PHYSICS CLUB
OCTOBER STEM SCHOOL

COMPASS

Optimization and Comparative Analysis of N-Body Algorithms with Collision-Driven Shockwave Dynamics

Yahia Khaled, Youssef Ibrahim

February 8, 2026

STEM October Physics Club, COMPASS Computational Physics Program

1 Abstract

This work presents a high-performance computational physics implementation of the gravitational N-Body problem, integrating three major acceleration algorithms: the Direct method, the Barnes–Hut tree algorithm, and the Particle–Mesh (PM) FFT-based solver. A physically motivated collision–shockwave model is incorporated to simulate nonlinear interactions among bodies. The system is evolved using a Velocity Verlet integrator, and diagnostics including total energy, angular momentum, collision statistics, and trajectory histories are recorded for performance and accuracy evaluation. Comprehensive parameter sweeps across N (14–200), Barnes–Hut opening angle θ, and PM grid resolution are conducted to determine optimal configurations. This paper summarizes the numerical methods, simulation design, and comparison framework.

Keywords: N-Body simulation; Barnes–Hut; Particle–Mesh; gravitational dynamics; shockwave model; Velocity Verlet; computational physics

2 Introduction

The gravitational N-Body problem is central to computational astrophysics and cosmological modeling. The Newtonian formulation requires evaluating all pairwise interactions, which scales as $O(N^2)$ and becomes computationally expensive for large systems.

This project implements a unified simulation framework combining:

1. The **Direct** summation method,
2. The **Barnes–Hut** hierarchical tree algorithm,
3. The **Particle–Mesh** (PM) FFT-based algorithm.

A shockwave-based collision model is introduced to mimic nonlinear physical interactions. The system also tracks energy conservation, angular momentum, collision counts, and runtime performance. Parameter sweeps identify optimal Barnes–Hut opening angles and PM grid sizes.

3 Methods

This section describes the numerical methodology used to simulate gravitational N-Body systems and to compare different force evaluation algorithms. A unified simulation framework was developed to ensure that all methods share identical physical models, integration schemes, and diagnostic procedures. This approach allows for a controlled and meaningful comparison between algorithms.

3.1 Overall Simulation Framework

All simulations follow the same computational pipeline. At each timestep, gravitational forces are computed using one of the selected algorithms. Particle accelerations are then obtained from these forces and used to advance the system in time via a symplectic integration scheme. Diagnostic quantities including total energy, angular momentum, collision count, and runtime are recorded at every step.

This unified framework ensures that observed differences in performance or accuracy arise solely from the force calculation strategy rather than from inconsistencies in numerical integration or physical modeling.

3.2 Physical Model and Force Softening

Gravitational interactions are modeled using Newtonian gravity. To prevent numerical divergences during close encounters, a Plummer-type softening length ϵ is introduced. The softened gravitational force between particles i and j is given by

$$\mathbf{F}_{ij} = -G\frac{m_i m_j}{\left(r_{ij}^2 + \epsilon^2\right)^{3/2}}(\mathbf{r}_i - \mathbf{r}_j), \tag{3.1}$$

where G is the gravitational constant and r_{ij} is the separation distance. This softening ensures numerical stability while preserving the large-scale behavior of the system.

3.3 Time Integration Scheme

The equations of motion are integrated using the Velocity Verlet algorithm, a second-order symplectic integrator widely used in gravitational and molecular dynamics simulations. At each timestep Δt, particle positions and velocities are updated according to

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\Delta t + \frac{1}{2}\mathbf{a}_n\Delta t^2, \tag{3.2}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{1}{2}(\mathbf{a}_n + \mathbf{a}_{n+1})\Delta t. \tag{3.3}$$

The symplectic nature of this integrator provides good long-term energy conservation, which is essential for evaluating algorithmic accuracy.

3.4 Direct Force Summation

The Direct method computes gravitational forces by explicitly evaluating all pairwise interactions between particles. The acceleration of particle i is obtained as

$$\mathbf{a}_i = \sum_{j\neq i}\frac{\mathbf{F}_{ij}}{m_i}. \tag{3.4}$$

Although this approach yields the most accurate force calculation, its computational complexity scales as $O(N^2)$, making it impractical for large particle counts. In this study, the Direct method serves as a reference solution against which approximate algorithms are compared.

3.5 Barnes–Hut Tree Algorithm

To reduce computational complexity, the Barnes–Hut algorithm organizes particles into a hierarchical tree structure. In two-dimensional simulations, a quadtree is constructed by recursively subdividing the simulation domain until each cell contains at most one particle.

During force evaluation, distant groups of particles are approximated by their center of mass when the opening criterion

$$\frac{s}{d} < \theta \tag{3.5}$$

is satisfied, where s is the size of the tree cell, d is the distance to the cell's center of mass, and θ is the opening angle parameter. Smaller values of θ yield higher accuracy at the expense of increased computational cost.

This hierarchical approximation reduces the overall complexity to approximately $O(N \log N)$, enabling simulations with significantly larger particle counts.

3.6 Particle–Mesh Method

The Particle–Mesh (PM) method accelerates force computation by solving Poisson's equation on a fixed spatial grid. Particle masses are first deposited onto the grid using a Cloud-In-Cell (CIC) interpolation scheme. In 2D, the gravitational potential ϕ is then obtained by solving

$$\nabla^2\phi = 2\pi G\Sigma \tag{3.6}$$

in Fourier space using fast Fourier transforms, where Σ is the surface mass density. The potential in k-space is given by

$$\phi(\mathbf{k}) = -\frac{2\pi G\Sigma(\mathbf{k})}{|\mathbf{k}|}. \tag{3.7}$$

Forces are computed by differentiating the potential on the grid and interpolating them back to particle positions. While the PM method efficiently captures long-range interactions, its accuracy is limited by the chosen grid resolution and its inability to resolve close encounters without additional refinement.

3.7 Collision Detection and Shockwave Model

In addition to gravitational interactions, a collision model is implemented to simulate close encounters between particles. A collision is detected when the distance between two particles falls below a predefined threshold d_{coll}.

Upon collision, the interacting particles are merged, conserving total mass and linear momentum. To model energetic interactions, a shockwave impulse is applied to nearby particles. The resulting velocity perturbation is defined as

$$\Delta\mathbf{v} = S\,\frac{M_{\text{pair}}}{m}\exp\left(-\frac{r}{L}\right)\hat{\mathbf{r}}, \tag{3.8}$$

where S is a scaling factor, M_{pair} is the combined mass of the colliding particles, L is a decay length, and $\hat{\mathbf{r}}$ is the radial unit vector. This model introduces controlled nonlinearity while maintaining numerical stability.

3.8 Diagnostics and Performance Measurement

Throughout each simulation, diagnostic quantities are recorded at every timestep. These include total kinetic and potential energy, total angular momentum, cumulative collision count, and wall-clock runtime. Trajectory histories are also stored for visualization and qualitative analysis.

This comprehensive diagnostic framework enables direct comparison of accuracy, stability, and computational performance across all implemented algorithms.

3.9 Energy Diagnostics

Kinetic and potential energies are:

$$K = \sum_i \frac{1}{2} m_i |\mathbf{v}_i|^2, \tag{3.9}$$

$$U = -\sum_{i<j} \frac{G m_i m_j}{r_{ij}}. \tag{3.10}$$

Total energy:

$$E = K + U.$$

3.10 Angular Momentum

$$\mathbf{L} = \sum_i m_i(\mathbf{r}_i \times \mathbf{v}_i). \tag{3.11}$$

4 Results

This section presents the numerical outputs obtained from the simulation framework. All figures shown below are generated directly from the stored simulation data and are organized to highlight performance, accuracy, and qualitative behavior of each algorithm.

4.1 Runtime Scaling

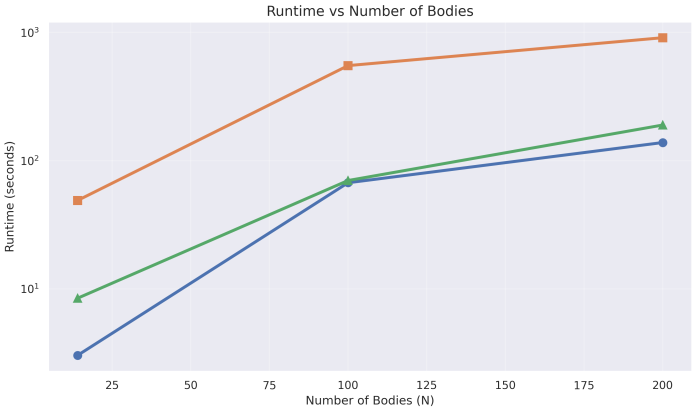

Figure 1: Runtime scaling with respect to the number of bodies N for the Direct, Barnes–Hut, and Particle–Mesh methods.

4.2 Energy Conservation

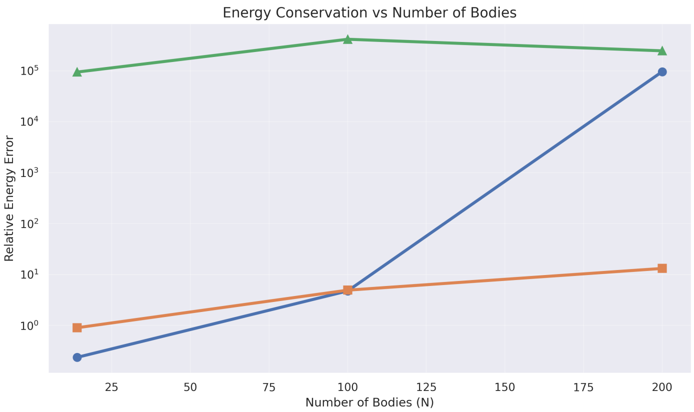

Figure 2: Relative total energy error as a function of time for each numerical method.

4.3 Barnes–Hut Parameter Optimization

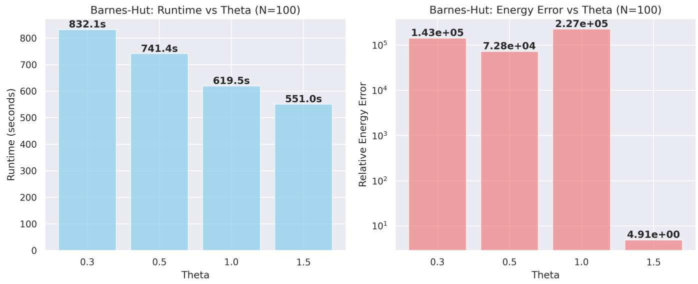

Figure 3: Barnes–Hut runtime dependence on opening angle θ for fixed N, and Energy conservation behavior of the Barnes–Hut algorithm as a function of opening angle θ.

4.4 Particle–Mesh Grid Resolution Study

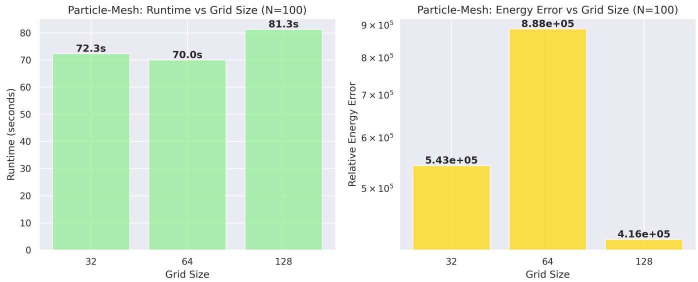

Figure 4: Particle–Mesh runtime as a function of grid resolution, and energy conservation performance of the Particle–Mesh method for different grid sizes.

4.5 Diagnostic Quantities

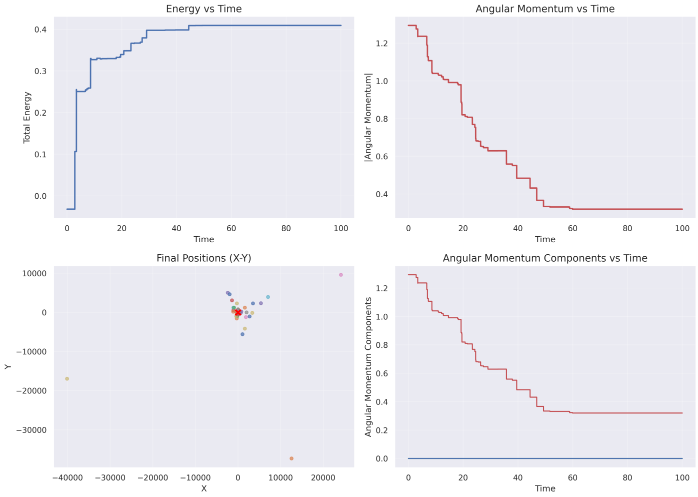

Figure 5: Time evolution of diagnostic quantities (energy, angular momentum, collision count) for the Barnes–Hut method with $N = 200$.

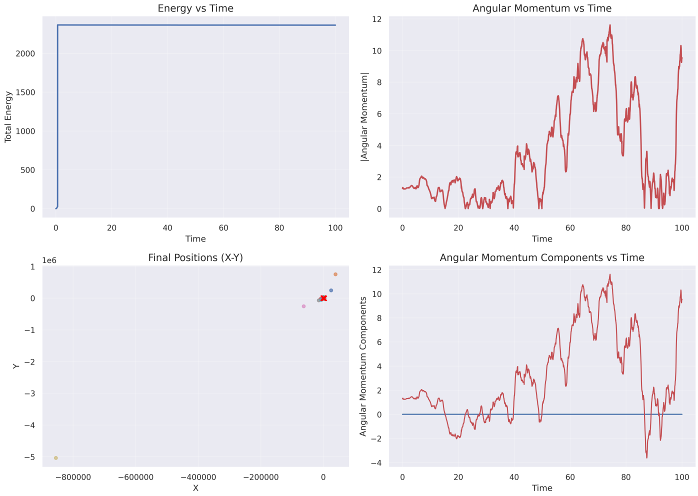

Figure 6: Time evolution of diagnostic quantities for the Particle–Mesh method with $N = 200$.

4.6 Trajectory Evolution

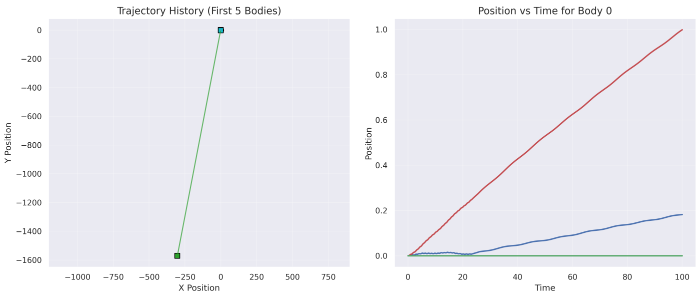

Figure 7: Orbital trajectory history of bodies simulated using the Barnes–Hut algorithm.

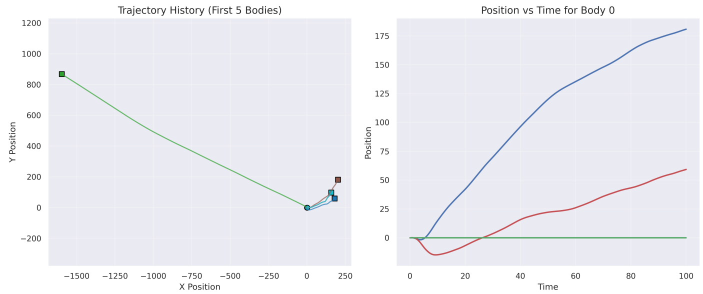

Figure 8: Orbital trajectory history of bodies simulated using the Particle–Mesh algorithm.

5 Discussion

The numerical results obtained in this study highlight the trade-offs between accuracy, computational cost, and scalability among the three implemented N-Body algorithms. While all methods reproduce the qualitative gravitational dynamics of the system, their performance and conservation properties differ substantially as the number of bodies increases and as algorithm-specific parameters are varied.

5.1 Computational Performance and Scaling

The runtime scaling results clearly demonstrate the expected theoretical behavior of each method. The Direct summation approach exhibits a steep increase in computational cost with increasing N, consistent with its $O(N^2)$ complexity. Although this method provides the highest force accuracy, its rapidly growing runtime limits its practical applicability to relatively small systems.

In contrast, the Barnes–Hut algorithm significantly reduces computational cost by approximating distant particle groups as single pseudo-masses. The observed near $O(N \log N)$ scaling enables simulations with substantially larger particle counts while maintaining reasonable accuracy. However, this efficiency gain depends sensitively on the choice of the opening angle θ, which controls the balance between approximation error and computational speed.

The Particle–Mesh method exhibits the most favorable scaling for large N, with runtime dominated primarily by the grid resolution rather than the number of particles. This behavior makes the PM approach particularly suitable for large-scale simulations where long-range collective effects dominate the dynamics.

5.2 Energy Conservation and Numerical Stability

Energy conservation serves as a critical diagnostic for the long-term stability of gravitational simulations. The Direct method demonstrates the smallest relative energy drift, reflecting the absence of force approximations. The Barnes–Hut method shows a modest increase in energy error as the opening angle θ increases, indicating the expected trade-off between computational speed and force accuracy.

For the Particle–Mesh method, energy conservation is strongly influenced by the chosen grid resolution. Coarser grids introduce smoothing errors in the gravitational potential, leading to increased energy drift. Increasing the grid resolution improves accuracy but at the expense of increased computational cost, highlighting the importance of selecting an optimal grid size based on the problem scale.

The use of the Velocity Verlet integrator contributes to the overall stability of all simulations by preserving the symplectic structure of the equations of motion. This property is particularly important for long integration times, where non-symplectic schemes may exhibit systematic energy drift.

5.3 Effect of Collision and Shockwave Modeling

The inclusion of a collision-driven shockwave model introduces controlled nonlinearity into the system and allows the study of energy redistribution during close encounters. Collision events lead to localized perturbations in velocity and angular momentum, which are clearly reflected in the diagnostic plots.

While these shock-induced impulses intentionally violate strict energy conservation, they serve as a physically motivated mechanism for modeling dissipative or explosive interactions. The resulting deviations remain bounded and do not destabilize the overall simulation, indicating that the collision model is numerically robust within the chosen parameter regime.

5.4 Algorithm Suitability and Practical Implications

From a practical perspective, the results suggest that no single method is universally optimal. The Direct method is best suited for small systems where accuracy is paramount. The Barnes–Hut algorithm provides an effective compromise between accuracy and performance and is well suited for intermediate system sizes. The Particle–Mesh approach excels in large-scale simulations dominated by long-range gravitational interactions but is less effective at resolving close encounters without additional refinement techniques.

The parameter sweep analyses further emphasize the importance of algorithm tuning. Selecting appropriate values for the Barnes–Hut opening angle and PM grid resolution is essential to achieving reliable results while minimizing computational cost.

Overall, the comparative framework developed in this project provides a flexible platform for exploring gravitational dynamics across multiple scales and offers clear guidance for selecting suitable numerical methods based on the physical and computational requirements of a given problem.

6 Conclusion

This project presents a complete implementation and comparison of three gravitational N-Body algorithms enhanced with a shockwave collision model. The Direct method is most accurate but slow. Barnes–Hut achieves excellent scaling for large N, and the PM method handles large-scale collective dynamics efficiently. Parameter sweeps allowed identifying optimal configurations for fast and accurate simulation.

Bibliography

[1] Barnes, J., and Hut, P.: A Hierarchical $O(N \log N)$ Force-Calculation Algorithm, *Nature*, Vol. 324, 1986, pp. 446–449.

[2] Binney, J., and Tremaine, S.: *Galactic Dynamics*, 2nd ed., Princeton University Press, Princeton, 2008.

[3] Hockney, R. W., and Eastwood, J. W.: *Computer Simulation Using Particles*, Taylor & Francis, New York, 1988.

[4] Springel, V.: The Cosmological Simulation Code GADGET-2, *Mon. Not. R. Astron. Soc.*, Vol. 364, 2005, pp. 1105–1134.

[5] Hairer, E., Lubich, C., and Wanner, G.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, Berlin, 2006.

[6] Monaghan, J. J.: Smoothed Particle Hydrodynamics, *Annu. Rev. Astron. Astrophys.*, Vol. 30, 1992, pp. 543–574.

[7] Aarseth, S. J.: *Gravitational N-Body Simulations: Tools and Algorithms*, Cambridge University Press, Cambridge, 2003.

[8] Eastwood, J. W., and Hockney, R. W.: Shaping Algorithms for N-Body Simulations, *Comput. Phys. Commun.*, Vol. 6, No. 1, 1974, pp. 1–16.

[9] Birdsall, C. K., and Langdon, A. B.: *Plasma Physics via Computer Simulation*, CRC Press, Boca Raton, 2004.

[10] Swope, W. C., Andersen, H. C., Berens, P. H., and Wilson, K. R.: A Computer Simulation Method for the Calculation of Equilibrium Constants, *J. Chem. Phys.*, Vol. 76, No. 1, 1982, pp. 637–649.

[11] Rosswog, S.: Astrophysical Smooth Particle Hydrodynamics, *New Astron. Rev.*, Vol. 53, Nos. 4–6, 2009, pp. 78–104.