Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Mean-element Parareal: Parallel-in-Time Integration of the Solar System

Michael Hippke

michael@hippke.org

Sonneberg Observatory

Research Article

Keywords: Celestial mechanics, Numerical methods, Planetary dynamics and stability, N-body simulations

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11578737/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Mean-element Parareal: Parallel-in-Time Integration of the Solar System

Michael Hippke[1*]

[1*]Sonneberg Observatory, Sternwartestr. 32, Sonneberg, 96515, Germany.

Corresponding author(s). E-mail(s): michael@hippke.org;

Abstract

Long-term integrations of planetary systems have long been considered inherently sequential: each time step must start from the *exact* state at the end of the previous one, because any error in that state, however small, is amplified by orbital shear and chaos and alters the trajectory. Because a single step of a modern Wisdom–Holman integrator takes only ∼100 ns—too short to split across cores—additional cores have previously only been used to run multiple simulations in parallel. We introduce mean-element parareal (MEP), a parallel-in-time scheme that accelerates a single Solar System integration (eight planets, general relativity) at the same accuracy as the serial integrator. Combined with a latency-optimized C kernel that integrates 5 Gyr on one core in 5.8 h, MEP completes a 5 Gyr integration in 52 min on 64 cores (a parallel speed-up of 6.7×, and 10.4× faster than the previous 9.0 h single-core record) or in 1.8 h on 16 cores (3.3× parallel speed-up, 5.0× faster than the previous record). MEP succeeds because its coarse propagator uses a numerical first-order mean-element theory to remove the short-period oscillations from the semi-major axes, preventing Keplerian shear from amplifying them into growing phase errors. Standard parareal with a coarse Wisdom–Holman integrator fails for planetary orbits because no inexpensive coarse integrator can resolve these oscillations. To our knowledge, this is the first measured multicore speed-up of a single long-term planetary integration and opens a pathway towards larger-scale parallelization. MEP is freely available in Parachron, a modified version of `REBOUND`, at https://github.com/hippke/parachron.

Keywords: Celestial mechanics, Numerical methods, Planetary dynamics and stability, N-body simulations

1 Introduction

Studying the long-term dynamical evolution of the Solar System and of exoplanetary systems requires integrating the gravitational N-body problem over billions of orbits (e.g., Sussman and Wisdom 1988; Laskar 1989; Sussman and Wisdom 1992; Laskar 2008; Batygin and Laughlin 2008; Laskar and Gastineau 2009; Zeebe 2015; Abbot et al. 2023; Brown and Rein 2023; Rein et al. 2025). For these integrations, mixed-variable symplectic maps of the Wisdom–Holman (WH) type (Wisdom and Holman 1991; Kinoshita et al. 1991) have become the standard tool, and successive implementations have steadily reduced their computational cost: from `WHFast` (Rein and Tamayo 2015) in the `REBOUND` package (Rein and Liu 2012) to the AVX-512 vectorized `WHFast512` (Javaheri et al. 2023) and its hand-written assembly successor, which integrates the eight planets of the Solar System with general relativistic (GR) corrections for 5 Gyr in 9.0 h on a single core (Dagli and Rein 2026). As Dagli and Rein (2026) argue, this code already approaches the hardware limit of a single core on present-day CPUs.

With single-core performance near its ceiling, further speed-ups must come from parallel hardware, yet conventional parallelism has not accelerated a single planetary integration. Because every time step must start from the exact end state of the previous one, the integration is inherently sequential; moreover, a single WH step takes only ∼100 ns, comparable to the core-to-core latency of a CPU, leaving too little work within a step to distribute across cores. Distributing force evaluations across particles pays for large N (Grimm and Stadel 2014; Grimm et al. 2022; Zhang and Gladman 2020; Lau and Lee 2023), but for few bodies a 16-core CPU outperforms a GPU by at least a factor of ten (Grimm and Stadel 2026), and vectorization within a step is already what makes `WHFast512` fast (Javaheri et al. 2023). Parallel computers have therefore been confined to running independent trajectories in ensembles (e.g., Laskar and Gastineau 2009; Brown and Rein 2022; Abbot et al. 2023) or parameter surveys (e.g., Tamayo et al. 2016; Obertas et al. 2017; Tamayo et al. 2020), while the wall-clock time of any individual trajectory remains fixed by single-core speed. Averaged (secular) equations of motion run orders of magnitude faster than direct integration (Laskar 1990, 2008; Mogavero and Laskar 2021; Hoang et al. 2022), but only by approximating the dynamics. Many questions instead demand exact, ultra-long single trajectories: determining the dynamical half-life of Mercury over 100 Gyr (Hoang et al. 2022), constructing orbital solutions for astrochronology (Laskar et al. 2004; Zeebe 2023), probing slow chaotic diffusion (Mogavero et al. 2023), and establishing the numerical convergence of Solar System ensembles (Hernandez et al. 2021; Rein et al. 2025, 2026).

In this paper, we show that a single long-term integration of the Solar System can be parallelized in time on a many-core workstation. Our method, mean-element parareal (MEP), integrates the benchmark of Dagli and Rein (2026)—the Sun and eight planets with GR corrections at a 6 d time step—for 5 Gyr in 52 min on 64 cores and 1.8 h on 16 cores, compared to 5.8 h on a single core with our C kernel (speed-ups of 6.7 and 3.3, or 10.4× and 5.0× faster than the 9.0 h record of Dagli and Rein 2026) at the full accuracy of the serial integrator (Figure 1). Across billion-year integrations, the parallel trajectories preserve the energy and angular-momentum conservation, the secular frequencies, and the long-term dynamical statistics of the serial trajectory.

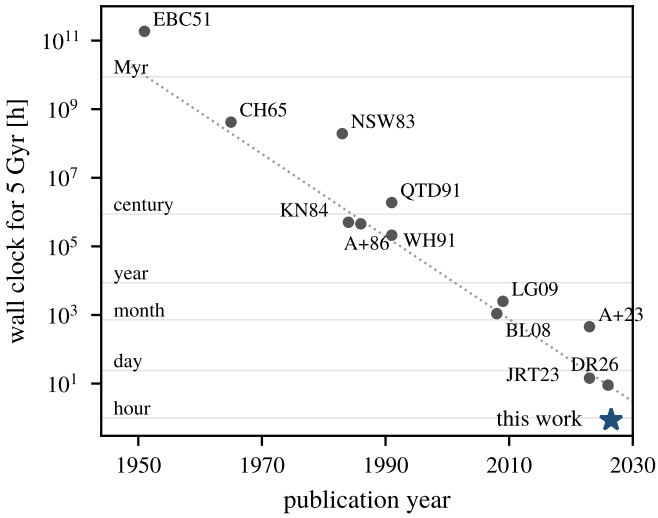


Fig. 1 Wall-clock time to integrate the eight planets of the Solar System for 5 Gyr, by publication year, normalized as in Dagli and Rein (2026, their figure 1): $t_{\mathrm{wall}}\,(5\,\mathrm{Gyr}/t_{\mathrm{sim}})\,(P_{\mathrm{min}}/88\,\mathrm{d})\,(8/N)^2$. Grey points: published integrations, compiled by S. Hadden (https://shadden.github.io/nbody_history/) and re-derived with this normalization (approximate); JRT23 is inferred from the statement of Dagli and Rein (2026) that their version of `WHFast512` is 60% faster than that of Javaheri et al. (2023). Dotted line: halving every 2.5 yr. This work (blue star): MEP at the same accuracy as the serial integrator, taking 52 min (0.9 h) on 64 cores (10.4× faster than the 9.0 h record of Dagli and Rein 2026, DR26) and 1.8 h on 16 cores (5.0× faster). Serial `WHFast512` with Parachron's C kernel takes 5.8 h on one core.

To parallelize along the time axis, parallel-in-time (PinT) methods, of which parareal (Lions et al. 2001) is the best known, divide the time span into slices, integrate many slices concurrently from approximate initial states, and iteratively correct the slice boundaries using a fast coarse propagator (see the reviews of Gander 2015; Ong and Schroder 2020). Although their application to planetary orbits has a long history, no wall-clock speed-up has previously been measured. Saha et al. (1997) noted that "simulations on serial machines suggest" a speed-up of about 50 on 1000 processors; Jiménez-Pérez and Laskar (2011) tested a refined scheme on model problems in a simulated parallel environment; Dai et al. (2013) reported a theoretical speed-up above 60 for the outer Solar System "provided a sufficient number of processors are available", requiring up to 15 iterations; and Gander and Hairer (2014) proved for the Kepler problem that rapid convergence demands a coarse solver of nearly the fine solver's accuracy. These difficulties reflect three broader obstacles documented in the PinT literature: parareal struggles with oscillatory problems because coarse phase errors stall convergence (Ruprecht 2018), loses the geometric properties of the fine solver unless symmetrized or projected (Bal and Wu 2008; Dai et al. 2013), and converges only over a limited interval for chaotic systems (Gander and Hairer 2008).

We identify the physical mechanism behind this barrier and show how to overcome it. For planetary orbits, parareal convergence is governed by how accurately the

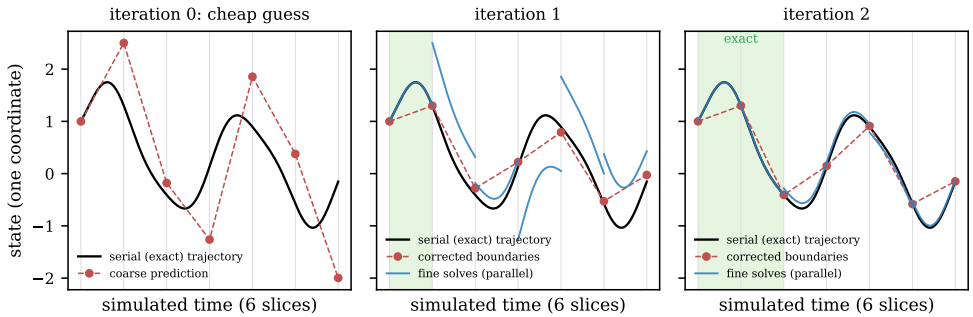


Fig. 2 Parareal on a toy problem with a one-dimensional state, six time slices and a coarse step of one Euler step per slice. Iteration 0: the coarse propagator predicts the slice boundaries (red), cheaply but inaccurately. Iteration $k \geq 1$: all six fine solves (blue) start in parallel from the boundaries of the previous iterate, and Equation (1) corrects the boundaries. Green: slices that are already exact. Two iterations suffice here to approach the serial trajectory (black) closely. In MEP the state has 48 dimensions (six orbital elements of each of eight planets), a window has 16 slices, the coarse step is the mean-element propagator of Section 2.4, and windows need 3–4 iterations.

coarse propagator resolves the *semi-major axes*. Through Keplerian shear, the dependence of mean motion on semi-major axis that disperses planetary rings and debris clouds along their orbits (e.g., Rein and Tremaine 2011), any discrepancy in the semi-major axis accumulates into a secularly growing phase error. However, the osculating semi-major axes are dominated by short-period fluctuations that no computationally inexpensive coarse integrator can resolve. For oscillatory problems in other fields, the most successful PinT remedies modify the coarse model rather than its time step: they align phases (Ariel et al. 2015), formulate micro–macro models in slow variables (Legoll et al. 2012), or average over the fast oscillations (Haut and Wingate 2013; Peddle et al. 2017). Adopting this principle, we build the first PinT coarse propagator based on a perturbation theory of celestial mechanics. By transforming to *mean* elements and removing the short-period terms, a low-cost coarse propagator meets the stringent semi-major-axis accuracy requirement. Perturbation theory thus steers the parallel convergence, while direct numerical integration guarantees that the converged result remains exact.

The remainder of this paper is organized as follows. Section 2 presents mean-element parareal, explains the physical mechanism that governs its convergence, and describes its parallel implementation. Section 3 validates the physical and statistical accuracy of the parallel trajectories, and Section 4 reports the measured wall-clock speed-up and scaling. Finally, Section 5 discusses when the method is advantageous and how it can scale to larger core counts. MEP is freely available for Python and C in Parachron, a modified version of `REBOUND`.

2 Mean-element parareal

2.1 Fine propagator and accuracy requirements

Our fine propagator F integrates the Sun and the eight planets in Jacobi coordinates using the Wisdom–Holman map of `WHFast512` (Javaheri et al. 2023; Dagli and Rein 2026) with a time step of $\Delta t = 6\,\mathrm{d}$, a 17th-order symplectic corrector, and the general relativistic (GR) potential of Nobili and Roxburgh (1986). Applied to a single time slice, F is a standard serial integration with this map. In Parachron, we port the hand-written AVX-512 assembly kernel of Dagli and Rein (2026) to C and shorten its critical instruction dependency chain in both the Kepler solver and the Jacobi coordinate transformations (Appendix A, Figure 7). Where AVX-512 is available, the compiled C kernel is faster than the assembly kernel (69 against 86 ns per step) and runs on any processor architecture with bitwise identical results; on CPUs without AVX-512, where `REBOUND` provides no `WHFast512`, it runs 44% slower than with AVX-512 but still six times faster than the older `WHFast` (Rein and Tamayo 2015). MEP works with either kernel, and all results in this paper use the C kernel. At the start and end of every slice, F maps between real and symplectic-corrector coordinates ($\sim$0.06% of the slice cost) and converts between inertial and Jacobi coordinates using recursive transformations rather than precomputed matrices, which eliminates a small energy offset per call (Appendix B) and otherwise alters trajectories only at the round-off level.

Because the Solar System is chaotic with a Lyapunov time of $\sim$5 Myr (Laskar 1989; Sussman and Wisdom 1992), two serial integrations that differ only in floating-point round-off decorrelate within $\sim$100 Myr (cf. Portegies Zwart and Boekholt 2014). Pointwise agreement with a single serial trajectory over Gyr timescales is therefore not a meaningful requirement. Instead, a valid parallel integration must satisfy four physical criteria: it must (i) introduce no systematic energy drift, (ii) keep energy and angular-momentum errors within the bounds of the serial integrator, (iii) reproduce the secular orbital frequencies to within the serial integrator's own time-step error, and (iv) preserve the statistical distribution of the long-term evolution, particularly Mercury's high-eccentricity excursions.

2.2 Parareal

Parareal divides a time span into slices of length T and combines the accurate fine propagator F with a fast, approximate coarse propagator G (Figure 2). Starting from an initial sequential coarse prediction $U_{n+1}^0 = G(U_n^0)$, iteration k advances the state U_n^k at the start of slice n according to

$$U_{n+1}^{k+1} = G\big(U_n^{k+1}\big) + F\big(U_n^k\big) - G\big(U_n^k\big)\,, \qquad U_0^k = U_0. \tag{1}$$

Within each iteration, all fine integrations $F(U_n^k)$ are independent and run in parallel across cores; only the coarse correction sweep is sequential. After k iterations, the first k slice boundaries match the serial fine trajectory exactly. Once the iteration converges ($U_n^{k+1} = U_n^k$), the coarse terms $G(U_n^{k+1})$ and $G(U_n^k)$ cancel, leaving the

exact serial trajectory $U_{n+1} = F(U_n)$ as the fixed point regardless of approximations in G: an inaccurate coarse model costs extra iterations, not final accuracy.

With pipelining across N slices (Section 2.6), K iterations, a fine slice cost T_F, and a coarse slice cost $T_G = rT_F$, the wall-clock speed-up is (Appendix C)

$$S \approx \frac{NT_F}{NT_G + K\,(T_F + T_G)} \leq \min\left(\frac{N}{K}, \frac{1}{r}\right). \tag{2}$$

Achieving a speed-up above 3 on $N = 16$ cores therefore requires both rapid convergence ($K \lesssim 4$) and a very inexpensive coarse propagator ($r \lesssim 0.03$). Because G enters Equation (1) only through the difference $G(U_n^{k+1}) - G(U_n^k)$, any state-independent offset in G cancels out; rapid convergence instead requires G to accurately capture how small perturbations at the start of a slice propagate to its end (Appendix D).

2.3 Why standard parareal fails: Keplerian shear

In planetary systems, the dominant sensitivity across a time slice is Keplerian shear. By Kepler's third law, a planet's mean motion scales with its semi-major axis as $n \propto a^{-3/2}$. A small fractional error $\delta a/a = \delta \ln a$ at the start of a slice of length T shifts the mean motion by $\delta n/n = -\frac{3}{2}\,\delta \ln a$, which accumulates linearly in time into an orbital phase error in the mean longitude λ:

$$\delta\lambda(T) = -\frac{3}{2}\,nT\,\frac{\delta a}{a} = -\frac{3}{2}\,nT\,\delta \ln a. \tag{3}$$

In MEP, each slice spans $T = 360\,000\,\mathrm{d} \approx 986\,\mathrm{yr}$ (6×10^4 fine steps), a duration long enough for each fine solve to amortize the overhead of the coarse step (Section 2.6). Over a single slice, Mercury completes ∼4100 orbits, and the shear amplification factor $\frac{3}{2}nT$ reaches 3.9×10^4 for Mercury, 1.5×10^4 for Venus, 9.3×10^3 for Earth, and 4.9×10^3 for Mars. A fractional semi-major-axis error of just 10^{-8} is thus amplified into a phase error of $\sim 4 \times 10^{-4}\,\mathrm{rad}$ within one slice. Perturbation tests of the slice map confirm that converging within $K \leq 3$ iterations requires the coarse propagator to track changes in $\ln a$ to an absolute accuracy of $\lesssim 10^{-8}$, whereas all other orbital elements tolerate relative errors of ∼1% (Appendix D).

No inexpensive coarse integrator operating in standard (*osculating*) coordinates can meet this 10^{-8} requirement. Mutual planetary perturbations cause the osculating semi-major axes to fluctuate on orbital and synodic periods with relative amplitudes of 10^{-6}–10^{-5}—up to three orders of magnitude larger than the allowable error. Reproducing these short-period oscillations to 10^{-8} requires resolving every orbit with small time steps and accurate phases, which is as expensive as the fine integration itself.

This requirement explains why the most natural coarse propagator—the same WH integrator run with a larger time step Δt_G—cannot accelerate the integration. Resolving Mercury's 88 d orbit restricts the coarse step to $\Delta t_G \lesssim 16\,\mathrm{d}$, which already costs at least $r = 0.42$ of the fine step. At the same time, coarse sampling aliases the short-period oscillations, corrupting the semi-major-axis sensitivities by 1–5% at $\Delta t_G = 16\,\mathrm{d}$ and by ∼80% at 24 d. Across ten 986 yr slices, a coarse WH propagator

therefore needs 3–4 iterations at $\Delta t_G = 8$–$16\,\mathrm{d}$ ($r = 0.42$–0.77) and fails to converge altogether at $\Delta t_G = 24\,\mathrm{d}$ ($r = 0.26$) or over ten times longer slices; by Equation (2), this yields $S = 0.3$–0.5, a net slowdown. Multilevel schemes with a Kepler-only coarse solver fail for the same reason: in the Solar System benchmark of the `pySDC` framework (Speck 2019), PFASST requires more than 15 iterations on average.

2.4 A hybrid coarse propagator in mean elements

To achieve 10^{-8} accuracy in the semi-major axes at a small fraction of the fine solver's cost, our coarse propagator G combines perturbation theory for the inner planets with direct coarse integration for the outer giants. For the *inner planets* (Mercury–Mars), G transforms the osculating Jacobi orbital elements into *mean* elements by subtracting the short-period oscillations before evolving the slice (Appendix E). For each pair of planets, we evaluate their mutual gravitational accelerations on a regular two-dimensional grid of their orbital phases (λ_i, λ_j) with all other orbital elements held fixed, convert these accelerations into element rates via Gauss' equations, and apply a two-dimensional fast Fourier transform (FFT). The zero-frequency Fourier mode gives the slow secular drift rate, while integrating the non-zero modes analytically over time yields the short-period displacements to subtract from the osculating elements. This procedure constitutes a numerical first-order Lie-series transformation (Brouwer 1959; Deprit 1969): because it evaluates perturbations on phase grids rather than expanding them in power series of eccentricity and inclination (Murray and Dermott 2000), it remains exact in e and i at first order in the planetary masses, which is essential for Mercury ($e \approx 0.2$). Removing these short-period terms reduces the scatter in $\ln a$ for the inner planets by factors of 300–2300—from 2×10^{-6} (Mercury) and 3×10^{-5} (Mars) in osculating elements down to 10^{-9}–3×10^{-8}—well below the threshold required for parareal convergence.

For the *giant planets* (Jupiter–Neptune), a first-order theory is insufficient because near-resonances—specifically the Jupiter–Saturn 5:2 great inequality (period $\approx 880\,\mathrm{yr}$) and the Uranus–Neptune 2:1 near-commensurability ($\approx 4200\,\mathrm{yr}$)—introduce small frequency divisors that produce large higher-order oscillations. However, because Jupiter's orbit is 50 times longer than Mercury's, the four giants can be integrated directly by the WH solver at a large time step of $360\,\mathrm{d}$ (with the inner planets' masses added to the Sun) at negligible cost. Over one slice, G transforms the inner planets to mean elements at the start, drifts them with their mean motions $n(\bar{a})$ and the secular forcing from the evolving giants, and transforms them back to osculating elements at the end.

Four design choices keep this coarse step both fast and unbiased. First, the Fourier tables are rebuilt from the current state at every coarse call because the short-period amplitudes depend on the eccentricities and inclinations; freezing the tables across iterations requires 6 iterations instead of 3 to reach the round-off floor. Second, the secular drift uses a symmetric trapezoidal (Heun) step rather than an Euler step, preventing one-sided coarse errors from biasing the secular frequencies. Third, the periodic transformation is applied only to $\ln a$ and λ, since the remaining orbital elements do not experience Keplerian shear, using 32×32 phase grids for the six inner–inner pairs and 16×16 grids for the eight pairs coupling an inner planet to

Jupiter or Saturn (short-period perturbations from Uranus and Neptune on the inner planets' semi-major axes scale as $(m_j/M_\odot)(a_i/a_j)^3 \lesssim 10^{-10}$, two orders of magnitude below the 10^{-8} threshold, while their secular forcing is included through the giants' integration). The close Venus–Earth pair (semi-major-axis ratio 0.72), whose mutual perturbations contain high harmonics of the synodic angle $\lambda_V - \lambda_E$ but only low harmonics in the other direction, is instead tabulated on a 16×64 grid of $(\lambda_V, \lambda_V - \lambda_E)$, which resolves synodic harmonics up to order 32 at the cost of a 32×32 grid (Appendix E). Fourth, the small frequency divisors of the transformation are evaluated with the mean motions of the mean rather than the osculating semi-major axes. With these choices, one coarse slice takes about 0.09 ms against 4–5 ms for a fine slice ($r \approx 0.02$), and 16-slice windows converge to the round-off floor of F in 3–4 iterations, each of which reduces the largest boundary update by one to two orders of magnitude.

2.5 Boundary corrections and conservation laws

The parareal update in Equation (1) is evaluated in osculating Jacobi orbital elements, with mean longitudes wrapped to $(-\pi, \pi]$. Applying the correction in orbital elements rather than Cartesian coordinates is essential: because coarse errors are predominantly phase errors along the orbit, adding Cartesian position and velocity vectors across a phase difference of 0.1–1 rad cuts across the chord of the ellipse, producing unphysical semi-major axes and causing the iteration to diverge.

Like other parareal schemes for Hamiltonian systems (Dai et al. 2013), the element update alone does not strictly conserve total energy E and angular momentum $\mathbf{L}$ across slice boundaries. We therefore project each corrected boundary state onto target values of E and $\mathbf{L}$ defined by the start-of-slice value plus the increment measured during the most recent fine integration of that slice. This projection is accomplished by applying a tiny uniform rescaling and an infinitesimal rigid rotation to the planets' barycentric velocities (Appendix D). Unlike manifold projections applied to non-symplectic integrators, here the underlying fine solver F is symplectic: the projection only removes the small nonlinear mismatch ($\Delta E/E \sim 10^{-14}$ at the accepted iterate) introduced by adding osculating elements in Equation (1), so each slice carries only the conservation error of a single fine solve and the converged parareal fixed point remains unchanged.

2.6 Windows, pipelining, and stopping criterion

Because the parareal error grows binomially with the number of slices from the last exact boundary, long intervals require more iterations to converge (3 iterations for 10 slices versus 4 for 20 or 40). We therefore integrate the trajectory in consecutive *windows* of N slices, with N equal to the number of cores and each window starting from the converged end state of the previous one. The slice length $T = 360\,000$ d ≈ 986 yr (6×10^4 fine steps) balances the coarse error, which grows with slice length as T^3, against the relative overhead of the coarse step, which rises for shorter slices (Appendix C).

Within each window, one thread is pinned to one core and owns one slice. As soon as thread n receives its updated boundary state U_n^k, it evaluates the coarse step

$G(U_n^k)$, forms U_{n+1}^k via Equation (1), passes it to thread $n+1$, and immediately begins the fine solve $F(U_n^k)$ needed for iteration $k+1$. Coarse sweeps of successive iterations thus propagate through the window like a wave while the fine integrations of consecutive iterations overlap (Minion 2010, Appendix C). Because the coarse sweep lies on the serial critical path, each coarse call is further accelerated by offloading its independent Fourier-grid evaluations to helper threads running on cores whose slices have already converged. All reductions are performed in a fixed order, making the parallel trajectory bitwise reproducible regardless of thread timing.

A window terminates at the first iteration $k \geq K_{\min}$ whose largest planet-position update across all slice boundaries, $u_k = \max_n |\mathbf{x}(U_n^k) - \mathbf{x}(U_n^{k-1})|$, is at most τ, and at the latest at $K_{\max} = 6$. Because u_k measures the error of iterate $k-1$ and each iteration contracts the error by a factor of 10–100, the accepted iterate k is one to two orders of magnitude more accurate than τ. We use $K_{\min} = 3$ and $\tau = 2 \times 10^{-6}$ AU. Because convergence is known only when the last slice completes its update, threads speculatively start their next fine solve and abort it once the window terminates. Once a window converges, all N slice boundaries and any intra-slice outputs from the final fine pass are available in chronological order, and Parachron supports saving and resuming exact snapshots at any point during the run.

2.7 Use of AI tools

AI tools were used to derive part of the solution and to write some of the code (Anthropic Claude Opus 5.5 and Fable 5.1; OpenAI Astra 6.0), and to spell-check the manuscript (Google Gemini Argon 4.0). The author checked all results and takes full responsibility for the content of this paper.

3 Accuracy

3.1 Trajectory fidelity and secular frequencies

Because the Solar System is chaotic ($\lambda_{\mathrm{L}} \approx 5\,\mathrm{Myr}$), even two serial integrations that differ only in floating-point round-off diverge to $\sim$1 AU within $\sim$100 Myr (Figure 3a, orange). In MEP (blue), stopping each 16-slice window at $\tau = 2 \times 10^{-6}$ AU leaves a small residual below τ: the parallel trajectory follows the round-off floor for the first $\sim 2 \times 10^5$ yr and remains within an order of magnitude of the serial round-off twin until both saturate near 100 Myr. Once separated by chaos, the trajectories exhibit the same qualitative orbital behaviour (Figure 3b).

Over 10 Myr intervals, before trajectories decorrelate, we measure the fundamental secular frequencies g_i and s_i via frequency analysis (NAFF; Laskar 1993). Compared against a reference serial integration at $\Delta t = 3\,\mathrm{d}$, the serial integrator at its standard $\Delta t = 6\,\mathrm{d}$ step has a finite-step truncation error $\propto \Delta t^2$ of up to $5.1 \times 10^{-6''}\,\mathrm{yr}^{-1}$ (in Mars' g_4). This truncation error—not floating-point round-off—is the physical yardstick for frequency accuracy. Across seven realizations of the initial conditions, MEP adds a random scatter of only $1.9 \times 10^{-8''}\,\mathrm{yr}^{-1}$ rms to the inner planets' secular frequencies and a systematic shift of at most 2% of the serial $\Delta t = 6\,\mathrm{d}$ truncation error. Using larger windows ($N = 64$ slices) reduces the frequency scatter by a further

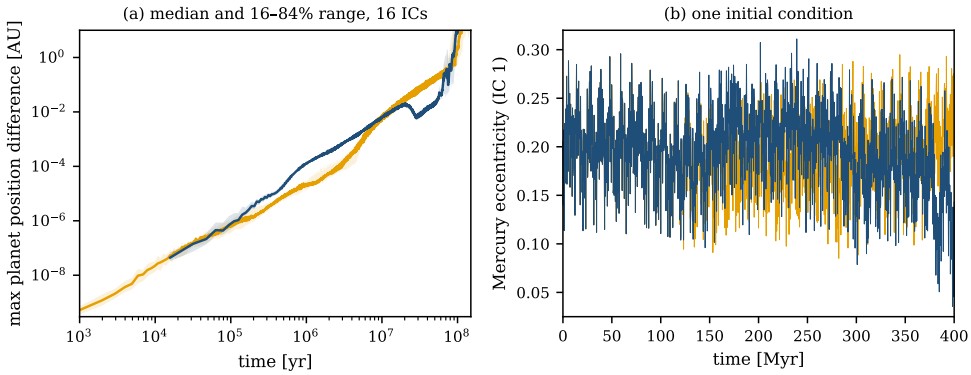


Fig. 3 (a) Largest planet position difference between pairs of trajectories from identical initial conditions (median and 16–84% range of 16 pairs): two serial integrations that differ only in floating-point round-off (orange), and MEP against the serial integration (blue). (b) Mercury's eccentricity for one initial condition, serial (orange) and MEP (blue): the trajectories agree until chaos separates them, with the same character afterwards.

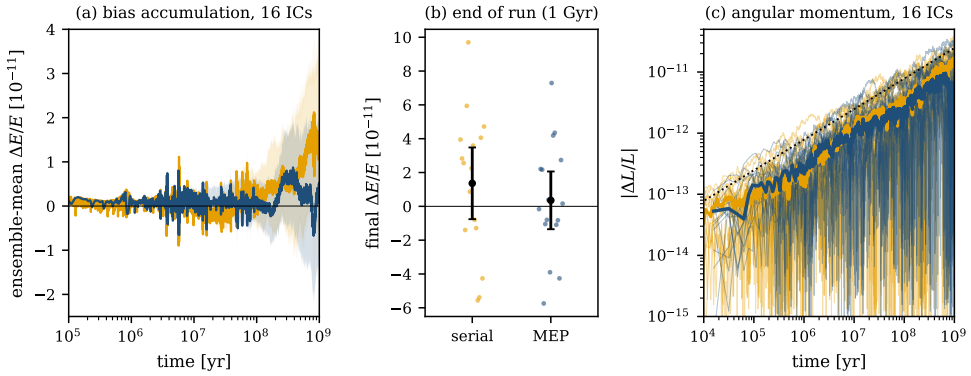


Fig. 4 Energy and angular-momentum conservation of 16-member ensembles over 1 Gyr for the serial integrator (orange) and MEP (blue). (a) Ensemble mean of the relative energy error with ±2 standard errors (shaded). (b) Final relative energy error of each run at 1 Gyr, with ensemble mean and ±2 standard errors. (c) Relative angular-momentum error of the 16 serial and 16 MEP runs (thin lines: individual runs; thick: median); the dotted line shows the Brouwer-law round-off growth ($10^{-16}\sqrt{N_{\mathrm{steps}}}$).

factor of three because most slices undergo additional iterations before the window terminates.

3.2 Energy and angular-momentum conservation

To test conservation over billion-year timescales (cf. Dagli and Rein 2026), we integrate a 16-member ensemble of the Solar System for 1 Gyr both serially and with MEP. After velocity projection at slice boundaries (Section 2.5), boundary energy jumps remain at the floating-point round-off floor ($\sim 5 \times 10^{-14}$ rms). Over the full 1 Gyr

integrations, both the relative energy error and the relative angular-momentum error follow the unbiased $\propto 10^{-16}\sqrt{N_{\rm steps}}$ random walk of Brouwer's law (Brouwer 1937) (Figure 4). At 1 Gyr, the ensemble-mean relative energy error is $(+0.4 \pm 0.9) \times 10^{-11}$ for MEP against $(+1.4 \pm 1.1) \times 10^{-11}$ for the serial ensemble, and the relative angular-momentum error is $(+0.1 \pm 0.4) \times 10^{-11}$ against $(+0.3 \pm 0.5) \times 10^{-11}$ serially. The paired differences between MEP and serial runs from the same initial conditions are consistent with zero within 1σ.

3.3 Long-term ensemble statistics

Because individual trajectories decorrelate after ∼100 Myr, billion-year accuracy must be evaluated statistically across ensembles (Laskar and Gastineau 2009; Abbot et al. 2023; Rein et al. 2025). While directly sampling the rare ∼1% probability of a planetary collision over 5 Gyr requires $\gtrsim 10^3$ integrations, Rein et al. (2025) showed that long-term stability is governed by chaotic diffusion driven by the fundamental secular frequencies g_i and s_i and the resulting eccentricity excursions of Mercury. For the 16-member 1 Gyr ensembles above, two-sample Kolmogorov–Smirnov (KS) tests comparing the distributions of Mercury's maximum eccentricity $e_{\rm max}$ and the eight inner secular frequencies $g_1 \dots g_4$ and $s_1 \dots s_4$ against the serial reference ensemble show no statistically significant differences, with $p \geq 0.21$ across all nine observables (Appendix F, Figure 9). Over 1 Gyr, Mercury's mean maximum eccentricity is 0.316 ± 0.004 for MEP versus 0.324 ± 0.009 for the serial ensemble ($p = 1.00$). The parallel ensemble is statistically indistinguishable from the serial reference.

4 Performance

4.1 Wall-clock speed-up and convergence

On 16 cores, MEP converges in an average of $\bar{K} = 3.63$ iterations per window and achieves a measured wall-clock speed-up of 3.3 (3.29–3.32 across three repetitions of a 100-window, 1.58 Myr benchmark) over a serial integration with the same C kernel, reducing a 5 Gyr integration of the Solar System from 5.8 h to 1.8 h. On 64 cores, the integration completes in 52 min, a speed-up of 6.7 over a single core of the C kernel. Compared to the 9.0 h single-core record of Dagli and Rein (2026), MEP reduces the wall-clock time at identical accuracy by a factor of 5.0 on 16 cores and 10.4 on 64 cores.

The number of iterations required per window is governed by the higher-order perturbations omitted from the first-order coarse model. At termination, the largest boundary update always occurs in the mean longitude of Venus, Earth, or Mars, where Keplerian shear (Equation 3) amplifies the residual semi-major-axis error: after three iterations, a residual error of 2.9×10^{-10} in Venus' $\ln a$ produces the measured phase update of 4.4×10^{-6} rad. The dominant omitted perturbation is the second-order Earth–Mars–Jupiter three-body near-resonance $4\lambda_{\rm E} - 8\lambda_{\rm M} + 3\lambda_{\rm J}$, which drives an oscillation in Mars' mean semi-major axis with a period of ≈ 1836 yr and a relative amplitude of 2.2×10^{-7}—the largest long-period term in Mars' semi-major axis in the planetary theory VSOP87 (period 1783 yr, amplitude 1.9×10^{-7}; Bretagnon and

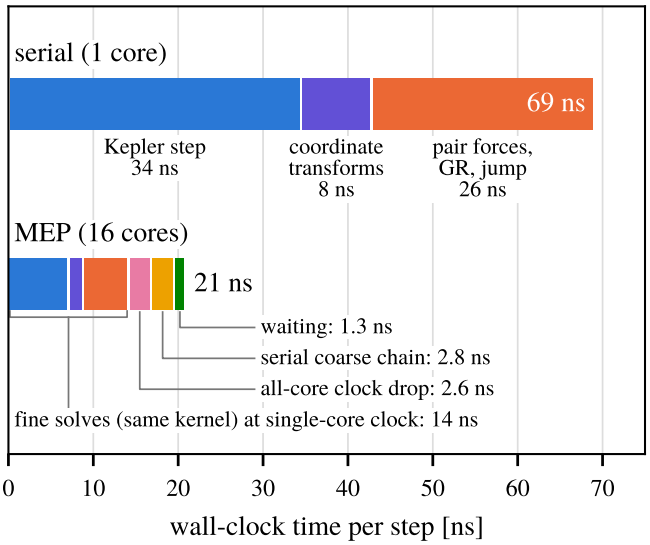


Fig. 5 Wall-clock time per integration step for the Solar System (eight planets with GR, $\Delta t = 6\,\mathrm{d}$), serial and with MEP on 16 cores (speed-up 3.3). Serial: 5.8 h for 5 Gyr with the C kernel, split by the cost fractions of its parts. The Kepler step comprises the series start, one evaluation of the Stiefel functions and the fifth-order correction; the interaction step comprises the transformations between Jacobi and heliocentric coordinates (prefix sums over the SIMD lanes) and the planet-pair forces with the GR terms and the jump step. MEP: one window (20 ms for $16 \times 6 \times 10^4$ steps), with the fine solves split in the same proportions.

Francou 1988). Including second-order mass terms in the coarse propagator reduces the average iteration count to $\bar{K} \approx 2.4$, at the expense of a heavier serial coarse step (Section 5.3).

4.2 Runtime breakdown and parallel overhead

Figure 5 compares the effective wall-clock time per integration step on one core and on 16 cores. In a serial integration (69 ns per step), the Kepler solver accounts for half of the runtime (34 ns), pairwise force evaluations and GR corrections for 38% (26 ns), and Jacobi coordinate transformations for the remaining 12% (8 ns). On 16 cores, MEP completes a 16-slice window (9.6×10^5 fine steps) in 20 ms, reducing the effective wall-clock time per step to 21 ns. Of this 21 ns, the parallel fine integrations account for 16.8 ns (80% of the runtime): 14 ns at the single-core clock rate plus 2.6 ns from the processor's clock-frequency reduction under all-core AVX-512 load (5.3 ms versus 4.4 ms per fine slice). The remaining 4.1 ns of parallel overhead comprises 2.8 ns on the serial coarse chain and 1.3 ns of thread synchronization and load-imbalance waiting across slices. Because the fine solver still dominates the wall-clock budget, any future acceleration of the fine kernel transfers almost directly to MEP.

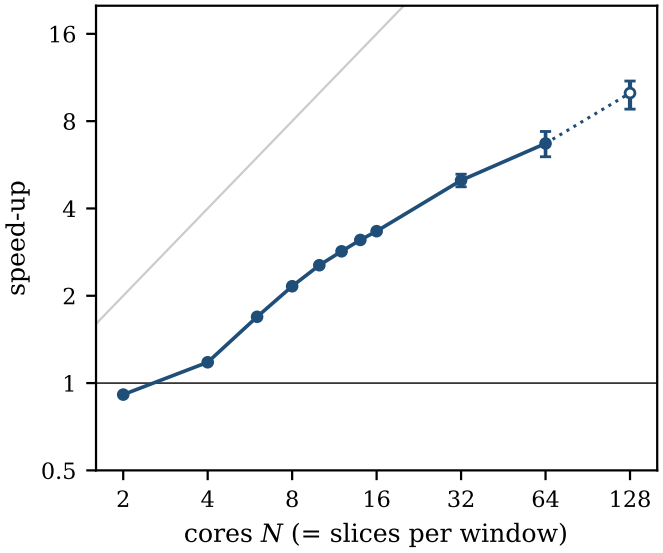


Fig. 6 Measured speed-up of MEP as a function of the number of cores N (= slices per window) up to 64 cores, and the estimated speed-up for 128 cores (open symbol; bars: range of the estimate). The grey line marks ideal linear scaling.

4.3 Parallel scaling

Figure 6 shows the parallel scaling of MEP from $N = 2$ to 64 cores, with one time slice assigned to each core. Two cores cannot provide a speed-up ($S = 0.91$) because two parareal iterations already render both slices exact. Beyond two cores, the measured speed-up rises steadily to 3.3 on 16 cores, 5.0 on 32 cores, and 6.7 on 64 cores (52 min for 5 Gyr, or 10.4× faster than the 9.0 h record of Dagli and Rein 2026), with a gentle curvature caused by the gradual increase in the mean iteration count ($\bar{K}$ rising from 3.0 at $N \leq 6$ to 3.6 at $N = 16$) and by the growing length of the serial coarse chain. Up to 16 cores, the measured performance attains 85%–91% of the theoretical pipelined schedule (Equation 4) evaluated with ideal single-core slice costs; the difference arises primarily from all-core clock throttling, with a smaller contribution from helper-thread contention during coarse steps.

Extrapolating the schedule model to 128 cores—with $\bar{K}$ increasing by 0.6–0.9 iterations per doubling of N—projects a speed-up of 10 (9–11) relative to a single core of the C kernel (open symbol in Figure 6).

5 Discussion

5.1 Regime of applicability

MEP reduces the wall-clock time of an individual trajectory rather than the total CPU cost of an ensemble: on 16 cores, it expends 4.8× more core-hours than a single-core serial integration. For large ensembles whose members can all run concurrently,

assigning one serial integration to each core remains the most economical strategy. Time parallelism becomes advantageous whenever a single long trajectory is the scientific product or the bottleneck: integrations spanning 10–100 Gyr (Hoang et al. 2022; Dagli and Rein 2026), astrochronological orbital solutions, sequential workflows such as rare-event splitting (Abbot et al. 2021) whose stages depend on one another, ensembles with fewer members than available cores, and rapid turnaround for exploratory runs. A 10 Gyr integration that previously required "several weeks" on one core (Rein and Tamayo 2017), or 5 Gyr runs with "a minimum wall-time of one month, even if a large number of cores is available" (Brown and Rein 2020), complete in 3.5 h and 1.8 h on 16 cores (and in 1.7 h and 52 min on 64 cores); a direct 100 Gyr integration takes 17 h on 64 cores. On a cluster, time parallelism and ensemble parallelism combine naturally, with one ensemble member assigned to each multi-core node and each trajectory accelerated across the cores of that node.

Because our speed-ups are measured against the fastest available serial algorithm for this problem, they do not stem from a slow baseline. Strictly speaking, MEP is not a symplectic map: like any parareal scheme, its accepted iterate equals the symplectic fine trajectory plus a residual bounded by τ. However, as demonstrated in Section 3, this residual preserves Brouwer-law conservation of energy and angular momentum over billion-year timescales and shifts the secular frequencies by at most 2% of the serial integrator's own finite-step truncation error.

5.2 Physical mechanism of convergence

In the parallel-in-time literature, parareal is commonly said to fail for oscillatory systems because the coarse propagator accumulates phase errors (Gander and Hairer 2014; Ruprecht 2018). For planetary orbits, this diagnosis is incomplete: state-independent phase offsets cancel out in the parareal update (Section 2.2), and convergence is governed instead by how accurately the coarse propagator resolves the sensitivity of the semi-major axes to the orbital state, which Keplerian shear amplifies by four orders of magnitude per kyr. Consequently, neither a higher-order coarse integrator (which would cost nearly as much as the fine solver) nor an artificially slowed orbital phase (Mikkola and Aarseth 1996; Wang et al. 2020, which preserves secular evolution but amplifies short-period fluctuations) can resolve the bottleneck. Instead, the solution is a canonical transformation to mean elements that strips the short-period oscillations from the semi-major axes, drawing on two centuries of perturbation theory in celestial mechanics (Brouwer 1959; Deprit 1969; Laskar 1990; Murray and Dermott 2000). The shear-amplified threshold of $\lesssim 10^{-8}$ in the semi-major axes then dictates precisely which perturbative terms the coarse model must retain.

5.3 Prospects for further acceleration and generalization

Throughout this work, we have required MEP to match the full accuracy of the $\Delta t = 6$ d serial integrator ($\tau = 2 \times 10^{-6}$ AU, $K_{\min} = 3$). However, Rein et al. (2025) recently demonstrated that Solar System ensembles remain statistically converged for serial time steps up to $\Delta t \approx 32$ d. Relaxing the parareal stopping criterion exploits this margin for an immediate 20%–40% additional speed-up while remaining well within

the regime of statistical convergence. Specifically, loosening τ to 10^{-5}–3×10^{-5} AU at $K_{\min} = 3$ lowers the mean iteration count to $\bar{K} = 3.0$–3.2 and increases the 16-core speed-up to 3.7–3.9 (1.5 h for 5 Gyr), while keeping secular-frequency shifts below 11% of the serial $\Delta t = 6$ d error. Allowing windows to terminate after $K_{\min} = 2$ iterations at $\tau = 3 \times 10^{-4}$ AU (combined with a 180 d giant-planet coarse step and an online linear least-squares calibration of the inner planets' secular rates across windows) lowers $\bar{K}$ to 2.36 and yields a speed-up of 4.6 on 16 cores (1.3 h for 5 Gyr, or 39 min on 64 cores), with frequency errors comparable to a $\Delta t \approx 8$ d serial run. Exploring optimal trade-offs between stopping tolerance and physical fidelity offers a straightforward path to further speed-ups.

Equation (2) bounds the speed-up by $\min(N/K, 1/r)$. Beyond the core-count scaling of Section 4.3, further gains are limited by the growth of $\bar{K}$ and by the serial coarse chain (N hops per window). Including second-order mass terms in the coarse propagator reduces $\bar{K}$ to ≈ 2.4 even at strict tolerances and becomes net-beneficial once a coarse hop takes $\lesssim 0.21$ ms, which may be achievable by fusing passes over the pair tables. For still larger core counts, natural extensions include multilevel PinT schemes in which a fast secular model drives the mean-element coarse level in the spirit of PFASST and MGRIT (Emmett and Minion 2012; Falgout et al. 2014), or low-rank quasi-Newton updates to the coarse propagator along the inner planets' $(\ln a, \lambda)$ directions (Vargas et al. 2022), whose cost falls on the parallel fine solves rather than the serial coarse chain.

Finally, while our hybrid coarse propagator takes advantage of the architecture of the Solar System—four inner planets treated by perturbation theory and four well-separated giants integrated directly—the underlying grid-based perturbation theory is general and requires no power-series expansion in eccentricity or inclination. As a result, the coarse transformation remains accurate even during large secular eccentricity excursions of Mercury, provided orbits do not cross; if a trajectory enters an orbit-crossing close encounter where first-order perturbation theory no longer contracts the window within $K_{\max}$, the window either converges slice-by-slice via the exact fine propagator or can fall back to serial integration once an encounter is flagged. Extending MEP to compact exoplanetary systems near mean-motion resonances, such as TRAPPIST-1 or the resonant chains discovered by TESS and CHEOPS (Tamayo et al. 2017; Leleu et al. 2021), will require integrating the resonant angles dynamically within the coarse propagator, just as we do here for the outer giants.

6 Conclusions

We have demonstrated that a single long-term integration of the Solar System can be parallelized across time at the full accuracy of the serial integrator. The enabling physical insight is that parareal convergence for planetary orbits is governed by how accurately the coarse propagator resolves the semi-major axes, where Keplerian shear amplifies any discrepancy by four orders of magnitude per kyr. By transforming the inner planets to mean elements via a numerical first-order perturbation theory evaluated on the fly while integrating the outer giants directly at a large time step, our

coarse propagator satisfies the resulting $\lesssim 10^{-8}$ accuracy threshold at 2% of the fine solver's cost and converges in 3–4 iterations.

Combined with a portable, latency-optimized C kernel for `WHFast512` that shortens the serial instruction dependency chain (reducing a single-core 5 Gyr integration to 5.8 h), mean-element parareal integrates the eight planets with general relativistic corrections for 5 Gyr in 1.8 h on 16 cores (a speed-up of 3.3 over one core of the C kernel, and 5.0× faster than the 9.0 h record of Dagli and Rein 2026) and in 52 min on 64 cores (6.7× and 10.4× faster, respectively). Across billion-year timescales, the parallel trajectories preserve Brouwer-law conservation of energy and angular momentum, reproduce the secular frequencies to within 2% of the serial integrator's finite-step truncation error, and yield ensemble statistics indistinguishable from serial runs; relaxing the stopping tolerance within the regime of statistical convergence (Rein et al. 2025) provides an additional 20%–40% speed-up. With single-core Wisdom–Holman integrators now near their hardware limits (Dagli and Rein 2026), time parallelism guided by celestial mechanics opens a new dimension for accelerating planetary dynamics.

Acknowledgements. We thank Hanno Rein and the `REBOUND` developers for making `REBOUND` and `WHFast512` freely available. This work made use of `REBOUND` (Rein and Liu 2012), `WHFast512` (Javaheri et al. 2023; Dagli and Rein 2026), Parachron (this work; https://github.com/hippke/parachron), NumPy, SciPy, Matplotlib.

Statements and Declarations

Funding. The author did not receive support from any organization for the submitted work.

Competing interests. The author has no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data and code availability. MEP is available in Parachron, a modified version of `REBOUND`, at https://github.com/hippke/parachron (Python: `parachron.MEP`; C: `reb_mep_*`).

Author contribution. Not applicable (single author).

Appendix A Hand-written assembly and compiled C

Dagli and Rein (2026) rewrote `WHFast512` in x86 assembly to control register allocation and instruction scheduling by hand, arguing that compilers cannot match hand-written assembly for this kernel. Parachron's C kernel began as a direct port of their assembly kernel, permitting an apples-to-apples comparison of the same operations before our algorithmic modifications. This comparison shows no measurable advantage from hand-written assembly, for four reasons. First, for identical mathematical operations, the compiler generates equally fast code: the compiled Kepler step,

which accounts for roughly two thirds of the runtime, required 114 ns per step compared to 122 ns for the hand-written assembly. Second, execution speed is governed by data-dependency chains rather than instruction ordering: the assembly kernel retires roughly one instruction per clock cycle on out-of-order processors capable of dispatching six or more, and the hardware scheduler reorders independent instructions dynamically at runtime. Third, hand-written assembly hardcodes microarchitectural trade-offs that are suboptimal on newer processors: for example, Dagli and Rein (2026) replace hardware square roots and divisions with an approximate reciprocal square root (`vrsqrt14pd`) followed by Newton–Raphson refinement, whereas on newer microarchitectures native square-root and division instructions make the full step ∼10% faster—a compile-time switch in C, but a rewrite in assembly. Fourth, manual bit-level tricks can compromise numerical edge cases for negligible gain: the assembly kernel halves floating-point numbers by subtracting one from their binary exponent, which saves less than 1% of the step time but maps 0.0 to $-\infty$. Finally, whereas x86 assembly is restricted to AVX-512 processors, a single C implementation compiles cleanly across AVX-512, AVX2, and ARM architectures with bitwise identical results.

Because `WHFast512` processes the eight planets in parallel across the lanes of a single SIMD register, its step time is limited by instruction latency along the critical dependency chain—a few clock cycles per multiplication or addition and 10–20 cycles per division or square root—while the processor's remaining arithmetic units sit idle (Hennessy and Patterson 2019, Figure 7). In Parachron's C kernel, we shorten this dependency chain in three ways. First, in the universal-variable Kepler solver (Stiefel and Scheifele 1971), we replace the two Halley and one to two Newton iterations of `WHFast512` with a sixth-order series reversion of Kepler's equation in Δt for the starting value (cf. Danby 1987) followed by a single fifth-order correction generalizing Danby and Burkardt (1983); all required derivatives follow from a single evaluation of the Stiefel functions at the starting value, and the Stiefel functions at the corrected value are obtained via their addition theorems rather than a second series evaluation. The Stiefel power series themselves are evaluated with Estrin's scheme (Estrin 1960; Knuth 1997), which reduces an n-term polynomial from n sequential steps to $\approx \log_2 n$ parallel stages. Second, in the pairwise force evaluation, inverse square roots are computed using hardware square root and division for half of the planet pairs and Newton–Raphson refinement for the other half, balancing the load between the divider and the fused multiply-add units. Third, the transformations between Jacobi and heliocentric coordinates are evaluated as three-stage SIMD prefix-sum scans across vector lanes (Hillis and Steele 1986) instead of 8×8 matrix multiplications. In a 10^6-step Kepler-only test from 4096 random phases (Δt = 3–8 d), the semi-major-axis drift of the C kernel is consistent with zero at the 10^{-21} level per step, whereas the assembly kernel exhibits a drift of up to 4×10^{-20} per step for Mercury.

Figure 7 compares a single step of the assembly kernel of Dagli and Rein (2026) and Parachron's C kernel instruction by instruction; shortening the dependency chain reduces the Kepler step from 65 ns to 36 ns on average and the full step from 86 ns to 69 ns. Wall-clock timings up to 16 cores were measured on an AMD Ryzen 9 9950X (16 cores, Zen 5, full-width 512-bit AVX-512) and 32- and 64-core timings on an AMD EPYC 9575F (64 cores, Zen 5). Additional validation and portability tests were

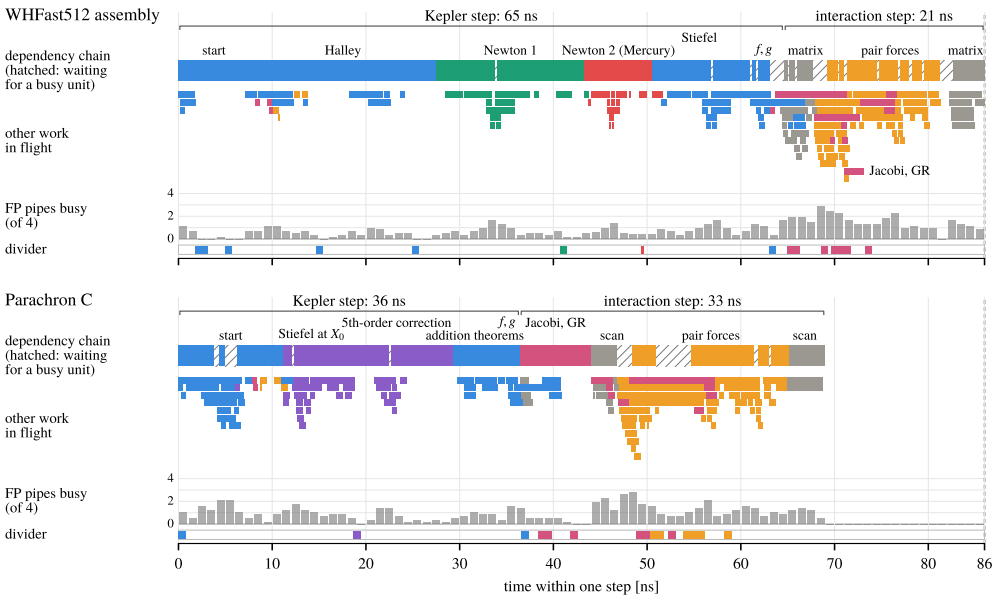


Fig. 7 Instruction-level schedule of one serial step for the Solar System (eight planets with GR, $\Delta t = 6\,\mathrm{d}$) for the assembly kernel of Dagli and Rein (2026) (top, 86 ns average step, with Mercury's second Newton iteration weighted by its 46% occurrence rate) and Parachron's C kernel (bottom, 69 ns). Dependency chain: the critical path of sequentially dependent instructions that determines the step time; each bar represents the latency of one instruction, hatched where the chain stalls waiting for a busy functional unit. Other work in flight: non-critical instructions executing concurrently. FP pipes busy: average number of active floating-point pipes (out of four) per ns. Divider: occupancy of the non-pipelined division/square-root unit. Schedule modelled on an ideal out-of-order core (four 512-bit floating-point pipes, six instructions dispatched per cycle) executing the traced instruction stream of each kernel.

performed on an Intel Xeon Scalable processor (Sapphire Rapids, native AVX-512), an AMD Ryzen 9 7950X (Zen 4, which executes AVX-512 instructions via double-pumped 256-bit execution units), an AMD Ryzen 9 5950X (Zen 3, AVX2 only), hybrid Intel processors combining Performance (P) and Efficient (E) cores without AVX-512 (Intel Core i5-13500 and Core Ultra 7 255), and an Apple M1 processor (ARM64 NEON).

Appendix B Synchronization energy offset in `WHFast512`

Because parareal restarts the fine integrator at every slice boundary, MEP exposed a small systematic energy bias in upstream `WHFast512` that also affects serial integrations with frequent output. Every synchronization call changes the relative energy of the Solar System by a constant offset $\Delta E/E = (+7.54 \pm 0.49) \times 10^{-17}$ (15σ, reproduced with unmodified `REBOUND`), independent of the orbital state. This offset arises from the pair of precomputed 8×8 matrices used to convert between inertial and Jacobi coordinates, which are not exact floating-point inverses: their product is diagonal to 10^{-19}, but the diagonal entries differ from unity by

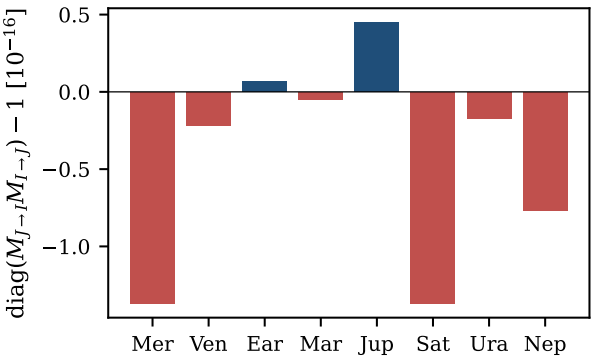


Fig. 8 Deviation from unity of the diagonal entries of the product of `WHFast512`'s precomputed Jacobi-to-inertial and inertial-to-Jacobi transformation matrices for the Solar System masses. Each synchronization rescales the Jacobi coordinates of each planet by these factors, adding $+7.5 \times 10^{-17}$ to the relative energy per call.

$(-1.37, -0.22, +0.07, -0.05, +0.45, -1.37, -0.17, -0.77) \times 10^{-16}$ for Mercury through Neptune (Figure 8). Each round-trip coordinate transformation rescales planet i's Jacobi position and velocity by $(1 + \epsilon_i)$ and shifts its energy by $\approx 2\epsilon_i|U_i|$; summing over the planets' potential energies predicts $+7.6 \times 10^{-17}$ per call, matching the measured offset. In a 5 Gyr integration with output every kyr, this bias accumulates to $\Delta E/E \approx +4 \times 10^{-10}$, exceeding the Brouwer-law round-off random walk of $\sim 10^{-16} N_{\mathrm{steps}}^{1/2} \approx 5 \times 10^{-11}$ (Brouwer 1937; Dagli and Rein 2026). Replacing the matrix multiplications at synchronization boundaries with the recursive Jacobi transformations of `REBOUND` eliminates the offset ($\leq 1.5 \times 10^{-17}$ per call at 2σ) at no measurable runtime cost.

Appendix C Pipelined schedule and parameter choices

With $t(k, n)$ denoting the wall-clock time at which the boundary state U_{n+1}^k becomes available, the wave-pipelined schedule (Section 2.6) obeys

$$t(k, n+1) = \max[t(k, n),\ t(k-1, n) + T_F(n)] + T_{\mathrm{hop}}. \tag{4}$$

In a window of N slices, the initial coarse sweep takes NT_G. In iteration $k \geq 1$, the coarse hop that produces U_{n+1}^k cannot begin until the fine integration $F(U_n^{k-1})$—which started when U_n^{k-1} became available—has finished. When all fine solves have equal duration T_F, the wave of iteration k trails that of iteration $k-1$ by $T_F + T_G$, so the final boundary of the window is ready after $NT_G + K(T_F + T_G)$, whereas the serial runtime for the window is NT_F; dividing the two yields Equation (2). For $K \ll N$ and $r = T_G/T_F \ll 1$, the speed-up approaches N/K; as $K \to N$ (no convergence), the speed-up falls below unity, as observed for a coarse WH integrator. When fine slice durations vary slightly, T_F is governed by the slowest slice in each iteration.

Table 1 Parameters of MEP: default, range explored, and outcome.

Parameter	Default	Explored	Outcome
slices per window N	16 (= cores)	12–24; 32 on SMT threads	fewer slices with more helpers: 2.7–2.8; 20–24 slices: 2.8–2.9; 32 SMT threads: 2.2–2.3
slice length T	986 yr (6×10^4 steps)	246–9856 yr	flat optimum: 789 and 1183 yr 3–8% slower, 1380 yr 10%; 246–493 yr raise r, $\geq$10 kyr raise K
tolerance τ	2×10^{-6} AU	10^{-6}–3×10^{-5} AU	$\bar{K}$ decreases from 3.63 at 2×10^{-6} AU to its floor of 3.0 at 3×10^{-5} AU (Section 5.3)
$K_{\min}$, $K_{\max}$	3, 6	$K_{\min} = 2$, 3	$K_{\min} = 2$ lowers $\bar{K}$ to 2.4 when combined with a 180 d giants' step and online secular calibration (Section 5.3)
stopping rule	largest update $\leq \tau$	error predicted from the contraction	no better accepted states at equal $\bar{K}$ (625 windows)
Fourier grids (inner–inner, inner–giant)	32^2, 16^2; Venus–Earth 16×64 synodic	8^2–64^2; synodic 8×64–32×64	smaller grids raise K; larger square grids lower K but cost more on the coarse chain; the synodic Venus–Earth grid reaches the accuracy of 64^2 for that pair at the cost of 32^2
giants' coarse step	360 d	180–1000 d	240–450 d within the scatter; $\geq$540 d no convergence
coarse helper threads	8, on SMT siblings of the first cores	0–8; placements	pinned placement +5%

Table 1 lists the algorithmic parameters of MEP, the range explored for each, and the resulting performance. Each parameter was tuned using both schedule-model predictions from measured iteration counts and full benchmark integrations. Most parameters lie on a broad plateau where neighbouring values alter the wall-clock speed-up by only a few per cent. The default stopping tolerance ($\tau = 2 \times 10^{-6}$ AU) is chosen for physical accuracy rather than maximum speed: it is the loosest threshold whose added secular-frequency shift is negligible compared to the serial integrator's own finite-step truncation error (Section 3.1).

Appendix D Linearized error recursion and conservation projection

Linearizing the parareal update (Equation 1) around the fine trajectory yields the error recursion

$$e_{n+1}^{k+1} = G'_n \, e_n^{k+1} + (F'_n - G'_n) \, e_n^k, \tag{5}$$

where F'_n and G'_n are the 48×48 Jacobians of the fine and coarse slice maps in Jacobi orbital elements. Convergence is therefore governed by the Jacobian mismatch $F' - G'$ rather than by state-independent errors in G. Evaluated with measured slice Jacobians, Equation (5) reproduces the nonlinear parareal iteration step by step. In osculating elements, short-period fluctuations produce off-diagonal fine-Jacobian entries $\partial \ln a_{\text{end}} / \partial \lambda_j \approx 10^{-6}$–$10^{-5}$ and, combined with Keplerian shear,

$\partial \ln a_{\mathrm{end}}/\partial \ln a_j$ up to 0.09 (Earth with respect to Venus); in mean elements, the $\ln a$ rows of the slice Jacobian are close to the identity.

To project each corrected boundary state onto the target energy $E_{\mathrm{target}} = E(U_n^k) + [E(F(U_n^{k-1})) - E(U_n^{k-1})]$ and target angular momentum $\mathbf{L}_{\mathrm{target}} = \mathbf{L}(U_n^k) + [\mathbf{L}(F(U_n^{k-1})) - \mathbf{L}(U_n^{k-1})]$ (Section 2.5), we adjust the barycentric velocities by $\delta\mathbf{v}_i = \alpha\,\mathbf{v}_i + \boldsymbol{\beta} \times \mathbf{r}_i$. The scalar α and three-vector $\boldsymbol{\beta}$ are obtained from two Newton iterations of the 4×4 linear system

$$\begin{pmatrix} 2E_{\mathrm{kin}} & \mathbf{L}^{\mathrm{T}} \\ \mathbf{L} & \mathsf{I} \end{pmatrix} \begin{pmatrix} \alpha \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} E_{\mathrm{target}} - E \\ \mathbf{L}_{\mathrm{target}} - \mathbf{L} \end{pmatrix}, \qquad \mathsf{I} = \sum_i m_i \left(r_i^2\, \mathbf{1} - \mathbf{r}_i \mathbf{r}_i^{\mathrm{T}} \right). \tag{6}$$

Two numerical details matter at the 10^{-16} level. First, the velocity correction must be applied incrementally as $\mathbf{v}_i \leftarrow \mathbf{v}_i + \delta\mathbf{v}_i$: the mathematically equivalent multiplicative rescaling by $\sqrt{1+x}$ with $x \sim 10^{-12}$ rounds asymmetrically near unity and removes 1.0×10^{-16} of the energy per projection, causing a measurable secular drift. Second, angular momentum must be projected alongside energy; with energy projection alone, $|\mathbf{L}|$ drifts by $-3.1 \times 10^{-10}\,\mathrm{Gyr}^{-1}$.

Appendix E Mean-element transformation and Gauss' equations

We represent each planet by its Jacobi equinoctial elements $E = (\ln a, \lambda, k, h, q, p)$, with $k + \mathrm{i}h = e\,\mathrm{e}^{\mathrm{i}\varpi}$, $q + \mathrm{i}p = \sin(i/2)\,\mathrm{e}^{\mathrm{i}\Omega}$, and mean longitude λ. For every pair of planets $i < j$, we evaluate the first-order perturbing accelerations in Jacobi coordinates on a regular grid of the two mean longitudes (λ_i, λ_j), holding the slow elements $(\ln a, k, h, q, p)$ fixed at their current values. For a planet with Jacobi position $\mathbf{r}$, velocity $\mathbf{v}$, and Keplerian parameter μ (the sum of the masses interior to and including the planet), the energy equation gives $\mathrm{d}\ln a/\mathrm{d}t = 2a\,\mathbf{v} \cdot \mathbf{a}_{\mathrm{pert}}/\mu$ exactly. The rates of the remaining elements follow from Gauss' equations, $\mathrm{d}E/\mathrm{d}t = (\partial E/\partial \mathbf{v}) \cdot \mathbf{a}_{\mathrm{pert}}$, where the partial derivatives at fixed position are evaluated using forward-mode dual numbers through the Cartesian-to-element conversion (finite differences are accurate to only $\sim 10^{-6}$ in some components). The interior planet of a pair experiences the direct term and the dipole indirect term, while the exterior planet experiences the interior mass distribution in the heliocentric-like Jacobi form; compared with a direct N-body sum, these accelerations agree to 3–5 $\times\, 10^{-5}$ (relative) for the inner planets, the residual being second order in the masses.

A two-dimensional FFT of each element rate yields Fourier harmonics $f_{\mathbf{k}}$ with wave vector $\mathbf{k} = (k_1, k_2)$ and frequency $\nu_{\mathbf{k}} = k_1 n_i + k_2 n_j$, where n_i and n_j are the mean motions corresponding to the mean semi-major axes (evaluated iteratively from the same Fourier tables). For the close Venus–Earth pair, the grid is instead spanned by λ_{V} and the synodic angle $\lambda_{\mathrm{V}} - \lambda_{\mathrm{E}}$ with 16×64 points; its harmonic (m, l) has wave vector $\mathbf{k} = (m + l, -l)$ and frequency $m n_{\mathrm{V}} + l(n_{\mathrm{V}} - n_{\mathrm{E}})$. Because the amplitude of a harmonic scales as $e^{|k_1 + k_2|}$ (Murray and Dermott 2000) but decays slowly with synodic order for a close pair, this skewed grid resolves harmonics $|m| \le 8$, $|l| \le 32$ at the same cost as a 32×32 square grid. The $\mathbf{k} = 0$ mode gives the first-order

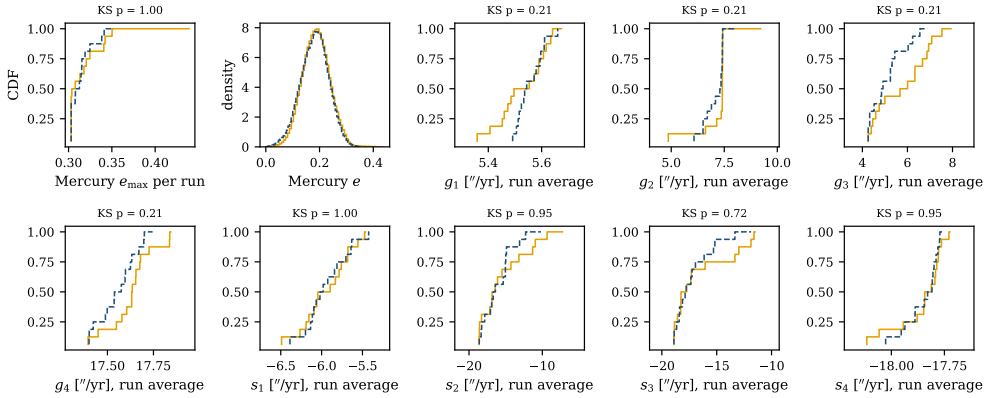


Fig. 9 Ensemble statistics over 1 Gyr for the 16-member serial ensemble (orange) and MEP ensemble (blue). Top left: cumulative distribution of Mercury's maximum eccentricity per run; next: distribution of Mercury's eccentricity; remaining panels: cumulative distributions of the run averages of the inner secular frequencies $g_1 \dots g_4$ and $s_1 \dots s_4$, with two-sample KS p-values shown above each panel.

secular rate, while the short-period displacement of each element at the current mean longitudes $\boldsymbol{\theta} = (\lambda_i, \lambda_j)$ is

$$\Delta E = \sum_{\mathbf{k}\neq 0} \frac{f_{\mathbf{k}}\, \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\boldsymbol{\theta}}}{\mathrm{i}\nu_{\mathbf{k}}}, \qquad \Delta\lambda_{\mathrm{shear}} = -\frac{3}{2}\, n \sum_{\mathbf{k}\neq 0} \frac{f^{(\ln a)}_{\mathbf{k}}\, \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\boldsymbol{\theta}}}{(\mathrm{i}\nu_{\mathbf{k}})^2}, \tag{7}$$

where $\Delta\lambda_{\mathrm{shear}}$ accounts for the perturbation of the Keplerian mean motion $n(a)$ by the short-period oscillation in $\ln a$. The GR correction enters as a one-planet table. Mean elements are obtained by subtracting the summed displacements from the osculating elements. In practice, the displacements of all elements are evaluated with a single FFT per planet pair, because $\sum_{\mathbf{k}} f_{\mathbf{k}} \mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\boldsymbol{\theta}}/\nu_{\mathbf{k}}$ reduces to a grid sum of the element rates weighted by the inverse-transformed kernel $\mathrm{e}^{\mathrm{i}\mathbf{k}\cdot\boldsymbol{\theta}}/\nu_{\mathbf{k}}$.

Appendix F Ensemble statistical tests

Figure 9 shows the cumulative distributions of Mercury's maximum eccentricity $e_{\max}$ (after 50 Myr), Mercury's eccentricity distribution, and the run-averaged inner secular frequencies $g_1 \dots g_4$ and $s_1 \dots s_4$ (measured in 20 Myr windows) for the 16-member 1 Gyr serial and MEP ensembles of Section 3.3. The two-sample KS p-values comparing the MEP ensemble against the serial ensemble are 1.00 for $e_{\max}$ and span 0.21–1.00 across all eight secular frequencies ($g_1 \dots g_4, s_1 \dots s_4$). None of the nine observables shows a statistically significant difference ($p < 0.05$).

References

Abbot, D.S., Hernandez, D.M., Hadden, S., Webber, R.J., Afentakis, G.P., Weare, J.: Simple physics and integrators accurately reproduce mercury instability statistics.

The Astrophysical Journal **944**(2), 190 (2023) https://doi.org/10.3847/1538-4357/acb6ff 2212.14844

Ariel, G., Kim, S.J., Tsai, R.: Parareal methods for highly oscillatory dynamical systems. arXiv e-prints, 1503–02094 (2015) arXiv:1503.02094. arXiv:1503.02094

Abbot, D.S., Webber, R.J., Hadden, S., Seligman, D., Weare, J.: Rare event sampling improves mercury instability statistics. The Astrophysical Journal **923**(2), 236 (2021) https://doi.org/10.3847/1538-4357/ac2fa8 2106.09091

Bretagnon, P., Francou, G.: Planetary theories in rectangular and spherical variables. VSOP 87 solutions. A&A **202**, 309–315 (1988)

Batygin, K., Laughlin, G.: On the dynamical stability of the solar system. The Astrophysical Journal **683**(2), 1207–1216 (2008) https://doi.org/10.1086/589232 0804.1946

Brown, G., Rein, H.: A repository of vanilla long-term integrations of the solar system. Research Notes of the AAS **4**(12), 221 (2020) https://doi.org/10.3847/2515-5172/abd103 2012.05177

Brown, G., Rein, H.: On the long-term stability of the solar system in the presence of weak perturbations from stellar flybys. Monthly Notices of the Royal Astronomical Society **515**(4), 5942–5950 (2022) https://doi.org/10.1093/mnras/stac1763 2206.14240

Brown, G., Rein, H.: General relativistic precession and the long-term stability of the solar system. Monthly Notices of the Royal Astronomical Society **521**(3), 4349–4355 (2023) https://doi.org/10.1093/mnras/stad719 2303.05567

Brouwer, D.: On the accumulation of errors in numerical integration. The Astronomical Journal **46**, 149 (1937) https://doi.org/10.1086/105423

Brouwer, D.: Solution of the problem of artificial satellite theory without drag. The Astronomical Journal **64**, 378 (1959) https://doi.org/10.1086/107958

Bal, G., Wu, Q.: Symplectic parareal. In: Domain Decomposition Methods in Science and Engineering XVII. Lecture Notes in Computational Science and Engineering, vol. 60, pp. 401–408. Springer, Berlin (2008)

Danby, J.M.A.: The solution of Kepler's equation, III. Celestial Mechanics **40**(3), 303–312 (1987) https://doi.org/10.1007/BF01235847

Danby, J.M.A., Burkardt, T.M.: The solution of Kepler's equation, I. Celestial Mechanics **31**(2), 95–107 (1983) https://doi.org/10.1007/BF01686811

Deprit, A.: Canonical transformations depending on a small parameter. Celestial Mechanics **1**(1), 12–30 (1969) https://doi.org/10.1007/bf01230629

Dai, X., Le Bris, C., Legoll, F., Maday, Y.: Symmetric parareal algorithms for hamiltonian systems. ESAIM: Mathematical Modelling and Numerical Analysis **47**(3), 717–742 (2013) https://doi.org/10.1051/m2an/2012046 1011.6222

Dagli, R., Rein, H.: A very fast symplectic integrator for planetary systems: Whfast512 in x86 assembly. arXiv e-prints, 2609–25339 (2026) arXiv:2609.25339. arXiv:2609.25339

Emmett, M., Minion, M.: Toward an efficient parallel in time method for partial differential equations. Communications in Applied Mathematics and Computational Science **7**(1), 105–132 (2012) https://doi.org/10.2140/camcos.2012.7.105

Estrin, G.: Organization of computer systems: the fixed plus variable structure computer. In: Papers Presented at the May 3–5, 1960, Western Joint IRE-AIEE-ACM Computer Conference, pp. 33–40. ACM, New York (1960). https://doi.org/10.1145/1460361.1460365

Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM Journal on Scientific Computing **36**(6), 635–661 (2014) https://doi.org/10.1137/130944230

Gander, M.J.: 50 Years of Time Parallel Time Integration, pp. 69–113. Springer, ??? (2015). https://doi.org/10.1007/978-3-319-23321-5_3

Gander, M.J., Hairer, E.: Nonlinear Convergence Analysis for the Parareal Algorithm, pp. 45–56. Springer, ??? (2008). https://doi.org/10.1007/978-3-540-75199-1_4

Gander, M.J., Hairer, E.: Analysis for parareal algorithms applied to hamiltonian differential equations. Journal of Computational and Applied Mathematics **259**, 2–13 (2014) https://doi.org/10.1016/j.cam.2013.01.011

Grimm, S.L., Stadel, J.G.: The genga code: Gravitational encounters in N-body simulations with gpu acceleration. The Astrophysical Journal **796**(1), 23 (2014) https://doi.org/10.1088/0004-637x/796/1/23 1404.2324

Grimm, S.L., Stadel, J.G.: Terrestrial planet formation in the era of GPU computing. arXiv e-prints, 2604–09184 (2026) arXiv:2604.09184. arXiv:2604.09184; chapter in Benz W. et al., eds, The NCCR PlanetS, Springer, in press

Grimm, S.L., Stadel, J.G., Brasser, R., Meier, M.M.M., Mordasini, C.: Genga. ii. gpu planetary n-body simulations with non-newtonian forces and high number of particles. The Astrophysical Journal **932**(2), 124 (2022) https://doi.org/10.3847/1538-4357/ac6dd2 2201.10058

Hoang, N.H., Mogavero, F., Laskar, J.: Long-term instability of the inner solar system: numerical experiments. Monthly Notices of the Royal Astronomical Society **514**(1), 1342–1350 (2022) https://doi.org/10.1093/mnras/stac1299 2205.04170

Hennessy, J.L., Patterson, D.A.: Computer Architecture: A Quantitative Approach, 6th edn. Morgan Kaufmann, Cambridge, MA (2019)

Hillis, W.D., Steele, G.L.: Data parallel algorithms. Communications of the ACM **29**(12), 1170–1183 (1986) https://doi.org/10.1145/7902.7903

Haut, T., Wingate, B.: An asymptotic parallel-in-time method for highly oscillatory pdes. arXiv e-prints, 1303–6615 (2013) arXiv:1303.6615. arXiv:1303.6615

Hernandez, D.M., Zeebe, R.E., Hadden, S.: Stepsize errors in the N-body problem: discerning mercury's true possible long-term orbits. Monthly Notices of the Royal Astronomical Society **510**(3), 4302–4307 (2021) https://doi.org/10.1093/mnras/stab3664 2111.08835

Jiménez-Pérez, H., Laskar, J.: A time-parallel algorithm for almost integrable hamiltonian systems. arXiv e-prints, 1106–3694 (2011) arXiv:1106.3694. arXiv:1106.3694

Javaheri, P., Rein, H., Tamayo, D.: Whfast512: A symplectic n-body integrator for planetary systems optimized with avx512 instructions. The Open Journal of Astrophysics **6** (2023) https://doi.org/10.21105/astro.2307.05683 2307.05683

Knuth, D.E.: The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd edn. Addison-Wesley, Reading, MA (1997)

Kinoshita, H., Yoshida, H., Nakai, H.: Symplectic integrators and their application to dynamical astronomy. CELESTIAL MECHANICS AND DYNAMICAL ASTRONOMY **50**(1), 59–71 (1991) https://doi.org/10.1007/bf00048986

Leleu, A., Alibert, Y., Hara, N.C., Hooton, M.J., Wilson, T.G., Robutel, P., Delisle, J.-B., Laskar, J., Hoyer, S., Lovis, C., Bryant, E.M., Ducrot, E., Cabrera, J., Delrez, L., Acton, J.S., Adibekyan, V., Allart, R., Allende Prieto, C., Alonso, R., Alves, D., Anderson, D.R., Angerhausen, D., Anglada Escudé, G., Asquier, J., Barrado, D., Barros, S.C.C., Baumjohann, W., Bayliss, D., Beck, M., Beck, T., Bekkelien, A., Benz, W., Billot, N., Bonfanti, A., Bonfils, X., Bouchy, F., Bourrier, V., Boué, G., Brandeker, A., Broeg, C., Buder, M., Burdanov, A., Burleigh, M.R., Bárczy, T., Cameron, A.C., Chamberlain, S., Charnoz, S., Cooke, B.F., Corral Van Damme, C., Correia, A.C.M., Cristiani, S., Damasso, M., Davies, M.B., Deleuil, M., Demangeon, O.D.S., Demory, B.-O., Di Marcantonio, P., Di Persio, G., Dumusque, X., Ehrenreich, D., Erikson, A., Figueira, P., Fortier, A., Fossati, L., Fridlund, M., Futyan, D., Gandolfi, D., García Muñoz, A., Garcia, L.J., Gill, S., Gillen, E., Gillon, M., Goad, M.R., González Hernández, J.I., Guedel, M., Günther, M.N., Haldemann, J., Henderson, B., Heng, K., Hogan, A.E., Isaak, K., Jehin, E., Jenkins, J.S., Jordán, A., Kiss, L., Kristiansen, M.H., Lam, K., Lavie, B., Etangs, A., Lendl, M., Lillo-Box, J., Lo Curto, G., Magrin, D., Martins, C.J.A.P., Maxted, P.F.L., McCormac, J., Mehner, A., Micela, G., Molaro, P., Moyano, M., Murray, C.A., Nascimbeni, V., Nunes, N.J., Olofsson, G., Osborn, H.P., Oshagh, M., Ottensamer, R., Pagano, I., Pallé, E., Pedersen, P.P., Pepe, F.A., Persson, C.M., Peter, G., Piotto, G., Polenta,

G., Pollacco, D., Poretti, E., Pozuelos, F.J., Queloz, D., Ragazzoni, R., Rando, N., Ratti, F., Rauer, H., Raynard, L., Rebolo, R., Reimers, C., Ribas, I., Santos, N.C., Scandariato, G., Schneider, J., Sebastian, D., Sestovic, M., Simon, A.E., Smith, A.M.S., Sousa, S.G., Sozzetti, A., Steller, M., Suárez Mascareño, A., Szabó, G.M., Ségransan, D., Thomas, N., Thompson, S., Tilbrook, R.H., Triaud, A., Turner, O., Udry, S., Van Grootel, V., Venus, H., Verrecchia, F., Vines, J.I., Walton, N.A., West, R.G., Wheatley, P.J., Wolter, D., Zapatero Osorio, M.R.: Six transiting planets and a chain of laplace resonances in toi-178. Astronomy & Astrophysics **649**, 26 (2021) https://doi.org/10.1051/0004-6361/202039767 2101.09260

Laskar, J.: A numerical experiment on the chaotic behaviour of the solar system. Nature **338**(6212), 237–238 (1989) https://doi.org/10.1038/338237a0

Laskar, J.: The chaotic motion of the solar system: A numerical estimate of the size of the chaotic zones. Icarus **88**(2), 266–291 (1990) https://doi.org/10.1016/0019-1035(90)90084-m

Laskar, J.: Frequency analysis for multi-dimensional systems. global dynamics and diffusion. Physica D: Nonlinear Phenomena **67**(1-3), 257–281 (1993) https://doi.org/10.1016/0167-2789(93)90210-r

Laskar, J.: Chaotic diffusion in the solar system. Icarus **196**(1), 1–15 (2008) https://doi.org/10.1016/j.icarus.2008.02.017 0802.3371

Laskar, J., Gastineau, M.: Existence of collisional trajectories of mercury, mars and venus with the earth. Nature **459**(7248), 817–819 (2009) https://doi.org/10.1038/nature08096

Lau, T.C.H., Lee, M.H.: Parallelization of the symplectic massive body algorithm (symba) n-body code. Research Notes of the AAS **7**(4), 74 (2023) https://doi.org/10.3847/2515-5172/accc8a 2304.07325

Legoll, F., Lelievre, T., Samaey, G.: A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations. arXiv e-prints, 1204–5926 (2012) arXiv:1204.5926. arXiv:1204.5926

Lions, J.-L., Maday, Y., Turinici, G.: Résolution d'edp par un schéma en temps «pararéel ». Comptes Rendus de l'Académie des Sciences - Series I - Mathematics **332**(7), 661–668 (2001) https://doi.org/10.1016/s0764-4442(00)01793-6

Laskar, J., Robutel, P., Joutel, F., Gastineau, M., Correia, A.C.M., Levrard, B.: A long-term numerical solution for the insolation quantities of the earth. Astronomy & Astrophysics **428**(1), 261–285 (2004) https://doi.org/10.1051/0004-6361:20041335

Mikkola, S., Aarseth, S.J.: A slow-down treatment for close binaries. Celestial Mechanics and Dynamical Astronomy **64**(3), 197–208 (1996) https://doi.org/10.1007/bf00728347

Murray, C.D., Dermott, S.F.: Solar System Dynamics. Cambridge University Press, ??? (2000). https://doi.org/10.1017/cbo9781139174817

Mogavero, F., Hoang, N.H., Laskar, J.: Timescales of chaos in the inner solar system: Lyapunov spectrum and quasi-integrals of motion. Physical Review X **13**(2) (2023) https://doi.org/10.1103/physrevx.13.021018 2305.01683

Minion, M.: A hybrid parareal spectral deferred corrections method. Communications in Applied Mathematics and Computational Science **5**(2), 265–301 (2010) https://doi.org/10.2140/camcos.2010.5.265

Mogavero, F., Laskar, J.: Long-term dynamics of the inner planets in the solar system. Astronomy & Astrophysics **655**, 1 (2021) https://doi.org/10.1051/0004-6361/202141007 2105.14976

Nobili, A., Roxburgh, I.W.: Simulation of general relativistic corrections in long term numerical integrations of planetary orbits. In: Kovalevsky, J., Brumberg, V.A. (eds.) Relativity in Celestial Mechanics and Astrometry. High Precision Dynamical Theories and Observational Verifications. IAU Symposium, vol. 114, pp. 105–110 (1986)

Ong, B.W., Schroder, J.B.: Applications of time parallelization. Computing and Visualization in Science **23**(1-4) (2020) https://doi.org/10.1007/s00791-020-00331-4

Obertas, A., Van Laerhoven, C., Tamayo, D.: The stability of tightly-packed, evenly-spaced systems of earth-mass planets orbiting a sun-like star. Icarus **293**, 52–58 (2017) https://doi.org/10.1016/j.icarus.2017.04.010 1703.08426

Peddle, A., Haut, T., Wingate, B.: Parareal convergence for oscillatory pdes with finite time-scale separation. arXiv e-prints, 1705–09565 (2017) arXiv:1705.09565. arXiv:1705.09565

Portegies Zwart, S., Boekholt, T.: On the minimal accuracy required for simulating self-gravitating systems by means of direct N-body methods. The Astrophysical Journal **785**(1), 3 (2014) https://doi.org/10.1088/2041-8205/785/1/l3

Rein, H., Brown, G., Kanda, M.: On the statistical convergence of n-body simulations of the solar system. The Open Journal of Astrophysics **8** (2025) https://doi.org/10.33232/001c.154745 2507.04987

Rein, H., Dey, K., Tamayo, D.: Democratic heliocentric coordinates underestimate the rate of instabilities in long-term integrations of the solar system. arXiv e-prints, 2601–08019 (2026) arXiv:2601.08019. arXiv:2601.08019

Rein, H., Liu, S.-F.: Rebound: an open-source multi-purpose N-body code for collisional dynamics. Astronomy & Astrophysics **537**, 128 (2012) https://doi.org/10.1051/0004-6361/201118085 1110.4876

Rein, H., Tremaine, S.: Symplectic integrators in the shearing sheet: Symplectic integrators. Monthly Notices of the Royal Astronomical Society **415**(4), 3168–3176 (2011) https://doi.org/10.1111/j.1365-2966.2011.18939.x 1103.1376

Rein, H., Tamayo, D.: 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 (2015) https://doi.org/10.1093/mnras/stv1257 1506.01084

Rein, H., Tamayo, D.: A new paradigm for reproducing and analyzing n-body simulations of planetary systems. Monthly Notices of the Royal Astronomical Society, 232 (2017) https://doi.org/10.1093/mnras/stx232 1701.07423

Ruprecht, D.: Wave propagation characteristics of parareal. Computing and Visualization in Science **19**(1-2), 1–17 (2018) https://doi.org/10.1007/s00791-018-0296-z 1701.01359

Speck, R.: Algorithm 997: pySDC—prototyping spectral deferred corrections. ACM Transactions on Mathematical Software **45**(3), 1–23 (2019) https://doi.org/10.1145/3310410 1808.02731

Stiefel, E.L., Scheifele, G.: Linear and Regular Celestial Mechanics: Perturbed Two-body Motion, Numerical Methods, Canonical Theory. Grundlehren der mathematischen Wissenschaften, vol. 174. Springer, Berlin (1971)

Saha, P., Stadel, J., Tremaine, S.: A parallel integration method for solar system dynamics. The Astronomical Journal **114**, 409 (1997) https://doi.org/10.1086/118485 astro-ph/9605016

Sussman, G.J., Wisdom, J.: Numerical evidence that the motion of pluto is chaotic. Science **241**(4864), 433–437 (1988) https://doi.org/10.1126/science.241.4864.433

Sussman, G.J., Wisdom, J.: Chaotic evolution of the solar system. Science **257**(5066), 56–62 (1992) https://doi.org/10.1126/science.257.5066.56

Tamayo, D., Cranmer, M., Hadden, S., Rein, H., Battaglia, P., Obertas, A., Armitage, P.J., Ho, S., Spergel, D.N., Gilbertson, C., Hussain, N., Silburt, A., Jontof-Hutter, D., Menou, K.: Predicting the long-term stability of compact multiplanet systems. Proceedings of the National Academy of Sciences **117**(31), 18194–18205 (2020) https://doi.org/10.1073/pnas.2001258117 2007.06521

Tamayo, D., Rein, H., Petrovich, C., Murray, N.: Convergent migration renders trappist-1 long-lived. The Astrophysical Journal Letters **840**(2), 19 (2017) https://doi.org/10.3847/2041-8213/aa70ea 1704.02957

Tamayo, D., Silburt, A., Valencia, D., Menou, K., Ali-Dib, M., Petrovich, C., Huang, C.X., Rein, H., Laerhoven, C.v., Paradise, A., Obertas, A., Murray, N.: A machine

learns to predict the stability of tightly packed planetary systems. The Astrophysical Journal Letters **832**(2), 22 (2016) https://doi.org/10.3847/2041-8205/832/2/l22 1610.05359

Vargas, D.A., Falgout, R.D., Günther, S., Schroder, J.B.: Multigrid reduction in time for chaotic dynamical systems. arXiv e-prints, 2208–12629 (2022) arXiv:2208.12629. arXiv:2208.12629

Wisdom, J., Holman, M.: Symplectic maps for the n-body problem. The Astronomical Journal **102**, 1528 (1991) https://doi.org/10.1086/115978

Wang, L., Nitadori, K., Makino, J.: A slow-down time-transformed symplectic integrator for solving the few-body problem. Monthly Notices of the Royal Astronomical Society **493**(3), 3398–3411 (2020) https://doi.org/10.1093/mnras/staa480 2002.07938

Zeebe, R.E.: Highly stable evolution of earth's future orbit despite chaotic behavior of the solar system. The Astrophysical Journal **811**(1), 9 (2015) https://doi.org/10.1088/0004-637x/811/1/9 1508.04518

Zeebe, R.E.: Orbitn: A symplectic integrator for planetary systems dominated by a central mass—insight into long-term solar system chaos. The Astronomical Journal **166**(1), 1 (2023) https://doi.org/10.3847/1538-3881/acd63b 2306.03737

Zhang, K., Gladman, B.J.: Glisse: A gpu-optimized planetary system integrator with application to orbital stability calculations. arXiv e-prints, 2011–03484 (2020) arXiv:2011.03484. arXiv:2011.03484