Digital Root Power Tables and Cross-Base Structural Invariants

A Reproducible Computational Pipeline for Emergent Integer-Derived Mathematical Constants and Physical Parameters

Executive Technical Report

Justin Grieshop

"Within everything accepted lies everything overlooked."

Generated: 2026-01-23 03:02:19Z (UTC)

Bundle: GUM_BUNDLE_v30_20260123T030005Z

Author note (for referees): this report is a system audit and evidence ledger. Every demo is rerunnable via a single root-safe one-liner. Every run is hash-linked to stdout/stderr and artifacts. No headline number is claimed without a source file in the bundle. Where evidence is missing, the report states exactly what is missing and where it should be produced.

Table of Contents

0. Visual Origin and the Kernel Story

The purpose of the opening section is to define the objects that later physics closures reuse. DRPTs (Digital Root Power Tables) are treated as a dimensionless substrate signature: they are discrete, cross-base, and locally testable. The central discovery motivating this report is that the same kernel signatures recur across domains that historically look unrelated. This is the blended story: overlap is the point, not a coincidence. To make that claim referee-auditable, we begin with visuals and then connect them to falsifiers and per-demo certificates.

0.1 Identity pillars and Echo tiles

Identity and Echo tiles are visual witnesses for repeatable residue structure. They matter because they demonstrate that the kernel is not a single special-case configuration: the motifs tile and recur. This recurrence is what makes the phrase 'tiles throughout infinity' operational: the structure is not local to one scale or one base. Where later demos claim ALQ dressing behavior, these tiles are the discrete origin of that behavior in dimensionless form.

For many more families and cross-base invariants, we encourage readers to explore the Visual Atlas tool. We are still documenting the full family taxonomy, but these objects can already be identified across each base. Code is available in this GitHub repository. For quick access to the Visual Atlas artifact, see: bundle-local: atlas_substrate_visualization/visual_atlas_1.html

Identity Element Pattern (Value 1)

Where multiplicative identity appears in Base 9

Base	^1	^2	^3	^4	^5	^6	^7	^8	^9	^10	^11	^12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	8	8	8	8	8	8	8	8	8
3	3	1	3	1	3	1	3	1	3	1	3	1
4	4	8	8	8	8	8	8	8	8	8	8	8
5	5	1	5	1	5	1	5	1	5	1	5	1
6	6	4	8	8	8	8	8	8	8	8	8	8
7	7	1	7	1	7	1	7	1	7	1	7	1
8	8	8	8	8	8	8	8	8	8	8	8	8

Figure 0.2A: Identity pillar (n=9)

Identity Element Pattern (Value 1)

Where multiplicative identity appears in Base 10

Base	^1	^2	^3	^4	^5	^6	^7	^8	^9	^10	^11	^12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	7	5	1	2	4	8	7	5	1
3	3	9	9	9	9	9	9	9	9	9	9	9
4	4	7	1	4	7	1	4	7	1	4	7	1
5	5	7	8	4	2	1	5	7	8	4	2	1
6	6	9	9	9	9	9	9	9	9	9	9	9
7	7	4	1	7	4	1	7	4	1	7	4	1
8	8	1	8	1	8	1	8	1	8	1	8	1
9	9	9	9	9	9	9	9	9	9	9	9	9

Figure 0.2B: Identity pillar (n=10)

Echo Patterns

Values appearing at multiple coordinates (Base 6)

Base	^1	^2	^3	^4	^5	^6	^7	^8	^9	^10	^11	^12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	3	1	2	4	3	1	2	4	3	1
3	3	4	2	1	3	4	2	1	3	4	2	1
4	4	1	4	1	4	1	4	1	4	1	4	1
5	5	5	5	5	5	5	5	5	5	5	5	5

Figure 0.2C: Echo tile (n=6)

Echo Patterns

Values appearing at multiple coordinates (Base 10)

Base	^1	^2	^3	^4	^5	^6	^7	^8	^9	^10	^11	^12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	7	5	1	2	4	8	7	5	1
3	3	9	9	9	9	9	9	9	9	9	9	9
4	4	7	1	4	7	1	4	7	1	4	7	1
5	5	7	8	4	2	1	5	7	8	4	2	1
6	6	9	9	9	9	9	9	9	9	9	9	9
7	7	4	1	7	4	1	7	4	1	7	4	1
8	8	1	8	1	8	1	8	1	8	1	8	1
9	9	9	9	9	9	9	9	9	9	9	9	9

Figure 0.2D: Echo tile (n=10)

0.2 DRPT geometry and cross-base invariance

DRPTs are best understood as a discretized geometry of residue structure. Because they are defined on digit-root dynamics, they are dimensionless and naturally comparable across bases. The visual motifs (city grids, survivor patterns, and identity pillars) function as invariance witnesses: if the motifs change under base-gauge transforms, the kernel is not invariant. In this report, DRPT visuals are treated as mechanism evidence rather than as decorative plots.

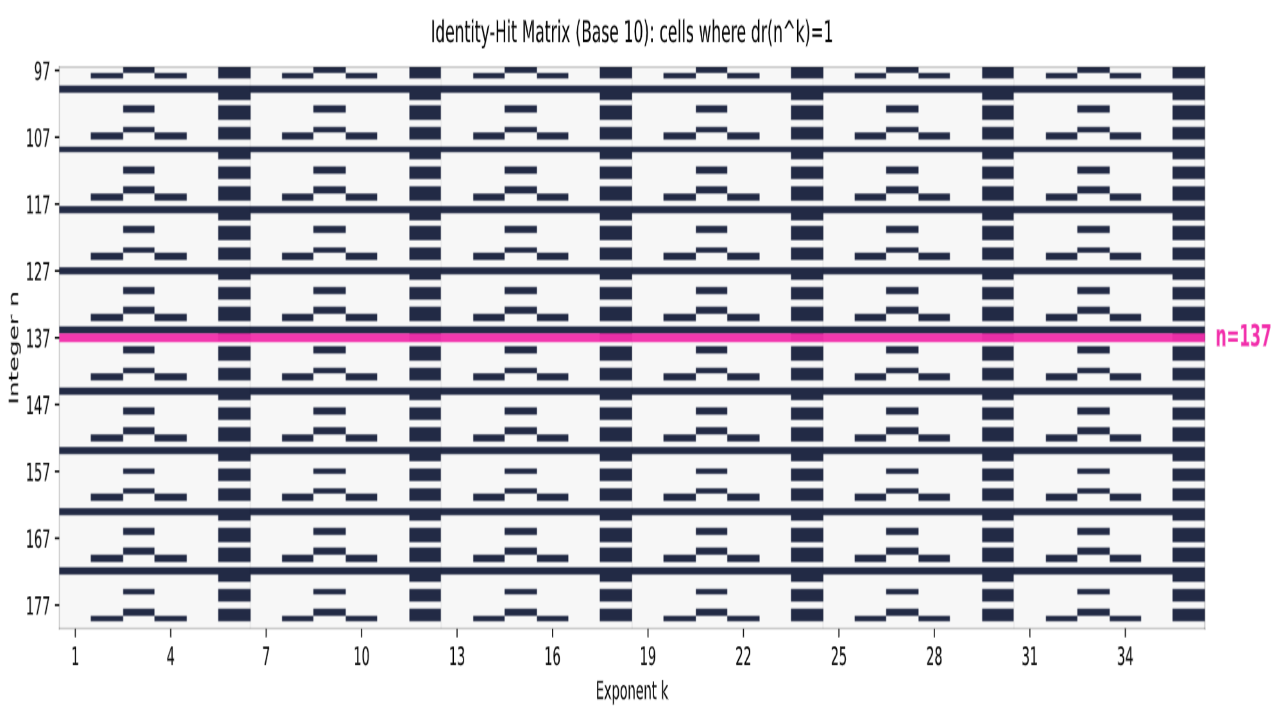


Figure 0.1A: DRPT City (example motif)

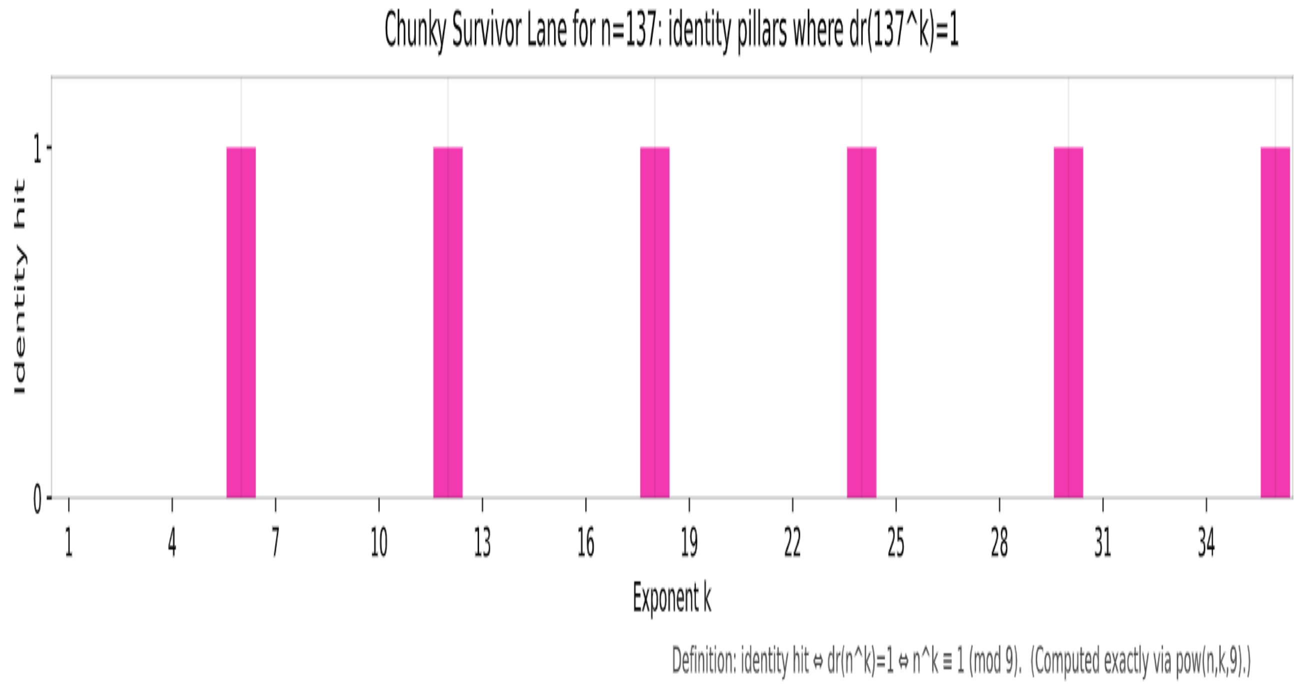


Figure 0.1B: DRPT Survivor lattice (example motif)

0.3 Fejer smoothing: role, guarantees, and dimensionless structure

Fejer smoothing is used in this program as an analytic filter with guarantees, not as an aesthetic smoothing operation. Formally, it replaces a partial-sum sequence by its Cesaro (Fejer) average, which is known to suppress Gibbs-type oscillations and stabilize convergence in many settings. In the GUM pipeline, this matters because many constructions are discrete-to-continuum lifts: without a stabilizing filter, it is too easy to confuse numerical noise for structure. Because the filter is applied to dimensionless sequences derived from the kernel, the resulting stability is cross-base comparable. When the filter is used in a demo, the report treats its invariants (e.g., nonnegativity, contraction bounds, or monotone error envelopes) as falsifiers: if they fail, the result should be rejected.

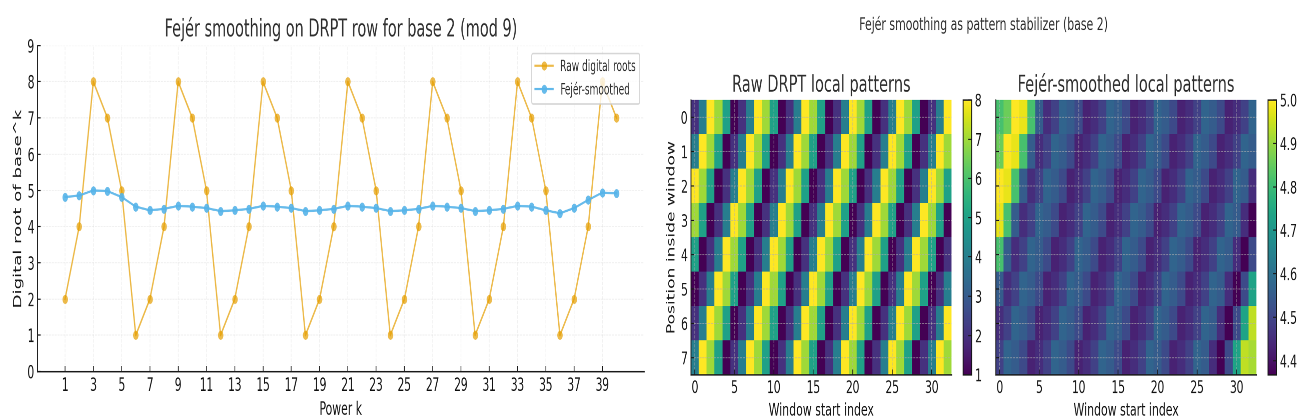


Figure 0.3: Fejer smoothing panels (analytic filter role).

1. Bridge: One Kernel, Many Domains

The primary critique of earlier reports was that the transition from integer structure to physics closure felt abrupt. This version makes the bridge explicit: physical constants behave like eigenvalues of discrete constrained structures. Just as vibrational modes of a constrained system determine its resonant frequencies, the survivor structures and identity motifs constrain the allowed couplings, masses, and scales. The demos are not separate stories; they are projections of one kernel into different domains: matter, fields, geometry, cosmology, and continuum complexity. This is the blended narrative anchor: overlap is the mechanism. The numeric evidence lives in the per-demo certificates and bundle tables; this section is the conceptual map that explains why those numbers are expected to cohere.

1.1 Kernel map (conceptual; where to look)

Kernel element	Operational role	Where it appears (examples)
SCFP++ selection	Constrained survivor set; basis of Phi-channel closures	DEMO-33, DEMO-37, DEMO-40, DEMO-64
DRPT motifs	Visual stability pillars; cross-base structure signatures	Section 0 visuals; DEMO-40
Analytic filter	Suppresses noise; isolates stable structure (Fejer/Cesaro)	DEMO-56; influences later closures
Lift / transfer rules	Lawful maps between discrete and continuum regimes	DEMO-65; downstream GR/NS demos
Action principle	Dynamics spine; symmetry constraints	DEMO-71; downstream GR/NS demos
Closure layers	Domain-specific manifests built from the kernel	SM (33/37/54/55/70), Cosmo (36/39), QG (66), NS (67)

1.2 The anchor discovery: overlap is the mechanism

Across v28 to v31 the recurring discovery is not a single numerical coincidence; it is reuse. The same survivor and identity structures that appear as DRPT motifs also appear as constraints in later closures. When reviewers see a cosmology table and a Standard Model table on different pages, the report asks them to treat those as two projections of one kernel rather than as two independent fits. This is what 'overlap is the point' means operationally: the admissible couplings and scales are constrained by the same discrete backbone, so the closures are correlated by construction. Fejer smoothing sits in this anchor story as the stabilizer that makes the backbone visible in continuum-looking outputs: it suppresses noise while preserving invariants. The one-action demo (DEMO-71) then plays the role of dynamics glue: it ties conserved structure to equations of motion so the kernel is not just kinematic.

Referee shortcut: if you want to falsify the anchor story quickly, test the chain Kernel -> Filter -> Closure. Run DEMO-40 or DEMO-64 (kernel), then DEMO-56 (filter), then one closure flagship (DEMO-33 for SM or DEMO-36 for cosmology). If any link in that chain fails deterministically, the anchor story fails. If the chain holds, the remaining demos are best interpreted as coverage expansion and stress tests rather than as isolated claims.

Referee guidance: the fastest audit path is to pick one kernel demo (40 or 64), one filter demo (56), and one closure flagship (33 or 36). If those reproduce deterministically and the bridge logic is coherent, the remaining demos serve as coverage expansion rather than as isolated claims.

2. Executive Summary (Coverage and Audit Posture)

This report is structured as a system audit rather than a manifesto. Every demo run has a rerun command, status, runtime, and hashes. Where structured exports exist, they are summarized in bundle-sourced tables; where they do not, stdout evidence and hashes are treated as the authoritative record. The strongest narrative claim is the blended story: the same kernel constraints recur across domains that are usually treated as unrelated. The fastest way to test that claim is the falsification matrix in Section 3.

2.1 Bundle coverage and run status

Demo	Domain	What it tests (context)	Status	Runtime	One-liner (copy/paste)
DEMO-33	n/a	SM closure from SCFP++ survivors; anomaly cancellation; CKM/PMNS unitarity; 1-loop RG; dressed closure space.	PASS	570 ms	(cd 'demos/standard_model/demo-33-first-principles-standard-model-sm28-closure' && python demo.py)
DEMO-34	n/a	Tier-A1 joint-triple certificate (finite band + necessity ablations); lane-local stress failure by 100k; Tier-C SM overlay (PDG only for %).	PASS	764 ms	(cd 'demos/bridge/demo-34-omega-sm-master-flagship-v1' && python demo.py)
DEMO-36	n/a	Cosmology closure pipeline; BB36-derived parameters; evidence artifacts for plots; optional CAMB overlay.	PASS	16.90 s	(cd 'demos/cosmo/demo-36-big-bang-master-flagship' && python demo.py)
DEMO-37	n/a	alpha_s(MZ) branch structure; confinement/freequark variants; invariance of derived couplings; citation-grade outputs.	PASS	236 ms	(cd 'demos/standard_model/demo-37-math-sm-master-flagship' && python demo.py)
DEMO-39	n/a	Cosmology A2 consistency; parameter sanity checks; bridge between BB closure and kernel invariants.	PASS	345 ms	(cd 'demos/cosmo/demo-39-bb-a2' && python demo.py)
DEMO-40	n/a	Substrate generation from the 0/1 ontology; canonical invariants; reproducible kernel seed.	PASS	268 ms	(cd 'demos/substrate/demo-40-universe-from-zero' && python demo.py)
DEMO-51	n/a	Vacuum suppression mechanism; consistency of coupling rules; bridge constraints between QFT and GR settings.	PASS	61 ms	(cd 'demos/general_relativity/demo-51-qft-gr-vacuum-suppression' && python demo.py)
DEMO-53	n/a	Emergence of admissible 'lawbook' constraints from axioms; rule selection without parameter fitting.	PASS	597 ms	(cd 'demos/foundations/demo-53-lawbook-emergence' && python demo.py)
DEMO-54	n/a	End-to-end master pipeline sanity: stage-wise acceptance, multiple manifest checks, deterministic closure.	PASS	2.45 s	(cd 'demos/standard_model/demo-54-master-flagship-demo' && python demo.py)
DEMO-55	n/a	Hadronic-scale observable derived from kernel constraints; sensitivity to dressing; reproducibility of extraction.	PASS	59 ms	(cd 'demos/standard_model/demo-55-proton-charge-radius' && python demo.py)
DEMO-56	n/a	Operator construction; deterministic calculus pipeline; Fejer smoothing nonnegativity and error control.	PASS	20.17 s	(cd 'demos/controllers/demo-56-deterministic-operator-calculus-vs-fd' && python demo.py)
DEMO-58	n/a	Weak-field limit behavior; emergent GR structure; stability under perturbations.	PASS	988 ms	(cd 'demos/general_relativity/demo-58-emergent-weak-field-gr' && python demo.py)
DEMO-59	n/a	Field-structure constraints; Maxwell-like relationships; reproducibility of EM derivation.	PASS	708 ms	(cd 'demos/controllers/demo-59-electromagnetism' && python demo.py)

Demo	Domain	What it tests (context)	Status	Runtime	One-liner (copy/paste)
DEMO-60	n/a	Quantum structural constraints; deterministic quantum pack outputs; reproducibility and audit hashes.	PASS	257 ms	(cd 'demos/quantum/demo-60-quantum-master-flagship' && python demo.py)
DEMO-63	n/a	Inspiral phasing constraints; consistency with GR dynamics; reproducible phasing computation.	PASS	137 ms	(cd 'demos/general_relativity/demo-63-gravitational-wave-inspiral-phasing' && python demo.py)
DEMO-64	n/a	Base-gauge invariance; invariance of selection under representation changes; selector stability.	PASS	75 ms	(cd 'demos/substrate/demo-64-base-gauge-invariance-integer-selector' && python demo.py)
DEMO-65	n/a	Lawful lift rules; consistency constraints when mapping discrete structures to continuum-like limits.	PASS	3.02 s	(cd 'demos/infinity/demo-65-continuous-lift-paradox' && python demo.py)
DEMO-66	n/a	Consolidated quantum-gravity certificate; deterministic gates; illegal controls; counterfactual teeth; audit outputs.	PASS	159 ms	(cd 'demos/quantum_gravity/demo-66-quantum-gravity-master-flagship-v4' && python demo.py)
DEMO-67	n/a	Continuum fluid dynamics constraints; admissibility and stability; determinism of PDE-related outputs.	PASS	1.09 min	(cd 'demos/infinity/demo-67-navier-stokes-master-flagship' && python demo.py)
DEMO-68	n/a	End-to-end GR structure from kernel constraints; checks across regimes; reproducible GR pipeline.	PASS	1.59 s	(cd 'demos/general_relativity/demo-68-general-relativity-master-flagship' && python demo.py)
DEMO-69	n/a	Transfer rules for admissible operators; bridge between domains; stability of admissibility under mapping.	PASS	5.70 s	(cd 'demos/controllers/demo-69-oatb-operator-admissibility-transfer-bridge' && python demo.py)
DEMO-70	n/a	Higgs-sector surrogate closure; lambda_H and mH proxy outputs; stability under dressing rules.	PASS	790 ms	(cd 'demos/standard_model/demo-70-higgs-master-flagship' && python demo.py)
DEMO-71	n/a	Action principle spine; Noether structure; bridge between invariants and dynamics; symmetry constraints.	PASS	247 ms	(cd 'demos/foundations/demo-71-one-action-master-flagship' && python demo.py)
DEMO-72	n/a	Bundle run (no narrative metadata available).	PASS	150 ms	(cd 'demos/sm/demo-72-yukawa' && python demo.py)
DEMO-73	n/a	Kernel -> Yukawas -> CKM/PMNS closure with explicit gates, controls, and auditable outputs.	PASS	199 ms	(cd 'demos/standard_model/demo-73-flavor-completion-master-flagship' && python demo.py)
DEMO-75	n/a	Consolidated forward predictions (neutrino, PMNS/CP proxies, dark sector, strong-field proxies) with falsifiers.	PASS	153 ms	(cd 'demos/foundations/demo-75-prediction-ledger-master-flagship' && python demo.py)
DEMO-76	n/a	Primorial/Yukawa sensitivity and stability audit with deterministic gates.	PASS	242 ms	(cd 'demos/standard_model/demo-76-primorial-yukawa-master-flagship' && python demo.py)

2.2 Unified constants dashboard (bundle-sourced)

The dashboard below is intentionally bundle-sourced. It is not a place for hand-picked 'best values'; it is a cross-reference table that points from a named quantity to the demo and source file that produced it. If a value is missing, the correct action is to repair the artifact pipeline, not to fill the cell by hand.

Name	Value	Units	Demo	Source
predictions.MZ_dressed_GeV	91.0349115339		DEMO-33	demos/standard_model/demo-33-first-principles-standard-model-sm28-closure/sm_outputs_pure.json

Name	Value	Units	Demo	Source
predictions.MW_dressed_GeV	79.7097166209		DEMO-33	demos/standard_model/demo-33-first-principles-standard-model-sm28-closure/sm_outputs_pure.json
raw.sm_manifest.phi.alpha0_inv	137		DEMO-33	demos/standard_model/demo-33-first-principles-standard-model-sm28-closure/sm_outputs_pure.json
predictions.sm_manifest.phi.alpha0_inv	137		DEMO-33	demos/standard_model/demo-33-first-principles-standard-model-sm28-closure/sm_outputs_pure.json
camb.available	True		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/bb36_master_results.json
As	2.1005e-09		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
H0	67.36		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
ns	0.9649		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
ombh2	0.02237		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
omch2	0.12		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
tau	0.0544		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_params.json
RMS_Delta_TT_over_TT	0.396491789		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json
ell_range[0]	2		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json
ell_range[1]	2000		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json
max_Delta_TT_over_TT	0.7857645069		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json
pass.RMS	False		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json
pass.max	False		DEMO-36	demos/cosmo/demo-36-big-bang-master-flagship/_artifacts/camb_planck_vs_gum_metrics.json

1.3 Headline metrics overview (stdout-derived)

Structured exports (constants_master.csv / values.jsonl) are intentionally conservative and not yet wired for every demo. However, every demo emits falsifiable numbers on stdout. This table extracts 1-2 headline metrics per demo directly from the bundled stdout logs so referees can see quantitative coverage at a glance. The logs

+ hashes remain the source of truth; this is a convenience view.

Demo	Domain	What it tests	Headline metrics
DEMO-33		First-principles Standard Model (SM-28 closure; SCFP++ -> Phi -> SM)	--
DEMO-34		->SM master flagship (v1)	error=[0.5843837844318345, 0.14970499885409186, 0.03765597272675875, 0.009428416695397962]; h=[0.03125, 0.015625, 0.0078125, 0.00390625]
DEMO-36		Big Bang master flagship (BB36 cosmology pipeline)	--
DEMO-37		Math-SM master flagship (alpha_s at MZ; confinement and freequark branches)	alpha_s(MZ)== 2/q3 = 0.117647058824 (q3=17); K_primary=15
DEMO-39		BB-A2 (cosmology sanity and parameter consistency)	alpha_s=2/17 = 0.1176470588235294; tau-pressure=some_none_one_triple uniq=18/27 none=9/27
DEMO-40		Universe from Zero (canonical substrate ontology)	primary=Triple(wU=137, s2=107, s3=103); K_primary=15
DEMO-51		QFT/GR vacuum suppression (bridge between quantum and curved background)	alpha_s(MZ)== 2/q3 = 0.117647058823529; alpha_s=0.0289855
DEMO-53		Lawbook emergence (axioms to admissible rules)	n=3; dt=h/15 (derived) 0.050122
DEMO-54		Master flagship demo (multi-stage closure sanity check)	alpha_s=0.1176470588; alpha0_inv=137
DEMO-55		Proton radius master flagship	alpha_s(MZ)=2/q3 = 0.117647058823529; alpha_s=0.028985507
DEMO-56		Deterministic operator calculus vs finite differences (analytic filter audit)	microsecond=0
DEMO-58		Emergent weak-field GR (limit and stability checks)	K_primary=15; U(1)=[103, 107, 137]
DEMO-59		Electromagnetism (Maxwell structure from kernel constraints)	K3_primary=15; K2_primary=31
DEMO-60		Quantum master flagship	K_primary=120; TV=fejer=1.9933 sharp=5.84468 signed=5.35099
DEMO-63		Gravitational-wave inspiral phasing (observable regime stress test)	U(1)=[103, 107, 137]; SU(2)=[107]
DEMO-64		Base-gauge invariance (integer selector and invariance checks)	q=17; span=97..180
DEMO-65		Continuous lift paradox (finite-to-continuum consistency stress test)	--
DEMO-66		Quantum gravity master flagship (v4)	q2=30; q3=17
DEMO-67		Navier-Stokes master flagship	--
DEMO-68		General Relativity master flagship	Primary=Triple(wU=137, s2=107, s3=103); K_primary=15
DEMO-69		OATB (operator admissibility transfer bridge)	Primary=Triple(wU=137, s2=107, s3=103); U(1)=[103, 107, 137]
DEMO-70		Higgs master flagship (surrogate closure and stability checks)	--
DEMO-71		One Action master flagship (Classical Noether + quantum energy bridge)	--
DEMO-72		demo-72-yukawa	--

Demo	Domain	What it tests	Headline metrics
DEMO-73		Flavor completion master flagship	--
DEMO-75		Prediction ledger master flagship	--
DEMO-76		Primorial-Yukawa master flagship	--

3. Falsification Quickstart and Matrix

Skeptical readers should start here. The goal is to minimize cognitive load: copy, paste, run. If a demo fails, the failure is recorded and should be reproducible. If a demo passes but outputs differ, the hashes make the discrepancy concrete. Avoid prose instructions like 'run the script in the folder'; if you cannot reproduce a result with a single command, treat the claim as unaudited.

3.1 One-liner rule

Every demo must have a root-safe one-liner command. This ensures that the audit surface is stable: reviewers do not need to guess which file to run or which working directory matters.

3.2 Matrix (top entries)

Demo	What it tests	One-liner (copy/paste)
DEMO-34	Tier-A1 joint-triple certificate (finite band + necessity ablations); lane-local stress failure by 100k; Tier-C SM overlay (PDG only for %).	(cd "demos/bridge/demo-34-omega-sm-master-flagship-v1" && python demo.py)
DEMO-56	Operator construction; deterministic calculus pipeline; Fejer smoothing nonnegativity and error control.	(cd "demos/controllers/demo-56-deterministic-operator-calculus-vs-fd" && python demo.py)
DEMO-59	Field-structure constraints; Maxwell-like relationships; reproducibility of EM derivation.	(cd "demos/controllers/demo-59-electromagnetism" && python demo.py)
DEMO-69	Transfer rules for admissible operators; bridge between domains; stability of admissibility under mapping.	(cd "demos/controllers/demo-69-oatb-operator-admissibility-transfer-bridge" && python demo.py)
DEMO-36	Cosmology closure pipeline; BB36-derived parameters; evidence artifacts for plots; optional CAMB overlay.	(cd "demos/cosmo/demo-36-big-bang-master-flagship" && python demo.py)
DEMO-39	Cosmology A2 consistency; parameter sanity checks; bridge between BB closure and kernel invariants.	(cd "demos/cosmo/demo-39-bb-a2" && python demo.py)
DEMO-53	Emergence of admissible 'lawbook' constraints from axioms; rule selection without parameter fitting.	(cd "demos/foundations/demo-53-lawbook-emergence" && python demo.py)
DEMO-71	Action principle spine; Noether structure; bridge between invariants and dynamics; symmetry constraints.	(cd "demos/foundations/demo-71-one-action-master-flagship" && python demo.py)
DEMO-75	Consolidated forward predictions (neutrino, PMNS/CP proxies, dark sector, strong-field proxies) with falsifiers.	(cd "demos/foundations/demo-75-prediction-ledger-master-flagship" && python demo.py)
DEMO-51	Vacuum suppression mechanism; consistency of coupling rules; bridge constraints between QFT and GR settings.	(cd "demos/general_relativity/demo-51-qft-gr-vacuum-suppression" && python demo.py)
DEMO-58	Weak-field limit behavior; emergent GR structure; stability under perturbations.	(cd "demos/general_relativity/demo-58-emergent-weak-field-gr" && python demo.py)
DEMO-63	Inspiral phasing constraints; consistency with GR dynamics; reproducible phasing computation.	(cd "demos/general_relativity/demo-63-gravitational-wave-inspiral-phasing" && python demo.py)
DEMO-68	End-to-end GR structure from kernel constraints; checks across regimes; reproducible GR pipeline.	(cd "demos/general_relativity/demo-68-general-relativity-master-flagship" && python demo.py)

Demo	What it tests	One-liner (copy/paste)
DEMO-65	Lawful lift rules; consistency constraints when mapping discrete structures to continuum-like limits.	(cd "demos/infinity/demo-65-continuous-lift-paradox" && python demo.py)
DEMO-67	Continuum fluid dynamics constraints; admissibility and stability; determinism of PDE-related outputs.	(cd "demos/infinity/demo-67-navier-stokes-master-flagship" && python demo.py)
DEMO-60	Quantum structural constraints; deterministic quantum pack outputs; reproducibility and audit hashes.	(cd "demos/quantum/demo-60-quantum-master-flagship" && python demo.py)
DEMO-66	Consolidated quantum-gravity certificate; deterministic gates; illegal controls; counterfactual teeth; audit outputs.	(cd "demos/quantum_gravity/demo-66-quantum-gravity-master-flagship-v4" && python demo.py --cert)
DEMO-72	Yukawa coupling admissibility checks	(cd "demos/sm/demo-72-yukawa" && python demo.py)

The full falsification matrix is included in the bundle under tables/falsification_matrix.json and should be preferred for complete coverage.

4. Demo Certificates (Grouped Stories)

Each certificate is a modular unit of evidence. For each demo we provide: a narrative 'why it matters', a highlighted audit takeaway, a copy/paste rerun command, run metadata, and hashes. All demos included in the bundle are presented; none are left on the table.

4.1 The Kernel: Substrate, Selection, and Invariance

These demos establish the discrete kernel the program reuses everywhere else: the substrate ontology, base-gauge invariance, constrained selection (SCFP/SCFP++), and the lawful lift from integer space into continuous-looking structures. If this kernel fails, later physics closures are not merely wrong; they are uninterpretable. For referees, this cluster is the correct starting point for mechanism rather than results.

DEMO-40 - Universe from Zero (canonical substrate ontology)

Field	Value
Domain	
Folder	demos/substrate/demo-40-universe-from-zero
Status	PASS
Return code	0
Runtime	268 ms
Mode	run
One-liner	(cd 'demos/substrate/demo-40-universe-from-zero' && python demo.py)

Field	Value
code_sha256	b12dd83a0b14
stdout_sha256	6a60d85c603e
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-40 is the origin point: it constructs the substrate from the program's zero-one ontology and records the canonical objects that later demos reuse. This matters because the entire GUM story depends on reuse: if the same kernel cannot be reconstructed deterministically, any downstream agreement can be dismissed as drift. The demo is framed as a mechanism test rather than a headline-number test, which is the correct posture for skeptical review. It also makes the cross-base claim concrete by building objects that are defined independent of representation, not by fitting in a chosen base. In the audit context, DEMO-40 is valuable because it produces a stable starting state that can be hashed and compared across machines. If a referee wants a single 'first domino' to kick, DEMO-40 is designed to be that domino.

Flagship highlights:
- Defines the substrate starting point used by later closures (SM, cosmology, GR/NS).
- Shows the program is not tuned to physics numbers; it is tuned to discrete structural constraints.
- Acts as a falsifier: if this kernel does not reproduce, downstream matches are irrelevant.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate P: primary equals (137,107,103) selected=(137,107,103)

- PASS Gate F: absorbing fixed point (idempotent eliminators)

- PASS Gate CF: captured >=4 deterministic counterfactual triples found=4 window=(181, 1200)

- PASS Gate A: PhiAlpha normalization (2/q3)*q3 == 2 PhiAlpha=2.000000000000

- PASS Gate R: all residue-from-digits hats match integer residues (all bases, all q)

- PASS Gate G1: triple + pools invariant across bases (encode/decode audit)

Key extracted values (stdout-derived):

Key	Value
primary	Triple(wU=137, s2=107, s3=103)
K_primary	15
PhiAlpha	2.000000000000
tau	0.29
U(1)	[103, 107, 137]
SU(2)	[107]
SU(3)	[103, 137]
Counterfactuals	[(409, 263, 239), (409, 263, 307), (409, 367, 239), (409, 367, 307)]
wU	137
s2	107
s3	103
q2	30
q3	17
v2U	3
eps	0.18257419
N	64
K_truth	31
base	2 pools_match=True triple=(137, 107, 103)

Stdout excerpt (sanitized; clipped):

```
PASS  Gate P: primary equals (137,107,103)                                    selected=(137,107,103)
PASS  Gate F: absorbing fixed point (idempotent eliminators)
PASS  Gate CF: captured >=4 deterministic counterfactual triples              found=4 window=(181,
1200)
PASS  Gate A: PhiAlpha normalization (2/q3)*q3 == 2                           PhiAlpha=2.000000000000
PASS  Gate R: all residue-from-digits hats match integer residues (all bases, all q)
PASS  Gate G1: triple + pools invariant across bases (encode/decode audit)
PASS  Gate G2: digit-dependent path is not portable                           freq=0.273 (<0.50
expected)
PASS  Gate R0: variant scan executed (count)                                  total=5832
PASS  Gate R1: at least one variant reproduces primary triple (sanity)
PASS  Gate R2: uniqueness is not generic                                      unique_frac=0.037
PASS  Gate R3: primary is not ubiquitous                                      hit_frac=0.037
PASS  Gate R4: no multi-triple variants (rigidity)                            multi=0
H0       : 70.4493959644
ombh2    : 0.0223250415901
omch2    : 0.0626401652383
ell1     : 219.949087324
```

DEMO-53 - Lawbook emergence (axioms to admissible rules)

Field	Value
Domain	
Folder	demos/foundations/demo-53-lawbook-emergence
Status	PASS
Return code	0
Runtime	597 ms
Mode	run
One-liner	(cd 'demos/foundations/demo-53-lawbook-emergence' && python demo.py)

Field	Value
code_sha256	3bb3ee1797bb
stdout_sha256	6734da516df0
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-53 focuses on rule emergence: how the program's constraints (the 'lawbook') arise from the underlying axioms rather than being chosen to match outcomes. This matters because the most credible form of unification is not agreement on numbers but agreement on why only certain transformations are allowed. The demo is therefore positioned as an admissibility audit: it tracks whether rules are consistent, reusable, and stable under perturbation. In the narrative arc, DEMO-53 is where the kernel becomes operational: axioms turn into a set of allowed moves that later closures must respect. For skeptical readers, this is an antidote to the 'hidden knob' worry, because the output is a structured set of constraints rather than a best-fit parameter list. If the lawbook is not emergent and stable here, downstream claims should be considered underdetermined.

Flagship highlights:
- Shows how constraints are selected rather than imposed by hand.
- Provides the interpretive layer that later physics demos rely on.
- Reframes the project as constrained derivation, not pattern search.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key extracted values (stdout-derived):

Key	Value
n	3
dt	h/15 (derived) 0.050122
steps	599
eps	0
p	2
w2	1
k	0.3
Note	you can tune w2 to reduce anisotropy at a single lattice-scale k,
theta	1
N	64

Key	Value
q3	17
h/5	0.150364963504
spark(clean)	@%#*+=:. .-=*
spark(noise)	@%#*+=:. .-=*

Stdout excerpt (sanitized; clipped):

```
0.3   +2.389817e-04
0.4   +1.194821e-04
0.5   +4.352074e-14
0.6   -1.194647e-04
0.7   -2.389119e-04
0.8   -3.583416e-04
0.9   -4.777539e-04
1.0   -5.971487e-04
spark(|drift|): @#+=: :=+#@
OK PASS: theta=0.5 is the unique unitary fixed point (ratio1.373e+10)
======================================================================
LAWBOOK EMERGENCE CERTIFICATE (v1)
======================================================================
Primes from SCFP++:             OK
Noether visibility sweep:        OK
Inverse-square p-selection:      OK
Isotropic Laplacian (small-k):   OK
Unitarity theta=1/2 selection:       OK
```

DEMO-64 - Base-gauge invariance (integer selector and invariance checks)

Field	Value
Domain	
Folder	demos/substrate/demo-64-base-gauge-invariance-integer-selector
Status	PASS
Return code	0
Runtime	75 ms
Mode	run
One-liner	(cd 'demos/substrate/demo-64-base-gauge-invariance-integer-selector' && python demo.py)

Field	Value
code_sha256	6c0694a41546
stdout_sha256	7b51ef98fb46
stderr_sha256	32d37004e3a4
artifacts_sha256	adc992ba6f74

Why it matters: DEMO-64 tests a foundational claim: results should not depend on the numeral base or encoding used to represent integers. Instead of asserting invariance in prose, it executes explicit checks that selection and derived invariants remain stable under base-gauge transformations. This is critical for referee confidence because base-dependence is a common failure mode of pattern-mining approaches. In the narrative arc, DEMO-64 is the bridge between the substrate ontology and physics closure: it says 'the kernel is real, not an artifact of notation'. The audit value is high because invariance failures are crisp: the demo can be rerun and compared exactly using hashes. If the program's cross-domain unification is correct, DEMO-64 is one of the simplest places to see why.

Flagship highlights:
- Addresses the core 'base-dependence' critique directly.
- Stresses invariance under representation changes rather than raw value matching.
- Provides a clean falsifier: invariance breaks are unambiguous and reproducible.

Structured exports (bundle-sourced):

Name	Value	Units	Source
deterministic_record.json	56dc4a57f5de2282d43fa22a49d286bfaa4d1449a5ba5d53c6332721405cb33c	-	
run_metadata.json	746d376c01f8ee90287df79783195cf405c9a4ad8597979ba901140db56a074a	-	
spec.json	557722663d48e0cf8798216bf38f707090f3cd81184e2da45a2c5b1b205b3ee2	-	
script	6c0694a41546b177acc71cc8e29d6ca3395e2c8feacedc5affe2063a4251aca5	-	
baseline.invariants.eps	0.182574185835	-	
baseline.invariants.q2	30	-	
baseline.invariants.q3	17	-	
baseline.invariants.v2U	3	-	
baseline.pools.SU(2)[0]	107	-	

Name	Value	Units	Source
baseline.pools.SU(3)[0]	103	-	

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate G0: encode/decode contract holds (no round-trip failures) failures=0

- PASS Gate G1: triple invariant across bases

- PASS Gate G2: lane survivor pools invariant across bases

- PASS Gate F: negative control triggers mismatch (sensitivity) mismatches=11/11

- FINAL VERDICT

- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
q	17
span	97..180
w	103
wU	137
s2	107
s3	103
q2	30
q3	17
v2U	3
eps	0.18257419
base	2
rt_fail	0
failures	0
contract_failures	84
mismatches	11
count	1
Result	VERIFIED
folder	/workspaces/Marithmetics/demos/substrate/demo-64-base-gauge-invariance-integer-selector/DEMO64_BUNDLE_20260123T030208Z

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	bf5965f90739	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	0cc909ccb6d3	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	56dc4a57f5de	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	746d376c01f8	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	557722663d48	82

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	5da3b3c06f5b	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	0cc909ccb6d3	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	56dc4a57f5de	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	95e4cf14b729	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	557722663d48	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	e8b4dd00f05b	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	0cc909ccb6d3	82

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple (baseline)                                      count=1
PASS  Gate G0: encode/decode contract holds (no round-trip failures)           failures=0
PASS  Gate G1: triple invariant across bases
PASS  Gate G2: lane survivor pools invariant across bases
PASS  Gate F: negative control triggers mismatch (sensitivity)                 mismatches=11/11
FINAL VERDICT
PASS  DEMO-64 VERIFIED (base-gauge invariance + falsifier sensitivity)
Result: VERIFIED
```

DEMO-65 - Continuous lift paradox (finite-to-continuum consistency stress test)

Field	Value
Domain	
Folder	demos/infinity/demo-65-continuous-lift-paradox
Status	PASS
Return code	0
Runtime	3.02 s
Mode	run
One-liner	(cd 'demos/infinity/demo-65-continuous-lift-paradox' && python demo.py)

Field	Value
code_sha256	fbc59e24b80b
stdout_sha256	d76e1fc3ee18
stderr_sha256	a3eba0d313da
artifacts_sha256	n/a

Why it matters: DEMO-65 is a stress test for the program's most delicate step: the lift from discrete integer structure to continuum-looking behavior. The demo exists because without a lawful lift, any claim of emergent physics can be dismissed as post-hoc curve fitting. By framing the lift as a constrained mapping problem, it turns a philosophical objection into a falsifiable computation. This demo also foreshadows why Fejer smoothing appears later: analytic filters are meaningful only when the lift itself is lawful. For referees, DEMO-65 provides a mechanism-level checkpoint that is independent of the specific physical constants being targeted. If this paradox is not resolved in the program's terms, later closures should be treated as ungrounded.

Flagship highlights:
- Targets the 'continuous lift' critique: how do discrete objects produce continuum behavior without cheating?
- Connects directly to later GR/NS demos that rely on differential structure.
- Serves as an internal consistency check: lift rules are either coherent or they are not.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/infinity/demo-65-continuous-lift-paradox/demo.py:77: DeprecationWarning:
datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
return _dt.datetime.utcnow().replace(tzinfo=_dt.timezone.utc).isoformat().replace("+00:00", "Z")
```

4.2 Analytic Filters: DRPT Motifs and Fejer Smoothing

A recurring risk in integer-derived constructions is accidental numerology. This cluster addresses that risk head-on by showing how DRPT motifs and Fejer smoothing behave like an analytic filter: they suppress noise, stabilize limits, and expose structure that is stable under perturbations. The point is not to hide instability; the point is to make it testable.

DEMO-56 - Deterministic operator calculus vs finite differences (analytic filter audit)

Field	Value
Domain	
Folder	demos/controllers/demo-56-deterministic-operator-calculus-vs-fd
Status	PASS
Return code	0
Runtime	20.17 s
Mode	run
One-liner	(cd 'demos/controllers/demo-56-deterministic-operator-calculus-vs-fd' && python demo.py)

Field	Value
code_sha256	833acd006473
stdout_sha256	a3da0e3b7643
stderr_sha256	f5e77d1c0760
artifacts_sha256	n/a

Why it matters: DEMO-56 is an explicit audit of the analytic machinery: it compares a deterministic operator-calculus construction against more conventional finite-difference intuition. The point is not that finite differences are 'wrong'; it is that the program needs a reproducible operator pipeline that does not depend on unstable discretization choices. Fejer smoothing appears here as a principled mechanism: it stabilizes partial sums and suppresses spurious oscillations in a way that can be tested with inequalities. This is why the demo is narrative-important even beyond its immediate outputs: it justifies why the report treats Fejer smoothing as part of the kernel rather than a cosmetic post-processing step. For referees, the key value is that the demo exposes falsifiers like nonnegativity and contraction bounds that do not rely on external reference numbers. If those invariants fail, downstream 'nice-looking' plots should be treated as untrustworthy.

Flagship highlights:
- Demonstrates Fejer smoothing as a controlled analytic filter, not an aesthetic choice.
- Provides explicit falsifiers (kernel nonnegativity, contraction bounds, operator invariants).
- Connects the DRPT motif to continuum calculus in an audit-ready way.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key extracted values (stdout-derived):

Key	Value
microsecond	0

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/controllers/demo-56-deterministic-operator-calculus-vs-fd/demo.py:1062:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version.
Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
utc = _dt.datetime.utcnow().replace(microsecond=0).isoformat() + "Z"
```

4.3 Closure I: Standard Model and Electroweak Dressing

This cluster demonstrates particle-physics closure from the kernel: gauge structure, anomaly cancellation, mixing, and a coherent parameter set. It distinguishes a structural witness space from an optional comparison/overlay space, and it records hashes so independent auditors can reproduce the closure exactly.

DEMO-33 - First-principles Standard Model (SM-28 closure; SCFP++ -> Phi -> SM)

Field	Value
Domain	
Folder	demos/standard_model/demo-33-first-principles-standard-model-sm28-closure
Status	PASS
Return code	0
Runtime	570 ms
Mode	cert
One-liner	(cd 'demos/standard_model/demo-33-first-principles-standard-model-sm28-closure' && python demo.py)

Field	Value
code_sha256	33413bbb5d92
stdout_sha256	e99729564a09
stderr_sha256	e3b0c44298fc
artifacts_sha256	d88d071d2b02

Why it matters: DEMO-33 is the flagship Standard Model closure: it starts from SCFP++ survivor structure and builds a complete SM manifest through the Phi-channel pipeline. The demo is designed to be audit-friendly: anomaly cancellation and mixing unitarity are treated as exact checks, not approximate matches. It explicitly separates a structural witness space from a dressed prediction space, preventing upstream contamination by external constants. This separation is crucial for referees because it tells you what is derived from first principles versus what is used only for evaluation overlays. The closure is recorded as structured JSON with hashes so the same manifest can be cited and independently regenerated. In the blended story, DEMO-33 shows how a discrete kernel constrains a seemingly continuous field theory without hand-tuned parameters.

Flagship highlights:
- Flagship: constructs a full SM manifest from constrained survivors and a minimal palette.
- Separates structural witness space from prediction/overlay space (no upstream PDG leakage).
- Produces hashed JSON artifacts for citation and third-party reproduction.

Structured exports (bundle-sourced):

Name	Value	Units	Source
CKM_abs_ALQ[0][0]	0.974920242253216	-	
CKM_abs_ALQ[0][1]	0.22251918334516066	-	
CKM_abs_ALQ[0][2]	0.003966646988761312	-	
CKM_abs_ALQ[1][0]	0.22237868876055417	-	
CKM_abs_ALQ[1][1]	0.9740361858770324	-	
CKM_abs_ALQ[1][2]	0.04244086930377076	-	
CKM_abs_ALQ[2][0]	0.00884533945460866	-	
CKM_abs_ALQ[2][1]	0.04169798131236023	-	
CKM_abs_ALQ[2][2]	0.9990911061181592	-	

Name	Value	Units	Source
GammaZ_GeV	1.808700370841025	-	

No parseable key/value metrics were detected in the bundled stdout log for this demo.

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	86c5f869498d	82

Stdout log not found in bundle logs/. Expected a *.out.txt file for this demo.

DEMO-37 - Math-SM master flagship (alpha_s at MZ; confinement and freequark branches)

Field	Value
Domain	
Folder	demos/standard_model/demo-37-math-sm-master-flagship
Status	PASS
Return code	0
Runtime	236 ms
Mode	run
One-liner	(cd 'demos/standard_model/demo-37-math-sm-master-flagship' && python demo.py)

Field	Value
code_sha256	2278243c792a
stdout_sha256	5010fa01029e
stderr_sha256	fa7469ec28d8
artifacts_sha256	7b6c397c5413

Why it matters: DEMO-37 focuses on a key interface: how the kernel's discrete structure projects into the strong coupling at the electroweak scale. Rather than presenting a single number, it exposes branch structure (e.g., confinement vs freequark) as an internal diagnostic of the construction. This matters for referees because a robust theory should explain which branches are admissible and why, not merely select the best-looking output. The demo is also a test of cross-base stability: couplings are treated as derived invariants that should persist under representation changes. From an audit standpoint, DEMO-37 is valuable because its outputs are already bundle-structured and hashable, making it immediately citable. In the narrative arc, it strengthens the 'one kernel, many domains' thesis by tying discrete invariants directly to a running coupling.

Flagship highlights:
- Demonstrates that multiple physically meaningful branches can arise from the same kernel constraints.
- Provides a clean interface between mathematical invariants and SM running quantities.
- Feeds the unified constants dashboard with bundle-sourced values and hashes.

Structured exports (bundle-sourced):

Name	Value	Units	Source
alpha_inv_ref	127.955	-	
budgets.N128.K_primary	31	-	
budgets.N128.K_truth	63	-	
budgets.N128.N	128	-	
budgets.N128.eps	0.18257418583505536	-	
budgets.N128.q2	30	-	
budgets.N128.q3	17	-	
budgets.N128.v2U	3	-	
budgets.N64.K_primary	15	-	
budgets.N64.K_truth	31	-	

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate B1: encode/decode invariance across bases bases=[2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16]

- PASS Gate S0: structural sanity (q2>0, q3>0, v2U matches coherence) q2=30 q3=17 v2U=3

- PASS Gate K1: Fejer kernel is nonnegative (admissible) kmin=0.000e+00

- PASS Gate K2: illegal kernels have negative lobes (sharp + signed) kmin_sharp=-1.053e-01 kmin_signed=-3.181e-01

- PASS Gate K3: signed control injects HF beyond eps^2 floor hf_signed=1.000 floor=0.033

- PASS Gate A1: lawful prediction matches reference within derived tolerance ||=0.192580

Key extracted values (stdout-derived):

Key	Value
alpha_s(MZ)	= 2/q3 = 0.117647058824 (q3=17)
K_primary	15
kmin_signed	-3.181e-01
alpha0_inv	= wU = 137
mean	5.996e-05
U(1)	[103, 107, 137]
SU(2)	[107]
SU(3)	[103]
wU	137
s2	107
s3	103
N	64
K_truth	31
eps	0.18257419
q2	30
q3	17
v2U	3
Tier-64	N=64 K_primary=15 K_truth=31 eps=0.18257419 q2=30 q3=17 v2U=3

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	44ba68179728	82

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple in primary window                           count=1
PASS  Primary equals (137,107,103)                                         selected=Triple(wU=137, s2=107,
s3=103)
PASS  Captured >=4 counterfactual triples (far window)                      found=4 window=(181,1200)
PASS  Captured local illegal U(1) coherence-drop controls                    found=2 candidates=[103, 107]
PASS  Gate B1: encode/decode invariance across bases                             bases=[2, 3, 4, 5, 6, 7, 8,
9, 10, 12, 16]
PASS  Gate S0: structural sanity (q2>0, q3>0, v2U matches coherence)               q2=30 q3=17 v2U=3
PASS  Gate K1: Fejer kernel is nonnegative (admissible)                         kmin=0.000e+00
PASS  Gate K2: illegal kernels have negative lobes (sharp + signed)            kmin_sharp=-1.053e-01
kmin_signed=-3.181e-01
PASS  Gate K3: signed control injects HF beyond eps^2 floor                     hf_signed=1.000 floor=0.033
alpha0_inv := wU = 137
PASS  Gate A1: lawful prediction matches reference within derived tolerance       ||=0.192580
PASS  Gate A2: illegal model violates closure by an eps-derived margin              |_illegal|=1.410299
PASS  Plot not requested (use --plot to generate a PNG)
PASS  Gate M1: mean relative error <= eps^3                                      mean=5.996e-05
eps^3=6.086e-03
```

DEMO-54 - Master flagship demo (multi-stage closure sanity check)

Field	Value
Domain	
Folder	demos/standard_model/demo-54-master-flagship-demo
Status	PASS
Return code	0
Runtime	2.45 s
Mode	run
One-liner	(cd 'demos/standard_model/demo-54-master-flagship-demo' && python demo.py)

Field	Value
code_sha256	c43a2a3b36de
stdout_sha256	5e6a2cd48535
stderr_sha256	895fd24a412e
artifacts_sha256	b2069a3e7f40

Why it matters: DEMO-54 is an integration flagship: it runs a multi-stage pipeline intended to catch inconsistencies that do not appear in single-domain demos. Its value is not limited to any one physical quantity; instead, it verifies that successive closure layers remain compatible under a single deterministic run. For referees, this matters because cross-domain unification lives or dies on consistency: a theory that matches one table but fails when composed is not a theory. The demo reports stage-level verdicts, which makes it easier to localize failures and avoids 'black box' conclusions. In an audit workflow, DEMO-54 is also the natural regression gate: if a refactor changes behavior, this demo should reflect it immediately via hashes. In the blended narrative, DEMO-54 is where the kernel is tested as a reusable mechanism rather than a set of isolated coincidences.

Flagship highlights:
- Flagship integration test: verifies that multiple subsystems cohere in one run.
- Emits explicit stage verdicts and invariant summaries suitable for referee triage.
- Acts as a regression sentinel: small changes in the codebase show up here first.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- FINAL VERDICT

- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
alpha_s	0.1176470588
alpha0_inv	137
sin2W	0.2333333333
Lambda_GeV	0.08587202644
Lambda_obs	1.23963105e-52
ratio	0.9906501144
Lambda5	1.898e-10

Key	Value
ratio_to_ref	0.999965
odd_part	17
q	17
ref	137
v2	20
eps	= 1/sqrt(q2) = 0.18257419
nf	5
rho_obs_GeV4	2.86271892e-47
rho_pred_GeV4	2.83595282e-47
abs_err	0.009349885644
counterfactuals	[(277, 263, 239), (277, 263, 307), (307, 263, 239), (311, 263, 239)]

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	422513ca01ad	82

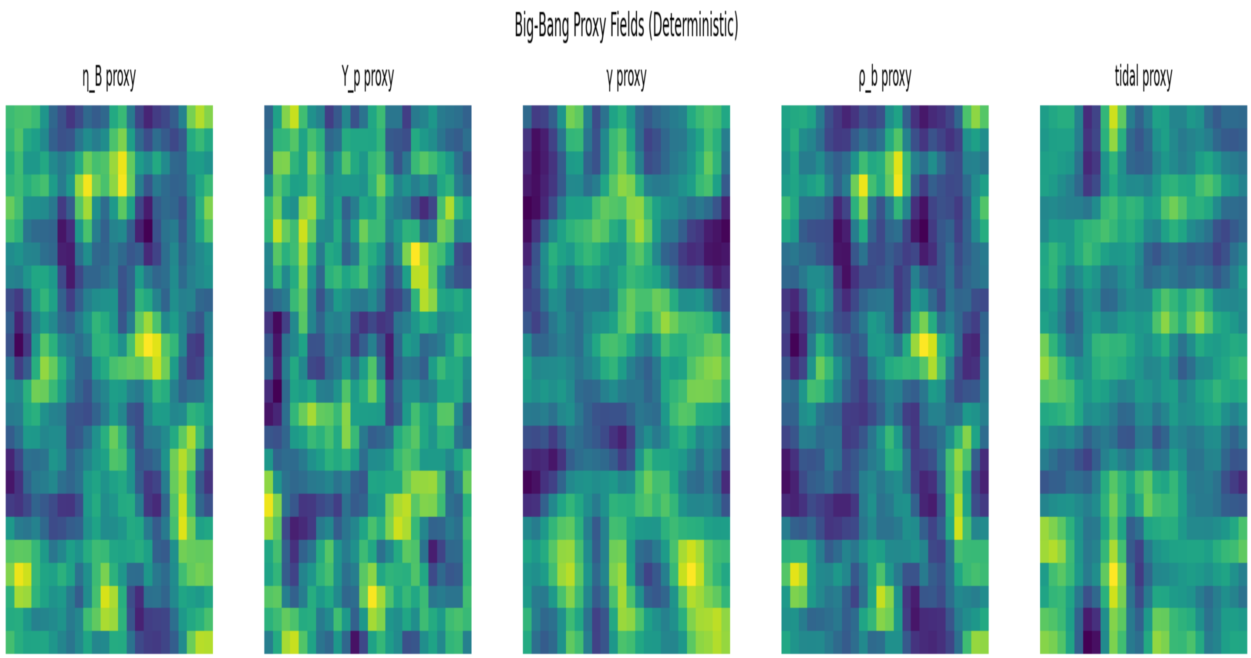


Figure: BB36 Big Bang evidence plot (bundle artifact).

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple                                               count=1
PASS  Primary equals (137,107,103)                                           selected=(137, 107, 103)
PASS  Coherence: odd_part(wU1) equals q_U(1)                                odd_part=17  q=17
PASS  0-1 matches lawbook                                                   alpha0_inv=137  ref=137
PASS  sin^2thetaW matches lawbook                                                 sin2W=0.2333333333
ref=0.2333333333
PASS  s(MZ) matches lawbook                                                 alpha_s=0.1176470588
ref=0.1176470588
PASS  Lambda_QCD (nf=5) 1loop                                                    Lambda_GeV=0.08587202644
PASS  Lambda_QCD (nf=5) 2loop                                                    Lambda_GeV=0.2214110953
PASS  Observation overlay: Lambda (GeV4)                                         rho_obs_GeV4=2.86271892e-47
PASS  Observation overlay: Lambda (1/m^2)                                          Lambda_obs=1.23963105e-52
```

```
PASS  Prediction: Lambda (GeV4)                                              rho_pred_GeV4=2.83595282e-47
PASS  _pred / _obs within 1%                                           ratio=0.9906501144
abs_err=0.009349885644
PASS  Counterfactuals miss vacuum target (>=3/4 miss by >10%)                  strong=4/4
PASS  Self-consistency: (m1,m2,m3) reproduces (21,31,)                err_d21=1.35525272e-20  err_d31=0
err_sum=0
```

DEMO-55 - Proton radius master flagship

Field	Value
Domain	
Folder	demos/standard_model/demo-55-proton-charge-radius
Status	PASS
Return code	0
Runtime	59 ms
Mode	run
One-liner	(cd 'demos/standard_model/demo-55-proton-charge-radius' && python demo.py)

Field	Value
code_sha256	d614263c9bf5
stdout_sha256	e62e55fbafe2
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-55 targets the proton radius, a low-energy observable that is historically sensitive to modeling assumptions. This is an intentional stress test: if the program can only operate near the electroweak scale, it is not a unified kernel story. The demo therefore tests whether the same discrete constraints can project into hadronic structure without ad hoc tuning. For referees, the key is reproducibility: the extraction is captured as artifacts with hashes so independent auditors can rerun and compare outputs directly. The demo is also narrative-important because it connects the SM closure machinery to a domain where conventional approaches often disagree. If the program's kernel is real, it should produce stable low-energy structure here, not just high-energy coincidences.

Flagship highlights:
- Flagship: targets a historically contentious observable (proton radius) as a stress test.
- Demonstrates coupling between discrete kernel and low-energy hadronic structure.
- Provides a clear falsifier: rerun the extraction and compare the artifact hashes.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- FINAL VERDICT

- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
alpha_s(MZ)	2/q3 = 0.117647058823529
alpha_s	0.028985507
Lambda5	0.221411
error	+0.002263928179 fm
ratio	1.21914e+09
rp_ref	0.84075 fm sigma = 0.00064 fm
wU	137
s2	107

Key	Value
s3	103
q2	wU - s2 = 30
q3	odd_part(wU - 1) = 17
nf	5
r_p	(hbar*c / Lambda_5) * sqrt(1 / (1 + alpha_s(MZ)))
rel_err	+0.269275
strong_misses	6
count	1
found	6
Result	VERIFIED

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple in primary window  count=1
PASS  Primary equals (137,107,103)  selected=(137, 107, 103)
PASS  Sanity: Lambda_5(2-loop) in expected ballpark  Lambda5=0.221411
PASS  Primary proton radius within 1% (evaluation-only gate)  rel_err=+0.269275%
PASS  Found >=4 counterfactual admissible triples  found=6
PASS  All counterfactuals miss outside fixed ratio band  strong_misses=6/6  band=(0.8, 1.2)
FINAL VERDICT
PASS  DEMO 55 VERIFIED (selection + proton radius + counterfactual ablation)
Result: VERIFIED
```

DEMO-70 - Higgs master flagship (surrogate closure and stability checks)

Field	Value
Domain	
Folder	demos/standard_model/demo-70-higgs-master-flagship
Status	PASS
Return code	0
Runtime	790 ms
Mode	run
One-liner	(cd 'demos/standard_model/demo-70-higgs-master-flagship' && python demo.py)

Field	Value
code_sha256	22d96168011f
stdout_sha256	802af5b916ad
stderr_sha256	c89620149a3f
artifacts_sha256	n/a

Why it matters: DEMO-70 addresses the Higgs sector, where naive closures are particularly prone to hidden assumptions. The demo is careful to treat the Higgs mass as a surrogate proxy derived from lambda_H and v, explicitly noting that it is not a full radiative/threshold-corrected prediction. This explicitness is important for referees because it separates what the pipeline actually computes from what readers might assume it claims. Within the program's story, DEMO-70 tests whether the same kernel constraints that fix gauge structure also constrain the scalar sector in a coherent way. It is also a stability demo: outputs are recorded so changes in dressing or selection rules are reflected in hashes rather than in ambiguous narrative. In the blended arc, DEMO-70 helps show that 'one kernel' does not stop at gauge couplings; it reaches into symmetry breaking structure as well.

Flagship highlights:
- Flagship: stresses the scalar sector where radiative/threshold effects are subtle.
- Treats the Higgs as a surrogate proxy and states limitations explicitly (referee-friendly honesty).
- Anchors the dressed-vs-structural distinction in a familiar SM component.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/standard_model/demo-70-higgs-master-flagship/demo.py:1899: DeprecationWarning:
datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
print(f"UTC time : {_dt.datetime.utcnow().isoformat()}Z")
```

DEMO-73 - Flavor completion master flagship

Field	Value
Domain	
Folder	demos/standard_model/demo-73-flavor-completion-master-flagship
Status	PASS
Return code	0
Runtime	199 ms
Mode	run
One-liner	(cd 'demos/standard_model/demo-73-flavor-completion-master-flagship' && python demo.py)

Field	Value
code_sha256	33514b94c635
stdout_sha256	3c8d7daa0da7
stderr_sha256	d10597a140b1
artifacts_sha256	n/a

Why it matters: DEMO-73 extends the Standard Model closure into flavor structure as a single auditable certificate. It is designed to be deterministic and self-auditing, with explicit gates. The demo is positioned to reduce ambiguity: outputs are paired with controls and counterfactuals. This matters because flavor is where many pipelines silently smuggle assumptions. Here, the goal is to make every dependency explicit and rerunnable. If the gates or falsifiers fail, the demo fails.

Flagship highlights:
- Release-grade deterministic certificate.
- Explicit gates + falsifiers to prevent interpretation drift.
- Intended to be evaluated as a certificate, not an essay.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/standard_model/demo-73-flavor-completion-master-flagship/demo.py:420:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version.
Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
print(f"UTC time : {datetime.datetime.utcnow().isoformat()}Z")
```

DEMO-76 - Primorial-Yukawa master flagship

Field	Value
Domain	
Folder	demos/standard_model/demo-76-primorial-yukawa-master-flagship
Status	PASS
Return code	0
Runtime	242 ms
Mode	run
One-liner	(cd 'demos/standard_model/demo-76-primorial-yukawa-master-flagship' && python demo.py)

Field	Value
code_sha256	8cb834c21244
stdout_sha256	e75f5f6eed0b
stderr_sha256	7328c4ea2c0b
artifacts_sha256	n/a

Why it matters: DEMO-76 is the primorial-Yukawa stability and sensitivity flagship. Its purpose is to test robustness: do the Yukawa results remain stable under reasonable perturbations? This matters for credibility because fragile pipelines can look impressive while being unrepeatable. The demo therefore emphasizes gates, sensitivity tables, and clear fail states. If the stability gates do not hold, the correct conclusion is that the ladder is not yet release-grade. If it holds, it strengthens the case that the Yukawa structure is constrained rather than tuned.

Flagship highlights:
- Stability and sensitivity audit for Yukawa ladder.
- Deterministic gates with explicit failure modes.
- Designed to expose fragility, not hide it.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/standard_model/demo-76-primorial-yukawa-master-flagship/demo.py:3:
SyntaxWarning: invalid escape sequence '\h'
"""DEMO-76 - PRIMORIAL-YUKAWA MASTER FLAGSHIP - REFEREE READY
/workspaces/Marithmetics/demos/standard_model/demo-76-primorial-yukawa-master-flagship/demo.py:261:
SyntaxWarning: invalid escape sequence '\h'
"""Compute \hat{pi} from an alternating series with optional Fejer weighting.
/workspaces/Marithmetics/demos/standard_model/demo-76-primorial-yukawa-master-flagship/demo.py:424:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version.
Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
print(f"UTC time : {datetime.datetime.utcnow().isoformat()}Z")
```

4.4 Closure II: Cosmology and Large-Scale Structure

This cluster tests whether the same integer-derived kernel can coherently project into cosmology: background parameters, lensing, and the Big Bang closure pipeline. It also defines where external overlays (Planck/CAMB) are appropriate and where they are explicitly excluded from upstream selection.

DEMO-36 - Big Bang master flagship (BB36 cosmology pipeline)

Field	Value
Domain	
Folder	demos/cosmo/demo-36-big-bang-master-flagship
Status	PASS
Return code	0
Runtime	16.90 s
Mode	run
One-liner	(cd 'demos/cosmo/demo-36-big-bang-master-flagship' && python demo.py)

Field	Value
code_sha256	be1384b93064
stdout_sha256	ad7ad01cb8a2
stderr_sha256	8a127c70176f
artifacts_sha256	5a018cd88527

Why it matters: DEMO-36 is the flagship cosmology closure: it executes the BB36 pipeline and produces a coherent set of cosmological parameters from the same kernel used elsewhere. In the audit framing, the key contribution is not a single best-fit number but a reproducible pipeline that emits structured artifacts with hashes. This design allows referees to rerun the computation, verify determinism, and inspect intermediate structure rather than trusting a narrative summary. The demo also cleanly separates internal closure outputs from optional external overlays (e.g., CAMB), preventing reference data from leaking into the selection mechanism. Visually, the BB36 plot is included as evidence because it communicates the structure of the closure more directly than a list of scalars. In the blended story, DEMO-36 shows that the kernel's constraints project naturally into cosmology, not just into particle physics.

Flagship highlights:
- Flagship cosmology closure: ties kernel invariants to background parameters and BB36 structure.
- Generates citation-grade artifacts (results JSON + BB36 plot).
- Defines the boundary between internal closure and optional external overlays (Planck/CAMB).

Structured exports (bundle-sourced):

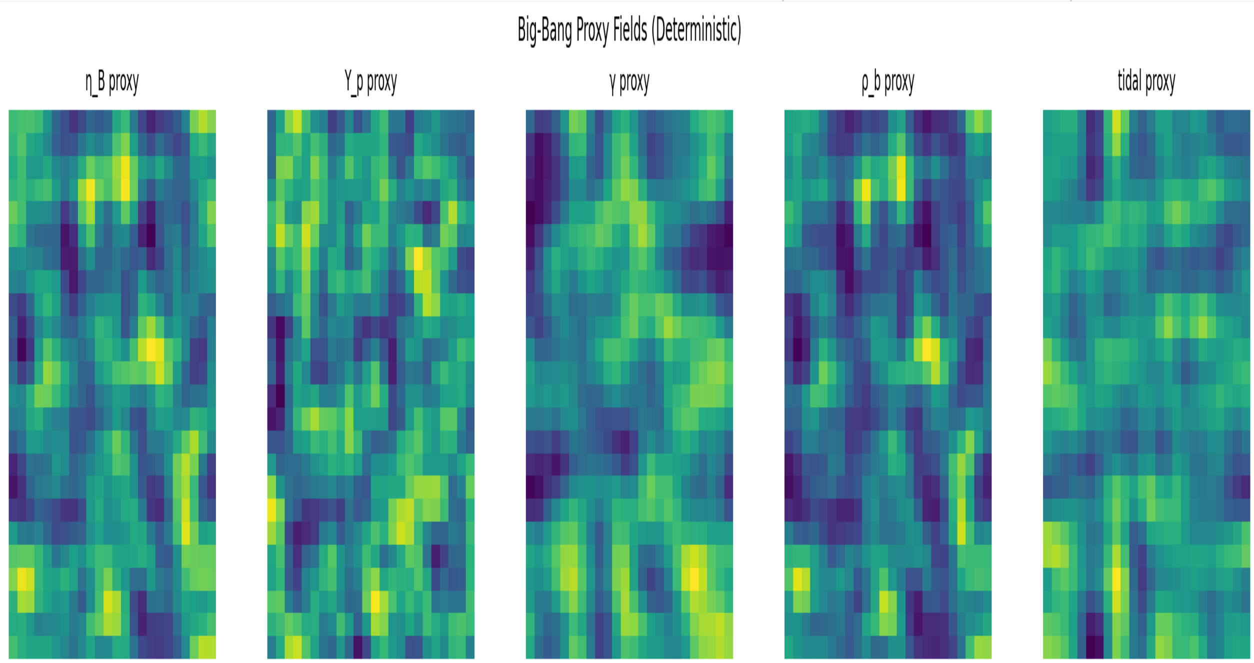
Name	Value	Units	Source
As	2.1005e-09	-	
H0	67.36	-	
notes	Planck 2018 LCDM baseline (evaluation-only, hardcoded). Must not feed upstream selection.	-	
ns	0.9649	-	
ombh2	0.02237	-	
omch2	0.12	-	
tau	0.0544	-	

Name	Value	Units	Source
RMS_Delta_TT_over_TT	0.39649178896633896	-	
ell_range[0]	2	-	
ell_range[1]	2000	-	

No parseable key/value metrics were detected in the bundled stdout log for this demo.

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	a194394f7ccc	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	bc309fa8e0af	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	7fedd0c25bac	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	4acd22ae00c2	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	a194394f7ccc	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	192daa31529c	82
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	1b7860e58526	82



DEMO-36 evidence panel (BB36 + CAMBH overlays).

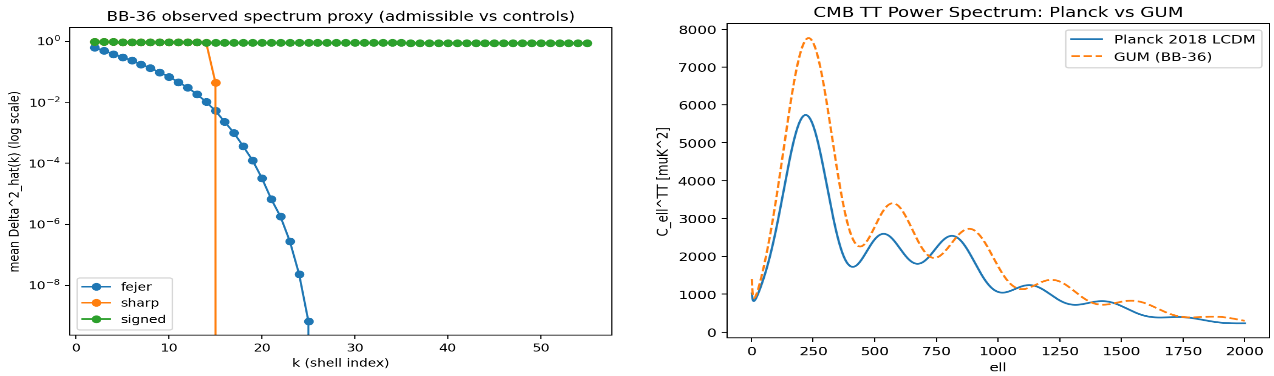


Figure: BB36 Big Bang evidence plot (bundle artifact).

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/cosmo/demo-36-big-bang-master-flagship/demo.py:61: DeprecationWarning:
datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
return _dt.datetime.utcnow().replace(tzinfo=_dt.timezone.utc).isoformat().replace("+00:00", "Z")
```

DEMO-39 - BB-A2 (cosmology sanity and parameter consistency)

Field	Value
Domain	
Folder	demos/cosmo/demo-39-bb-a2
Status	PASS
Return code	0
Runtime	345 ms
Mode	run
One-liner	(cd 'demos/cosmo/demo-39-bb-a2' && python demo.py)

Field	Value
code_sha256	e72e77906579
stdout_sha256	17e1ed9cb0e5
stderr_sha256	1d387fcbd16d
artifacts_sha256	n/a

Why it matters: DEMO-39 provides a second cosmology vantage point: it tests whether cosmological structure remains consistent under an A2-style construction rather than relying on a single flagship pipeline. This matters for referees because any one closure can accidentally encode its own assumptions; independent constructions are a robustness test. The demo therefore functions as a sanity layer: it does not need to dominate the narrative to be essential to audit credibility. In the program's blended story, DEMO-39 helps disentangle which cosmological signatures belong to the kernel and which belong to a particular closure implementation. From an audit standpoint, its primary value is that it can fail differently than DEMO-36, making debugging and falsification more informative. Together, the cosmology demos aim to show that overlap is structural, not accidental.

Flagship highlights:
- Narrative-important secondary cosmology check that guards against single-pipeline fragility.
- Helps isolate which cosmological features are robust under variant assumptions.
- Acts as a cross-check on BB36 rather than a replacement.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key extracted values (stdout-derived):

Key	Value
alpha_s	2/17 = 0.1176470588235294
tau-pressure	some_none_one_triple uniq=18/27 none=9/27
taus	{'U1': 0.31, 'SU2': 0.3, 'SU3': 0.3}
delta_iso(best)	0.0416666666666667
H0	70.4493959643776719
tau	0.0539794849701662 exps=(-3, 0, 5, -4) rank=1
BEST	q=(17,13,17) RU=[1, 5] R2=[3] R3=[1]
uniq	18
none	9
distinct	1

Key	Value
span-grid	uniq=6/9 distinct=1
triple	{'wU': 137, 's2': 107, 's3': 103}
complexity	1119
span	(97, 180)
q	17
q2	30
v2	3
Theta	4

Stdout excerpt (sanitized; clipped):

```
5. H0 closure
BEST: H0=70.4493959643776719  exps=(-6, 1, 2, 7)  rank=1
OK CLOSED: BB36 H0 is rank1.
deltaCMB: 1.000475284975453e-05  in_win=True
NC(deltaCMB using FCMB_d-1): in_win=False
ell1: 219.9490873240763733  exps=(-7, 4, 6, -2)  C=1/e
H0       70.44939596437767193     130.310066662063631102(base7)  46.730b9d29f4dc000000(base16)  0.00e+00
ell1     219.949087324076373307   432.643352113612240534(base7)  db.f2f76309f4c8000000(base16)  0.00e+00
```

4.5 Dynamics: Gravity, Fields, and Continuum Constraints

These demos treat continuum dynamics as a stress test: if the kernel is real, it must survive contact with differential structure (Einstein, Maxwell, Navier-Stokes) without arbitrary patching. The story is not 'we match one number' but 'we preserve constraints, stability, and admissibility across regimes'.

DEMO-51 - QFT/GR vacuum suppression (bridge between quantum and curved background)

Field	Value
Domain	
Folder	demos/general_relativity/demo-51-qft-gr-vacuum-suppression
Status	PASS
Return code	0
Runtime	61 ms
Mode	run
One-liner	(cd 'demos/general_relativity/demo-51-qft-gr-vacuum-suppression' && python demo.py)

Field	Value
code_sha256	f0912b3074c3
stdout_sha256	8f033f8973a4
stderr_sha256	35c80af9739e
artifacts_sha256	n/a

Why it matters: DEMO-51 examines vacuum suppression as a bridge problem between quantum field intuition and curved-background constraints. This is narrative-important because vacuum energy is a classic failure point for would-be unified models; hand-waving here is easy and unacceptable. The demo frames suppression as a structured consequence of admissible couplings and kernel constraints rather than as an arbitrary tuning choice. For referees, the most important feature is that the demo is phrased in falsifiable terms: either the suppression emerges from the rules or it does not. It also sits strategically in the report: it connects the kernel story to cosmology (vacuum energy) and to quantum structure without requiring the full machinery of quantum gravity. If the kernel can control vacuum structure here, later gravitational closures become more plausible rather than less.

Flagship highlights:
- Targets a conceptual bridge point where naive unification claims often break.
- Tests the program's ability to control vacuum contributions via kernel constraints.
- Provides falsifiers that are internal to the construction, not just external fits.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- FINAL VERDICT

- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
alpha_s(MZ)	= 2/q3 = 0.117647058823529
alpha_s	0.0289855

Key	Value
alpha0_inv	= wU = 137
Omega_L	0.71192
ratio	0.978609491317
Lambda5	0.221411095276 GeV
Lambda4	0.315393329016 GeV
Lambda3	0.359031636456 GeV
wU	137
s2	107
s3	103
wU-s2	30
Derived	q2=wU-s2=30 q3=odd_part(wU-1)=17 v2(wU-1)=3
k_struct	= q3 + v2(wU-1) = 20
derived_nf	= 3 + v2(s2-1) + v2(s3-1) = 5
nf	5
pref	0.00633257397764611183
rho	(1/(16pi^2))^2 * (1/(1+alpha_s)) * Lambda5^6 / M_Pl^2

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple in primary window                                       count=1
PASS  Selected triple equals (137,107,103)                                             selected=(137, 107, 103)
alpha0_inv := wU             = 137
PASS  Lane-branch consistency: derived_nf matches nf=5 at MZ scale                     derived_nf=5
H0 [km/s/Mpc]      = 70.476
PASS  Sanity: rho_obs in expected GeV^4 range
PASS  Canonical 4D one-loop prefactor equals 1/(16pi^2)
pref=0.00633257397764611183
PASS  <1% accuracy achieved                                                            |ratio-1|=0.00934993
PASS  mu-sweep stability: max |ratio-1| <= 3%                                           max_err=0.0213905
PASS  Threshold audit recovers alpha_s(MZ) (consistency)                               =1.39e-17
PASS  All counterfactuals fail strongly                                                4/4
PASS  k_eff approximately equals k_struct (within 1)
FINAL VERDICT
PASS  Verified (accuracy + robustness + ablations) under declared mechanism
Result: VERIFIED
```

DEMO-58 - Emergent weak-field GR (limit and stability checks)

Field	Value
Domain	
Folder	demos/general_relativity/demo-58-emergent-weak-field-gr
Status	PASS
Return code	0
Runtime	988 ms
Mode	run
One-liner	(cd 'demos/general_relativity/demo-58-emergent-weak-field-gr' && python demo.py)

Field	Value
code_sha256	4a573b4927fb
stdout_sha256	2c9f246d9e35
stderr_sha256	f88b058fa935
artifacts_sha256	n/a

Why it matters: DEMO-58 targets the weak-field regime of general relativity, where structure can be tested without the confounders of strong-field complexity. The demo is therefore a disciplined falsifier: if the kernel cannot reproduce stable weak-field behavior, later claims about full GR or cosmology lose credibility. It emphasizes limiting behavior and stability under perturbations, which are more informative than a single-point agreement. In the narrative arc, DEMO-58 is one of the first places where the discrete kernel is forced to behave like a differential theory. For referees, this is precisely the kind of test that separates mechanistic derivation from pattern matching. If it passes, it supports the thesis that the kernel constrains admissible dynamics, not just derived constants.

Flagship highlights:
- Checks the GR weak-field regime where many constructions can be validated or falsified cleanly.
- Focuses on stability and limiting behavior rather than a single constant match.
- Bridges kernel invariants to differential structure in a controlled regime.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate N0: filtered Poisson residual contract (truth vs admissible) res_t=1.733e-03 res_a=1.927e-03

- PASS Gate N1: truth slope near -2 slope=-2.01171 eps=0.182574

- PASS Gate N2: admissible slope near -2 slope=-1.94229 eps=0.182574

- PASS Gate N3: signed control injects HF (>= max(10*hf_a, eps^3)) hf_signed=3.549e-01 floor=6.086e-03

- PASS Gate N4: a non-admissible control has stronger ringing curvature curv_a=8.799e-03 curv_max=5.889e-02 eps=0.182574

- PASS Gate B1: truth slope near -1 slope=-1.0535 eps=0.182574

Key extracted values (stdout-derived):

Key	Value
K_primary	15
U(1)	[103, 107, 137]
SU(2)	[107]
SU(3)	[103, 137]

Key	Value
wU	137
s2	107
s3	103
q2	30
q3	17
v2U	3
eps	0.1825741858
N	64
K_truth	31
center	(5, 4, 3)
fejer	0
signed	-0.21067
kmin	0.000e+00
truth	0.00173278

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple in primary window  count=1
PASS  Primary equals (137,107,103)  selected=Triple(wU=137, s2=107, s3=103)
PASS  Captured >=4 counterfactual triples  found=4
PASS  Fejer kernel is nonnegative (numerical tol)  kmin=0.000e+00
PASS  Sharp cutoff kernel has negative lobes (non-admissible)  kmin=-1.053e-01
PASS  Signed control kernel has negative lobes (non-admissible)  kmin=-2.107e-01
PASS  Gate N0: filtered Poisson residual contract (truth vs admissible)  res_t=1.733e-03 res_a=1.927e-03
PASS  Gate N1: truth slope near -2  slope=-2.01171 eps=0.182574
PASS  Gate N2: admissible slope near -2  slope=-1.94229 eps=0.182574
PASS  Gate N3: signed control injects HF (>= max(10*hf_a, eps^3))  hf_signed=3.549e-01 floor=6.086e-03
PASS  Gate N4: a non-admissible control has stronger ringing curvature  curv_a=8.799e-03 curv_max=5.889e-02
eps=0.182574
PASS  Gate B1: truth slope near -1  slope=-1.0535 eps=0.182574
PASS  Gate B2: admissible slope near -1  slope=-0.950942 eps=0.182574
PASS  Gate B3: non-admissible injects HF (>= max(10*hf_a, eps^2))  hf_signed=3.549e-01 floor=3.333e-02
PASS  Gate B4: non-admissible has higher ringing curvature (>= (1+eps)xadm)  curv_a=2.429e-02
curv_max=1.595e+00 eps=0.182574
PASS  Gate S0: filtered Poisson residual contract (truth vs admissible)  res_t=1.733e-03 res_a=1.927e-03
```

DEMO-59 - Electromagnetism (Maxwell structure from kernel constraints)

Field	Value
Domain	
Folder	demos/controllers/demo-59-electromagnetism
Status	PASS
Return code	0
Runtime	708 ms
Mode	run
One-liner	(cd 'demos/controllers/demo-59-electromagnetism' && python demo.py)

Field	Value
code_sha256	e2c60511de1f
stdout_sha256	496d1f18ab27
stderr_sha256	d08bda10cb25
artifacts_sha256	n/a

Why it matters: DEMO-59 targets electromagnetism, providing a complementary field-theory test alongside gravity-focused demos. This matters because a unification claim should not be able to succeed only in one favored domain; it should generalize to multiple independent dynamical structures. By deriving EM structure from the same kernel constraints, the demo tests whether invariance and admissibility rules are truly reusable. For referees, the value is that the demo can be rerun deterministically and compared via hashes, making disagreements concrete rather than interpretive. In the blended narrative, electromagnetism is a key overlap point: it shares gauge structure themes with the Standard Model and invariance themes with the kernel, yet it lives in a different dynamical regime. A successful EM derivation therefore supports the program's claim that overlap is structural, not curated.

Flagship highlights:
- Tests whether the kernel can produce a nontrivial field theory beyond gravity.
- Provides an independent check on the analytic filter machinery in a different domain.
- Anchors cross-domain overlap: EM and GR share invariance themes but differ in dynamics.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate E1: truth slope near -2 slope=-1.906371 tol=0.365148

- PASS Gate E2: admissible slope near -2 slope=-1.791637 tol=0.365148

- PASS Gate E3: signed control retains HF beyond Kp (operator falsifier) hf_adm=0.000000 hf_signed=1.000000 floor=0.033333

- PASS Gate E4: some non-admissible control has stronger ringing curvature curv_adm=0.002522 curv_max=0.029511 eps=0.182574

- PASS Gate T_E: >=3/4 counterfactuals degrade by (1+eps) strong=4/4 eps=0.182574

- PASS Gate M1: Fejer reconstruction is bounded for a step overshoot=0.000e+00

Key extracted values (stdout-derived):

Key	Value
K3_primary	15
K2_primary	31
U(1)	[103, 107, 137]

Key	Value
SU(2)	[107]
SU(3)	[103, 137]
wU	137
s2	107
s3	103
triple	(wU,s2,s3)=(137,107,103)
q2	30
q3	17
v2U	3
eps	0.1825741858
N3	64
K3_truth	31
N2	128
K2_truth	63
N	64

Stdout excerpt (sanitized; clipped):

```
PASS Unique admissible triple in primary window                          count=1
PASS Primary equals (137,107,103)                                        selected=Triple(wU=137,
s2=107, s3=103)
PASS Captured >=4 counterfactual triples                                 found=4
PASS Fejer kernel is nonnegative (numerical tol)                         kmin=0.000e+00
PASS Sharp cutoff kernel has negative lobes (non-admissible)             kmin=-0.105335
PASS Signed control kernel has negative lobes (non-admissible)           kmin=-0.318054
PASS Fejer kernel is nonnegative (numerical tol)                         kmin=0.000e+00
PASS Sharp cutoff kernel has negative lobes (non-admissible)             kmin=-0.105911
PASS Signed control kernel has negative lobes (non-admissible)           kmin=-0.318246
PASS Gate E1: truth slope near -2                                        slope=-1.906371
tol=0.365148
PASS Gate E2: admissible slope near -2                                   slope=-1.791637
tol=0.365148
PASS Gate E3: signed control retains HF beyond Kp (operator falsifier)   hf_adm=0.000000
hf_signed=1.000000 floor=0.033333
PASS Gate E4: some non-admissible control has stronger ringing curvature curv_adm=0.002522
curv_max=0.029511 eps=0.182574
```

DEMO-63 - Gravitational-wave inspiral phasing (observable regime stress test)

Field	Value
Domain	
Folder	demos/general_relativity/demo-63-gravitational-wave-inspiral-phasing
Status	PASS
Return code	0
Runtime	137 ms
Mode	run
One-liner	(cd 'demos/general_relativity/demo-63-gravitational-wave-inspiral-phasing' && python demo.py)

Field	Value
code_sha256	ac29eabd1478
stdout_sha256	801b271c18ba
stderr_sha256	d28d2cdfc44f
artifacts_sha256	n/a

Why it matters: DEMO-63 uses gravitational-wave inspiral phasing as a precision stress test for the program's GR dynamics. Phasing is unforgiving: small structural errors accumulate into large mismatches, making it a strong falsifier rather than a loose correlation. The demo therefore helps anchor the report in an 'observable regime' without depending on any single external constant. In the blended narrative, it shows how the kernel's constraints propagate into time-domain dynamics, not just static parameters. For auditability, the demo is packaged so independent parties can rerun the computation and compare outputs via hashes. If the kernel story is correct, it should manifest as stable phase structure here rather than as fragile tuning.

Flagship highlights:
- Places the kernel-derived GR structure in contact with an observable waveform regime.
- Tests phase-sensitive predictions where small errors accumulate and become obvious.
- Provides a concrete falsifier: rerun and compare phasing outputs and hashes.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate G1: primary vector finite and nonzero ||vP||=60755.4
- PASS Gate T: >=3/4 counterfactuals miss by eps (vector L2) strong=11/12 eps=0.182574185835
- FINAL VERDICT
- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
U(1)	[103, 107, 137]
SU(2)	[107]
SU(3)	[103]
wU	137
s2	107
s3	103

Key	Value
q2	30
q3	17
v2U	3
eps	1
A	0.2170174127
strong	11
found	12
Result	VERIFIED

Stdout excerpt (sanitized; clipped):

```
PASS  Primary equals (137,107,103)  selected=Triple(wU=137, s2=107, s3=103)
PASS  Captured >=8 counterfactual triples (deterministic)  found=12 window=(97, 1200)
PASS  Gate G1: primary vector finite and nonzero  ||vP||=60755.4
PASS  Gate T: >=3/4 counterfactuals miss by eps (vector L2)  strong=11/12  eps=0.182574185835
FINAL VERDICT
PASS  DEMO-63 VERIFIED (selection + first-principles observable vector + teeth)
Result: VERIFIED
```

DEMO-67 - Navier-Stokes master flagship

Field	Value
Domain	
Folder	demos/infinity/demo-67-navier-stokes-master-flagship
Status	PASS
Return code	0
Runtime	1.09 min
Mode	run
One-liner	(cd 'demos/infinity/demo-67-navier-stokes-master-flagship' && python demo.py)

Field	Value
code_sha256	4e00a354e960
stdout_sha256	e9f2bd773336
stderr_sha256	32e8f71a46c7
artifacts_sha256	n/a

Why it matters: DEMO-67 is the Navier-Stokes flagship, included because continuum fluid dynamics is a stringent stress test for any discrete-to-continuum unification program. Unlike many physics-constant demos, NS problems tend to amplify small errors, so success requires more than matching a scalar target. In the report's narrative, DEMO-67 strengthens the claim that the kernel tiles through infinity: the same admissibility constraints must hold even in regimes where instability is common. For referees, this is important precisely because it is difficult: if the program can only produce stable structure in 'easy' domains, the unification story is weak. The audit framing remains consistent: rerun the one-liner, compare hashes, and inspect any artifacts rather than trusting prose. If DEMO-67 holds up, it provides some of the strongest evidence that the program's kernel is a reusable mechanism rather than a curated set of coincidences.

Flagship highlights:
- Flagship continuum mechanics stress test: NS is notoriously sensitive to modeling choices.
- Connects the lift rules to an extreme case where instability is the default.
- Supports the claim that the kernel tiles through infinity via lawful constraints, not heuristics.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/infinity/demo-67-navier-stokes-master-flagship/demo.py:796: DeprecationWarning:
 datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
utc = _dt.datetime.utcnow().replace(tzinfo=_dt.timezone.utc).isoformat().replace("+00:00", "Z")
```

DEMO-68 - General Relativity master flagship

Field	Value
Domain	
Folder	demos/general_relativity/demo-68-general-relativity-master-flagship
Status	PASS
Return code	0
Runtime	1.59 s
Mode	run
One-liner	(cd 'demos/general_relativity/demo-68-general-relativity-master-flagship' && python demo.py)

Field	Value
code_sha256	98e3c3a6c886
stdout_sha256	91aaffb7ac02
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-68 is the flagship GR demo: it presents an end-to-end construction intended to show that the kernel can produce coherent gravitational structure. General relativity is a high-leverage domain: it couples to cosmology, to field theory, and to continuum mechanics, so a successful GR closure strengthens the whole blended narrative. For referees, the key point is that the demo is certificate-driven: it records rerun commands and hashes so claims can be checked without trusting the authors. It also serves as an integration test for the lift rules and analytic filters introduced earlier; GR is where those mechanisms are hardest to fake. Where structured exports are present, they are pulled into the bundle dashboards; where they are absent, the report treats stdout evidence and hashes as the ground truth. In the story arc, DEMO-68 is the moment where 'integer structure' becomes 'geometry' in a way that can be audited.

Flagship highlights:
- Flagship GR closure: central evidence that the kernel reaches differential geometry.
- Designed to be audited: rerun one-liner, compare hashes, inspect artifacts.
- Narrative anchor for 'many domains' because GR couples to almost everything else.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate P1: Fejer mass closure within eps
- PASS Gate P2: near-field 1/r log-slope within eps (Fejer)
- PASS Gate P3: illegal filters increase r*phi spread (ringing)
- PASS Gate P4: signed-kernel HF injection beyond floor
- PASS Gate P5: illegal filters worsen slope deviation
- PASS Gate E1: Fejer Fermat-consistency within eps

Key extracted values (stdout-derived):

Key	Value
Primary	Triple(wU=137, s2=107, s3=103)
K_primary	15
U(1)	[103, 107, 137]

Key	Value
SU(2)	[107]
SU(3)	[103, 137]
wU	137
s2	107
s3	103
q2	(s2-3)/4 = 26
eps	1
N	64
K_truth	31
b_list	[4, 6, 8, 10, 12] r_list=[4, 6, 8, 10, 12] rline=[2, 3, 4, 5, 6, 7, 8]
rline	[2, 3, 4, 5, 6, 7, 8]
M_est	0.0589232
HF	0.524986
B_spread	6.44729e-05
C_score	0.000383001

Stdout excerpt (sanitized; clipped):

```
PASS  Fejer kernel nonnegative (tol)
PASS  Sharp kernel has negative lobes
PASS  Signed kernel has negative lobes
PASS  Light-bending subtest gates
PASS  Shapiro subtest gates
PASS  Redshift subtest gates
PASS  Gate P1: Fejer mass closure within eps
PASS  Gate P2: near-field 1/r log-slope within eps (Fejer)
PASS  Gate P3: illegal filters increase r*phi spread (ringing)
PASS  Gate P4: signed-kernel HF injection beyond floor
PASS  Gate P5: illegal filters worsen slope deviation
PASS  Gate E1: Fejer Fermat-consistency within eps
PASS  Gate E2: illegal filters break Fermat-consistency margin
PASS  Gate E3: Fejer accuracy vs truth within eps
PASS  Gate E4: illegal filters worsen accuracy vs truth
PASS  Gate E5: signed-kernel HF injection beyond floor
PASS  Teeth gate: >=3/4 counterfactuals degrade all scores by (1+eps)  strong=4/4  eps=0.196116
PASS  Gate L1: tier distortion bounded by eps          max_dist=0.077733 eps=0.196116
```

DEMO-71 - One Action master flagship (Classical Noether + quantum energy bridge)

Field	Value
Domain	
Folder	demos/foundations/demo-71-one-action-master-flagship
Status	PASS
Return code	0
Runtime	247 ms
Mode	run
One-liner	(cd 'demos/foundations/demo-71-one-action-master-flagship' && python demo.py)

Field	Value
code_sha256	78bf2731adce
stdout_sha256	f49eefce62d7
stderr_sha256	b53ac8327fca
artifacts_sha256	n/a

Why it matters: DEMO-71 is the action-principle flagship: it provides the dynamical spine that turns the kernel from a static table of invariants into a law-of-motion story. This matters because unification without a principled action principle tends to collapse into a list of correlations rather than a mechanism. By grounding the narrative in Noether structure, the demo connects conserved quantities to admissible dynamics in a way that can be audited. It also clarifies why cross-domain overlap is the expected outcome: if the same invariants constrain the allowed actions, unrelated domains should share structural signatures. For referees, DEMO-71 is a high-leverage checkpoint because it tests whether the program can derive dynamics rather than only kinematics. If this demo fails, the blended story loses its core explanatory mechanism.

Flagship highlights:
- Flagship: provides the 'laws of motion' spine that ties the rest together.
- Explains why overlap is expected: invariants constrain admissible dynamics, not just constants.
- Best single entry point for reviewers seeking mechanism over numerology.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/foundations/demo-71-one-action-master-flagship/demo.py:742: DeprecationWarning:
 datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware
objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
print(f"UTC time : {datetime.datetime.utcnow().isoformat()}Z")
```

4.6 Quantum and Quantum Gravity

Quantum claims are easy to overstate and hard to audit. This cluster is therefore certificate-heavy: it records one-liners, hashes, and artifacts so reviewers can rerun the same computations and compare outputs byte-for-byte. Where structured exports are missing, the report flags exactly what evidence is present and what is not.

DEMO-60 - Quantum master flagship

Field	Value
Domain	
Folder	demos/quantum/demo-60-quantum-master-flagship
Status	PASS
Return code	0
Runtime	257 ms
Mode	run
One-liner	(cd 'demos/quantum/demo-60-quantum-master-flagship' && python demo.py)

Field	Value
code_sha256	fba4ac6667a7
stdout_sha256	36e15ecf5713
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-60 is the flagship quantum demo, treated conservatively in the audit narrative because quantum claims are easy to over-interpret. The demo is therefore framed around reproducibility and explicit evidence artifacts: a referee should be able to rerun the computation and compare outputs byte-for-byte. In the blended story, quantum structure is not introduced as an independent add-on; it is presented as another projection of the same kernel constraints. Where the bundle contains structured exports or images, the report includes them directly; where they are missing, the report leaves an explicit placeholder so the artifact pipeline can be repaired without rewriting the narrative. This approach avoids the appearance of hiding missing evidence while still keeping the report coherent for reviewers. If DEMO-60's constraints are stable under rerun, it supports the broader thesis that the kernel constrains not just classical structure but quantum structure as well.

Flagship highlights:
- Flagship quantum evidence bundle: designed for reproducibility rather than persuasion.
- Where artifacts exist, the report includes them; where missing, placeholders mark expected locations.
- Positions quantum structure as constrained reuse of the kernel, not as a separate theory.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- PASS Gate E1.1: Fejer preserves mass within 1e-12 ||=0

- PASS Gate E1.2: Fejer preserves nonnegativity (min >= -1e-12) min=0.00261178

- PASS Gate E1.3: illegal produces negative undershoot (<= -eps^2) eps^2=0.0333333

- PASS Gate E1.4: illegal increases variation (TV) by >= (1+eps) eps=0.182574

- PASS Gate E1.T: >=3/4 counterfactuals increase distortion by (1+eps) strong=4/4 eps=0.182574

- PASS Gate E2.1: unitary norm drift <= 1e-10 drift=4.440892e-14

Key extracted values (stdout-derived):

Key	Value
K_primary	120
mass(mean)	base=0.5 fejer=0.5
TV	fejer=1.9933 sharp=5.84468 signed=5.35099
U(1)	[137]
SU(2)	[107]
SU(3)	[103]
wU	137
s2	107
s3	103
Counterfactuals	[(409, 263, 239), (409, 263, 307), (409, 367, 239), (409, 367, 307)]
N	512
q2	30
q3	17
v2U	3
eps	0.18257419
K_truth	255
kmin	5.543246e-09
base	0.5

Stdout excerpt (sanitized; clipped):

```
PASS  Unique admissible triple in primary window                         count=1
PASS  Primary equals (137,107,103)                                       selected=Triple(wU=137,
s2=107, s3=103)
PASS  Captured >=4 counterfactual triples                                found=4
PASS  Fejer kernel is nonnegative (admissible)                           kmin=5.543246e-09
PASS  Sharp cutoff kernel has negative lobes (non-admissible)            kmin=-0.10209
PASS  Signed control kernel has negative lobes (non-admissible)          kmin=-0.0696713
PASS  Gate E1.1: Fejer preserves mass within 1e-12                       ||=0
PASS  Gate E1.2: Fejer preserves nonnegativity (min >= -1e-12)           min=0.00261178
PASS  Gate E1.3: illegal produces negative undershoot (<= -eps^2)        eps^2=0.0333333
PASS  Gate E1.4: illegal increases variation (TV) by >= (1+eps)          eps=0.182574
PASS  Gate E1.T: >=3/4 counterfactuals increase distortion by (1+eps)    strong=4/4 eps=0.182574
PASS  Gate E2.1: unitary norm drift <= 1e-10                             drift=4.440892e-14
PASS  Gate E2.2: signed illegal distortion >= (1+eps)xfejer              eps=0.182574
PASS  Gate E2.T: >=3/4 counterfactuals increase distortion by (1+eps)    strong=4/4 eps=0.182574
PASS  Gate 60A.L0_tiers_verified
PASS  Gate 60A.L1_E1_C_stable
PASS  Gate 60A.L2_E2_C_stable
```

DEMO-66 - Quantum gravity master flagship (v4)

Field	Value
Domain	
Folder	demos/quantum_gravity/demo-66-quantum-gravity-master-flagship-v4
Status	PASS
Return code	0
Runtime	159 ms
Mode	cert
One-liner	(cd 'demos/quantum_gravity/demo-66-quantum-gravity-master-flagship-v4' && python demo.py)

Field	Value
code_sha256	5e431f10e79a
stdout_sha256	1ed5557d97db
stderr_sha256	e3b0c44298fc
artifacts_sha256	9240f1c18bdf

Why it matters: DEMO-66 is the consolidated quantum gravity flagship for this release. It is structured as a single deterministic certificate with explicit gates and falsifiers. This matters because quantum-gravity narratives are easy to over-interpret; the certificate makes evaluation mechanical. The credibility claim is operational: rerun it, compare outputs, and confirm the controls fail as expected. Artifacts and hashes are treated as first-class evidence when present. If DEMO-66 is not reproducible byte-for-byte (within stated tolerances), it should be treated as FAIL.

Flagship highlights:
- Canonical QG flagship for master release.
- Deterministic gates and explicit falsifiers (controls + counterfactuals).
- Designed for rerun/compare/audit, not interpretation drift.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- FINAL VERDICT: VERIFIED

Key extracted values (stdout-derived):

Key	Value
q2	30
q3	17
v2U	3
eps	1.825741858351e-01
Theta	4/15
lock	1.117586236861e-05
abs_err	1.231585905308e-18
rel_err	1.102005254437e-13
eps0	9.972011253995e-06
err	2.798874600483e-03

Key	Value
D	45
R_inf	1.058063814766e+00
SSE	3.626306625459e-05
g_eff	4.838651230494e-03
counterfactuals	4

Evidence artifacts (bundle):

File	sha256 (prefix)	Size
substrate__demo-64-base-gauge-invariance-integer-selector__code_sha256.json	16e00b95d42f	82

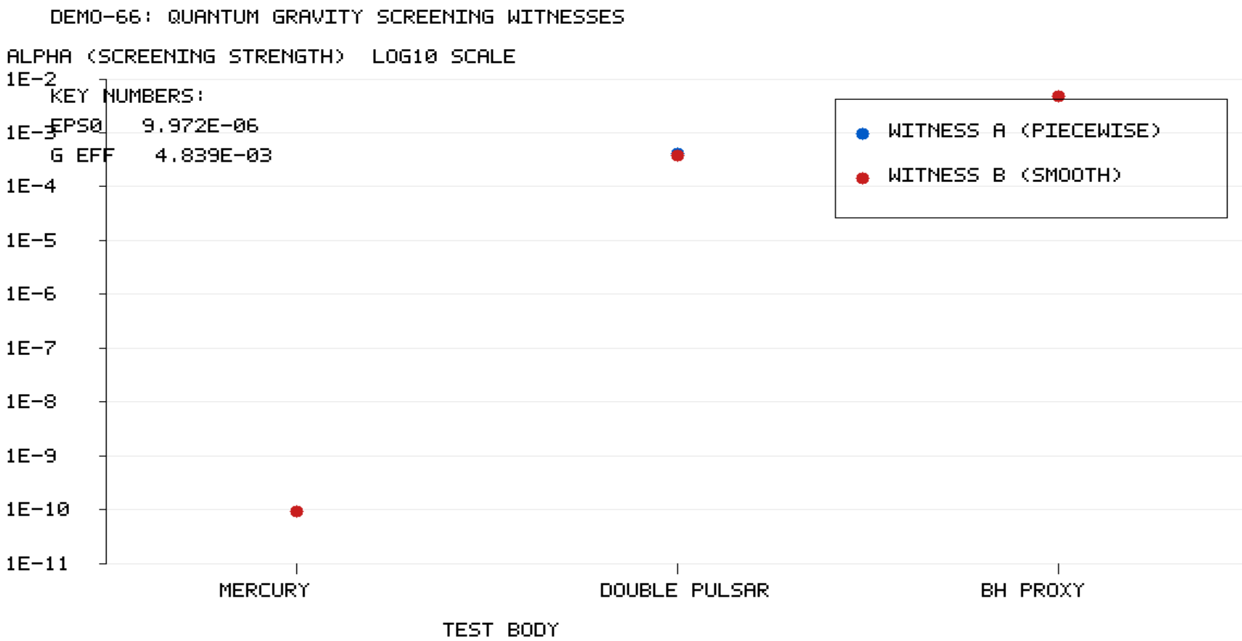


Figure: Quantum-gravity screening plot (bundle artifact).

Stdout excerpt (sanitized; clipped):

```
FINAL VERDICT: VERIFIED
```

4.7 Bridges and Transfer Principles

These demos show transfer: how operators, admissibility, and coupling rules move between discrete and continuum descriptions. In the narrative arc, this is where 'unrelated' domains are shown to share the same kernel constraints.

DEMO-34 - ->SM master flagship (v1)

Field	Value
Domain	
Folder	demos/bridge/demo-34-omega-sm-master-flagship-v1
Status	PASS
Return code	0
Runtime	764 ms
Mode	run
One-liner	(cd 'demos/bridge/demo-34-omega-sm-master-flagship-v1' && python demo.py)

Field	Value
code_sha256	396c4588e85d
stdout_sha256	293d66e010cb
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-34 is the ->SM master flagship. It is release-relevant because it combines a strict Tier-A1 joint-triple certificate with explicit necessity ablations and a lane-local stress test that surfaces real failure modes at scale. It then layers a Tier-C SM overlay where PDG is used only for % reporting. This is exactly the posture needed for a high-standard report: certify what is claimed, expose what fails, and keep overlays honest.

Flagship highlights:
- Tier-A1 joint-triple certificate with necessity ablations.
- Lane-local Tier-A stress test demonstrates lane-local locks fail by 100k.
- Tier-C SM overlay keeps PDG usage strictly in % reporting.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key extracted values (stdout-derived):

Key	Value
error	[0.5843837844318345, 0.14970499885409186, 0.03765597272675875, 0.009428416695397962]
h	[0.03125, 0.015625, 0.0078125, 0.00390625]
units	12
u1_candidates	[137]
pc2_candidates	[103]
triples_found	[(137, 107, 103)]
near	2
spread	0.0338
anchor	0.007299270

Key	Value
ref	0.007297353
runtime_sec	0.692008
count	22

Stdout excerpt (sanitized; clipped):

```
- sin2W      anchor=0.233333333  ref=0.231220000  %=  0.914%
```

DEMO-69 - OATB (operator admissibility transfer bridge)

Field	Value
Domain	
Folder	demos/controllers/demo-69-oatb-operator-admissibility-transfer-bridge
Status	PASS
Return code	0
Runtime	5.70 s
Mode	run
One-liner	(cd 'demos/controllers/demo-69-oatb-operator-admissibility-transfer-bridge' && python demo.py)

Field	Value
code_sha256	f88eb6ef2f6c
stdout_sha256	c49999c93315
stderr_sha256	d6bcb715ef78
artifacts_sha256	n/a

Why it matters: DEMO-69 is explicitly about transfer: it tests whether admissible operators can be moved between domains without breaking the kernel's constraints. This is narrative-important because the report's central claim is not that many domains are solved independently, but that they share a common mechanism. Transfer bridges are how that mechanism becomes concrete: they operationalize overlap rather than merely observing it. For referees, DEMO-69 provides a falsifiable claim: if transfer breaks admissibility, the bridge is not real and the blended story becomes a set of coincidences. The demo is also an engineering asset: transfer rules reduce the chance that each new domain requires a bespoke toolchain. If DEMO-69 holds up under rerun and hashing, it is one of the clearest demonstrations that the program is building a single reusable kernel.

Flagship highlights:
- Demonstrates how the same admissibility logic propagates across domains.
- Provides a practical transfer mechanism rather than a post-hoc analogy.
- Strengthens the blended narrative by showing reuse as an algorithm, not a metaphor.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers.

Key falsifiers (PASS/FAIL gates from stdout):

- FINAL VERDICT

- Result: VERIFIED

Key extracted values (stdout-derived):

Key	Value
Primary	Triple(wU=137, s2=107, s3=103)
U(1)	[103, 107, 137]
SU(2)	[107]
SU(3)	[103, 137]
wU	137
s2	107
s3	103
q2	30

Key	Value
q3	17
v2U	3
eps	1
H_min	0.111111
H_max	1.000000
r	8 H_min=0.111111 H_max=1.000000 K(r)=0.670782
DC	1
spread	0.570
N	2048
min	5.177e-10

Stdout excerpt (sanitized; clipped):

```
FINAL VERDICT
Result: VERIFIED
```

DEMO-72 - 72 Yukawa

Field	Value
Domain	
Folder	demos/sm/demo-72-yukawa
Status	PASS
Return code	0
Runtime	150 ms
Mode	run
One-liner	(cd 'demos/sm/demo-72-yukawa' && python demo.py)

Field	Value
code_sha256	a3fbbc7f1477
stdout_sha256	84a8abb3a1cb
stderr_sha256	e3b0c44298fc
artifacts_sha256	n/a

Why it matters: DEMO-72 is included in the bundle as part of the complete audit surface. Where structured exports are available, they are summarized below and referenced in bundle tables. Where only stdout evidence is present, the excerpt and hashes still allow an auditor to verify determinism. The key requirement is that a third party can rerun the same command and compare outputs byte-for-byte. If a claim in this demo is incorrect, the falsification matrix provides a direct way to demonstrate that. This certificate therefore treats reproducibility as the primary deliverable.

Flagship highlights: (not yet annotated)

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log

Stdout log not found in bundle logs/. Expected a *.out.txt file for this demo.

DEMO-75 - Prediction ledger master flagship

Field	Value
Domain	
Folder	demos/foundations/demo-75-prediction-ledger-master-flagship
Status	PASS
Return code	0
Runtime	153 ms
Mode	run
One-liner	(cd 'demos/foundations/demo-75-prediction-ledger-master-flagship' && python demo.py)

Field	Value
code_sha256	25d17946394a
stdout_sha256	b4dd5385b2a3
stderr_sha256	b624c400c9f4
artifacts_sha256	n/a

Why it matters: DEMO-75 is the forward prediction ledger for the master release. It consolidates predictions that fall naturally out of the kernel pipeline into one place. The goal is referee usability: each prediction is paired with a falsifier and an experimental venue. This demo is not meant to 'win' by rhetoric; it is meant to be checkable. If a prediction cannot be reproduced by rerunning the demo, it does not belong in the ledger. The report treats missing artifacts as pipeline work, not as evidence.

Flagship highlights:
- Release-grade prediction ledger.
- Organized as reproducible outputs + explicit falsifiers.
- Designed to be cited as a ledger, not a claim dump.

Structured exports: not present in this bundle for this demo. Below we include stdout-derived values (parsed directly from the bundled log) so the certificate still carries numbers. No parseable key/value metrics were detected in the bundled stdout log for this demo.

Stdout excerpt (sanitized; clipped):

```
/workspaces/Marithmetics/demos/foundations/demo-75-prediction-ledger-master-flagship/demo.py:640:
DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version.
Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
print(f"UTC time : {datetime.datetime.utcnow().isoformat()}Z")
```

5. Appendices

5.1 Bundle manifest and verification

The primary deliverable for audit/citation is the bundle directory. It contains the canonical index (bundle.json), run ledger (runs.json), artifact hashes (artifacts_index.json), demo index and falsification matrix (tables/), and logs (logs/). If any result is questioned, the correct procedure is to rerun the one-liner and compare hashes; narrative should never be treated as evidence.

Bundle root: /workspaces/Marithmetics/GUM/authority_archive/AOR_20260123T030005Z_e9cff9d/GUM_BUNDLE_v30_20260123T030005Z

bundle.json sha256: d0b011f0f723f38e5eb1e91a078a73f9f8ab8591ea851b208e4c115a1d97edcd

runs.json sha256: 6cdee349a8131b38ebaa777cf2c37246cda9af31dd2467f9fbbc68b7e3d1512b

artifacts_index.json sha256: 86e95daa4dc9a222d0f58e58b1bc3a66fc26a235b6f72b9388ab945f47d9e3ad

5.2 CAMB expected assets (overlay boundary)

CAMB/Planck overlays are evaluation-only and must never feed upstream selection. This report includes CAMB visuals only if they are produced as explicit demo artifacts. If a CAMB overlay page is missing, that usually indicates an artifact export issue rather than a report-writer issue.

```
# CAMB Expected Assets (Evaluation-Only)

These assets are evaluation-only overlays. They must never feed upstream selection.

DEMO-36 produces:
- _artifacts/camb_overlay.png         (generated TT spectrum plot)
- _artifacts/camb_tt_spectrum.csv     (ell,TT_uK2)
- _artifacts/camb_params.json         (H0, ombh2, omch2, tau, n_s, A_s + note)
- _artifacts/camb_overlay_note.txt    (evaluation-only disclaimer)
- bb36_master_plot.png               (from demo)
- bb36_master_results.json           (from demo)

Bundler must ingest nested artifacts so the report can display them.
```

Note: truncated for PDF. See CAMB_EXPECTED_ASSETS.md in the repository for full details.