N-body galaxy with a migrating black hole

Chad Fairservice – 1008283956,
Daniel Harrington – 1006539355,
Nicholas Lucianetti – 1008427383,
Zac Zhang – 1006667224

PHYD57

Department of Physical and Environmental Sciences
University of Toronto Scarborough
November 28, 2024

Introduction

N-body Simulations and Galactic Mergers

For the better part of the last century, research in astrophysics and cosmology has made significant progress thanks to the advent of numerical modeling, particularly through the wide applicability of N-body simulations. From investigating the dynamics of a few-body system, to understanding the evolution of the large-scale structure of the universe, N-body simulations provide insight into a dynamical system of particles under the influence of physical forces. Conceived in the mid-20th century, N-body simulations have evolved alongside advances in computing power and more effective and suitable numerical integrators, allowing more complex interactions to be studied in a shorter amount of time.

The study of merging and interacting galaxies is one of the most dynamic areas of research in computational astrophysics and is a perfect candidate for investigation through N-body simulations. During such interactions, stars are rarely involved in direct collisions; instead, their motions are governed by the collective gravitational potential of the system as they gradually coalesce. These phenomena have been shown to be ubiquitous processes in the universe; observations reveal that many of the most massive galaxies and dense star clusters likely formed through a series of mergers.

A 2020 study by Adamo et al. [1], based on the HiPEEC (Hubble imaging Probe of Extreme Environments and Clusters) survey, examined star cluster formation in six merging galaxies. Using Hubble Space Telescope data, researchers were able to derive the ages and masses of star clusters, and estimate star formation rates. Massive clusters with ages ranging from 1 to 500 million years and masses exceeding 10^7 solar masses were discovered, with more advanced-stage galactic mergers hosting more massive clusters compared to earlier-stage mergers. These six merging galaxies were found to be among the most efficient star-forming regions in the local universe, and further investigation is necessary to better understand the unique physical conditions that drive this rapid cluster formation.

Challenges of N-body simulations

Despite the utility of N-body simulations, there are numerous considerations and challenges in modeling the interaction of galaxies. Chief among these is the computational complexity of modeling the gravitational interactions among millions or even billions of particles. Direct calculation of the gravitational force of each particle scales as N^2, making this method impractical for large systems. To address this, researchers employ efficient algorithms such as the Fast Fourier Transform (FFT) to evaluate forces on a grid or tree-based methods like the Barnes-Hut algorithm. Additionally, simulations must carefully balance resolution with computational feasibility. High spatial and temporal resolution is necessary to accurately capture the dynamics of dense regions, such as galactic centers or tidal tails, but this comes at a steep computational cost.

Another challenge is ensuring numerical accuracy and stability over long timescales. Small errors in force calculations or integration schemes can accumulate, leading to non-physical results. Techniques such as symplectic integration and adaptive time-stepping can be useful to maintain energy and momentum conservation while resolving both global and local dynamics.

Force Calculations and Integration Schemes

Computational astrophysicists studying galaxy and black hole mergers have used various techniques to achieve their desired accuracy and computational efficiency, and these choices largely depend on the nature of the research being conducted. In 2023, Hao et al. [2] investigated the merging process of super-massive black holes using a direct summation N-body simulation of 64,000 particles, which included 3 SMBHs that each made up 1% of the system's mass. By using a direct summation method to calculate forces and zero softening, an extremely high level of accuracy was maintained. The N-body code NBODY6++ was used to integrate the system, which employs a hybrid parallelization strategy, combining GPU acceleration, MPI, and OpenMP to dramatically increase computational efficiency.

For this project, we focus on the dynamics of a host galaxy interacting with a smaller, colliding galaxy. Using an FFT-based gravity solver and a symplectic integrator, we model the evolution of these systems and explore how the presence of SMBHs influences their post-merger configurations. We employ the 3D graphics software, Houdini, to visualize the galactic merger over 100000 timesteps.

Method

Particle Initialization and Interpolation to Density Array

A total of 10^8 particles are initialized on a $512 \times 512 \times 256$ grid. The `initialize_particles` subroutine arranges the particles in a spiral galaxy formation, where the tightness, pitch angle, spiral arm separation, and initial velocity are specified. At the center of this galaxy is a super-massive black hole represented by a particle of mass $1000m$, where $m = 1/(2N)$ is the mass of a single particle. A second black hole of mass $100m$ is introduced to simulate the interaction of two galaxies at an intermediate stage of their merger.

The `particles_to_grid` subroutine is the first to be called in each step, and its purpose is to interpolate the particles to a mass density array. This allows a continuous density distribution to be represented by discrete grid points that will later be used to compute the gravitational forces. Our method of choice is a linear interpolation known as cloud-in-cell. This method is first order but computationally inexpensive. It also conserves quantities like mass during interpolation; that is, the total mass in the grid is equal to the total mass of particles. This interpolation step is implemented by assigning weights to particle positions based on their proximity to grid nodes. Say a particle has mass m and position $\mathbf{r} = (x, y, z)$. We first calculate the fractional position of the particle within its cell. In the x dimension this looks like:

$$d_x = \frac{x - x_{\text{node}}}{\Delta x},$$

where x_{node} is the position of the node at the lower bound of the cell and Δx is the width of the cell. We then assign the particle's mass to the eight surrounding nodes using a trilinear weighting scheme, where weights are calculated based on the particle distance:

$$w_x(i) = \begin{cases} 1 - d_x, & \text{if } i = \text{ lower node} \\ d_x, & \text{if } i = \text{ upper node.} \end{cases}$$

An analogous formula is used for the y-direction ($w_y(j)$) and the z-direction ($w_z(k)$). These weights are combined to determine the density contribution to a given node:

$$\rho_{\text{node}} = m \cdot w_x(i) \cdot w_y(j) \cdot w_z(k).$$

Figure 1 (left) below shows the discrete particles of an initialized spiral galaxy, visualized in the 3D graphics software, Blender. On the right, we see the interpolated density grid visualized as a heat map. The super-massive black holes contributes substantially more mass, and thus greater density to their surrounding nodes.

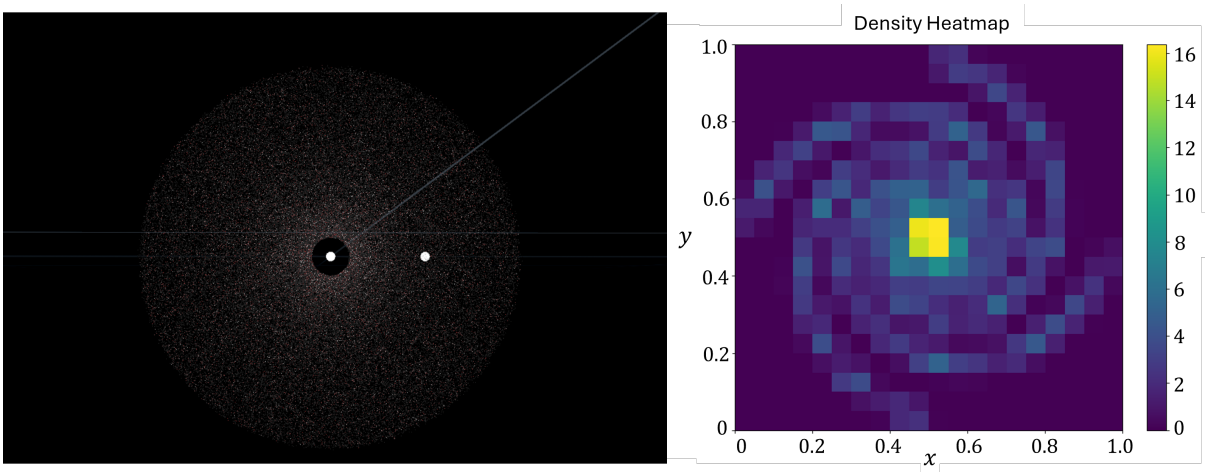

Figure 1: Left: Blender screenshot of the initialized galaxy. Right: Example of an interpolated density grid of a previous version of the initialized system.

Fast Fourier Transforms and Computing Accelerations

Once the particles have been interpolated to a density grid, the `compute_accelerations` subroutine employs the Nvidia cuFFT library to perform a forward FFT. The cuFFT library makes use of GPU parallelization to perform an FFT algorithm with $N\log(N)$ efficiency, greatly improving upon the N^2 efficiency of direct summation. At this stage the density field has been converted to its frequency components $\hat{\rho}(\mathbf{k})$, where $\mathbf{k} = (k_x, k_y, k_z)$ is the wave vector. To compute the force field, we must retrieve these wavevector components. For a 3D grid of size $N_x \times N_y \times N_z$, the wave numbers are

$$k_x = \frac{2\pi n_x}{L_x}, \qquad k_y = \frac{2\pi n_y}{L_y}, \qquad k_z = \frac{2\pi n_z}{L_z},$$

where n_x, n_y, and n_z represent the indices of the Fourier transform grid and L_x, L_y, and L_z are the physical lengths of the simulation domain. In real space, the gravitational force $\mathbf{F}$ is the negative gradient of the gravitational potential Φ,

$$\mathbf{F} = -\nabla\mathbf{\Phi},$$

however, in frequency space, the gradient operator becomes a multiplication by $i\mathbf{k}$, and the force components are calculated as

$$\hat{\mathbf{F}}_i(\mathbf{k}) = ik_i\hat{\mathbf{\Phi}}(\mathbf{k}).$$

In frequency space, the potential is defined as

$$\hat{\Phi} = \frac{4\pi G}{k^2}\hat{\rho}(\mathbf{k}),$$

and the final force computation becomes

$$\hat{\mathbf{F}}_i(\mathbf{k}) = -i\frac{4\pi G k_i}{k^2}\hat{\rho}(\mathbf{k}).$$

It is also important to note that when using cuFFT to transform data, input and output data sizes will differ. Both real to complex and complex to real transformations require a separate array to be created to account for this discrepancy. The newly computed force grid is mapped onto the density grid to save memory, as this process must be performed every time step. The cuFFT library is once again used to perform an inverse FFT so that we can apply the forces in real space.

After the accelerations have been calculated using the density grid, they are interpolated back into the particles by the `grid_to_particles` subroutine. This subroutine calculates the weights of the particles in the exact same way as mentioned in the previous interpolation step. It then uses those weights to interpolate the accelerations—which are located on all eight nodes of the cube—to the particles. For example, if a_x is the x-component of a particle's acceleration, we have

$$a_{\mathrm{x}} = a_{\mathrm{x}} + acceleration\_grid(i, j, k) \cdot w_x(i) \cdot w_y(j) \cdot w_z(k),$$

$$a_{\mathrm{x}} = a_{\mathrm{x}} + acceleration\_grid(i+1, j, k) \cdot w_x(i+1) \cdot w_y(j) \cdot w_z(k).$$

Since there are 100 million particles in the simulation, both subroutines are parallelized using CUDA.

Integration Scheme

We use a 2nd-order leapfrog scheme to integrate the simulation forward in time by using the accelerations calculated in the previous step. We solve the equation of motion

$$\ddot{\mathbf{r}} = \frac{d^2\mathbf{r}}{dt^2} = A(\mathbf{r}).$$

If we consider the acceleration of some particle at time step i, a_i, the first stage of the integration is to calculate an initial "kick" on the velocity, half a time step ahead:

$$v_{i+1/2} = v_i + a_i\frac{\Delta t}{2}.$$

This is then used to calculate the position of the particle a full step ahead, in what is called the "drift" step:

$$x_{i+1} = x_i + v_{i+1/2}\Delta t.$$

Finally, in the second kick step, we update the velocity of the particle one time step ahead:

$$v_{i+1} = v_{i+1/2} + a_{i+1}\frac{\Delta t}{2}.$$

The benefit of using the leapfrog integration scheme in gravitational dynamics simulations is its ability to handle a large number of particles while maintaining a reasonable balance of accuracy and efficiency. The leapfrog scheme is a symplectic integrator, and as such, energy and momentum can be conserved over long timescales. Leapfrog's 2nd order accuracy is better than 1st order schemes such as Euler's method, while still achieving the order $O(n)$ efficiency required to simulate two galaxies of order 10^8 particles.

Additionally, in galactic merger simulations, strong gravitational interactions frequently occur in dense galactic cores and in very close encounters between stars. In non-symplectic methods, small energy errors can grow exponentially during strong interactions, potentially destabilizing the simulation.

Challenges and Techniques Used to Overcome Them

1. GPU bandwidth limitations/host-device transfer bottleneck: Instead of simply parallelizing the heavy computation and transferring back to the CPU for simpler tasks, it is essential to keep all large datasets (ie. particles and grids) on device memory after initialization. Since the data is roughly 7GB, even at a peak theoretical transfer speed of 11GB/s, this amounts to over half a second of latency per step. As N approaches 100 million, this latency is often longer than the computation itself. Keeping data on the GPU eliminates this transfer latency entirely. Conversely, for scalars, pass by value from the host is often faster than pass by reference, as the number of GPU cycles is lower and the transfer data is miniscule.

2. Using cuFFT optimal grid sizes: The cuFFT library is optimized for grid sizes that are products of small prime factors, with powers of two being the most efficient. This is because the library uses specialized algorithms to exploit regular factorization patterns in the input size. For this reason, we stick to a $512 \times 512 \times 256$ grid.

3. Dimension difference of FFT transform grid due to Hermitian symmetry: A dimensional distinction must be accounted for when creating the gravity grid in complex space using the cuFFT transform functions. This is discussed in detail on the following page: https://docs.nvidia.com/cuda/cufft/index.html#multidimensional-transforms

Results

Visualization of the Merger

The evolution of the galactic merger is visualized over 10000 timesteps, highlighting the dynamics of the SMBH interactions and the eventual coalescence of the two galaxies. The visualizations reveal distinct phases in the merging process:

1. Initial mass ejection: At the very start of the simulation, a large number of particles are ejected from the system, which is partially expected from Jeans equations. We believe this may also be due to a combination of the masses used for the black holes and the initial velocities of the particles, and that this is a somwhat unphysical result. In a true galaxy merger, there would be a loss of mass due to tidal stripping as the galaxies pass through one another.

2. Core coalescence phase: A fraction of particles do not escape from the system in the initial mass loss phase. The black holes approach one another and the remaining particles in the system orbit these black holes.

3. Post-merger configuration: The final configuration of the system shows a super-massive black hole binary that continues to orbit at the same distance for the duration of the simulation. A fraction of the initial particle distribution remains in orbit.

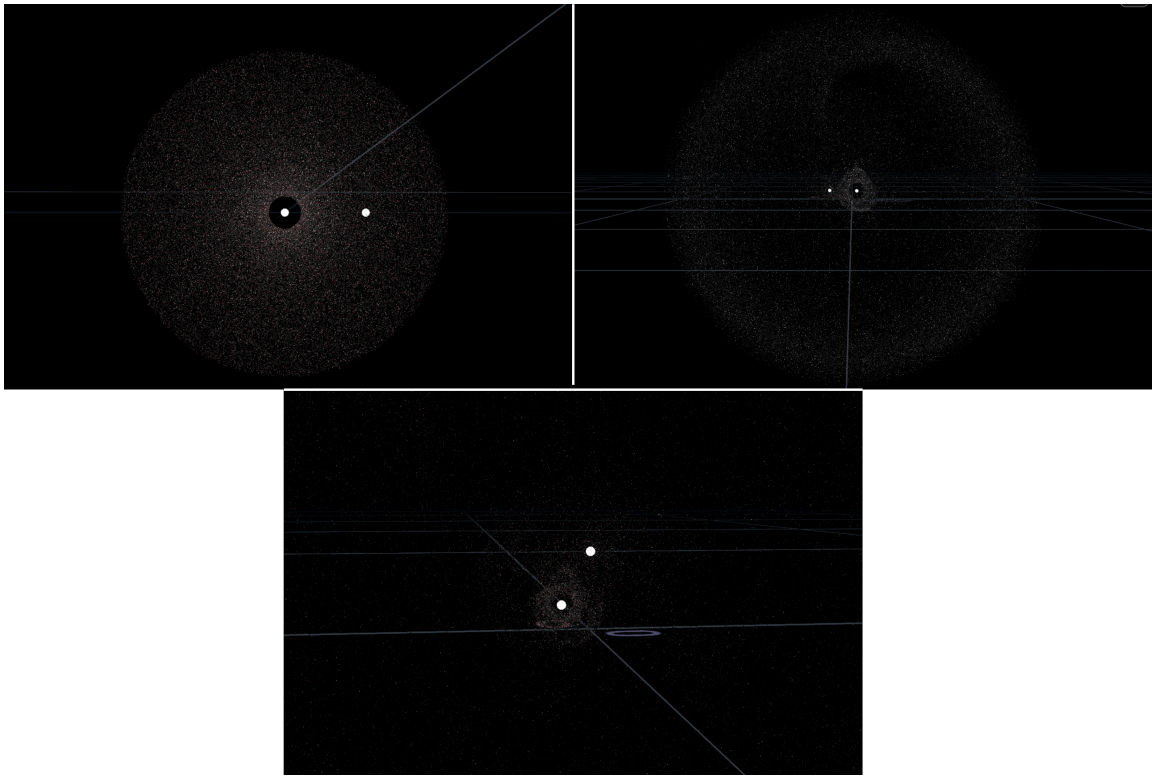

Figure 2: The initial distribution of particles (top left), an intermediate stage of the merger (top right), and the final timestep (bottom) are shown above.

Energy Conservation

We were not able to achieve the desired energy conservation in our final simulation. We consistently have a stable energy error of approximately 0.27 with small fluctuations. An energy timeseries for 1000 timesteps is included below.

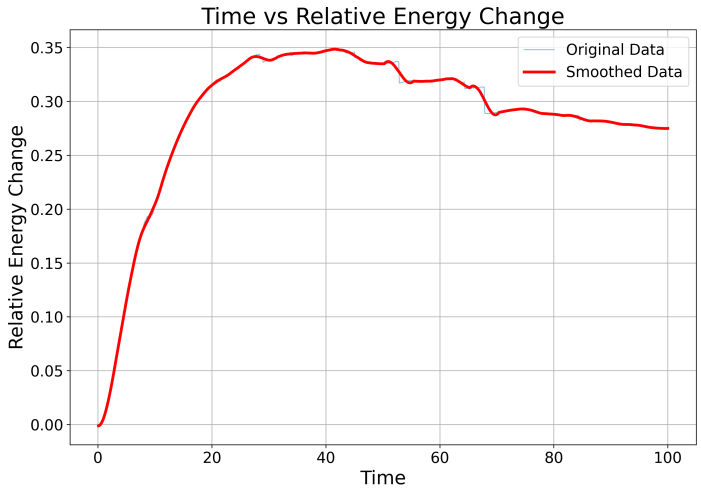

Figure 3: An relative energy error plot for 1000 timesteps (at dt=0.1) of the final simulation.

Discussion

The Final Parsec Problem and Its Implications for Galaxy Evolution

It is thought that a super-massive black hole exists at the center of every galaxy. Observations show a correlation between a galaxy's star mass and the mass of its central SMBH, suggesting that the processes driving galaxy and black hole growth are linked. When galaxies merge, we expect their central SMBHs—drawn together by violent gravitational interactions—to coalesce as well. This process begins with dynamical friction, where the SMBHs lose energy by gravitationally interacting with stars, gas, and other material. Over time, this energy loss allows the SMBHs to migrate toward the center of the merged galaxy.

Despite the effectiveness of dynamical friction in the initial stages, it becomes less efficient as the SMBHs approach one other. This inefficiency arises because the SMBHs, being orders of magnitude more massive than the surrounding stars and gas, require countless interactions over long timescales to lose sufficient energy. Eventually, the two black holes form a bound binary system. At separations on the order of a parsec, the density of nearby material capable of inducing further energy loss diminishes. This leads to the final parsec problem, where the binary stalls at a separation of roughly one parsec.

This issue was first addressed in a 1980 paper by Begelman, Blandford, and Rees [3], who demonstrated that SMBH binaries could remain stalled, unable to merge within the age of the universe, unless additional mechanisms facilitated further energy loss. If SMBHs fail to merge, our

interpretation of galaxy evolution must change. Conversely, if SMBHs do merge, existing theoretical models may be incomplete or require revision.

Several potential solutions to the final parsec problem have been proposed:

- Three-body interactions involving a third SMBH during subsequent galaxy mergers could expel one black hole, driving the remaining two closer together (Iwasawa, Funato, and Makino) [4].
- Interaction with a gas disk might provide an efficient mechanism for energy dissipation, enabling the binary to shrink further (Berczik et al.) [5].
- Star replenishment, where new stars are brought into the vicinity of the SMBHs through dynamical processes, could maintain energy dissipation over time (Milosavljević and Merritt) [6].

Despite these proposed mechanisms, the resolution to the final parsec problem remains an open question in astrophysics.

What our Simulation Tells Us

Although we believe to have some unphysical results due to complications with our code, we do observe that the two super-massive black holes approach one another before they form an orbiting binary. This appears to mimic the final parsec problem discussed above.

It is important to note that in addition to potential errors in our code, we are unable to account for certain phenomena that may have given a different result, such as those discussed in the literature mentioned in the previous section. Ultimately, we believe our simulation describes some of the fundamental aspects of galactic merger dynamics.

Physical Limitations of our Model

While our simulation successfully models the foundational dynamical features of a galaxy merger, there are several limitations that arise from simplifications and assumptions. In this section we identify potential directions we could have gone, given more time and resources.

1. Gravitational Wave Emission from SMBH Binary: Our simulation aims to track the orbital decay of the SMBHs due to dynamical friction, but it does not model the generation of gravitational waves. At small separations, gravitational wave emission becomes the dominant mechanism driving the SMBH merger. Without this, the final coalescence timescale and configuration of the SMBH binary is over-simplified.
2. No Consideration of Gas Physics: Another simplification of our model is the absence of a gas component, which contributes to processes such as angular momentum exchange, dynamical friction, and energy dissipation in the system. Without accounting for gas physics, our model may underestimate the rate at which the SMBH binary spirals inward, particularly during later stages where gas drag might complement gravitational radiation.

Computational Limitations of our Model

1. High Memory Overhead: Our model uses both an interpolated density grid and computed force grid to increase computational efficiency. The density grid dimension is $512 \times 512 \times 265$ and the force grid is 3 times larger since it holds x, y, and z components. This leads to a very high memory overhead on the GPU, which pushes the limits of the hardware.

References

[1] A. Adamo et al. "Star cluster formation in the most extreme environments: insights from the HiPEEC survey". In: 499.3 (Dec. 2020), pp. 3267–3294. DOI: 10.1093/mnras/staa2380. arXiv: 2008.12794 [astro-ph.GA].

[2] Wei Hao et al. "Analysis of Kozai cycles in equal-mass hierarchical triple supermassive black hole mergers in the presence of a stellar cluster". In: 527.4 (Feb. 2024), pp. 10705–10725. DOI: 10.1093/mnras/stad3908. arXiv: 2312.16986 [astro-ph.GA].

[3] M. C. Begelman, R. D. Blandford, and M. J. Rees. "Massive black hole binaries in active galactic nuclei". In: 287.5780 (Sept. 1980), pp. 307–309. DOI: 10.1038/287307a0.

[4] Masaki Iwasawa, Yoko Funato, and Junichiro Makino. "Evolution of Massive Black Hole Triples. I. Equal-Mass Binary-Single Systems". In: 651.2 (Nov. 2006), pp. 1059–1067. DOI: 10.1086/507473. arXiv: astro-ph/0511391 [astro-ph].

[5] Peter Berczik et al. "Efficient Merger of Binary Supermassive Black Holes in Nonaxisymmetric Galaxies". In: 642.1 (May 2006), pp. L21–L24. DOI: 10.1086/504426. arXiv: astro-ph/0601698 [astro-ph].

[6] Miloš Milosavljević and David Merritt. "Long-Term Evolution of Massive Black Hole Binaries". In: 596.2 (Oct. 2003), pp. 860–878. DOI: 10.1086/378086. arXiv: astro-ph/0212459 [astro-ph].