Interim Report — High-Precision N-Body Solar System Gravitational Simulation Simulator

Nik Szczepanski and Marlow Lichty

Problem definition

There are few free, open-source N-body gravitational simulators optimized for consumer hardware that combine state-of-the-art integration algorithms with an interactive three-dimensional user interface. Our goal is to deliver an MIT-licensed Rust application that provides high-fidelity solar-system simulation by combining long-term symplectic evolution with adaptive, high-order precision for close encounters (Rein & Tamayo, 2015; Rein & Spiegel, 2014).

Computational plan & UI

We implement a coarse/fine integrator architecture: WHFast (a fast Wisdom–Holman symplectic scheme) for efficient long-term evolution, and IAS15 (an adaptive, 15th-order Gauss–Radau integrator) for high-precision phases such as close encounters or non-conservative forces (Rein & Tamayo, 2015; Rein & Spiegel, 2014). Key collocation and Runge–Kutta coefficients will be computed at extended precision at build time and cached; runtime arithmetic will use double precision for performance. Initial conditions are sourced from NASA JPL's HORIZONS ephemerides and simulated states will be validated against matched HORIZONS epochs (NASA Jet Propulsion Laboratory, n.d.). Visualization and user interaction are implemented with the Bevy engine, with nalgebra supplying linear-algebra primitives and transformations for the physics and rendering layers (Bevy Engine, n.d.; Nalgebra Developers, n.d.).

Progress to date

A Bevy test environment is complete and has been used to exercise rendering, camera controls, and the I/O pipeline. The literature review covering WHFast, IAS15, and reference implementations (e.g., REBOUND) is finished. Implementation of the integrator designs has begun in Rust using nalgebra for vector and matrix operations. A prototype pipeline for retrieving and parsing ephemerides from JPL HORIZONS is also in development.

Validation & accuracy plan

We will validate by running matched-epoch comparisons against HORIZONS vector outputs and report RMS position error, maximum absolute error, relative energy drift, and angular-momentum conservation.

Integrator control & switching

A control layer will switch from the coarse to fine integrator when encounter metrics exceed thresholds (e.g., distance-to-body less than a factor of the Hill radius or an adaptive error estimator above tolerance). Hand-offs will synchronize state and preserve consistency to minimize accumulated numerical error.

Expected results

Deliverables will include a working Rust codebase (MIT Licensed), WHFast and IAS15 implementations, a Bevy visualization with controls and diagnostics, ephemeris ingestion tooling, and validation reports comparing results to JPL HORIZONS and existing codes. Success criteria are meeting the stated accuracy benchmarks, reproducibility of test cases, and responsive visualization on consumer hardware.

References

Rein, H., & Tamayo, D. (2015). WHFast: A fast and unbiased implementation of a symplectic Wisdom–Holman integrator for long-term gravitational simulations. Monthly Notices of the Royal Astronomical Society, 452(1), 376–388. https://doi.org/10.1093/mnras/stv1257

Rein, H., & Spiegel, D. S. (2014). IAS15: A fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. Monthly Notices of the Royal Astronomical Society, 446(2), 1424–1437. https://doi.org/10.1093/mnras/stu2164

Rein, H., & Liu, S.-F. (2012). REBOUND: An open-source multi-purpose N-body code for collisional dynamics. Astronomy & Astrophysics, 537, A128. https://arxiv.org/abs/1110.4876

NASA Jet Propulsion Laboratory. (n.d.). Horizons system (ephemerides service). Jet Propulsion Laboratory, California Institute of Technology. https://ssd.jpl.nasa.gov/horizons/app.html

Bevy Engine. (n.d.). Bevy — A refreshingly simple data-driven game engine for Rust. https://bevy.org/

Nalgebra Developers. (n.d.). nalgebra — Linear algebra library for Rust. https://nalgebra.org/