

Galaxy Collisions With CUDA and FFT

Dylan Thomas Adams
Pouya Lajevardi
Chantal LeBlanc
Katayoun Movaseghi

April 15, 2020

Abstract

A simulation of a collision between two galaxies resembling the Milky Way and Andromeda was computed using CUDA and FFT. The simulations performed included one collision at a higher velocity and another collision at half the speed. In both of these simulations, a particle representing the sun was tracked in trajectory throughout the entirety of the collision. The results do not completely compliment the theorized outcome of the Andromeda- Milky Way collision. This is mainly due to the fact that gas density in both galaxies is not accounted for and the black holes in the center of the galaxies do not account for mass density but rather just force.

1 Introduction

Over the last couple decades, theories regarding the formation and evolution of galaxies have developed from slow evolutions and steady chemical element buildups to recurrent collisions and mergers that are separated by periods of relaxation [1]. Due to this development in theories, one of the primary processes in galaxy evolution is now considered to be galaxy collisions [1]. While it is quite possible for a galaxy to undergo multiple collisions in its lifetime, the outcomes of these collisions are hard to predict due to the variability in galaxy mass and kinematic distributions. The amount of energy involved in galaxy collisions is comparable to the binding energy of the galaxies [1]. This dictates the importance of galaxy collisions as evolutionary drivers.

Despite the nature of collisions, it is important to note that the majority of matter involved in galaxy collisions does not actually collide [1]. This implies that while a galaxy may die in the aftermath of a galactic collision, entire solar systems and life-supporting planets may remain intact and unharmed. This is the direct result of the large distances between stars and the relatively small cross sections of said stars [1]. In addition to this, the rest of a galaxy is made up of collisionless dark matter.

Regardless of stellar density increasing towards the center of the galaxy, the low probability for star collision rates remains true.

Separated by 2.5 million lightyears, the Andromeda and Milky Way galaxies are racing towards each other at 402,000 kilometers per hour fueled by gravitational forces [2]. Despite the magnitude at which these galaxies are coming together, the galaxies will not collide for another four billion years with the collision itself lasting a couple billion years. However, the low probability of star collision in galactic collision situations dictates that our solar system will be left relatively unscathed [2]. Modelling a simulation of this collision provides insight into what the future holds for mankind's home galaxy. The collision between the two spiral galaxies- Andromeda and the Milky Way- will result in a massive galaxy that is elliptical in shape [2]. This is where Earth and the rest of the solar system will continue to exist. Within a collision simulation, it is possible to track a particle- the particle representing a sun- in order to understand the effects the collision will have on its trajectory over a period of time.

The galaxy collision simulator with fast Fourier transforms (FFT) simulates the collision of two galaxies similar to the one predicted to happen between the Andromeda and Milky Way galaxies. This was done through using a variety of arrays to model the density, particles, and potential of the respective galaxies. The simulation performed had both galaxies moving towards each other at the time of collision. In both of these simulations, a particle was tracked through its new trajectory post-collision.

1.1 Modelling Galaxies

The Milky Way and Andromeda galaxies are both spiral galaxies with complex structures that make it difficult to model their mass distributions [3]. Modelling the mass distribution in order to correctly simulate the collision of these galaxies plays an important role in modelling the evolution and dynamics of the particles involved throughout the entirety of the collision; especially in tracking the trajectory of a single particle. Correctly modelling the mass distribution of a galaxy is important for estimating the total mass of the galaxy [3]. Having the correct distancing between particles at the time of collision increases accuracy in the collision projections. The reality of low probabilities in star collisions during galactic collisions due to the large distances between consecutive stars is properly represented by the stellar mass density described in equation 1.

$$\Sigma_S(r) = \frac{\Sigma_{S0} l_c}{\sqrt{(r - r_b)^2 + l_c^2}} \quad (1)$$

Equation 1 is derived from the Gaia star number distribution [4]. Defining constants can be found in Table 1.

Bulge radius	$r_b = 2.0 \text{ kpc}$
Maximum surface density	$\Sigma_{S0} = 611 \pm 44 M_{\odot}/pc^2$
Characteristic length	$l_c = 2.5 \text{ kpc}$
Mass within a 25 kpc radius	$M_s = 2.32 \pm 0.24 \times 10^{11} M_{\odot}$

Table 1: The defined constants for Equation (1).

The stellar mass distribution of galaxies derived and modeled from this equation ensures a more reliable simulation of the galaxies as it eliminates uncertainties caused by surface brightness and constant mass-to-light ratio assumptions [4]. Mass-to-light assumption uncertainties are minimized through diminishing the local mass-discrepancy-acceleration relation [5]. The total mass distribution of the galaxy only relies on the stars and gases observed within it; spiral galaxies’ mass distributions are made up of these two factors [5]. While the determination of the stellar mass within a galaxy still has many uncertainties, the use of baryonic mass being accurately found significantly decreases the uncertainty in calculations [5]. This allows for the more accurate modeling of galaxy mass distributions. Galaxy mass distributions are fundamental in understanding the structure and formation of galaxy modelling [6].

2 Methods

2.1 Algorithm Overview

The general algorithm of the galaxy collision simulation involved the creation of a particle array, which was populated based off a rough idea of galaxy distribution data and their velocities, populating a three dimensional density array based on the particle distribution, and performing the fast fourier transform (FFT) on the density array. The FFT array was then manipulated as needed and its inverse was applied to retrieve the potential array, which was converted to accelerations to update the particle array using the leap frog method. The next few sections go into these parts in more detail.

The density array was a $256 \times 256 \times 256$ grid representing space with each grid being one kpc^3 . Masses observed in each grid were added to the array value to increase the density. Similarly, masses which moved away from the grid were to be subtracted from the values of the array to decrease the density at that grid entry.

None of the subroutines had a return value; rather, they took the array as an entry, updated it and passed it on to the following subroutine.

Figure 1 illustrates the potential of an earth-like planet. The planet sits at the center, with a high potential profile, while the potential radially decreases. This figure is the x-y projection of the 3-space surrounding the particle. This is to illustrate the content of the potential array generated by the FFT. As expected, the density needs to drop radially for a point particle.

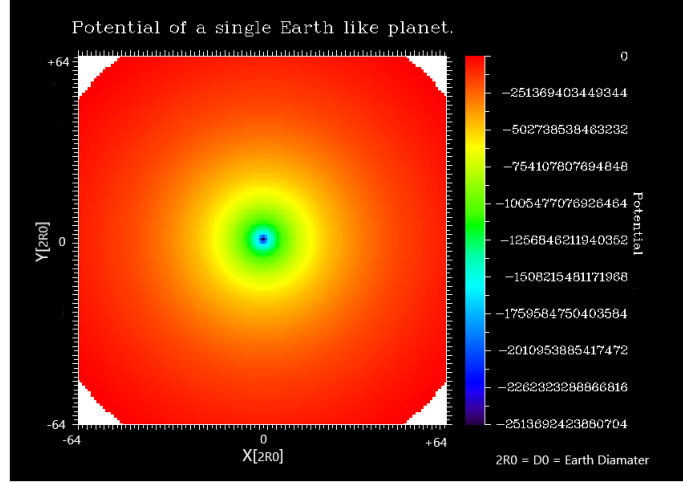


Figure 1: Potential of Earth like planet.

2.1.1 Populating Arrays with Initial Configurations

The particles were generated in radial grids based on their characteristic mass density distributions. Their initial positions were generated based on the mass density as a function of radius, from equation (1). From then particles were proportionally randomly positioned to reflect the initial mass density of the galaxies. The inner density was a randomly generated bulge up to a radius of $2\sqrt{2}$ kpc. The portions outside of the bulge were populated with particles proportional to the radii within that range so that it is less dense than the inner radial bin, and so on, for 11 radial steps, each of distance $\sqrt{2}$ kpc. Each step contained the same amount of mass, so that each step had less density than the one before (due to higher area covered). The mass densities of the two galaxies were considered identical.

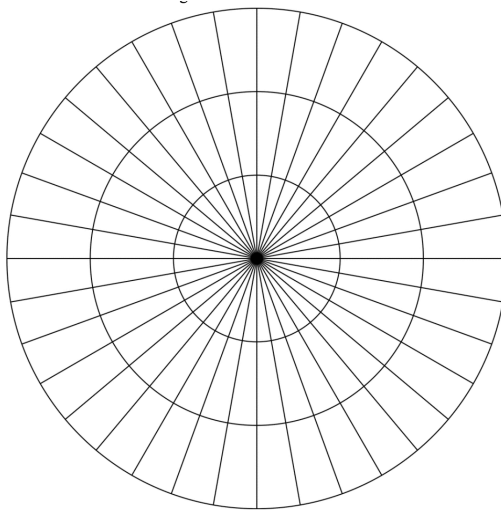


Figure 2: Depiction of radius grid. Each step has length of $\sqrt{2}kpc$. 11 radial steps were used in total, with decreasing particle density with each step.

Unit particles were given masses of $840 M_{\odot}$. The initial velocities of each particle were the sum of their orbital velocities and the velocity of the entire galaxy moving through space (and towards the other galaxy). The orbital velocity component was created using the following equation

$$v_{orb} = \sqrt{\frac{1190.0G}{R}}. \quad (2)$$

The 1190 is the fraction of the central blackholes gravitational acceleration with respect to the mass of the galaxy. The radius R is the distance from each particle to the center of mass. To make sure the velocity remains perpendicular to the radial direction, we make use of the unit circle and concept of sines and cosines to find the correct values of V_x and V_y . Namely $v_x = Y/RV$, $V_y = X/RV$. Later, we add the velocities to all particles to have the galaxies move toward each other.

2.1.2 Using FFT For Potential

One of the biggest problems with running N-Body simulations of galaxies is finding the gravitational force of each particle. If calculated directly it is an $O(N^2)$ operation, with hundreds of millions of particles that need to be updated every step. This time complexity is not good enough for these types of compilations, so Fast Fourier Transforms (FFT's) are used to decrease the calculation to $O(N \log N)$.

FFT operations allow computations to use the gravitational potential to find the gravitational force generated in a system. Gravitational force is related to gravitational potential by

$$F = -\delta\Phi, \quad (3)$$

where F is the force and Φ is the gravitational potential. Potential can be calculated from its relationship to the density of the system using

$$\delta^2\Phi(x, y, z) = 4\rho(x, y, z), \quad (4)$$

where ρ is the density and G is the gravitational constant. This is the gravitational form of Poisson's equation and can be solved easily with the Fourier transformation. When the Fourier transformation is applied to Poisson's equation, it takes the form:

$$-k^2\Phi_f(k) = F(k), \quad (5)$$

where Φ_f is the Fourier transformation of Φ and F is the Fourier transformation the left hand side of the original equation. From that equation it is easy to see that the potential can then be calculated with

$$\Phi = FT^{-1}\left(-\frac{F(k)}{k^2}\right), \quad (6)$$

where FT^{-1} is the inverse Fourier transformation. For the system in this experiment, the density matrix was used as ρ , and the Fourier transformation was done with the CUDA software cuFFT. cuFFT used the GPU to parallelize and complete the FFT of

the system. When starting a CUDA computation the first step was allocating memory in the GPU for the system. When ρ was copied to the GPU it was collapsed into a 1 dimensional array with index $x + yN + zN^2$ in order to keep a 3D view in the 1D array. After that, cuFFT has a built in FFT and FFT^{-1} function that can be called to switch the values. k had three dimensions, one in each direction. Each dimension of k scales the same way, starting from 0 and moving up to $N/2$, and then suddenly shifting to $-N/2$ and increasing again towards 0. This created a symmetric stencil of the system, which was required because the gravitational potential is symmetric.

2.1.3 Parallelization

CPU parallelization was done on many iterations including initial particle array creation, updating the particle array and updating the density array. This was easily taken care of since at each iteration in the loop there is no call or dependency to a prior step which allowed for the calculation to be done quite faster.

3 Results

The mass loss began once the cores collide.

For the first run (Figure 3, the distance between the center of masses of the two galaxies were 90 kpc, with initial velocities $4v$, where $v = 125000$ miles/hour is the actual velocity of the galaxies as calculated by astronomers in an external reference frame. This run also had the galaxies starting at a distance much closer to each other than reality. The simulation was terminated at 40×10^7 years.

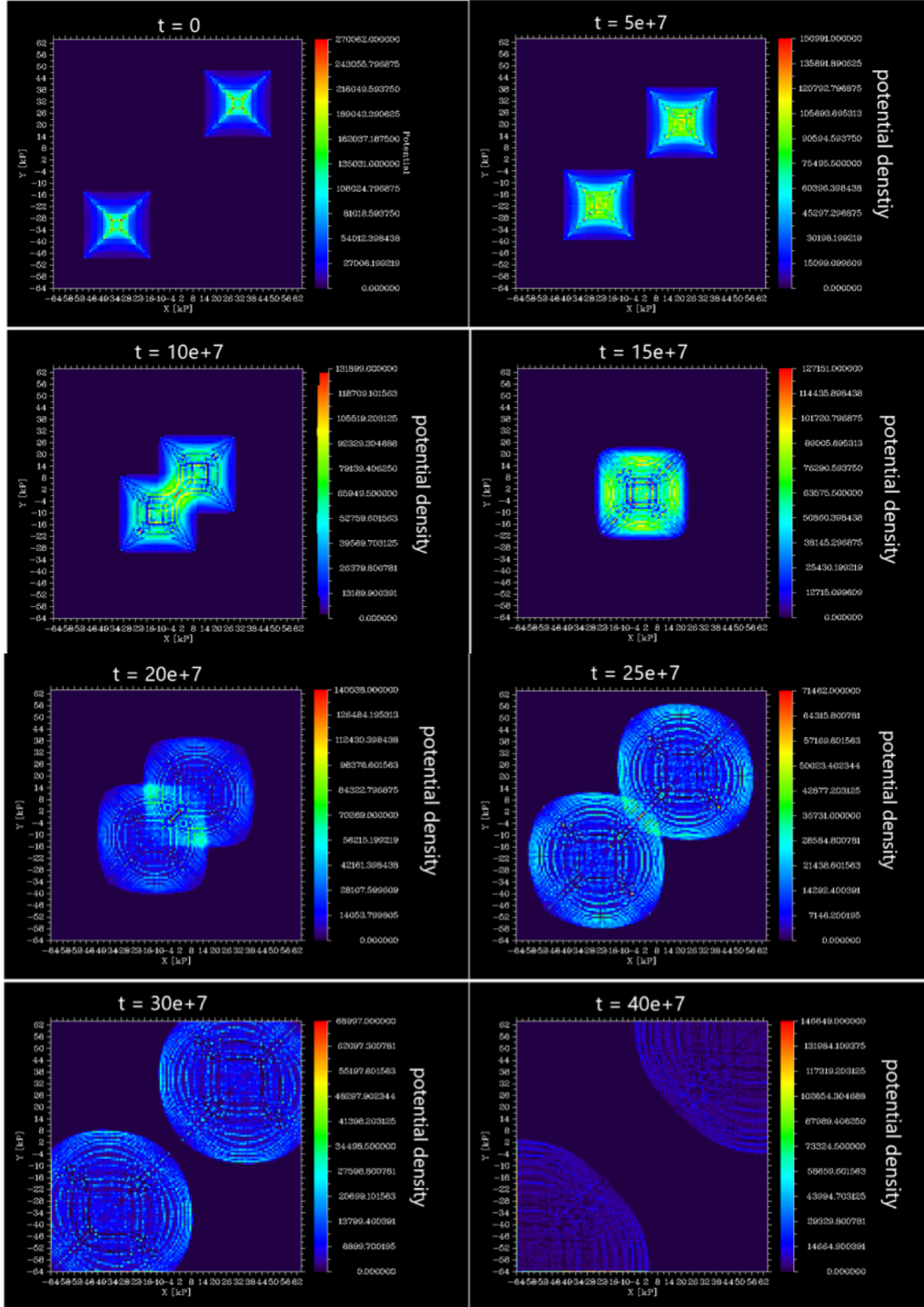


Figure 3: Spatial densities of two particles colliding with given times in years for first simulation.

For the second run, their initial velocities were changed to $2v$. The simulation was terminated at 45×10^7 years.

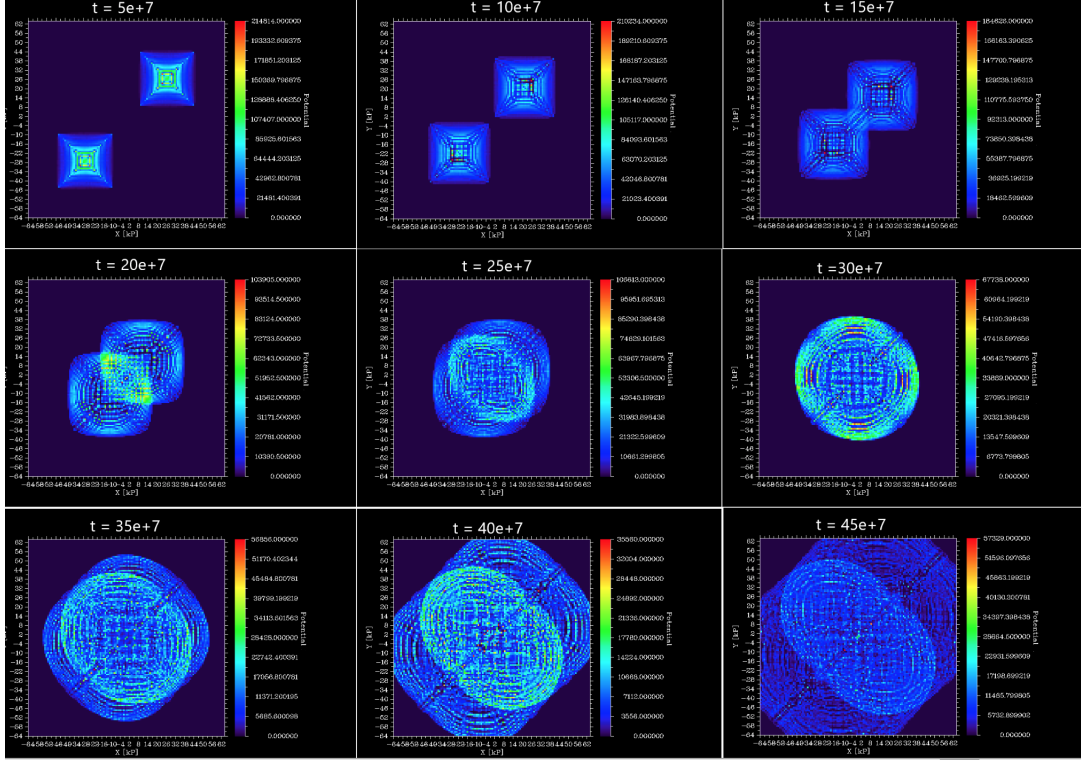


Figure 4: Spatial densities of two particles colliding with given times in years for second simulation.

The position of a particle with a mass of $840 M_{\odot}$ was tracked throughout the simulations. Initially, it had a position of 8.32 ± 0.14 kpc [14] from the Galactic Center. Its evolution for the first simulation is shown in Figure 5 and for the slower simulation Figure 6.

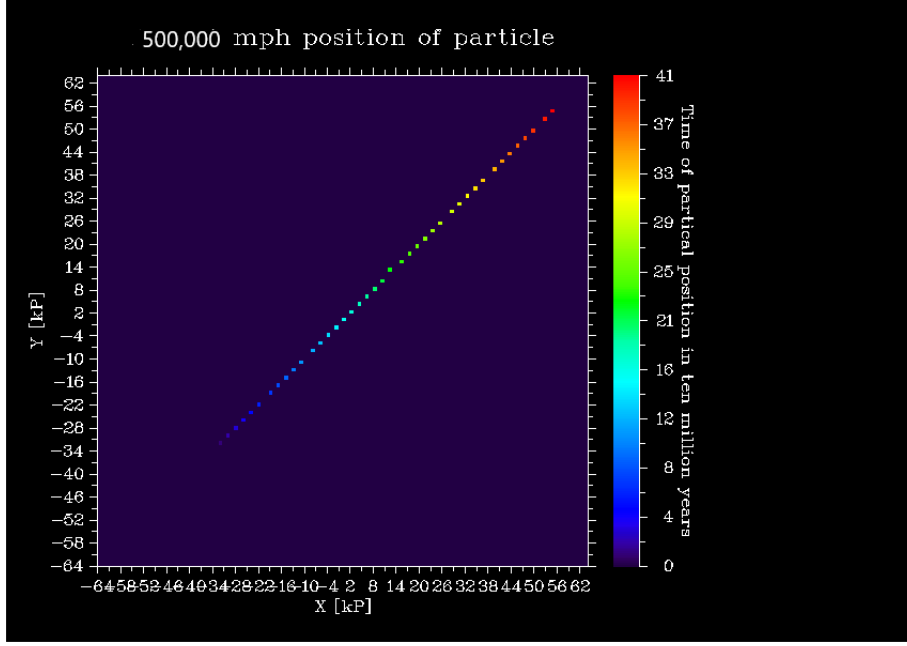


Figure 5: Evolution of single particle in first simulation, with initial velocity $4v$.

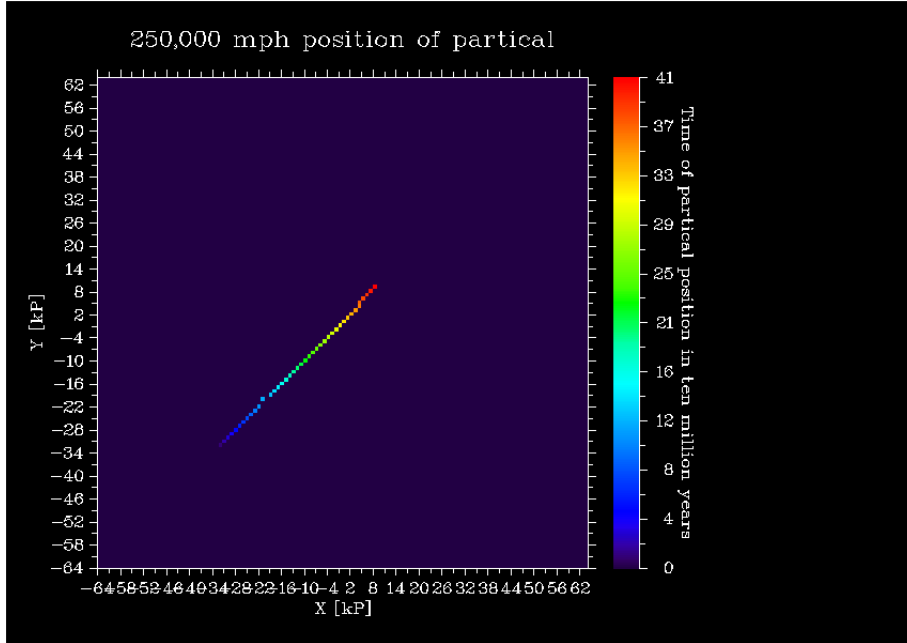


Figure 6: Evolution of single particle in second simulation with initial velocity $2v$.

4 Discussion

The majority of duration the simulation took was to calculate and update the configuration of particles in the particle array. This step took the value of the gravitational

potential, found its differential and evolved the particles and updated the particle array with new information. All of this took place parallel on CPU and as discussed since the steps are not dependent on their prior step, this parallelization was done without the need for special consideration of dependency on prior steps. The FFT which was done parallel on GPU using CUDA, was one of the fastest major tasks in each time step of the program. As seen in the evolution of the first simulation (Figure 2), the higher speed and closer initial proximity resulted in the galaxies colliding and continuing to pass right through each other. It may also be noted that the density distribution of particles in each galaxy slowly dissipated throughout the entire interaction. The hypothesized theory is that when Andromeda and the Milky Way collide in 4 billion years, they will continue to bounce apart and come together over the time period of another couple billion years before forming a new massive galaxy. This was not represented in this set of results. However, these results did show the lack of star-to-star collisions that was initially predicted. As the galaxies passed through one another, the particles also remained intact as they passed through each other. As seen in the evolution in the second simulation (Figure 4, the slightly lower speed in this set resulted in the galaxies colliding without passing straight through each other. These results did not show the galaxies bouncing apart and together but still correctly represent the lack of star-to-star collisions. The density distribution in this set of results also dissipated over time.

This lack of repeated collisions did not occur because we did not consider gas pressure (because they do not collide), the black holes and the black holes in the model only contributed force, with no realistic excess mass to represent them.

4.1 Future Work

The galaxy densities, as illustrated in Figures 3 and 4 show square shaped galaxies, when in theory the shapes of Andromeda and Milky Way type galaxies are spiral. This could be made more realistic with linear interpolation to smoothen out the boundaries of the particle distributions.

More realistic results may be obtained through taking into account the gas mass distribution along with stellar mass distribution as the current model only takes into account the stellar masses. It is known that the total mass distribution of the galaxy only relies on the stars and gases observed within it [5]. By not accounting for the gas distribution of each galaxy, the final results do not account from a large portion of the total mass. This simulation does not account for the presence of dark matter in each of the galaxies. However, dark matter does not change the total mass of the galaxies nor does in effect the way in which the galaxies interact as they pass through [5]. Due to this, we do not believe that the presence of dark matter in the simulation would significantly change the results. It is also important to note that the galaxies that are represented in these simulations are not exact representations of the Milky Way and Andromeda galaxies. The size, velocities, and different mass distributions all vary with reality. This affects the interpretation of the data as we are comparing the results to what is expected out of the Andromeda and Milky Way collision. Larger time periods for analysis would provide a clearer interpretation of

the results and of the lifecycle of the galactic collision. By combining all of the above stated improvements, it may be possible to simulate the collision all the way through to the formation of a new massive galaxy.

5 Conclusions

The first simulation used velocities four times greater than reality and much smaller initial distances between the galaxies. The results had the galaxies pass through each other as their densities dissipated over time. The second simulation used velocities twice as large as reality and, again, much smaller initial distances between the galaxies. These results had the galaxies collide without passing straight through each other. The simulated results would more accurately represent the collision between the Milky Way and Andromeda galaxies if gas mass distribution was accounted for, and proper initial velocities, sizes and distances were used.

References

- [1] Struck, C. (1999, November). Galaxy Collisions. *Physics Reports*, 1-137. doi: [https://doi-org.myaccess.library.utoronto.ca/10.1016/S0370-1573\(99\)00030-7](https://doi-org.myaccess.library.utoronto.ca/10.1016/S0370-1573(99)00030-7)
- [2] Drake, N. (2014, March 24). Milky Way Has 4 Billion Years to Live — But Our Sun Will Survive. In *National Geographic*. Retrieved from <https://www.nationalgeographic.com/science/phenomena/2014/03/24/scientists-predict-our-galaxys-death/>
- [3] Sparke, L., Gallagher, J. (2007). *Galaxies in the Universe* (2nd ed.). Cambridge, NY: Cambridge University Press.
- [4] Li, E. Modelling mass distribution of the Milky Way galaxy using Gaia’s billion-star map. Faculty of Engineering and Information Sciences, University of Wollongong. Retrieved from <https://arxiv.org/pdf/1612.07781.pdf>
- [5] McGaugh, S. (2005). The Baryonic Tully-Fisher Relation of Galaxies with Extended Rotation Curves and the Stellar Mass of Rotating Galaxies. *The Astrophysical Journal*, 632(2). Retrieved from https://iopscience.iop.org/article/10.1086/432968?fbclid=IwAR36DsIRCMIA2KDMgFkIl9imh69_bkIn-0zbsCw-qDRRHoZkIby28ynr22M
- [6] Courteau, S. (2014, January 14). Galaxy Masses. *Reviews of Modern Physics*, 86(47). Retrieved from <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.86.47>
- [7] Techie Delight. (n.d.). Retrieved April 12, 2020, from <https://www.techiedelight.com/>

- [8] Bhattacharjee, P., Chaudhury, S., Kundu, S., Majumdar, S. (2013). Deriving the velocity distribution of Galactic dark matter particles from the rotation curve data. *Physical Review D*, 87(8). doi: 10.1103/physrevd.87.083525
- [9] Brigham, E. O. (1988). The fast Fourier transform and its applications.
- [10] C Pointers. (n.d.). Retrieved April 12, 2020, from <https://www.programiz.com/c-programming/c-pointers>
- [11] Cuda C Programming Guide. (2018). [PG-02829-001_v9.2 — April 2018]. Retrieved from https://docs.nvidia.com/cuda/archive/9.1/pdf/CUDA_C_Programming_Guide.pdf
- [12] Cufft Library User'S Guide. (2019). [DU-06707-001_v10.1 — August 2019]. Retrieved from https://docs.nvidia.com/cuda/archive/10.1/pdf/CUFFT_Library.pdf
- [13] Difference between pointer to an array and array of pointers. (2019, June 13). Retrieved from <https://www.geeksforgeeks.org/difference-between-pointer-to-an-array-and-array-of-pointers/>
- [14] Sanders, J., Kandrot, E., Dongarra, J. J. (2015). Cuda by example: an introduction to general-purpose Gpu programming.
- [15] Springel, V. (n.d.). Retrieved from <https://obswww.unige.ch/lastro/conferences/sf2013/pdf/lecture2.pdf>
- [16] Storti, D., Yurkowski, M. (2016). Cuda for engineers: an introduction to high-performance parallel computing.
- [17] Tremaine, S., Binney, J. (2008). *Galactic Dynamics: (Second Edition)*. Retrieved from <https://press.princeton.edu/books/paperback/9780691130279/galactic-dynamics>