Technical Specification for the ArcSecs Dual-Tier Physics and Cosmological Simulation Engine

The development of interactive web-based physics engines has historically forced a compromise between two conflicting paradigms: the accessibility required for educational platforms and the mathematical rigor demanded by computational physicists, astronomers, and software engineers.[1] By analyzing the ArcSecs web physics platform, this technical framework outlines how to design an engine that bridges this gap.[4] The engine achieves this by using a dual-tier architecture that exposes complex underlying parameters while providing intuitive visual representations.[1]

The name "ArcSecs" refers to the arcsecond, a fundamental unit of angular measurement equal to $1/3600$ of a degree.[7] This connection provides an opportunity to expand the engine beyond basic rigid-body dynamics.[1] By integrating astronomical optics [11], cosmological redshift theories [13], dark matter halo modeling [15], and relational coordinate-free spacetime emergence [5], the engine connects classical mechanical loops with the mathematical foundations of astrophysics and quantum cosmology.[5]

Front-End Frameworks and Web Engine Architecture

To maximize accessibility for beginners without sacrificing execution speed, the front-end must support a dual-tier execution loop.[1] Beginners require zero-install, declarative HTML elements that render immediate visual feedback.[3] In contrast, advanced users require low-overhead API bindings, deterministic execution, and configuration of the underlying solver mechanics.[2]

The beginner-facing layer is built using encapsulated Web Components, registering custom HTML elements such as <game-component>, <platform-component>, and <crate-component>.[3] This design leverages the browser's native DOM parser, allowing users to define scenes through HTML structures.[3] Behind these custom elements, a high-performance rendering pipeline uses an HTML5 <canvas> element.[3] To maintain smooth rendering, the DOM positions of these components are updated via CSS 3D transforms (translate3d), which offloads the drawing overhead to the GPU.[3]

[redacted]

For advanced users, the JavaScript DOM is bypassed entirely in favor of compiled execution pipelines.[2] The engine's core is compiled into WebAssembly (Wasm) using a C++ or Odin compiler toolchain.[1] This pipeline utilizes Raylib and Emscripten to enable high-performance calculations directly on the web.[2]

To support modular code architectures, the engine is packaged using modern TypeScript with .mts and .mjs extensions.[18] This approach ensures native ESModule compatibility across server-side Node.js or Deno runtimes and client-side web browsers without requiring heavy bundlers.[18] The performance of different web physics engines under stress testing highlights the need for this dual-tier architecture.

Engine Implementation	**Framework Language**	**Average Performance (60 Bodies)**	**Rendering Target**	**Extensibility and Constraint**

				Support
box2d.ts [19]	TypeScript [19]	$27\ \text{FPS}$ [19]	Canvas / WebGL	Comprehensive joint, spring, and contact solvers.
planck.js [19]	JavaScript [19]	$17\ \text{FPS}$ [19]	Canvas	Port of Box2D; robust but bound by JS runtime.
matter.js [19]	JavaScript [3]	$43\ \text{FPS}$ [19]	HTML5 DOM / Canvas [3]	Lacks advanced spring constraints; breaks under stress. [19]
ArcSecs Core [1]	Odin / C++ via Wasm [2]	$60+$ (Deterministic)	WebGL Context [2]	Exposed Gauss-Seidel solver, custom sub-stepping. [6]

Rigid-Body Collision Pipelines and Visual Diagnostics

A major challenge when designing an accessible yet advanced physics simulation is presenting complex spatial and numerical calculations in an understandable format.[1] The ArcSecs engine solves this by using real-time visual diagnostics that overlay mathematical constructs directly onto the simulated bodies.[6]

[redacted]

[redacted]

Broadphase Spatial Partitioning

To maintain interactive performance, the engine avoids checking every body against every other body, which has a quadratic computational complexity of $\mathcal{O}(N^2)$.[2] Instead, it uses a hierarchical spatial partitioning pipeline.[2] Beginners can toggle a visual overlay to view the Axis-Aligned Bounding Box (AABB) trees or spatial grid lines.[2] This visualization helps novices understand how the engine groups nearby bodies before calculating exact collisions.[2] Advanced users can directly modify the broadphase configuration. They can adjust the cell sizes of the uniform spatial grid [2] or configure the tree-rebuilding heuristic of the dynamic AABB tree.[10]

Narrowphase Collision and Contact Manifolds

Once the broadphase identifies potential collisions, they are sent to the narrowphase pipeline to calculate exact geometric overlaps.[1] The engine supports multiple colliders, including spheres, boxes (rectangular prisms), cylinders, static triangles, and planes.[1] For advanced configurations, the system uses two main narrowphase pipelines:

- **Separation Axis Theorem (SAT):** This method projected vertices along face normals to detect overlaps in box-to-box and box-to-sphere collisions.[1] The engine visually renders the calculated projection axes and identifies the "Separating Axis" where overlap is minimized.[1]
- **GJK and EPA Pipeline:** For arbitrary convex shapes, the engine uses the Gilbert-Johnson-Keerthi (GJK) algorithm to calculate the minimum distance between bodies in Minkowski space.[10] If an overlap is detected, the Expansion Polytope Algorithm (EPA) expands the Minkowski difference hull outward.[10] This process calculates the exact penetration depth vector and generates a contact manifold via face clipping.[10]

To make this understandable to beginners, the engine displays interactive vectors of these mathematical operations.[6] Red arrows represent the penetration depth, yellow circles mark the contact points, and blue lines show the surface contact normals.[6]

Numerical Integration and Constraint Resolution

Once the contact manifold is generated, the engine resolves the physical forces, torques, and constraints.[1] The rigid-body states are advanced over time using semi-implicit Euler integration.[1] This integration maps forces and torques to linear and angular velocities before updating positions and orientations [1]:

$$v_{t+\Delta t} = v_t + a_t \Delta t$$

$$x_{t+\Delta t} = x_t + v_{t+\Delta t} \Delta t$$

Resting contacts, spring connections, and joint constraints are formulated as a system of inequalities solved using an iterative projected Gauss-Seidel solver.[10] To make this numerical process understandable, the interface provides visual diagnostics that show how forces are resolved [6]:

- **Velocity and Force Vectors:** Real-time green arrows scale with the linear velocity of each body, while orange curved vectors represent angular velocity and torque.[1]
- **Constraint Island Coloring:** When multiple bodies contact each other, the engine groups them into isolated computational units called "islands".[6] By color-coding these islands in real time, the engine visually demonstrates how forces propagate through stacked objects.[6]
- **Fixed-Time Step Manipulation:** Novices often struggle to understand why simulation

physics break at high velocities.[18] The engine provides a step-by-step control panel.[6] Users can pause the simulation, manually execute a single frame, adjust the time-step duration (Δt), or configure the maximum sub-steps.[6] This interactive control demonstrates how numerical integration error grows when the time step is too large relative to the velocities involved.[1]

Astronomical Optics and Plate Scale Metrics

The term "arcsec" is a foundational unit of angular resolution in astronomical optics and stellar astrometry.[7] To bridge physics simulation with astronomical observation, the platform includes an interactive optics module.[11] This module simulates how light behaves across different optical apertures, focal lengths, and detector pixel dimensions.[11]

[redacted]

Trigonometric Parallax and Proper Motion

The arcsecond is historically defined through trigonometric stellar parallax, where the distance of one parsec ($1\ \mathrm{pc}$) is the distance at which an astronomical unit ($1\ \mathrm{AU}$) subtends an angle of exactly one arcsecond ($1''$).[7] This relationship is governed by the small-angle approximation [9]:

$$d = \frac{1\text{ AU}}{\tan(p)} \approx \frac{1}{p}\text{ parsecs}$$

where p is the parallax angle in arcseconds.[9]

The simulation lets users interactively adjust the baseline distance (e.g., Earth's orbit of 1 AU) and observe the apparent motion of a target star against distant, stationary background stars.[9]

By incorporating proper motion (μ) measured in arcseconds per year, the engine calculates the tangential velocity (v_t) of stellar bodies [23]:

$$v_t = \mu \cdot d$$

To convert this value to metric SI units (m/s), the engine computes the conversion factor between arcseconds, parsecs, and metric time [23]:

$$\mu\ [\text{arcsec/yr}] \cdot d\ [\text{pc}] \approx \left(\frac{2\pi}{360 \times 3600}\text{ rad}\right) \cdot (3.086 \times 10^{16}\text{ m}) \cdot \left(\frac{1}{3.156 \times 10^{7}\text{ s}}\right) \approx 4740\text{ m}$$

This calculation shows how a proper motion of one arcsec per year at a distance of one parsec corresponds to a tangential velocity of 4.74 km/s.[23] The engine's conversion database is structured to bridge these astronomical scales with standard metric SI units.[7]

Astronomical Unit Name	Symbol	Definition / Origin	Angular Value	Metric SI Equivalent
Arcsecond [7]	$1''$ or arcsec	$1/360($ of a degree [7]	$4.848 \times$ [22]	N/A (Pure Angular Unit) [7]
Astronomical Unit [7]	AU	Mean distance from Earth to Sun [22]	Subtends $1''$ at 1 pc [7]	$1.496 \times$ [8]
Parsec [7]	pc	Distance where 1 AU	Angle of $1''$ [22]	$3.086 \times$ [23]

		subtends $1''$ [7]		
Light-Year [7]	LY	Distance light travels in one Julian year	N/A	$9.461 \times$ [9]

Plate Scale Calculations

For advanced users studying astrometry and focal plane imaging, the engine simulates plate scale dynamics.[12] This system determines how many arcseconds of the sky map onto a physical millimeter of an image sensor [12]:

$$\text{Plate Scale [arcsec/mm]} = \frac{206265}{F\ [\text{mm}]}$$

where F is the focal length of the telescope, and 206265 is the number of arcseconds in one radian.[12] To translate this physical metric to pixel-level data on a modern charge-coupled device (CCD), the engine implements the standard camera plate scale formula [12]:

$$\text{Scale [arcsec/pixel]} = \frac{206.265 \times w\ [\backslash\text{mum}]}{F\ [\text{mm}]}$$

where w represents the physical width of a single pixel in micrometers.[12]

Beginners can interactively adjust the focal ratio (f-ratio) and pixel size to observe how these parameters affect image scale.[12] Fast focal ratios (small f-ratios) increase the light per pixel but cover more arcseconds per pixel, which reduces the spatial resolution.[12]
Advanced users can simulate high-precision research instruments by loading specific aperture and focal configurations into the optics panel.[11]

Astronomical Instrument	Aperture Diameter (D)	Operating Wavelength (λ)	Angular Resolution (θ)	Analytical Formula / Type
Multiplexed	$6.5\ \mathrm{m}$ [24]	$370 -$	$1.2\ \mathrm{arcsec}\ (\mathrm{Fil}$	Cassegrain

Survey Telescope (MUST) [24]		[24]	[24]	Focus with WFC [24]
Hubble Space Telescope (HST) [11]	$2.4\ \mathrm{m}$ [25]	$500\ \mathrm{nm}$ (Visible) [11]	$0.04\ \mathrm{arcsec}$ [11]	Diffraction-Limited Space Telescope [11]
James Webb Space Telescope (JWST) [11]	$6.5\ \mathrm{m}$	$2000\ \mathrm{nm}$ (Infrared) [11]	$0.10\ \mathrm{arcsec}$ [11]	Segmented Space Telescope [11]
Very Large Telescope / PIONIER [11]	Equivalent Array	$1.0 -$ [11]	$0.001\ \mathrm{arcsec}$ ([11]	Astronomical Interferometer [11]
Global mm-VLBI Array [11]	Earth-Scale Array	$1.3\ \mathrm{cm}$ (Radio) [11]	$0.000012\ \mathrm{arcs}$ [11]	Very Long Baseline Interferometry [11]

Cosmological Models: Expanding Space versus Static "Tired Light"

To provide a playground for advanced theoretical physics, the engine includes a cosmological wave simulation module.[14] This module lets users contrast standard expanding FLRW cosmologies with historical alternatives, such as Fritz Zwicky's 1929 "tired light" proposal.[14]

[redacted]

```
Source * ===(Energy Lost to Space Resistance)===> [Observer]
- Photon energy decreases exponentially: E = E0 * e^(-d/R)
- No cosmic expansion or time dilation: dt_obs = dt_src
- Surface Brightness dims gently: I ~ (1 + z)^-1
```

Pure Kinematics vs. Relativistic Dynamics

The engine lets users compare different expansion dynamics.[27] For example, users can toggle the kinematics of the Edward Arthur Milne model against standard Friedmann-Lemaître-Robertson-Walker (FLRW) dynamics.[27]

The Milne model describes a kinematically expanding universe of non-interacting test particles moving at constant velocity.[27] This kinematic model produces an apparent linear relationship between velocity and distance, mimicking Hubble's law ($v = H_0 \cdot d$) without requiring gravitational curvature or dynamic dark energy.[27]

Advanced users can toggle this kinematic model to observe how it compares to standard general relativistic expansion.[27] In the general relativistic model, expansion rates are dynamically determined by the matter, radiation, and dark energy densities of the Friedmann equations.[27]

The Tolman Surface Brightness Test

The core visual tool in this module is the Tolman surface brightness test.[13] Surface brightness (I) measures the received flux density per unit solid angle of an extended astronomical source.[12] The engine calculates the apparent surface brightness of simulated galaxies as they are moved to high redshifts (z) [13]:

- **Expanding Universe (FLRW Metric):** The surface brightness of an expanding universe dims by a factor of $(1+z)^4$ when using bolometric VEGA magnitudes [13]:

$$I_{\text{expanding}} \propto \frac{1}{(1+z)^4}$$

This dimming is caused by two physical mechanisms.[13] First, individual photon energies decrease by $(1+z)$ due to redshifting.[13] Second, the photon arrival rate decreases by another factor of $(1+z)$ due to relativistic time dilation.[13] The remaining $(1+z)^2$ factor comes from the aberration of light, which makes the distant object appear larger than its physical size would suggest.[13] When using flux density per unit frequency (AB magnitudes), the relation simplifies to dimming by a factor of $(1 +$.[13]

- **Static Universe with Tired Light:** In Zwicky's tired light model, the redshift is not caused by metric expansion.[14] Instead, photons lose energy exponentially through interactions with the "resistance of space" over distance d [14]:

$$E(d) = E_0 e^{-\frac{d}{R}}$$

Because the universe is static, there is no cosmic time dilation or expansion-based angular distortion.[13] Consequently, the surface brightness dims by only a single factor of $(1 + z)$ under standard VEGA magnitudes [13]:

$$I_{\text{static}} \propto \frac{1}{1 + z}$$

When measured in frequency-specific AB magnitudes, the surface brightness of a static universe remains completely constant, regardless of distance or redshift.[13]

Physical Falsification through Blurring

To show why the pure tired-light model is physically falsified, the engine simulates wave-particle scattering.[14] If the energy loss in a tired light model were caused by physical scattering (such as the Compton or Raman effects) with intergalactic dust or plasma, the momentum transfer would deflect the photons' paths.[14]

The engine uses a particle diffusion simulation to demonstrate this effect.[14] As photons propagate through a scattering medium, their directions scatter, causing distant point sources to appear blurry.[14]

The engine compares this simulated blurring with actual data from the Hubble Ultra Deep Field (HUDF).[13] In the real universe, galaxies at redshifts $z > 5$ remain perfectly sharp, which rules out simple scattering as a mechanism for cosmic redshift.[13]

Cosmological Property	**Expanding FLRW Universe Model**	**Static Tired-Light Model (Zwicky)**	**Hybrid CCC + Tired-Light Model**

Underlying Physical Mechanism	Metric expansion of spacetime. [14]	Photons lose energy via spatial resistance. [14]	Covarying Coupling Constants + energy decay. [26]
Time Dilation of Distant Events	Present (proportional to $1 +$). [14]	Absent (static time). [14]	Present (governed by metrics). [26]
Surface Brightness Dimming (AB)	Dims as $(1 +$. [13]	Constant over distance z. [13]	Fits Pantheon+ and JWST data. [26]
CMB Isotropy Explanation	Naturally explained via cooled plasma. [26]	Fails to explain isotropic background. [26]	Partially compliant with hybrid metrics. [26]
Visual Resolution (Sharpness)	Retains sharp, unscattered profiles. [14]	Suffers scattering-induced blur. [14]	Retains sharpness (assumes no scattering). [26]

Gravitational Lensing, Dark Matter Halos, and Spacetime Emergence

At the intersection of general relativity and cosmology, the engine includes a module for simulating gravitational lensing and dark matter halo dynamics.[16] This system models how light deflections can map invisible mass distributions in the universe.[16]

[redacted]

I

Gravitational Lensing and Lensing Profiles

The lensing engine simulates how light rays from a background source (such as an elliptical core-Sérsic galaxy) are deflected by a foreground lens.[16] Advanced users can configure several mathematical profiles for the lens [16]:

- **Singular Isothermal Sphere (SIS):** This profile assumes a flat velocity dispersion, projecting a mass distribution that models typical galaxy-scale lenses.[16]
- **Truncated Navarro-Frenk-White (tNFW) Profile:** This profile models the density distribution of cold dark matter halos.[16] It is parameterized by the halo scale radius (r_s), the truncation radius (τ), and the concentration parameter (c_{CDM}).[16]

The simulation calculates the Einstein Radius (θ_E) to determine where multiple images or Einstein arcs form.[16]
The engine's rendering panel uses a ray-tracing shader to compute these deflections.[16] Beginners can drag the dark matter halo across the screen to observe how the background galaxy stretches and warps into arcs in real time.[16]

Dark Matter Halo Shapes and the Inertia Tensor

To analyze the shapes of dark matter structures, the advanced interface exposes the calculations for the halo shape inertia tensor.[15] Dark matter halos are modeled as triaxial ellipsoids with principal semi-axes lengths of $a > b > c$.[15] The components of the shape inertia tensor (I_{ij}) are computed from the spatial coordinates ($x_{i,n}$) of the halo's simulated particles [15]:

$$I_{ij} = \sum_n x_{i,n} x_{j,n}$$

Diagonalizing this tensor yields eigenvectors that define the orientations of the principal axes, while the eigenvalues determine the axial ratios c/a (minor-to-major) and b/a (semimajor-to-major).[15]
The engine uses these calculations to demonstrate structural differences between galactic scales and massive galaxy clusters.[15]

Halo Simulation Class	Median Axial Ratio c/a (Sphericity)	Median Axial Ratio b/a (Elongation)	Typical Mass Bounds (M⊙)	Structure and Composition
Milky Way-Mass Halo [15]	$0.74 \pm$ [15]	$0.85 \pm$ [15]	$\sim$ [30]	More spherical; contains dwarf satellite halos. [15]
Cluster-Mass Halo [15]	$0.67 \pm$ [15]	$0.77 \pm$ [15]	$\geq$ [30]	Less spherical, highly triaxial ellipsoids. [15]

By adjusting the telescope's angular resolution, users can see how resolving power affects our ability to observe these dark matter structures.[30] Improving the resolution from 9 to 4.5 arcseconds can increase the detection of faint, dwarf satellite galaxies by up to six times, helping map the low-mass limits of dark matter halos.[30]

The Radial Acceleration Relation (RAR)

To explore alternative gravity models, the engine includes a module for simulating the Radial Acceleration Relation (RAR).[17] The RAR connects the observed acceleration (g_{obs}) derived from rotation curves with the acceleration calculated purely from visible baryonic mass (g_{bar}) [17]:

$$g_{\text{obs}} = \frac{g_{\text{bar}}}{1 - e^{-\sqrt{g_{\text{bar}}/g_0}}}$$

where $g_0 \approx 1.2 \times 10^{-10} \text{ m/s}^2$ is a fundamental acceleration scale.[17]

At high accelerations ($g_{\text{bar}} \gg g_0$), the relation is one-to-one, meaning baryonic matter accounts for all the observed dynamics.[17] At low accelerations ($g_{\text{bar}} \ll g_0$), the observed acceleration flattens to a power-law slope of approximately 0.5, demonstrating the classic flat rotation curves discovered by Vera Rubin and Albert Bosma.[17]

Users can toggle between standard Newtonian gravity with dark matter halos and modified gravity models (such as MOND) to see how each model matches observed galaxy rotation curves.[17]

[redacted]

Atemporal Graph Projections and Spacetime Emergence

To align with the theoretical physics essays published on the ArcSecs domain, the platform includes a quantum cosmology sandbox.[4] This module explores the concept that physical spacetime is not a fundamental background.[5] Instead, spacetime can emerge from a non-spatiotemporal Hilbert space ($\mathcal{H}$) composed of tensor products of smaller spaces [5]:

$$\mathcal{H} = \bigotimes_i \mathcal{H}_i$$

[redacted]

[redacted]

Quantum Entanglement and Emergent Geometry

In this sandbox, the traditional coordinate-based simulation window is replaced by an atemporal graph $G = (V, E, \omega)$.[31] Nodes (V) represent physical qubits, which are not defined by spatial coordinates but by quantum states like energy or angular momentum.[5]

Edges (E) represent the quantum entanglement or mutual information between these qubits.[5] To make this concept understandable to beginners, the engine uses a spring-mass visualizer.[5] Qubits with high entanglement are pulled closer together by strong spring forces, while unentangled qubits drift apart.[5]
As users draw entanglement links between nodes, the graph dynamically clusters into recognizable shapes.[5] This clustering visually demonstrates how a continuous, multi-dimensional space can emerge from non-spatial quantum information.[5]

Quantum Error Correction and Gravity

For advanced users, this module simulates spacetime as a quantum error-correcting code.[5] In this framework, stable macroscopic "logical qubits" are constructed from entangled networks of "physical qubits".[5]
The engine lets users simulate local perturbations on the physical qubits.[5] If the entanglement network is robust, the logical spacetime metric remains stable, demonstrating error tolerance.[5]
Advanced users can also explore models where dark matter is simulated as a projection effect.[31] In these models, what we observe as dark matter is represented as the gravitational "shadow" of structures in neighboring planes of the atemporal graph, projected into our local spacetime coordinate system.[31]

Technical Recommendations for the ArcSecs Engine

To successfully implement these dual-tier physics and cosmological modules, the development of the ArcSecs platform should focus on several key areas:

- **Implement a Fixed-Update/Multi-Threaded Loop:** To prevent high-velocity bodies from clipping through boundaries, the physics engine should run at a high, deterministic frequency (such as $120\ \mathrm{Hz}$ or $240\ \mathrm{Hz}$) on a background Web Worker.[3] The rendering pipeline on the main browser thread should run independently, interpolating positions between physics steps to match the screen's refresh rate.[6]
- **Use Shader-Based Ray-Tracing for Lensing and Wave Optics:** Calculating gravitational lensing and light deflection across millions of pixels is computationally heavy.[16] These calculations should be offloaded to WebGL Fragment Shaders.[2] This allows the platform to render complex cosmological phenomena, like Einstein rings, at fluid frame rates.[16]
- **Build Context-Aware Mathematical Help Overlays:** The engine's user interface should dynamically adapt to user actions.[6] When a user selects a spring constraint, the platform should display the spring's physical parameters alongside its equations.[20] Clicking on these equations should open an interactive panel where advanced users can directly modify the solver's math.[6]
- **Connect Theoretical Modules to Real Astronomical Datasets:** To make the advanced modules more engaging, the engine should let users load actual cosmological data.[26] For example, users should be able to load galaxy rotation curves from the SPARC dataset to test gravity models, or import redshift-distance data from the Pantheon+ supernova dataset to analyze cosmic expansion.[26]

By combining high-performance web engineering with interactive, educational visual diagnostics, the ArcSecs engine can serve as a powerful tool for a wide audience.[1] It can provide a clear and intuitive introduction to physics for beginners while offering a rigorous, mathematically detailed simulation sandbox for advanced researchers.[6]

Works cited

1. Physics Engine in Odin from Scratch, Part I - Marian Pekár | Blog, accessed May 28, 2026, https://marianpekar.com/blog/physics-engine-in-odin-from-scratch-part-i
2. 2D C++ Physics Engine - Fabian Huber, accessed May 28, 2026, https://fabianhbr.ch/projects/physics-engine
3. Adding physics to web components - DEV Community, accessed May 28, 2026, https://dev.to/eerk/adding-physics-to-web-components-4kh2
4. Plato and Nietzsche - Leiter Reports, accessed May 28, 2026, https://leiterreports.com/2026/05/12/plato-and-nietzsche/
5. Spacetime does not exist - Leiter Reports, accessed May 28, 2026, https://leiterreports.com/2026/04/23/spacetime-does-not-exist/
6. Springs demo - p2-es physics engine, accessed May 28, 2026, https://p2-

es.pmnd.rs/demos/springs.html
7. 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
8. Angular diameter - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Angular_diameter
9. Calculating the Distance to Nearby Stars: The Stellar Parallax, accessed May 28, 2026, https://aaa.org/2023/07/01/calculating-the-distance-to-nearby-stars-the-stellar-parallax/
10. RomanPPP/TypeScript-Physics-Engine - GitHub, accessed May 28, 2026, https://github.com/RomanPPP/TypeScript-Physics-Engine
11. Angular resolution - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Angular_resolution
12. Ast 401/Phy 580 Fall 2015 - Lowell Observatory, accessed May 28, 2026, http://www2.lowell.edu/users/massey/6_Telescopes.pdf
13. UV surface brightness of galaxies from the local universe to z ~ 5 - ResearchGate, accessed May 28, 2026, https://www.researchgate.net/publication/262071666_UV_surface_brightness_of_galaxies_from_the_local_universe_to_z_5
14. Tired light - Wikipedia, accessed May 28, 2026, https://en.wikipedia.org/wiki/Tired_light
15. Subhaloes are anisotropically distributed and aligned with the smooth matter distribution of their host haloes - Oxford Academic, accessed May 28, 2026, https://academic.oup.com/mnras/article/538/2/963/8030530
16. A forward-modelling method to infer the dark matter particle mass from strong gravitational lenses - Oxford Academic, accessed May 28, 2026, https://academic.oup.com/mnras/article/511/2/3046/6516982
17. MIGHTEE-HI: The radial acceleration relation with resolved stellar mass measurements, accessed May 28, 2026, https://arxiv.org/html/2504.20857v1
18. Physics Engine 13x Faster than Matter.js : r/typescript - Reddit, accessed May 28, 2026, https://www.reddit.com/r/typescript/comments/14ogysu/physics_engine_13x_faster_than_matterjs/
19. Javascript/Typescript Physics Engines - HTML5 Game Devs Forum, accessed May 28, 2026, https://www.html5gamedevs.com/topic/26608-javascripttypescript-physics-engines/
20. 15-462 Project 5: Simple Physics Engine, accessed May 28, 2026, https://www.cs.cmu.edu/afs/cs/academic/class/15462-s12/www/project/p5.pdf
21. Parallax, accessed May 28, 2026, https://ircamera.as.arizona.edu/NatSci102/NatSci/text/parallax.htm
22. Parsec confusion - astronomy - Physics Stack Exchange, accessed May 28, 2026, https://physics.stackexchange.com/questions/762253/parsec-confusion
23. 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
24. MUltiplexed Survey Telescope: Perspectives for Large-Scale Structure Cosmology in the Era of Stage-V Spectroscopic Survey - arXiv, accessed May 28, 2026, https://arxiv.org/html/2411.07970v3
25. The Galileo of Palomar - Halton Arp, accessed May 28, 2026, http://haltonarp.com/inc/memorial/TheGalileoOfPalomar.pdf
26. JWST early Universe observations and ΛCDM cosmology | Monthly Notices of the Royal Astronomical Society | Oxford Academic, accessed May 28, 2026, https://academic.oup.com/mnras/article/524/3/3385/7221343?login=false
27. Is "Hubble-Independent Distance Calculator to Cosmological Objects" article a rebranding of Tired Light Model? | ResearchGate, accessed May 28, 2026, https://www.researchgate.net/post/Is_Hubble-Independent_Distance_Calculator_to_Cosmological_Objects_article_a_rebranding_of_Tired_Light_Model
28. AaS! 267: Is the Universe Older than We Think? - Paul M. Sutter, accessed May 28, 2026, https://www.pmsutter.com/shows/askaspaceman-archive/2026/03/03/aas-267-is-the-universe-older-than-we-think
29. Testing CCC+TL Cosmology with Galaxy Rotation Curves - MDPI, accessed May 28, 2026, https://www.mdpi.com/2075-4434/13/5/108
30. Hunting for the faintest galaxies | astrobites, accessed May 28, 2026, https://astrobites.org/2017/11/29/hunting-for-the-faintest-galaxies/
31. What if what we call “dark” components are not new particles but inter projection effects: shadows of massive structures in neighbouring planes of the atemporal graph. : r/HypotheticalPhysics - Reddit, accessed May 28, 2026, https://www.reddit.com/r/HypotheticalPhysics/comments/1sosr6c/what_if_what_we_call_dark_components_are_not_new/