)-1:
                    continue
                tpl = canonical_class_triplet(qx,qy,qz)
                if tpl is None:
                    continue
                gid = class_to_id.get(tpl)
                if gid is None:
                    gid = len(classes)
                    class_to_id[tpl] = gid
                    classes.append(tpl)
                key = (sbin, gid)
                h = stable_hash_bucket_q(seed_split, sbin, gid, qx, qy, qz)
                prev = best_h.get(key)
                if prev is None or h < prev:
                    best_h[key] = h
                    seeded_q[key] = (qx,qy,qz)
    return np.array(classes, dtype=np.int16), class_to_id, seeded_q

def spectrum_from_energy_residual(phi: np.ndarray, eps_seed: float, edges, class_to_id: dict):
    N = phi.shape[0]
    E = (np.abs(phi).astype(np.float64)**2)
```

 doi:10.20944/preprints202405.2028.v8


```
    mu = float(E.mean())
    res = E - mu

    res_norm = (res / eps_seed).astype(np.float64)
    res_norm -= float(res_norm.mean())

    F = np.fft.fftn(res_norm, s=res_norm.shape, axes=(0,1,2))
    Pn = (F.conj()*F).real / (N**3)

    idx = np.arange(N)
    ki = idx.copy()
    ki[ki > N//2] -= N
    kx, ky, kz = np.meshgrid(ki, ki, ki, indexing='ij')
    kx = kx.ravel(); ky = ky.ravel(); kz = kz.ravel()
    Pflat = Pn.ravel()

    mask = ~((kx==0)&(ky==0)&(kz==0))
    kx = kx[mask]; ky = ky[mask]; kz = kz[mask]; Pflat = Pflat[mask]

    fm = np.sqrt(kx*kx + ky*ky + kz*kz)/float(N)
    s_idx = np.digitize(fm, edges) - 1

    P_sg_norm = np.zeros((len(edges)-1, len(class_to_id)), dtype=np.float64)
    for i in range(Pflat.size):
        sbin = int(s_idx[i])
        if sbin<0 or sbin>=len(edges)-1:
            continue
        tpl = canonical_class_triplet(int(kx[i]), int(ky[i]), int(kz[i]))
        gid = class_to_id.get(tpl)
        if gid is None:
            continue
        P_sg_norm[sbin, gid] += float(Pflat[i])

    return (eps_seed**2) * P_sg_norm

# ============================================================================
# 8) Calibration T + \kappa-fit + reporting
# ============================================================================
def project_cl_batched(A_mat, w, Psi, bands):
    mu_cols=(w[:,None]*A_mat).sum(axis=0)/(4.0*np.pi)
    delta=A_mat-mu_cols[None,:]
    WAm=w[:,None]*delta
    coeffs=Psi.T@WAm
    Lmax=int(bands[-1,0])
    C_all=np.zeros((Lmax+1, A_mat.shape[1]), dtype=np.float64)
    for ell,s,e in bands:
        num=np.sum(coeffs[s:e,:]**2, axis=0)
        C_all[ell,:]=(num/(2*ell+1))/(4.0*np.pi)
    return C_all

def calibrate_T_from_seeded_q(coords64, w, PsiK, band_rows, N, edges, qmap, eps, chunk_q, n_classes):
    two_pi_over_N = 2.0*np.pi/float(N)
    Sbins = len(edges)-1
    Lrows = int(band_rows[-1,0]) + 1
    T_lsg = np.zeros((Lrows, Sbins, n_classes), dtype=np.float64)
```

 doi:10.20944/preprints202405.2028.v8


```
    cnt   = np.zeros((Sbins, n_classes), dtype=np.int32)
    P_unit = (eps*eps)*(N**3)/2.0

    by_s=[[] for _ in range(Sbins)]
    for (sbin,gid), q in qmap.items():
        by_s[int(sbin)].append((int(gid), q))

    for sbin in range(Sbins):
        items=by_s[sbin]
        if not items: continue
        start=0
        while start < len(items):
            end=min(start+chunk_q, len(items))
            chunk=items[start:end]
            gids=np.array([g for (g,_) in chunk], dtype=np.int32)
            Q=np.array([q for (_,q) in chunk], dtype=np.int64).T
            D=coords64@Q
            ang=two_pi_over_N*D

            Emat_cos = 1.0 + eps*np.cos(ang)
            Emat_sin = 1.0 + eps*np.sin(ang)

            mu_c=(w[:,None]*Emat_cos).sum(axis=0)/(4.0*np.pi)
            mu_s=(w[:,None]*Emat_sin).sum(axis=0)/(4.0*np.pi)

            A_cos=0.25*(Emat_cos/mu_c[None,:] - 1.0)
            A_sin=0.25*(Emat_sin/mu_s[None,:] - 1.0)

            A_cat=np.concatenate([A_cos,A_sin], axis=1)
            gids2=np.concatenate([gids,gids], axis=0)

            C_cols=project_cl_batched(A_cat, w, PsiK, band_rows)

            for j,g in enumerate(gids2):
                gg=int(g)
                T_lsg[:,sbin,gg]+=C_cols[:,j]
                cnt[sbin,gg]+=1
            start=end

    nz=cnt>0
    T_lsg[:,nz]/=(cnt[nz]*P_unit)
    return T_lsg

def kappa_fit(C_pred, ell_min, ell_max):
    ells=np.arange(ell_min, ell_max+1, dtype=int)
    Cm=np.array([C_pred[ell] for ell in ells], dtype=np.float64)
    Co=np.array([OBS_MEAN[OBS_L_TO_I[ell]] for ell in ells], dtype=np.float64)
    wts=1.0/np.array([OBS_VAR[OBS_L_TO_I[ell]] for ell in ells], dtype=np.float64)
    den=float(np.sum(wts*Cm*Cm))
    require(den > 0.0, "\kappa-fit denominator nonpositive.")
    num=float(np.sum(wts*Cm*Co))
    k2=max(0.0, num/den)
    chi2=float(np.sum(wts*(k2*Cm - Co)**2))
    return math.sqrt(k2), k2, chi2
```

 doi:10.20944/preprints202405.2028.v8


```
def chi2_eval(C_model, ell_min, ell_max):
    tot = 0.0
    for ell in range(ell_min, ell_max+1):
        Co, So = get_obs(ell)
        Cm = float(C_model[ell])
        z = (Cm - Co)/So
        tot += z*z
    return float(tot)

def rotate_volume_axes(Phi, perm):
    return np.transpose(Phi, axes=perm).copy()

def eps_linearity_probe(coords, w, PsiK, band_rows, N, samples, eps_vals, ell_max):
    rng = np.random.default_rng(20240229)
    two_pi_over_N = 2.0*np.pi/float(N)
    coords64 = coords.astype(np.int64)
    half = N//2
    qs=[]
    while len(qs) < samples:
        q = tuple(int(x) for x in rng.integers(-half, half+1, size=3))
        if q==(0,0,0): continue
        qs.append(q)
    Q = np.array(qs, dtype=np.int64).T
    D = coords64 @ Q
    ang = two_pi_over_N * D

    C_by_eps=[]
    for eps in eps_vals:
        Emat_cos = 1.0 + eps*np.cos(ang)
        Emat_sin = 1.0 + eps*np.sin(ang)
        mu_c = (w[:,None]*Emat_cos).sum(axis=0)/(4.0*np.pi)
        mu_s = (w[:,None]*Emat_sin).sum(axis=0)/(4.0*np.pi)
        A_cos = 0.25*(Emat_cos/mu_c[None,:] - 1.0)
        A_sin = 0.25*(Emat_sin/mu_s[None,:] - 1.0)
        A_cat = np.concatenate([A_cos, A_sin], axis=1)
        C_cols = project_cl_batched(A_cat, w, PsiK, band_rows)
        C_by_eps.append(C_cols)

    up = min(int(ell_max), int(band_rows[-1,0]))
    spreads=[]
    for j in range(C_by_eps[0].shape[1]):
        ratios=[]
        for t,eps in enumerate(eps_vals):
            val = C_by_eps[t][:up+1, j]
            ratios.append(val / max(1e-300, eps*eps))
        ratios = np.stack(ratios, axis=0)
        mu = np.mean(ratios, axis=0)
        if np.all(mu==0): continue
        dev = np.sqrt(np.mean((ratios - mu[None,:])**2, axis=0)) / np.maximum(1e-18, np.abs(mu))
        spreads.append(float(np.mean(dev)))
    return (float(np.mean(spreads)) if spreads else np.nan,
            float(np.max(spreads)) if spreads else np.nan)

# ============================================================================
# 9) Main
```

 doi:10.20944/preprints202405.2028.v8


```
# ==============================================================================
def main():
    warnings.filterwarnings("ignore", category=UserWarning)
    require(DO_COLLAPSE_EVERY == 0, "DO_COLLAPSE_EVERY must be 0.")
    T = Timer()

    print("DO CMB — Validation I (DO-text t=0 init, \epsilon_seed=1e-15) — \ellmax=50", flush=True)
    print("Efficiency: \Phi is flat 1D; per-tick pairs precomputed; tight Numba loop per tick", flush=True)
    print(f"NBOX={NBOX}, \theta(1)={THETA_1:.6f}, DO_COLLAPSE_EVERY={DO_COLLAPSE_EVERY}", flush=True)
    print(f"t=0 seeds: EPS_SEED={EPS_SEED:.1e}, SEED_ANISO={SEED_ANISO}", flush=True)
    print(f"Schedule: R={SCHED_LEN} ticks per sweep (13 offsets × 3 modes) | Cycle1 ticks={K_TICKS_CYCLE1}
    ↪  (production)", flush=True)
    print(f"Cycle2 ticks={K_TICKS_CYCLE2} (verification only) | enabled={DO_SECOND_CYCLE_CHECK}",
    ↪  flush=True)
    print(f"Shell: R={R_SHELL}, K_EIG={K_EIG} (\ell_cap\approx{ELL_MAX}+pad={ELL_PAD}), BINS={BINS}",
    ↪  flush=True)
    print(f"\kappa-fit window: \ell={ELL_FIT_MIN}..{ELL_FIT_MAX}\n", flush=True)

    # t=0 init
    phi0 = init_phi_do_text(NBOX, EPS_SEED, SEED_ANISO)
    print(f"\Phi(t=0): sha256[:8]={hashlib.sha256(phi0.tobytes()).hexdigest()[:8]}", flush=True)
    print(f"Bell-sum E0_init={float(np.sum(np.abs(phi0)**2)):.6e}", flush=True)
    energy_residual_stats(phi0, "t0")
    T.beat("t=0 init")

    # schedule arrays + precompute pairs
    dxs,dys,dzs,axs,modes,cos_arr,sin_arr = build_schedule_arrays(OFFSETS, float(THETA_1))
    pair_i, pair_j, ptr = precompute_pairs_flat(NBOX, dxs, dys, dzs, axs, modes)

    # convert phi to flat for fast ticking
    phi = phi0.ravel().copy()
    E0 = float(np.sum((phi.real*phi.real + phi.imag*phi.imag), dtype=np.float64))

    # warm compile
    _tmp = phi.copy()
    apply_matching_pairs(_tmp, pair_i, pair_j, ptr, cos_arr, sin_arr, 0)
    del _tmp

    T.beat("precompute pairs")

    # cycle1 evolution
    for K in range(1, K_TICKS_CYCLE1+1):
        sidx = (K-1) % SCHED_LEN
        apply_matching_pairs(phi, pair_i, pair_j, ptr, cos_arr, sin_arr, sidx)
        if K in PROG_TICKS_1:
            Ecur = float(np.sum((phi.real*phi.real + phi.imag*phi.imag), dtype=np.float64))
            relk = abs(Ecur - E0) / max(1e-300, E0)
            print(f"[gen] tick {K:2d}/{K_TICKS_CYCLE1}: r(K)={sidx+1:2d} | Bell drift={relk:.3e}",
            ↪  flush=True)

    E1 = float(np.sum((phi.real*phi.real + phi.imag*phi.imag), dtype=np.float64))
    relE = abs(E1 - E0) / max(1e-300, E0)

    phi3 = phi.reshape((NBOX,NBOX,NBOX))
    print(f"\n\Phi(after cycle1): sha256[:8]={hashlib.sha256(phi3.tobytes()).hexdigest()[:8]}", flush=True)
```

 doi:10.20944/preprints202405.2028.v8


```
print(f"Bell-sum audit: E0={E0:.6e}  E1={E1:.6e}  rel_drift={relE:.3e}", flush=True)
require(relE <= BELL_SUM_REL_MAX, "Bell-sum drift too large.")
energy_residual_stats(phi3, "t_end_cycle1")
T.beat("cycle1 evolution")

# shell geometry
coords, L, w = build_graph_and_weights(R_SHELL)
coords64 = coords.astype(np.int64)
rel_meas = abs(w.sum() - 4.0*np.pi)/(4.0*np.pi)
print("\n===== Guardrails (geometry) =====", flush=True)
print(f"shell nodes={coords.shape[0]}, weights sum={w.sum():.12f} (4\pi={4*np.pi:.12f}) |
↪  rel.err={rel_meas:.3e}", flush=True)
require(rel_meas <= SHELL_WSUM_REL_MAX, "Shell weights sum deviates from 4\pi.")
T.beat("shell geometry")

# eigensolve
print(f"\n[eigensolve] computing K={K_EIG} lowest modes ...", flush=True)
PsiK, resid, tsolve = compute_modes_eigsh(L, w, K_EIG)
print(f"[eigensolve] ||\PsiA^{\mathrm{T}} M \Psi $-$ I||_F = {resid:.3e} | t={tsolve:.1f}s", flush=True)
require(resid <= EIG_M_ORTHO_MAX, "Eigensolve residual too large.")
band_rows = bands_cover_all(PsiK.shape[1])
T.beat("eigensolve")

# qmap + P_sg
edges = radial_bins_edges(BINS)
classes, class_to_id, q_seeded = scan_classes_and_seeded_q(NBOX, edges, SEED_SPLIT)
P_sg = spectrum_from_energy_residual(phi3, EPS_SEED, edges, class_to_id)
print("\n[prep]", flush=True)
print(f"classes={classes.shape[0]}, buckets_used={len(q_seeded)}", flush=True)
T.beat("qmap")

# calibration
print("\n[calibration] building T (SEEDED-Q full buckets) ...", flush=True)
T_lsg = calibrate_T_from_seeded_q(coords64, w, PsiK, band_rows, NBOX, edges, q_seeded,
                                  CALIB_EPS, CALIB_CHUNK_Q, int(classes.shape[0]))
T.beat("T calibration")

# predict + fit
C_pred = np.tensordot(T_lsg, P_sg, axes=([1,2],[0,1]))
kappa, k2, chi2 = kappa_fit(C_pred, ELL_FIT_MIN, ELL_FIT_MAX)
C_model = k2 * C_pred

n_pts = ELL_FIT_MAX - ELL_FIT_MIN + 1
print("\n===== FIT SUMMARY (Cycle 1 production) =====", flush=True)
print(f"\kappa = {kappa:.15e}   \chi²(\ell=5-50) = {chi2:.3f} over n_pts={n_pts}
↪  \chi²/(n_pts-1)={chi2/max(1,n_pts-1):.3f}", flush=True)

# bandwise \chi²
print("\n===== Bandwise \chi² contributions (\kappa frozen; Cycle 1) =====", flush=True)
bands = [("5-12",5,12), ("13-20",13,20), ("21-30",21,30), ("31-40",31,40), ("41-50",41,50)]
print(f"{'Band':>7} {'\chi²_band':>10} {'n':>4} {'\chi²/n':>8}", flush=True)
for label,a,b in bands:
    ch = chi2_eval(C_model, a, b)
    n = b - a + 1
    print(f"{label:>7} {ch:10.3f} {n:4d} {ch/max(1,n):8.3f}", flush=True)
```

 doi:10.20944/preprints202405.2028.v8


```
    # per-\ell table
    print("\n===== PER-\ell TABLE (Cycle 1; \ell=2..50; \kappa frozen) =====", flush=True)
    print("(*) marks \ell in \kappa fit window 5..50.", flush=True)
    print(f"{'\ell':>3} {'*':>1} {'C_obs':>14} {'\sigma_obs':>14} {'C_model':>14} {'z':>9}
    ↪  {'\Delta\chi²=z²':>10}", flush=True)
    dchi_list = []
    for ell in range(2, 51):
        Co, So = get_obs(ell)
        Cm = float(C_model[ell])
        z  = (Cm - Co)/So
        dchi = z*z
        star = "*" if (ELL_FIT_MIN <= ell <= ELL_FIT_MAX) else " "
        print(f"{ell:3d} {star:>1} {Co:14.6e} {So:14.6e} {Cm:14.6e} {z:9.3f} {dchi:10.3f}", flush=True)
        if 5 <= ell <= 50:
            dchi_list.append((ell, float(dchi), float(z), float(Cm)))

    dchi_list.sort(key=lambda t: -t[1])
    print("\nTop \Delta\chi² contributors (Cycle 1; fit window 5..50)", flush=True)
    print(f"{'\ell':>3} {'\Delta\chi²_\ell':>10} {'z':>9} {'C_model':>14}", flush=True)
    for ell, dchi, z, Cm in dchi_list[:25]:
        print(f"{ell:3d} {dchi:10.3f} {z:9.3f} {Cm:14.6e}", flush=True)

    # rotation gate (refit \kappa)
    def fit_for_phi(Phi_in):
        P_r = spectrum_from_energy_residual(Phi_in, EPS_SEED, edges, class_to_id)
        C_r = np.tensordot(T_lsg, P_r, axes=([1,2],[0,1]))
        _, k2_r, chi2_r = kappa_fit(C_r, 5, 50)
        return chi2_r

    chi_main = chi2
    chi_xy = fit_for_phi(rotate_volume_axes(phi3, (1,0,2)))
    chi_yz = fit_for_phi(rotate_volume_axes(phi3, (0,2,1)))
    chi_xz = fit_for_phi(rotate_volume_axes(phi3, (2,1,0)))
    spread = float(max(chi_main, chi_xy, chi_yz, chi_xz) - min(chi_main, chi_xy, chi_yz, chi_xz))

    print("\n[Rotation gate] (Cycle 1)", flush=True)
    print(f"\chi² main={chi_main:.3f} | x\leftrightarrowy={chi_xy:.3f}, y\leftrightarrowz={chi_yz:.3f},
    ↪  x\leftrightarrowz={chi_xz:.3f} | spread={spread:.3f}", flush=True)
    require(spread <= ROT_CHI2_SPREAD_MAX, "Rotation \chi² spread too large.")
    T.beat("rotation gate")

    # \epsilon-linearity probe
    mean_spread, max_spread = eps_linearity_probe(coords, w, PsiK, band_rows, NBOX,
                                                  EPS_CHECK_NSAMPLES, EPS_CHECK_VALUES, 50)
    print("\n[\epsilon-linearity probe] (Cycle 1)", flush=True)
    print(f"Fractional spread across \epsilon: mean={mean_spread:.3e}, max={max_spread:.3e}", flush=True)
    require(mean_spread <= EPS_MEAN_MAX and max_spread <= EPS_MAX_MAX, "\epsilon-linearity spreads too
    ↪  large.")
    T.beat("\epsilon-linearity")

    print(f"\n[done] wall time = {time.perf_counter() - T.t0:.1f}s", flush=True)

if __name__ == "__main__":
    main()
```

 doi:10.20944/preprints202405.2028.v8

Appendix B.3.1.2. Appendix B.6.2. Console transcript

```
DO CMB — Validation I (DO-text t=0 init, \epsilon_seed=1e-15) — \ellmax=50
Efficiency: \Phi is flat 1D; per-tick pairs precomputed; tight Numba loop per tick
NBOX=69, \theta(1)=0.020000, DO_COLLAPSE_EVERY=0
t=0 seeds: EPS_SEED=1.0e-15, SEED_ANISO=20250101
Schedule: R=39 ticks per sweep (13 offsets × 3 modes) | Cycle1 ticks=39 (production)
Cycle2 ticks=78 (verification only) | enabled=False
Shell: R=30, K_EIG=2916 (\ell_cap\approx50+pad=3), BINS=32
\kappa-fit window: \ell=5..50

\Phi(t=0): sha256[:8]=b78dde34
Bell-sum E0_init=3.285090e+05
[t0] mean(|\Phi|^2)=1.0000000000000000e+00  mean(res)=-8.578e-17
[t0] std(res)/mean=2.101e-16  max|res|/mean=1.110e-15
[time] t=0 init: +0.20s | total 0.20s
[time] precompute pairs: +0.97s | total 1.17s
[gen] tick  1/39: r(K)= 1 | Bell drift=0.000e+00
[gen] tick 10/39: r(K)=10 | Bell drift=0.000e+00
[gen] tick 20/39: r(K)=20 | Bell drift=0.000e+00
[gen] tick 30/39: r(K)=30 | Bell drift=0.000e+00
[gen] tick 39/39: r(K)=39 | Bell drift=0.000e+00

\Phi(after cycle1): sha256[:8]=736c80d7
Bell-sum audit: E0=3.285090e+05  E1=3.285090e+05  rel_drift=0.000e+00
[t_end_cycle1] mean(|\Phi|^2)=1.0000000000000000e+00  mean(res)=-8.581e-17
[t_end_cycle1] std(res)/mean=1.765e-16  max|res|/mean=1.332e-15
[time] cycle1 evolution: +2.28s | total 3.45s

===== Guardrails (geometry) =====
shell nodes=21602, weights sum=12.566370614359 (4\pi=12.566370614359) | rel.err=1.414e-16
[time] shell geometry: +3.39s | total 6.84s

[eigensolve] computing K=2916 lowest modes ...
[eigensolve] ||\PsiA^{\mathrm{T}} M \Psi $-$ I||_F = 1.024e-12 | t=10999.8s
[time] eigensolve: +11011.70s | total 11018.54s

[prep]
classes=6205, buckets_used=7744
[time] qmap: +4.99s | total 11023.53s

[calibration] building T (SEEDED-Q full buckets) ...
[time] T calibration: +114.05s | total 11137.58s

===== FIT SUMMARY (Cycle 1 production) =====
\kappa = 2.089482181263314e+13   \chi²(\ell=5–50) = 49.876 over n_pts=46   \chi²/(n_pts-1)=1.108

===== Bandwise \chi² contributions (\kappa frozen; Cycle 1) =====
   Band    \chi²_band    n     \chi²/n
   5–12     18.464     8    2.308
  13–20      4.757     8    0.595
  21–30      3.310    10    0.331
  31–40     12.218    10    1.222
  41–50     11.127    10    1.113

===== PER-\ell TABLE (Cycle 1; \ell=2..50; \kappa frozen) =====
```

 doi:10.20944/preprints202405.2028.v8

(*) marks \ell in \kappa fit window 5..50.

\ell	*	C_obs	\sigma_obs	C_model	z	\Delta\chi²=z²
2		6.371975e-11	2.993180e-10	8.908343e-11	0.085	0.007
3		3.757952e-10	2.353983e-10	8.357874e-11	-1.241	1.541
4		2.097323e-10	1.355616e-10	8.630735e-11	-0.910	0.829
5	*	3.780877e-10	1.055243e-10	8.463421e-11	-2.781	7.733
6	*	1.585670e-10	8.444205e-11	8.442203e-11	-0.878	0.771
7	*	2.677040e-10	7.081953e-11	8.506053e-11	-2.579	6.651
8	*	1.158792e-10	6.020004e-11	8.529687e-11	-0.508	0.258
9	*	1.070531e-10	5.238777e-11	8.636957e-11	-0.395	0.156
10	*	1.388228e-10	4.284038e-11	8.932857e-11	-1.155	1.335
11	*	1.345381e-10	3.776291e-11	8.871242e-11	-1.214	1.473
12	*	9.810913e-11	3.624853e-11	8.739637e-11	-0.296	0.087
13	*	1.131737e-10	3.325981e-11	8.619288e-11	-0.811	0.658
14	*	1.020729e-10	3.108198e-11	8.553932e-11	-0.532	0.283
15	*	1.277477e-10	2.841248e-11	8.456297e-11	-1.520	2.310
16	*	8.742653e-11	2.920708e-11	8.353835e-11	-0.133	0.018
17	*	1.152576e-10	2.732701e-11	8.386029e-11	-1.149	1.320
18	*	8.023027e-11	2.615252e-11	8.302540e-11	0.107	0.011
19	*	9.230326e-11	2.510391e-11	8.377455e-11	-0.340	0.115
20	*	7.651180e-11	2.447305e-11	8.148608e-11	0.203	0.041
21	*	6.577979e-11	2.362083e-11	8.300530e-11	0.729	0.532
22	*	6.058016e-11	2.245348e-11	8.168142e-11	0.940	0.883
23	*	7.057404e-11	2.155320e-11	8.109804e-11	0.488	0.238
24	*	7.610091e-11	2.072381e-11	8.050426e-11	0.212	0.045
25	*	6.643222e-11	2.075658e-11	8.015239e-11	0.661	0.437
26	*	7.658001e-11	2.019592e-11	8.011302e-11	0.175	0.031
27	*	7.705872e-11	1.872003e-11	7.838957e-11	0.071	0.005
28	*	9.224762e-11	1.746402e-11	7.928334e-11	-0.742	0.551
29	*	8.635307e-11	1.703794e-11	7.744346e-11	-0.523	0.273
30	*	6.934708e-11	1.693556e-11	7.883539e-11	0.560	0.314
31	*	7.896162e-11	1.654951e-11	7.775123e-11	-0.073	0.005
32	*	1.046747e-10	1.584769e-11	7.620962e-11	-1.796	3.226
33	*	6.971541e-11	1.573058e-11	7.665153e-11	0.441	0.194
34	*	6.677449e-11	1.515098e-11	7.562096e-11	0.584	0.341
35	*	5.700753e-11	1.468349e-11	7.538046e-11	1.251	1.566
36	*	6.273364e-11	1.416504e-11	7.395141e-11	0.792	0.627
37	*	8.080809e-11	1.354318e-11	7.443945e-11	-0.470	0.221
38	*	7.631162e-11	1.319514e-11	7.329431e-11	-0.229	0.052
39	*	7.849050e-11	1.292744e-11	7.198556e-11	-0.503	0.253
40	*	1.015036e-10	1.247094e-11	7.164799e-11	-2.394	5.731
41	*	6.266490e-11	1.225876e-11	7.050133e-11	0.639	0.409
42	*	7.400832e-11	1.203643e-11	7.059375e-11	-0.284	0.080
43	*	6.635391e-11	1.187754e-11	6.978908e-11	0.289	0.084
44	*	6.328508e-11	1.169798e-11	6.871604e-11	0.464	0.216
45	*	5.380897e-11	1.154021e-11	6.858349e-11	1.280	1.639
46	*	4.857735e-11	1.131740e-11	6.705621e-11	1.633	2.666
47	*	7.537541e-11	1.106781e-11	6.738496e-11	-0.722	0.521
48	*	4.103847e-11	1.090678e-11	6.635170e-11	2.321	5.386
49	*	6.836774e-11	1.073460e-11	6.538659e-11	-0.278	0.077
50	*	6.750351e-11	1.052116e-11	6.517858e-11	-0.221	0.049

Top \Delta\chi² contributors (Cycle 1; fit window 5..50)

\ell	\Delta\chi²_\ell	z	C_model
5	7.733	-2.781	8.463421e-11

 doi:10.20944/preprints202405.2028.v8


```
  7      6.651    -2.579   8.506053e-11
 40      5.731    -2.394   7.164799e-11
 48      5.386     2.321   6.635170e-11
 32      3.226    -1.796   7.620962e-11
 46      2.666     1.633   6.705621e-11
 15      2.310    -1.520   8.456297e-11
 45      1.639     1.280   6.858349e-11
 35      1.566     1.251   7.538046e-11
 11      1.473    -1.214   8.871242e-11
 10      1.335    -1.155   8.932857e-11
 17      1.320    -1.149   8.386029e-11
 22      0.883     0.940   8.168142e-11
  6      0.771    -0.878   8.442203e-11
 13      0.658    -0.811   8.619288e-11
 36      0.627     0.792   7.395141e-11
 28      0.551    -0.742   7.928334e-11
 21      0.532     0.729   8.300530e-11
 47      0.521    -0.722   6.738496e-11
 25      0.437     0.661   8.015239e-11
 41      0.409     0.639   7.050133e-11
 34      0.341     0.584   7.562096e-11
 30      0.314     0.560   7.883539e-11
 14      0.283    -0.532   8.553932e-11
 29      0.273    -0.523   7.744346e-11

[Rotation gate] (Cycle 1)
\chi² main=49.876 | x\leftrightarrowy=49.876, y\leftrightarrowz=49.876, x\leftrightarrowz=49.876 |
↪  spread=0.000
[time] rotation gate: +3.52s | total 11141.10s

[\epsilon-linearity probe] (Cycle 1)
Fractional spread across \epsilon: mean=4.163e-05, max=1.485e-03
[time] \epsilon-linearity: +3.31s | total 11144.41s

[done] wall time = 11144.4s
```

Display convention. Where a D_ℓ display is used elsewhere, $D_\ell := \frac{\ell(\ell+1)}{2\pi} C_\ell$. This validation reports C_ℓ directly in the transcript.

Appendix C.2. Appendix B.7. Validation II: CHSH (Tsirelson bound)

Cross-reference. Appendix B, Validation II (Sec. B.2).

Appendix B.3.2.1. Appendix B.7.1. Executable script

```
import numpy as np

# ============================================================================
# 1. CONFIGURATION AND MODEL PARAMETERS
# ============================================================================

# The number of Monte Carlo samples used to numerically integrate over all
# possible orientations of the entangled state. This represents a statistical
# sample of the vast population of Bell Spheres constituting the single system.
N_SAMPLES = 50000

# Canonical CHSH measurement angles in degrees.
```

 doi:10.20944/preprints202405.2028.v8


```
ANGLES_DEG = {
    'A': 0.0,
    'A_prime': 90.0,
    'B': 45.0,
    'B_prime': 135.0
}

# Convert angles to radians to define measurement orientations in the
# abstract 2D computational space.
MEASUREMENT_AXES = {
    key: np.array([np.cos(np.deg2rad(val)), np.sin(np.deg2rad(val))])
    for key, val in ANGLES_DEG.items()
}

# ==============================================================================
# 2. DO MODEL PRINCIPLES IMPLEMENTED AS FUNCTIONS
# ==============================================================================

def initialize_singlet_state(n_samples):
    """
    Implements Principle 1: Models the indeterminate orientation of the single
    entangled state by creating a statistical sample of possible orientations.
    """
    random_angles = np.random.uniform(0, 2 * np.pi, n_samples)
    s_alice = np.vstack([np.cos(random_angles), np.sin(random_angles)]).T
    return s_alice

def calculate_correlation_from_first_principles(s_alice, m_alice, m_bob):
    """
    Calculates the quantum correlation E(A, B) by applying DO principles.
    """
    # Principle 3: The Measurement Amplification Constant (d).
    # Per the model's d = \nu rule, this is set to 2 because the planar
    # interaction has 2 degrees of freedom (\nu = 2).
    AMPLIFICATION_FACTOR_D = 2.0

    # Principle 2: Measurement as Relational Alignment.
    # The dot product calculates the scalar alignment value for each sample.
    # Note: s_bob = -s_alice is used implicitly in the second term.
    local_projections_A = np.einsum('ij,j->i', s_alice, m_alice)
    local_projections_B = np.einsum('ij,j->i', -s_alice, m_bob)

    # The final correlation is the estimated physical average (the sample mean)
    # amplified by the required factor d.
    correlation_E = AMPLIFICATION_FACTOR_D * np.mean(local_projections_A * local_projections_B)
    return correlation_E

# ==============================================================================
# 3. MAIN SIMULATION EXECUTION
# ==============================================================================

print("Starting DO First-Principles CHSH Bell Test Simulation...")
print(f"Estimating physical average with {N_SAMPLES} Monte Carlo samples.\n")

# Initialize the sample space for the singlet state once.
```

 doi:10.20944/preprints202405.2028.v8


```
s_alice_samples = initialize_singlet_state(N_SAMPLES)

# Define the four measurement settings for the CHSH test.
measurement_settings = {
    'AB':   (MEASUREMENT_AXES['A'], MEASUREMENT_AXES['B']),
    'ABp':  (MEASUREMENT_AXES['A'], MEASUREMENT_AXES['B_prime']),
    'ApB':  (MEASUREMENT_AXES['A_prime'], MEASUREMENT_AXES['B']),
    'ApBp': (MEASUREMENT_AXES['A_prime'], MEASUREMENT_AXES['B_prime'])
}

# Calculate the correlation E for each setting from first principles.
E_values = {}
for setting, (m_a, m_b) in measurement_settings.items():
    E_values[setting] = calculate_correlation_from_first_principles(
        s_alice_samples, m_a, m_b
    )

# Calculate the final CHSH S-value using the standard formula.
S = E_values['AB'] - E_values['ABp'] + E_values['ApB'] + E_values['ApBp']

# ==============================================================================
# 4. RESULTS AND CONCLUSION
# ==============================================================================

print("--- Explicit DO-Compliant CHSH Bell Test Results ---")
print("Correlations derived directly from the DO model's first principles:\n")

# Theoretical expected QM correlation is E(a,b) = -cos(theta_a - theta_b)
th_AB   = -np.cos(np.deg2rad(ANGLES_DEG['A'] - ANGLES_DEG['B']))
th_ABp  = -np.cos(np.deg2rad(ANGLES_DEG['A'] - ANGLES_DEG['B_prime']))
th_ApB  = -np.cos(np.deg2rad(ANGLES_DEG['A_prime'] - ANGLES_DEG['B']))
th_ApBp = -np.cos(np.deg2rad(ANGLES_DEG['A_prime'] - ANGLES_DEG['B_prime']))

print(f"Correlation E(A, B)      = {E_values['AB']:.4f} (Theoretical: {th_AB:.4f})")
print(f"Correlation E(A, B')     = {E_values['ABp']:.4f} (Theoretical: {th_ABp:.4f})")
print(f"Correlation E(A', B)     = {E_values['ApB']:.4f} (Theoretical: {th_ApB:.4f})")
print(f"Correlation E(A', B')    = {E_values['ApBp']:.4f} (Theoretical: {th_ApBp:.4f})")

print("\n------------------------------------------------------")
print(f"Final CHSH S-value = {S:.4f}")
print("------------------------------------------------------")
print("\nClassical Limit (Local Realism) <= 2.0")
print("Quantum Maximum (Tsirelson's Bound) \u2248 2.8284\n")
```

Appendix B.3.2.2. Appendix B.7.2. Console transcript

```
Starting DO First-Principles CHSH Bell Test Simulation...
Estimating physical average with 50000 Monte Carlo samples.

--- Explicit DO-Compliant CHSH Bell Test Results ---
Correlations derived directly from the DO model's first principles:

Correlation E(A, B)      = -0.7052 (Theoretical: -0.7071)
Correlation E(A, B')     =  0.7125 (Theoretical:  0.7071)
Correlation E(A', B)     = -0.7017 (Theoretical: -0.7071)
Correlation E(A', B')    = -0.7090 (Theoretical: -0.7071)
```

 doi:10.20944/preprints202405.2028.v8


```
--------------------------------------------------
Final CHSH S-value = -2.8284
--------------------------------------------------

Classical Limit (Local Realism) <= 2.0
Quantum Maximum (Tsirelson's Bound) \approx 2.8284
```

Appendix C.3. Appendix B.8. Validation III–A1: Earth–Moon two-body baseline (1-year)

Cross-reference. Appendix B, Validation III–A1.

Appendix B.3.3.1. Appendix B.8.1. Executable script

```
import numpy as np
import matplotlib.pyplot as plt
from numba import njit
import math
import time
from mpl_toolkits.mplot3d import Axes3D  # noqa: F401

# ============================================================
# 1-YEAR EARTH-MOON ANALOG (DO-NATIVE, FLUX KERNEL, H_DO)
# Candidate A flux kernel + Manhattan direction + FR + H_DO
# ============================================================

# ---------------------------
# USER CONTROLS
# ---------------------------

DT          = 1.0e-2        # time step

# Discrete geometry / orbit setup
D0          = 30.0          # Chebyshev launch radius (Earth-Moon analog)
R_MAX       = 200           # Max Chebyshev radius for kernel

# Low-mass Earth-Moon DO parameters
M1, M2      = 81.297, 1.0   # "Earth" and "Moon" masses in DO units

# Gravitational coupling and flux kernel parameter
CHI_FLUX    = 24.0
CHI_E       = 0.00057032984  # base DO coupling (no high-energy scaling)

# Tangential velocity factor (to be tuned by hand)
V_FACTOR    = 0.958454       # adjust for defect / eccentricity

# Fixed steps per "year" (to be tuned by hand)
STEPS_YEAR  = 443_850
N_YEARS     = 1
STEPS_TOTAL = N_YEARS * STEPS_YEAR

# Sampling: target number of samples per orbit
TARGET_SAMPLES_PER_YEAR = 4000  # -> ~4000 samples per year

# Optional target eccentricity (for reporting only)
E_TARGET = 0.0  # mildly-elliptical Earth-Moon analog
```

 doi:10.20944/preprints202405.2028.v8


```
# ---------------------------
# 1. Ellipse fit helper
# ---------------------------

def fit_ellipse_parameters(x, y):
    """
    Fit a general conic Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0
    and extract ellipse parameters (e, a). Returns (e,a).
    Returns (0.0, 0.0) if the conic is not an ellipse.
    """
    x = np.asarray(x)
    y = np.asarray(y)
    if x.size < 20:
        return 0.0, 0.0

    D = np.vstack([x**2, x*y, y**2, x, y, np.ones_like(x)]).T
    A_mat, b_vec = D[:, :-1], -np.ones(len(x))
    A, B, C, D_c, E = np.linalg.lstsq(A_mat, b_vec, rcond=None)[0]
    F = 1.0

    delta = B**2 - 4.0 * A * C
    if delta >= 0.0:
        return 0.0, 0.0

    # Center of ellipse
    x0 = (2.0 * C * D_c - B * E) / delta
    y0 = (2.0 * A * E - B * D_c) / delta

    # Semi-axes from standard conic formulas
    num = 2.0 * (A*x0**2 + C*y0**2 + B*x0*y0 - F)
    tmp = math.sqrt((A - C)**2 + B**2)

    den1 = (A + C) + tmp
    den2 = (A + C) - tmp
    if den1 == 0.0 or den2 == 0.0:
        return 0.0, 0.0

    a = math.sqrt(abs(num / den2))
    b = math.sqrt(abs(num / den1))
    if a < b:
        a, b = b, a

    e = math.sqrt(1.0 - (b / a)**2)
    return e, a

# ---------------------------
# 2. Candidate-A flux kernel
# ---------------------------

def build_flux_kernel(r_max, chi_flux):
    """
    Candidate-A radial kernel:
      N_d = 24 d^2 + 2 (for d>=1), N_0 = 1
      rho_0 = 1, rho_{d>0} = 0
```

 doi:10.20944/preprints202405.2028.v8


```
      Phi(d) = sum_{k<=d} N_k * rho_k  (monopole at d=0)
      gamma(d) = -chi_flux * Phi(d) / N_d (for d>=1), gamma(0)=0
      w(d)-w(d+1) = -2 gamma(d+1), with w(R_max)=0
    Returns arrays w[d], gamma[d] for d=0..r_max
    """
    k = np.arange(1, r_max + 1, dtype=np.float64)

    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k * k + 2.0  # N_d

    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0

    Phi = np.cumsum(counts * rho)

    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])

    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
        w_arr[d] = s

    return w_arr, gamma_arr

@njit(fastmath=True)
def get_field(d, w_arr, gamma_arr):
    """
    Interpolate w(d), gamma(d) for real d between 0 and r_max.
    """
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]
    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0

    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1

    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]

    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w

    return w_val, g_val

# ---------------------------
# 3. DO-native state: Manhattan direction + Candidate A
# ---------------------------
```

 doi:10.20944/preprints202405.2028.v8


```
@njit(fastmath=True)
def compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr):
    """
    Given positions pos[2,3], compute:
      - accelerations on each body (acc[2,3])
      - potential energy pe
      - force magnitude f_mag
      - Manhattan-fraction unit direction u (graph-based)
    """
    r_vec = pos[0] - pos[1]

    ax = abs(r_vec[0])
    ay = abs(r_vec[1])
    az = abs(r_vec[2])

    # Chebyshev radius
    d_inf = ax
    if ay > d_inf:
        d_inf = ay
    if az > d_inf:
        d_inf = az

    w_val, g_val = get_field(d_inf, w_arr, gamma_arr)

    # Manhattan distance
    r1 = ax + ay + az

    # Graph-based direction (L1 normalized, with signs)
    u = np.zeros(3, dtype=np.float64)
    if r1 > 0.0:
        fx = ax / r1
        fy = ay / r1
        fz = az / r1

        if r_vec[0] > 0.0:
            u[0] = fx
        elif r_vec[0] < 0.0:
            u[0] = -fx

        if r_vec[1] > 0.0:
            u[1] = fy
        elif r_vec[1] < 0.0:
            u[1] = -fy

        if r_vec[2] > 0.0:
            u[2] = fz
        elif r_vec[2] < 0.0:
            u[2] = -fz

    f_mag = chi_e * m1 * m2 * g_val

    acc = np.zeros((2, 3), dtype=np.float64)
    inv_m1 = 1.0 / m1
    inv_m2 = 1.0 / m2
```

 doi:10.20944/preprints202405.2028.v8


```
    acc[0, 0] =  f_mag * inv_m1 * u[0]
    acc[0, 1] =  f_mag * inv_m1 * u[1]
    acc[0, 2] =  f_mag * inv_m1 * u[2]

    acc[1, 0] = -f_mag * inv_m2 * u[0]
    acc[1, 1] = -f_mag * inv_m2 * u[1]
    acc[1, 2] = -f_mag * inv_m2 * u[2]

    pe = -chi_e * m1 * m2 * w_val

    return acc, pe, f_mag, u

# ---------------------------
# 4. FR + H_DO integrator with sampling
# ---------------------------

@njit(fastmath=True)
def run_orbit(
    dt, steps, v_factor, d0,
    m1, m2, chi_e, w_arr, gamma_arr,
    sample_stride
):
    """
    DO-native 2-body FR+H_DO integrator for 'steps' major steps.
    Samples every 'sample_stride' steps and returns:
      t_hist, x_hist, y_hist, z_hist, r_hist,
      E_naive_hist, H_DO_hist, Lz_hist,
      pos1_hist, pos2_hist
    """
    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    # Initial circular speed at radius d0 for this kernel
    w0, g0 = get_field(d0, w_arr, gamma_arr)
    f_mag_init = chi_e * m1 * m2 * abs(g0)
    mu = (m1 * m2) / (m1 + m2)
    v_circ = math.sqrt(f_mag_init * d0 / mu)
    v_init = v_factor * v_circ

    # Initial positions: body 2 at +D0 on x-axis, then shift for COM at 0
    pos = np.zeros((2, 3), dtype=np.float64)
    pos[1, 0] = d0
    Mtot = m1 + m2
    comx = (pos[0, 0] * m1 + pos[1, 0] * m2) / Mtot
    pos[0, 0] -= comx
    pos[1, 0] -= comx

    # Initial velocities: tangential +/-v_init, COM at rest (y-direction)
    vel = np.zeros((2, 3), dtype=np.float64)
    vel[1, 1] = v_init
```

 doi:10.20944/preprints202405.2028.v8


```
mass_ratio = m2 / m1
vel[0, 1] = -mass_ratio * v_init

# Sampling arrays
n_samples = steps // sample_stride + 1
t_hist    = np.zeros(n_samples, dtype=np.float64)
x_hist    = np.zeros(n_samples, dtype=np.float64)
y_hist    = np.zeros(n_samples, dtype=np.float64)
z_hist    = np.zeros(n_samples, dtype=np.float64)
r_hist    = np.zeros(n_samples, dtype=np.float64)
E_hist    = np.zeros(n_samples, dtype=np.float64)  # naive K+U
H_hist    = np.zeros(n_samples, dtype=np.float64)  # H_DO
Lz_hist   = np.zeros(n_samples, dtype=np.float64)
pos1_hist = np.zeros((n_samples, 3), dtype=np.float64)
pos2_hist = np.zeros((n_samples, 3), dtype=np.float64)

work_defect = 0.0

# Initial energies
acc_s, pe_s, fmag_s, u_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
ke = 0.5 * (m1 * (vel[0, 0]**2 + vel[0, 1]**2 + vel[0, 2]**2)
            + m2 * (vel[1, 0]**2 + vel[1, 1]**2 + vel[1, 2]**2))
r_vec = pos[1] - pos[0]
r2 = r_vec[0]**2 + r_vec[1]**2 + r_vec[2]**2
r = math.sqrt(r2)
Lz = (pos[0, 0] * (m1 * vel[0, 1]) - pos[0, 1] * (m1 * vel[0, 0])
      + pos[1, 0] * (m2 * vel[1, 1]) - pos[1, 1] * (m2 * vel[1, 0]))

E_naive = ke + pe_s
H_DO    = E_naive - work_defect

# First sample
t_hist[0]  = 0.0
x_hist[0]  = r_vec[0]
y_hist[0]  = r_vec[1]
z_hist[0]  = r_vec[2]
r_hist[0]  = r
E_hist[0]  = E_naive
H_hist[0]  = H_DO
Lz_hist[0] = Lz
pos1_hist[0, :] = pos[0, :]
pos2_hist[0, :] = pos[1, :]

sample_idx = 1

# Main FR loop
for step in range(1, steps + 1):
    for i in range(4):
        # start-of-drift
        acc_s, pe_s, fmag_s, u_s = compute_state_DOnative(
            pos, m1, m2, chi_e, w_arr, gamma_arr
        )
        F_s = np.zeros(3, dtype=np.float64)
        F_s[0] = fmag_s * u_s[0]
        F_s[1] = fmag_s * u_s[1]
```

 doi:10.20944/preprints202405.2028.v8


```
        F_s[2] = fmag_s * u_s[2]

        # drift
        dr = c[i] * dt * vel
        pos = pos + dr

        # end-of-drift
        acc_e, pe_e, fmag_e, u_e = compute_state_DOnative(
            pos, m1, m2, chi_e, w_arr, gamma_arr
        )
        F_e = np.zeros(3, dtype=np.float64)
        F_e[0] = fmag_e * u_e[0]
        F_e[1] = fmag_e * u_e[1]
        F_e[2] = fmag_e * u_e[2]

        F_avg = 0.5 * (F_s + F_e)

        dr_rel = np.zeros(3, dtype=np.float64)
        dr_rel[0] = dr[0, 0] - dr[1, 0]
        dr_rel[1] = dr[0, 1] - dr[1, 1]
        dr_rel[2] = dr[0, 2] - dr[1, 2]

        W_mech  = F_avg[0] * dr_rel[0] + F_avg[1] * dr_rel[1] + F_avg[2] * dr_rel[2]
        delta_U = pe_e - pe_s
        work_defect += (W_mech + delta_U)

        # kick
        if dcoef[i] != 0.0:
            vel = vel + dcoef[i] * dt * acc_e

    # sampling
    if step % sample_stride == 0:
        ke = 0.5 * (m1 * (vel[0, 0]**2 + vel[0, 1]**2 + vel[0, 2]**2)
                    + m2 * (vel[1, 0]**2 + vel[1, 1]**2 + vel[1, 2]**2))
        acc_c, pe_c, fmag_c, u_c = compute_state_DOnative(
            pos, m1, m2, chi_e, w_arr, gamma_arr
        )
        r_vec = pos[1] - pos[0]
        r2 = r_vec[0]**2 + r_vec[1]**2 + r_vec[2]**2
        r = math.sqrt(r2)
        Lz = (pos[0, 0] * (m1 * vel[0, 1]) - pos[0, 1] * (m1 * vel[0, 0])
              + pos[1, 0] * (m2 * vel[1, 1]) - pos[1, 1] * (m2 * vel[1, 0]))
        E_naive = ke + pe_c
        H_DO    = E_naive - work_defect

        t_hist[sample_idx]  = step * dt
        x_hist[sample_idx]  = r_vec[0]
        y_hist[sample_idx]  = r_vec[1]
        z_hist[sample_idx]  = r_vec[2]
        r_hist[sample_idx]  = r
        E_hist[sample_idx]  = E_naive
        H_hist[sample_idx]  = H_DO
        Lz_hist[sample_idx] = Lz
        pos1_hist[sample_idx, :] = pos[0, :]
        pos2_hist[sample_idx, :] = pos[1, :]
```

 doi:10.20944/preprints202405.2028.v8


```
            sample_idx += 1

    return (t_hist[:sample_idx], x_hist[:sample_idx], y_hist[:sample_idx],
            z_hist[:sample_idx], r_hist[:sample_idx],
            E_hist[:sample_idx], H_hist[:sample_idx],
            Lz_hist[:sample_idx],
            pos1_hist[:sample_idx, :], pos2_hist[:sample_idx, :])

# ----------------------------
# 5. Main: run 1-year orbit, analyze & plot
# ----------------------------

if __name__ == "__main__":
    print("Building Candidate-A flux kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)

    print("Warming up Numba JIT...")
    _ = run_orbit(DT, 1000, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, 10)
    print("Warm-up done.")

    sample_stride = max(1, STEPS_YEAR // TARGET_SAMPLES_PER_YEAR)
    print(f"\nRunning 1-year Earth-Moon simulation: "
          f"STEPS_TOTAL = {STEPS_TOTAL}, sample_stride = {sample_stride}")
    t0 = time.time()

    (t_hist, x_hist, y_hist, z_hist, r_hist,
     E_hist, H_hist, Lz_hist,
     pos1_hist, pos2_hist) = run_orbit(
        DT, STEPS_TOTAL, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, sample_stride
    )

    t1 = time.time()
    print(f"Simulation complete. Elapsed time: {t1 - t0:.2f} s")

    # Relative orbit and basic stats
    x_all = x_hist
    y_all = y_hist
    r_all = r_hist

    # Ellipse fit & e_rmin on full 1-year track
    e_fit, a_fit = fit_ellipse_parameters(x_all, y_all)
    r_min = float(r_all.min())
    r_max = float(r_all.max())
    e_rmin = (r_max - r_min) / (r_max + r_min)

    # Energy & Lz diagnostics over that 1-year run
    E0   = E_hist[0]
    H0   = H_hist[0]
    Lz0  = Lz_hist[0]

    dE_naive = (E_hist - E0) / abs(E0)
    dH = (H_hist - H0) / abs(H0)
    dLz = (Lz_hist - Lz0) / abs(Lz0)

    H_net_drift   = dH[-1]
```

 doi:10.20944/preprints202405.2028.v8


```
H_rms         = np.std(dH)
E_net_drift   = dE_naive[-1]
E_rms         = np.std(dE_naive)
Lz_rms        = np.std(dLz)
e_target_diff = abs(e_fit - E_TARGET)

# COM and per-body radial metrics
Mtot = M1 + M2
COM = (pos1_hist * M1 + pos2_hist * M2) / Mtot
COM_drift = np.linalg.norm(COM - COM[0], axis=1)
max_COM_drift = float(COM_drift.max())

r_E = np.linalg.norm(pos1_hist - COM, axis=1)
r_M = np.linalg.norm(pos2_hist - COM, axis=1)

# Relative phase and delta-theta
rel_angles = np.arctan2(y_all, x_all)
phase = np.unwrap(rel_angles)
delta_theta = np.degrees(phase[-1] - phase[0])

# Periapsis angle (single 1-year track)
k_min = int(np.argmin(r_all))
x_min = x_all[k_min]
y_min = y_all[k_min]
omega = math.atan2(y_min, x_min)
omega_deg = math.degrees(omega)

# ------------------------
# Printed metrics (IV + Mercury style)
# ------------------------
print("\n1-YEAR EARTH-MOON DO-NATIVE ORBIT (FLUX KERNEL, H_DO)")
print("------------------------------------------------------")
print(f"{'Parameter':<30} | {'Value':>20}")
print("-" * 55)
print(f"{'D0 (hops)':<30} | {D0:>20.6f}")
print(f"{'DT':<30} | {DT:>20.6e}")
print(f"{'V_FACTOR':<30} | {V_FACTOR:>20.6f}")
print(f"{'STEPS_YEAR':<30} | {STEPS_YEAR:>20d}")
print(f"{'STEPS_TOTAL':<30} | {STEPS_TOTAL:>20d}")

print("\n--- Two-Body Dynamics (1 Year / DO-native) ---")
print(f"{'Delta Theta (deg)':<30} | {delta_theta:>20.6f}")
print(f"{'Max COM Drift (hops)':<30} | {max_COM_drift:>20.6e}")
print(f"{'H_DO net drift':<30} | {H_net_drift:>20.3e}")
print(f"{'H_DO RMS':<30} | {H_rms:>20.3e}")
print(f"{'Naive E net drift':<30} | {E_net_drift:>20.3e}")
print(f"{'Naive E RMS':<30} | {E_rms:>20.3e}")
print(f"{'Lz RMS':<30} | {Lz_rms:>20.3e}")

print("\n--- Per-Body Radial Metrics (Hops, barycentric) ---")
print(f"{'Body':<10} | {'r_mean':>12} | {'r_std':>12}")
print("-" * 40)
print(f"{'Earth':<10} | {np.mean(r_E):>12.6f} | {np.std(r_E):>12.6f}")
print(f"{'Moon':<10} | {np.mean(r_M):>12.6f} | {np.std(r_M):>12.6f}")
```

 doi:10.20944/preprints202405.2028.v8


```
print("\n--- Ellipse / Shape Diagnostics (Moon-Earth relative orbit) ---")
print(f"{'e_fit (conic fit)':<30} | {e_fit:>20.6f}")
print(f"{'a_fit (semi-major, hops)':<30} | {a_fit:>20.6f}")
print(f"{'e_rmin from r_min/r_max':<30} | {e_rmin:>20.6f}")
print(f"{'r_min (hops)':<30} | {r_min:>20.6f}")
print(f"{'r_max (hops)':<30} | {r_max:>20.6f}")
print(f"{'|e_fit - E_target|':<30} | {e_target_diff:>20.6e}")
print(f"{'periapsis omega (deg)':<30} | {omega_deg:>20.6f}")

# ------------------------
# Plots
# ------------------------

# 3D trajectory: Earth, Moon, COM
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
ax.plot(pos1_hist[:, 0], pos1_hist[:, 1], pos1_hist[:, 2], '-r', label='Earth')
ax.plot(pos2_hist[:, 0], pos2_hist[:, 1], pos2_hist[:, 2], '-b', label='Moon')
ax.plot(COM[:, 0], COM[:, 1], COM[:, 2], 'k--', label='COM')
ax.set_title('1-Year Earth-Moon Orbit (DO-native flux kernel)')
ax.set_xlabel('X (hops)')
ax.set_ylabel('Y (hops)')
ax.set_zlabel('Z (hops)')
ax.legend()
plt.tight_layout()
plt.show()

# 2D relative orbit in x-y
plt.figure(figsize=(6, 6))
plt.plot(x_all, y_all, lw=0.8)
plt.scatter([0.0], [0.0], label='Earth (relative frame)')
plt.xlabel('x (relative)')
plt.ylabel('y (relative)')
plt.title('Earth-Moon relative orbit (1-year track, DO-native)')
plt.axis('equal')
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

# Radial distance vs time
plt.figure(figsize=(7, 4))
plt.plot(t_hist, r_all, lw=0.8)
plt.xlabel('t')
plt.ylabel('r (hops)')
plt.title('Moon-Earth separation r(t) over 1 DO-year')
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

# Energy diagnostics (H_DO and naive E)
plt.figure(figsize=(7, 4))
plt.plot(t_hist, dH, label="\DeltaH_DO / H0")
plt.plot(t_hist, dE_naive, label="\DeltaE_naive / E0", alpha=0.7)
plt.xlabel('t')
```

 doi:10.20944/preprints202405.2028.v8


```
    plt.ylabel('\DeltaE / E0')
    plt.title('Energy diagnostics over 1 DO-year (Earth-Moon)')
    plt.legend()
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # Lz diagnostics
    plt.figure(figsize=(7, 4))
    plt.plot(t_hist, dLz, lw=0.8)
    plt.xlabel('t')
    plt.ylabel('\DeltaLz / Lz0')
    plt.title('Angular momentum diagnostics over 1 DO-year (Earth-Moon)')
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()
```

Appendix B.3.3.2. Appendix B.8.2. Console transcript

```
Building Candidate-A flux kernel...
Warming up Numba JIT...
Warm-up done.

Running 1-year Earth-Moon simulation: STEPS_TOTAL = 443850, sample_stride = 110
Simulation complete. Elapsed time: 1.61 s

1-YEAR EARTH-MOON DO-NATIVE ORBIT (FLUX KERNEL, H_DO)
------------------------------------------------------
Parameter                      |                Value
------------------------------------------------------
D0 (hops)                      |            30.000000
DT                             |         1.000000e-02
V_FACTOR                       |             0.958454
STEPS_YEAR                     |               443850
STEPS_TOTAL                    |               443850

--- Two-Body Dynamics (1 Year / DO-native) ---
Delta Theta (deg)              |           360.000208
Max COM Drift (hops)           |         2.402499e-13
H_DO net drift                 |            5.880e-13
H_DO RMS                       |            4.128e-12
Naive E net drift              |            2.378e-06
Naive E RMS                    |            1.960e-01
Lz RMS                         |            3.069e-14

--- Per-Body Radial Metrics (Hops, barycentric) ---
Body       |       r_mean |        r_std
-----------------------------------------
Earth      |     0.347484 |     0.014862
Moon       |    28.249416 |     1.208257

--- Ellipse / Shape Diagnostics (Moon-Earth relative orbit) ---
e_fit (conic fit)              |             0.054875
a_fit (semi-major, hops)       |            28.545052
e_rmin from r_min/r_max        |             0.061861
r_min (hops)                   |            26.638620
```

 doi:10.20944/preprints202405.2028.v8


```
r_max (hops)                  |            30.151742
|e_fit - E_target|            |         5.487468e-02
periapsis omega (deg)         |          -179.792378
```

Appendix B.3.3.3. Appendix B.9. Validation III–A2: Earth–Moon Two-Body Long Horizon (1000-Orbit)

Cross-reference. Appendix B, Validation III–A2.

Appendix B.3.3.4. Appendix B.9.1. Executable Script

```
import numpy as np
import math
import time
from numba import njit
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D  # noqa: F401

# ============================================================
# EARTH-MOON 1000-ORBIT TILTED RUN (DO-NATIVE, DIAGNOSTICS-HEAVY)
#
# Dynamics unchanged vs your 100-orbit tilted run:
#   Candidate-A flux kernel + Manhattan (L1) direction + Forest-Ruth
#
# Primary conserved quantity:
#   I_DO = (K + U) - W_closure
#
# Diagnostics improvements (no dynamics change):
#   • Multiple apsis/orientation estimators (peri, harmonic, axis/PCA)
#   • Conditioning metrics (e_rmin, e_harm, cond_ratio ~ 1 => near-circular)
#   • Robust stats + block summaries (exclude first orbit in each block)
#   • Selected-orbit table only (no full spam)
#   • Full plot suite (3D/2D/r(t)/invariants/e_rmin/\Delta\omega/inc/node)
# ============================================================

# ---------------------------
# TUNABLE PARAMETERS (ALL HERE)
# ---------------------------

# Integrator / orbit setup
DT                = 3.0e-3
D0                = 30.0
R_MAX             = 200

# Masses (Earth, Moon) in DO units
M1, M2            = 81.297, 1.0

# DO kernel / coupling
CHI_FLUX          = 24.0
CHI_E             = 0.00057032984

# Launch tuning (from your tilted 1-orbit calibration)
V_FACTOR          = 0.939199

# Tilt (about x-axis)
INCL_DEG          = 5.145
INCL_RAD          = INCL_DEG * math.pi / 180.0
```

 doi:10.20944/preprints202405.2028.v8


```
# Horizon
STEPS_PER_ORBIT   = 1_499_675
N_ORBITS_TARGET   = 1000
EXTRA_ORBITS_RUN  = 5
STEPS_TOTAL       = STEPS_PER_ORBIT * (N_ORBITS_TARGET + EXTRA_ORBITS_RUN)

# Sampling
SAMPLES_PER_ORBIT = 8000
SAMPLES_MAX       = SAMPLES_PER_ORBIT * (N_ORBITS_TARGET + EXTRA_ORBITS_RUN)

# Printing policy
PRINT_FIRST_N     = 10
PRINT_EVERY_N     = 20
PRINT_LAST_N      = 10

# Block summaries
BLOCK_SIZE        = 100

# Conditioning thresholds (diagnostics only)
E_HARM_MIN_FOR_OMEGA  = 0.020   # below this, \omega is typically ill-conditioned
COND_RATIO_MAX_FOR_OMEGA = 0.995 # cond_ratio ~ 1 => near-circular (ill-conditioned)

# Plot downsampling target
PLOT_DS_TARGET    = 120000

np.set_printoptions(suppress=True)

# ============================================================
# Candidate-A flux kernel
# ============================================================

def build_flux_kernel(r_max, chi_flux):
    k = np.arange(1, r_max + 1, dtype=np.float64)
    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k**2 + 2.0

    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0

    Phi = np.cumsum(counts * rho)

    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])

    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
        w_arr[d] = s

    return w_arr, gamma_arr

@njit(fastmath=True)
```

 doi:10.20944/preprints202405.2028.v8


```
def get_field(d, w_arr, gamma_arr):
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]
    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0
    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1
    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]
    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w
    return w_val, g_val

# ============================================================
# DO-native two-body state (Chebyshev radius + Manhattan direction)
# ============================================================

@njit(fastmath=True)
def compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr):
    # r_vec = Earth - Moon
    rx = pos[0,0] - pos[1,0]
    ry = pos[0,1] - pos[1,1]
    rz = pos[0,2] - pos[1,2]

    ax = abs(rx); ay = abs(ry); az = abs(rz)

    # Chebyshev shell index
    d_inf = ax
    if ay > d_inf:
        d_inf = ay
    if az > d_inf:
        d_inf = az

    w_val, g_val = get_field(d_inf, w_arr, gamma_arr)

    # Manhattan distance
    r1 = ax + ay + az

    # Manhattan-fraction direction (signed)
    ux = 0.0; uy = 0.0; uz = 0.0
    if r1 > 0.0:
        ux = rx / r1
        uy = ry / r1
        uz = rz / r1

    # Force magnitude (g_val < 0 gives attraction)
    f_mag = chi_e * m1 * m2 * g_val

    # Accelerations
    inv_m1 = 1.0 / m1
    inv_m2 = 1.0 / m2
```

 doi:10.20944/preprints202405.2028.v8


```
    acc = np.zeros((2,3), dtype=np.float64)
    acc[0,0] =  f_mag * inv_m1 * ux
    acc[0,1] =  f_mag * inv_m1 * uy
    acc[0,2] =  f_mag * inv_m1 * uz

    acc[1,0] = -f_mag * inv_m2 * ux
    acc[1,1] = -f_mag * inv_m2 * uy
    acc[1,2] = -f_mag * inv_m2 * uz

    # Pair potential energy
    U = -chi_e * m1 * m2 * w_val

    return acc, U, f_mag, ux, uy, uz

# ============================================================
# Forest-Ruth + I_DO integrator (tilted) with sampling
# Stores: t, E_naive, I_DO, L-vector, r, positions
# ============================================================

@njit(fastmath=True)
def run_multi_orbit_tilted_IDO(
    dt, steps, v_factor, d0, incl_rad,
    m1, m2, chi_e, w_arr, gamma_arr,
    samples_max
):
    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    # Initial circular speed and v_init
    _, g0 = get_field(d0, w_arr, gamma_arr)
    f_mag_init = chi_e * m1 * m2 * abs(g0)
    mu = (m1 * m2) / (m1 + m2)
    v_circ = math.sqrt(f_mag_init * d0 / mu)
    v_init = v_factor * v_circ

    # Initial positions: Moon at +D0 on x; shift COM to origin
    pos = np.zeros((2,3), dtype=np.float64)
    pos[1,0] = d0
    Mtot = m1 + m2
    comx = (pos[0,0]*m1 + pos[1,0]*m2) / Mtot
    pos[0,0] -= comx
    pos[1,0] -= comx

    # Tilted tangential velocity in y-z plane
    vy = v_init * math.cos(incl_rad)
    vz = v_init * math.sin(incl_rad)

    vel = np.zeros((2,3), dtype=np.float64)
    vel[1,1] = vy
    vel[1,2] = vz
```

 doi:10.20944/preprints202405.2028.v8


```
mass_ratio = m2 / m1
vel[0,1] = -mass_ratio * vy
vel[0,2] = -mass_ratio * vz

# Sampling stride
sample_stride = steps // samples_max
if sample_stride < 1:
    sample_stride = 1

n_alloc = steps // sample_stride + 2

t_hist  = np.zeros(n_alloc, dtype=np.float64)
E_hist  = np.zeros(n_alloc, dtype=np.float64)   # naive
I_hist  = np.zeros(n_alloc, dtype=np.float64)   # I_DO
Lx_hist = np.zeros(n_alloc, dtype=np.float64)
Ly_hist = np.zeros(n_alloc, dtype=np.float64)
Lz_hist = np.zeros(n_alloc, dtype=np.float64)
r_hist  = np.zeros(n_alloc, dtype=np.float64)
pos1_hist = np.zeros((n_alloc,3), dtype=np.float64)
pos2_hist = np.zeros((n_alloc,3), dtype=np.float64)

W_closure = 0.0

# Initial sample
acc_s, U_s, fmag_s, ux_s, uy_s, uz_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)

ke = 0.5 * (
    m1 * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
    m2 * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2)
)
E_naive = ke + U_s
I_DO = E_naive - W_closure

# COM and L about COM
COMx = (pos[0,0]*m1 + pos[1,0]*m2) / Mtot
COMy = (pos[0,1]*m1 + pos[1,1]*m2) / Mtot
COMz = (pos[0,2]*m1 + pos[1,2]*m2) / Mtot

r0x = pos[0,0] - COMx; r0y = pos[0,1] - COMy; r0z = pos[0,2] - COMz
r1x = pos[1,0] - COMx; r1y = pos[1,1] - COMy; r1z = pos[1,2] - COMz

p0x = m1*vel[0,0]; p0y = m1*vel[0,1]; p0z = m1*vel[0,2]
p1x = m2*vel[1,0]; p1y = m2*vel[1,1]; p1z = m2*vel[1,2]

Lx = (r0y*p0z - r0z*p0y) + (r1y*p1z - r1z*p1y)
Ly = (r0z*p0x - r0x*p0z) + (r1z*p1x - r1x*p1z)
Lz = (r0x*p0y - r0y*p0x) + (r1x*p1y - r1y*p1x)

rvx = pos[1,0] - pos[0,0]
rvy = pos[1,1] - pos[0,1]
rvz = pos[1,2] - pos[0,2]
r = math.sqrt(rvx*rvx + rvy*rvy + rvz*rvz)

idx = 0
t_hist[idx] = 0.0
```

 doi:10.20944/preprints202405.2028.v8


```
E_hist[idx] = E_naive
I_hist[idx] = I_DO
Lx_hist[idx] = Lx
Ly_hist[idx] = Ly
Lz_hist[idx] = Lz
r_hist[idx] = r
pos1_hist[idx,:] = pos[0,:]
pos2_hist[idx,:] = pos[1,:]
idx += 1

for step in range(1, steps + 1):
    for j in range(4):
        # start-of-drift
        acc_s, U_s, fmag_s, ux_s, uy_s, uz_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr,
        ↪  gamma_arr)
        F_sx = fmag_s * ux_s
        F_sy = fmag_s * uy_s
        F_sz = fmag_s * uz_s

        # drift
        dr0x = c[j]*dt*vel[0,0]; dr0y = c[j]*dt*vel[0,1]; dr0z = c[j]*dt*vel[0,2]
        dr1x = c[j]*dt*vel[1,0]; dr1y = c[j]*dt*vel[1,1]; dr1z = c[j]*dt*vel[1,2]
        pos[0,0] += dr0x; pos[0,1] += dr0y; pos[0,2] += dr0z
        pos[1,0] += dr1x; pos[1,1] += dr1y; pos[1,2] += dr1z

        # end-of-drift
        acc_e, U_e, fmag_e, ux_e, uy_e, uz_e = compute_state_DOnative(pos, m1, m2, chi_e, w_arr,
        ↪  gamma_arr)
        F_ex = fmag_e * ux_e
        F_ey = fmag_e * uy_e
        F_ez = fmag_e * uz_e

        # average force (on Earth) over drift
        F_avgx = 0.5*(F_sx + F_ex)
        F_avgy = 0.5*(F_sy + F_ey)
        F_avgz = 0.5*(F_sz + F_ez)

        # relative displacement (Earth - Moon) over drift
        dr_relx = dr0x - dr1x
        dr_rely = dr0y - dr1y
        dr_relz = dr0z - dr1z

        W_mech = F_avgx*dr_relx + F_avgy*dr_rely + F_avgz*dr_relz
        W_closure += (W_mech + (U_e - U_s))

        # kick
        if dcoef[j] != 0.0:
            vel[0,0] += dcoef[j]*dt*acc_e[0,0]; vel[0,1] += dcoef[j]*dt*acc_e[0,1]; vel[0,2] +=
            ↪  dcoef[j]*dt*acc_e[0,2]
            vel[1,0] += dcoef[j]*dt*acc_e[1,0]; vel[1,1] += dcoef[j]*dt*acc_e[1,1]; vel[1,2] +=
            ↪  dcoef[j]*dt*acc_e[1,2]

    if (step % sample_stride == 0) and (idx < n_alloc):
        acc_c, U_c, fmag_c, ux_c, uy_c, uz_c = compute_state_DOnative(pos, m1, m2, chi_e, w_arr,
        ↪  gamma_arr)
```

 doi:10.20944/preprints202405.2028.v8


```
            ke = 0.5 * (
                m1 * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
                m2 * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2)
            )
            E_naive = ke + U_c
            I_DO = E_naive - W_closure

            COMx = (pos[0,0]*m1 + pos[1,0]*m2) / Mtot
            COMy = (pos[0,1]*m1 + pos[1,1]*m2) / Mtot
            COMz = (pos[0,2]*m1 + pos[1,2]*m2) / Mtot

            r0x = pos[0,0] - COMx; r0y = pos[0,1] - COMy; r0z = pos[0,2] - COMz
            r1x = pos[1,0] - COMx; r1y = pos[1,1] - COMy; r1z = pos[1,2] - COMz

            p0x = m1*vel[0,0]; p0y = m1*vel[0,1]; p0z = m1*vel[0,2]
            p1x = m2*vel[1,0]; p1y = m2*vel[1,1]; p1z = m2*vel[1,2]

            Lx = (r0y*p0z - r0z*p0y) + (r1y*p1z - r1z*p1y)
            Ly = (r0z*p0x - r0x*p0z) + (r1z*p1x - r1x*p1z)
            Lz = (r0x*p0y - r0y*p0x) + (r1x*p1y - r1y*p1x)

            rvx = pos[1,0] - pos[0,0]
            rvy = pos[1,1] - pos[0,1]
            rvz = pos[1,2] - pos[0,2]
            r = math.sqrt(rvx*rvx + rvy*rvy + rvz*rvz)

            t_hist[idx] = step*dt
            E_hist[idx] = E_naive
            I_hist[idx] = I_DO
            Lx_hist[idx] = Lx
            Ly_hist[idx] = Ly
            Lz_hist[idx] = Lz
            r_hist[idx] = r
            pos1_hist[idx,:] = pos[0,:]
            pos2_hist[idx,:] = pos[1,:]
            idx += 1

    return idx, sample_stride, t_hist, E_hist, I_hist, Lx_hist, Ly_hist, Lz_hist, r_hist, pos1_hist,
    ↪  pos2_hist

# ============================================================
# Diagnostics helpers
# ============================================================

def orbital_plane_rotate(rel_xyz, incl_rad):
    c = math.cos(incl_rad)
    s = math.sin(incl_rad)
    x = rel_xyz[:,0]
    y = rel_xyz[:,1]
    z = rel_xyz[:,2]
    xp = x
    yp = c*y + s*z
    zp = -s*y + c*z
    return xp, yp, zp
```

 doi:10.20944/preprints202405.2028.v8


```
def find_boundaries_signed(Rx, Ry, t, k_max):
    phi = np.unwrap(np.arctan2(Ry, Rx))
    dphi = float(phi[-1] - phi[0])
    sgn = 1.0 if dphi >= 0.0 else -1.0
    phi_s = phi * sgn
    phi0 = float(phi_s[0])
    two_pi = 2.0 * math.pi

    boundaries = [(0, 0, 0.0, float(t[0]))]  # (k, i_lo, frac, t_boundary)
    target_k = 1
    i = 1
    while target_k <= k_max and i < len(phi_s):
        thr = phi0 + target_k * two_pi
        if phi_s[i] >= thr:
            i_lo = i - 1
            i_hi = i
            denom = (phi_s[i_hi] - phi_s[i_lo])
            frac = 0.0 if denom == 0.0 else (thr - phi_s[i_lo]) / denom
            frac = min(1.0, max(0.0, frac))
            t_b = float(t[i_lo] + frac * (t[i_hi] - t[i_lo]))
            boundaries.append((target_k, i_lo, frac, t_b))
            target_k += 1
        i += 1
    return boundaries, sgn

def quad_vertex_offset(y_m1, y0, y_p1):
    denom = (y_m1 - 2.0*y0 + y_p1)
    if denom == 0.0:
        return 0.0
    d = 0.5 * (y_m1 - y_p1) / denom
    return max(-1.0, min(1.0, d))

def quad_interp_at(d, f_m1, f0, f_p1):
    a = 0.5*(f_p1 + f_m1 - 2.0*f0)
    b = 0.5*(f_p1 - f_m1)
    return a*d*d + b*d + f0

def wrap_pm_pi(theta):
    return (theta + math.pi) % (2.0*math.pi) - math.pi

def unwrap_pi(prev, raw):
    if prev is None:
        return raw
    best = raw
    best_d = abs(raw - prev)
    for k in range(-6, 7):
        cand = raw + k*math.pi
        d = abs(cand - prev)
        if d < best_d:
            best = cand
            best_d = d
    return best

def harmonic_fit_omega_e(Rx_seg, Ry_seg, r_seg):
```

 doi:10.20944/preprints202405.2028.v8


```
    # Fit 1/r = A + B cos\Phi + C sin\Phi
    rho = np.sqrt(Rx_seg*Rx_seg + Ry_seg*Ry_seg)
    rho = np.maximum(rho, 1e-18)
    cphi = Rx_seg / rho
    sphi = Ry_seg / rho
    y = 1.0 / np.maximum(r_seg, 1e-18)

    S00 = float(len(y))
    S01 = float(np.sum(cphi))
    S02 = float(np.sum(sphi))
    S11 = float(np.sum(cphi*cphi))
    S12 = float(np.sum(cphi*sphi))
    S22 = float(np.sum(sphi*sphi))

    b0 = float(np.sum(y))
    b1 = float(np.sum(y*cphi))
    b2 = float(np.sum(y*sphi))

    M = np.array([[S00, S01, S02],
                  [S01, S11, S12],
                  [S02, S12, S22]], dtype=np.float64)
    b = np.array([b0, b1, b2], dtype=np.float64)

    try:
        A, B, C = np.linalg.solve(M, b)
    except np.linalg.LinAlgError:
        return 0.0, 0.0, 0.0  # omega, e, okflag

    if A <= 1e-18:
        return 0.0, 0.0, 0.0

    e_over_p = math.sqrt(B*B + C*C)
    e = e_over_p / A
    omega = math.atan2(C, B)
    return omega, e, 1.0

def pca_axis_angle_and_cond(Rx_seg, Ry_seg):
    # PCA on (Rx,Ry) gives principal axis angle and cond_ratio
    x = Rx_seg - float(np.mean(Rx_seg))
    y = Ry_seg - float(np.mean(Ry_seg))

    Sxx = float(np.mean(x*x))
    Syy = float(np.mean(y*y))
    Sxy = float(np.mean(x*y))

    tr = Sxx + Syy
    det = Sxx*Syy - Sxy*Sxy
    disc = math.sqrt(max(0.0, tr*tr - 4.0*det))
    lam1 = 0.5*(tr + disc)
    lam2 = 0.5*(tr - disc)

    if lam1 <= 1e-30:
        return False, 0.0, 1.0, 0.0  # ok, theta, cond_ratio, e_cov

    # principal axis angle (major)
```

 doi:10.20944/preprints202405.2028.v8


```
    theta = 0.5 * math.atan2(2.0*Sxy, (Sxx - Syy))

    cond_ratio = math.sqrt(max(0.0, lam2/lam1))  # near 1 => near circle
    e_cov = math.sqrt(max(0.0, 1.0 - lam2/lam1))
    return True, theta, cond_ratio, e_cov

def robust_stats(x):
    x = np.asarray(x, dtype=np.float64)
    if x.size == 0:
        return (np.nan, np.nan, np.nan, np.nan)
    mu = float(np.mean(x))
    sig = float(np.std(x))
    med = float(np.median(x))
    mad = float(np.median(np.abs(x - med)))
    mad_sigma = 1.4826 * mad
    return (mu, sig, med, mad_sigma)

# ============================================================
# MAIN
# ============================================================

if __name__ == "__main__":
    print("Building Candidate-A flux kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)

    print("Warming up Numba JIT...")
    _ = run_multi_orbit_tilted_IDO(
        DT, 100_000, V_FACTOR, D0, INCL_RAD,
        M1, M2, CHI_E, w_arr, gamma_arr, 50_000
    )
    print("Warm-up done.\n")

    print("Running Earth-Moon tilted horizon:")
    print(f"  N_ORBITS_TARGET    = {N_ORBITS_TARGET}")
    print(f"  EXTRA_ORBITS_RUN   = {EXTRA_ORBITS_RUN}")
    print(f"  dt                 = {DT}")
    print(f"  D0                 = {D0}")
    print(f"  V_FACTOR           = {V_FACTOR}")
    print(f"  tilt               = {INCL_DEG} deg")
    print(f"  STEPS_PER_ORBIT    = {STEPS_PER_ORBIT}")
    print(f"  STEPS_TOTAL        = {STEPS_TOTAL}")
    print(f"  SAMPLES_PER_ORBIT  = {SAMPLES_PER_ORBIT}")
    print(f"  SAMPLES_MAX        = {SAMPLES_MAX}\n")

    t0 = time.time()
    (n_samples, sample_stride,
     t_hist, E_hist, I_hist,
     Lx_hist, Ly_hist, Lz_hist,
     r_hist, pos1_hist, pos2_hist) = run_multi_orbit_tilted_IDO(
        DT, STEPS_TOTAL, V_FACTOR, D0, INCL_RAD,
        M1, M2, CHI_E, w_arr, gamma_arr, SAMPLES_MAX
    )
    t1 = time.time()
    print(f"Simulation complete. Elapsed time: {t1 - t0:.2f} s\n")
```

 doi:10.20944/preprints202405.2028.v8


```
# Trim
t_hist = t_hist[:n_samples]
E_hist = E_hist[:n_samples]
I_hist = I_hist[:n_samples]
Lx_hist = Lx_hist[:n_samples]
Ly_hist = Ly_hist[:n_samples]
Lz_hist = Lz_hist[:n_samples]
r_hist = r_hist[:n_samples]
pos1_hist = pos1_hist[:n_samples,:]
pos2_hist = pos2_hist[:n_samples,:]

# Relative vectors and orbital-plane rotation
rel = pos2_hist - pos1_hist
Rx, Ry, Rz = orbital_plane_rotate(rel, INCL_RAD)

# Find orbit boundaries
boundaries, sgn = find_boundaries_signed(Rx, Ry, t_hist, N_ORBITS_TARGET + 1)
n_orb = min(len(boundaries) - 1, N_ORBITS_TARGET)
print(f"Found {n_orb} complete orbits for analysis.\n")

# Analysis window end
_, i_lo_end, _, _ = boundaries[n_orb]
end_idx = min(i_lo_end + 2, n_samples)

# GLOBAL INVARIANTS
I0 = float(I_hist[0])
E0 = float(E_hist[0])

dI = (I_hist[:end_idx] - I0) / abs(I0)
dE = (E_hist[:end_idx] - E0) / abs(E0)

Lmag = np.sqrt(Lx_hist[:end_idx]**2 + Ly_hist[:end_idx]**2 + Lz_hist[:end_idx]**2)
L0 = float(Lmag[0])
dL = (Lmag - L0) / abs(L0)

Mtot = M1 + M2
COM = (pos1_hist[:end_idx]*M1 + pos2_hist[:end_idx]*M2) / Mtot
COM_vec = COM - COM[0]
COM_mag = np.linalg.norm(COM_vec, axis=1)
COM_max = float(COM_mag.max())
COM_rms = float(np.sqrt(np.mean(COM_mag**2)))

inc_raw = np.degrees(np.arccos(np.clip(Lz_hist[:end_idx] / np.maximum(Lmag, 1e-30), -1.0, 1.0)))
inc_eff = np.minimum(inc_raw, 180.0 - inc_raw)
node = np.degrees(np.arctan2(Lx_hist[:end_idx], -Ly_hist[:end_idx]))

print("GLOBAL INVARIANTS (analysis horizon)")
print("------------------------------------")
print(f"I_DO net drift        = {float(dI[-1]):.3e}")
print(f"I_DO RMS              = {float(np.std(dI)):.3e}")
print(f"Naive E net drift      = {float(dE[-1]):.3e}")
print(f"Naive E RMS            = {float(np.std(dE)):.3e}")
print(f"|L| net drift          = {float(dL[-1]):.3e}")
print(f"|L| RMS                = {float(np.std(dL)):.3e}")
print(f"COM drift max (hops)   = {COM_max:.3e}")
```

 doi:10.20944/preprints202405.2028.v8


```
print(f"COM drift RMS (hops)   = {COM_rms:.3e}")
print(f"Inclination mean/std   = {float(inc_eff.mean()):.6f} / {float(inc_eff.std()):.6f}")
print(f"Node start/end         = {float(node[0]):.6f} -> {float(node[-1]):.6f}\n")

# ORBIT STEP COUNT (from boundary times)
step_counts = np.zeros(n_orb, dtype=np.int64)
for j in range(1, n_orb + 1):
    _, _, _, t0b = boundaries[j-1]
    _, _, _, t1b = boundaries[j]
    step_counts[j-1] = int(round((t1b - t0b) / DT))

print("ORBIT STEP COUNT (from boundary times)")
print("--------------------------------------")
print(f"mean={step_counts.mean():.3f}  std={step_counts.std():.3f}  min={step_counts.min()}
↪  max={step_counts.max()}\n")

# Selected-orbit print set
sel = set(range(1, PRINT_FIRST_N + 1))
for k in range(PRINT_EVERY_N, n_orb + 1, PRINT_EVERY_N):
    sel.add(k)
for k in range(max(1, n_orb - PRINT_LAST_N + 1), n_orb + 1):
    sel.add(k)

# PER-ORBIT diagnostics
e_rmin = np.zeros(n_orb, dtype=np.float64)
rmin = np.zeros(n_orb, dtype=np.float64)
rmax = np.zeros(n_orb, dtype=np.float64)
a_r  = np.zeros(n_orb, dtype=np.float64)
Rz_rms = np.zeros(n_orb, dtype=np.float64)

omega_peri_u = np.zeros(n_orb, dtype=np.float64)
omega_harm_u = np.zeros(n_orb, dtype=np.float64)
omega_axis_u = np.zeros(n_orb, dtype=np.float64)

domega_peri = np.zeros(n_orb, dtype=np.float64)
domega_harm = np.zeros(n_orb, dtype=np.float64)
domega_axis = np.zeros(n_orb, dtype=np.float64)

e_harm = np.zeros(n_orb, dtype=np.float64)
cond_ratio = np.zeros(n_orb, dtype=np.float64)
e_cov = np.zeros(n_orb, dtype=np.float64)

prev_peri = None
prev_harm = None
prev_axis = None

print("PER-ORBIT TABLE (selected)")
print("--------------------------")
hdr = (f"{'orb':>4} {'Nsteps':>9} {'e_rmin':>8} {'rmin':>9} {'rmax':>9} {'a_r':>9} "
       f"{'\omega_peri(deg)':>12} {'\Delta\omega_p':>10} "
       f"{'\omega_harm(deg)':>12} {'\Delta\omega_h':>10} {'e_h':>7} "
       f"{'\omega_axis(deg)':>12} {'\Delta\omega_a':>10} {'cond':>7} {'Rz_RMS':>10}")
print(hdr)

for j in range(1, n_orb + 1):
```

 doi:10.20944/preprints202405.2028.v8


```
_, i0, _, t0b = boundaries[j-1]
_, i1, _, t1b = boundaries[j]
seg_start = i0
seg_end = min(i1 + 2, end_idx)

Rx_seg = Rx[seg_start:seg_end]
Ry_seg = Ry[seg_start:seg_end]
Rz_seg = Rz[seg_start:seg_end]
r_seg  = r_hist[seg_start:seg_end]

if len(r_seg) < 7:
    continue

# extrema (quadratic)
i_min = int(np.argmin(r_seg))
i_max = int(np.argmax(r_seg))

# refined periapsis location in plane
if 0 < i_min < (len(r_seg) - 1):
    dmin = quad_vertex_offset(r_seg[i_min-1], r_seg[i_min], r_seg[i_min+1])
    rmin_j = quad_interp_at(dmin, r_seg[i_min-1], r_seg[i_min], r_seg[i_min+1])
    Rxmin = quad_interp_at(dmin, Rx_seg[i_min-1], Rx_seg[i_min], Rx_seg[i_min+1])
    Rymin = quad_interp_at(dmin, Ry_seg[i_min-1], Ry_seg[i_min], Ry_seg[i_min+1])
else:
    rmin_j = float(r_seg[i_min]); Rxmin = float(Rx_seg[i_min]); Rymin = float(Ry_seg[i_min])

if 0 < i_max < (len(r_seg) - 1):
    dmax = quad_vertex_offset(r_seg[i_max-1], r_seg[i_max], r_seg[i_max+1])
    rmax_j = quad_interp_at(dmax, r_seg[i_max-1], r_seg[i_max], r_seg[i_max+1])
else:
    rmax_j = float(r_seg[i_max])

e = (rmax_j - rmin_j) / (rmax_j + rmin_j) if (rmax_j + rmin_j) != 0.0 else 0.0
ar = 0.5*(rmax_j + rmin_j)

# periapsis angle
omega_p_raw = math.atan2(Rymin, Rxmin)
omega_p_u = omega_p_raw if prev_peri is None else (prev_peri + wrap_pm_pi(omega_p_raw - prev_peri))
dwp = 0.0 if prev_peri is None else (omega_p_u - prev_peri) * 180.0/math.pi * 3600.0
prev_peri = omega_p_u

# harmonic fit omega + e_harm
omega_h_raw, eh, ok_h = harmonic_fit_omega_e(Rx_seg, Ry_seg, r_seg)
omega_h_u = omega_h_raw if prev_harm is None else (prev_harm + wrap_pm_pi(omega_h_raw - prev_harm))
dwh = 0.0 if prev_harm is None else (omega_h_u - prev_harm) * 180.0/math.pi * 3600.0
prev_harm = omega_h_u

# PCA axis angle + conditioning
ok_pca, theta_axis, cond, ec = pca_axis_angle_and_cond(Rx_seg, Ry_seg)
# orient axis toward periapsis direction (\pi ambiguity)
theta0 = theta_axis
d0 = abs(wrap_pm_pi(theta0 - omega_p_raw))
d1 = abs(wrap_pm_pi((theta0 + math.pi) - omega_p_raw))
omega_a_raw = theta0 if d0 <= d1 else (theta0 + math.pi)
omega_a_u = unwrap_pi(prev_axis, omega_a_raw)
```

 doi:10.20944/preprints202405.2028.v8


```
    dwa = 0.0 if prev_axis is None else (omega_a_u - prev_axis) * 180.0/math.pi * 3600.0
    prev_axis = omega_a_u

    RzRMS = float(np.sqrt(np.mean(Rz_seg*Rz_seg)))

    # store
    e_rmin[j-1] = e
    rmin[j-1] = rmin_j
    rmax[j-1] = rmax_j
    a_r[j-1] = ar
    Rz_rms[j-1] = RzRMS

    omega_peri_u[j-1] = omega_p_u
    omega_harm_u[j-1] = omega_h_u
    omega_axis_u[j-1] = omega_a_u

    domega_peri[j-1] = dwp
    domega_harm[j-1] = dwh
    domega_axis[j-1] = dwa

    e_harm[j-1] = eh
    cond_ratio[j-1] = cond
    e_cov[j-1] = ec

    if j in sel:
        # print angles modulo 360 for readability, deltas from unwrapped
        op = (math.degrees(omega_p_u) % 360.0 + 360.0) % 360.0
        oh = (math.degrees(omega_h_u) % 360.0 + 360.0) % 360.0
        oa = (math.degrees(omega_a_u) % 360.0 + 360.0) % 360.0
        print(f"{j:4d} {int(step_counts[j-1]):9d} {e:8.5f} {rmin_j:9.4f} {rmax_j:9.4f} {ar:9.4f} "
              f"{op:12.4f} {dwp:10.3f} "
              f"{oh:12.4f} {dwh:10.3f} {eh:7.4f} "
              f"{oa:12.4f} {dwa:10.3f} {cond:7.5f} {RzRMS:10.3e}")

# ============================================================
# SUMMARY + BLOCKS
# ============================================================

print("\nPER-ORBIT SUMMARY (analysis horizon)")
print("------------------------------------")
print(f"e_rmin: mean={float(e_rmin.mean()):.6f}  std={float(e_rmin.std()):.6f}
↪  first={float(e_rmin[0]):.6f}  last={float(e_rmin[-1]):.6f}")

# Exclude first orbit deltas
dp = domega_peri[1:]
dh = domega_harm[1:]
da = domega_axis[1:]

mu_p, sig_p, med_p, mad_p = robust_stats(dp)
mu_h, sig_h, med_h, mad_h = robust_stats(dh)
mu_a, sig_a, med_a, mad_a = robust_stats(da)

print("\nAPSIS STATISTICS (exclude first orbit)")
print("--------------------------------------")
```

 doi:10.20944/preprints202405.2028.v8


```
print(f"\Delta\omega_peri: mean={mu_p:.6f}  \sigma={sig_p:.6f}   median={med_p:.6f}
↪  MAD\sigma={mad_p:.6f}")
print(f"\Delta\omega_harm: mean={mu_h:.6f}  \sigma={sig_h:.6f}   median={med_h:.6f}
↪  MAD\sigma={mad_h:.6f}")
print(f"\Delta\omega_axis: mean={mu_a:.6f}  \sigma={sig_a:.6f}   median={med_a:.6f}
↪  MAD\sigma={mad_a:.6f}")

# Well-conditioned mask (diagnostics only)
well = (e_harm >= E_HARM_MIN_FOR_OMEGA) & (cond_ratio <= COND_RATIO_MAX_FOR_OMEGA)
well[0] = False  # first orbit excluded
da_w = domega_axis[well]
dh_w = domega_harm[well]
dp_w = domega_peri[well]

mu_pw, sig_pw, med_pw, mad_pw = robust_stats(dp_w)
mu_hw, sig_hw, med_hw, mad_hw = robust_stats(dh_w)
mu_aw, sig_aw, med_aw, mad_aw = robust_stats(da_w)

print("\nAPSIS STATISTICS (well-conditioned subset)")
print("-----------------------------------------")
print(f"N_well = {int(well.sum())} / {n_orb-1}")
print(f"\Delta\omega_peri: mean={mu_pw:.6f}  \sigma={sig_pw:.6f}   median={med_pw:.6f}
↪  MAD\sigma={mad_pw:.6f}")
print(f"\Delta\omega_harm: mean={mu_hw:.6f}  \sigma={sig_hw:.6f}   median={med_hw:.6f}
↪  MAD\sigma={mad_hw:.6f}")
print(f"\Delta\omega_axis: mean={mu_aw:.6f}  \sigma={sig_aw:.6f}   median={med_aw:.6f}
↪  MAD\sigma={mad_aw:.6f}")

# Block summary (exclude first orbit in each block for \Delta\omega)
print("\nBLOCK SUMMARY (block size = 100 orbits)")
print("---------------------------------------")
print(f"{'block':>5} {'orbits':>11} {'<e_rmin>':>10} {'\sigma(e_rmin)':>10} {'<\Delta\omega_axis>':>10}
↪  {'\sigma_axis':>10} {'N_well':>8}")
nb = n_orb // BLOCK_SIZE
for b in range(nb):
    a = b*BLOCK_SIZE
    z = (b+1)*BLOCK_SIZE
    e_blk = e_rmin[a:z]
    # deltas: exclude first orbit in the block
    da_blk = domega_axis[a+1:z]
    well_blk = well[a+1:z]
    da_wblk = domega_axis[a+1:z][well_blk]

    mu_da, sig_da, _, _ = robust_stats(da_blk)
    mu_w, sig_w, _, _ = robust_stats(da_wblk)

    print(f"{b+1:5d} {a+1:4d}–{z:4d} {float(e_blk.mean()):10.6f} {float(e_blk.std()):10.6f} "
          f"{mu_da:10.3f} {sig_da:10.3f} {int(well_blk.sum()):8d}")

# ============================================================
# PLOTS
# ============================================================

# Downsample for plotting
ds = max(1, end_idx // PLOT_DS_TARGET)
```

 doi:10.20944/preprints202405.2028.v8


```
idxs = np.arange(0, end_idx, ds, dtype=int)

COM_ds = COM[::ds]
pos1_ds = pos1_hist[:end_idx:ds]
pos2_ds = pos2_hist[:end_idx:ds]

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")
ax.plot(pos1_ds[:,0], pos1_ds[:,1], pos1_ds[:,2], label="Earth")
ax.plot(pos2_ds[:,0], pos2_ds[:,1], pos2_ds[:,2], label="Moon")
ax.plot(COM_ds[:,0], COM_ds[:,1], COM_ds[:,2], label="COM")
ax.set_title("3D trajectories (tilted, downsampled)")
ax.set_xlabel("x (hops)"); ax.set_ylabel("y (hops)"); ax.set_zlabel("z (hops)")
ax.legend()
plt.tight_layout()
plt.show()

plt.figure(figsize=(6, 6))
plt.plot(Rx[:end_idx:ds], Ry[:end_idx:ds], lw=0.6)
plt.scatter([0.0], [0.0], s=20)
plt.xlabel("x' (hops)")
plt.ylabel("y' (hops)")
plt.title("Relative orbit in orbital plane (downsampled)")
plt.axis("equal")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], r_hist[:end_idx], lw=0.45)
plt.xlabel("t (code units)")
plt.ylabel("r (hops)")
plt.title("r(t) over analysis horizon")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], dI, lw=0.6, label="\DeltaI_DO/I0")
plt.plot(t_hist[:end_idx], dE, lw=0.6, label="\DeltaE_naive/E0", alpha=0.7)
plt.xlabel("t (code units)")
plt.ylabel("\Delta/initial")
plt.title("Energy diagnostics")
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], dL, lw=0.6)
plt.xlabel("t (code units)")
plt.ylabel("\Delta|L|/|L0|")
plt.title("|L| drift")
plt.grid(True, alpha=0.3)
plt.tight_layout()
```

 doi:10.20944/preprints202405.2028.v8


```
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], COM_mag, lw=0.6)
plt.xlabel("t (code units)")
plt.ylabel("|COM(t)-COM(0)|")
plt.title("COM drift magnitude")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(np.arange(1, n_orb+1), e_rmin, lw=0.8)
plt.xlabel("Orbit index")
plt.ylabel("e_rmin (interp)")
plt.title("e_rmin per orbit")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(np.arange(1, n_orb+1), domega_axis, lw=0.8)
plt.xlabel("Orbit index")
plt.ylabel("\Delta\omega_axis (arcsec/orbit)")
plt.title("\Delta\omega_axis per orbit")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], inc_eff, lw=0.6)
plt.xlabel("t (code units)")
plt.ylabel("inclination (deg)")
plt.title("Inclination vs time (from L-vector)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(t_hist[:end_idx], node, lw=0.6)
plt.xlabel("t (code units)")
plt.ylabel("node angle \omega (deg)")
plt.title("Node angle \omega vs time (from L-vector)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

Appendix B.3.3.5. Appendix B.9.2. Console Transcript

```
Building Candidate-A flux kernel...
Warming up Numba JIT...
Warm-up done.

Running Earth–Moon tilted horizon:
  N_ORBITS_TARGET    = 1000
  EXTRA_ORBITS_RUN   = 5
```

 doi:10.20944/preprints202405.2028.v8


```
  dt                = 0.003
  D0                = 30.0
  V_FACTOR          = 0.939199
  tilt              = 5.145 deg
  STEPS_PER_ORBIT    = 1499675
  STEPS_TOTAL        = 1507173375
  SAMPLES_PER_ORBIT  = 8000
  SAMPLES_MAX        = 8040000

Simulation complete. Elapsed time: 867.39 s

Found 1000 complete orbits for analysis.

GLOBAL INVARIANTS (analysis horizon)
------------------------------------
I_DO net drift        = 2.245e-12
I_DO RMS              = 2.242e-12
Naive E net drift      = -6.041e-02
Naive E RMS            = 1.924e-01
|L| net drift          = -6.467e-12
|L| RMS                = 3.589e-12
COM drift max (hops)   = 3.296e-09
COM drift RMS (hops)   = 2.065e-09
Inclination mean/std   = 5.145000 / 0.000000
Node start/end         = 0.000000 -> -0.000000

ORBIT STEP COUNT (from boundary times)
--------------------------------------
mean=1497696.780  std=946.737  min=1496221  max=1499847

PER-ORBIT TABLE (selected)
--------------------------
 orb    Nsteps   e_rmin      rmin      rmax       a_r  \omega_peri(deg)       \Delta\omega_p
  ↪  \omega_harm(deg)       \Delta\omega_h     e_h  \omega_axis(deg)       \Delta\omega_a    cond     Rz_RMS
   1   1499563  0.05490   26.8796   30.0024   28.4410     180.5839      0.000     180.1532      0.000  0.0553
    ↪      180.0134      0.000 0.98443  4.946e-13
   2   1499562  0.05498   26.8797   30.0075   28.4436     181.2337   2339.131     180.4596   1102.886  0.0552
    ↪      180.0395     94.065 0.98443  6.725e-13
   3   1499559  0.05506   26.8799   30.0124   28.4462     181.7031   1689.732     180.7657   1101.914  0.0552
    ↪      180.0656     93.876 0.98443  7.330e-13
   4   1499555  0.05513   26.8803   30.0171   28.4487     182.0928   1402.966     181.0713   1100.294  0.0552
    ↪      180.0916     93.559 0.98443  6.830e-13
   5   1499550  0.05520   26.8808   30.0216   28.4512     182.4344   1229.844     181.3763   1098.025  0.0552
    ↪      180.1175     93.117 0.98443  6.318e-13
   6   1499544  0.05525   26.8815   30.0259   28.4537     182.7430   1110.842     181.6805   1095.108  0.0552
    ↪      180.1432     92.554 0.98442  7.110e-13
   7   1499536  0.05531   26.8823   30.0299   28.4561     183.0267   1021.306     181.9837   1091.526  0.0551
    ↪      180.1709     99.830 0.98453  3.128e-13
   8   1499527  0.05535   26.8834   30.0338   28.4586     183.2908    951.051     182.2857   1087.340  0.0551
    ↪      180.1933     80.588 0.98442  1.683e-13
   9   1499517  0.05539   26.8845   30.0374   28.4610     183.5391    893.625     182.5864   1082.459  0.0550
    ↪      180.2183     90.108 0.98441  3.218e-13
  10   1499505  0.05542   26.8859   30.0408   28.4633     183.7738    845.062     182.8856   1076.941  0.0550
    ↪      180.2431     89.096 0.98441  2.803e-13
```

 doi:10.20944/preprints202405.2028.v8


```
 20   1499321  0.05534   26.9090   30.0618   28.4854     185.6401    564.409     185.7517    983.797  0.0540
 ↪       180.4775     96.197 0.98454  1.947e-12
 40   1498718  0.05360   26.9860   30.0426   28.5143     187.7614    239.357     190.0691    560.204  0.0511
 ↪       180.6884    -30.050 0.98450  1.634e-12
 60   1498030  0.05105   27.0704   29.9828   28.5266     188.7181    137.400     192.5140    382.943  0.0479
 ↪       180.7324     -6.263 0.98461  1.219e-12
 80   1497370  0.04843   27.1538   29.9180   28.5359     189.4561    138.766     194.7125    434.056  0.0448
 ↪       180.7273    -55.674 0.98460  1.619e-12
100   1496805  0.04613   27.2314   29.8651   28.5482     190.3462    183.380     197.6103    614.998  0.0420
↪       180.7450     46.907 0.98475  8.238e-13
120   1496399  0.04448   27.2953   29.8366   28.5660     191.5205    235.327     201.6616    826.663  0.0400
↪       180.7603     57.173 0.98477  5.487e-13
140   1496223  0.04436   27.3070   29.8423   28.5747     255.4772    143.460     206.6364    930.090  0.0391
↪       180.7930      6.253 0.98481  1.350e-12
160   1496323  0.04652   27.2285   29.8855   28.5570     256.0942     81.290     211.6302    843.183  0.0398
↪       180.8434    -57.409 0.98479  2.708e-12
180   1496664  0.04938   27.1391   29.9585   28.5488     256.3820     28.546     215.7740    654.178  0.0417
↪       180.9648    118.853 0.98491  3.645e-12
200   1497155  0.05251   27.0482   30.0461   28.5471     256.4618      8.383     218.9493    514.602  0.0442
↪       181.1036     43.783 0.98489  4.807e-12
220   1497636  0.05543   26.9625   30.1272   28.5448     256.5913     53.981     221.8305    576.898  0.0467
↪       181.2831     51.193 0.98487  6.741e-12
240   1497909  0.05763   26.8845   30.1727   28.5286     257.2117    165.264     225.8991    892.321  0.0481
↪       181.4748    158.945 0.98498  8.532e-12
260   1498061  0.05932   26.8027   30.1831   28.4929     258.4957    295.076     232.1134   1344.957  0.0489
↪       181.5698     16.473 0.98504  9.061e-12
280   1498276  0.06068   26.7091   30.1599   28.4345     260.5447    413.817     240.9644   1719.605  0.0496
↪       181.5389    -18.952 0.98517  1.052e-11
300   1498450  0.06075   26.6463   30.0933   28.3698     262.8017    392.995     249.6054   1364.329  0.0501
↪       181.2639    -65.598 0.98530  9.452e-12
320   1497998  0.05748   26.6819   29.9366   28.3092     265.0358    427.448     256.7096   1290.555  0.0475
↪       180.8625   -101.366 0.98532  9.450e-12
340   1497378  0.05334   26.7382   29.7514   28.2448     267.7557    564.748     264.5550   1522.516  0.0446
↪       180.4805     42.306 0.98538  9.152e-12
360   1497254  0.05245   26.7452   29.7058   28.2255     271.4990    633.406     273.2883   1568.369  0.0443
↪       180.0608   -184.784 0.98528  9.070e-12
380   1497718  0.05634   26.6965   29.8843   28.2904     274.4143    454.352     281.4485   1357.255  0.0467
↪       179.6291   -164.088 0.98522  9.863e-12
400   1498416  0.06019   26.6461   30.0593   28.3527     276.7089    392.026     288.5538   1286.294  0.0497
↪       179.2602     44.784 0.98544  1.009e-11
420   1498571  0.06088   26.6882   30.1487   28.4184     278.9364    415.190     296.6715   1661.888  0.0499
↪       178.9602     40.158 0.98537  1.075e-11
440   1498328  0.05965   26.7826   30.1802   28.4814     281.1104    336.125     305.9064   1503.312  0.0490
↪       178.7289   -101.920 0.98508  1.168e-11
460   1498133  0.05806   26.8669   30.1788   28.5228     282.5614    198.283     312.7778   1013.598  0.0483
↪       178.6356    -12.883 0.98509  1.285e-11
480   1497883  0.05604   26.9446   30.1437   28.5441     283.3226     83.838     317.3118    658.925  0.0471
↪       178.6288      2.836 0.98497  1.471e-11
500   1497445  0.05328   27.0283   30.0702   28.5492     283.5262     11.564     320.3463    501.326  0.0449
↪       178.7316     56.481 0.98502  1.462e-11
520   1496944  0.05015   27.1185   29.9818   28.5502     283.5886     18.867     323.3536    601.411  0.0423
↪       178.8996     29.466 0.98491  1.478e-11
540   1496555  0.04717   27.2090   29.9031   28.5561     283.8146     64.822     327.2345    790.127  0.0402
↪       179.0873     30.796 0.98488  1.383e-11
```

 doi:10.20944/preprints202405.2028.v8


```
560  1496377  0.04480   27.2909   29.8506   28.5707    284.3539   126.868    332.0752   923.443  0.0391
↪      179.2572     27.148 0.98485  1.306e-11
580  1496455  0.04424   27.3067   29.8345   28.5706    348.1784   246.959    337.1575   875.554  0.0396
↪      179.3829     -0.214 0.98473  1.273e-11
600  1496773  0.04566   27.2473   29.8544   28.5508    349.4157   198.318    341.4936   681.496  0.0414
↪      179.4649     11.753 0.98477  1.479e-11
620  1497274  0.04783   27.1720   29.9017   28.5368    350.3651   148.400    344.6601   478.294  0.0441
↪      179.4993      6.664 0.98467  1.568e-11
640  1497906  0.05039   27.0894   29.9646   28.5270    351.1153   131.083    346.9633   380.107  0.0471
↪      179.5136      4.438 0.98466  1.692e-11
660  1498630  0.05301   27.0048   30.0281   28.5165    351.9713   193.355    349.2268   470.414  0.0503
↪      179.5470     13.945 0.98459  1.787e-11
680  1499386  0.05508   26.9244   30.0631   28.4938    353.7217   460.759    352.9536   880.020  0.0534
↪      179.6908     40.703 0.98455  1.712e-11
700  1499831  0.05524   26.8805   30.0237   28.4521    357.7339  1166.304    358.6263  1094.620  0.0552
↪      179.9359     49.216 0.98445  1.737e-11
720  1499660  0.05550   26.8965   30.0572   28.4769      4.8896   663.692      4.6379  1029.487  0.0545
↪      180.2085     38.956 0.98460  1.609e-11
740  1498973  0.05407   26.9674   30.0502   28.5088      7.4348   300.095      9.4056   655.684  0.0517
↪      180.4099     26.985 0.98455  1.430e-11
760  1498222  0.05159   27.0514   29.9942   28.5228      8.5419   147.706     12.0807   395.150  0.0485
↪      180.4782      7.415 0.98460  1.654e-11
780  1497544  0.04895   27.1354   29.9288   28.5321      9.2856   133.279     14.2373   410.934  0.0454
↪      180.4926     -8.673 0.98470  1.651e-11
800  1496979  0.04656   27.2148   29.8729   28.5438     10.1230   171.594     16.9426   572.700  0.0425
↪      180.5143     -6.552 0.98475  1.745e-11
820  1496572  0.04476   27.2827   29.8394   28.5610     11.2318   224.657     20.7581   788.904  0.0403
↪      180.5723     15.589 0.98479  1.687e-11
840  1496380  0.04403   27.3220   29.8386   28.5803     75.2919   156.232     25.6062   924.420  0.0392
↪      180.6783     23.781 0.98483  1.559e-11
860  1496445  0.04601   27.2474   29.8755   28.5615     75.9861    95.323     30.6834   873.567  0.0395
↪      180.8295     28.661 0.98492  1.641e-11
880  1496748  0.04876   27.1596   29.9442   28.5519     76.3419    38.324     35.0345   692.705  0.0412
↪      181.0154     31.137 0.98490  1.659e-11
900  1497214  0.05188   27.0683   30.0306   28.5494     76.4514     8.971     38.3739   532.401  0.0437
↪      181.1965     -5.133 0.98498  1.408e-11
920  1497712  0.05489   26.9810   30.1149   28.5480     76.5356    32.393     41.2132   530.746  0.0463
↪      181.3449     12.797 0.98496  1.408e-11
940  1498038  0.05722   26.9018   30.1675   28.5347     77.0183   139.336     44.9424   819.494  0.0479
↪      181.3829     -3.004 0.98499  1.316e-11
960  1498236  0.05897   26.8220   30.1836   28.5028     78.1432   263.400     50.6683  1244.496  0.0487
↪      181.3269     32.806 0.98509  1.244e-11
980  1498461  0.06038   26.7308   30.1661   28.4484     80.0323   403.985     59.0641  1713.584  0.0494
↪      181.1649     35.652 0.98512  1.201e-11
991  1498588  0.06097   26.6795   30.1438   28.4116     81.2956   414.361     64.2530  1644.361  0.0500
↪      181.0391    -42.975 0.98527  1.162e-11
992  1498593  0.06100   26.6754   30.1411   28.4082     81.4104   413.230     64.7040  1623.847  0.0501
↪      181.0272    -42.951 0.98528  1.138e-11
993  1498595  0.06102   26.6715   30.1381   28.4048     81.5247   411.526     65.1483  1599.461  0.0501
↪      180.9904   -132.262 0.98540  1.132e-11
994  1498595  0.06104   26.6679   30.1350   28.4014     81.6384   409.459     65.5855  1573.690  0.0501
↪      181.0024     43.062 0.98528  1.130e-11
995  1498594  0.06104   26.6645   30.1315   28.3980     81.7516   407.282     66.0157  1548.860  0.0502
↪      180.9900    -44.515 0.98529  1.131e-11
```

 doi:10.20944/preprints202405.2028.v8


```
 996   1498591  0.06104   26.6615   30.1279   28.3947      81.8642    405.318      66.4392   1524.665  0.0502
 ↪       180.9774    -45.379 0.98529  1.146e-11
 997   1498587  0.06103   26.6587   30.1240   28.3914      81.9762    403.291      66.8563   1501.320  0.0502
 ↪       180.9646    -46.174 0.98529  1.145e-11
 998   1498581  0.06101   26.6562   30.1199   28.3881      82.0877    401.517      67.2671   1478.865  0.0502
 ↪       180.9516    -46.896 0.98528  1.157e-11
 999   1498574  0.06098   26.6539   30.1156   28.3848      82.1988    399.720      67.6718   1457.192  0.0502
 ↪       180.9121   -142.043 0.98540  1.183e-11
1000   1498566  0.06094   26.6519   30.1110   28.3815      82.3094    398.182      68.0709   1436.581  0.0502
↪       180.9241     43.188 0.98528  1.201e-11

PER-ORBIT SUMMARY (analysis horizon)
------------------------------------
e_rmin: mean=0.052571  std=0.005288  first=0.054901  last=0.060940

APSIS STATISTICS (exclude first orbit)
--------------------------------------
\Delta\omega_peri: mean=943.154670  \sigma=12356.468237   median=191.539555  MAD\sigma=179.154346
\Delta\omega_harm: mean=893.396954  \sigma=385.307264   median=827.078232  MAD\sigma=406.704012
\Delta\omega_axis: mean=3.281830  \sigma=60.677576   median=14.756248  MAD\sigma=36.814146

APSIS STATISTICS (well-conditioned subset)
------------------------------------------
N_well = 999 / 999
\Delta\omega_peri: mean=943.154670  \sigma=12356.468237   median=191.539555  MAD\sigma=179.154346
\Delta\omega_harm: mean=893.396954  \sigma=385.307264   median=827.078232  MAD\sigma=406.704012
\Delta\omega_axis: mean=3.281830  \sigma=60.677576   median=14.756248  MAD\sigma=36.814146

BLOCK SUMMARY (block size = 100 orbits)
---------------------------------------
block      orbits   <e_rmin>  \sigma(e_rmin)  <\Delta\omega_axis>     \sigma_axis   N_well
    1    1- 100   0.051797   0.003094     26.604     45.084       99
    2  101- 200   0.046752   0.002564     13.096     54.254       99
    3  201- 300   0.058061   0.002529      5.382     97.196       99
    4  301- 400   0.055918   0.002838    -72.178     86.152       99
    5  401- 500   0.058330   0.002322    -17.649     52.173       99
    6  501- 600   0.047029   0.002819     26.786     18.569       99
    7  601- 700   0.051452   0.003236     17.015     18.303       99
    8  701- 800   0.052168   0.002950     20.506     20.447       99
    9  801- 900   0.046511   0.002298     24.722     18.852       99
   10  901-1000   0.057687   0.002683    -10.506     47.798       99
```

Appendix C.4. Appendix B.10. Validation III–B1: Three-body baseline (1-year inner-orbit window)

Cross-reference. Appendix B, Validation III–B1.

Appendix B.3.4.1. Appendix B.10.1. Executable script

```
import numpy as np
import math
import time
from numba import njit
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D  # noqa: F401

# ============================================================
```

 doi:10.20944/preprints202405.2028.v8


```
# THREE-BODY (IV-B REDO) — 1 INNER ORBIT — "NEAR-PERFECT" SETUP
# Flux + Tilt + H_DO + High sampling + Sub-sample peri/apo
#
# Upgrades applied:
#  (1) Two-stage V_OUT refinement around best (coarse -> fine)
#  (2) Higher sampling density (>= 12,000 samples/inner-orbit)
#  (3) Quadratic sub-sample peri/apo (r_min/r_max, omega) + interpolated orbit boundary time
#
# Goal for THIS run:
#  - Inner completes 1 full orbit (2\pi crossing) — that's the "1 year" window
#  - Outer is mildly elliptical, tuned to OUTER_E_TARGET via e_rmin over inner-orbit window
#  - Earth mass M_E fixed identical to your two-body value (81.297)
# ============================================================

# ---------------------------
# TUNABLE PARAMETERS (all here)
# ---------------------------

DT = 3.0e-3

# Masses (Earth fixed for comparability)
M_E   = 81.297
M_IN  = 1.0
M_OUT = 1.0
MASSES = np.array([M_E, M_IN, M_OUT], dtype=np.float64)

# Radii hierarchy
D_IN  = 30.0
D_OUT = 40.0

# Tilt (same as your two-body/three-body tilted work)
INCL_DEG = 5.145
INCL_RAD = INCL_DEG * math.pi / 180.0

# Flux kernel + coupling
R_MAX = 200
CHI_FLUX = 24.0
CHI_E = 0.00057032984

# Inner speed factor (hold fixed for now)
V_FACTOR_IN = 0.939199

# Outer ellipticity target (radius-defined, over inner-orbit window)
OUTER_E_TARGET = 0.0800

# Inner-orbit step estimate (from your last 3-body run)
STEPS_PER_ORBIT_EST_INNER = 1_448_537

# Run length = (1 inner orbit + safety margin)
EXTRA_ORBITS = 1
STEPS_TOTAL = STEPS_PER_ORBIT_EST_INNER * (1 + EXTRA_ORBITS)

# Sampling density (upgrade #2)
SAMPLES_PER_ORBIT = 12000
SAMPLES_MAX = SAMPLES_PER_ORBIT * (1 + EXTRA_ORBITS)
```

 doi:10.20944/preprints202405.2028.v8


```
# Two-stage V_OUT refinement (upgrade #1)
STAGE1_MIN  = 1.025
STAGE1_MAX  = 1.035
STAGE1_STEP = 0.001

STAGE2_HALF_WIDTH = 0.001
STAGE2_STEP       = 0.0002

# Optional: tighten the “reporting” resolution
PRINT_V_DECIMALS = 9

# ----------------------------
# Candidate-A flux kernel
# ----------------------------

def build_flux_kernel(r_max, chi_flux):
    k = np.arange(1, r_max + 1, dtype=np.float64)
    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k**2 + 2.0

    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0

    Phi = np.cumsum(counts * rho)

    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])

    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
        w_arr[d] = s

    return w_arr, gamma_arr

@njit(fastmath=True)
def get_field(d, w_arr, gamma_arr):
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]
    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0
    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1
    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]
    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w
    return w_val, g_val
```

 doi:10.20944/preprints202405.2028.v8


```
# ---------------------------
# Orbital-plane basis (fixed rotation about x-axis)
# ---------------------------

def orbital_plane_basis(incl_rad):
    c = math.cos(incl_rad)
    s = math.sin(incl_rad)
    e_xp = np.array([1.0, 0.0, 0.0], dtype=np.float64)
    e_yp = np.array([0.0, c,  s], dtype=np.float64)
    e_zp = np.array([0.0, -s, c], dtype=np.float64)
    return e_xp, e_yp, e_zp

# ---------------------------
# Initial conditions (COM=0, total momentum=0)
# ---------------------------

def init_three_body(w_arr, gamma_arr, v_in_factor, v_out_factor):
    # Positions: Earth at 0, inner at +D_IN, outer at -D_OUT along x
    pos = np.zeros((3, 3), dtype=np.float64)
    pos[0, 0] = 0.0
    pos[1, 0] = D_IN
    pos[2, 0] = -D_OUT

    # Shift COM to origin
    Mtot = MASSES.sum()
    COM0 = (pos[0] * MASSES[0] + pos[1] * MASSES[1] + pos[2] * MASSES[2]) / Mtot
    pos -= COM0

    # Tilted tangential direction (y-z)
    t_hat = np.array([0.0, math.cos(INCL_RAD), math.sin(INCL_RAD)], dtype=np.float64)

    # Flux-native circular speeds for Earth-inner and Earth-outer
    _, g_in = get_field(D_IN, w_arr, gamma_arr)
    F_in = CHI_E * MASSES[0] * MASSES[1] * abs(g_in)
    mu_in = (MASSES[0] * MASSES[1]) / (MASSES[0] + MASSES[1])
    v_circ_in = math.sqrt(F_in * D_IN / mu_in)
    v_rel_in = v_in_factor * v_circ_in

    _, g_out = get_field(D_OUT, w_arr, gamma_arr)
    F_out = CHI_E * MASSES[0] * MASSES[2] * abs(g_out)
    mu_out = (MASSES[0] * MASSES[2]) / (MASSES[0] + MASSES[2])
    v_circ_out = math.sqrt(F_out * D_OUT / mu_out)
    v_rel_out = v_out_factor * v_circ_out

    # Scalar speeds along t_hat with total momentum = 0:
    # (b-a)=v_rel_in, (c-a)=v_rel_out, mE*a+mI*b+mO*c=0
    mE, mI, mO = MASSES[0], MASSES[1], MASSES[2]
    Mtot = mE + mI + mO
    a = -(mI * v_rel_in + mO * v_rel_out) / Mtot
    b = a + v_rel_in
    c = a + v_rel_out

    vel = np.zeros((3, 3), dtype=np.float64)
    vel[0] = a * t_hat
```

 doi:10.20944/preprints202405.2028.v8


```
    vel[1] = b * t_hat
    vel[2] = c * t_hat
    return pos, vel

# ---------------------------
# Multi-body forces + potential (in-place; avoids allocations)
# ---------------------------

@njit(fastmath=True)
def forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_out, F_out):
    for i in range(3):
        F_out[i, 0] = 0.0
        F_out[i, 1] = 0.0
        F_out[i, 2] = 0.0
    U = 0.0

    for i in range(2):
        mi = masses[i]
        for j in range(i + 1, 3):
            mj = masses[j]
            rx = pos[i, 0] - pos[j, 0]
            ry = pos[i, 1] - pos[j, 1]
            rz = pos[i, 2] - pos[j, 2]

            ax = abs(rx); ay = abs(ry); az = abs(rz)
            d_inf = ax
            if ay > d_inf: d_inf = ay
            if az > d_inf: d_inf = az

            w_val, g_val = get_field(d_inf, w_arr, gamma_arr)
            U += -chi_e * mi * mj * w_val

            r1 = ax + ay + az
            if r1 > 0.0:
                ux = rx / r1; uy = ry / r1; uz = rz / r1
                f_mag = chi_e * mi * mj * g_val  # g_val<0 => attraction
                fx = f_mag * ux; fy = f_mag * uy; fz = f_mag * uz
                F_out[i, 0] += fx; F_out[i, 1] += fy; F_out[i, 2] += fz
                F_out[j, 0] -= fx; F_out[j, 1] -= fy; F_out[j, 2] -= fz

    for i in range(3):
        inv_m = 1.0 / masses[i]
        acc_out[i, 0] = F_out[i, 0] * inv_m
        acc_out[i, 1] = F_out[i, 1] * inv_m
        acc_out[i, 2] = F_out[i, 2] * inv_m

    return U

# ---------------------------
# Forest-Ruth + H_DO integrator (3-body, sampled)
# ---------------------------

@njit(fastmath=True)
def run_fr_3body(
    dt, steps, masses, chi_e, w_arr, gamma_arr,
```

 doi:10.20944/preprints202405.2028.v8


```
    pos0, vel0, samples_max
):
    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    pos = pos0.copy()
    vel = vel0.copy()

    sample_stride = steps // samples_max
    if sample_stride < 1:
        sample_stride = 1
    n_alloc = steps // sample_stride + 2

    t_hist = np.zeros(n_alloc, dtype=np.float64)
    E_hist = np.zeros(n_alloc, dtype=np.float64)
    H_hist = np.zeros(n_alloc, dtype=np.float64)
    Lx_hist = np.zeros(n_alloc, dtype=np.float64)
    Ly_hist = np.zeros(n_alloc, dtype=np.float64)
    Lz_hist = np.zeros(n_alloc, dtype=np.float64)
    pos_hist = np.zeros((n_alloc, 3, 3), dtype=np.float64)

    work_defect = 0.0

    F_s = np.zeros((3, 3), dtype=np.float64)
    F_e = np.zeros((3, 3), dtype=np.float64)
    acc_s = np.zeros((3, 3), dtype=np.float64)
    acc_e = np.zeros((3, 3), dtype=np.float64)

    U_s = forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_s, F_s)

    Mtot = masses[0] + masses[1] + masses[2]
    COM = (pos[0] * masses[0] + pos[1] * masses[1] + pos[2] * masses[2]) / Mtot

    ke = 0.5 * (
        masses[0] * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
        masses[1] * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2) +
        masses[2] * (vel[2,0]**2 + vel[2,1]**2 + vel[2,2]**2)
    )
    E_naive = ke + U_s
    H_DO = E_naive - work_defect

    r0 = pos[0] - COM; r1 = pos[1] - COM; r2 = pos[2] - COM
    p0 = masses[0]*vel[0]; p1 = masses[1]*vel[1]; p2 = masses[2]*vel[2]
    L = np.cross(r0, p0) + np.cross(r1, p1) + np.cross(r2, p2)

    idx = 0
    t_hist[idx] = 0.0
    E_hist[idx] = E_naive
    H_hist[idx] = H_DO
    Lx_hist[idx] = L[0]; Ly_hist[idx] = L[1]; Lz_hist[idx] = L[2]
    pos_hist[idx] = pos
```

 doi:10.20944/preprints202405.2028.v8


```
idx += 1

for step in range(1, steps + 1):
    for s in range(4):
        dr0x = c[s] * dt * vel[0,0]; dr0y = c[s] * dt * vel[0,1]; dr0z = c[s] * dt * vel[0,2]
        dr1x = c[s] * dt * vel[1,0]; dr1y = c[s] * dt * vel[1,1]; dr1z = c[s] * dt * vel[1,2]
        dr2x = c[s] * dt * vel[2,0]; dr2y = c[s] * dt * vel[2,1]; dr2z = c[s] * dt * vel[2,2]

        pos[0,0] += dr0x; pos[0,1] += dr0y; pos[0,2] += dr0z
        pos[1,0] += dr1x; pos[1,1] += dr1y; pos[1,2] += dr1z
        pos[2,0] += dr2x; pos[2,1] += dr2y; pos[2,2] += dr2z

        U_e = forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_e, F_e)

        work_total = 0.0
        for i in range(3):
            favg0 = 0.5 * (F_s[i,0] + F_e[i,0])
            favg1 = 0.5 * (F_s[i,1] + F_e[i,1])
            favg2 = 0.5 * (F_s[i,2] + F_e[i,2])
            if i == 0:
                work_total += favg0*dr0x + favg1*dr0y + favg2*dr0z
            elif i == 1:
                work_total += favg0*dr1x + favg1*dr1y + favg2*dr1z
            else:
                work_total += favg0*dr2x + favg1*dr2y + favg2*dr2z

        work_defect += (work_total + (U_e - U_s))

        if dcoef[s] != 0.0:
            vel[0,0] += dcoef[s]*dt*acc_e[0,0]; vel[0,1] += dcoef[s]*dt*acc_e[0,1]; vel[0,2] +=
            ↪  dcoef[s]*dt*acc_e[0,2]
            vel[1,0] += dcoef[s]*dt*acc_e[1,0]; vel[1,1] += dcoef[s]*dt*acc_e[1,1]; vel[1,2] +=
            ↪  dcoef[s]*dt*acc_e[1,2]
            vel[2,0] += dcoef[s]*dt*acc_e[2,0]; vel[2,1] += dcoef[s]*dt*acc_e[2,1]; vel[2,2] +=
            ↪  dcoef[s]*dt*acc_e[2,2]

        tmpF = F_s; F_s = F_e; F_e = tmpF
        tmpA = acc_s; acc_s = acc_e; acc_e = tmpA
        U_s = U_e

    if (step % sample_stride == 0) and (idx < n_alloc):
        COM = (pos[0] * masses[0] + pos[1] * masses[1] + pos[2] * masses[2]) / Mtot
        ke = 0.5 * (
            masses[0] * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
            masses[1] * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2) +
            masses[2] * (vel[2,0]**2 + vel[2,1]**2 + vel[2,2]**2)
        )
        E_naive = ke + U_s
        H_DO = E_naive - work_defect

        r0 = pos[0] - COM; r1 = pos[1] - COM; r2 = pos[2] - COM
        p0 = masses[0]*vel[0]; p1 = masses[1]*vel[1]; p2 = masses[2]*vel[2]
        L = np.cross(r0, p0) + np.cross(r1, p1) + np.cross(r2, p2)

        t_hist[idx] = step * dt
```

 doi:10.20944/preprints202405.2028.v8


```
            E_hist[idx] = E_naive
            H_hist[idx] = H_DO
            Lx_hist[idx] = L[0]; Ly_hist[idx] = L[1]; Lz_hist[idx] = L[2]
            pos_hist[idx] = pos
            idx += 1

    return idx, t_hist, E_hist, H_hist, Lx_hist, Ly_hist, Lz_hist, pos_hist

# ----------------------------
# Orbit window: boundary time via interpolation in plane-angle
# ----------------------------

def inner_orbit_boundary(t, Rx, Ry):
    phi = np.unwrap(np.arctan2(Ry, Rx))
    phi0 = float(phi[0])
    thr = phi0 + 2.0 * math.pi
    for i in range(1, len(phi)):
        if phi[i] >= thr:
            i_lo = i - 1
            i_hi = i
            denom = (phi[i_hi] - phi[i_lo])
            frac = 0.0 if denom == 0.0 else (thr - phi[i_lo]) / denom
            frac = min(1.0, max(0.0, frac))
            t_b = t[i_lo] + frac * (t[i_hi] - t[i_lo])
            return i_lo, i_hi, frac, t_b, float((phi[i_hi] - phi0) * 180.0 / math.pi)
    # no crossing
    return None

# ----------------------------
# Quadratic sub-sample extremum interpolation
# ----------------------------

def quad_vertex_offset(y_m1, y0, y_p1):
    denom = (y_m1 - 2.0 * y0 + y_p1)
    if denom == 0.0:
        return 0.0
    d = 0.5 * (y_m1 - y_p1) / denom
    return max(-1.0, min(1.0, d))

def quad_interp_at(d, f_m1, f0, f_p1):
    a = 0.5 * (f_p1 + f_m1 - 2.0 * f0)
    b = 0.5 * (f_p1 - f_m1)
    return a * d * d + b * d + f0

def extrema_interp(r, Rx, Ry, Rz):
    # peri (min)
    i_min = int(np.argmin(r))
    if 0 < i_min < (len(r) - 1):
        d = quad_vertex_offset(r[i_min-1], r[i_min], r[i_min+1])
        r_min = quad_interp_at(d, r[i_min-1], r[i_min], r[i_min+1])
        Rx_min = quad_interp_at(d, Rx[i_min-1], Rx[i_min], Rx[i_min+1])
        Ry_min = quad_interp_at(d, Ry[i_min-1], Ry[i_min], Ry[i_min+1])
    else:
        r_min = float(r[i_min]); Rx_min = float(Rx[i_min]); Ry_min = float(Ry[i_min])
```

 doi:10.20944/preprints202405.2028.v8


```
    # apo (max)
    i_max = int(np.argmax(r))
    if 0 < i_max < (len(r) - 1):
        d = quad_vertex_offset(r[i_max-1], r[i_max], r[i_max+1])
        r_max = quad_interp_at(d, r[i_max-1], r[i_max], r[i_max+1])
    else:
        r_max = float(r[i_max])

    e_r = (r_max - r_min) / (r_max + r_min) if (r_max + r_min) != 0.0 else 0.0
    a_r = 0.5 * (r_max + r_min)
    omega_deg = math.degrees(math.atan2(Ry_min, Rx_min))
    Rz_rms = float(np.std(Rz))
    return float(e_r), float(r_min), float(r_max), float(a_r), float(omega_deg), Rz_rms

# ---------------------------
# Evaluate one V_OUT (returns dict + optional arrays)
# ---------------------------

def eval_vout(v_out, w_arr, gamma_arr, return_series=False):
    e_xp, e_yp, e_zp = orbital_plane_basis(INCL_RAD)
    pos0, vel0 = init_three_body(w_arr, gamma_arr, V_FACTOR_IN, v_out)

    n_samples, t_hist, E_hist, H_hist, Lx, Ly, Lz, pos_hist = run_fr_3body(
        DT, STEPS_TOTAL, MASSES, CHI_E, w_arr, gamma_arr,
        pos0, vel0, SAMPLES_MAX
    )

    t_hist = t_hist[:n_samples]
    E_hist = E_hist[:n_samples]
    H_hist = H_hist[:n_samples]
    Lx = Lx[:n_samples]; Ly = Ly[:n_samples]; Lz = Lz[:n_samples]
    pos_hist = pos_hist[:n_samples]

    # COM + invariants
    Mtot = MASSES.sum()
    COM = (pos_hist[:,0,:]*MASSES[0] + pos_hist[:,1,:]*MASSES[1] + pos_hist[:,2,:]*MASSES[2]) / Mtot
    COM_drift = np.linalg.norm(COM - COM[0], axis=1).max()

    Lmag = np.sqrt(Lx**2 + Ly**2 + Lz**2)
    dL = (Lmag - Lmag[0]) / abs(Lmag[0])

    inc_raw = np.degrees(np.arccos(np.clip(Lz / Lmag, -1.0, 1.0)))
    inc_eff = np.minimum(inc_raw, 180.0 - inc_raw)
    node = np.degrees(np.arctan2(Lx, -Ly))

    # Relative vectors to Earth
    rel_in = pos_hist[:,1,:] - pos_hist[:,0,:]
    rel_out = pos_hist[:,2,:] - pos_hist[:,0,:]

    Rx_in = rel_in @ e_xp; Ry_in = rel_in @ e_yp; Rz_in = rel_in @ e_zp
    Rx_out = rel_out @ e_xp; Ry_out = rel_out @ e_yp; Rz_out = rel_out @ e_zp

    # orbit boundary
    b = inner_orbit_boundary(t_hist, Rx_in, Ry_in)
    if b is None:
```

 doi:10.20944/preprints202405.2028.v8


```
    return None

i_lo, i_hi, frac, t_b, dtheta_raw_deg = b
k_end = i_hi  # include i_hi sample

# window arrays
t_w = t_hist[:k_end+1]
E_w = E_hist[:k_end+1]
H_w = H_hist[:k_end+1]
dH = (H_w - H_w[0]) / abs(H_w[0])
dE = (E_w - E_w[0]) / abs(E_w[0])

# inner/outer radii in window
r_in = np.linalg.norm(rel_in[:k_end+1], axis=1)
r_out = np.linalg.norm(rel_out[:k_end+1], axis=1)

e_in, rmin_in, rmax_in, ar_in, omega_in, RzRMS_in = extrema_interp(r_in, Rx_in[:k_end+1],
↪  Ry_in[:k_end+1], Rz_in[:k_end+1])
e_out, rmin_out, rmax_out, ar_out, omega_out, RzRMS_out = extrema_interp(r_out, Rx_out[:k_end+1],
↪  Ry_out[:k_end+1], Rz_out[:k_end+1])

Nsteps = int(round((t_b - t_hist[0]) / DT))
dtheta_overshoot = dtheta_raw_deg - 360.0

out = {
    "v_out": float(v_out),
    "inner_Nsteps": int(Nsteps),
    "inner_dtheta_raw": float(dtheta_raw_deg),
    "inner_dtheta_overshoot": float(dtheta_overshoot),
    "Hnet": float(dH[-1]),
    "HRMS": float(np.std(dH)),
    "Enet": float(dE[-1]),
    "ERMS": float(np.std(dE)),
    "L_RMS": float(np.std(dL[:k_end+1])),
    "COM_drift": float(np.linalg.norm(COM[:k_end+1] - COM[0], axis=1).max()),
    "inc_mean": float(inc_eff[:k_end+1].mean()),
    "inc_std": float(inc_eff[:k_end+1].std()),
    "node_start": float(node[0]),
    "node_end": float(node[k_end]),
    "inner_e": float(e_in),
    "inner_rmin": float(rmin_in),
    "inner_rmax": float(rmax_in),
    "inner_ar": float(ar_in),
    "inner_omega": float(omega_in),
    "inner_RzRMS": float(RzRMS_in),
    "outer_e": float(e_out),
    "outer_rmin": float(rmin_out),
    "outer_rmax": float(rmax_out),
    "outer_ar": float(ar_out),
    "outer_omega": float(omega_out),
    "outer_RzRMS": float(RzRMS_out),
    "t_end": float(t_b),
    "k_end": int(k_end),
}
```

 doi:10.20944/preprints202405.2028.v8


```
    if return_series:
        out["_series"] = {
            "t": t_w,
            "Rx_in": Rx_in[:k_end+1], "Ry_in": Ry_in[:k_end+1], "Rz_in": Rz_in[:k_end+1],
            "Rx_out": Rx_out[:k_end+1], "Ry_out": Ry_out[:k_end+1], "Rz_out": Rz_out[:k_end+1],
            "r_in": r_in, "r_out": r_out,
            "dH": dH, "dE": dE,
            "dL": dL[:k_end+1],
            "pos": pos_hist[:k_end+1],
            "COM": COM[:k_end+1],
            "inc": inc_eff[:k_end+1],
            "node": node[:k_end+1],
        }
    return out

# ----------------------------
# Plot helper
# ----------------------------

def plot_final(res):
    s = res["_series"]
    t = s["t"]

    # 3D trajectories (downsample for speed)
    pos = s["pos"]
    COM = s["COM"]
    ds = max(1, len(t)//5000)
    idxs = np.arange(0, len(t), ds, dtype=int)

    fig = plt.figure(figsize=(7, 6))
    ax = fig.add_subplot(111, projection="3d")
    ax.plot(pos[idxs, 0, 0], pos[idxs, 0, 1], pos[idxs, 0, 2], label="Earth")
    ax.plot(pos[idxs, 1, 0], pos[idxs, 1, 1], pos[idxs, 1, 2], label="Inner")
    ax.plot(pos[idxs, 2, 0], pos[idxs, 2, 1], pos[idxs, 2, 2], label="Outer")
    ax.plot(COM[idxs, 0], COM[idxs, 1], COM[idxs, 2], label="COM")
    ax.set_title("3-body trajectories (inner-orbit window)")
    ax.set_xlabel("x (hops)"); ax.set_ylabel("y (hops)"); ax.set_zlabel("z (hops)")
    ax.legend()
    plt.tight_layout()
    plt.show()

    # orbital-plane orbits
    plt.figure(figsize=(6, 6))
    plt.plot(s["Rx_in"], s["Ry_in"], lw=0.8, label="Inner (plane)")
    plt.plot(s["Rx_out"], s["Ry_out"], lw=0.8, label="Outer (plane)")
    plt.scatter([0.0], [0.0], s=20, label="Earth")
    plt.xlabel("x' (hops)"); plt.ylabel("y' (hops)")
    plt.title("Orbital-plane relative orbits (inner-orbit window)")
    plt.axis("equal"); plt.grid(True, alpha=0.3); plt.legend()
    plt.tight_layout()
    plt.show()

    # r(t)
    plt.figure(figsize=(7, 4))
    plt.plot(t, s["r_in"], lw=0.9, label="Inner r(t)")
```

 doi:10.20944/preprints202405.2028.v8


```
    plt.plot(t, s["r_out"], lw=0.9, label="Outer r(t)")
    plt.xlabel("t (code units)"); plt.ylabel("r (hops)")
    plt.title("r(t) over inner-orbit window")
    plt.grid(True, alpha=0.3); plt.legend()
    plt.tight_layout()
    plt.show()

    # energy diagnostics
    plt.figure(figsize=(7, 4))
    plt.plot(t, s["dH"], lw=0.9, label="\DeltaH_DO/H0")
    plt.plot(t, s["dE"], lw=0.9, alpha=0.7, label="\DeltaE_naive/E0")
    plt.xlabel("t (code units)"); plt.ylabel("\DeltaE/E0")
    plt.title("Energy diagnostics (inner-orbit window)")
    plt.grid(True, alpha=0.3); plt.legend()
    plt.tight_layout()
    plt.show()

    # |L|
    plt.figure(figsize=(7, 4))
    plt.plot(t, s["dL"], lw=0.9)
    plt.xlabel("t (code units)"); plt.ylabel("\Delta|L|/|L0|")
    plt.title("|L| drift (inner-orbit window)")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # inc/node
    plt.figure(figsize=(7, 4))
    plt.plot(t, s["inc"], lw=0.9)
    plt.xlabel("t (code units)"); plt.ylabel("inclination (deg, eff)")
    plt.title("Inclination vs time (eff convention)")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(7, 4))
    plt.plot(t, s["node"], lw=0.9)
    plt.xlabel("t (code units)"); plt.ylabel("node angle (deg)")
    plt.title("Node angle vs time")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

# ----------------------------
# MAIN
# ----------------------------

if __name__ == "__main__":
    print("Building Candidate-A flux kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)

    print("Warming up Numba JIT...")
    pos_w, vel_w = init_three_body(w_arr, gamma_arr, V_FACTOR_IN, 1.03)
    _ = run_fr_3body(DT, 50_000, MASSES, CHI_E, w_arr, gamma_arr, pos_w, vel_w, 5000)
    print("Warm-up done.\n")
```

 doi:10.20944/preprints202405.2028.v8


```
# -----------------------
# Stage 1 scan
# -----------------------
v1 = np.arange(STAGE1_MIN, STAGE1_MAX + 0.5*STAGE1_STEP, STAGE1_STEP)
stage1 = []
print("STAGE 1: V_OUT scan")
print("--------------------")
print(f"Target outer e_rmin = {OUTER_E_TARGET:.6f}\n")
print(f"{'idx':>3} {'V_OUT':>14} {'outer_e':>10} {'|err|':>12} {'outer_rmin':>11} {'outer_rmax':>11} "
      f"{'inner_e':>10} {'Nsteps':>9} {'\Delta\theta_raw':>10} {'\Delta\theta-360':>10} {'HRMS':>11}")

tscan0 = time.time()
for i, vout in enumerate(v1):
    res = eval_vout(float(vout), w_arr, gamma_arr, return_series=False)
    if res is None:
        print(f"{i:3d} {vout:14.{PRINT_V_DECIMALS}f}  NO_ORBIT")
        continue
    err = abs(res["outer_e"] - OUTER_E_TARGET)
    stage1.append((err, res))
    print(f"{i:3d} {vout:14.{PRINT_V_DECIMALS}f} {res['outer_e']:10.6f} {err:12.6e} "
          f"{res['outer_rmin']:11.4f} {res['outer_rmax']:11.4f} "
          f"{res['inner_e']:10.6f} {res['inner_Nsteps']:9d} "
          f"{res['inner_dtheta_raw']:10.4f} {res['inner_dtheta_overshoot']:10.6f} "
          f"{res['HRMS']:11.3e}")

tscan1 = time.time()
if len(stage1) == 0:
    raise SystemExit("Stage 1 produced no valid orbits. Increase STEPS_TOTAL or widen scan.")

stage1.sort(key=lambda x: x[0])
best1 = stage1[0][1]
v_best1 = best1["v_out"]
print(f"\nStage 1 best V_OUT = {v_best1:.{PRINT_V_DECIMALS}f}  (outer_e={best1['outer_e']:.6f},
↪  |err|={stage1[0][0]:.3e})")
print(f"Stage 1 time: {tscan1 - tscan0:.2f} s\n")

# -----------------------
# Stage 2 scan (fine)
# -----------------------
v2_min = v_best1 - STAGE2_HALF_WIDTH
v2_max = v_best1 + STAGE2_HALF_WIDTH
v2 = np.arange(v2_min, v2_max + 0.5*STAGE2_STEP, STAGE2_STEP)

stage2 = []
print("STAGE 2: V_OUT refinement")
print("-------------------------")
print(f"Refine in [{v2_min:.{PRINT_V_DECIMALS}f}, {v2_max:.{PRINT_V_DECIMALS}f}] step={STAGE2_STEP}\n")
print(f"{'idx':>3} {'V_OUT':>14} {'outer_e':>10} {'|err|':>12} {'outer_rmin':>11} {'outer_rmax':>11} "
      f"{'inner_e':>10} {'Nsteps':>9} {'\Delta\theta-360':>10} {'HRMS':>11}")

for i, vout in enumerate(v2):
    res = eval_vout(float(vout), w_arr, gamma_arr, return_series=False)
    if res is None:
        print(f"{i:3d} {vout:14.{PRINT_V_DECIMALS}f}  NO_ORBIT")
```

 doi:10.20944/preprints202405.2028.v8


```
        continue
    err = abs(res["outer_e"] - OUTER_E_TARGET)
    stage2.append((err, res))
    print(f"{i:3d} {vout:14.{PRINT_V_DECIMALS}f} {res['outer_e']:10.6f} {err:12.6e} "
          f"{res['outer_rmin']:11.4f} {res['outer_rmax']:11.4f} "
          f"{res['inner_e']:10.6f} {res['inner_Nsteps']:9d} "
          f"{res['inner_dtheta_overshoot']:10.6f} {res['HRMS']:11.3e}")

if len(stage2) == 0:
    raise SystemExit("Stage 2 produced no valid orbits.")

stage2.sort(key=lambda x: x[0])
best2 = stage2[0][1]
v_best2 = best2["v_out"]
print(f"\nStage 2 best V_OUT = {v_best2:.{PRINT_V_DECIMALS}f}  (outer_e={best2['outer_e']:.6f},
↪  |err|={stage2[0][0]:.3e})\n")

# Plot scan curve (outer_e vs V_OUT)
v_plot = np.array([x[1]["v_out"] for x in stage2], dtype=np.float64)
e_plot = np.array([x[1]["outer_e"] for x in stage2], dtype=np.float64)
o = np.argsort(v_plot)
v_plot = v_plot[o]; e_plot = e_plot[o]

plt.figure(figsize=(7, 4))
plt.plot(v_plot, e_plot, marker="o", lw=0.8, label="outer e_rmin")
plt.axhline(OUTER_E_TARGET, lw=0.8, label="target")
plt.xlabel("V_OUT")
plt.ylabel("outer e_rmin (over inner-orbit window)")
plt.title("Stage 2 refinement: outer eccentricity vs V_OUT")
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

# ------------------------
# Final run with plots
# ------------------------
print("FINAL DETAILED RUN (best V_OUT, plots)")
print("--------------------------------------")
final = eval_vout(float(v_best2), w_arr, gamma_arr, return_series=True)

print("\nFINAL 3-BODY 1-ORBIT SUMMARY (INNER ORBIT WINDOW)")
print("-------------------------------------------------")
print(f"dt               = {DT}")
print(f"tilt (deg)       = {INCL_DEG}")
print(f"masses           = [Earth={MASSES[0]}, Inner={MASSES[1]}, Outer={MASSES[2]}]")
print(f"radii            = [Inner={D_IN}, Outer={D_OUT}]")
print(f"V_IN             = {V_FACTOR_IN}")
print(f"V_OUT (best)     = {final['v_out']:.{PRINT_V_DECIMALS}f}")
print(f"inner_Nsteps     = {final['inner_Nsteps']}")
print(f"inner_\Delta\theta_raw      = {final['inner_dtheta_raw']:.6f}  (overshoot
↪  {final['inner_dtheta_overshoot']:.6f} deg)")
print(f"H_DO net/RMS     = {final['Hnet']:.3e} / {final['HRMS']:.3e}")
print(f"Naive E net/RMS  = {final['Enet']:.3e} / {final['ERMS']:.3e}")
print(f"|L| RMS          = {final['L_RMS']:.3e}")
```

 doi:10.20944/preprints202405.2028.v8


```
    print(f"COM drift (hops) = {final['COM_drift']:.3e}")
    print(f"inc mean/std     = {final['inc_mean']:.6f} / {final['inc_std']:.6f}")
    print(f"node start/end   = {final['node_start']:.6f} -> {final['node_end']:.6f}")
    print()
    print(f"INNER  e_rmin={final['inner_e']:.6f}  rmin={final['inner_rmin']:.6f}
    ↪  rmax={final['inner_rmax']:.6f}  a_r={final['inner_ar']:.6f}  omega={final['inner_omega']:.6f}
    ↪  RzRMS={final['inner_RzRMS']:.3e}")
    print(f"OUTER  e_rmin={final['outer_e']:.6f}  rmin={final['outer_rmin']:.6f}
    ↪  rmax={final['outer_rmax']:.6f}  a_r={final['outer_ar']:.6f}  omega={final['outer_omega']:.6f}
    ↪  RzRMS={final['outer_RzRMS']:.3e}")
    print()

    plot_final(final)
```

Appendix B.3.4.2. Appendix B.10.2. Console transcript

```
Building Candidate-A flux kernel...
Warming up Numba JIT...
Warm-up done.

STAGE 1: V_OUT scan
--------------------
Target outer e_rmin = 0.080000

idx          V_OUT    outer_e        |err|  outer_rmin  outer_rmax    inner_e    Nsteps     \Delta\theta_raw
↪      \Delta\theta-360        HRMS
  0    1.025000000   0.067462 1.253836e-02     40.0000     45.7874   0.075667   1449691   360.0064   0.006350
  ↪     2.417e-13
  1    1.026000000   0.069592 1.040812e-02     40.0000     45.9838   0.075690   1449734   360.0232   0.023195
  ↪     2.539e-13
  2    1.027000000   0.071720 8.279691e-03     40.0000     46.1809   0.075713   1449777   360.0139   0.013864
  ↪     2.944e-13
  3    1.028000000   0.073848 6.152157e-03     40.0000     46.3789   0.075736   1449819   360.0047   0.004655
  ↪     2.774e-13
  4    1.029000000   0.075976 4.023880e-03     40.0000     46.5778   0.075759   1449860   360.0219   0.021863
  ↪     2.810e-13
  5    1.030000000   0.078108 1.892262e-03     40.0000     46.7780   0.075781   1449901   360.0129   0.012941
  ↪     2.876e-13
  6    1.031000000   0.080243 2.433781e-04     40.0000     46.9795   0.075803   1449941   360.0042   0.004166
  ↪     2.738e-13
  7    1.032000000   0.082383 2.382626e-03     40.0000     47.1823   0.075825   1449980   360.0218   0.021816
  ↪     3.668e-13
  8    1.033000000   0.084525 4.525311e-03     40.0000     47.3864   0.075847   1450019   360.0134   0.013413
  ↪     3.442e-13
  9    1.034000000   0.086672 6.671852e-03     40.0000     47.5917   0.075869   1450056   360.0053   0.005273
  ↪     2.537e-13
 10    1.035000000   0.088823 8.822582e-03     40.0000     47.7985   0.075891   1450092   360.0237
  ↪  0.023651   2.525e-13

Stage 1 best V_OUT = 1.031000000  (outer_e=0.080243, |err|=2.434e-04)
Stage 1 time: 13.28 s

STAGE 2: V_OUT refinement
--------------------------
Refine in [1.030000000, 1.032000000] step=0.0002
```

 doi:10.20944/preprints202405.2028.v8


```
idx          V_OUT    outer_e       |err|  outer_rmin  outer_rmax    inner_e    Nsteps     \Delta\theta-360
↪         HRMS
  0    1.030000000   0.078108 1.892262e-03     40.0000     46.7780   0.075781   1449901   0.012941   2.876e-13
  1    1.030200000   0.078535 1.465469e-03     40.0000     46.8182   0.075786   1449909   0.011174   2.400e-13
  2    1.030400000   0.078961 1.038511e-03     40.0000     46.8585   0.075790   1449917   0.009413   3.163e-13
  3    1.030600000   0.079389 6.113863e-04     40.0000     46.8988   0.075794   1449925   0.007658   3.269e-13
  4    1.030800000   0.079816 1.840908e-04     40.0000     46.9391   0.075799   1449933   0.005909   3.996e-13
  5    1.031000000   0.080243 2.433781e-04     40.0000     46.9795   0.075803   1449941   0.004166   2.738e-13
  6    1.031200000   0.080671 6.710039e-04     40.0000     47.0200   0.075808   1449949   0.002429   3.395e-13
  7    1.031400000   0.081099 1.098721e-03     40.0000     47.0605   0.075812   1449957   0.000698   3.105e-13
  8    1.031600000   0.081527 1.526558e-03     40.0000     47.1010   0.075817   1449965   0.025245   2.918e-13
  9    1.031800000   0.081955 1.954525e-03     40.0000     47.1417   0.075821   1449973   0.023526   2.238e-13
 10    1.032000000   0.082383 2.382626e-03     40.0000     47.1823   0.075825   1449980   0.021816   3.668e-13

Stage 2 best V_OUT = 1.030800000  (outer_e=0.079816, |err|=1.841e-04)

FINAL DETAILED RUN (best V_OUT, plots)
------------------------------------

FINAL 3-BODY 1-ORBIT SUMMARY (INNER ORBIT WINDOW)
------------------------------------------------
dt               = 0.003
tilt (deg)       = 5.145
masses           = [Earth=81.297, Inner=1.0, Outer=1.0]
radii            = [Inner=30.0, Outer=40.0]
V_IN             = 0.939199
V_OUT (best)     = 1.030800000
inner_Nsteps     = 1449933
inner_\Delta\theta_raw      = 360.005909  (overshoot 0.005909 deg)
H_DO net/RMS     = 5.047e-13 / 3.996e-13
Naive E net/RMS  = 3.849e-02 / 1.423e-01
|L| RMS          = 5.380e-13
COM drift (hops) = 1.860e-13
inc mean/std     = 5.145000 / 0.000000
node start/end   = 180.000000 -> 180.000000

INNER  e_rmin=0.075799  rmin=25.772503  rmax=30.000000  a_r=27.886252  omega=177.655553  RzRMS=9.686e-14
OUTER  e_rmin=0.079816  rmin=40.000000  rmax=46.939125  a_r=43.469563  omega=180.000000  RzRMS=1.340e-13
```

Appendix C.5. Appendix B.11. Validation III–B2: Three-body long horizon (1000 inner orbits)

Cross-reference. Appendix B, Validation III–B2.

Appendix B.3.5.1. Appendix B.11.1. Executable script

```
import numpy as np
import math
import time
from numba import njit
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D  # noqa: F401

# ============================================================
# IV-B THREE-BODY — FLUX + TILT + H_DO — 1000 INNER ORBITS
# SAME AS LAST RUN, WITH TWO ADDED PLOTS ONLY:
#   (1) 3D trajectories (Earth, Inner, Outer, COM) downsampled
```

 doi:10.20944/preprints202405.2028.v8


```
#   (2) 2D orbital-plane tracks (Inner about Earth; Outer about COM) downsampled
#
# NO OTHER CHANGES.
# ============================================================

# ---------------------------
# PARAMETERS (UNCHANGED)
# ---------------------------

DT = 3.0e-3

M_E, M_IN, M_OUT = 81.297, 1.0, 1.0
MASSES = np.array([M_E, M_IN, M_OUT], dtype=np.float64)

D_IN, D_OUT = 30.0, 40.0

INCL_DEG = 5.145
INCL_RAD = INCL_DEG * math.pi / 180.0

R_MAX = 200
CHI_FLUX = 24.0
CHI_E = 0.00057032984

V_FACTOR_IN  = 0.939199
V_FACTOR_OUT = 1.030800

N_INNER_ORBITS_TARGET = 1000
EXTRA_INNER_ORBITS_RUN = 12

STEPS_PER_ORBIT_EST_INNER = 1_449_933
STEPS_TOTAL = STEPS_PER_ORBIT_EST_INNER * (N_INNER_ORBITS_TARGET + EXTRA_INNER_ORBITS_RUN)

SAMPLES_MAX = 2_000_000

PLOT_DS_TARGET = 100000  # downsample for 3D/2D plots (UNCHANGED)


# ============================================================
# FLUX KERNEL
# ============================================================

def build_flux_kernel(r_max, chi_flux):
    k = np.arange(1, r_max + 1, dtype=np.float64)
    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k**2 + 2.0

    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0

    Phi = np.cumsum(counts * rho)

    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])
```

 doi:10.20944/preprints202405.2028.v8


```
    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
        w_arr[d] = s

    return w_arr, gamma_arr


@njit(fastmath=True)
def get_field(d, w_arr, gamma_arr):
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]
    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0
    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1
    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]
    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w
    return w_val, g_val


# ============================================================
# GEOMETRY + ICs
# ============================================================

def orbital_plane_basis(incl_rad):
    c = math.cos(incl_rad)
    s = math.sin(incl_rad)
    e_xp = np.array([1.0, 0.0, 0.0], dtype=np.float64)
    e_yp = np.array([0.0, c,  s], dtype=np.float64)
    e_zp = np.array([0.0, -s, c], dtype=np.float64)
    return e_xp, e_yp, e_zp


def init_three_body(w_arr, gamma_arr, v_in_factor, v_out_factor):
    pos = np.zeros((3, 3), dtype=np.float64)
    pos[0, 0] = 0.0
    pos[1, 0] = D_IN
    pos[2, 0] = -D_OUT

    Mtot = MASSES.sum()
    COM0 = (pos[0] * MASSES[0] + pos[1] * MASSES[1] + pos[2] * MASSES[2]) / Mtot
    pos -= COM0

    t_hat = np.array([0.0, math.cos(INCL_RAD), math.sin(INCL_RAD)], dtype=np.float64)

    _, g_in = get_field(D_IN, w_arr, gamma_arr)
    F_in = CHI_E * MASSES[0] * MASSES[1] * abs(g_in)
```

 doi:10.20944/preprints202405.2028.v8


```
    mu_in = (MASSES[0] * MASSES[1]) / (MASSES[0] + MASSES[1])
    v_circ_in = math.sqrt(F_in * D_IN / mu_in)
    v_rel_in = v_in_factor * v_circ_in

    _, g_out = get_field(D_OUT, w_arr, gamma_arr)
    F_out = CHI_E * MASSES[0] * MASSES[2] * abs(g_out)
    mu_out = (MASSES[0] * MASSES[2]) / (MASSES[0] + MASSES[2])
    v_circ_out = math.sqrt(F_out * D_OUT / mu_out)
    v_rel_out = v_out_factor * v_circ_out

    mE, mI, mO = MASSES[0], MASSES[1], MASSES[2]
    Mtot = mE + mI + mO
    a = -(mI * v_rel_in + mO * v_rel_out) / Mtot
    b = a + v_rel_in
    c = a + v_rel_out

    vel = np.zeros((3, 3), dtype=np.float64)
    vel[0] = a * t_hat
    vel[1] = b * t_hat
    vel[2] = c * t_hat
    return pos, vel


# ============================================================
# DYNAMICS (FR + H_DO)
# ============================================================

@njit(fastmath=True)
def forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_out, F_out):
    for i in range(3):
        F_out[i, 0] = 0.0
        F_out[i, 1] = 0.0
        F_out[i, 2] = 0.0
    U = 0.0

    for i in range(2):
        mi = masses[i]
        for j in range(i + 1, 3):
            mj = masses[j]
            rx = pos[i, 0] - pos[j, 0]
            ry = pos[i, 1] - pos[j, 1]
            rz = pos[i, 2] - pos[j, 2]

            ax = abs(rx); ay = abs(ry); az = abs(rz)
            d_inf = ax
            if ay > d_inf: d_inf = ay
            if az > d_inf: d_inf = az

            w_val, g_val = get_field(d_inf, w_arr, gamma_arr)
            U += -chi_e * mi * mj * w_val

            r1 = ax + ay + az
            if r1 > 0.0:
                ux = rx / r1; uy = ry / r1; uz = rz / r1
                f_mag = chi_e * mi * mj * g_val
```

 doi:10.20944/preprints202405.2028.v8


```
                fx = f_mag * ux; fy = f_mag * uy; fz = f_mag * uz
                F_out[i, 0] += fx; F_out[i, 1] += fy; F_out[i, 2] += fz
                F_out[j, 0] -= fx; F_out[j, 1] -= fy; F_out[j, 2] -= fz

    for i in range(3):
        inv_m = 1.0 / masses[i]
        acc_out[i, 0] = F_out[i, 0] * inv_m
        acc_out[i, 1] = F_out[i, 1] * inv_m
        acc_out[i, 2] = F_out[i, 2] * inv_m

    return U


@njit(fastmath=True)
def run_fr_3body(dt, steps, masses, chi_e, w_arr, gamma_arr, pos0, vel0, samples_max):
    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    pos = pos0.copy()
    vel = vel0.copy()

    sample_stride = steps // samples_max
    if sample_stride < 1:
        sample_stride = 1
    n_alloc = steps // sample_stride + 2

    t_hist = np.zeros(n_alloc, dtype=np.float64)
    E_hist = np.zeros(n_alloc, dtype=np.float64)
    H_hist = np.zeros(n_alloc, dtype=np.float64)
    Lx_hist = np.zeros(n_alloc, dtype=np.float64)
    Ly_hist = np.zeros(n_alloc, dtype=np.float64)
    Lz_hist = np.zeros(n_alloc, dtype=np.float64)
    pos_hist = np.zeros((n_alloc, 3, 3), dtype=np.float64)

    work_defect = 0.0

    F_s = np.zeros((3, 3), dtype=np.float64)
    F_e = np.zeros((3, 3), dtype=np.float64)
    acc_s = np.zeros((3, 3), dtype=np.float64)
    acc_e = np.zeros((3, 3), dtype=np.float64)

    U_s = forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_s, F_s)

    Mtot = masses[0] + masses[1] + masses[2]
    COM = (pos[0] * masses[0] + pos[1] * masses[1] + pos[2] * masses[2]) / Mtot

    ke = 0.5 * (
        masses[0] * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
        masses[1] * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2) +
        masses[2] * (vel[2,0]**2 + vel[2,1]**2 + vel[2,2]**2)
    )
```

 doi:10.20944/preprints202405.2028.v8


```
E_naive = ke + U_s
H_DO = E_naive - work_defect

r0 = pos[0] - COM; r1 = pos[1] - COM; r2 = pos[2] - COM
p0 = masses[0]*vel[0]; p1 = masses[1]*vel[1]; p2 = masses[2]*vel[2]
L = np.cross(r0, p0) + np.cross(r1, p1) + np.cross(r2, p2)

idx = 0
t_hist[idx] = 0.0
E_hist[idx] = E_naive
H_hist[idx] = H_DO
Lx_hist[idx] = L[0]; Ly_hist[idx] = L[1]; Lz_hist[idx] = L[2]
pos_hist[idx] = pos
idx += 1

for step in range(1, steps + 1):
    for s in range(4):
        dr0x = c[s] * dt * vel[0,0]; dr0y = c[s] * dt * vel[0,1]; dr0z = c[s] * dt * vel[0,2]
        dr1x = c[s] * dt * vel[1,0]; dr1y = c[s] * dt * vel[1,1]; dr1z = c[s] * dt * vel[1,2]
        dr2x = c[s] * dt * vel[2,0]; dr2y = c[s] * dt * vel[2,1]; dr2z = c[s] * dt * vel[2,2]

        pos[0,0] += dr0x; pos[0,1] += dr0y; pos[0,2] += dr0z
        pos[1,0] += dr1x; pos[1,1] += dr1y; pos[1,2] += dr1z
        pos[2,0] += dr2x; pos[2,1] += dr2y; pos[2,2] += dr2z

        U_e = forces_potential_inplace(pos, masses, chi_e, w_arr, gamma_arr, acc_e, F_e)

        work_total = 0.0
        for i in range(3):
            favg0 = 0.5 * (F_s[i,0] + F_e[i,0])
            favg1 = 0.5 * (F_s[i,1] + F_e[i,1])
            favg2 = 0.5 * (F_s[i,2] + F_e[i,2])
            if i == 0:
                work_total += favg0*dr0x + favg1*dr0y + favg2*dr0z
            elif i == 1:
                work_total += favg0*dr1x + favg1*dr1y + favg2*dr1z
            else:
                work_total += favg0*dr2x + favg1*dr2y + favg2*dr2z

        work_defect += (work_total + (U_e - U_s))

        if dcoef[s] != 0.0:
            vel[0,0] += dcoef[s]*dt*acc_e[0,0]; vel[0,1] += dcoef[s]*dt*acc_e[0,1]; vel[0,2] +=
            ↪ dcoef[s]*dt*acc_e[0,2]
            vel[1,0] += dcoef[s]*dt*acc_e[1,0]; vel[1,1] += dcoef[s]*dt*acc_e[1,1]; vel[1,2] +=
            ↪ dcoef[s]*dt*acc_e[1,2]
            vel[2,0] += dcoef[s]*dt*acc_e[2,0]; vel[2,1] += dcoef[s]*dt*acc_e[2,1]; vel[2,2] +=
            ↪ dcoef[s]*dt*acc_e[2,2]

        tmpF = F_s; F_s = F_e; F_e = tmpF
        tmpA = acc_s; acc_s = acc_e; acc_e = tmpA
        U_s = U_e

    if (step % sample_stride == 0) and (idx < n_alloc):
        COM = (pos[0] * masses[0] + pos[1] * masses[1] + pos[2] * masses[2]) / Mtot
```

 doi:10.20944/preprints202405.2028.v8


```
            ke = 0.5 * (
                masses[0] * (vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
                masses[1] * (vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2) +
                masses[2] * (vel[2,0]**2 + vel[2,1]**2 + vel[2,2]**2)
            )
            E_naive = ke + U_s
            H_DO = E_naive - work_defect

            r0 = pos[0] - COM; r1 = pos[1] - COM; r2 = pos[2] - COM
            p0 = masses[0]*vel[0]; p1 = masses[1]*vel[1]; p2 = masses[2]*vel[2]
            L = np.cross(r0, p0) + np.cross(r1, p1) + np.cross(r2, p2)

            t_hist[idx] = step * dt
            E_hist[idx] = E_naive
            H_hist[idx] = H_DO
            Lx_hist[idx] = L[0]; Ly_hist[idx] = L[1]; Lz_hist[idx] = L[2]
            pos_hist[idx] = pos
            idx += 1

    return idx, t_hist, E_hist, H_hist, Lx_hist, Ly_hist, Lz_hist, pos_hist


# ============================================================
# \omega helpers
# ============================================================

def quad_vertex_offset(y_m1, y0, y_p1):
    denom = (y_m1 - 2.0*y0 + y_p1)
    if denom == 0.0:
        return 0.0
    d = 0.5 * (y_m1 - y_p1) / denom
    return max(-1.0, min(1.0, d))


def quad_interp_at(d, f_m1, f0, f_p1):
    a = 0.5 * (f_p1 + f_m1 - 2.0*f0)
    b = 0.5 * (f_p1 - f_m1)
    return a*d*d + b*d + f0


def extrema_interp_with_indices(r):
    r = np.asarray(r, dtype=np.float64)
    i_min = int(np.argmin(r))
    i_max = int(np.argmax(r))
    if 0 < i_min < (len(r)-1):
        d = quad_vertex_offset(r[i_min-1], r[i_min], r[i_min+1])
        rmin = quad_interp_at(d, r[i_min-1], r[i_min], r[i_min+1])
    else:
        rmin = float(r[i_min])
    if 0 < i_max < (len(r)-1):
        d = quad_vertex_offset(r[i_max-1], r[i_max], r[i_max+1])
        rmax = quad_interp_at(d, r[i_max-1], r[i_max], r[i_max+1])
    else:
        rmax = float(r[i_max])
    e = (rmax - rmin) / (rmax + rmin) if (rmax + rmin) != 0.0 else 0.0
```

 doi:10.20944/preprints202405.2028.v8


```
    a = 0.5*(rmin + rmax)
    return float(e), float(rmin), float(rmax), float(a), i_min


def pericenter_angle_from_rmin(Rx, Ry, r, i_min):
    if 0 < i_min < (len(r)-1):
        d = quad_vertex_offset(r[i_min-1], r[i_min], r[i_min+1])
        Rxp = quad_interp_at(d, Rx[i_min-1], Rx[i_min], Rx[i_min+1])
        Ryp = quad_interp_at(d, Ry[i_min-1], Ry[i_min], Ry[i_min+1])
    else:
        Rxp = float(Rx[i_min])
        Ryp = float(Ry[i_min])
    return math.atan2(Ryp, Rxp)


def wrap_pi(theta):
    return (theta + math.pi) % (2.0*math.pi) - math.pi


def unwrap_pi(prev, raw):
    if prev is None:
        return raw
    best = raw
    best_d = abs(raw - prev)
    for k in range(-6, 7):
        cand = raw + k*math.pi
        d = abs(cand - prev)
        if d < best_d:
            best = cand
            best_d = d
    return best


def ellipse_axis_angle(x, y):
    x = np.asarray(x, dtype=np.float64)
    y = np.asarray(y, dtype=np.float64)
    if x.size < 25:
        return False, 0.0
    xm = x.mean(); ym = y.mean()
    xc = x - xm; yc = y - ym
    D = np.vstack([xc*xc, xc*yc, yc*yc, xc, yc, np.ones_like(xc)]).T
    try:
        _, _, Vt = np.linalg.svd(D, full_matrices=False)
        p = Vt[-1, :]
    except Exception:
        return False, 0.0
    A, B, C, Dx, Ey, F = p
    if (B*B - 4.0*A*C) >= 0.0:
        return False, 0.0
    theta = 0.5 * math.atan2(B, (A - C))
    return True, theta


# ============================================================
# ORBIT BOUNDARIES (inner)
```

 doi:10.20944/preprints202405.2028.v8


```
# ============================================================

def find_boundaries_signed(x, y, t, k_max):
    phi = np.unwrap(np.arctan2(y, x))
    dphi_end = float(phi[-1] - phi[0])
    sgn = 1.0 if dphi_end >= 0.0 else -1.0
    phi_s = phi * sgn
    phi0 = float(phi_s[0])
    two_pi = 2.0 * math.pi
    boundaries = [(0, 0, 0.0, t[0])]
    target_k = 1
    i = 1
    while target_k <= k_max and i < len(phi_s):
        thr = phi0 + target_k * two_pi
        if phi_s[i] >= thr:
            i_lo = i - 1
            i_hi = i
            denom = (phi_s[i_hi] - phi_s[i_lo])
            frac = 0.0 if denom == 0.0 else (thr - phi_s[i_lo]) / denom
            frac = min(1.0, max(0.0, frac))
            t_b = t[i_lo] + frac * (t[i_hi] - t[i_lo])
            boundaries.append((target_k, i_lo, frac, t_b))
            target_k += 1
        i += 1
    return boundaries, sgn


def full_run_metrics(theta_deg, r):
    th = np.unwrap(np.radians(theta_deg))
    dth = th - th[0]
    delta_deg_signed = float(math.degrees(dth[-1]))
    delta_deg_abs = abs(delta_deg_signed)
    norbits_abs = delta_deg_abs / 360.0
    sgn = 1.0 if delta_deg_signed >= 0.0 else -1.0
    return delta_deg_signed, delta_deg_abs, float(norbits_abs), sgn, float(r.mean()), float(r.std())


# ============================================================
# MAIN
# ============================================================

if __name__ == "__main__":
    e_xp, e_yp, e_zp = orbital_plane_basis(INCL_RAD)

    print("Building Candidate-A flux kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)
    pos0, vel0 = init_three_body(w_arr, gamma_arr, V_FACTOR_IN, V_FACTOR_OUT)

    print("Warming up Numba JIT...")
    _ = run_fr_3body(DT, 50_000, MASSES, CHI_E, w_arr, gamma_arr, pos0, vel0, 5000)
    print("Warm-up done.\n")

    print("Running THREE-BODY 1000-INNER-ORBIT test (flux + tilt + H_DO)")
    print(f"  dt = {DT}, tilt = {INCL_DEG} deg")
    print(f"  masses = [Earth=81.297, Inner=1.0, Outer=1.0]")
```

 doi:10.20944/preprints202405.2028.v8


```
print(f"  radii  = [Inner=30.0, Outer=40.0]")
print(f"  V_IN = {V_FACTOR_IN}, V_OUT = {V_FACTOR_OUT}")
print(f"  steps_total (incl extra) = {STEPS_TOTAL}")
print(f"  samples_max (Option B) = {SAMPLES_MAX}\n")

t0 = time.time()
n_samples, t_hist, E_hist, H_hist, Lx, Ly, Lz, pos_hist = run_fr_3body(
    DT, STEPS_TOTAL, MASSES, CHI_E, w_arr, gamma_arr, pos0, vel0, SAMPLES_MAX
)
t1 = time.time()
print(f"Simulation complete. Elapsed time: {t1 - t0:.2f} s\n")

# trim
t_hist = t_hist[:n_samples]
E_hist = E_hist[:n_samples]
H_hist = H_hist[:n_samples]
Lx = Lx[:n_samples]; Ly = Ly[:n_samples]; Lz = Lz[:n_samples]
pos_hist = pos_hist[:n_samples]

# COM and L
Mtot = MASSES.sum()
COM = (pos_hist[:,0,:]*MASSES[0] + pos_hist[:,1,:]*MASSES[1] + pos_hist[:,2,:]*MASSES[2]) / Mtot
Lmag = np.sqrt(Lx**2 + Ly**2 + Lz**2)

inc_raw = np.degrees(np.arccos(np.clip(Lz / Lmag, -1.0, 1.0)))
inc_eff = np.minimum(inc_raw, 180.0 - inc_raw)
node = np.degrees(np.arctan2(Lx, -Ly))

# inner about Earth for orbit clock
rel_in_E = pos_hist[:,1,:] - pos_hist[:,0,:]
Rx_in = rel_in_E @ e_xp
Ry_in = rel_in_E @ e_yp
Rz_in = rel_in_E @ e_zp

# outer about COM for table count
out_B = pos_hist[:,2,:] - COM
Rx_out = out_B @ e_xp
Ry_out = out_B @ e_yp
Rz_out = out_B @ e_zp

# boundaries (inner)
b_in, sgn_in = find_boundaries_signed(Rx_in, Ry_in, t_hist, N_INNER_ORBITS_TARGET + 1)
n_in = min(len(b_in) - 1, N_INNER_ORBITS_TARGET)
print(f"Found {n_in} complete INNER orbits for analysis.\n")

# horizon
_, i_lo_end, _, t_end = b_in[n_in]
end_idx = min(i_lo_end + 2, n_samples)

# invariants window
dE = (E_hist[:end_idx] - E_hist[0]) / abs(E_hist[0])
dH = (H_hist[:end_idx] - H_hist[0]) / abs(H_hist[0])
dL = (Lmag[:end_idx] - Lmag[0]) / abs(Lmag[0])

COM_vec = COM[:end_idx] - COM[0]
```

 doi:10.20944/preprints202405.2028.v8


```
COM_drift_max = float(np.linalg.norm(COM_vec, axis=1).max())
COM_drift_rms = float(np.sqrt(np.mean(np.sum(COM_vec*COM_vec, axis=1))))

r_in_series = np.sqrt(Rx_in[:end_idx]**2 + Ry_in[:end_idx]**2 + Rz_in[:end_idx]**2)
r_in_mean = float(r_in_series.mean())
COM_rel = COM_drift_max / r_in_mean if r_in_mean != 0.0 else float("inf")

print("GLOBAL INVARIANTS (first 1000 inner orbits)")
print("-------------------------------------------")
print(f"H_DO net drift        = {float(dH[-1]):.3e}")
print(f"H_DO RMS              = {float(np.std(dH)):.3e}")
print(f"Naive E net drift      = {float(dE[-1]):.3e}")
print(f"Naive E RMS            = {float(np.std(dE)):.3e}")
print(f"|L| net drift          = {float(dL[-1]):.3e}")
print(f"|L| RMS                = {float(np.std(dL)):.3e}")
print(f"COM drift max (hops)   = {COM_drift_max:.3e}")
print(f"COM drift RMS (hops)   = {COM_drift_rms:.3e}")
print(f"COM drift max / <r_in> = {COM_rel:.3e}")
print(f"Inclination mean/std   = {float(inc_eff[:end_idx].mean()):.6f} /
↪  {float(inc_eff[:end_idx].std()):.6f}")
print(f"Node start/end         = {float(node[0]):.6f} -> {float(node[:end_idx][-1]):.6f}\n")

# outer table-style count
theta_out = np.degrees(np.unwrap(np.arctan2(Ry_out[:end_idx], Rx_out[:end_idx])))
r_out = np.sqrt(Rx_out[:end_idx]**2 + Ry_out[:end_idx]**2 + Rz_out[:end_idx]**2)
dOsgn, dOabs, nO, sgnO, rOmean, rOstd = full_run_metrics(theta_out, r_out)

print("OUTER ORBIT COUNT (Table-style, same horizon)")
print("---------------------------------------------")
print(f"Outer \Delta\theta_full = {dOsgn:.6f} deg  (|\Delta\theta|={dOabs:.6f} deg)")
print(f"Outer Norbits = {nO:.6f}  (sgn={sgnO:.0f})")
print(f"Outer r_mean/std (barycentric) = {rOmean:.6f} / {rOstd:.6f}\n")

# inner step counts
inner_steps = []
for j in range(1, n_in + 1):
    _, _, _, t0b = b_in[j - 1]
    _, _, _, t1b = b_in[j]
    inner_steps.append(int(round((t1b - t0b) / DT)))
inner_steps = np.array(inner_steps, dtype=int)

print("INNER ORBIT STEP COUNT")
print("----------------------")
print(f"mean={inner_steps.mean():.3f}  std={inner_steps.std():.3f}  min={inner_steps.min()}
↪  max={inner_steps.max()}\n")

# per-orbit arrays
e_rmin = np.zeros(n_in, dtype=np.float64)
omega_deg = np.zeros(n_in, dtype=np.float64)
domega_arcsec = np.zeros(n_in, dtype=np.float64)
fit_ok = np.zeros(n_in, dtype=np.int64)

omega_prev = None

for j in range(1, n_in + 1):
```

 doi:10.20944/preprints202405.2028.v8


```
    _, i0, _, _ = b_in[j - 1]
    _, i1, _, _ = b_in[j]
    seg_start = i0
    seg_end = min(i1 + 2, end_idx)

    Rx_seg = Rx_in[seg_start:seg_end]
    Ry_seg = Ry_in[seg_start:seg_end]
    r_seg = np.sqrt(Rx_seg**2 + Ry_seg**2 + Rz_in[seg_start:seg_end]**2)

    e, rmn, rmx, ar, i_min = extrema_interp_with_indices(r_seg)
    peri_ang = pericenter_angle_from_rmin(Rx_seg, Ry_seg, r_seg, i_min)

    ok, theta = ellipse_axis_angle(Rx_seg, Ry_seg)
    if ok:
        d0 = abs(wrap_pi(theta - peri_ang))
        d1 = abs(wrap_pi((theta + math.pi) - peri_ang))
        omega_raw = theta if d0 <= d1 else (theta + math.pi)
        fit = 1
    else:
        omega_raw = peri_ang
        fit = 0

    omega_u = unwrap_pi(omega_prev, omega_raw)
    if omega_prev is None:
        dw = 0.0
    else:
        dw = math.degrees(omega_u - omega_prev) * 3600.0
    omega_prev = omega_u

    e_rmin[j-1] = e
    omega_deg[j-1] = (math.degrees(omega_u) % 360.0 + 360.0) % 360.0
    domega_arcsec[j-1] = dw
    fit_ok[j-1] = fit

print("INNER SUMMARY (1000 orbits, ellipse-axis \omega)")
print("-----------------------------------------")
print(f"e_rmin: mean={e_rmin.mean():.6f}  std={e_rmin.std():.6f}  first={e_rmin[0]:.6f}
↪  last={e_rmin[-1]:.6f}")
print(f"fit_ok: {int(fit_ok.sum())} / {len(fit_ok)}")
if n_in > 2:
    di = domega_arcsec[1:]
    print(f"\Delta\omega: mean={di.mean():.6f} arcsec/orbit  \sigma={di.std():.6f}")

# ============================================================
# ADDED PLOTS (ONLY ADDITIONS REQUESTED)
# ============================================================

# downsample indices for plotting
ds = max(1, end_idx // PLOT_DS_TARGET)
idxs = np.arange(0, end_idx, ds, dtype=int)

# (1) 3D trajectories
fig = plt.figure(figsize=(9, 7))
ax = fig.add_subplot(111, projection="3d")
ax.plot(pos_hist[idxs, 0, 0], pos_hist[idxs, 0, 1], pos_hist[idxs, 0, 2], label="Earth")
```

 doi:10.20944/preprints202405.2028.v8


```
ax.plot(pos_hist[idxs, 1, 0], pos_hist[idxs, 1, 1], pos_hist[idxs, 1, 2], label="Inner")
ax.plot(pos_hist[idxs, 2, 0], pos_hist[idxs, 2, 1], pos_hist[idxs, 2, 2], label="Outer")
ax.plot(COM[idxs, 0], COM[idxs, 1], COM[idxs, 2], label="COM")
ax.set_title("Three-body 3D trajectories (downsampled)")
ax.set_xlabel("x (hops)"); ax.set_ylabel("y (hops)"); ax.set_zlabel("z (hops)")
ax.legend(loc="upper right")
plt.tight_layout()
plt.show()

# (2) 2D orbital-plane tracks
plt.figure(figsize=(7, 7))
plt.plot(Rx_in[idxs], Ry_in[idxs], lw=0.6, label="Inner about Earth (plane)")
plt.plot(Rx_out[idxs], Ry_out[idxs], lw=0.6, label="Outer about COM (plane)")
plt.scatter([0.0], [0.0], s=20, label="Origin")
plt.title("Orbital-plane tracks (downsampled)")
plt.xlabel("x' (hops)"); plt.ylabel("y' (hops)")
plt.axis("equal")
plt.grid(True, alpha=0.3)
plt.legend(loc="upper right")
plt.tight_layout()
plt.show()

# existing time-series plots
plt.figure(figsize=(7,4))
plt.plot(np.arange(1, n_in+1), e_rmin, lw=0.8)
plt.xlabel("Inner orbit index"); plt.ylabel("e_rmin")
plt.title("Inner e_rmin per orbit (1000)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(7,4))
plt.plot(np.arange(1, n_in+1), omega_deg, lw=0.8)
plt.xlabel("Inner orbit index"); plt.ylabel("\omega (deg)")
plt.title("Inner \omega per orbit (ellipse axis, 1000)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(7,4))
plt.plot(np.arange(1, n_in+1), domega_arcsec, lw=0.8)
plt.xlabel("Inner orbit index"); plt.ylabel("\Delta\omega (arcsec/orbit)")
plt.title("Inner \Delta\omega per orbit (ellipse axis, 1000)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

Appendix B.3.5.2. Console transcript

```
Building Candidate-A flux kernel...
Warming up Numba JIT...
Warm-up done.

Running THREE-BODY 1000-INNER-ORBIT test (flux + tilt + H_DO)
  dt = 0.003, tilt = 5.145 deg
  masses = [Earth=81.297, Inner=1.0, Outer=1.0]
```

 doi:10.20944/preprints202405.2028.v8


```
  radii  = [Inner=30.0, Outer=40.0]
  V_IN = 0.939199, V_OUT = 1.0308
  steps_total (incl extra) = 1467332196
  samples_max (Option B) = 2000000

Simulation complete. Elapsed time: 613.41 s

Found 1000 complete INNER orbits for analysis.

GLOBAL INVARIANTS (first 1000 inner orbits)
-------------------------------------------
H_DO net drift        = 3.981e-12
H_DO RMS              = 3.580e-12
Naive E net drift      = -1.152e-02
Naive E RMS            = 1.404e-01
|L| net drift          = 5.104e-11
|L| RMS                = 1.883e-11
COM drift max (hops)   = 8.063e-09
COM drift RMS (hops)   = 3.651e-09
COM drift max / <r_in> = 2.883e-10
Inclination mean/std   = 5.145000 / 0.000000
Node start/end         = 180.000000 -> 180.000000

OUTER ORBIT COUNT (Table-style, same horizon)
---------------------------------------------
Outer \Delta\theta_full = -186464.059185 deg  (|\Delta\theta|=186464.059185 deg)
Outer Norbits = 517.955720  (sgn=-1)
Outer r_mean/std (barycentric) = 43.128234 / 1.607408

INNER ORBIT STEP COUNT
----------------------
mean=1460802.733  std=6324.984  min=1444198  max=1477787

INNER SUMMARY (1000 orbits, ellipse-axis \omega)
------------------------------------------------
e_rmin: mean=0.055389  std=0.015386  first=0.075799  last=0.046756
fit_ok: 1000 / 1000
\Delta\omega: mean=-75.029073 arcsec/orbit  \sigma=57718.116028
```

Appendix C.6. Appendix B.12. Validation IV1: Mercury–Sun baseline (1-year)

Cross-reference. Appendix B, Validation IV1.

Appendix B.3.6.1. Executable script

```
import numpy as np
import matplotlib.pyplot as plt
from numba import njit
import math
import time

# ============================================================
# 1-YEAR MERCURY-SUN VALIDATION (DO-NATIVE, D0 = 60)
# Candidate A flux kernel + Manhattan direction + FR + H_DO
#
# Fixed-step approach:
```

 doi:10.20944/preprints202405.2028.v8


```
#   - Use pre-calibrated STEPS_YEAR for D0=60
#   - Run exactly that many steps (one orbit)
#   - Fit ellipse and compute diagnostics on the full track
# ============================================================

# ----------------------------
# USER CONTROLS
# ----------------------------

DT          = 1.0e-4

# Discrete geometry / orbit setup
D0          = 60.0        # Chebyshev launch radius (Mercury-Sun analogue)
R_MAX       = 200         # Max Chebyshev radius (n=401 grid)

# High-mass Sun-Mercury DO parameters
M1, M2      = 1000.0, 1.0  # "Sun" and "Mercury" masses in DO units

# Gravitational coupling and flux kernel parameter
CHI_FLUX    = 24.0
CHI_E_BASE  = 0.00057032984
CHI_E_SCALE = 100.0
CHI_E       = CHI_E_BASE * CHI_E_SCALE

# Tangential velocity factor (from your D0=60 calibration)
V_FACTOR    = 1.107167

# Fixed steps per year (from your calibration)
STEPS_YEAR  = 5_797_311    # one orbit at D0=60
STEPS_TOTAL = STEPS_YEAR   # 1-year run

# Sampling: target number of samples over full orbit
TARGET_SAMPLES = 4000
# Target e (for reporting)
E_TARGET = 0.205630

# ----------------------------
# 1. Ellipse fit helper
# ----------------------------

def fit_ellipse_parameters(x, y):
    """
    Fit a general conic Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0
    and extract ellipse parameters (e, a). Returns (e,a).
    If the best-fit conic is not an ellipse (discriminant >= 0),
    returns (0.0, 0.0).
    """
    x = np.asarray(x)
    y = np.asarray(y)
    if x.size < 20:
        return 0.0, 0.0

    D = np.vstack([x**2, x*y, y**2, x, y, np.ones_like(x)]).T
    A_mat, b_vec = D[:, :-1], -np.ones(len(x))
    A, B, C, D_c, E = np.linalg.lstsq(A_mat, b_vec, rcond=None)[0]
```

 doi:10.20944/preprints202405.2028.v8


```
    F = 1.0

    delta = B**2 - 4.0 * A * C
    if delta >= 0.0:
        return 0.0, 0.0

    # Center of ellipse
    x0 = (2.0 * C * D_c - B * E) / delta
    y0 = (2.0 * A * E - B * D_c) / delta

    # Semi-axes from standard conic formulas
    num = 2.0 * (A*x0**2 + C*y0**2 + B*x0*y0 - F)
    tmp = math.sqrt((A - C)**2 + B**2)
    den1 = (A + C) + tmp
    den2 = (A + C) - tmp
    if den1 == 0.0 or den2 == 0.0:
        return 0.0, 0.0

    a = math.sqrt(abs(num / den2))
    b = math.sqrt(abs(num / den1))
    if a < b:
        a, b = b, a

    e = math.sqrt(1.0 - (b / a)**2)
    return e, a

# ----------------------------
# 2. Candidate-A flux kernel
# ----------------------------

def build_flux_kernel(r_max, chi_flux):
    """
    Build Candidate-A radial kernel:
      N_d = 24 d^2 + 2 (for d>=1), N_0 = 1
      Phi(d) = sum_{k<=d} N_k * rho_k, with rho_0=1, rho_{d>0}=0
      gamma(d) = -chi_flux * Phi(d) / N_d (for d>=1), gamma(0)=0
      w(d)-w(d+1) = -2 gamma(d+1), with w(R_max)=0
    Returns arrays w[d], gamma[d] for d=0..r_max
    """
    k = np.arange(1, r_max + 1, dtype=np.float64)
    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k * k + 2.0  # N_d

    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0
    Phi = np.cumsum(counts * rho)

    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])

    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
```

 doi:10.20944/preprints202405.2028.v8


```
        w_arr[d] = s

    return w_arr, gamma_arr

@njit(fastmath=True)
def get_field(d, w_arr, gamma_arr):
    """
    Interpolate w(d), gamma(d) for real d between 0 and r_max.
    """
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]

    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0

    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1

    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]

    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w

    return w_val, g_val

# ----------------------------
# 3. DO-native state: Manhattan direction + Candidate A
# ----------------------------

@njit(fastmath=True)
def compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr):
    """
    Given positions pos[2,3], compute:
      - accelerations on each body (acc[2,3])
      - potential energy pe
      - force magnitude fmag
      - unit direction u (graph-based, Manhattan fractions)
    """
    r_vec = pos[0] - pos[1]

    ax = abs(r_vec[0])
    ay = abs(r_vec[1])
    az = abs(r_vec[2])

    # Chebyshev radius
    d_inf = ax
    if ay > d_inf:
        d_inf = ay
    if az > d_inf:
        d_inf = az
```

 doi:10.20944/preprints202405.2028.v8


```
    w_val, g_val = get_field(d_inf, w_arr, gamma_arr)

    # Manhattan distance
    r1 = ax + ay + az

    # Graph-based direction (L1 normalized, with signs)
    u = np.zeros(3, dtype=np.float64)
    if r1 > 0.0:
        fx = ax / r1
        fy = ay / r1
        fz = az / r1

        if r_vec[0] > 0.0:
            u[0] = fx
        elif r_vec[0] < 0.0:
            u[0] = -fx

        if r_vec[1] > 0.0:
            u[1] = fy
        elif r_vec[1] < 0.0:
            u[1] = -fy

        if r_vec[2] > 0.0:
            u[2] = fz
        elif r_vec[2] < 0.0:
            u[2] = -fz

    fmag = chi_e * m1 * m2 * g_val

    acc = np.zeros((2, 3), dtype=np.float64)
    inv_m1 = 1.0 / m1
    inv_m2 = 1.0 / m2

    acc[0, 0] =  fmag * inv_m1 * u[0]
    acc[0, 1] =  fmag * inv_m1 * u[1]
    acc[0, 2] =  fmag * inv_m1 * u[2]

    acc[1, 0] = -fmag * inv_m2 * u[0]
    acc[1, 1] = -fmag * inv_m2 * u[1]
    acc[1, 2] = -fmag * inv_m2 * u[2]

    pe = -chi_e * m1 * m2 * w_val

    return acc, pe, fmag, u

# ---------------------------
# 4. FR + H_DO integrator with sampling
# ---------------------------

@njit(fastmath=True)
def run_orbit(
    dt, steps, v_factor, d0,
    m1, m2, chi_e, w_arr, gamma_arr,
    sample_stride
```

 doi:10.20944/preprints202405.2028.v8


```
):
    """
    Run DO-native 2-body FR+H_DO integrator for 'steps' major steps.
    Sample every 'sample_stride' steps and return:
      t_hist, x_hist, y_hist, r_hist, E_naive_hist, H_DO_hist, Lz_hist
    """

    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    # Initial circular speed at radius d0 for this kernel
    w0, g0 = get_field(d0, w_arr, gamma_arr)
    f_mag_init = chi_e * m1 * m2 * abs(g0)
    mu = (m1 * m2) / (m1 + m2)
    v_circ = math.sqrt(f_mag_init * d0 / mu)
    v_init = v_factor * v_circ

    # Initial positions: body 2 at +D0 on x-axis, body 1 shifted for COM at 0
    pos = np.zeros((2, 3), dtype=np.float64)
    pos[1, 0] = d0
    Mtot = m1 + m2
    comx = (pos[0, 0] * m1 + pos[1, 0] * m2) / Mtot
    pos[0, 0] -= comx
    pos[1, 0] -= comx

    # Initial velocities: tangential ±v_init, COM at rest
    vel = np.zeros((2, 3), dtype=np.float64)
    vel[1, 1] = v_init
    mass_ratio = m2 / m1
    vel[0, 1] = -mass_ratio * v_init

    # Sampling arrays
    n_samples = steps // sample_stride + 1
    t_hist   = np.zeros(n_samples, dtype=np.float64)
    x_hist   = np.zeros(n_samples, dtype=np.float64)
    y_hist   = np.zeros(n_samples, dtype=np.float64)
    r_hist   = np.zeros(n_samples, dtype=np.float64)
    E_hist   = np.zeros(n_samples, dtype=np.float64)  # naive K+U
    H_hist   = np.zeros(n_samples, dtype=np.float64)  # H_DO
    Lz_hist  = np.zeros(n_samples, dtype=np.float64)

    work_defect = 0.0

    # Initial energies
    acc_s, pe_s, fmag_s, u_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
    ke = 0.5 * (m1 * (vel[0, 0]**2 + vel[0, 1]**2 + vel[0, 2]**2)
                + m2 * (vel[1, 0]**2 + vel[1, 1]**2 + vel[1, 2]**2))
    r_vec = pos[1] - pos[0]
    r2 = r_vec[0]**2 + r_vec[1]**2 + r_vec[2]**2
    r = math.sqrt(r2)
    Lz = (pos[0, 0] * (m1 * vel[0, 1]) - pos[0, 1] * (m1 * vel[0, 0])
```

 doi:10.20944/preprints202405.2028.v8


```
      + pos[1, 0] * (m2 * vel[1, 1]) - pos[1, 1] * (m2 * vel[1, 0]))

E_naive = ke + pe_s
H_DO    = E_naive - work_defect

# First sample
t_hist[0] = 0.0
x_hist[0] = r_vec[0]
y_hist[0] = r_vec[1]
r_hist[0] = r
E_hist[0] = E_naive
H_hist[0] = H_DO
Lz_hist[0] = Lz

sample_idx = 1

# Main FR loop
for step in range(1, steps + 1):

    for i in range(4):
        # start-of-drift
        acc_s, pe_s, fmag_s, u_s = compute_state_DOnative(
            pos, m1, m2, chi_e, w_arr, gamma_arr
        )
        F_s = np.zeros(3, dtype=np.float64)
        # FIXED: use fmag_s (not f_mag_s)
        F_s[0] = fmag_s * u_s[0]
        F_s[1] = fmag_s * u_s[1]
        F_s[2] = fmag_s * u_s[2]

        # drift
        dr = c[i] * dt * vel
        pos = pos + dr

        # end-of-drift
        acc_e, pe_e, fmag_e, u_e = compute_state_DOnative(
            pos, m1, m2, chi_e, w_arr, gamma_arr
        )
        F_e = np.zeros(3, dtype=np.float64)
        F_e[0] = fmag_e * u_e[0]
        F_e[1] = fmag_e * u_e[1]
        F_e[2] = fmag_e * u_e[2]

        F_avg = 0.5 * (F_s + F_e)

        dr_rel = np.zeros(3, dtype=np.float64)
        dr_rel[0] = dr[0, 0] - dr[1, 0]
        dr_rel[1] = dr[0, 1] - dr[1, 1]
        dr_rel[2] = dr[0, 2] - dr[1, 2]

        W_mech  = F_avg[0] * dr_rel[0] + F_avg[1] * dr_rel[1] + F_avg[2] * dr_rel[2]
        delta_U = pe_e - pe_s
        work_defect += (W_mech + delta_U)

        # kick
```

 doi:10.20944/preprints202405.2028.v8


```
            if dcoef[i] != 0.0:
                vel = vel + dcoef[i] * dt * acc_e

        # sampling
        if step % sample_stride == 0:
            ke = 0.5 * (m1 * (vel[0, 0]**2 + vel[0, 1]**2 + vel[0, 2]**2)
                        + m2 * (vel[1, 0]**2 + vel[1, 1]**2 + vel[1, 2]**2))
            acc_c, pe_c, fmag_c, u_c = compute_state_DOnative(
                pos, m1, m2, chi_e, w_arr, gamma_arr
            )
            r_vec = pos[1] - pos[0]
            r2 = r_vec[0]**2 + r_vec[1]**2 + r_vec[2]**2
            r = math.sqrt(r2)
            Lz = (pos[0, 0] * (m1 * vel[0, 1]) - pos[0, 1] * (m1 * vel[0, 0])
                  + pos[1, 0] * (m2 * vel[1, 1]) - pos[1, 1] * (m2 * vel[1, 0]))
            E_naive = ke + pe_c
            H_DO    = E_naive - work_defect

            t_hist[sample_idx] = step * dt
            x_hist[sample_idx] = r_vec[0]
            y_hist[sample_idx] = r_vec[1]
            r_hist[sample_idx] = r
            E_hist[sample_idx] = E_naive
            H_hist[sample_idx] = H_DO
            Lz_hist[sample_idx] = Lz
            sample_idx += 1

    return (t_hist[:sample_idx], x_hist[:sample_idx], y_hist[:sample_idx],
            r_hist[:sample_idx], E_hist[:sample_idx], H_hist[:sample_idx],
            Lz_hist[:sample_idx])

# ---------------------------
# 5. Main: run 1-year orbit, analyze & plot
# ---------------------------

if __name__ == "__main__":
    print("Building Candidate-A kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)

    print("Warming up Numba JIT...")
    _ = run_orbit(DT, 1000, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, 10)
    print("Warm-up done.")

    # Choose sampling stride
    sample_stride = max(1, STEPS_YEAR // TARGET_SAMPLES)
    print(f"\nRunning 1-year simulation: STEPS_YEAR = {STEPS_YEAR}, sample_stride = {sample_stride}")
    t0 = time.time()
    (t_hist, x_hist, y_hist, r_hist,
     E_hist, H_hist, Lz_hist) = run_orbit(
        DT, STEPS_YEAR, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, sample_stride
    )
    t1 = time.time()
    print(f"Simulation complete. Elapsed time: {t1 - t0:.2f} s")

    # Treat entire run as one orbit
```

 doi:10.20944/preprints202405.2028.v8


```
t_year = t_hist
x_year = x_hist
y_year = y_hist
r_year = r_hist
E_year = E_hist
H_year = H_hist
Lz_year = Lz_hist

# Ellipse fit & e_rmin
e_fit, a_fit = fit_ellipse_parameters(x_year, y_year)
r_min = r_year.min()
r_max = r_year.max()
e_rmin = (r_max - r_min) / (r_max + r_min)

# Energy & Lz diagnostics over that year
E0   = E_year[0]
H0   = H_year[0]
Lz0  = Lz_year[0]

dE_naive = (E_year - E0) / abs(E0)
dH = (H_year - H0) / abs(H0)
dLz = (Lz_year - Lz0) / abs(Lz0)

print("\n1-YEAR ORBIT DIAGNOSTICS (D0 = 60, fixed-step approach):")
print(f"  V_FACTOR           = {V_FACTOR:.6f}")
print(f"  STEPS_YEAR         = {STEPS_YEAR}")
print(f"  e_fit               = {e_fit:.6f}")
print(f"  a_fit               = {a_fit:.6f}")
print(f"  e_rmin              = {e_rmin:.6f}")
print(f"  |e_fit - E_target|  = {abs(e_fit - E_TARGET):.6e}")
print(f"  H_DO net drift      = {dH[-1]:.3e}")
print(f"  H_DO RMS            = {np.std(dH):.3e}")
print(f"  Naive E net drift   = {dE_naive[-1]:.3e}")
print(f"  Naive E RMS         = {np.std(dE_naive):.3e}")
print(f"  Lz RMS              = {np.std(dLz):.3e}")

# 1) x-y orbit plot
plt.figure(figsize=(6, 6))
plt.plot(x_year, y_year, lw=0.8)
plt.scatter([0.0], [0.0], c='r', label='Sun')
plt.xlabel("x")
plt.ylabel("y")
plt.title(f"1-year DO-native Mercury-Sun orbit (D0={D0}, e\approx{e_fit:.3f})")
plt.axis('equal')
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

# 2) r(t) plot
plt.figure(figsize=(7, 4))
plt.plot(t_year, r_year, lw=0.8)
plt.xlabel("t")
plt.ylabel("r")
plt.title("Radial distance r(t) over 1 DO-year (D0=60)")
```

 doi:10.20944/preprints202405.2028.v8


```
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # 3) energy diagnostics
    plt.figure(figsize=(7, 4))
    plt.plot(t_year, dH, label="\DeltaH_DO / H0")
    plt.plot(t_year, dE_naive, label="\DeltaE_naive / E0", alpha=0.7)
    plt.xlabel("t")
    plt.ylabel("\DeltaE / E0")
    plt.title("Energy diagnostics over 1 DO-year (D0=60)")
    plt.legend()
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # 4) Lz diagnostics
    plt.figure(figsize=(7, 4))
    plt.plot(t_year, dLz, lw=0.8)
    plt.xlabel("t")
    plt.ylabel("\DeltaLz / Lz0")
    plt.title("Angular momentum diagnostics over 1 DO-year (D0=60)")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()
```

Appendix B.3.6.2. Console Transcript

```
Building Candidate-A kernel...
Warming up Numba JIT...
Warm-up done.

Running 1-year simulation: STEPS_YEAR = 5797311, sample_stride = 1449
Simulation complete. Elapsed time: 20.60 s

1-YEAR ORBIT DIAGNOSTICS (D0 = 60, fixed-step approach):
  V_FACTOR           = 1.107167
  STEPS_YEAR         = 5797311
  e_fit               = 0.205634
  a_fit               = 78.217892
  e_rmin              = 0.228116
  |e_fit - E_target|  = 4.291054e-06
  H_DO net drift      = 4.526e-13
  H_DO RMS            = 1.744e-13
  Naive E net drift   = 7.370e-04
  Naive E RMS         = 3.011e-01
  Lz RMS              = 1.140e-13
```

Appendix C.7. Validation IV2: Mercury–Sun Long Horizon (1000 Orbits)

Cross-reference. Appendix B, Validation IV2.

Appendix B.3.7.1. Executable Script

```
import numpy as np
import math
```

 doi:10.20944/preprints202405.2028.v8


```
import time
from numba import njit
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D  # noqa: F401

# ============================================================
# MERCURY-SUN 1000-ORBIT OVERNIGHT RUN (DO-NATIVE)
# Candidate-A flux kernel + Manhattan (L1) direction + Forest-Ruth
#
# PURPOSE:
#   Repeat the 1000-orbit run with:
#     - Correct stationarity reporting (blockwise \Delta\omega excluding the first orbit in each block)
#     - No per-orbit table spam (print first/last 10 only)
#     - Full diagnostic suite + 2D/3D plots (three-body style)
#
# PRIMARY CONSERVED QUANTITY:
#   I_DO = (K + U) - W_closure   (Appendix B)
#
# ============================================================

# ---------------------------
# TUNABLE PARAMETERS (all here)
# ---------------------------

DT                = 1.0e-4
D0                = 60.0
R_MAX             = 200

M1, M2            = 1000.0, 1.0

CHI_FLUX          = 24.0
CHI_E_BASE        = 0.00057032984
CHI_E_SCALE       = 100.0
CHI_E             = CHI_E_BASE * CHI_E_SCALE

V_FACTOR          = 1.107167

STEPS_PER_ORBIT   = 5_797_311
N_ORBITS_TARGET   = 1000
EXTRA_ORBITS_RUN  = 1
STEPS_TOTAL       = STEPS_PER_ORBIT * (N_ORBITS_TARGET + EXTRA_ORBITS_RUN)

SAMPLES_PER_ORBIT = 4000
SAMPLES_MAX       = SAMPLES_PER_ORBIT * (N_ORBITS_TARGET + EXTRA_ORBITS_RUN)

PLOT_DS_TARGET    = 120000
BLOCK_SIZE        = 100

PRINT_FIRST_N      = 10
PRINT_LAST_N       = 10

# ============================================================
# Candidate-A kernel
# ============================================================
```

 doi:10.20944/preprints202405.2028.v8


```
def build_flux_kernel(r_max, chi_flux):
    k = np.arange(1, r_max + 1, dtype=np.float64)
    counts = np.zeros(r_max + 1, dtype=np.float64)
    counts[0] = 1.0
    counts[1:] = 24.0 * k**2 + 2.0
    rho = np.zeros(r_max + 1, dtype=np.float64)
    rho[0] = 1.0
    Phi = np.cumsum(counts * rho)
    gamma_arr = np.zeros(r_max + 1, dtype=np.float64)
    gamma_arr[1:] = -chi_flux * (Phi[1:] / counts[1:])
    w_arr = np.zeros(r_max + 1, dtype=np.float64)
    s = 0.0
    for d in range(r_max - 1, -1, -1):
        s += -2.0 * gamma_arr[d + 1]
        w_arr[d] = s
    return w_arr, gamma_arr

@njit(fastmath=True)
def get_field(d, w_arr, gamma_arr):
    if d <= 0.0:
        return w_arr[0], gamma_arr[0]
    n_max = w_arr.shape[0] - 1
    if d >= n_max:
        return 0.0, 0.0
    idx = int(d)
    if idx >= n_max:
        idx = n_max - 1
    frac = d - idx
    g0 = gamma_arr[idx]
    g1 = gamma_arr[idx + 1]
    g_val = (1.0 - frac) * g0 + frac * g1
    delta_w = 2.0 * (g0 * frac + 0.5 * (g1 - g0) * frac * frac)
    w_val = w_arr[idx] + delta_w
    return w_val, g_val

# ============================================================
# Operator: Manhattan direction
# ============================================================

@njit(fastmath=True)
def compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr):
    r_vec = pos[0] - pos[1]
    ax = abs(r_vec[0]); ay = abs(r_vec[1]); az = abs(r_vec[2])
    d_inf = ax
    if ay > d_inf: d_inf = ay
    if az > d_inf: d_inf = az
    w_val, g_val = get_field(d_inf, w_arr, gamma_arr)

    r1 = ax + ay + az
    u = np.zeros(3, dtype=np.float64)
    if r1 > 0.0:
        u[0] = r_vec[0] / r1
        u[1] = r_vec[1] / r1
        u[2] = r_vec[2] / r1
```

 doi:10.20944/preprints202405.2028.v8


```
    f_mag = chi_e * m1 * m2 * g_val
    acc = np.zeros((2, 3), dtype=np.float64)
    inv_m1 = 1.0 / m1
    inv_m2 = 1.0 / m2
    acc[0, 0] =  f_mag * inv_m1 * u[0]
    acc[0, 1] =  f_mag * inv_m1 * u[1]
    acc[0, 2] =  f_mag * inv_m1 * u[2]
    acc[1, 0] = -f_mag * inv_m2 * u[0]
    acc[1, 1] = -f_mag * inv_m2 * u[1]
    acc[1, 2] = -f_mag * inv_m2 * u[2]

    U = -chi_e * m1 * m2 * w_val
    return acc, U, f_mag, u

# ============================================================
# Forest-Ruth + I_DO integrator with sampling
# ============================================================

@njit(fastmath=True)
def run_multi_orbit(
    dt, steps, v_factor, d0,
    m1, m2, chi_e, w_arr, gamma_arr,
    samples_max
):
    theta_c = 1.0 / (2.0 - 2.0**(1.0 / 3.0))
    c = np.array([theta_c / 2.0,
                  (1.0 - theta_c) / 2.0,
                  (1.0 - theta_c) / 2.0,
                  theta_c / 2.0], dtype=np.float64)
    dcoef = np.array([theta_c, 1.0 - 2.0 * theta_c, theta_c, 0.0], dtype=np.float64)

    # v_init = v_factor * v_circ(d0)
    w0, g0 = get_field(d0, w_arr, gamma_arr)
    f_mag_init = chi_e * m1 * m2 * abs(g0)
    mu = (m1 * m2) / (m1 + m2)
    v_circ = math.sqrt(f_mag_init * d0 / mu)
    v_init = v_factor * v_circ

    # initial positions (Mercury at +d0 on x), COM to origin
    pos = np.zeros((2, 3), dtype=np.float64)
    pos[1, 0] = d0
    Mtot = m1 + m2
    comx = (pos[0, 0] * m1 + pos[1, 0] * m2) / Mtot
    pos[0, 0] -= comx
    pos[1, 0] -= comx

    # initial velocities (tangential in y), COM at rest
    vel = np.zeros((2, 3), dtype=np.float64)
    vel[1, 1] = v_init
    vel[0, 1] = -(m2 / m1) * v_init

    # sampling
    sample_stride = steps // samples_max
    if sample_stride < 1:
        sample_stride = 1
```

 doi:10.20944/preprints202405.2028.v8


```
n_alloc = steps // sample_stride + 2

t_hist   = np.zeros(n_alloc, dtype=np.float64)
E_hist   = np.zeros(n_alloc, dtype=np.float64)  # naive
I_hist   = np.zeros(n_alloc, dtype=np.float64)  # I_DO
Lz_hist  = np.zeros(n_alloc, dtype=np.float64)
r_hist   = np.zeros(n_alloc, dtype=np.float64)
COM_hist = np.zeros((n_alloc, 3), dtype=np.float64)
pos1_hist = np.zeros((n_alloc, 3), dtype=np.float64)
pos2_hist = np.zeros((n_alloc, 3), dtype=np.float64)

W_closure = 0.0

# initial sample
acc_s, U_s, fmag_s, u_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
ke = 0.5 * (m1*(vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
            m2*(vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2))
E_naive = ke + U_s
I_DO = E_naive - W_closure

COM = (pos[0]*m1 + pos[1]*m2) / Mtot
r0 = pos[0] - COM
r1 = pos[1] - COM
Lz = (r0[0]*(m1*vel[0,1]) - r0[1]*(m1*vel[0,0]) +
      r1[0]*(m2*vel[1,1]) - r1[1]*(m2*vel[1,0]))

rv = pos[1] - pos[0]
r = math.sqrt(rv[0]**2 + rv[1]**2 + rv[2]**2)

idx = 0
t_hist[idx] = 0.0
E_hist[idx] = E_naive
I_hist[idx] = I_DO
Lz_hist[idx] = Lz
r_hist[idx] = r
COM_hist[idx, :] = COM
pos1_hist[idx, :] = pos[0, :]
pos2_hist[idx, :] = pos[1, :]
idx += 1

for step in range(1, steps + 1):
    for j in range(4):
        acc_s, U_s, fmag_s, u_s = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
        F_sx = fmag_s*u_s[0]; F_sy = fmag_s*u_s[1]; F_sz = fmag_s*u_s[2]

        dr = c[j] * dt * vel
        pos = pos + dr

        acc_e, U_e, fmag_e, u_e = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
        F_ex = fmag_e*u_e[0]; F_ey = fmag_e*u_e[1]; F_ez = fmag_e*u_e[2]

        F_avgx = 0.5*(F_sx + F_ex)
        F_avgy = 0.5*(F_sy + F_ey)
        F_avgz = 0.5*(F_sz + F_ez)
```

 doi:10.20944/preprints202405.2028.v8


```
            dr_relx = dr[0,0] - dr[1,0]
            dr_rely = dr[0,1] - dr[1,1]
            dr_relz = dr[0,2] - dr[1,2]

            W_mech = F_avgx*dr_relx + F_avgy*dr_rely + F_avgz*dr_relz
            W_closure += (W_mech + (U_e - U_s))

            if dcoef[j] != 0.0:
                vel = vel + dcoef[j] * dt * acc_e

        if (step % sample_stride == 0) and (idx < n_alloc):
            acc_c, U_c, fmag_c, u_c = compute_state_DOnative(pos, m1, m2, chi_e, w_arr, gamma_arr)
            ke = 0.5 * (m1*(vel[0,0]**2 + vel[0,1]**2 + vel[0,2]**2) +
                        m2*(vel[1,0]**2 + vel[1,1]**2 + vel[1,2]**2))
            E_naive = ke + U_c
            I_DO = E_naive - W_closure

            COM = (pos[0]*m1 + pos[1]*m2) / Mtot
            r0 = pos[0] - COM
            r1 = pos[1] - COM
            Lz = (r0[0]*(m1*vel[0,1]) - r0[1]*(m1*vel[0,0]) +
                  r1[0]*(m2*vel[1,1]) - r1[1]*(m2*vel[1,0]))

            rv = pos[1] - pos[0]
            r = math.sqrt(rv[0]**2 + rv[1]**2 + rv[2]**2)

            t_hist[idx] = step*dt
            E_hist[idx] = E_naive
            I_hist[idx] = I_DO
            Lz_hist[idx] = Lz
            r_hist[idx] = r
            COM_hist[idx, :] = COM
            pos1_hist[idx, :] = pos[0, :]
            pos2_hist[idx, :] = pos[1, :]
            idx += 1

    return idx, t_hist, E_hist, I_hist, Lz_hist, r_hist, COM_hist, pos1_hist, pos2_hist

# ============================================================
# Diagnostics helpers
# ============================================================

def quad_vertex_offset(y_m1, y0, y_p1):
    denom = (y_m1 - 2.0*y0 + y_p1)
    if denom == 0.0:
        return 0.0
    d = 0.5*(y_m1 - y_p1) / denom
    return max(-1.0, min(1.0, d))

def quad_interp_at(d, f_m1, f0, f_p1):
    a = 0.5*(f_p1 + f_m1 - 2.0*f0)
    b = 0.5*(f_p1 - f_m1)
    return a*d*d + b*d + f0

def find_boundaries_xy(Rx, Ry, t, k_max):
```

 doi:10.20944/preprints202405.2028.v8


```
    phi = np.unwrap(np.arctan2(Ry, Rx))
    phi0 = float(phi[0])
    two_pi = 2.0*math.pi
    boundaries = [(0, 0, 0.0, float(t[0]))]
    target_k = 1
    i = 1
    while target_k <= k_max and i < len(phi):
        thr = phi0 + target_k*two_pi
        if phi[i] >= thr:
            i_lo = i-1; i_hi = i
            denom = (phi[i_hi] - phi[i_lo])
            frac = 0.0 if denom == 0.0 else (thr - phi[i_lo]) / denom
            frac = min(1.0, max(0.0, frac))
            t_b = float(t[i_lo] + frac*(t[i_hi] - t[i_lo]))
            boundaries.append((target_k, i_lo, frac, t_b))
            target_k += 1
        i += 1
    return boundaries

def wrap_pm_pi(x):
    return (x + math.pi) % (2.0*math.pi) - math.pi

# ============================================================
# MAIN
# ============================================================

if __name__ == "__main__":
    print("Building Candidate-A flux kernel...")
    w_arr, gamma_arr = build_flux_kernel(R_MAX, CHI_FLUX)

    print("Warming up Numba JIT...")
    _ = run_multi_orbit(DT, 200_000, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, 50_000)
    print("Warm-up done.\n")

    print("Running MERCURY-SUN 1000-ORBIT test (overnight, compact printout)")
    print(f"  dt={DT}  D0={D0}  V_FACTOR={V_FACTOR}")
    print(f"  steps_total={STEPS_TOTAL}  samples_max={SAMPLES_MAX}\n")

    t0 = time.time()
    n_samples, t_hist, E_hist, I_hist, Lz_hist, r_hist, COM_hist, pos1_hist, pos2_hist = run_multi_orbit(
        DT, STEPS_TOTAL, V_FACTOR, D0, M1, M2, CHI_E, w_arr, gamma_arr, SAMPLES_MAX
    )
    t1 = time.time()
    print(f"Simulation complete. Elapsed time: {t1 - t0:.2f} s\n")

    # trim
    t_hist = t_hist[:n_samples]
    E_hist = E_hist[:n_samples]
    I_hist = I_hist[:n_samples]
    Lz_hist = Lz_hist[:n_samples]
    r_hist = r_hist[:n_samples]
    COM_hist = COM_hist[:n_samples,:]
    pos1_hist = pos1_hist[:n_samples,:]
    pos2_hist = pos2_hist[:n_samples,:]
```

 doi:10.20944/preprints202405.2028.v8


```
rel = pos2_hist - pos1_hist
Rx = rel[:,0]; Ry = rel[:,1]; Rz = rel[:,2]

boundaries = find_boundaries_xy(Rx, Ry, t_hist, N_ORBITS_TARGET + 1)
n_orb = min(len(boundaries)-1, N_ORBITS_TARGET)
print(f"Found {n_orb} complete orbits for analysis.\n")

_, i_lo_end, _, _ = boundaries[n_orb]
end_idx = min(i_lo_end + 2, n_samples)

# GLOBAL INVARIANTS
I0 = I_hist[0]; E0 = E_hist[0]; L0 = Lz_hist[0]
dI = (I_hist[:end_idx] - I0) / abs(I0)
dE = (E_hist[:end_idx] - E0) / abs(E0)
dLz = (Lz_hist[:end_idx] - L0) / abs(L0)

COM_vec = COM_hist[:end_idx,:] - COM_hist[0,:]
COM_mag = np.linalg.norm(COM_vec, axis=1)

r_mean = float(r_hist[:end_idx].mean())
r_std  = float(r_hist[:end_idx].std())

print("GLOBAL INVARIANTS (first 1000 orbits)")
print("-------------------------------------")
print(f"I_DO net drift        = {float(dI[-1]):.3e}")
print(f"I_DO RMS              = {float(np.std(dI)):.3e}")
print(f"Naive E net drift      = {float(dE[-1]):.3e}")
print(f"Naive E RMS            = {float(np.std(dE)):.3e}")
print(f"Lz net drift           = {float(dLz[-1]):.3e}")
print(f"Lz RMS                 = {float(np.std(dLz)):.3e}")
print(f"COM drift max (hops)   = {float(COM_mag.max()):.3e}")
print(f"COM drift RMS (hops)   = {float(np.sqrt(np.mean(COM_mag**2))):.3e}")
print(f"COM drift max / <r>    = {float(COM_mag.max()/r_mean):.3e}\n")

# ORBIT COUNT
theta_deg = np.degrees(np.unwrap(np.arctan2(Ry[:end_idx], Rx[:end_idx])))
dtheta = float(theta_deg[-1] - theta_deg[0])
print("ORBIT COUNT (Table-style)")
print("-------------------------")
print(f"\Delta\theta_full = {dtheta:.6f} deg   Norbits = {abs(dtheta)/360.0:.6f}\n")

# STEP COUNT
step_counts = np.zeros(n_orb, dtype=int)
for j in range(1, n_orb+1):
    _, _, _, t0b = boundaries[j-1]
    _, _, _, t1b = boundaries[j]
    step_counts[j-1] = int(round((t1b - t0b)/DT))
print("ORBIT STEP COUNT")
print("----------------")
print(f"mean={step_counts.mean():.3f}  std={step_counts.std():.3f}  min={step_counts.min()}
↪  max={step_counts.max()}\n")

# PER-ORBIT arrays (no full print)
e_r = np.zeros(n_orb, dtype=np.float64)
a_r = np.zeros(n_orb, dtype=np.float64)
```

 doi:10.20944/preprints202405.2028.v8


```
domega_c = np.zeros(n_orb, dtype=np.float64)
domega_r = np.zeros(n_orb, dtype=np.float64)
omega_c = np.zeros(n_orb, dtype=np.float64)
omega_r = np.zeros(n_orb, dtype=np.float64)

# Store a compact per-orbit record for first/last printing
per_orbit_rows = []

for j in range(1, n_orb+1):
    _, i0, _, t0b = boundaries[j-1]
    _, i1, _, t1b = boundaries[j]
    seg_start = i0
    seg_end = min(i1+2, end_idx)
    r_seg = r_hist[seg_start:seg_end]
    Rx_seg = Rx[seg_start:seg_end]
    Ry_seg = Ry[seg_start:seg_end]

    i_min = int(np.argmin(r_seg))
    i_max = int(np.argmax(r_seg))

    # refined turning points and refined perihelion position
    if 0 < i_min < (len(r_seg)-1):
        dmin = quad_vertex_offset(r_seg[i_min-1], r_seg[i_min], r_seg[i_min+1])
        rmin = quad_interp_at(dmin, r_seg[i_min-1], r_seg[i_min], r_seg[i_min+1])
        Rxmin = quad_interp_at(dmin, Rx_seg[i_min-1], Rx_seg[i_min], Rx_seg[i_min+1])
        Rymin = quad_interp_at(dmin, Ry_seg[i_min-1], Ry_seg[i_min], Ry_seg[i_min+1])
    else:
        rmin = float(r_seg[i_min]); Rxmin = float(Rx_seg[i_min]); Rymin = float(Ry_seg[i_min])

    if 0 < i_max < (len(r_seg)-1):
        dmax = quad_vertex_offset(r_seg[i_max-1], r_seg[i_max], r_seg[i_max+1])
        rmax = quad_interp_at(dmax, r_seg[i_max-1], r_seg[i_max], r_seg[i_max+1])
    else:
        rmax = float(r_seg[i_max])

    e = (rmax - rmin) / (rmax + rmin) if (rmax + rmin) != 0.0 else 0.0
    ar = 0.5*(rmax + rmin)

    # omega coarse (index) and refined
    oc = math.atan2(float(Ry_seg[i_min]), float(Rx_seg[i_min]))
    orf = math.atan2(float(Rymin), float(Rxmin))

    omega_c[j-1] = oc
    omega_r[j-1] = orf
    e_r[j-1] = e
    a_r[j-1] = ar

    if j == 1:
        domega_c[j-1] = 0.0
        domega_r[j-1] = 0.0
    else:
        dc = wrap_pm_pi(omega_c[j-1] - omega_c[j-2])
        dr = wrap_pm_pi(omega_r[j-1] - omega_r[j-2])
        domega_c[j-1] = dc * 180.0/math.pi * 3600.0
        domega_r[j-1] = dr * 180.0/math.pi * 3600.0
```

 doi:10.20944/preprints202405.2028.v8


```
    per_orbit_rows.append((j, t0b, t1b, e, rmin, rmax, ar, step_counts[j-1],
                           oc*180.0/math.pi, domega_c[j-1],
                           orf*180.0/math.pi, domega_r[j-1]))

# Print first/last N orbits only
print("PER-ORBIT SNAPSHOT (first and last orbits only)")
print("-----------------------------------------------")
hdr = (f"{'orb':>4} {'t_start':>10} {'t_end':>10} "
       f"{'e_rmin':>9} {'r_min':>9} {'r_max':>9} {'a_r':>9} {'Nsteps':>9} "
       f"{'\omegar_deg':>9} {'\Delta\omegar':>10}")
print(hdr)
for row in per_orbit_rows[:PRINT_FIRST_N]:
    j, t0b, t1b, e, rmin, rmax, ar, ns, _, _, ordeg, dor = row
    print(f"{j:4d} {t0b:10.4f} {t1b:10.4f} {e:9.5f} {rmin:9.4f} {rmax:9.4f} {ar:9.4f} {ns:9d}
    ↪  {ordeg:9.6f} {dor:10.3f}")
print("   ...")
for row in per_orbit_rows[-PRINT_LAST_N:]:
    j, t0b, t1b, e, rmin, rmax, ar, ns, _, _, ordeg, dor = row
    print(f"{j:4d} {t0b:10.4f} {t1b:10.4f} {e:9.5f} {rmin:9.4f} {rmax:9.4f} {ar:9.4f} {ns:9d}
    ↪  {ordeg:9.6f} {dor:10.3f}")

# Precession stats (exclude first orbit)
dc = domega_c[1:]
dr = domega_r[1:]
print("\nPRECESSION STATISTICS (exclude first orbit):")
print(f"  COARSE : <\Delta\omega> = {dc.mean():.6f} arcsec/orbit   \sigma = {dc.std():.6f}")
print(f"  REFINED: <\Delta\omega> = {dr.mean():.6f} arcsec/orbit   \sigma = {dr.std():.6f}\n")

# Correct stationarity blocks: exclude the first orbit in EACH block for \Delta\omega
print("STATIONARITY CHECKS (100-orbit blocks, refined \Delta\omega)")
print("-----------------------------------------------------")
n_blocks = n_orb // BLOCK_SIZE
for b in range(n_blocks):
    a = b*BLOCK_SIZE
    z = (b+1)*BLOCK_SIZE
    e_blk = e_r[a:z]
    ar_blk = a_r[a:z]
    # \Delta\omega for orbits (a+1 ... z-1) inside block, mapped to domega_r indices
    # domega_r[j] is \Delta\omega for orbit (j+1) in 1-based indexing.
    # So for block a..z-1 (0-based), exclude a (first orbit of block):
    dr_blk = domega_r[a+1:z]
    print(f"Block {b+1:2d} orbits {a+1:4d}-{z:4d}: "
          f"e_rmin mean/std = {e_blk.mean():.6f}/{e_blk.std():.6f}   "
          f"a_r mean/std = {ar_blk.mean():.6f}/{ar_blk.std():.6f}   "
          f"\Delta\omega_ref mean/std = {dr_blk.mean():.6f}/{dr_blk.std():.6f}")

# ============================================================
# PLOTS (2D + 3D)
# ============================================================

ds = max(1, end_idx // PLOT_DS_TARGET)
idxs = np.arange(0, end_idx, ds, dtype=int)

fig = plt.figure(figsize=(9, 7))
```

 doi:10.20944/preprints202405.2028.v8


```
ax = fig.add_subplot(111, projection="3d")
ax.plot(pos1_hist[idxs,0], pos1_hist[idxs,1], pos1_hist[idxs,2], label="Sun")
ax.plot(pos2_hist[idxs,0], pos2_hist[idxs,1], pos2_hist[idxs,2], label="Mercury")
ax.plot(COM_hist[idxs,0], COM_hist[idxs,1], COM_hist[idxs,2], label="COM")
ax.set_title("Mercury-Sun 3D trajectories (downsampled)")
ax.set_xlabel("x (hops)"); ax.set_ylabel("y (hops)"); ax.set_zlabel("z (hops)")
ax.legend(loc="upper right")
plt.tight_layout()
plt.show()

plt.figure(figsize=(7, 7))
plt.plot(Rx[:end_idx:ds], Ry[:end_idx:ds], lw=0.6, label="Mercury about Sun (x-y)")
plt.scatter([0.0], [0.0], s=20, label="Origin")
plt.title("Orbital-plane rosette (downsampled)")
plt.xlabel("x (hops)"); plt.ylabel("y (hops)")
plt.axis("equal")
plt.grid(True, alpha=0.3)
plt.legend(loc="upper right")
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,4))
plt.plot(t_hist[:end_idx], r_hist[:end_idx], lw=0.45)
plt.xlabel("t"); plt.ylabel("r")
plt.title("r(t) over 1000 orbits")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,4))
plt.plot(t_hist[:end_idx], dI, lw=0.6, label="\DeltaI_DO/I0")
plt.plot(t_hist[:end_idx], dE, lw=0.6, label="\DeltaE_naive/E0", alpha=0.7)
plt.xlabel("t"); plt.ylabel("\Delta/initial")
plt.title("Energy diagnostics")
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,4))
plt.plot(t_hist[:end_idx], dLz, lw=0.6)
plt.xlabel("t"); plt.ylabel("\DeltaLz/Lz0")
plt.title("Lz drift")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,4))
plt.plot(t_hist[:end_idx], COM_mag, lw=0.6)
plt.xlabel("t"); plt.ylabel("|COM(t)-COM(0)|")
plt.title("COM drift magnitude")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

 doi:10.20944/preprints202405.2028.v8


```
    plt.figure(figsize=(8,4))
    plt.plot(np.arange(1, n_orb+1), e_r, lw=0.7)
    plt.xlabel("orbit"); plt.ylabel("e_rmin")
    plt.title("e_rmin vs orbit")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(8,4))
    plt.plot(np.arange(1, n_orb+1), a_r, lw=0.7)
    plt.xlabel("orbit"); plt.ylabel("a_r")
    plt.title("a_r vs orbit")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(8,4))
    plt.plot(np.arange(1, n_orb+1), domega_r, lw=0.7)
    plt.xlabel("orbit"); plt.ylabel("\Delta\omega_refined (arcsec/orbit)")
    plt.title("\Delta\omega_refined vs orbit")
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()
```

Appendix B.3.7.2. Console transcript

```
Building Candidate-A flux kernel...
Warming up Numba JIT...
Warm-up done.

Running MERCURY-SUN 1000-ORBIT test (overnight, compact printout)
  dt=0.0001  D0=60.0  V_FACTOR=1.107167
  steps_total=5803108311  samples_max=4004000

Simulation complete. Elapsed time: 15155.18 s

Found 1000 complete orbits for analysis.

GLOBAL INVARIANTS (first 1000 orbits)
-------------------------------------
I_DO net drift        = 1.862e-11
I_DO RMS              = 1.427e-11
Naive E net drift      = 4.189e-02
Naive E RMS            = 3.009e-01
Lz net drift           = -5.492e-12
Lz RMS                 = 3.072e-12
COM drift max (hops)   = 5.011e-10
COM drift RMS (hops)   = 3.524e-10
COM drift max / <r>    = 6.227e-12

ORBIT COUNT (Table-style)
-------------------------
\Delta\theta_full = 360000.061264 deg   Norbits = 1000.000170

ORBIT STEP COUNT
----------------
```

 doi:10.20944/preprints202405.2028.v8


```
mean=5795321.239  std=797.244  min=5793692  max=5796861

PER-ORBIT SNAPSHOT (first and last orbits only)
-----------------------------------------------
 orb    t_start      t_end    e_rmin     r_min     r_max       a_r    Nsteps    \omegar_deg       \Delta\omegar
   1     0.0000   579.6837   0.22812   60.0000   95.4639   77.7319   5796837  0.000000      0.000
   2   579.6837  1159.3673   0.22812   60.0000   95.4638   77.7319   5796835  0.031532    113.514
   3  1159.3673  1739.0503   0.22812   60.0001   95.4637   77.7319   5796830  0.063195    113.987
   4  1739.0503  2318.7327   0.22811   60.0001   95.4635   77.7318   5796824  0.094733    113.539
   5  2318.7327  2898.4141   0.22811   60.0002   95.4632   77.7317   5796814  0.126098    112.911
   6  2898.4141  3478.0943   0.22811   60.0004   95.4628   77.7316   5796803  0.157498    113.040
   7  3478.0943  4057.7732   0.22811   60.0005   95.4624   77.7315   5796789  0.188927    113.147
   8  4057.7732  4637.4504   0.22810   60.0007   95.4619   77.7313   5796772  0.220275    112.852
   9  4637.4504  5217.1257   0.22810   60.0009   95.4614   77.7311   5796753  0.251370    111.943
  10  5217.1257  5796.7990   0.22809   60.0011   95.4608   77.7309   5796732  0.282520    112.140
   ...
 991 573737.7874 574317.2146   0.23399   60.5634   97.5629   79.0632   5794272 24.368767     96.184
 992 574317.2146 574896.6425   0.23399   60.5646   97.5652   79.0649   5794279 24.395501     96.243
 993 574896.6425 575476.0714   0.23399   60.5658   97.5675   79.0667   5794289 24.422252     96.304
 994 575476.0714 576055.5016   0.23399   60.5669   97.5699   79.0684   5794302 24.449021     96.366
 995 576055.5016 576634.9334   0.23400   60.5680   97.5724   79.0702   5794317 24.475807     96.430
 996 576634.9334 577214.3668   0.23400   60.5691   97.5750   79.0720   5794335 24.502611     96.494
 997 577214.3668 577793.8023   0.23401   60.5701   97.5776   79.0739   5794355 24.529432     96.555
 998 577793.8023 578373.2400   0.23401   60.5711   97.5803   79.0757   5794377 24.556269     96.614
 999 578373.2400 578952.6803   0.23402   60.5721   97.5830   79.0776   5794402 24.583121     96.667
1000 578952.6803 579532.1233   0.23402   60.5731   97.5858   79.0795   5794430 24.609986     96.713

PRECESSION STATISTICS (exclude first orbit):
  COARSE : <\Delta\omega> = 88.612524 arcsec/orbit   \sigma = 226.768353
  REFINED: <\Delta\omega> = 88.684633 arcsec/orbit   \sigma = 7.406472

STATIONARITY CHECKS (100-orbit blocks, refined \Delta\omega)
-----------------------------------------------------
Block  1 orbits    1- 100: e_rmin mean/std = 0.227894/0.000121   a_r mean/std = 77.724226/0.004485
↪  \Delta\omega_ref mean/std = 104.422207/5.306753
Block  2 orbits  101- 200: e_rmin mean/std = 0.227808/0.000114   a_r mean/std = 77.728889/0.006699
↪  \Delta\omega_ref mean/std = 92.780683/2.940062
Block  3 orbits  201- 300: e_rmin mean/std = 0.227921/0.000159   a_r mean/std = 77.763056/0.018161
↪  \Delta\omega_ref mean/std = 86.295362/0.947914
Block  4 orbits  301- 400: e_rmin mean/std = 0.228490/0.000321   a_r mean/std = 77.854610/0.038927
↪  \Delta\omega_ref mean/std = 86.815440/0.584096
Block  5 orbits  401- 500: e_rmin mean/std = 0.229636/0.000422   a_r mean/std = 78.015363/0.055667
↪  \Delta\omega_ref mean/std = 87.920764/1.166196
Block  6 orbits  501- 600: e_rmin mean/std = 0.231171/0.000552   a_r mean/std = 78.227405/0.068534
↪  \Delta\omega_ref mean/std = 88.397307/1.200356
Block  7 orbits  601- 700: e_rmin mean/std = 0.232344/0.000158   a_r mean/std = 78.432579/0.049872
↪  \Delta\omega_ref mean/std = 76.481439/3.096826
Block  8 orbits  701- 800: e_rmin mean/std = 0.233027/0.000295   a_r mean/std = 78.618817/0.061256
↪  \Delta\omega_ref mean/std = 85.398176/4.545903
Block  9 orbits  801- 900: e_rmin mean/std = 0.234075/0.000124   a_r mean/std = 78.835297/0.054431
↪  \Delta\omega_ref mean/std = 87.929003/3.094245
Block 10 orbits  901-1000: e_rmin mean/std = 0.234192/0.000143   a_r mean/std = 79.003725/0.042567
↪  \Delta\omega_ref mean/std = 90.584814/5.649679
```

 doi:10.20944/preprints202405.2028.v8

Appendix D. Future Research Directions and Experimental Tests

This section lists near–term, falsifiable tests that directly probe the claims of the Dual Ontology (DO) in mathematics, quantum physics, relativity, and cosmology. The list is scoped to work that can be initiated and meaningfully advanced within the next decade. Each item states a concise method, the readout, and a single–line purpose. Numerical items are labeled **Simulation** (or **Derivation/Observation**); laboratory items are labeled **Setup**. No external frameworks are assumed beyond the DO and standard numerical or experimental practice, except where explicitly labeled external.

1. **Operator split test:** $K_{\text{eff}} = K_{\text{kin}} + K_{\text{grav}}$
   Simulation. Run the same initial condition on the same graph, schedule, and time step in three modes: (a) kinetic exchange only, (b) gravitational exchange only, and (c) both terms enabled. Compare the combined evolution to the two single-term evolutions using the same read-only diagnostics, and repeat on a small time-step ladder to check that the "both terms" behavior is explained by the sum of the two components rather than by hidden cross-terms.
   Readout. Clear separation of effects (kinetic controls spreading/transport; gravity controls relational curvature response), with conservation and neutrality gates passing in all three modes; consistency of the combined run with the two component runs across the time-step ladder.
   Purpose. Provide a direct, DO-native demonstration that the effective operator decomposes into distinct kinetic and gravitational exchange components, rather than being a single unanalyzed update rule.
2. **Large-scale approximation and convergence of the discrete rule**
   Derivation. Provide a clear large-scale approximation of the discrete relational gravity rule and state the conditions under which it is used for interpretation and error control. Establish, in a simple and explicit way, that neutrality (uniform backgrounds do not drive motion) and the conservation audits used in the validations follow from the DO rule form, and report how key outputs converge as graph size and resolution are increased.
   Readout. Convergence trends and error bounds for the main orbital and lensing readouts; pass/fail reporting for neutrality and conservation gates across increasing graph sizes.
   Purpose. Make the DO gravity program mathematically audit-ready by showing controlled convergence from the discrete rule to its large-scale behavior.
3. **Kernel and operator realization refinement (DO-compliant)**
   Simulation. Systematically improve the realized gravity kernel and operator choices used in the hard-mass validations, staying strictly within the DO admissible kernel class and the fixed, background-independent update structure. Treat the kernel profile, cutoff/range choice, and the realized direction rule as realization data to be selected once, then held fixed. Use only DO-internal, read-only selection criteria (neutrality under uniform backgrounds, rotation/relabeling invariance, conservation-audit stability, barycenter and angular-momentum control, suppression of orientation texture, and suppression of slow secular orbit-shape drift under fixed initial conditions). After selecting a candidate realization, freeze it and re-run the multi-regime validation ladder under one χ_E.
   Readout. A ranked set of candidate realizations with gate summaries (pass/fail plus margins) for neutrality, invariants, rotation/relabeling, and long-horizon stability; before/after comparisons of orbit drift metrics and anisotropy indicators under the same initial conditions; cross-regime consistency under one fixed χ_E.
   Purpose. Separate DO law-level claims from realization artifacts by identifying a more isotropic, more stable kernel/operator realization that improves accuracy without adding new physics or mid-run tuning.
4. **Search for the true χ_E of 4D spacetime**
   Simulation. Fix one kernel scale-fixing convention, one direction rule, and one auditing suite. On an increasing graph-size ladder (and cutoff radii), fit the effective $\chi_E^{(N)}$ from one clean anchor readout (two-body period and/or perihelion advance), holding all other realization choices fixed.

 doi:10.20944/preprints202405.2028.v8

Extract the converged large-graph value $\chi_E^{(\infty)}$, then freeze $\chi_E = \chi_E^{(\infty)}$ and cross-check it across independent regimes (three-body stability, scattering/deflection curves, strong-field orbits, and lensing).

Readout. $\chi_E^{(N)}$ versus graph size with an uncertainty band; convergence summaries; audit-gate stability at each size (energy gate, angular momentum, barycenter); cross-regime residuals under the frozen χ_E.

Purpose. Determine whether the DO supports a graph-independent gravitational coupling for physical 4D spacetime, and separate finite-size artifacts from the law-level coupling strength.

5. **Relational gravity scale-up program (discrete, background-independent)**
Simulation. Using one fixed DO relational gravity rule and one coupling χ_E, run a staged validation ladder under the same fixed rule throughout: (a) short-horizon diagnostics that constrain the realization quickly (close-encounter scattering/deflection curves, flybys, and "massless-probe" trajectory diagnostics treated as read-only probes under the same steering rule); (b) full Solar System scale runs (major planets and moons, plus optional additional bodies) with fixed initial conditions and no mid-run tuning; (c) strong-field orbital tests in Mercury-like and compact regimes, including perihelion advance and stable turning-point behavior; (d) gravitational lensing readouts for realistic galaxy and cluster mass maps (weak/strong lensing and time-delay patterns); (e) Galactic Center star-orbit propagation (and pulsar timing where available); (f) galaxy and cluster scale comparisons using observed mass distributions as inputs and comparing rotation and lensing outputs; (g) extreme-horizon robustness audits reported as stability metrics (including slow secular shape changes) rather than pointwise trajectory claims. Apply the same rotation/relabeling gates and uniform-background neutrality checks at each tier.
Readout. Deflection versus impact parameter and audit drift during close encounters; orbit periods and secular rates for Solar System runs; perihelion advance and strong-field orbit stability metrics; lensing observables (shear/magnification/cusp relations/time delays); Galactic Center track residuals and precession signatures; rotation-curve residuals and cluster-scale lensing residuals; long-horizon drift rates and stability summaries reported as robustness metrics.
Purpose. Show that one discrete, purely relational gravity law scales from fast diagnostics to Solar System, strong-field, lensing, and galaxy/cluster regimes, and remains stable under long-horizon auditing, without introducing a spacetime metric or case-by-case retuning.
6. **Cosmic Microwave Background mapping to the full multipole range**
Simulation. Extend the existing CMB demonstration from the current limited band to the full measured temperature and polarization multipole range. Keep a single global amplitude on power for comparison only, and keep the same rotation and linearity gates.
Readout. Goodness-of-fit for spectral shape over stated multipole windows; residual structure tests; gate summaries.
Purpose. Test whether the same DO evolution and read-only mapping can match CMB structure across all measured angular scales, not just a segment.
7. **CMB initial conditions to first stars to galaxies to clusters (external baryonic module)**
Simulation. Starting from DO-consistent initial conditions, continue the DO evolution forward and couple it to an explicit baryonic astrophysics module (gas dynamics, cooling/chemistry, radiative transfer, and feedback), stated as *external to the DO* and documented separately with a sensitivity panel.
Readout. First-star opening times and counts; star-formation-rate history; first-galaxy assembly times and mass functions; clustering and correlation statistics; comparison with reionization and Lyman–alpha constraints; sensitivity of key outputs to the external module choices.
Purpose. Test whether DO-consistent initial conditions can be carried forward to realistic structure formation, while clearly separating DO-law outputs from added baryonic physics assumptions.

 doi:10.20944/preprints202405.2028.v8

8. **Entanglement-insensitive relational gravity (constituent sourcing)**
   Setup. Prepare two physical-source configurations with the same 4D spacetime energy distribution (same source energies, same geometry, same shielding and environment), but with different joint quantum-state structure: (a) an entangled singlet preparation (unified Bell Point in $S^{(PD)}$), and (b) a non-entangled preparation engineered to have the same local 4D energy distributions. Measure gravity-sensitive motion or deflection using the same probe and geometry in both cases.
   Readout. The gravitational readout (force/acceleration/deflection signal in the probe) is identical across (a) and (b) within experimental error; any reproducible dependence on entanglement state is recorded as a failure.
   Purpose. Test the DO claim that gravity does not couple to the unified Planck-Domain Bell Point, but only to the constituent 4D spacetime source distributions of the component states.
9. **Gravity-mediated entanglement (DO falsifier)**
   Setup. Prepare two well-isolated mesoscopic masses in controlled quantum states and arrange an interaction window in which gravitational coupling dominates while electromagnetic and other interactions are independently bounded. After the interaction window, perform an entanglement-witness readout between the two systems.
   Readout. A statistically significant entanglement signature attributable to gravity alone (after bounding non-gravitational couplings), or a null result reported as an upper bound.
   Purpose. Test a direct disproof condition for the DO: gravity-alone generation of entanglement would contradict the DO claim that gravity has no independent quantum mediator degrees of freedom.

References

1. Howard D. Who invented the "Copenhagen interpretation"? A study in mythology. Philos Sci. 2004;71(5):669–682. doi:10.1086/425941.
2. Dürr D, Goldstein S, Zanghi N. Bohmian mechanics as the foundation of quantum mechanics. ArXiv.org. 1995. doi:10.48550/arXiv.quant-ph/9511016.
3. Goldstein S, Zanghi N. Reality and the role of the wavefunction in quantum theory. ArXiv. 2011. doi:10.48550/arXiv.1101.4575.
4. Bassi A, Ghirardi G. Dynamical reduction models. Phys Rep. 2003;379(5–6):257–426. doi:10.48550/arXiv.quant-ph/0302164.
5. Bassi A, Lochan K, Satin S, Singh TP, Ulbricht H. Models of wave-function collapse, underlying theories, and experimental tests. Rev Mod Phys. 2012;85(2):471–527. doi:10.48550/arXiv.1204.4325.
6. Deutsch D. Quantum theory as a universal physical theory. Int J Theor Phys. 1985;24(1):1–41. doi:10.1007/bf00670071.
7. Wallace D. The emergent multiverse. Oxford (UK): Oxford University Press; 2012.
8. Vaidman L. Many-worlds interpretation of quantum mechanics. The Stanford Encyclopedia of Philosophy. 2021. Available from: https://plato.stanford.edu/archives/fall2021/entries/qm-manyworlds.
9. Allori V. Primitive ontology and the structure of fundamental physical theories. In: Ney A, Alberts D, editors. The wave function: Essays on the metaphysics of quantum mechanics. Oxford (UK): Oxford University Press; 2013. p. 58–75. Available from: https://philarchive.org/rec/ALLPOA.
10. Allori V. How to make sense of quantum mechanics: Fundamental physical theories and primitive ontology. PhilPapers. 2016. Available from: https://philpapers.org/rec/ALLQTM.
11. Carroll S. Reality as a vector in Hilbert space. In: Allori V, editor. Quantum mechanics and fundamentality. Cham (Switzerland): Springer; 2022. p. 211–225.
12. Cramer JG. The transactional interpretation of quantum mechanics. Rev Mod Phys. 1986;58(3):647–687. doi:10.1103/revmodphys.58.647.
13. Fuchs CA, Mermin ND, Schack R. An introduction to QBism with an application to the locality of quantum mechanics. Am J Phys. 2014;82(8):749–754. doi:10.1119/1.4874855.
14. Griffiths RB. Consistent quantum theory. Cambridge (UK): Cambridge University Press; 2003.
15. Hubert M, Romano D. The wave-function as a multi-field. Eur J Philos Sci. 2018;8(3):521–537. doi:10.1007/s13194-017-0198-9.

 doi:10.20944/preprints202405.2028.v8

16. Norsen T, Marian D, Oriols X. Can the wave function in configuration space be replaced by single-particle wave functions in physical space? Synthese. 2015;192(10):3125–3151. doi:10.1007/s11229-014-0577-0.
17. Rovelli C. Relational quantum mechanics. Int J Theor Phys. 1996;35(8):1637–1678. doi:10.1007/bf02302261.
18. Ambjørn J, Görlich A, Jurkiewicz J, Loll R. Quantum gravity via causal dynamical triangulations. In: Oriti D, editor. Approaches to quantum gravity: Toward a new understanding of space, time and matter. Cambridge (UK): Cambridge University Press; 2009. p. 723–741. doi:10.1007/978-3-642-41992-8_34.
19. Niedermaier M, Reuter M. The asymptotic safety scenario in quantum gravity. Living Rev Relativ. 2006;9(1):5. doi:10.12942/lrr-2006-5.
20. Hooft GT. Dimensional reduction in quantum gravity. ArXiv. 1993. Available from: https://arxiv.org/abs/gr-qc/9310026.
21. Danielsson U. Introduction to string theory. Rep Prog Phys. 2000;64(1):51–96. doi:10.1088/0034-4885/64/1/202.
22. Rovelli C, Smolin L. Spin networks and quantum gravity. Phys Rev D. 1995;52(10):5743–5759. doi:10.1103/physrevd.52.5743.
23. Monton B. Quantum mechanics and 3N-dimensional space. Philos Sci. 2006;73(5):778–789. doi:10.1086/518633.
24. Monton B. Wave function ontology. Synthese. 2002;130(2):265–277. Available from: https://philarchive.org/rec/MONWFO-2.
25. Ylikoski P, Kuorikoski J. Dissecting explanatory power. Philos Stud. 2010;148(2):201–219. doi:10.1007/s11098-008-9324-z.
26. Wolf WJ, Thébault KPY. Explanatory depth in primordial cosmology: A comparative study of inflationary and bouncing paradigms. Br J Philos Sci. 2023;725096. doi:10.1086/725096.
27. Azhar F, Loeb A. Finely tuned models sacrifice explanatory depth. Found Phys. 2021;51(5):1–22. doi:10.1007/s10701-021-00493-2.
28. Crouse DT. On the nature of discrete spacetime: The atomic theory of spacetime and its effects on Pythagoras's theorem, time versus duration, inertial anomalies of astronomical bodies, and special relativity at the Planck scale. ArXiv. 2016. doi:10.48550/arXiv.1608.08506.
29. Hagar A. Discrete or continuous?: The quest for fundamental length in modern physics. Cambridge (UK): Cambridge University Press; 2015. Kindle edition.
30. Hossenfelder S. Minimal length scale scenarios for quantum gravity. Living Rev Relativ. 2013;16(1). doi:10.12942/lrr-2013-2.
31. Hossenfelder S. Theory and phenomenology of spacetime defects. Adv High Energy Phys. 2014;2014:1–6. doi:10.1155/2014/950672.
32. Smolin L. Atoms of space and time. Sci Am. 2004;290(1):66–75. Available from: https://www.jstor.org/stable/26172656.
33. Lucretius. On the nature of things. Translated, with introduction and notes, by Smith MF. Indianapolis (IN): Hackett Publishing Company, Inc.; 2001.
34. Rovelli C. Reality is not what it seems: The journey to quantum gravity. New York (NY): Penguin Publishing Group; 2017. Kindle edition.
35. Carroll S. What is nothing? Vice Magazine Online. 2018 Oct 31. Available from: https://www.vice.com/en/article/vbk5va/what-is-nothing.
36. Barrow JD. The book of nothing. London (UK): Vintage; 2001. Available from: https://philpapers.org/rec/BARTBO-11.
37. Grünbaum A. Why is there a world at all, rather than just nothing? Ontology Stud. 2009;9:7–19.
38. Holt J. Why does the world exist?: An existential detective story. New York (NY): Liveright; 2012. Kindle edition.
39. Leslie J, Kuhn RL. The mystery of existence. Hoboken (NJ): John Wiley & Sons; 2013.
40. Moghri M. Much ado about nothingness? Kriterion (Salzburg). 2020;34(3):79–98. doi:10.1515/krt-2020-340305.
41. Lewis PJ. Dimension and illusion. In: Ney A, Albert D, editors. The wave function: Essays on the metaphysics of quantum mechanics. Oxford (UK): Oxford University Press; 2013. p. 110–125. Available from: http://philsci-archive.pitt.edu/8841/.
42. Banerjee S, Bera S, Singh TP. Quantum nonlocality and the end of classical spacetime. Int J Mod Phys D. 2016;25(12):1644005. doi:10.1142/s0218271816440053.

 doi:10.20944/preprints202405.2028.v8

43. Adams FC. The degree of fine-tuning in our universe — and others. Phys Rep. 2019;807:1–111. doi:10.1016/j.physrep.2019.02.001.
44. To be completed following peer review.
45. Norsen T. Foundations of quantum mechanics: An exploration of the physical meaning of quantum theory. Cham (Switzerland): Springer; 2017.
46. Maudlin T. The nature of the quantum state. In: Albert D, Ney A, editors. The wave function: Essays on the metaphysics of quantum mechanics. Oxford (UK): Oxford University Press; 2013. p. 126–153. doi:10.1093/acprof:oso/9780199790807.003.0006.
47. Maudlin T. Philosophy of physics: Quantum theory. Princeton (NJ): Princeton University Press; 2019. Kindle edition.
48. Monton B. Quantum mechanics and 3N-dimensional space. Philos Sci. 2006;73(5):778–789. doi:10.1086/518633.
49. Pusey MF, Barrett J, Rudolph T. On the reality of the quantum state. Nat Phys. 2012;8(6):475–478. doi:10.1038/nphys2309.
50. Albert D. Wave function realism. In: Albert D, Ney A, editors. The wave function: Essays on the metaphysics of quantum mechanics. Oxford (UK): Oxford University Press; 2013. p. 51–56. doi:10.1093/acprof:oso/9780199790807.003.0006.
51. Chen EK. Realism about the wave function. Philos Compass. 2019;14(7). doi:10.1111/phc3.12611.
52. Ney A. The world in the wave function: A metaphysics for quantum physics. Oxford (UK): Oxford University Press; 2021. Kindle edition.
53. Ney A. Three arguments for wave function realism. Eur J Philos Sci. 2023;13(4). doi:10.1007/s13194-023-00554-5.
54. Allori V. Spontaneous localization theories. In: Bacciagaluppi G, Freire O Jr, editors. Oxford handbook of the history of interpretations and foundations of quantum mechanics. Oxford (UK): Oxford University Press; 2022. p. 1103–34. Available from: https://philpapers.org/rec/ALLSLT-3.
55. Bricmont J. Making sense of quantum mechanics. Cham (Switzerland): Springer; 2016. Available from: https://philpapers.org/rec/BRIMSO.
56. de Broglie L. The current interpretation of wave mechanics. Amsterdam (Netherlands): Elsevier Publishing Company; 1964.
57. Norsen T. Einstein's boxes. Am J Phys. 2005;73(2):164–176. doi:10.1119/1.1811620.
58. Sebens CT. Electron charge density: A clue from quantum chemistry for quantum foundations. Found Phys. 2021;51(4). doi:10.1007/s10701-021-00480-7.
59. Sebens CT. Eliminating electron self-repulsion. Found Phys. 2023;53(4). doi:10.1007/s10701-023-00702-0.
60. Lewis PJ. Quantum ontology: A guide to the metaphysics of quantum mechanics. Oxford (UK): Oxford University Press; 2016. Kindle edition.
61. Licata I, Chiatti L. Event-based quantum mechanics: A context for the emergence of classical information. Symmetry. 2019;11(2):181. doi:10.3390/sym11020181.
62. Bassi A, Ulbricht H. Collapse models: From theoretical foundations to experimental verifications. J Phys Conf Ser. 2014;504:012023. doi:10.1088/1742-6596/504/1/012023.
63. Ghirardi G. Sneaking a look at God's cards. Princeton (NJ): Princeton University Press; 2004.
64. Allori V, Bassi A, Durr D, Zanghi N. Do wave functions jump?: Perspectives of the work of GianCarlo Ghirardi. Cham (Switzerland): Springer; 2021. Available from: https://link.springer.com/book/10.1007/978-3-030-46777-7.
65. Bacciagaluppi G, Valentini A. Quantum theory at the crossroads: Reconsidering the 1927 Solvay conference. ArXiv (Cornell University). 2009. Available from: https://arxiv.org/abs/quant-ph/0609184.
66. McQueen KJ. Four tails problems for dynamical collapse theories. Stud Hist Philos Sci B. 2015;49:10–18. doi:10.1016/j.shpsb.2014.12.001.
67. Hossenfelder S, Palmer TN. Rethinking superdeterminism. Front Phys. 2020;8:139. doi:10.3389/fphy.2020.00139.
68. Howard D. Einstein on locality and separability. Stud Hist Philos Sci. 1985;16(3):171–201. doi:10.1016/0039-3681(85)90001-9.
69. Maudlin T. Quantum nonlocality & relativity: Metaphysical intimations of modern physics. 3rd ed. Hoboken (NJ): John Wiley & Sons; 2011. Kindle edition.
70. Wiseman HM. From Einstein's theorem to Bell's theorem: A history of quantum nonlocality. Contemp Phys. 2006;47(2):79–88. doi:10.1080/00107510600581011.

 doi:10.20944/preprints202405.2028.v8

71. Gao S. Quantum theory is incompatible with relativity: A new proof beyond Bell's theorem and a test of unitary quantum theories. PhilSci Archive. 2019. Available from: http://philsci-archive.pitt.edu/16155/.
72. Ney A. Separability, locality, and higher dimensions in quantum mechanics. In: Dasgupta S, Weslake B, editors. Current controversies in philosophy of science. London: Routledge; 2020. p. 75–90.
73. Goldstein S, et al. Bell's theorem. Scholarpedia. 2011;6(10):8378.
74. Maudlin T. What Bell did. J Phys A Math Theor. 2014;47(42):424010. doi:10.1088/1751-8113/47/42/424010.
75. Bell M, Gao S. Quantum nonlocality and reality: 50 years of Bell's theorem. Cambridge (UK): Cambridge University Press; 2016.
76. Brunner N, Cavalcanti D, Pironio S, Scarani V, Wehner S. Bell nonlocality. Rev Mod Phys. 2014;86(2):419–78. doi:10.1103/revmodphys.86.419.
77. Norsen T. John S. Bell's concept of local causality. Am J Phys. 2011;79(12):1261–1275. doi:10.1119/1.3630940.
78. Albert DZ. Time and chance. Cambridge (MA): Harvard University Press; 2000.
79. Bahrami M, Bassi A, Donadi S, Ferialdi L, León G. Irreversibility and collapse models. In: Müller A, Filk T, editors. Re-thinking time at the interface of physics and philosophy. Vol. 4. Cham (Switzerland): Springer; 2015. p. 125–146. doi:10.1007/978-3-319-10446-1_6.
80. Doyle R. The origin of irreversibility. Department of Astronomy, Harvard University. 2014. Available from: https://www.informationphilosopher.com/problems/reversibility/Irreversibility.pdf.
81. Price H. On the origins of the arrow of time: Why there is still a puzzle about the low entropy past. In: Hitchcock C, editor. Contemporary debates in the philosophy of science. Hoboken (NJ): Blackwell Publishing Ltd; 2004. p. 219–239. Available from: https://philpapers.org/rec/PRIOTO-2.
82. Curiel E. Singularities and black holes. In: Nalta EA, Nodelman U, editors. The Stanford Encyclopedia of Philosophy (Summer 2023 Edition). 2009. Available from: https://plato.stanford.edu/archives/sum2023/entries/spacetime-singularities/.
83. Bedingham D. Collapse models, relativity, and discrete spacetime. In: Allori V, Bassi A, Durr D, Zanghi N, editors. Do wave functions jump? Perspectives of the work of GianCarlo Ghirardi. Cham (Switzerland): Springer; 2021. p. 191–203. doi:10.1007/978-3-030-46777-7_15.
84. Oppenheim J. A postquantum theory of classical gravity? Phys Rev X. 2023;13(4). doi:10.1103/physrevx.13.041040.
85. Norsen T. Foundations of quantum mechanics: An exploration of the physical meaning of quantum theory. Cham (Switzerland): Springer; 2017.
86. Smolin L. The case for background independence. In: Rickles D, French S, Saatsi J, editors. The structural foundations of quantum gravity. Oxford (UK): Oxford University Press; 2006. p. 196–239. Available from: http://arxiv.org/abs/hep-th/0507235v1.
87. Barceló C, Carballo-Rubio R, Liberati S. Generalized no-hair theorems without horizons. Class Quantum Grav. 2019;36(13):13LT01. doi:10.1088/1361-6382/ab23b6.
88. López-Corredoira M. Physics and reality. J Phys Conf Ser. 2025;2948:012001. doi:10.1088/1742-6596/2948/1/012001.
89. Bojowald M. Quantum cosmology: A review. Rep Prog Phys. 2015;78(2):023901. doi:10.1088/0034-4885/78/2/023901.
90. Aghanim N, et al. Planck 2018 results. VI. Cosmological parameters. ArXiv. 2019. Available from: https://arxiv.org/abs/1807.06209v4.
91. Davies P. The last three minutes: Conjectures about the ultimate fate of the universe. New York (NY): Basic Books; 1995.
92. Ijjas A, Steinhardt PJ, Loeb A. Inflationary paradigm in trouble after Planck2013. ArXiv. 2013. doi:10.48550/arXiv.1304.2785.
93. Melia F. The Friedmann–Lemaître–Robertson–Walker metric and the principle of equivalence. Z Naturforsch A. 2023;78(6):525–533. doi:10.1515/zna-2022-0307.
94. Penrose R. Before the big bang: An outrageous new perspective and its implications for particle physics. Conf Proc C. 2006;060626:2759–2767.
95. Guth AH. Inflationary universe: A possible solution to the horizon and flatness problems. Phys Rev D. 1981;23(2):347–356. doi:10.1103/physrevd.23.347.
96. León G, Landau SJ, Piccirilli MP. Quantum collapse as a source of the seeds of cosmic structure during the radiation era. Phys Rev D. 2014;90(8). doi:10.1103/physrevd.90.083525.
97. Okon E, Sudarsky D. The weight of collapse: Dynamical reduction models in general relativistic contexts. ArXiv Gen Relat Quantum Cosmol. 2017. doi:10.48550/arxiv.1701.02963.

 doi:10.20944/preprints202405.2028.v8

98. Pérez A, Sahlmann H, Sudarsky D. On the quantum origin of the seeds of cosmic structure. Class Quantum Grav. 2006;23(7):2317–2354. doi:10.1088/0264-9381/23/7/008.
99. Sudarsky D. The quantum origin of the seeds of cosmic structure: The case for a missing link. AIP Conf Proc. 2010;1256:107–121. doi:10.1063/1.3473844.
100. Wald RM. The arrow of time and the initial conditions of the universe. Stud Hist Philos Sci B. 2006;37(3):394–398. doi:10.1016/j.shpsb.2006.03.005.
101. Riess AG, et al. Observational evidence from supernovae for an accelerating universe and a cosmological constant. Astron J. 1998;116(3):1009–1038. doi:10.1086/300499.
102. Allori V. What if we lived in the best of all possible (quantum) worlds? PhilSci-Archive. 2023. Available from: http://philsci-archive.pitt.edu/21840/1/what-if-latest.pdf.