Computational Specification for a Relational Static Cosmology Physics Engine

Ontological Relationalism and the Elimination of Coordinate Spacetime

Modern general relativity represents the universe using a continuous four-dimensional pseudo-Riemannian manifold, interpreting gravity as the physical warping of coordinate space and time.[1] However, this geometric formulation is not mathematically unique, suggesting that treating spacetime as a physical, stretchable fabric is an instrumentalist shortcut.[1] To build a simulation engine that functions entirely without a physical spacetime metric, space and time must be modeled as emergent, relational properties of discrete matter configurations.[1] This approach aligns with Leibnizian relationalism, which asserts that space is merely an abstract web of relations among coexisting bodies, disappearing entirely if all matter is removed.[1]

Point-Coincidences and the Hole Argument

The ontological rejection of coordinate spacetime is mathematically supported by Einstein's Hole Argument.[1] If individual points on a coordinate manifold possessed an independent physical existence, the application of an active diffeomorphism—a smooth shifting of the coordinate grid within a matter-free region—would generate mathematically distinct metric fields that satisfy the identical gravitational field equations.[1] This mathematically distinct mapping would result in a failure of physical determinism.[1]

To preserve determinism, the concept of a physical manifold container must be discarded.[1] Instead, physical reality is defined strictly through point-coincidences, which are the localized intersections of material worldlines and field values.[1] Space and time do not exist as absolute backgrounds; they are generated dynamically by the matter fields contained within the system.[1]

Teleparallelism and Flat Weitzenböck connections

To simulate gravitational interactions on a flat coordinate background without deforming coordinate intervals, the physics engine can implement the Teleparallel Equivalent of General Relativity (TEGR).[1] Instead of using the metric-compatible, torsion-free Levi-Civita connection ($\Gamma^{\lambda}_{\mu\nu}$) of general relativity, TEGR operates on a flat Weitzenböck spacetime.[1] The simulation calculates dynamics using a dynamical tetrad field (e^{a}_{μ}) to define a flat connection [1]:

$$\Gamma^{\lambda}_{\mu\nu} = e^{\lambda}_{a}\partial_{\nu}e^{a}_{\mu}$$

In this representation, the Riemann curvature tensor of the Weitzenböck connection vanishes identically everywhere [1]:

$$R^{\rho}_{\sigma\mu\nu} = 0$$

All gravitational degrees of freedom are instead mapped to a non-vanishing torsion tensor $T^{\lambda}_{\mu\nu}$.[1] Rather than warping space or time, gravity behaves as a translation gauge force mediated by a translational gauge potential (B^{a}_{μ}).[1] This gauge force physically deflects material bodies from their straight-line inertial paths within a flat, non-relativistic coordinate system.[1] Because TEGR is dynamically equivalent to standard general relativity, it yields identical physical predictions for all observable phenomena while operating entirely on flat, Euclidean coordinate systems.[1]

The Discrete Informational Lattice

To replace continuous spatial backgrounds with a computationally decidable framework, the simulation engine represents physical configurations on a discrete informational lattice.[1] This model draws on digital physics frameworks proposed by Konrad Zuse and Edward Fredkin, where physical laws emerge as the outputs of local, algebraic state-machine updates on a cellular automaton.[1] Macroscopic spatial coordinates and distances are modeled as "logical qubits" reconstructed via quantum error-correction codes from highly entangled underlying "physical qubits".[1]

Instead of deconstructing the continuous Einstein field equations into complex, mathematically unstable numerical subroutines—such as York-ADM or BSSN formulations to maintain metric constraints—the engine computes step-by-step algebraic transitions.[1] This structure eliminates the coordinate singularities and infinite processing overhead associated with continuous differential equations.[1]

Computational Domain	Standard General Relativity (Numerical BSSN/ADM)	Relational Lattice Engine (Spacetime-Free)
Space & Time Representation	Continuous 4D pseudo-Riemannian metric manifold [1]	Discrete, emergent relational lattice of entangled informational states [1]
Mathematical Connection	Torsion-free, curved Levi-Civita connection ($\Gamma^{\lambda}_{\mu\nu}$) [1]	Flat Weitzenböck connection with non-vanishing torsion ($T^{\lambda}_{\mu\nu}$) [1]
Interaction Mechanism	Matter deforms metric coordinates; light travels on null geodesics [1]	Translational gauge potentials exert physical forces on flat coordinates [1]

Algorithmic Complexity	Extremely high; requires solving continuous non-linear partial differential equations [1]	Low to moderate; bounded by local, algebraic state-machine updates [1]
Singularity Vulnerability	Subject to coordinate and physical singularities (e.g., black hole collapses) [1]	Coordinate-free; inherently stable against coordinate and mathematical singularities [1]

Rigid-Body Dynamics and Engine-Level Solver Mechanics

To build a practical, real-time interactive simulation of this spacetime-free cosmology, the engine must merge these advanced cosmological equations with a robust, deterministic rigid-body physics solver.[1] In the web demo application, physical entities (such as platforms, dynamic crates, and test meshes) are modeled as discrete mass nodes residing in flat coordinates.[1]

Contact Resolution, Restitution, and Friction Models

At the narrowphase collision level, the engine resolves physical interactions between overlapping rigid bodies.[1] Collision boundaries are represented by discrete meshes, including standard testing shapes such as a Stanford bunny mesh.[4] When the narrowphase detector identifies an intersection, it calculates the collision normal and penetration depth.[1]

The contact solver corrects the velocities of colliding bodies using a configurable bounciness parameter (restitution, e_r) and a surface friction parameter (μ_f).[4] For dynamic impacts, the normal impulse J_n is computed as:

$$J_n = \frac{-(1 + e_r)(v_{\text{rel}} \cdot n)}{m_1^{-1} + m_2^{-1} + (r_1 \times n)^2 I_1^{-1} + (r_2 \times n)^2 I_2^{-1}}$$

where v_{rel} is the relative velocity at the contact point, n is the collision normal, and I represents the moment of inertia.[4]

Friction is implemented via a tangential force that opposes sliding velocity.[4] When modeling slippery environments—such as a dynamic crate sliding across a low-friction ice platform—the tangential friction impulse J_t is clamped to a maximum value governed by Coulomb's law [4]:

$$|J_t| \leq \mu_f J_n$$

By dynamically modifying μ_f and e_r, the engine simulates realistic bouncing and sliding behaviors across distinct material surfaces.[4]

Constraints, Springs, and Structural Assemblies

To simulate complex mechanical assemblies, the engine resolves joint constraints using Lagrange multipliers and Jacobians.[5] For example, assemblies like the Trammel of Archimedes are simulated inside a dynamic containment box to test coordinate stability and constraint limits.[6] For rolling constraints and complex circular trajectories, the mathematical formulations require exact differentiation of coordinate constraints, which are prone to floating-point arithmetic errors if solved with low-precision integrators.[5]

Elastic connections are modeled using dynamic spring-mass-damper systems.[5] The force exerted by a spring connector is defined as:

$$F_s = -k_s(x - x_0) - d_s v$$

where k_s is the spring stiffness, d_s is the damping coefficient, and v is the relative velocity.[5]

To prevent numerical instability and sudden explosive energy gains, the engine matches the spring stiffness to the global integration frequency (system frequency, f_{sys}) [5]:

$$f_{\text{sys}} = \frac{1}{2\pi}\sqrt{\frac{k_s}{m}}$$

If f_{sys} approaches the Nyquist limit of the physics time step (Δt), the integration errors cascade, causing constraint drift.[5] The engine solves these configurations by assembling rigid structural members using triangular geometries.[5] Triangulation distributes mechanical stresses across multiple nodes, ensuring structural rigidity without relying on high-frequency spring constraints.[5]

Scene Serialization and the JSON Scene-Graph

To manage and share complex simulation environments, the engine incorporates a dedicated serialization editor, code-named "slingshot".[6] Generating sophisticated physical scenarios by hand is highly inefficient and error-prone.[6]

The slingshot editor serializes the entire physical state—including node coordinates, mass profiles, restitution values, spring constants, and rotation constraints—into structured JSON scene-graphs.[6] This allows the physics engine to instantly reload, mutate, and parse custom scenarios.[6]

JSON

```json
{
  "simulation_metadata": {
    "engine_version": "1.7.0",
    "global_damping": 0.01,
    "integration_timestep": 0.01667
  },
  "rigid_bodies": [
    {
      "id": "crate_01",
      "type": "dynamic",
      "mesh": "cube",
      "mass": 5.0,
      "restitution": 0.5,
      "friction": 0.3,
      "position": [0.0, 10.0, 0.0],
      "velocity": [0.0, -2.0, 0.0]
    },
    {
      "id": "ice_platform",
      "type": "static",
      "mesh": "plane",
      "mass": 0.0,
      "restitution": 0.1,
      "friction": 0.02,
      "position": [0.0, 0.0, 0.0],
      "velocity": [0.0, 0.0, 0.0]
    }
  ],
  "constraints": [
    {
      "type": "spring",
      "body_a": "crate_01",
      "body_b": "ice_platform",
      "stiffness": 150.0,
      "damping": 5.0,
      "rest_length": 5.0
    }
  ]
}
```


Relational Gravity and the Entropic Beta-Matter Formulation

Because the simulation engine does not contain a physical spacetime manifold, it must reject standard metric-based gravitational acceleration.[1] Instead, the engine implements an entropic, relational gravity model where gravitational forces emerge from informational gradients among discrete baryonic mass nodes, completely replacing the concept of metric curvature.[1]

Holographic Entropy and Verlinde's Framework

Following the holographic principle and Erik Verlinde's emergent gravity framework, space is represented as a series of informational boundary screens.[8] The informational entropy S encoded on a spherical boundary screen of radius r enclosing a central mass M is defined as [7]:

$$S = \frac{k_B A}{4\ell_P^2} = \frac{k_B \pi r^2}{\ell_P^2}$$

where $A = 4\pi r^2$ is the screen area and $\ell_P = \sqrt{G\hbar/c^3}$ is the Planck length.[7]

The total number of informational degrees of freedom N on the boundary screen is written as [7]:

$$N = \frac{A}{\ell_P^2} = \frac{4\pi r^2 c^3}{G\hbar}$$

By applying the equipartition theorem, the total energy associated with these degrees of freedom is equated to the enclosed rest mass, $E = \frac{1}{2} N k_B T = Mc^2$.[7] By substituting the relations and applying the Unruh temperature formula ($T = \hbar a / 2\pi c k_B$), the standard Newtonian gravitational acceleration is derived [7]:

$$a = \frac{GM}{r^2}$$

This demonstrates that gravitational attraction can be modeled as an entropic force arising from information gradients on a flat coordinate grid without invoking curved spacetime.[7]

The Missing 2/3 Factor Correction

When simulating entropic gravity in low-acceleration regimes, a critical mathematical correction must be integrated.[10] Erik Verlinde's original derivations of emergent gravity contained a missing geometric factor of $2/3$ when integrating the spatial volume-law entropy of dark energy in de Sitter space.[10] In the presence of a central baryonic mass M , the volume-law entropy of space competes with the area-law entropy of the boundary screen.[10]

Verlinde's original formulation incorrectly calculated Milgrom's critical acceleration threshold a_0 as [10]:

$$a_0 = \frac{cH_0}{6} \approx 1.1 \times 10^{-10}\ \mathrm{m/s}^2$$

By correctly integrating the volume-law contribution, the missing $2/3$ factor is restored, modifying the critical acceleration relation [10]:

$$a_0 = \frac{2}{3} \cdot \frac{cH_0}{6} = \frac{cH_0}{9} \approx 0.76 \times 10^{-10}\ \mathrm{m/s}^2$$

When applied to the dynamics of spiral galaxies, this corrected value provides a superior fit to galactic rotation curves, resolving the anomalous "wiggles" or oscillations that occurred when using the uncorrected constant.[10]

The Relational Coupling Parameter β and MOND Equivalence

To simulate flat rotation curves without introducing physical dark matter particles, the engine implements a localized variation parameter β.[1] Gravity is calculated directly between baryonic mass nodes, with its coupling strength scaled by β [1]:

$$F_G = -\beta(r)\frac{GMm}{r^2}$$

In high-density, high-acceleration environments, the parameter stabilizes at $\beta = 1$, recovering standard Newtonian dynamics.[1] In low-acceleration regimes (where the local acceleration drops below a_0), the coupling strength increases [1]:

$$\beta = \frac{1}{\mu(x)}$$

where $x = a/a_0$ is the dimensionless acceleration ratio.[13] The engine implements a smooth, continuous arc-tangent interpolating function to transition between regimes [13]:

$$\mu(x) = \frac{2}{\pi}\arctan\left(\frac{\pi x}{2}\right)$$

In deep-MOND, low-acceleration regimes ($x \ll 1$), this interpolating function approximates $\mu(x) \approx x$.[13] Substituting this limit into the force equation yields:

$$a^2 \approx a_N a_0 \implies a \approx \frac{\sqrt{GMa_0}}{r}$$

The orbital velocity of a test mass orbiting at radius r becomes independent of distance [13]:

$$v(r) = \sqrt{ar} = (MGa_0)^{1/4}$$

This mathematically reproduces the constant asymptotic velocity observed in galactic rotation curves and derives the Baryonic Tully-Fisher relation ($v^4 \propto M$) directly from the visible baryonic distribution.[13] This local scaling of gravitational strength creates a phantom dark matter effect, coined "β-matter," without requiring non-baryonic matter particles.[1]

The Rotation Curve Fitting Model (RCFM)

To process and predict rotation curves, the engine also supports the Rotation Curve Fitting Model (RCFM).[17] Rather than altering the underlying gravity laws at a distance, the RCFM attributes the flat rotation curves of spiral galaxies to relativistic frame-dragging and frame-shifting effects between curved emitter and receiver frames.[17] Historically, these frame differences were ignored by applying simple Galilean subtraction to relative velocities.[17]
The RCFM instead models these frame variations using Schwarzschild gravitational redshift terms, performing a one-to-one Lorentz mapping across radial coordinates [17]:

$$\text{Lorentz-mapped Doppler Shift} = \gamma\left(1 - \frac{v_{\text{emitter}}}{c}\right)\sqrt{1 - \frac{2GM_{\text{emitter}}}{c^2 r}}$$

Using this formulation, the engine fits rotation curve data across dynamic ranges with fewer free parameters than standard dark matter halo models.[17]

Rotation Curve Parameter	**Standard Dark Matter Halo Models (NFW)**	**Relational MOND / Beta-Matter Solver**	**Relational Frame-Dragging Solver (RCFM)**
Baryonic Mass	Calculated from visible light [14]	Calculated from visible light [14]	Calculated from visible light [17]
Dark Matter Fraction	Typically $\geq$ of total galaxy mass [18]	0% (Completely absent) [1]	0% (Completely absent) [17]
Free Parameters	Multiple halo scaling and density parameters [17]	Single critical acceleration parameter $a_0 =$ [10]	Single model parameter mapping receiver/emitter [17]

Tully-Fisher Relation	Requires fine-tuning of halo and baryonic ratios [17]	Naturally derived as a direct consequence of the low-acceleration limit [13]	Derived from relativistic frame-mapping of luminous mass [17]

Modernized Tired Light and Covarying Coupling Constants

To explain cosmological redshift without metric space expansion, the engine implements the modern Covarying Coupling Constants and Tired Light (CCC+TL) hybrid model.[1]

Resolving Historical Redshift Objections

Historically, Fritz Zwicky's 1929 tired light hypothesis was discarded because simple photon scattering off free electrons would produce angular blurring, introduce color-dependent wavelength dispersion, and fail to match the $(1+z)^{-4}$ Tolman surface brightness relation.[3] The CCC+TL framework resolves these objections by combining tired light energy decay with the systematic, covarying evolution of fundamental coupling constants over cosmic time.[1]

The gravitational constant $G(t)$, the speed of light $c(t)$, Planck's constant $h(t)$, and the elementary charge $e(t)$ vary in a covarying, interrelated manner.[1] This covariance is structured such that all dimensionless ratios—including the fine-structure constant α—remain strictly invariant over cosmic time [1]:

$$\alpha = \frac{e(t)^2}{2\epsilon_0 h(t) c(t)} = \text{Constant}$$

Because these constants co-evolved, ancient atomic configurations had stronger binding energies, emitting photons at lower frequency states.[1] The total observed cosmological redshift z is divided between these historical atomic states and the energy decayed during transit [1]:

$$1 + z = (1 + z_{\text{atomic}})(1 + z_{\text{TL}})$$

This hybrid model satisfies the Tolman surface brightness test, fits high-redshift Type Ia supernova data, and extends the calculated age of the universe to 26.7 billion years, explaining the mature galaxies observed at cosmic dawn by the James Webb Space Telescope.[1]

Mathematical Derivation of Tired Light Energy Decay

The tired light component of the engine calculates continuous energy attenuation as a function of the distance traveled through the intergalactic medium.[1] Let E_0 be the initial photon energy at emission, and E_d be the energy at a distance d from the source.[23] If a photon transfers a small, constant fraction f of its energy to intergalactic matter for every unit of distance (light-year) traveled, its remaining energy after n light-years is [23]:

$$E_d = E_0 f^n$$

Taking the natural logarithm of both sides yields [23]:

$$\ln\left(\frac{E_d}{E_0}\right) = n\ln(f)$$

Since the distance d is related to the integer number of light-years by $d = n \times (\mathrm{ly})$, we substitute to obtain [23]:

$$\ln\left(\frac{E_d}{E_0}\right) = \left[\frac{\ln(f)}{\mathrm{ly}}\right] d$$

Since f represents a fraction ($0 < f < 1$), the term $\ln(f)$ is negative.[23] We define a positive attenuation coefficient β with units of $(\mathrm{ly})^{-1}$ [23]:

$$\beta = -\frac{\ln(f)}{\mathrm{ly}}$$

Substituting β into the logarithmic equation yields the exponential energy decay relation [23]:

$$E_d = E_0 e^{-\beta d}$$

To express this decay in terms of transit time t, we substitute $d = ct$ [23]:

$$E_t = E_0 e^{-(\beta c)t}$$

We define the attenuation time constant as $\tau = \beta c$, which governs the rate of energy decay over time [23]:

$$E_t = E_0 e^{-\tau t}$$

The attenuation half-time $\theta_{1/2}$ —the time required for a photon to lose half of its energy—is written as [23]:

$$\theta_{1/2} = \frac{\ln(2)}{\tau} \approx \frac{0.692}{\tau} = \frac{0.692}{\beta c}$$

The cosmological redshift z_G produced by this tired light decay is defined by the wavelength ratio [23]:

$$1 + z_G = \frac{\lambda_d}{\lambda_0} = \frac{E_0}{E_d} = e^{\beta d}$$

Because this energy attenuation is frequency-independent, it redshifts all spectral lines uniformly, preserving the blackbody profile of the cosmic microwave background and preventing color-dependent dispersion.[21]

Cosmological Variable	**Standard ΛCDM Cosmology**	**Modernized CCC+TL Static Cosmology**
Redshift Mechanism	Metric expansion of the spatial manifold [1]	Covarying coupling constants + tired light energy decay [1]
Universe Age	$13.8 \times$ years [20]	$26.7 \times$ years [1]
Fine-Structure Constant	Assumed strictly static over cosmic time [1]	α remains strictly static; e, c, and h covary [1]
Speed of Light	Strictly static in vacuum (c_0) [1]	Decays over cosmic time ($c(t) \propto$) [1]
Particle Masses	Strictly static [1]	Scale inversely with the speed of light ($m(t) \propto$) [1]

Proca Electrodynamics and Fluid Media Subroutines

To simulate flat-space gravitational lensing and resolve cosmological time dilation without metric expansion, the engine replaces Maxwellian electrodynamics with Proca's massive vector field equations.[1]

Proca Vacuum Dispersion and Supernova Time Dilation

Under standard Maxwellian electromagnetism, the photon is massless ($m_\gamma = 0$), meaning all electromagnetic waves propagate at a constant velocity c.[1] In the Proca formulation, the photon is assigned a non-zero rest mass m_γ, which modifies the vacuum dispersion relation.[1]
The velocity $v(E)$ of a Proca photon is dependent on its energy E [1]:

$$v(E) = c\sqrt{1 - \left(\frac{m_\gamma c^2}{E}\right)^2}$$

As a polychromatic wave packet (such as the light emitted from a distant supernova) travels across the universe, its distinct frequency components propagate at slightly different velocities.[1] This velocity differential physically stretches the wave packet along the line of sight.[1]
The accumulated width $w(d)$ of a wave packet that has traveled a distance d is stretched in proportion to its redshift [1]:

$$w(d) = w_0(1 + z)$$

This massive photon vacuum dispersion reproduces the observed kinematic time dilation of Type Ia supernova light curves in a static universe, matching the observational data that rule out classical, non-dispersive tired light models.[1]

Flat-Space Gravitational Lensing and Spin Deflection

Rather than modeling gravitational lensing as light following null geodesics in warped spacetime, the engine treats lensing as a vector field scattering interaction in flat coordinates.[1]
The effective path of light passing through a gravitational potential $\Phi(r) = -GM/r$ can be mapped to propagation through an optical medium with a spatially varying refractive index $n(r)$ [28]:

$$n(r) = 1 - \frac{2\Phi(r)}{c^2} = 1 + \frac{2GM}{c^2 r}$$

Under standard Newtonian optics, treating a photon as a simple massive scalar particle yields a deflection angle of $\theta_{\text{Newton}} = 2GM/c^2 b$, which is exactly half of the observed relativistic deflection.[1]
By modeling the photon as a massive spin-1 vector field under Proca electrodynamics, the spin-gravitational coupling of the vector potential A_μ with the gravitational field tensor introduces a spin-dependent correction to the trajectory.[25]
Under the eikonal approximation, the flat-space deflection angle $\Delta\phi$ of a Proca photon is

written as [29]:

$$\Delta\phi \approx \frac{2r_0}{r_m}\left(1 + \frac{\hbar}{r_m c m_\gamma}\right)$$

where r_m is the radius of closest approach.[29] To align with precision gravitational lensing observations, this spin-dependent modification must satisfy the constraint [29]:

$$1 \gg \frac{\hbar}{r_m c m_\gamma}$$

For highly relativistic photons, this vector coupling matches the observed relativistic deflection angle of $4GM/c^2b$ in flat space, providing a physical scattering mechanism that replaces the geometric concept of metric warping.[1]

Plasma Interactions and Electromagnetically Induced Transparency

To simulate propagation through real intergalactic environments, the engine incorporates fluid medium subroutines [1]:

- **Microscopic Compton Scattering Subroutine:** Simulates discrete collision events where photons scatter off free electrons in intergalactic plasma.[1] This transfers kinetic energy to recoiling electrons and shifts wavelengths, generating a Dispersion Measure proportional to redshift.[1]
- **Electromagnetically Induced Transparency (EIT) Subroutine:** Models quantum interference within dense plasma regions.[1] Using Rabi frequencies and coupling constants, the subroutine maps electromagnetic wave packets into slow-moving "Dark-State Polaritons," simulating localized reductions in the speed of light.[1]

Relational Space Travel and Matter Profiling

Within a static, non-expanding universe, objects traveling through the intergalactic medium do not experience the coordinate "stretching" of metric expansion.[1] Instead, the engine models physical resistance and energy-state transformations to govern spatial dynamics.[1]

Dynamic Drag Laws in a Static Medium

Any physical body (such as a simulated spacecraft) propagating through the intergalactic medium experiences retarding forces.[1] The engine models this resistance using two distinct dynamic drag laws [1]:

1. **Baryonic Drag Force (F_{baryonic}):** Caused by direct mechanical collisions with the warm-hot intergalactic gas and plasma.[1] It scales quadratically with velocity and is written as [1]:

$$F_{\text{baryonic}} = -\frac{1}{2}\rho_b C_d A v^2$$

where ρ_b is the local baryonic gas density, C_d is the drag coefficient, and A is the cross-sectional area of the body.[1]

2. **Radiative Drag Force ($F_{\text{radiative}}$):** Generated by the coherent scattering of the cosmic microwave background and ambient starlight.[1] It scales linearly with velocity, dominating primarily at relativistic or sub-atomic scales [1]:

$$F_{\text{radiative}} = -\frac{4}{3}\sigma_T U_{\text{rad}} A \left(\frac{v}{c}\right)$$

where σ_T is the Thomson scattering cross-section and U_{rad} is the radiation energy density.[1]

Thermodynamic Photon Condensation and Graviballs

Over cosmological timescales, propagating photons continuously lose kinetic energy via the tired light mechanism.[1] The engine models this energy decay as a thermodynamic transition.[1] When ancient, redshifted photons decay to extremely low energy states, they undergo a phase change.[1]
They freeze out into a cold, sub-luminal condensate of stable, non-relativistic bound states called "graviballs".[1] These massive, non-luminous graviballs accumulate over billions of years, physically constituting the cold dark matter substrate observed in galactic halos and clusters.[1]

Web UI Usability and Real-Time Diagnostics Architecture

To render these complex relational mechanics in an accessible, interactive web application, the simulation engine is designed around a dual-tier usability architecture.[1]

Dual-Tier Web Architecture

To accommodate both educational users and advanced physics researchers, the application implements a dual-tier web framework [1]:

- **Declarative Beginner Tier:** Novices can construct and manipulate physical scenes without writing code.[1] The engine exposes custom, zero-install HTML elements (such as <game-component>, <platform-component>, and <crate-component>) that leverage the browser's native DOM parser, allowing scenes to be defined using simple HTML markup.[1]
- **Low-Overhead Developer Tier:** Offers direct, low-overhead JavaScript/TypeScript API bindings.[1] Advanced users can configure solver subroutines, alter time-step integration, adjust the restitution and friction parameters of dynamic bodies, and directly monitor coordinate states.[1]

- **GPU-Accelerated Rendering:** Computations are performed inside a high-performance WebAssembly (Wasm) core compiled from C++ or Odin.[1] To maintain a smooth 60 frames-per-second refresh rate during dense calculations, the HTML5 <canvas> rendering pipeline offloads draw calls to the GPU using CSS 3D transforms (translate3d).[1]

Real-Time Visual Overlays and Diagnostics

To make abstract, non-relativistic and relational concepts intuitive, the engine overlays visual diagnostics directly onto the simulated bodies [1]:

- **Broadphase Spatial Overlays:** Renders interactive visual boundaries of the Axis-Aligned Bounding Box (AABB) trees and uniform spatial grids, demonstrating how the engine prunes non-colliding objects to optimize computation.[1]
- **Narrowphase Contact Geometry:** When rigid bodies (such as crates and bunny meshes) collide, the engine overlays geometric vectors in real time [1]:
  - **Red Arrows:** Represent the exact penetration depth and overlap distance.[1]
  - **Yellow Circles:** Mark the precise coordinates of contact points.[1]
  - **Blue Lines:** Trace the surface contact normals and force reflection axes.[1]
- **Force Propagation Mapping:** Displays green arrows scaling with linear velocity and orange curved vectors depicting angular velocity and torque.[1] The engine color-codes isolated mechanical islands in real time, demonstrating how forces propagate through stacked assemblies.[1]

Educational Subroutine Modules

The physics engine integrates interactive modules to demonstrate observational physical concepts [1]:

- **Stellar Parallax Module:** Users can interactively scale the baseline distance (such as Earth's orbit) and observe the resulting changes in stellar parallax.[1] The module displays how proper motion (arcseconds per year) and distance (parsecs) convert to tangential velocity, making scale-based astronomy intuitive.[32]
- **Plate Scale Module:** Demonstrates the trade-offs of astronomical detectors.[1] Users can adjust the focal ratio and CCD pixel size to see how shorter focal ratios maximize light collection per pixel while degrading the angular resolution (arcseconds per pixel), illustrating the limits of optical systems.[1]

Synthesis and Implementation Roadmap

To successfully implement this relational cosmology within the web physics engine and demo, the software architecture must execute several development phases:

1. **Deploy flat Weitzenböck coordinates:** Replace curved metric geodesics with a flat coordinate system governed by translational torsion. This allows gravitational forces to be computed using standard, high-performance Euclidean vector math.[1]
2. **Implement the slingshot JSON scene-graph:** Expose an editor to serialize rigid-body properties, contact parameters (friction, bounciness), and structural constraints.[4] This enables deterministic reload states for testing numerical stability.[5]
3. **Integrate the corrected entropic gravity solver:** Compute gravitational attraction

directly between baryonic mass nodes, scaled by the localized coupling parameter β.[1] Incorporate the corrected Milgrom constant $a_0 = cH_0/9 \approx 0.76 \times 10^{-10}\ \mathrm{m/s^2}$ to fit rotation curves without dark matter.[10]

4. **Deploy the CCC+TL decay cascade:** Redshift incoming light using the joint covarying constants and tired light formulas.[1] Compute continuous energy attenuation using the exponential decay subroutine ($E_d = E_0 e^{-\beta d}$).[23]
5. **Apply Proca massive photon dispersion:** Assign a non-zero rest mass to photons, using vacuum dispersion to stretch wave packets over distance.[1] This reproduces supernova time dilation and the $(1+z)^{-4}$ Tolman surface brightness relation in a static coordinate framework.[1]

Works cited

1. Alternative Physics Simulation Frameworks.md
2. Speed of Light and Rates of Clocks in the Space Generation Model of Gravitation, Part 1 - viXra.org, accessed May 28, 2026, https://vixra.rxiv.org/pdf/1406.0090v1.pdf
3. Tired light - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Tired_light
4. I Made a Physics Engine - YouTube, accessed May 28, 2026, https://www.youtube.com/watch?v=QlLy4QWIb44
5. Building a Physics Engine with C++ and Simulating Machines - YouTube, accessed May 28, 2026, https://www.youtube.com/watch?v=TtgS-b191V0
6. Physics Engine from Scratch - YouTube, accessed May 28, 2026, https://www.youtube.com/watch?v=BtJfHoxAc4w
7. Informational Entropic Gravity (IEG): Gravitational Dynamics as Informational Symmetry Emergent from Entropy - Preprints.org, accessed May 28, 2026, https://www.preprints.org/manuscript/202601.2277
8. Entropic gravity - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Entropic_gravity
9. Emergent gravity and the dark universe, accessed May 28, 2026, https://www.quantumuniverse.nl/emergent-gravity-and-the-dark-universe?pdf=1113
10. Comment on “Emergent Gravity and the Dark Universe” by Erik Verlinde - SciPost, accessed May 28, 2026, https://scipost.org/comments/5434/attachment
11. Comment on “Emergent Gravity and the Dark Universe” by Erik Verlinde - arXiv, accessed May 28, 2026, https://arxiv.org/html/1909.01734v3
12. The Gravity of the Situation: Erik Verlinde's Emergent Gravity and Its Implications for Dark Matter and Dark Energy - Pete SonEarth, accessed May 28, 2026, https://petesonearth.medium.com/the-gravity-of-the-situation-erik-verlindes-emergent-gravity-and-its-implications-for-dark-matter-941c8be99bdb
13. A MOND model applied to the rotation curve of galaxies - World Scientific Publishing, accessed May 28, 2026,

https://www.worldscientific.com/doi/10.1142/S0218271825500014
14. Rotation Curve Fitting, accessed May 28, 2026, https://astroweb.case.edu/ssm/mond/mondprocedure.html
15. A MOND model applied to the rotation curve of galaxies - arXiv, accessed May 28, 2026, https://arxiv.org/html/2502.20599v1
16. The MOND paradigm of modified dynamics - Scholarpedia, accessed May 28, 2026, http://www.scholarpedia.org/article/The_MOND_paradigm_of_modified_dynamics
17. Rotation Curve Fitting Model - arXiv, accessed May 28, 2026, https://arxiv.org/html/2310.04372v4
18. Dark Matter in Galaxies and Clusters - Status of Dark Matter in the Universe - Katherine Freese - Caltech, accessed May 28, 2026, https://ned.ipac.caltech.edu/level5/Sept17/Freese/Freese2.html
19. Cosmic Microwave Background Radiation within the Zwicky Tired Light Hypothesis - arXiv, accessed May 28, 2026, https://arxiv.org/html/2504.10510v1
20. (PDF) Testing CCC+TL Cosmology with Observed Baryon Acoustic Oscillation Features, accessed May 28, 2026, https://www.researchgate.net/publication/379007561_Testing_CCCTL_Cosmology_with_Observed_Baryon_Acoustic_Oscillation_Features
21. Tired Light Is Solved | by Michael Aaron Cody - Medium, accessed May 28, 2026, https://medium.com/@mac123456/tired-light-solved-7fb837b1aa6c
22. The evolution of various energy densities in the CCC+TL model plotted... - ResearchGate, accessed May 28, 2026, https://www.researchgate.net/figure/The-evolution-of-various-energy-densities-in-the-CCC-TL-model-plotted-against-the_fig1_400549105
23. The Tired-Light Hypothesis: Derivations of Basic Relations, accessed May 28, 2026, https://www.gsjournal.net/Science-Journals/Communications-Cosmology/Download/9433
24. Redshift - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Redshift
25. Dynamical and observational properties of weakly Proca-charged black holes - arXiv, accessed May 28, 2026, https://arxiv.org/html/2601.19409v1
26. Force Acting on the Photon. Elementary Theory and Astrophysical Implications - SCIRP, accessed May 28, 2026, https://www.scirp.org/journal/paperinformation?paperid=146055
27. Errors in Tired Light Cosmology, accessed May 28, 2026, https://www.astro.ucla.edu/~wright/tiredlit.htm
28. Gravitational Lensing as an Optical Framework for Modified Gravity Theories - arXiv, accessed May 28, 2026, https://arxiv.org/html/2604.06791v1
29. Closing a window for massive photons - SciSpace, accessed May 28, 2026, https://scispace.com/pdf/closing-a-window-for-massive-photons-3upv0qlei7.pdf
30. PhysicsEngine[v1.7.0] Demo : r/learnjavascript - Reddit, accessed May 28, 2026, https://www.reddit.com/r/learnjavascript/comments/1t7l22a/physicsenginev170_demo/
31. Parsec confusion - astronomy - Physics Stack Exchange, accessed May 28, 2026, https://physics.stackexchange.com/questions/762253/parsec-confusion

32. Re: parsecs and arcsecs - UCLA Physics & Astronomy, accessed May 28, 2026, https://www.physics.ucla.edu/wwwboard/voh/fall97_quarter/physics/3_Huffman/messages/6.html
33. How do I apply specific astronomy concepts I have learned to calculations? | Wyzant Ask An Expert, accessed May 28, 2026, https://www.wyzant.com/resources/answers/952462/how-do-i-apply-specific-astronomy-concepts-i-have-learned-to-calculations
34. Angular resolution - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Angular_resolution