N-Body Simulation - Barnes-Hut Approximation

Measuring Performance Improvements with Quadtrees

Jared Michael Wildermuth & Kalju Jake Nekvasil

Institute of Computer Science, University of Tartu

Abstract

This project focuses on the implementation and analysis of the Barnes-Hut algorithm for n-body simulations, which model the motion of bodies under gravitational forces. Direct methods for computing interactions in these systems suffer from a high computational cost of $\mathcal{O}(n^2)$, making them inefficient for large-scale simulations. The Barnes-Hut algorithm overcomes this limitation by approximating distant body clusters as single points, reducing the computational complexity to $\mathcal{O}(n \log n)$. This is achieved through the use of a quadtree data structure to partition the simulation volume and efficiently calculate gravitational forces. The primary objectives of this project are benchmark the performance differences between the Barnes-Hut and brute-force methods and implement a functional body simulation using Barnes-Hut.

Introduction

The n-body problem is a classic challenge in physics and computational science, involving the prediction of the motion of bodies under gravitational forces. While direct methods for solving this problem are straightforward, they scale poorly. The Barnes-Hut algorithm addresses this limitation by approximating the gravitational influence of distant body clusters. This project implements a 2D n-body simulation using the Barnes-Hut algorithm and compares its performance to the brute-force method. By implementing a quadtree data structure, the simulation efficiently computes gravitational forces while maintaining a balance between accuracy and computational efficiency.

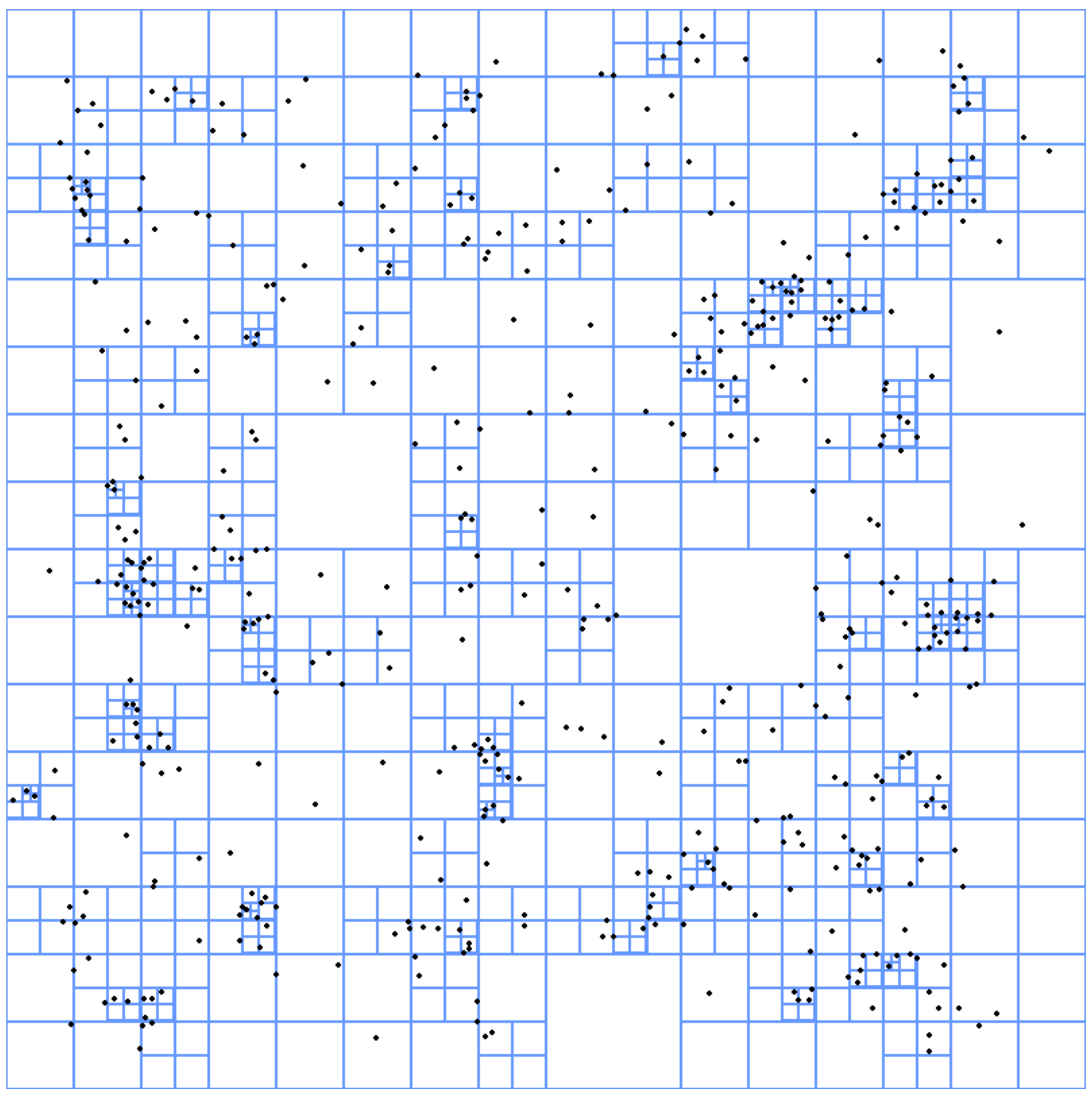

Figure 1: Quadtree visualization during a 500-body Barnes-Hut simulation.

Methods

Simulation Constants

The gravitational constant ($G = 1$) scales the strength of gravitational forces between bodies. The force between two bodies is calculated as:

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2} \tag{1}$$

where m_1 and m_2 are the masses of the bodies, and r is the distance between them. The Barnes-Hut parameter ($\theta = 0.5$) controls the accuracy of the force approximation. A cluster of bodies is treated as a single point if:

$$\frac{\text{width of cluster}}{\text{distance to body}} < \theta \tag{2}$$

A smaller θ increases accuracy but reduces performance, while a larger θ speeds up the simulation at the cost of accuracy. Finally, the time step ($\Delta t = 0.1$) defines the interval for updating body positions and velocities. Smaller time steps improve accuracy but increase computational cost.

Algorithmic Approach

The process begins with the initialization of bodies, which are randomly placed in the simulation space with random masses and initial velocities. Next, the simulation space is recursively subdivided using a quadtree data structure. Each node in the quadtree represents a region of space and stores the total mass and center of mass of the bodies within it. This hierarchical partitioning allows the algorithm to group bodies into clusters, significantly reducing the number of force calculations required. For each body, the gravitational force is computed by traversing the quadtree. Distant clusters of bodies are approximated as single points, using the Barnes-Hut parameter θ to determine when this approximation is appropriate. Nearby bodies, however, are treated individually to ensure accurate force calculations. Once the forces are calculated, they are applied to update the velocities of the bodies. The positions of the bodies are then updated based on these new velocities, using a time step Δt. Finally, any collisions between bodies are evaluated.

Results

The Barnes-Hut algorithm significantly outperforms the brute-force method for large numbers of bodies. For example, with $n = 1000$ bodies, the Barnes-Hut algorithm completes each simulation step in a fraction of the time required by the brute-force method. This is due to the algorithm's ability to approximate the gravitational influence of distant bodies clusters, reducing the computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. As a result, the brute-force method becomes impractical for large n, while the Barnes-Hut algorithm enables efficient simulations with thousands of bodies.

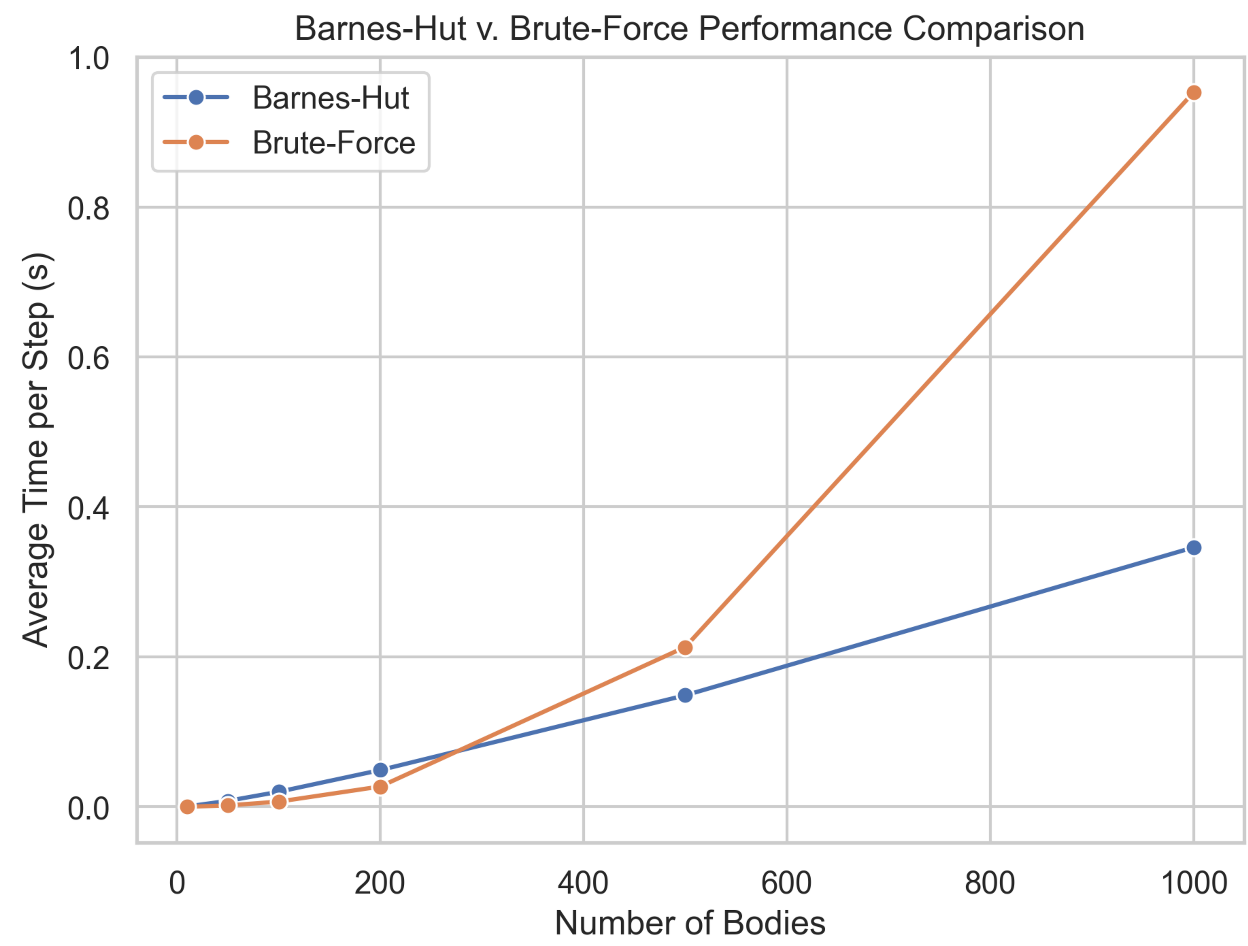

Figure 2: Brute-force's quadratic scaling becomes noticable for large n.

In terms of memory usage, the Barnes-Hut algorithm uses slightly more memory than the brute-force method due to the overhead of the quadtree data structure. However, this increase is minor and scales well with the number of bodies. Both methods show linear growth in memory usage with respect to (n), but the Barnes-Hut algorithm remains more efficient for large-scale simulations, as it avoids the quadratic scaling of force calculations.

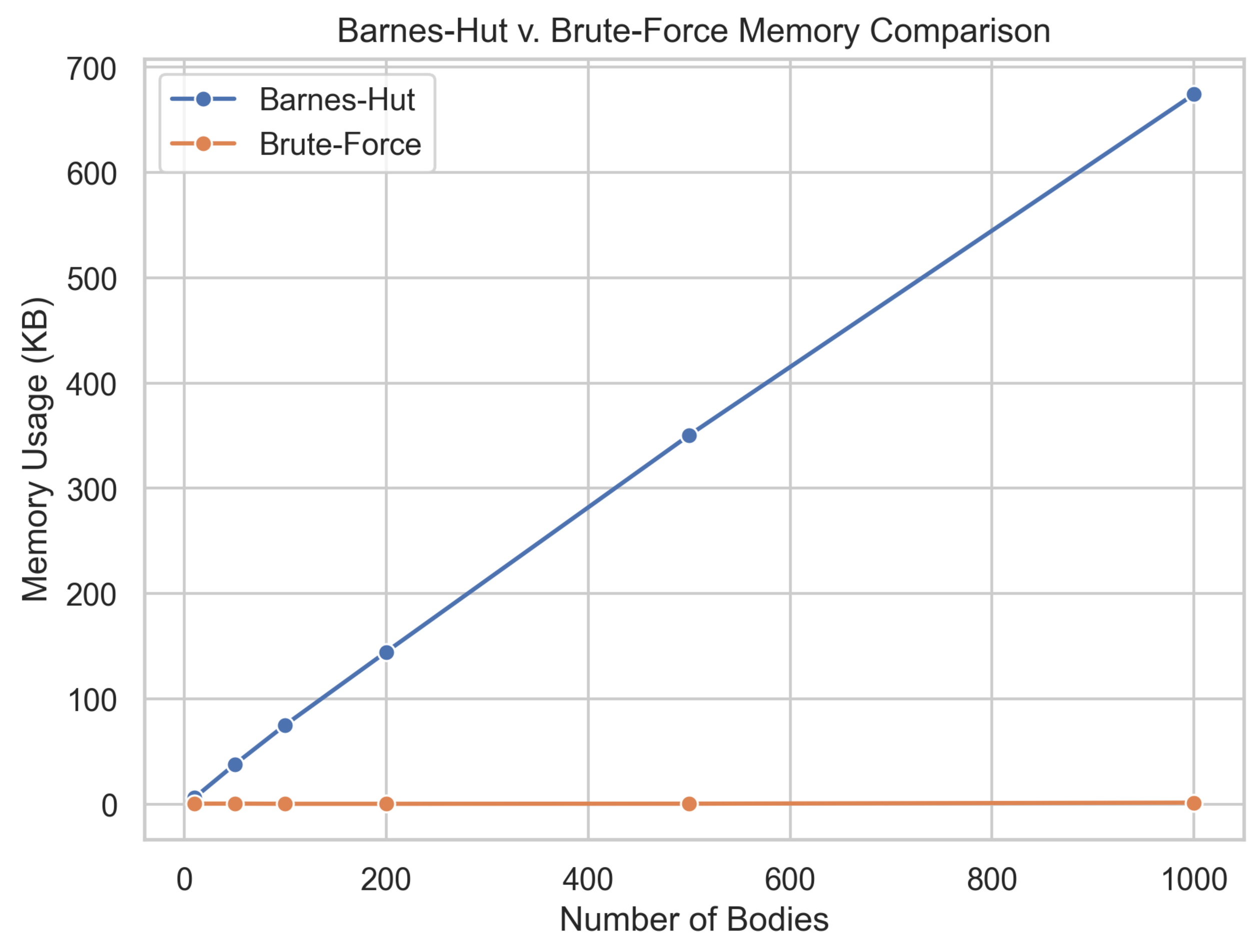

Figure 3: Although brute-force outperforms Barnes-Hut in terms of memory usage, both growth rates are linear.

Conclusion

The Barnes-Hut algorithm provides an efficient and scalable solution to the n-body problem, enabling simulations with thousands of bodies while maintaining reasonable accuracy. By approximating the gravitational influence of distant body clusters, the algorithm reduces computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$, making it suitable for large-scale simulations. The results demonstrate that the Barnes-Hut algorithm outperforms the brute-force method in both speed and scalability, with only a minor increase in memory usage.

Figure 4: Visualization of a 200 step, 500-body Barnes-Hut simulation.

References

[1] J. Barnes and P. Hut. *A Hierarchical $\mathcal{O}(n \log n)$ Force-Calculation Algorithm*, volume 324. Nature, 1986.