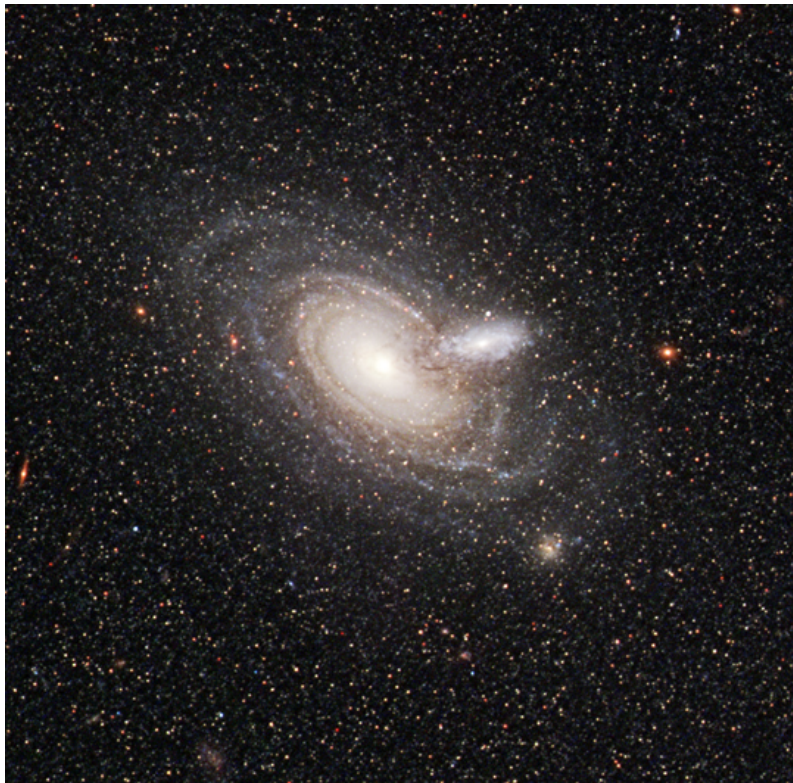


# Simulating the Merging of Two Disk Galaxies

**Project 1:** Simulating colliding galaxies  
Modelling and Simulation

**Group 21:**  
Jordan Barkai (s3972097)

October 18, 2020



Credit: [\[11\]](#)

**Project Description:** “Simulate the collision of two disk-shaped galaxies in 3D. Experiment with various initial conditions, resulting in realistically shaped galaxies. Galaxies typically comprise  $10^{11}$  stars. It is not feasible to do a simulation with that number of stars. Determine experimentally what is feasible. An important simplification/approximation is the use of the Barnes-Hut method. The essence of this method is that stars are grouped when they are close together. In this way the complexity of the algorithm is reduced from  $N^2$  to  $N \log(N)$  [\[5\]](#).”



**Github Repo**

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Method</b>	<b>3</b>
2.1	Initial Positions and Velocities - Rejection Method . . . . .	4
2.2	Modelling Approach - Barnes Hut Algorithm . . . . .	7
2.3	Integrating the Particle Orbits . . . . .	9
2.3.1	Leap Frog Algorithm . . . . .	9
2.4	Tools - Python . . . . .	9
<b>3</b>	<b>Results and Discussion</b>	<b>10</b>
3.1	Comparing collision angle . . . . .	10
3.2	Comparing radii ratios . . . . .	12
<b>4</b>	<b>Conclusion</b>	<b>12</b>
<b>5</b>	<b>References</b>	<b>14</b>

# 1 Introduction

While the familiar spiral structure of galaxies like our own, the Milky Way, are better known outside the field of Astronomy, galaxies come in a variety of shapes and sizes. The history behind the formation of some galaxies can be explained by the violent environments which cause gravitational interactions between them. For example, the collision of two spiral galaxies, like the Milky Way, form elliptical galaxies which lack such structure. These collisions which result in a single gravitationally bound stellar system are known as galaxy mergers and are studied in great depth to help understand the formation and evolution of galaxies in our Universe.

However, these mergers occur on timescales far beyond a human lifetime and therefore require simulations to better understand them. Two of the most widely used approaches for simulating galaxy dynamics is smoothed particle hydrodynamics (SPH) and N-body simulations. The first, SPH simulations, "smoothes" the particles in order to simulate the interactions with fluid dynamics. The latter, N-body simulations, traces the motion of many (N) massive particles and their gravitational interactions. While both of these methods are widely used in modelling galaxy mergers, they are both very computationally expensive and a lot of work has been done to try and optimize them. For the scale of this project we will be using N-body simulations to model the collision of two disk galaxies.

Although galaxies typically comprise of  $N_* = 10^{11}$  stars, not even the largest computers can simulate more than about  $10^{10}$  particles efficiently [8]. There are therefore two different approaches to choosing how many particles,  $N$ , to numerically integrate the equations of motion. Collisional N-body simulations choose  $N = N_*$  particles while collisionless N-body simulations integrate the motion of  $N \ll N_*$  particles. The choice between the two depends on the relaxation time, the time taken for equilibrium to be reached again after a perturbation. For collisional N-body simulations the integration time is larger than the relaxation time and for this reason, it is the approach that will be used for this project.

In this project we aim to investigate how the different physical properties of two disk-shaped galaxies merging affect the morphology of the final galaxy produced in their interaction. We plan to do this by modelling the two disk-shaped galaxies with collisional N-body simulations in three dimensions with various different initial conditions and analysing the different resulting spatial distributions. The parameters we investigate in this project are the collision angle of the two galaxies and the ratio of their radii.

Simulating the merging of two disk galaxies, we expect to form an elliptical galaxy. However, we also expect that the physical properties and morphology of this final galaxy will change depending on the chosen initial conditions. Based on previous simulations [4] we expect the collision angle and radii ratio to have very little effect since it is the mass ratio that has been found to have a large effect and this remains constant throughout the simulations.

The details of the modelling methodology, physics and optimization of the model are found in section 2, the results and discussion of the simulation can be found in section 3, and finally what has been concluded from these results can be found in section 4.

## 2 Method

In this section we will explain how an N-body simulation was used to model the collision of two disk galaxies. An explanation of how the initial conditions of the system were setup is followed by an explanation of the modelling approach and finally the tools used to execute this. Four simulations were run, each with different initial conditions, a list of which can be found in Table 1.

Table 1: Summary of the simulations that were run and the various physical parameters used. Note that  $M_A$  and  $M_B$  are the masses of galaxies A and B respectively, while  $R_A$  and  $R_B$  are their radii and  $\Theta$  is the collision angle between them. Also note that the scale length of both galaxies is 8kpc in all the simulations

Simulation	$M_A$ [ $M_\odot$ ]	$M_B$ [ $M_\odot$ ]	$R_A$ [kpc]	$R_B$ [kpc]	$\Theta$ [radians]
1	$10^6$	$10^6$	15	15	0
2	$10^6$	$10^6$	15	15	$\pi/4$
3	$10^6$	$10^6$	15	15	$\pi/2$
4	$10^6$	$10^6$	15	5	$\pi/2$

## 2.1 Initial Positions and Velocities - Rejection Method

To setup the simulation, first the initial positions and velocities of each particle of the two galaxy systems are needed. When modelling a disk galaxy, the three components of it need to be considered, namely the bulge, the dark matter halo and the disk. However, for the simplicity of this project only the disk component has been modelled with the insertion of a massive central particle to represent the bulge. To begin, 1000 particles were setup in initial positions with initial velocities to create the two spiral-disk galaxies initially placed in equilibrium. The initial conditions shown in Table 1 were chosen to create realistic properties comparable to observed disk galaxies.

The process to find the initial position and velocity follows an adaptation of [2], using the thickened disk Miyamoto-Nagai potential [13] instead of the Plummer potential, defined as:

$$\Phi(r, z) = -\frac{GM}{\sqrt{r^2 + (\sqrt{z^2 + b^2} + a)^2}} \quad (1)$$

where  $b$  is the scale height and  $a$  is the scale length of the system. For simplicity, we set  $b=z=0$ , which makes a razor thin disk with what is known as the Kuzmin potential:

$$\Phi(r) = -\frac{GM}{\sqrt{r^2 + a^2}} \quad (2)$$

For the scale of this project we will use this potential to model the disk galaxies but in fact a bulge and a dark matter halo component are necessary to represent the total mass distribution of a galaxy. We begin by assuming that the 1000 stars in the galaxy are of equal mass,  $m=M/N$ , and define the mass enclosed in a disk of radius  $r$  as follows:

$$M(r) = \frac{r^2}{G} \frac{d\Phi}{dr} \quad (3)$$

$$= \frac{Mr^3}{(r^2 + a^2)^{3/2}} \quad (4)$$

$$(5)$$

Therefore by assigning a random number from a uniform distribution,  $M(r) = x_1$ , to be the enclosed mass between 0 and the total mass of the galaxy, the radius,  $r$ , can be found:

$$r = \frac{(x_1/M)^{1/3}a}{\sqrt{1 - (x_1/M)^{2/3}}} \quad (6)$$

$$(7)$$

Using the rejection method, only particles that fell within the radius of the disk were taken. The radius was then used to find the cartesian co-ordinates (where  $z=0$  in a disk) by choosing a random angle for  $\theta$  between 0 and  $2\pi$  as follows:

$$z = 0 \quad (8)$$

$$x = \sqrt{r^2 - z^2} \cos \theta = r \cos \theta \quad (9)$$

$$y = \sqrt{r^2 - z^2} \sin \theta = r \sin \theta \quad (10)$$

$$(11)$$

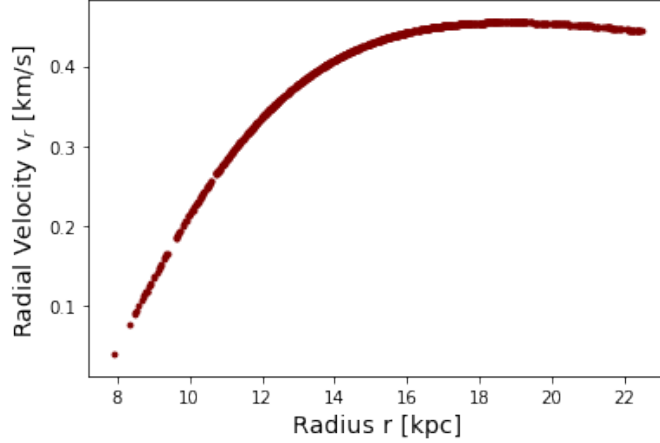


Figure 1: The initial radial velocity as a function of radius for each particle of galaxy A with the initial conditions of simulation 1 in Table 1.

Using the radius and the enclosed mass, the angular velocity was found:

$$V_{\theta} = \sqrt{GM_r/r^3} \quad (12)$$

$$(13)$$

and used to calculate the Cartesian components as follows:

$$v_z = 0 \quad (14)$$

$$v_x = -rV_{\theta} \sin \theta \quad (15)$$

$$v_y = +rV_{\theta} \cos \theta \quad (16)$$

$$(17)$$

To ensure the resulting velocities were correct, the magnitude of the particles' velocities was plotted as a function of the radius and can be seen in Figure 1. This procedure was performed for all 1000 particles in each of the galaxy systems, galaxy A and B. The calculated initial positions and velocities of the particles in each system were then shifted on either side of the x-axis by 1/2 times their radii to create a new initial setup of two disk galaxies side by side for simulation 1, see Figure 2.

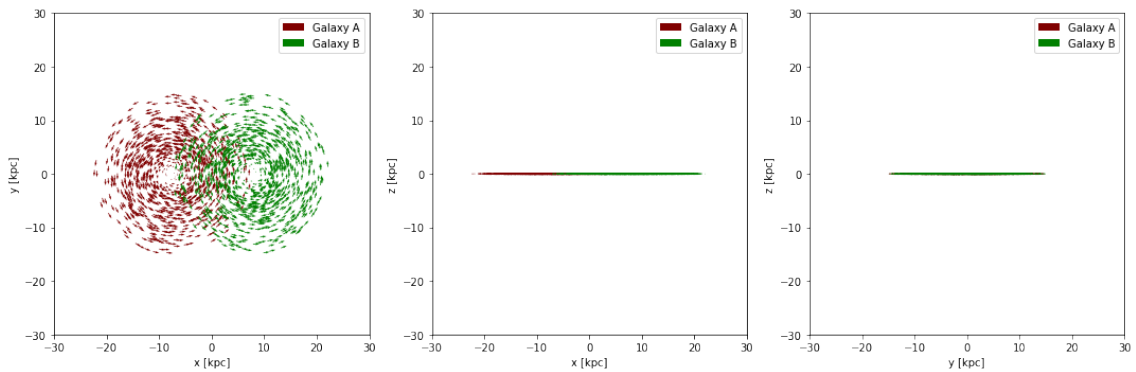


Figure 2: Initial set-up of the two disk galaxies prior to their collision in simulation 1 where the collision angle is 0 radians. The x-y, x-z and y-z planes are plotted with each point showing the initial positions with an arrow indicating the direction of the initial velocities and the length indicating the magnitude of the initial velocities. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles.

For simulation 2, galaxy A was shifted half its radius down in the z-direction and rotated  $\pi/4$  radians about the y-axis, see Figure 3.

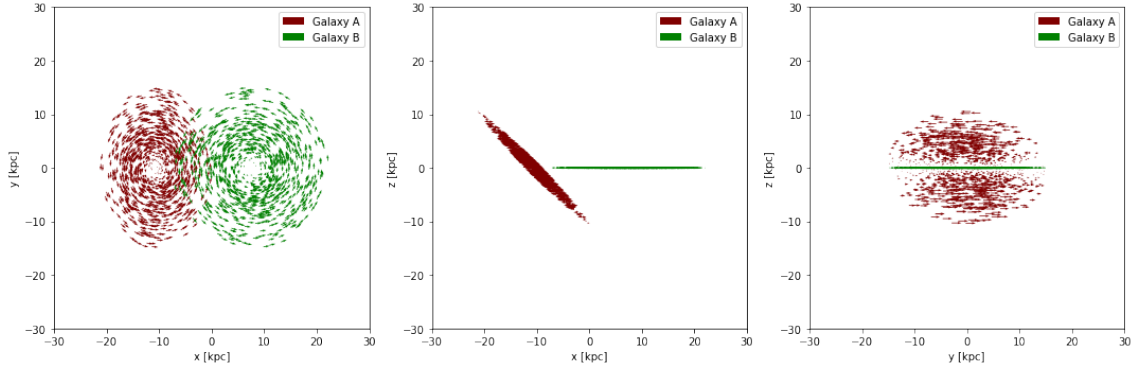


Figure 3: Initial set-up of the two disk galaxies prior to their collision in simulation 2 where the collision angle is  $\pi/4$  radians. The x-y, x-z and y-z planes are plotted with each point showing the initial positions with an arrow indicating the direction of the initial velocities and the length indicating the magnitude of the initial velocities. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles.

For simulation 3, galaxy B was then also shifted half its radius but up in the z-direction and rotated  $-\pi/4$  radians about the y-axis, see Figure 4.

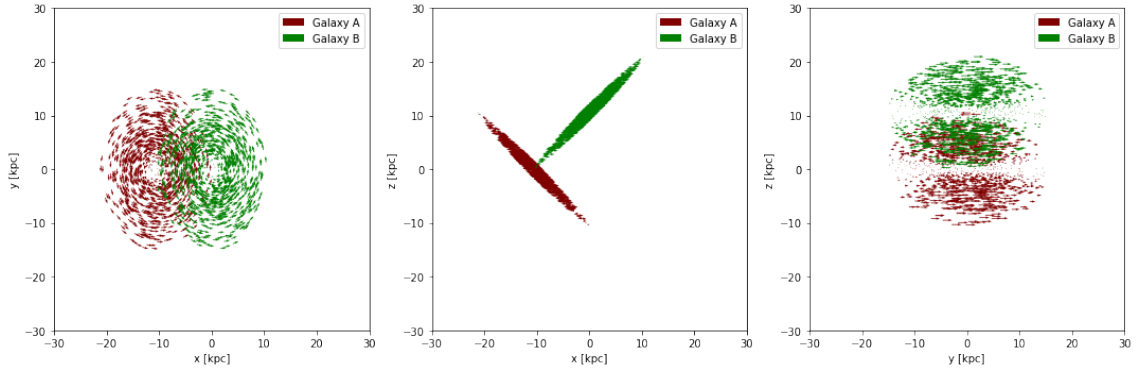


Figure 4: Initial set-up of the two disk galaxies prior to their collision in simulation 3 where the collision angle is  $\pi/2$  radians. The x-y, x-z and y-z planes are plotted with each point showing the initial positions with an arrow indicating the direction of the initial velocities and the length indicating the magnitude of the initial velocities. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles.

The final set of initial conditions, simulation 4, was set up by then reducing the radius of galaxy B to 5kpc, see Figure 5.

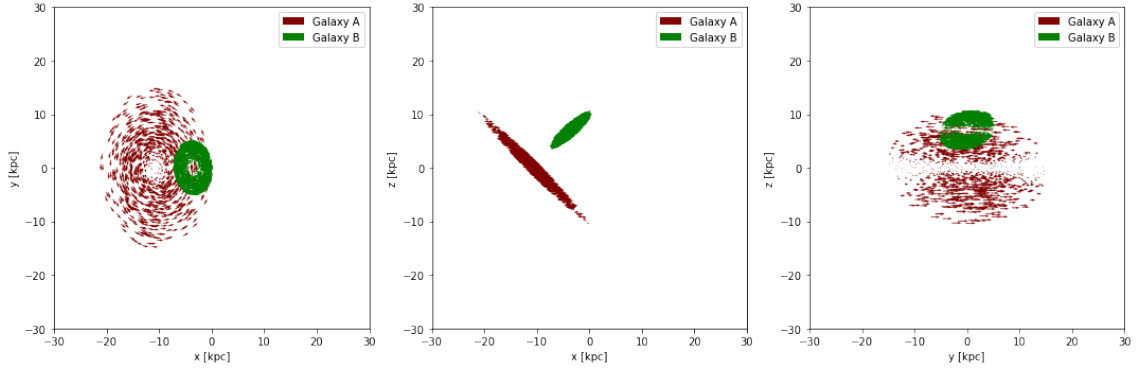


Figure 5: Initial set-up of the two disk galaxies prior to their collision in simulation 4 where the collision angle is  $\pi/2$  radians and the radius ratio is 1:3. The x-y, x-z and y-z planes are plotted with each point showing the initial positions with an arrow indicating the direction of the initial velocities and the length indicating the magnitude of the initial velocities. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles.

Additionally, a very massive particle is added to the centre of each system to represent the bulge and black hole of the galaxies, equal to the size of the galaxy to stabilise the system. This results in 2002 particles in each simulation.

## 2.2 Modelling Approach - Barnes Hut Algorithm

To model this problem we take the N-body approach, dividing the system into discrete N particles of equal mass. These particles could be stars, fluid elements, or even grid cells. In the merging of two disk galaxies the dominating force at these physical scales is gravity and therefore it is responsible for the physical outcome of the simulation. Therefore to model the change in position and velocity of these particles over time we need to calculate how they are gravitationally influenced by the other particles. For this we can treat them as point masses to calculate their gravitational fields, which requires a numerical solution for large N.

Due to the limited resources available for this project we have chosen to use the Barnes-Hut algorithm [5] to optimize the numerical approach, which has a computational complexity of  $O(N \log N)$  as opposed to a direct-sum algorithm of  $O(N^2)$ . The algorithm begins by dividing the simulated volume into cubic cells using an octree (a tree in which each internal node has 8 children). The root of the tree contains the entire space that needs to be simulated. At each level this space is recursively subdivided into 8 regions (represented by 8 nodes), until each node contains only 1 or even 0 bodies. Each internal stores the center-of-mass of its children. In order to compute the force acting on a specific body, the tree is traversed starting from the root; if the center-of-mass stored in a node is far from the required body, then it is treated as a single particle and its interaction with said body is added in the total being accumulated. Otherwise, if the center-of-mass is close, then the process is repeated recursively for each of its children. Note that the vague terms "far" and "close" depend on a fixed accuracy parameter. This approach is shown visually with a schematic example in Figure 6.



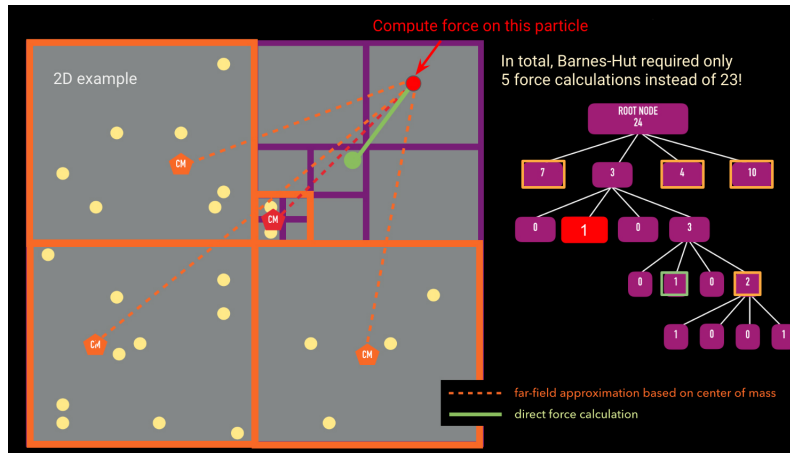


Figure 6: A schematic diagram explaining the process of the Barnes-Hut method [6]. The left shows the domain divided into quadrants, with the North East quadrant divided even further. In this image it shows the computation of the gravitational force on the red particle due to the other particles. On the right, the oct tree with the corresponding nodes are shown, demonstrating that with this Barnes-hut method only 5 force calculations are needed to compute the force on this particle rather than the 23 that would be needed for a regular N-body simulation.

Using the established initial setup of the system, the Barnes Hut method was used to calculate the gravitational acceleration,  $g$ , of each particle. This is done by looping through the octree with the following procedure for each node and particle whose acceleration you seek to calculate:

1. If the node has no children and is not equal to the particle whose acceleration you are trying to find, just calculate the gravitational acceleration the particle experiences due to the mass defined by this node:

$$g_x = \frac{GM_{node} dx}{r^3} \quad (18)$$

where  $M_{node}$  is the mass of the leaf node and  $dx$  is the leaf's centre of mass subtracted from the particle's centre of mass in the x-direction with  $r = \sqrt{dx^2 + dy^2 + dz^2 + \epsilon^2}$  being the magnitude of the vectors. The same goes for the y and z components. Note here that  $\epsilon$  is the softening parameter which prevents the accelerations from exploding to infinity when particles are too close together (due to the inverse square-law) which is caused by the assumptions that these are point masses. The softening parameter is chosen depending on the number of particles and the mass distribution describing them and was therefore taken by modifying [3] as follows:

$$\epsilon_{opt} = 10 \times 0.98N^{-0.26} \quad (19)$$

2. If it does have children the theta criterion is applied. This is done by calculating the ratio of the side length,  $s$ , of the domain and the distance from the particle to the centre of mass of the domain,  $d$ , and comparing it to the open angle,  $\alpha$ . The choice of this is made by the associated force accuracy and for this project was chosen to be  $\alpha = 0.5$ , following [15].
  - (a) If  $s/d < \alpha$  we continue as in step 1 and add the acceleration to the net acceleration that the particle experiences.
  - (b) If  $s/d > \alpha$  the algorithm continues recursively through the children of the node.

The Barnes–Hut method is advantageous over the other potential solving codes due the accuracy of local interactions as well as its lack of geometrical assumptions and its flexibility to be used in a wide range of applications. Compared to other hierarchical tree codes, this method is also advantageous for its simplicity as well as its potential for rigorous error analysis.



## 2.3 Integrating the Particle Orbits

Once the particle's accelerations have been calculated, an integrator will be needed to update their positions and velocities at each time step. A few factors need to be taken into account when choosing the kind of integrator for this, as well as the size of the integration time step. In attempt to maximize the number of particles we can simulate we need to optimize the memory available for a galaxy simulation which would require an integrator that predicts future positions and velocities of particles based of their current position and gravitational field rather than from the past. Due to the time constraint of the project it is also important to consider the integration time, however, for a galaxy merger this should only be around a few crossing times. Since we are working in the scale of galaxies and not clusters the gravitational potential will not place too many demands on the integrator and we do not need to worry as much about the smoothness of the potential [7]. Finally the computational expense of the integrator must be considered.

### 2.3.1 Leap Frog Algorithm

The Leapfrog Algorithm [12] has been chosen due to its conservation of energy (with the accuracy depending on the chosen time step) over many time steps. It is a symplectic first-order integrator, meaning that it is time-reversible and is use to solve the differential equations of motion for the particles. The time step was taken to be  $dt = 2Gyr$  due to computational constraints but it is recommended for future work to use  $dt = 0.01Myr$ , following the tests of [6]. The number of time steps was taken to be 50 which makes the integration time 100 Gyr. Unlike most integrators, the leap frog integrator's position and velocities steps are asynchronous which is advantageous in second-order systems. The velocities are in fact taken at half time steps as you can see in Figure 7.

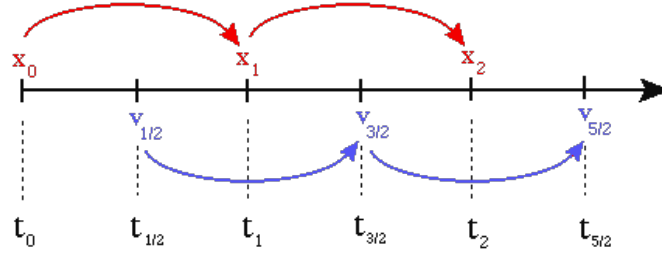


Figure 7: A schematic diagram demonstrating the asynchronous time steps of the velocities and positions calculated when using the Leap Frog Algorithm [1].

The algorithm calculates the velocity at half a time step ahead of the previous position time step,  $i$ , known as the kick step:

$$v_{i+1/2} = v_{i-1/2} + a_i dt \quad (20)$$

where  $dt$  is the time step size and  $a_i$  is the acceleration at time  $t = i$  taken from the Barnes-hut method. The leap frog then uses these half-step velocities to calculate the position in the next full time step, known as the drift step:

$$x_{i+1} = x_i + v_{i+1/2} dt \quad (21)$$

Since the initial positions and velocities are created in-sync, another kind of integrator is needed to act as a “self-starting” scheme to first calculate the velocities at half a time-step, before the leap frog can be used. Due to the simplification of this project, it is assumed that the algorithm is self-starting, i.e. that  $v(t=0)=v(t=1/2)$ . All the simulations were integrated using this method for a time consistent with the time scale of disk galaxy mergers in the literature, which is around 100Gyr.

## 2.4 Tools - Python

Python 3.8 was used for the implementation of this simulation. Python has become one of most widely used programming languages in the field of astronomy as well as many other fields. Therefore

it has been chosen primarily for the re-usability of the code for different applications in the future and its abundant existing resources for modelling astrophysical systems. In addition, python's celebrated user-friendliness and flexibility still allows for speed due to it's ability to wrap low-level languages such as C, which most of the widely used packages do.

One of the most useful packages for any astronomical work is **Astropy** [9]. Aimed at creating a central location for all astronomy tools in python regularly used for research in the field. Broadly, **Astropy** capabilities include astronomical units and conversions, co-ordinate systems and transformations, absolute date-times, dealing with common astronomical file formats such as fits files, and cosmological calculations. Majority of these utilities were used for this project. Besides complimenting the more general scientific packages such as **SciPy** [16] and **NumPy** [10] it also has many affiliated packages for more specialised fields, such as **Gala** [14].

### 3 Results and Discussion

The collision of two disk galaxies was simulated for three different collision angles and two different radii ratios. The spatial evolution of these simulations are plotted in this section, however, for a more detailed understanding of their evolution and for the sake of better comparison please watch the attached videos of each of the simulations.

#### 3.1 Comparing collision angle

In the first simulation, see Figure 8, two identical galaxies were placed next to each other and evolved for 100Gyr. In the second simulation, galaxy A was rotated about the y-axis, making a collision angle of  $\pi/4$  radians, and the again integrated for 100Gyr, as you can see in Figure 9. For the final collision, integrated for the same period of time, galaxy B was also rotated about the y-axis and shifted up the z-axis such that the galaxies were perpendicular, having a collision angle of  $\pi/2$  radians, see Figure 10. From the videos of each simulation it is clear that the merger is complete and the system is stable by the end of the 100Gyr as desired and that the original disk structure of the two galaxies is stripped to form an entirely new structure.

We begin by taking a look at the first “head-on” simulation in Figure 8. As you can see from the two panels showing the different planes, this merger results in a larger disk galaxy rather than a spherical one, remaining only in the x-y plane. This could be due to the over-simplification of the initial positions and velocities having no z components. After about 50Gyr, see video **sim\_0.mp4**, the merger appears to have reached an equilibrium with the particles orbiting a new central nuclei.

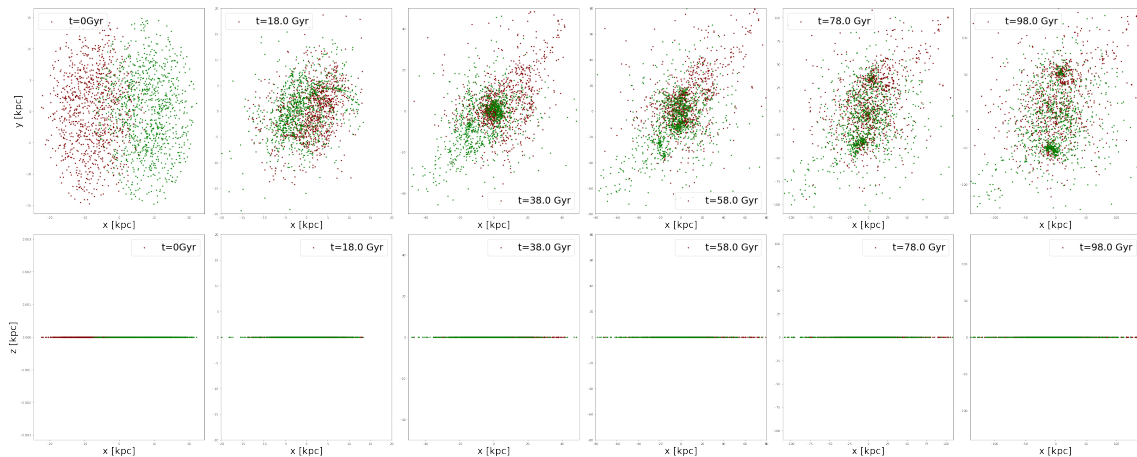


Figure 8: The spatial evolution of the merging between two galaxies of equal mass and a collision angle of 0 radians over 100Gyr. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles. The top row shows the x-y plane, while the bottom row shows the x-z plane, each incrementing in time from left to right with the time.

An investigation Figure 9 and video `sim_pi_4.mp4` shows an already very different result with the slight increase in collision angle to  $\pi/4$  radians. In this simulation, unlike the “head-on” scenario, the spatial evolution does not remain in the x-y plane and shows some shape in the x-z plane one the galaxies start interacting. Although it is more clear in the video, this simulation again results in a single central nuclei, with the system stabilising at around 60Gyr.

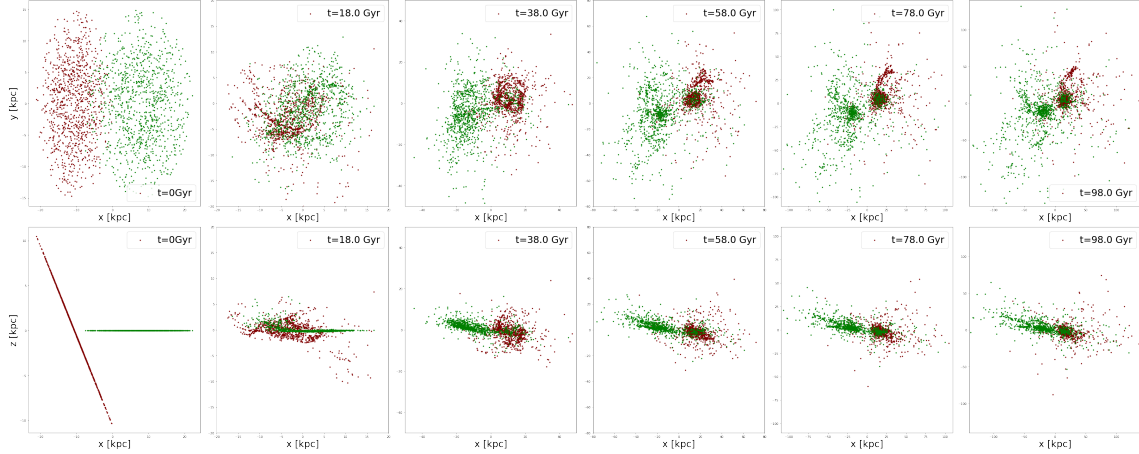


Figure 9: The spatial evolution of the merging between two galaxies of equal mass and a collision angle of  $\pi/4$  radians over 100Gyr. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles. The top row shows the x-y plane, while the bottom row shows the x-z plane, each incrementing in time from left to right with with the time.

The final simulation with the largest collision angle of  $\pi/2$ , with the two galaxies perpendicular to each other can be seen in Figure 10 and the `sim_pi_2.mp4` video. This increase in collision angle finally shows a much more spherical structure, with the x-z plane appearing to be spatially distributed similarly to the x-y plane. This increased collision angle also shows a much more destructive interaction, with the galaxies twisting and tearing in the duration of the merger. Although the resulting galaxy shows a single nuclei again, it is not at the centre of the system but instead is offset to the right and this time taking about 70Gyr to stabilise.

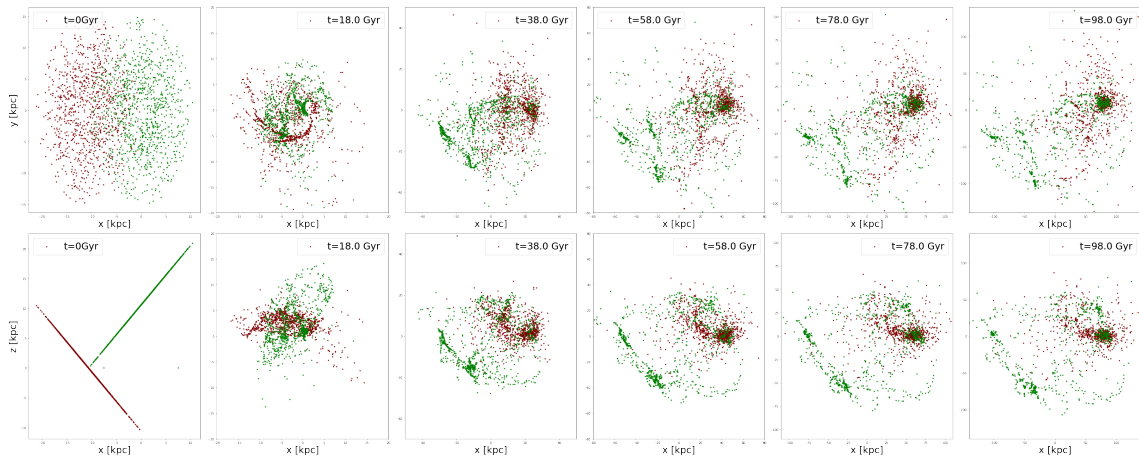


Figure 10: The spatial evolution of the merging between two galaxies of equal mass and a collision angle of  $\pi/2$  radians over 100Gyr. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles. The top row shows the x-y plane, while the bottom row shows the x-z plane, each incrementing in time from left to right with with the time.

### 3.2 Comparing radii ratios

The final simulation with a collision angle of  $\pi/2$  was rerun with a reduction in the radii of galaxy B to create a scenario with a radii ratio of 1:3. The results can be seen in Figure 11 and the `sim_radii.mp4` video and are used to compare to Figure 10 and find the effect of the radii ratio on the resulting merged galaxy.

In the video it is clear that the process of the merger is much shorter in duration than the rest, reaching a state of equilibrium at about 45Gyr. Just as in simulation 3, the more spherical shape of the resulting galaxy is prevalent, with activity in all directions. However, unlike all the previous simulations, the different radii ratio results in two prominent final nuclei rather than one and with majority of the particles remaining gravitationally bound to their original systems. However, the two original systems are still stripped of their original structure and result in one gravitationally bound system.

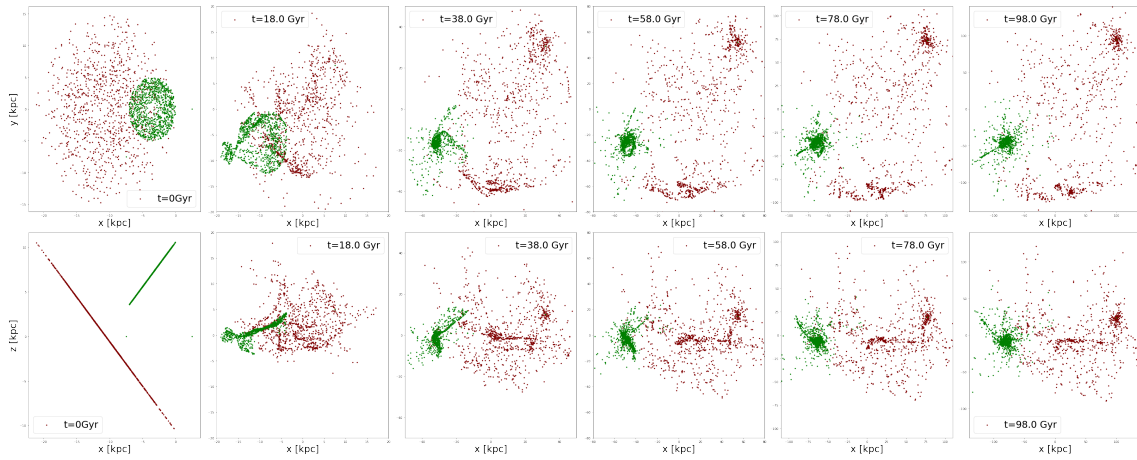


Figure 11: The spatial evolution of the merging between two galaxies of equal mass but unequal radii and a collision angle of  $\pi/2$  radians over 100Gyr. The red stars are gravitationally bound to galaxy A and the green stars are bound to galaxy B. Both galaxies consist of 1000 particles. The top row shows the x-y plane, while the bottom row shows the x-z plane, each incrementing in time from left to right with the time.

## 4 Conclusion

The simulation of the merging of two disk galaxies was performed for 3 different collision angles and two different radii radius. The success of integrating the full duration of these mergers is apparent in the equilibrium reached by all simulations within the integration time of 100Gyr. As an overview, all the simulations stripped the original structure of the two disk galaxies and resulted in an entirely new structure all together, representative of observed galaxies.

Focusing in the simulations of the identical galaxies colliding at angles from 0 to  $\pi/2$  radians, there are some clear effects of the collision angle on the outcome of the merger. The increase in collision angle caused more activity or violence and destruction of the original galaxy shapes, leading to a longer duration before reaching a stabilised state. Additionally, the increase in collision angle results in a more spherically distributed interaction and resulting structure, as expected. While all the simulations of the merging of identical galaxies resulted in a single nuclei, a perhaps less expected result in increasing the collision angle is the offset of this nuclei from the centre, resulting in more irregular galaxies rather than elliptical.

A final comparison of the effects of changing the two galaxies radii to be unequal shows the formation of two nuclei in the final system. This is an interesting result which has been observed in existing galaxies and could be due to the maintenance of an 1:1 mass ratio of the galaxies despite changing their radii ratio. Again, this interaction resulted in an even more violent and quicker merger than the simulations of galaxies with equal radii but does not seem to have an effect on

the spherical distribution of the resulting shape.

All the resulting shapes of the systems produced in the simulations are comparable to real observations. Perhaps the only relatively unrealistic result is that of the “head-on” collision because this scenario is very unlikely and therefore so is the result of a purely disk shaped galaxy with no  $z$  components. Additionally, the lack of spiral structure due to the small number of particles and the initial conditions of the two disk galaxies don’t allow the investigation of the effect of the collision angle and radii ratio on the spiral arms that are usual observed in real galaxies.

There are many ways in which this project could be taken further. A further investigation of the collision angle and radii ratio could be performed with more simulations of varying these parameters. In addition, the investigation of other parameters such as the mass ratio and the direction of rotation would also be very interesting to explore.

The time constraints as well as the computational limitations of this project left plenty room for improvement which is recommended if the topic is to be interrogated further. The many parameter choices were chosen based on these constraints or based on the tests of others, but if this project is repeated it would be preferable to run tests to choose these parameters. This includes finding the optimal time step and integration time for the leap frog integration that best conserves energy and shows the full evolution of the merger. Additionally, the choice of the softening parameter should be tested with the number of particles and the mass distribution thoroughly to prevent the forces being too noisy but also to prevent them from being systematically misrepresented. A final important parameter that should undergo tests for its choice is the theta criterion for the Barnes-hut method. The choice of theta determines the force accuracy and should be chosen optimally for the best accuracy.

Beyond the potential for more rigorous parameter choices, the simulations also come with the flaws of their many assumptions. The first being the fundamental assumption of an N-body simulation, which is that each particle behaves as a point mass. This is of course not the case, and therefore attempts to smooth over the masses with different methods such as smooth particle hydrodynamics provide much more accurate representations of the behavior of these masses. However, the complexity of these kind of simulations favour our approach for short-term simple projects. Perhaps the assumption that most simplifies the simulation is that the disk is purely 2-dimensional and has no  $z$  components, while real disk galaxies are not razor-thin in this way. Again, although it is a crude simplification of a galaxy, the small effect on the results is outweighed by the ease of modelling it provides. Similarly, the assumption that the leap frog method is self-starting is incorrect, but the minimal effect proves it a useful assumption for projects of this scale.

Perhaps the biggest short-coming of which is the adding of a massive particle at the centre of each galaxy to stabilize the system. This is an inaccurate representation of how the mass in galaxies are distributed and it is highly recommended that in future work there is more focus given to improving the initial mass distribution so that the insertion of such a particle is not necessary. Additionally, it is not to be underestimated the potential to optimize the code behind the simulation which could allow for the simulation of more particles and smaller time steps despite a lack of computational power.

While the Barnes-hut method is known to speed up the computation time it is excessive for the small amount of particles simulated in this project. It is therefore recommended to use a simpler approach unless the computational power to simulate much more particles is available. Reflecting on this very simplified model, it is clear there is room for many improvements. However, for a simple short-term project such as this, the assumptions made and the use of parameters chosen in similar literature have proved to be sufficient and beneficial for an initial investigation of such systems with minimal computing power and perhaps an even simpler approach would suffice for the simulation of so few particles.

## 5 References

- [1] Leapfrog integrator. [http://www.physics.drexel.edu/~steve/Courses/Comp\\_Phys/Integrators/leapfrog/](http://www.physics.drexel.edu/~steve/Courses/Comp_Phys/Integrators/leapfrog/). (Accessed on 10/06/2020).
- [2] M. Wielen R. Aarseth, J. Henon. A comparison of numerical methods for the study of star cluster dynamics.
- [3] E. Athanassoula et al. Optimal softening for force calculations in collisionless n-body simulations. 2000. (Accessed on 10/16/2020).
- [4] M Balcells and A González. Formation of Kinematic Subsystems in Stellar Spiral-Spiral Mergers. *THE ASTROPHYSICAL JOURNAL*, 1998.
- [5] J. Barnes and P. Hut. A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature*, 324(4), dec 1986.
- [6] BEHALF. Barnes-hut algorithm for cs205. <https://anaroxanapop.github.io/behalf/#Citations>, 2018. (Accessed on 10/04/2020).
- [7] J Binney and S. Tremaine. Galactic Dynamics. *Princeton Series in Astrophysics*, pages 1–22, 2008.
- [8] A Burkert, W B Burton, M A Dopita, A Eckart, T Encrenaz, E K Grebel, B Leibundgut, J Lequeux, A Maeder, and V Trimble. *Galaxy Formation*. 2008.
- [9] The Astropy Collaboration. Astropy: A community python package for astronomy. 2013. (Accessed on 10/02/2020).
- [10] C. Harris et al. Array programming with numpy. 2020. (Accessed on 10/06/2020).
- [11] B. Holwerda and J. Dalcanton. Galaxy silhouettes. [https://www.nasa.gov/mission\\_pages/hubble/science/hst\\_img\\_20080916.html](https://www.nasa.gov/mission_pages/hubble/science/hst_img_20080916.html). (Accessed on 09/11/2020).
- [12] P. Hut et al. Building a better leap frog. 1995. (Accessed on 10/11/2020).
- [13] M. Miyamoto and R. Nagai. Three-dimensional models for the distribution of mass in galaxies. *Astronomical Society*, 1975.
- [14] A. M. Price-Whelan et al. Gala: A python package for galactic dynamics. 2017. (Accessed on 10/02/2020).
- [15] Volker Springel. The cosmological simulation code gadget-2. *Monthly Notices of the Royal Astronomical Society*, 364(4):1105–1134, dec 2005.
- [16] P. Virtanen et al. Scipy 1.0: fundamental algorithms for scientific computing in python. 2020. (Accessed on 10/06/2020).