





Swiftest: An N -body Integrator for Gravitational Systems

Carlisle Wishard ^{1,4*}, Jennifer Pouplin^{1,3}, Jacob Elliott¹, Dana Singh⁵, Kaustub Anand^{1,2}, and David Minton ^{1*}

1 Department of Earth, Atmospheric, and Planetary Sciences, Purdue University, USA **2** School of Aeronautics and Astronautics, Purdue University, USA **3** Department of Physics and Astronomy, Purdue University, USA **4** Evansville Museum of Arts, History & Science, USA **5** SAIC, Princeton, NJ, USA 
Corresponding author * These authors contributed equally.

DOI: [10.21105/joss.05409](https://doi.org/10.21105/joss.05409)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Warrick Ball](#) 

Reviewers:

- [@hannorein](#)
- [@mikegrudic](#)
- [@aatrani](#)

Submitted: 04 March 2023

Published: 10 October 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Swiftest is a software package designed to model the long-term dynamics of system of bodies in orbit around a dominant central body, such a planetary system around a star, or a satellite system around a planet. The main body of the program is written in Modern Fortran, taking advantage of the object-oriented capabilities included with Fortran 2003 and the parallel capabilities included with Fortran 2008 and Fortran 2018. Swiftest also includes a Python package that allows the user to quickly generate input, run simulations, and process output from the simulations. Swiftest uses a NetCDF output file format which makes data analysis with the Swiftest Python package a streamlined and flexible process for the user.

Statement of Need

Building off a strong legacy, including its predecessors Swifter ([Kaufmann, n.d.](#)) and Swift ([Levison & Duncan, 1994](#)), Swiftest takes the next step in modeling the dynamics of planetary systems by improving the performance and ease of use of software, and by introducing a new collisional fragmentation model. Currently, Swiftest includes the four main symplectic integrators included in its predecessors:

WHM Wisdom-Holman method. WHM is a symplectic integrator that is best suited for cases in which the orbiting bodies do not have close encounters with each other. For details see Wisdom & Holman ([1991](#)).

RMVS Regularized Mixed Variable Symplectic. RMVS is an extension of WHM that handles close approaches between test particles and massive bodies. For details, see Levison & Duncan ([1994](#)).

HELIO Democratic Heliocentric method. This is a basic symplectic integrator that uses democratic heliocentric coordinates instead of the Jacobi coordinates used by WHM. Like WHM it is not suited for simulating close encounters between orbiting bodies. For details, see Duncan et al. ([1998](#)).

SyMBA Symplectic Massive Body Algorithm. This is an extension of HELIO that handles close approaches between massive bodies and any of the other objects in the simulation. It also includes semi-interacting massive bodies that can gravitationally influence fully massive bodies but not each other. This algorithm is described in the Duncan et al. ([1998](#)). See also Levison & Duncan ([2000](#)).

All of the integrators that are currently implemented are based symplectic integrators, which are designed to model massive bodies (e.g. planets) or test particles (e.g. asteroids) in orbit of

a single dominant central body (e.g. the Sun) for very long periods of times (e.g. Gy). The core components of *Swiftest* were inherited from the *Swift*/*Swifter* codebase, but have been somewhat modified for performance. The *Swift*-family of integrators are most comparable to other symplectic integrators, which are among the most popular numerical methods used to study the long-term dynamics of planetary systems.

The *SyMBA* integrator included in *Swiftest* is most similar to the hybrid symplectic integrator *MERCURY6* (Chambers, 1999), the *MERCURIUS* integrator of *REBOUND* (Rein & Liu, 2012; Rein & Tamayo, 2015), and the GPU-enabled hybrid symplectic integrators such as *QYMSYM* (Moore & Quillen, 2011) and *GENGA II* (Grimm et al., 2022), with some important distinctions. The hybrid symplectic integrators typically employ a symplectic method, such as the original *WHM* method in Jacobi coordinates or the modified method that uses the Democratic Heliocentric coordinates, only when bodies are far from each other relative to their gravitational spheres of influence (some multiple of a Hill's sphere). When bodies approach each other, the integration is smoothly switched to a non-symplectic method, such as *Bulirsch-Stoer* or *IAS15*. In contrast, *SyMBA* is a multi-step method that recursively subdivides the step size of bodies undergoing close approach with each other.

In addition *Swiftest* contains a number of significant enhancements relative to its predecessors:

- It includes a fully symplectic implementation of the post-Newtonian correction (General Relativity) as described in Saha & Tremaine (1994).
- Output data is stored in *NetCDF4* (*HDF5*) format, making data analysis, cross-platform compatibility, and portability far more robust than the flat binary file formats used by its predecessors.
- It makes use of CPU-based parallelization via *OpenMP*. It also makes use of *SIMD* instructions, but these are available only when compiled for target host machine rather than when installed from *PyPI* via *pip*.
- Simulations can be run by means of a standalone binary driver program, similar to its predecessors, or from a compiled Python module. Both the standalone driver and Python module link to the same compiled library, so simulations run by either method are identical.
- It comes with an extensive set of Python scripts to help generate simulation initial conditions and post-process simulation results.
- It includes the *Fraggle* collisional fragmentation model that is used to generate collisional fragments on trajectories that conserve linear and angular momentum and lose the appropriate amount of collisional energy for inelastic collisions between massive bodies in *SyMBA* simulations. The collisional outcome is determined using standard methods based on the work of Leinhardt & Stewart (2012) and Stewart & Leinhardt (2012).

Performance

Modeling the behavior of thousands of fully interacting bodies over long timescales is computationally expensive, with typical runs taking weeks or months to complete. The addition of collisional fragmentation can quickly generate hundreds or thousands of new bodies in a short time period, creating further computational challenges for traditional *n*-body integrators. As a result, enhancing computational performance was a key aspect of the development of *Swiftest*. Here we show a comparison between the performance of *Swift*, *Swifter-OMP* (a parallel version of *Swifter*), and *Swiftest* on simulations with 1k, 2k, 8k, and 16k fully interacting bodies. The number of cores dedicated to each run is varied from 1 to 24 to test the parallel performance of each program.

[Figure 1](#) shows the results of this performance test. We can see that *Swiftest* outperforms *Swifter-OMP* and *Swift* in each simulation set, even when run in serial. When run in parallel, *Swiftest* shows a significant performance boost when the number of bodies is increased. The improved performance of *Swiftest* compared to *Swifter-OMP* and *Swift* is a critical step

forward in n -body modeling, providing a powerful tool for modeling the dynamical evolution of planetary systems.

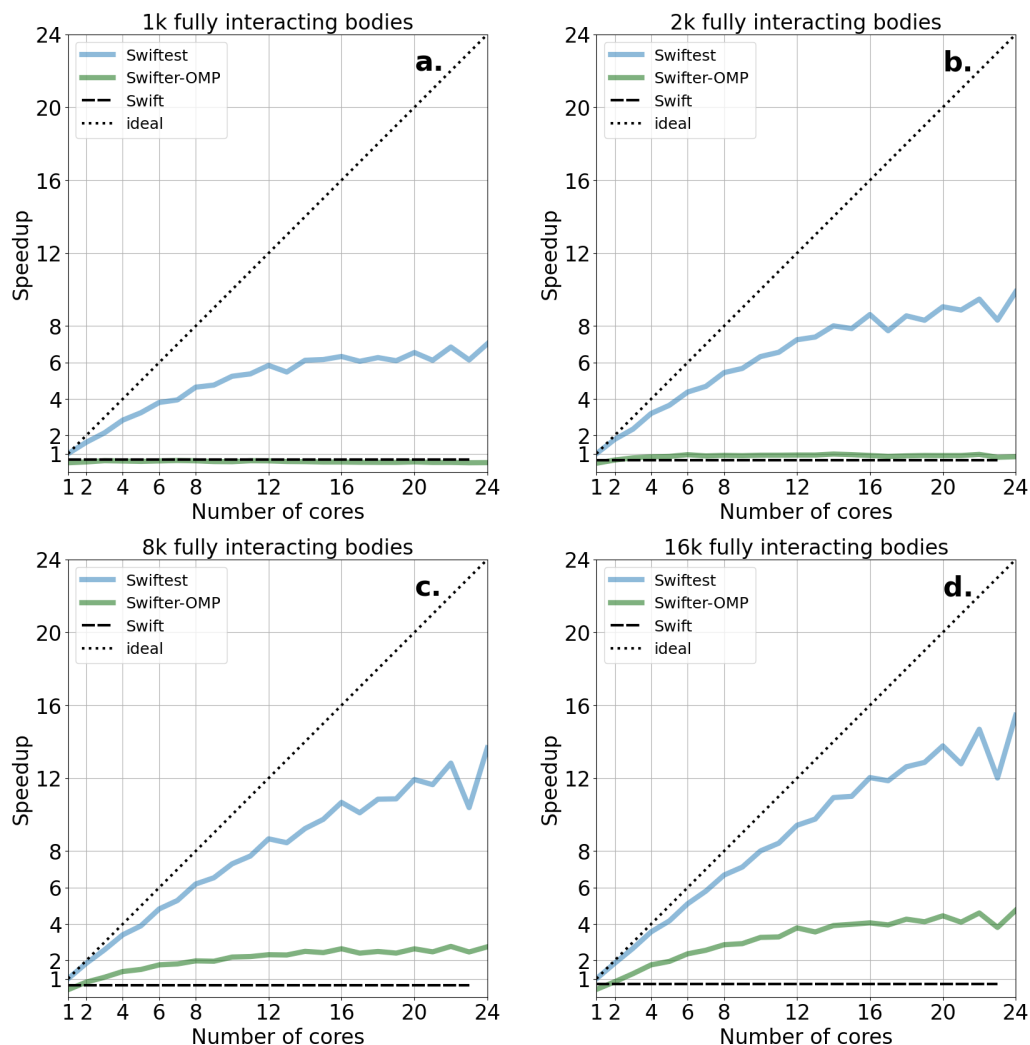


Figure 1: Performance testing of Swiftest on systems of (a) 1k, (b) 2k, (c) 8k, and (d) 16k fully interacting massive bodies. All simulations were run using the *SyMBA* integrator included in Swift, Swifter-OMP (and earlier attempt to parallelize Swifter), and Swiftest. Speedup is measured relative to Swiftest for 1 CPU (dashed), with an ideal 1:1 speedup shown as an upper limit (dotted). The performance of Swifter-OMP is shown in green while the performance of Swiftest is shown in blue. All simulations were run on the Purdue University Rosen Center for Advanced Computing Brown Community Cluster. Brown contains 550 Dell compute nodes, with each node containing 2 12-core Intel Xeon Gold Sky Lake processors, resulting in 24 cores per node. Each node has 96 GB of memory.

Acknowledgements

Swiftest was developed at Purdue University and was funded under the NASA Emerging Worlds and Solar System Workings programs. Active development by the Purdue Swiftest Team is ongoing and contributions from the community are highly encouraged.

References

- Chambers, J. E. (1999). A hybrid symplectic integrator that permits close encounters between massive bodies. *Monthly Notices of the Royal Astronomical Society*, 304(4), 793–799. <https://doi.org/10.1046/j.1365-8711.1999.02379.x>
- Duncan, M. J., Levison, H. F., & Lee, M. H. (1998). A Multiple Time Step Symplectic Algorithm for Integrating Close Encounters. *The Astronomical Journal*, 116(4), 2067–2077. <https://doi.org/10.1086/300541>
- Grimm, S. L., Stadel, J. G., Brasser, R., Meier, M. M. M., & Mordasini, C. (2022). GENGA. II. GPU Planetary N-body Simulations with Non-Newtonian Forces and High Number of Particles. *The Astrophysical Journal*, 932(2), 124. <https://doi.org/10.3847/1538-4357/ac6dd2>
- Kaufmann, D. E. (n.d.). Swifter. In *Swifter — an improved solar system integration software package*. Retrieved October 6, 2023, from <https://www.boulder.swri.edu/swifter/>
- Leinhardt, Z. M., & Stewart, S. T. (2012). Collisions between Gravity-dominated Bodies. I. Outcome Regimes and Scaling Laws. *The Astrophysical Journal*, 745(1), 79. <https://doi.org/10.1088/0004-637X/745/1/79>
- Levison, H. F., & Duncan, M. (1994). The Long-Term Dynamical Behavior of Short-Period Comets. *Icarus*, 108. <https://doi.org/10.1006/icar.1994.1039>
- Levison, H. F., & Duncan, M. J. (2000). Symplectically Integrating Close Encounters with the Sun. *The Astronomical Journal*, 120(4), 2117–2123. <https://doi.org/10.1086/301553>
- Moore, A., & Quillen, A. C. (2011). QYMSYM: A GPU-accelerated hybrid symplectic integrator that permits close encounters. *New Astronomy*, 16(7), 445–455. <https://doi.org/10.1016/j.newast.2011.03.009>
- Rein, H., & Liu, S. F. (2012). REBOUND: An open-source multi-purpose N-body code for collisional dynamics. *Astronomy & Astrophysics*, 537, A128. <https://doi.org/10.1051/0004-6361/201118085>
- 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>
- Saha, P., & Tremaine, S. (1994). Long-term planetary integration with individual time steps. *The Astronomical Journal*, 108, 1962–1969. <https://doi.org/10.1086/117210>
- Stewart, S. T., & Leinhardt, Z. M. (2012). Collisions between Gravity-dominated Bodies. II. The Diversity of Impact Outcomes during the End Stage of Planet Formation. *The Astrophysical Journal*, 751(1), 32. <https://doi.org/10.1088/0004-637X/751/1/32>
- Wisdom, J., & Holman, M. (1991). Symplectic maps for the n-body problem. *The Astronomical Journal*, 102, 1528–1538. <https://doi.org/10.1086/115978>