

Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Eric Zhang

April 10, 2023

Constructing and Simulating the Dynamics of Galaxies in AMUSE:
A Continuation of a Very BRAVE Program

By

Eric Zhang

Merida Batiste, Ph.D.
Adviser

Physics

Merida Batiste, Ph.D.
Adviser

Alissa Bans, Ph.D.
Committee Member

Meredith Schweig, Ph.D.
Committee Member

2023

Constructing and Simulating the Dynamics of Galaxies in AMUSE:
A Continuation of a Very BRAVE Program

By

Eric Zhang

Merida Batiste, Ph.D.
Adviser

A abstract of
a thesis submitted to the Faculty of the College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements for the degree of
Bachelors of Science with Honors

Physics

2023

Abstract

Constructing and Simulating the Dynamics of Galaxies in AMUSE:

A Continuation of a Very BRAVE Program

By Eric Zhang

Many observations of galaxies have been conducted with spectral analysis, but it is difficult to distinguish the extent of different structural components within a single galaxy based solely on observations. One way to solve this problem is by using simulations, as we can define specific stars to different structural components of the galaxy. Therefore, simulations of galaxies of various morphologies will enable us to understand the effects of different types of galaxy structures on the measured properties of galaxies. Specifically, we want to find a way to isolate and separate the bulge dynamics from the disk and any other structures. We use the simulation software Astrophysical Multipurpose Software Environment (AMUSE) to create particle sets, allowing us to modify and analyze specific stars within the Plummer model. AMUSE also lets us use other simulations with robust hydrodynamics and solve observational problems such as line of sight and structure identification. Our final goal is to create a simple simulation that has physical dynamics and structure to previously observed spiral galaxies. We created two simulations that held galactic structure through a period of 10 Gyr. One has a total galactic mass of $1 \times 10^{10} M_{\odot}$, which has parameters close to the Milky Way galaxy, and one with a mass of $1 \times 10^8 M_{\odot}$, which has parameters similar to dwarf galaxies.

Constructing and Simulating the Dynamics of Galaxies in AMUSE:
A Continuation of a Very BRAVE Program

By

Eric Zhang

Merida Batiste, Ph.D.
Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Physics

2023

Acknowledgements

I would like to thank everyone within the BRAVE Lab, specifically Dr. Merida Batiste and my fellow peer Lucas Alonso-Munoyerro. I would like to thank my friends, in particular my roommates David Goldberg, George Poppitz, and Josh Kim for their continued support in my physical, emotional, and professional life. I would not be nearly as successful without you all pushing me to be as great as you are. I would like to thank the Emory University Symphony Orchestra and all of my musical teachers and friends for giving me a creative outlet few get to pursue. I would like to thank all of my teammates on Juice Ultimate for physically keeping me healthy, mentally strengthening my confidence, and helping me unexpectedly grow as a person. I would like to thank my committee members, Dr. Alissa Bans and Dr. Meredith Schweig, for helping me pursue my professional career. Each of you have given me life-long lessons that I will remember forever. I would like to thank my parents and my brother for supporting my goals no matter the cost or time. Your unwavering support has let me thrive in all pursuits in my life. Finally, I would like to once again thank Dr. Merida Batiste. My work as a researcher and student have been influenced by you more than anyone else, and I strive to live up to the high expectations you set for me. None of this would have been possible without the enormous time and energy investment Dr. Batiste has committed to me.

Contents

1	Introduction	1
2	Methods	7
2.1	Galaxy Structure	7
2.2	Half-light Radius	12
2.3	An Introduction to Galactic Simulations And AMUSE	15
3	The Simulation Itself	18
3.1	Problems with the Original Script	18
3.2	The <i>make_galaxy</i> Definition	20
3.3	The <i>simulation</i> Definition	24
3.4	The <i>new_option_parser</i> Definition	25
3.5	Reference Galaxies	26
4	Discussion and Analysis	29
4.1	The $1 \times 10^8 M_{\odot}$ Galaxy and Adjusting Parameters	31
4.2	The $1 \times 10^{10} M_{\odot}$ Galaxy and MW Parameters	34
4.3	The “Repulsion Force” Problem	36

5	Future Works	42
5.1	Simulation Problems	42
5.2	Compare with Observed Galaxies	43
6	Conclusion	45
A	Milky Way Galaxy Simulation	46
A.1	Original Simulation Script	46
A.2	Modified Simulation Script	50
A.3	Table of Simulations	55
	Bibliography	57

List of Figures

2.1	The Hubble Tuning Fork classification system. Shows ellipticals on the left and spirals on the right, with regular spirals on the top tine of the fork, and barred spirals on the bottom tine.	8
2.2	General structure of a spiral galaxy. For our purposes, we will not differentiate between the thick and thin disk.	10
2.3	Some examples of previously observed spiral galaxies from the Hubble Space Telescope (Batiste et al. 2017). Our simulated galaxies should have similar structure.	11
2.4	An illustrative example of R_e with NGC 3599. (a) shows NGC 3599 as viewed by the Sloan Digital Sky Survey (SDSS) ⁶ . (b) places an effective radius estimate, which is overlayed on top of the original image. Note that a luminosity profile was <i>not</i> fitted to this image, and the estimated radius was made purely by eye to serve as an exaggerated example. (c) is a higher resolution image from the Hubble Space Telescope superimposed on top of the original image, which shows a more accurate representation of how much light is being emitted. (d) shows the effective radius estimated with the higher resolution image, superimposed on top of the original SDSS image.	13

2.5	NGC 5273 ⁷ . Effective radius measurements in ATLAS 3D were found to be five times too large.	14
2.6	The output of all simulations specifies which stellar particles are within the bulge, and which are within the disk. These structures are then combined to create the galaxy as a whole. (a) shows the bulge of the galaxy with preset parameters in the default simulation. (b) shows only the disk of the galaxy. Notice how the bulge and disk occupy the same area within the plot. (c) shows the galaxy as a whole by combining the two previous figures.	16
3.1	The <i>makegalaxy</i> definition	21
3.2	Almost all possible ways a galaxy can be rotated within our simulation. Within the default simulation ($1 \times 10^{10} M_{\odot}$, a radius of 10 kpc, 10^3 particles in the bulge, 10^3 particles in the disk, and 2×10^4 particles in the halo), the axes are shown in each figure but to be descriptive; x points “into and out of the page”, while y and z make up the plane that the figures lie on. (a) shows the galaxy face-on, (b) shows the galaxy rotated 90° along the x-axis, (c) shows the galaxy rotated 90° along the y-axis. It is not very clear to compare the galaxy rotated along the z-axis since it would still be face-on. Thus, (d) shows the galaxy rotated 90° along the x and y axis.	23
3.3	the <i>simulate</i> definition	24
3.4	The <i>new_option_parser</i> Definition	26
3.5	The merge_two_galaxies simulation from the AMUSE textbook. Two identical galaxies collide with one another across 2 Gyr.	27
3.6	The galaxy_merger simulation ⁶ . Two identical galaxies are once again colliding with one another. However, this simulation iterates over 3 trillion years, and uses a different simulation definition.	28

4.1 First simulated galaxy with a total mass of $1 \times 10^8 M_{\odot}$, radius of 10 kpc, 10^4 particles in the bulge, 10^4 particles in the disk, and 2×10^4 particles in the halo. These parameters are emblematic of a dwarf galaxy. 32

4.2 Galaxy with a mass of $1 \times 10^{10} M_{\odot}$, a radius of 30 kpc, 4×10^3 particles in the bulge, 1.3×10^4 particles in the disk, and 2.3×10^4 particles in the halo iterated over 20 billion years 35

4.3 The repulsion force effects seen within the default simulation ($1 \times 10^{10} M_{\odot}$, a radius of 10 kpc, 10^3 particles in the bulge, 10^3 particles in the disk, and 2×10^4 particles in the halo) where the radius was relatively small (10 kpc). Between an iteration of 100 Myr, stars within the galaxy can be seen flying radially outward, which doesn't make physical sense. The galaxy then seemingly recombines before exploding again. 37

4.4 The repulsion force seems to force particles within the center of the galaxy to travel out radially in all directions. This is the default simulation viewed along the x-axis by 90° 38

4.5 Both of the converter functions within the *make_galaxy* and *simulate* definitions . . . 39

Chapter 1

Introduction

It is well understood that at the center of every galaxy, there is a super-massive black hole (SMBH). SMBHs have masses between $10^6 - 10^{10} M_{\odot}$, contained in a region smaller than the solar system (Peterson, 2013). While SMBHs are enormously massive objects, a SMBH makes up less than 1% of the total mass of the host galaxy. This, combined with their very compact nature, means that the gravitational sphere of influence (SOI) of a SMBH is a small fraction of the total size of the galaxy. Therefore, while the SMBH is located at the center of the galaxy, it is not responsible for holding the galaxy together, and its gravitational influence does not affect objects outside the very center of the galaxy.

However, despite this lack of gravitational interaction, tight correlations have been observed between the mass of the SMBH (M_{BH}) and various large scale properties of the host galaxy, implying black hole-galaxy co-evolution. Even though we know that the SMBH *doesn't* have a gravitational effect on the large scale of a galaxy, we see that the mass of the SMBH *does* correlate with various galaxy properties, including: bulge mass (Ramsden, Lanning, Nicholl, & McGee, 2022); gravitational binding energy (Aller & Richstone, 2007); absolute blue luminosity (Dullo, Bouquin, de Paz, Knapen,

& Gorgas, 2020); and bulge stellar velocity dispersion σ_* , which is a measure of the statistical spread of velocities of stars within the bulge of the galaxy (Ferrarese & Merritt, 2000; Beifiori, Courteau, M., & Y., 2012; Shankar et al., 2016; Kormendy & Richstone, 1995).

By 2000, relations between properties of SMBHs and their host galaxies had provided insight into the evolution of both (Ferrarese & Merritt, 2000; Gebhardt et al., 2000; Gültekin et al., 2009). An extremely tight correlation was found between M_{BH} and σ_* , with negligible intrinsic scatter. Unlike all other relations discussed previously, all the scatter in the relation could be attributed to measurement error, suggesting that this so called $M_{BH} - \sigma_*$ relation was the most physically meaningful. The $M_{BH} - \sigma_*$ relation therefore became the starting point for many prediction theories on galaxy evolution and is used to anchor cosmological simulations (Adams, Graff, & Richstone, 2003; Burkert & Silk, 2001).

If all scatter in the $M_{BH} - \sigma_*$ relation can be attributed to measurement error, then it implies it is a perfect relation with no intrinsic scatter. This means that if σ_* within the bulge of the galaxy can be measured, then we can also measure M_{BH} using this relation. This is important because M_{BH} is inherently difficult to measure, and it is *impossible* to measure directly for galaxies outside the local universe, except in a small subset of cases. This means that if we want to track the evolution of SMBHs across cosmic time, well constrained scaling relations provide a vital tool for doing so, allowing large numbers of M_{BH} to be measured with relative ease.

Extensive work went into reducing the amount of error in M_{BH} measurements, since they were known to be the dominant source of measurement error (Ferrarese & Merritt, 2000; Gebhardt et al., 2000). These studies were so successful that we are now at the point where the dominant source of measurement error is now σ_* . However, the scatter in the $M_{BH} - \sigma_*$ relation remained high, despite the significant reduction in measurement error. This means that there is an intrinsic component to the scatter due to properties of the galaxy that are not accounted for in the fit. Error in σ_* needs

to be reduced in order to probe the cause of the intrinsic scatter.

Differentiating structural components within a galaxy is a difficult undertaking, as defining structure within a galaxy can be very complicated. Even simple looking galaxies can contain different structural components that are not obvious within an image, and it is difficult to see where one component ends and another begins (see discussion in Section 2.1). Heightening the problem, there are many observational constraints that further distort the structure of the galaxy, making the components very difficult to define. As observers on or near Earth, we are limited to one line-of-sight to view of galaxies. Therefore, if we view a galaxy “edge on” where the disk overlaps with the bulge of the galaxy, it becomes impossible to separate light coming from within the bulge and light coming from the disk. Moreover, we are limited by resolution of the telescopes we use to image galaxies, which can blur or distort the appearance of the galaxy. Thus, being able to define galaxy structure, and identify which light is coming from within the bulge of the galaxy, are some of the key motivations of this project. Reducing the scatter due to error in measurements will allow us to probe the underlying nature of the relation, and for example, make it possible to see if different types of galaxies follow different relations, if there are structural dependencies, if it changes with time, etc. (Gültekin et al., 2009; Tempel & Tenjes, 2006). Reducing the measurement error in σ_* , and improving our ability to truly isolate different structural components from one another (as discussed in Section 2.1) will allow us to determine exactly how to measure σ_* to get the tightest, most physically meaningful relation, which is a major motivation of this project.

Computer simulations are a way to understand how these different issues affect observational data. By comparing mock observations of simulations with real-world observed data, we can gain more insight into the $M_{BH} - \sigma_*$ relation and the effects of galaxy structure on observed dynamics. By simulating each star within the galaxy as a particle, it is then possible to define which stars are in which part of the galaxy, eliminating problems with line-of-sight and instrumentation limitations.

We can then simulate the issues we know are present (e.g. changing viewing angle, adding noise to mock observations) and see how they affect the observational data. In addition, the number of stars located in each part of the galaxy can be adjusted, along with the mass of the entire galaxy, and the lifetime of the galaxy. By utilizing sophisticated stellar dynamical effects from simulations, it is possible to modify galaxy properties such as galaxy mass and the total number of stars within the galaxy, which should allow us to simulate galaxies with very similar structures to those that have been previously observed. Since simulations give us the specific position of stars at every iteration in time, we can calculate σ_* for stars that we define to be within the bulge. Ultimately, a simulation makes it possible to completely constrain the σ_* of the bulges within these galaxies, providing insight into the $M_{BH} - \sigma_*$ relation.

As of 2023, there are few galactic simulations available to the public. There are however many large-scale cosmological simulations, which analyze the formation of large-scale structure within the Universe. The Illustris simulation studies the properties of SMBHs and their host galaxies within a 106.5 Mpc^3 volume (Sijacki et al., 2015). This is a large volume of space and results in an enormous number of galaxies, which would take us far too much time to realistically process through. Also, while Illustris does provide their data to the public, they do not have their source code easily available to the public. Having access to the source code is vital when creating a simulation software as it provides insight into how a simulation processes dynamics. The Evolution and Assembly of GaLaxies and their Environments (EAGLE) simulation project follows galaxy evolution of a period cube that's 50 cMpc on each side (Crain et al., 2015), which is similarly too large. Both of these cosmological simulations analyze galaxy clusters and the expansion of the Universe (Sijacki et al., 2015; Crain et al., 2015), which is a much larger scale than is appropriate for our purposes. Large-scale cosmological simulations instead treat galaxies as individual particles, and ignore the interstellar dynamics. Smaller-scale cosmological simulations such as the IllustrisTNG project or

subsets of the EAGLE project do not have source code easily available to the public. At the other end of the scale, there are many simulations that analyze interstellar winds, dust interactions, and other substructures of galaxies, which are too small in scale for the kind of analysis we're interested in (Aoyama, hou, Hirashita, Nagamine, & Shimizu, 2018; Matsuda, Fujimoto, Shima, Sawada, & Inaguchi, 1989). For this reason, a simulation software must be chosen that contains the sophisticated dynamic modeling seen within cosmological simulations, but has the resolution to simulate individual stars as particles.

The simulation software we use is the Astrophysical Multipurpose Software Environment (AMUSE) (Pelupessy et al., 2013). AMUSE is written in the python coding language, and allows the user to make use of a range of different simulation codes at once, without needing expertise in each one. Notably, AMUSE provides user-friendly access to the GADGET-2 simulation software, which is widely used and forms the basis for most galactic and cosmological simulations (Volker, 2005; Spiridon & Slusanschi, 2013). AMUSE also has a very accessible and active GitHub repository, where Steven Rieder, creator of AMUSE and Computational Astronomer at Université de Genève, personally answers questions on the open forum. Many other GitHub repositories don't have an active community that engage in discussion on forums or submit their code. There is also an online textbook available within the GitHub repository which contains example code, as well as code from other people's external projects¹. All of this makes AMUSE very accessible to coding novices, while also providing access to state-of-the-art simulation code. AMUSE also has a relatively low cost in terms of simulation time, taking on average around 45 minutes to simulate a galaxy with 40,000 stars for 10 billion years.

In this project I aim to build a physically reasonable galaxy simulation, which is comparable to galaxies that have been previously observed. This means that we want to make galaxy simulations that have mass, radius, and distribution of particles similar to that of the Milky Way and other

¹<https://github.com/amusecode/amuse/tree/main/examples/applications>

spiral galaxies. This thesis is divided into five main sections: Methods, The Simulation Itself, Discussion and Analysis, and Future works. The Methods section gives some important background for the project, including galaxy structure, radius, and a general introduction to galaxy simulations. The Simulation Itself details the simulation code in great detail, and explains the most important definitions and sections of code that let the simulation function. Discussion and Analysis describes the full set of simulations produced, and discusses how and why they function, what problems we identified, and how we have gone about explaining and solving those problems. Finally, we discuss what future work need to be done on the simulation in terms of fixing the dynamics and developing a framework which will allow us to compare our galaxy simulation with previously observed galaxies.

Chapter 2

Methods

2.1 Galaxy Structure

There are trillions of galaxies in the Universe, each of which is made up of gas, dust, and billions of stars. The structure of these galaxies may vary significantly from simple ellipticals all the way to complex spirals with bars, disk, rings, etc. Yet, galaxies have only been known to exist for around a century. In the 1920s, Edwin Hubble measured the distances of what he thought were nebulous clouds. However, the values he found were so dramatically larger than anything else that had been measured within the sky, that the only explanation for the distances of these nebulous clouds was that they lay outside of our own galaxy¹. Thus, the concept of galaxies became real. Hubble then began trying to classify galaxies and came up with the Hubble Tuning Fork, which remains in use today as the primary galaxy classification system (Fig. 2.1)².

¹<https://www.nasa.gov/content/about-story-edwin-hubble>

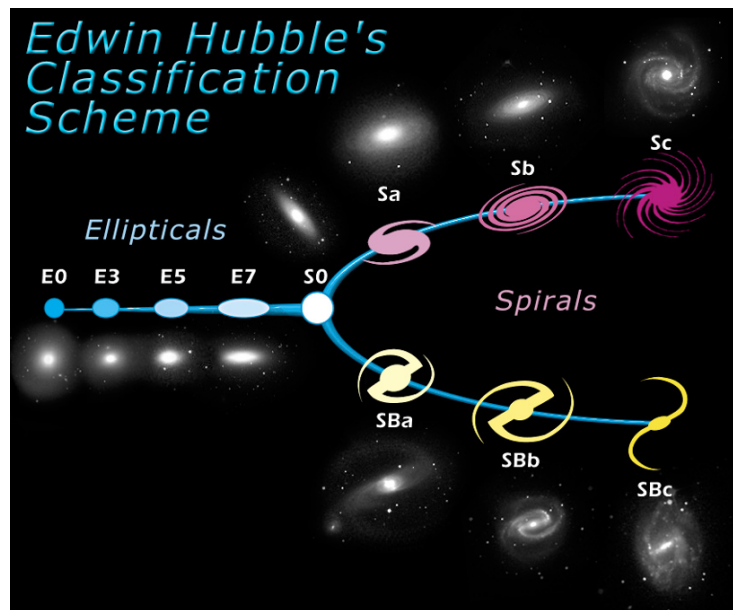


Figure 2.1: The Hubble Tuning Fork classification system. Shows ellipticals on the left and spirals on the right, with regular spirals on the top tine of the fork, and barred spirals on the bottom tine.

First, Hubble divided the galaxies into two broad categories: elliptical and spiral galaxies. This initial classification was based on early observations of these galaxies, which appeared to be fuzzy elliptical clouds. Other galaxies instead had a bright central bulge that was surrounded by a disk containing spiral arms, which extended out from the bulge. Within each of these groups, there is significant variation, thus necessitating subgroups.

Elliptical galaxies are classified by how round or flat they look. An E0 galaxy appears very round, while an E7 galaxy appears very flat. The number after the “E” is determined by the galaxy’s ellipticity, or the ratio of the major and minor axes. Higher ellipticities having higher numbers.

Spiral galaxies are classified by two parameters: how tightly wound the spiral arms are, and the relative size of the central bulge when compared to the disk. These classification are notated by lower-case letters “a, b, and c”. Type “Sa” galaxies have their arms wound very tightly with

²<https://esahubble.org/images/heic9902o/>

large central bulges, and type “Sc” galaxies have loosely wound arms with small central bulges. During Hubble’s original observations, he noticed that some spiral galaxies have a bright bar that runs through the center of the galaxy. Thus, these galaxies are called barred spiral galaxies, and a capital “B” is added to the naming scheme (e.g. a tightly wound, barred galaxy would be denoted as a type “SBa” galaxy).

Galaxies that appear to be in a transitional period between spirals and ellipticals are labeled “S0” on the tuning fork. These are known as lenticular galaxies and have both a central bulge and a disk, but no spiral arms. Finally, galaxies that don’t fall into any of these categorizations are known as irregular galaxies. Irregular galaxies are neither spiral nor elliptical and can have any number of shapes. They are frequently the product of recent galaxy collisions, or significant external gravitational interactions³.

In this project, we are mainly focused on spiral galaxies, where the bulge lies at the center of the galaxy, while a disk of gas, dust, and stars orbits the bulge. This is because spiral galaxies have the most complex structure, and thus it is the hardest to measure σ_* (Batiste, Bentz, Manne-Nicholas, Onken, & Bershady, 2017).

An example of a simple spiral galaxy can be seen in Fig. 2.2⁴. It consists of a bulge, disk, and stellar halo, which are fundamental components of any disk galaxy. In general, it is difficult to distinguish and isolate these structural components from each other. We are specifically interested in isolating the bulge because this is where σ_* must be measured for the $M_{BH} - \sigma_*$ relation. The bulge is difficult to define, since stars within the bulge overlap into regions where the disk and halo lie. It is therefore very difficult to distinguish where the bulge of the galaxy ends and the disk begins.

When trying to reduce error within σ_* , there are many properties of the galaxy that must be considered. The volume and size of the galaxy may have an effect on σ_* , both of which can be

³<https://askanearthspacescientist.asu.edu>

⁴<https://evolution.calpoly.edu/milky-way-galaxy>

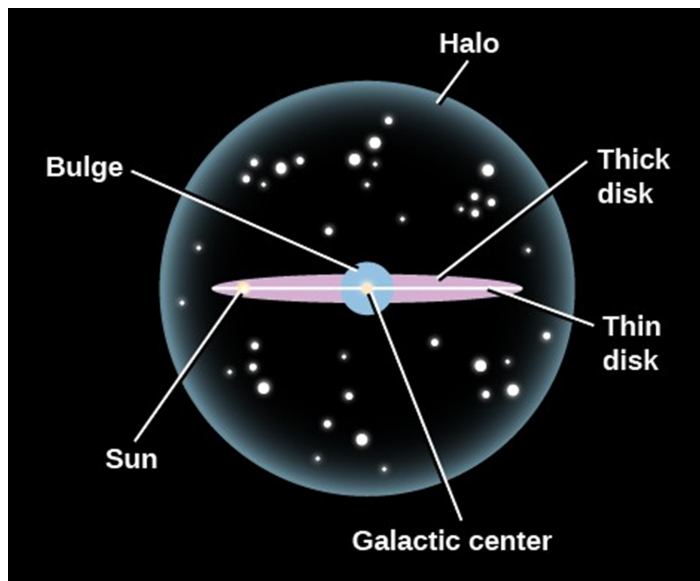


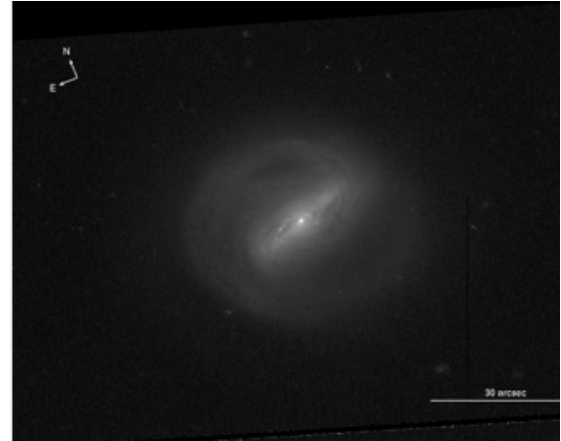
Figure 2.2: General structure of a spiral galaxy. For our purposes, we will not differentiate between the thick and thin disk.

easily modified within a simulation. If scatter in the $M_{BH} - \sigma_*$ relation is due to variations in σ_* , then σ_* may be dependent on the substructure elements of the galaxy and the age of the galaxy (Gültekin et al., 2009). Another possible source of intrinsic scatter is the structure of the bulge. The photometric bulge is the bulge that appears in images. A luminosity function (the amount of light within a galaxy as a function of radius) is fit onto the image, which differentiates structure based on how much light is produced at every location within an image. (this is described in more detail in Section 2.2) (Gadotti, 2009). However, some luminosity profiles reveal that there is a small disk within the bulge⁵. In order to constrain the $M_{BH} - \sigma_*$ relation, the structure of the bulge must be defined.

⁵<http://ned.ipac.caltech.edu/level5/March12/Gadotti/Gadotti3.html>



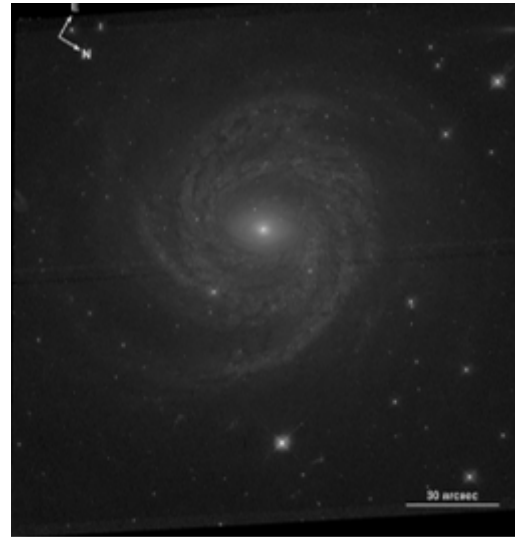
(a) NGC 3368: a spiral galaxy that has two main spiral arms and ring.



(b) MRK 776: a spiral galaxy with a predominant bar spanning from the middle.



(c) NGC 4594: a galaxy that has its line-of-sight edge on.



(d) NGC 6814: a galaxy that's almost perfectly face-on with many spiral arms

Figure 2.3: Some examples of previously observed spiral galaxies from the Hubble Space Telescope (Batiste et al. 2017). Our simulated galaxies should have similar structure.

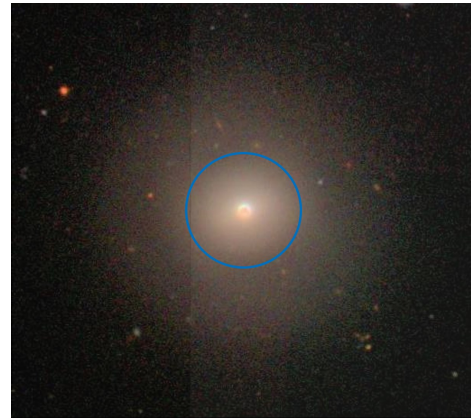
2.2 Half-light Radius

The radius of the bulge of a galaxy can be estimated by calculating the half-light (or effective) radius, which contains half of the total light emitted by the bulge. In order to calculate how much light is emitted by different components of a galaxy, a luminosity function is created. A luminosity function essentially fits how the luminosity of the galaxy changes with radial distance (Stahler & Palla, 2004). Most galaxies require multiple functions to fully fit the light distribution, and each function represents a different structural component. This method allows us to identify different structural components in observed galaxies. By fitting luminosity functions to an observed image, one may estimate the effective radius (R_e) of the bulge, as well as the size of the disk, and any other structural components. An illustrative example of an effective radius (the blue circle) can be seen on galaxy NGC 3559 as shown in Fig. 2.4b⁶. As an illustrative example, the resolution of the image directly impacts the calculation of the effective radius.

⁶https://en.wikipedia.org/wiki/NGC_3599 and <https://in-the-sky.org/data/object.php?id=NGC3504>



(a) NGC 3599



(b) An effective radius estimate



(c) A higher resolution image superimposed on top



(d) The new effective radius estimated

Figure 2.4: An illustrative example of R_e with NGC 3599. (a) shows NGC 3599 as viewed by the Sloan Digital Sky Survey (SDSS)⁶. (b) places an effective radius estimate, which is overlaid on top of the original image. Note that a luminosity profile was *not* fitted to this image, and the estimated radius was made purely by eye to serve as an exaggerated example. (c) is a higher resolution image from the Hubble Space Telescope superimposed on top of the original image, which shows a more accurate representation of how much light is being emitted. (d) shows the effective radius estimated with the higher resolution image, superimposed on top of the original SDSS image.



Figure 2.5: NGC 5273⁷. Effective radius measurements in ATLAS 3D were found to be five times too large.

However, luminosity profiles are dependent on the resolution of an image itself, since it is only possible to fit what is in the image. A lower resolution image will blur out structure, where a higher resolution image reveals a more accurate light distribution, thus allowing for a better profile fit. One example that serves to illustrate this point is the galaxy NGC 5273⁷ (Fig. 2.5), which is an elliptical galaxy that was included in the ATLAS 3D survey, which collected over 260 early type galaxies into their library and database⁸. They used a single component luminosity function to fit the galaxy and find R_e , assuming that this was a simple elliptical (Cappellari et al., 2013). However, analysis by Bentz et al. (2014) later found a 2 component fit was needed to fully describe the light distribution. The difference between these fits meant that the original ATLAS 3D estimate of effective radius was found to have been overestimated by almost five times (Batiste, Bentz, Raimundo, & Vestergaard, 2017).

The problem with distinguishing galaxy structure is further exacerbated when taking into account that we can only see a galaxy as it is oriented along our line-of-sight, meaning that we are only able to view galaxies from one inclination angle. This means that when we take measurements of galaxies,

⁷https://de.wikipedia.org/w/index.php?title=NGC_5273&oldid=192301524

⁸<http://www-astro.physics.ox.ac.uk/atlas3d/>

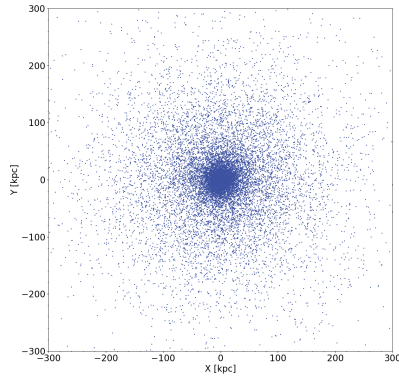
we are condensing a three-dimensional object into a two-dimensional image. Therefore, if the disk lies along our line-of-sight, the disk and bulge will lie within the same area with our image. This means that observations will include light from both components integrated along the line of sight, which will be impossible to distinguish.

One way to solve these problems of galaxy structure identification is through computer simulations. The motivation of this project is to create simulations with similar structure to a set of galaxies that have been previously observed (Batiste, Bentz, Raimundo, & Vestergaard, 2017), some examples of which can be seen in Fig. 2.3. By simulating each star within the galaxy as a particle, we can individually define which stars are in the bulge and which are in the disk. We can also solve the line-of-sight problem by manually changing the inclination angle within the simulation so that no part of the bulge is obstructed. This project is a first attempt at building a physically reasonable simulated galaxy which can act as a basis for future mock observations.

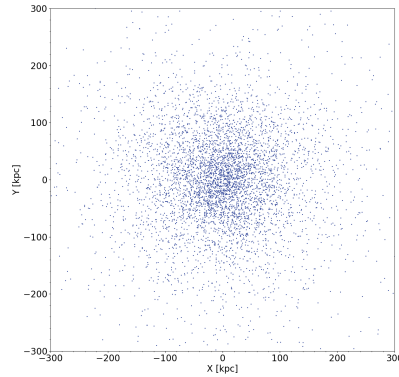
2.3 An Introduction to Galactic Simulations And AMUSE

The ultimate goal of the simulation is to construct a physically reasonable, simple spiral galaxy, which could form the basis for analysis of mock observations. This means that our simulation should contain the main sub-structures of a spiral galaxy (See Fig. 2.6 and discussion in Chapter 3). In order to define substructure, we need to treat individual stars as point particles so that each star can be attributed to a specific substructure.

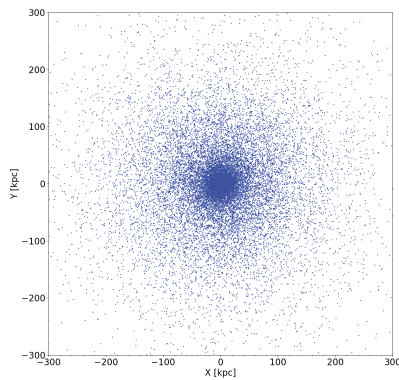
AMUSE uses a fundamental data structure in which all particles are saved as sets (or a list of particles). Based on the attributes of the elements in the sets, selections can be made, which return subsets (in our case, positions of particles). These sets of stars are saved within a “Plummer model”, which is a standard model that simulates density within multiple-particle simulations of



(a) The bulge of a galaxy



(b) The disk of a galaxy



(c) The galaxy as a whole

Figure 2.6: The output of all simulations specifies which stellar particles are within the bulge, and which are within the disk. These structures are then combined to create the galaxy as a whole. (a) shows the bulge of the galaxy with preset parameters in the default simulation. (b) shows only the disk of the galaxy. Notice how the bulge and disk occupy the same area within the plot. (c) shows the galaxy as a whole by combining the two previous figures.

stellar dynamics⁹. AMUSE constantly refers back to and iterates through the Plummer model over the course of a simulation (Zwart & McMillan, 2018).

For many computer simulations, a framework is put in place that handles units and unit conversions. AMUSE does this by creating a basic data object called a “quantity”. A quantity is made up of a value and a unit. Within AMUSE, quantities are created by a bar (|) operator, where the value is specified on the left of the bar, and the unit on the right. All of these quantities are treated like numbers within the code (see Section 3). However, AMUSE by default does not support the use of units. Gravitational dynamics usually work with N-body units internally¹⁰. Therefore, we must use a converter to tell the code how to convert these N-body units into natural units of a specific system. For example, as described in Section 3, a converter is used to append astronomical units onto parameters. The radius is defined as units of kiloparsecs, and mass as solar masses. Within the simulation, the gravitational and interstellar dynamics are controlled through the **new_galactics_model** package, which controls the initial dynamics of the galaxy as it’s being made (see Section 3.2). The fluid-like stellar dynamics are responsible for the disk and spiral arm formation (hydrodynamics), which are controlled by GADGET-2.

⁹<https://amuse.readthedocs.io/en/latest/tutorial/index.html>

¹⁰<https://www.amusecode.org/getting-started>

Chapter 3

The Simulation Itself

3.1 Problems with the Original Script

We began with a simulation script that had been previously in development within the BRAVE program¹, which was constructed to have similar structure and properties to the Milky Way (MW) galaxy. However, the simulation never appeared to reach a dynamically stable state. Within a few 100 Myr, the galaxy would lose structure, and stars would escape the gravitational potential of the galaxy (see Appendix C). Thus, work immediately went into trying to understand the simulation from the original script, in order to have it hold together and understand the problem. Copies of both the original script and the final script can be seen within the Appendix.

The first goal was to understand exactly how the script functioned. That required going through line by line and developing an understanding of every package or function not well understood or documented within the code. Comments describe the purpose of each line, and are denoted by a # before the comment itself, and the text is bolded within copies in the Appendix.

To more accurately understand the role of each piece, attention was brought to why the simulation

¹Simulation script was originally developed and created by Diego Gonzalez

was failing. An initial examination directed our focus on the properties of the galaxy within the simulation. Within the code, the user may control the mass and radius of the galaxy as well as the distribution of particles (explained more in depth in Section 3.4). Originally, the simulation had the total mass of the galaxy as $1 \times 10^{12} M_{\odot}$, the radius of the galaxy as 10 kpc, and 10^4 stars within each of the bulge and disk. $1 \times 10^{12} M_{\odot}$ is a reasonable estimate for the mass of the MW (Peterson, 2013). However, measurements of the radius for the MW are closer to 15 kpc (Peterson, 2013), and the bulge to disk ratio is closer to a 1/3 ratio, rather than an even distribution (Habing, Olmon, Chester, Gillett, & Rowan-Robinson, 1985). Thus, modifications were made to these parameters to be more physically realistic to the MW galaxy.

There was great confusion as to what the halo physically represented. I initially interpreted the halo parameter as representative of a dark matter halo. Thus, the initial ratio of the parameters were set as a 1-3-6 ratio for the bulge, disk, and halo respectively (see Section 2.1 and Fig. 3.6). This is because there are three times as many stars in the disk as the bulge, and there is twice as much mass in a dark matter halo as the disk². Even though the dark matter halo should not have any stars, I believed that adding more particles would change how much mass was in the dark matter halo. However, simulations with these ratios produced non-physical results and the structure of the galaxy broke down. After extensive efforts to find a ratio that would work, it became clear that the halo was in fact defined to be the number of particles within the bulge and the disk due to how it is defined within the code (see Section 3.2). Thus to produce physical results, the galaxy should have the total number of particles in both the bulge and disk be equal to the number of particles in the halo, effectively treating the halo as the galaxy as a whole.

A great deal of time was also spent trying to understand how the the simulation attached units to values in the converter function. Initially, I believed that the converter also had a scale factor associated with the conversion. This is due to python's bitwise operator function sharing the same

²https://chandra.harvard.edu/xray_sources/pdf/milky_way.pdf

notation as AMUSE’s quantity notation (`()`). If it were a bitwise operator, then an additive operation would have been performed on every unit conversion which was perceived to be a scale factor. Therefore, a scale factor was applied to all definitions involving the converter and the parameters themselves (bulge, disk, and halo particle numbers). The number of particles were adjusted so that they would be multiplied by a scale factor (e.g. 10 stellar particles was thought to represent 10,000 stars). However, it was later found within the AMUSE tutorial documentation that the `|` operator was instead a quantity. Instead of a scale factor, a quantity appends a unit value onto a numerical object. I then understood that the converter function also serves to contain knowledge of the units and values through the quantity object. Thus, the parameters were removed of their scale factor, and returned to being a true value. These converters can be seen in Sections 3.2 and 3.3.

3.2 The *make_galaxy* Definition

To begin, the script first creates plots and a comma separated value (CSV) list which saves the position of every star every 100 Myr. The default units used are kiloparsecs, and the plot range is from -300kpc to 300 kpc. Initial parameters are set so the viewing angle is pointed directly at the galaxy “face on” so that no particles from the disk would obscure the bulge.

The next part of the code creates the galaxy as a function of parameters the user can control. The *make_galaxy* definition (Fig. 3.1) creates a galaxy as a function of the mass of the galaxy (in units of M_{\odot}), the radius of the galaxy (in units of kpc), and the number of particles in the halo, and how they are distributed between the bulge and disk. These are then defined as variables **M_galaxy**, **R_galaxy**, **n_halo**, **n_bulge**, and **n_disk** respectively. First, a converter is run to convert the simulation from “AMUSE units” to astronomical units. The galaxy is created by using the in-built **new_galactics_model** package, which uses the variables we have defined earlier. This package is responsible for how AMUSE builds a galaxy, and contains important physics detailing the

```

1 def make_galaxy(M_galaxy, R_galaxy, n_halo, n_bulge, n_disk):
2     converter=nbody_system.nbody_to_si(M_galaxy, R_galaxy)
3     galaxy = new_galactics_model(n_halo,
4                                 converter,
5                                 #do_scale = True,
6                                 bulge_number_of_particles=n_bulge,
7                                 disk_number_of_particles=n_disk)
8
9     #Change line-of-sight (rotate along y axis, rotate along x axis, rotate along z
10    galaxy.rotate(0.0, 0.0, 0.0)
11
12    galaxy.velocity += [0.0, 0.0, 0.0] | units.km/units.s
13    return galaxy, converter

```

Figure 3.1: The *makegalaxy* definition

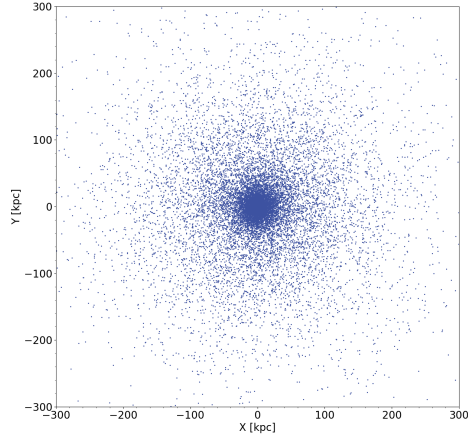
gravitational and interstellar dynamics. The original source code for **new_galactics_model** could not be found, so it is not clear exactly how this galaxy is built. This is a core package in AMUSE, so we have reason to trust it. However, it is a “black box” where we input values, and trust that the vales at the end are physical. This is explained more in Section 5.1. The *make_galaxy* definition can be seen below:

It should be noted that when running simulations, it was found that there seemed to be a force exerted on all of the particles within the bulge. A possible source of this repulsion force is within the **new_galactics_model** package. This repulsion force became a problem within all simulations as particles would seemingly fly out of the bulge of the galaxy, leading to the galaxy eventually dissociating. Thus, the radius and mass of the galaxy were modified to reduce these effects. This is discussed more in Section 4.3.

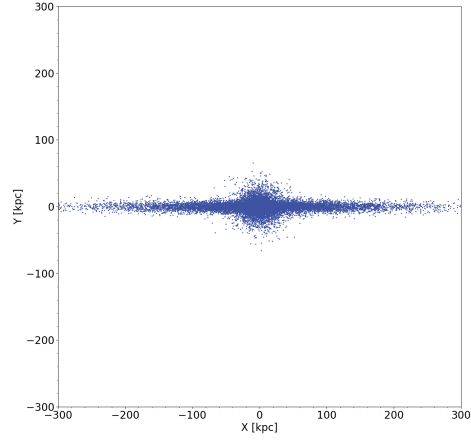
As discussed previously, simulations help us resolve line-of-sight problems by allowing for particles to be defined as being in the bulge or disk without observational obstructions. Our simulation can be rotated so that it can be observed at any inclination angle. The code that does this is defined within the **galaxy.rotate** function in line 10 of Fig. 3.1, which can be used to rotate the galaxy in any orientation we want. Each of the three values allows for the galaxy to rotate along an x, y, and

z axis. An example of how the galaxy can be rotated is shown in Fig. 3.2. It should be noted that this was not found within the original script, and is one of the pieces of functionality added from the `merge_two_galaxies` example found within the AMUSE textbook examples³ (Fig. 3.5).

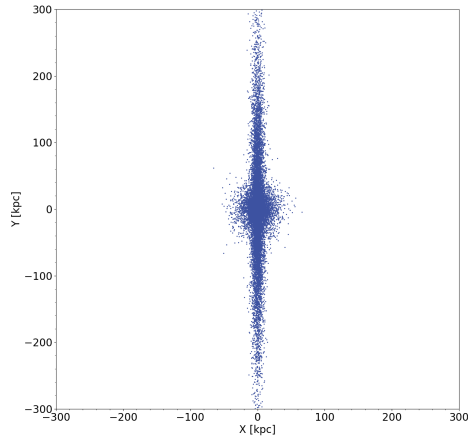
³<https://github.com/amusecode/amuse/tree/main/examples/textbook>



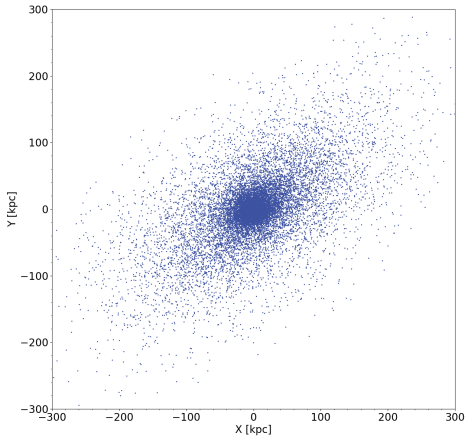
(a) Galaxy face on



(b) Galaxy rotated around the x-axis



(c) Galaxy rotated around the y-axis



(d) Galaxy rotated on the x and y axis

Figure 3.2: Almost all possible ways a galaxy can be rotated within our simulation. Within the default simulation ($1 \times 10^{10} M_{\odot}$, a radius of 10 kpc, 10^3 particles in the bulge, 10^3 particles in the disk, and 2×10^4 particles in the halo), the axes are shown in each figure but to be descriptive; x points “into and out of the page”, while y and z make up the plane that the figures lie on. (a) shows the galaxy face-on, (b) shows the galaxy rotated 90° along the x-axis, (c) shows the galaxy rotated 90° along the y-axis. It is not very clear to compare the galaxy rotated along the z-axis since it would still be face-on. Thus, (d) shows the galaxy rotated 90° along the x and y axis.

```

1 def simulate(galaxy, converter, n_bulge, n_halo, t_end):
2
3     converter = nbody_system.nbody_to_si(1.0e12|units.MSun, 100|units.kpc)
4     dynamics = Gadget2(converter, number_of_workers=8)
5     dynamics.parameters.epsilon_squared = (100 | units.parsec)**2
6     set1 = dynamics.particles.add_particles(galaxy)
7     dynamics.particles.move_to_center()
8
9     disk = set1[:n_bulge] # This is just the thin and thick disks #:n_halo]
10    bulge = set1[n_bulge:n_halo] # This is just the bulge
11    disk_and_bulge = set1[:n_halo] # This is everything except dark matter halo

```

Figure 3.3: the *simulate* definition

3.3 The *simulation* Definition

Next, we simulate the galaxy dynamics using some of the same parameters we defined within the *make_galaxy* definition; **galaxy**, **converter**, **n_halo**, **n_bulge**, and **t_end**. **t_end** is the time we want to run the simulation (for our purposes, around 10 billion years, which is roughly the age of the MW). In the section of the script shown below (Fig. 3.3), we set the mass of the galaxy to be $1 \times 10^{12} M_{\odot}$ and the radius to be 10 kpc. Note that these parameters should be changed if we want to vary the mass and radius of the galaxy. We then use GADGET-2 to simulate the interstellar dynamics of stars within the galaxy in line 4. **dynamics.parameters.epsilon_squared** is a smoothing parameter responsible for the fluid-like (hydrodynamic) physics of stars within the disk/spiral arms of the galaxy. We then attribute these particles and their dynamical interactions within a set, and move the particles to the center of the plot.

As mentioned in Section 3.1, the halo parameter represents all of the stars within both the bulge and disk, acting essentially as the galaxy. Thus, for the simulation to function properly, the sum of particles within the bulge and disk should equal the number of particles within the halo. However, it is possible for the simulation to run with the halo being unequal (as mentioned in Section 3.1). At extremes (where the halo is 10 times as many particles as the bulge + disk), the simulation produces something non-physical or doesn't run at all. However, at more moderate values, the simulation

accounts for this mismatch between the number of particles by adding the excess to the bulge. This is because of how the bulge is defined within the *simulate* definition. As seen in lines 9 - 11, The disk is defined to be all of the particles up to **n_bulge**. The bulge is then defined to be the all the particles between **n_bulge** and **n_halo**. Thus, any extra particles defined to be in **n_halo** are appended to the end of the list, which is then recorded as being part of **n_bulge**. We iterate through the simulation every 100 Myr, and record the positions and plots during each iteration.

3.4 The *new_option_parser* Definition

Finally, we define all of our variables (**M_galaxy**, **R_galaxy**, **n_halo**, **n_bulge**, **n_disk**, and **t_end**) within a *new_option_parser* (OP) definition (Fig. 3.4). Here, the *simulate* and *make_galaxy* definitions refer back to the OP definition to set the variables that the rest of the entire simulation uses. This means that the user may change the “default” values to whatever they choose (within physical reason where the parameters are similar to those of observed galaxies). The original default number of particles was determined when the script was originally created. However, those numbers are not physically representative in observations. Based on standard observations, it’s been found that the disk should have roughly three times as many stars as the bulge, while the default values have equivalent numbers of stars in each⁴. The OP definition also allows the user to change the mass of the galaxy (in units of M_{\odot}) and the radius of the galaxy. For spiral galaxies, the mass of the galaxy is chosen to be around $10^{10} - 10^{15} M_{\odot}$, and the radius around 15 - 30 kpc (Peterson, 2013). The lifetime of the galaxy/length of the simulation can also be edited here.

⁴https://chandra.harvard.edu/xray_sources/pdf/milky_way.pdf

```

1 def new_option_parser():
2     from amuse.units.optparse import OptionParser
3     result = OptionParser()
4     result.add_option("-M", unit=units.MSun,
5                       dest="M_galaxy", default = 1.0e12 | units.MSun,
6                       help="Galaxy mass [%default]")
7     result.add_option("-R", unit=units.kpc,
8                       dest="R_galaxy", default = 10 | units.kpc,
9                       help="Galaxy size [%default]")
10    result.add_option("--n_bulge", dest="n_bulge", default = 10000,
11                      help="number of stars in the bulge [%default]")
12    result.add_option("--n_disk", dest="n_disk", default = 10000,
13                      help="number of stars in the disk [%default]")
14    result.add_option("--n_halo", dest="n_halo", default = 20000,
15                      help="number of stars in the halo [%default]")
16    result.add_option("--t_end", unit=units.Myr,
17                      dest="t_end", default = 2000|units.Myr,
18                      help="End of the simulation [%default]")
19    return result

```

Figure 3.4: The *new_option_parser* Definition

3.5 Reference Galaxies

In addition to the MW galaxy simulation, two galaxy merger simulations were also used as a sanity check, as these two simulations share many parts of code with the one we worked on. The first simulation, titled `merge_two_galaxies`, comes from the AMUSE textbook example⁵. This simulation seemed to be the basis of the original simulation, and contains very similar *make_galaxy*, *simulate*, and *new_option_parser* definitions. However, it should be noted that this simulation is a galaxy merger, which may explain some of the odd repulsion force effects described earlier. This is discussed more in depth in the Discussion and Analysis section. The second simulation titled `galaxy_merger`⁶ uses the `new_galactics_model` package in a similar way, but instead creates two galaxies with initial velocities opposite to one another, resulting in them colliding. This simulation uses the same `new_option_parser` package and definition as our simulation, allowing us to compare the interstellar dynamics.

We also have the two reference simulations that were used as sanity checks throughout the

⁵<https://github.com/amusecode/amuse/tree/main/examples/textbook>

⁶https://github.com/amusecode/amuse/blob/main/examples/applications/galaxy_merger.py

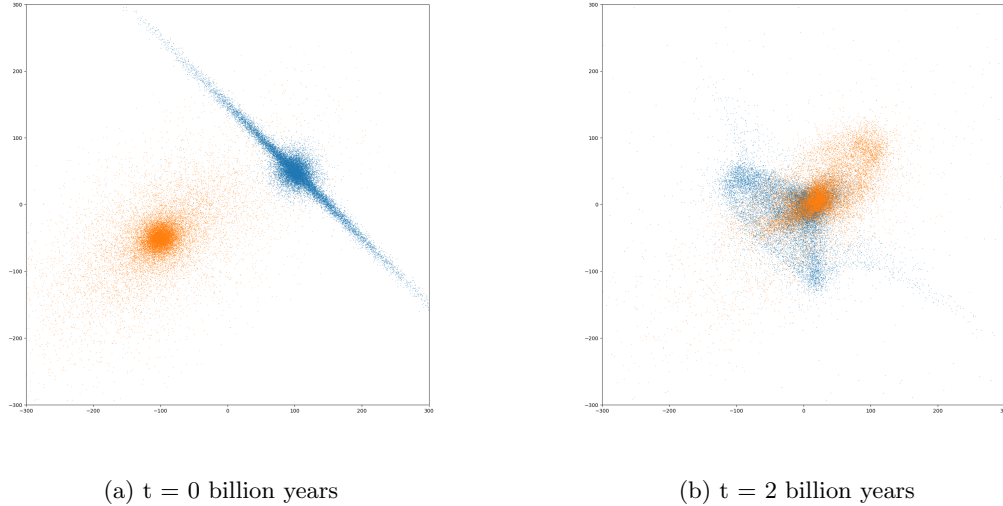


Figure 3.5: The `merge_two_galaxies` simulation from the AMUSE textbook. Two identical galaxies collide with one another across 2 Gyr.

project. We know that much of the code is shared between these two simulations. Therefore, galaxy structure and dynamics within these two simulations should be relatively similar to our own simulation. The first is the `merge_two_galaxies` simulation⁷ (Fig. 3.5), which was found within the AMUSE textbook, and the source code is provided within the GitHub repository. This simulation flings two galaxies together with an initial velocity so they collide. This simulation then produces two plots: one at an initial time where no dynamics are run, and one 2 billion years later. The other simulation was found online and titled `galaxy_merger`. It should be noted that this simulation provided the rotational line of sight code which we implemented into our own simulation. It also uses the `new_galactics_model` package that our simulation uses. This simulation also flings two galaxies together, though spanned over a time of 3 trillion years. By comparing how this simulation with the previous galaxy merger, we now have two different simulations code which should produce similar results. From there, we can use the definitions within each code and implement them into our own simulation.

⁷https://github.com/amusecode/amuse/blob/main/examples/textbook/merge_two_galaxies.py

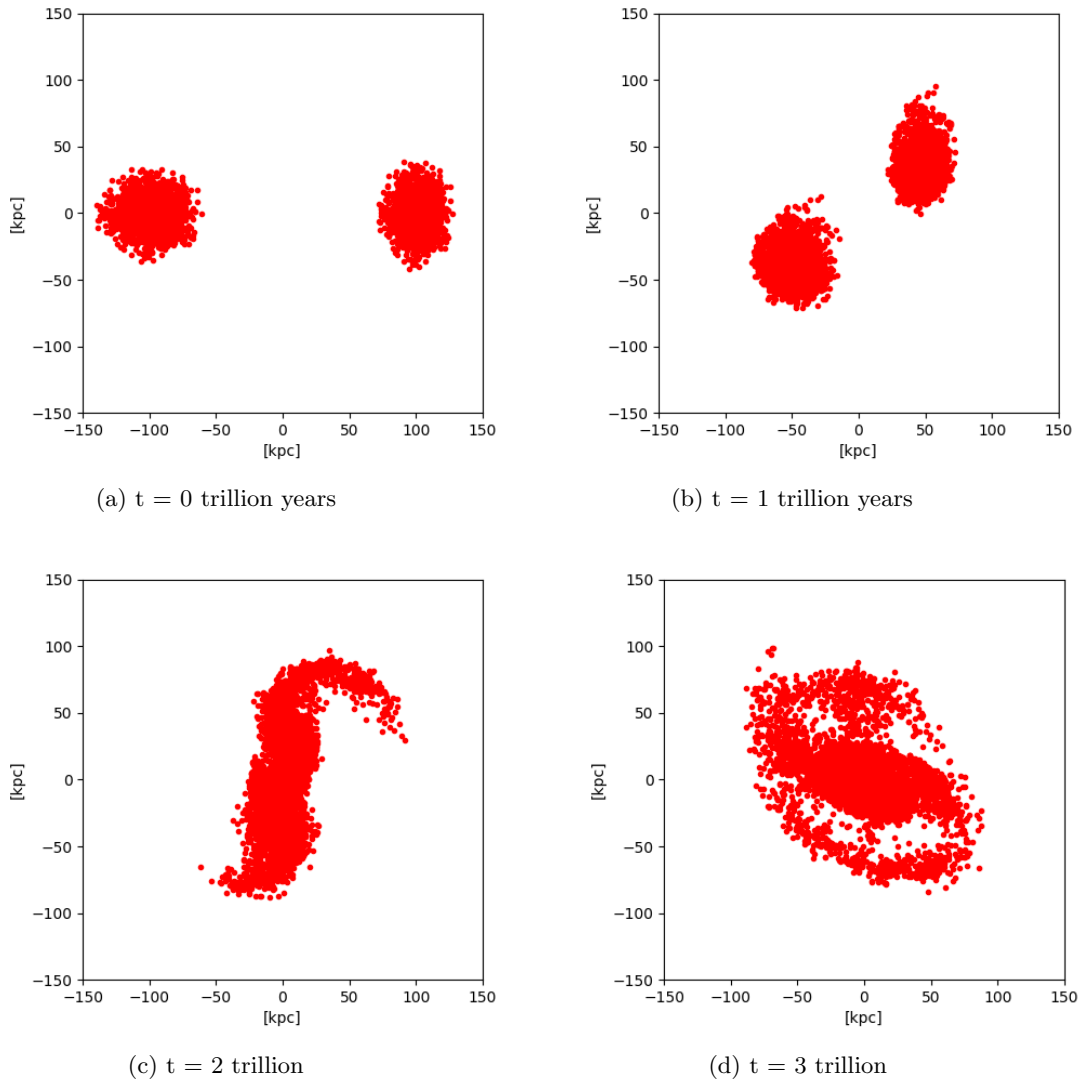


Figure 3.6: The `galaxy_merger` simulation⁶. Two identical galaxies are once again colliding with one another. However, this simulation iterates over 3 trillion years, and uses a different simulation definition.

Chapter 4

Discussion and Analysis

After the original script was modified so that everything was relatively well understood, work was done to create physical galaxies that exhibit structure similar to the MW. In order to do this, the parameters of the galaxy were varied across a physically reasonable range, to probe the full parameter space (within the OP package). This means that we changed the values of the galactic mass and radius, as well as the number of stellar particles within the bulge and disk.

As a starting point, the original script had a galactic mass of $1 \times 10^{12} M_{\odot}$, a radius of 10 kpc, and 10^4 particles within the bulge and disk, leading to a halo with 2×10^4 particles. As discussed previously, these ratios of particles are different than observed in the MW-type galaxies where the bulge and disk are more of a one-to-two ratio. The radius is also a bit small for the MW (which ranges from 15 - 30 kpc). When the simulation was run with these “default” settings, the galaxy was found to disperse within a few 100 million years. As seen in Appendix C, 25% of the particles had left the field of view (FOV). Ultimately, it is necessary for the simulated galaxy to remain intact and stable up to at least the current age of the universe (13 billion years). However, they still provided a good basis to start modifying the parameters.

Our first goal was to create a stable simulation which held structure until the end of the simulation by modifying only one parameter. Thus, we wanted to change the mass of the galaxy without changing any of the other parameters to see how the mass affected our simulation. To start, simulations were run with the total mass of the galaxy at $1 \times 10^{10} M_{\odot}$, a radius of 10 kpc, 10^4 particles in the bulge, 10^4 particles in the disk, and 2×10^4 particles in the halo iterated over 20 billion years (the default simulation parameters). As mentioned in Section 3.1, this simulation immediately lost structure after a by 700 Myr, and particles were evenly distributed across the whole FOV becoming homogeneous. Since the galaxy is falling apart so quickly, we thought that this may be due to stars within the bulge of the galaxy not having significant enough gravity to hold itself together. Thus, the mass of the galaxy was increased.

Five simulations were run at a range of masses from $1 \times 10^5 M_{\odot}$ to $2 \times 10^{15} M_{\odot}$ in increments of five magnitudes ($1 \times 10^5 M_{\odot}$, $1 \times 10^{10} M_{\odot}$, etc.), some of which can be seen within Appendix C. An additional simulation was also run at $1 \times 10^8 M_{\odot}$ once it was seen that the $1 \times 10^5 M_{\odot}$ simulation did not produce galactic structure (see Section 4.1). No other parameters were modified, so they share the same parameters as the default simulation. It was found that an increase in mass exaggerated the effects of dissociation more. In this thesis, dissociation will be defined as when particles would appear to no longer exhibit galactic structure and become homogenized across the entire FOV. It should be noted that a galaxy could completely lose structure and become dissociated while not leaving the FOV. In the $2 \times 10^{15} M_{\odot}$ simulation, it was found that by 2.5 Gyr (1/4 of the run-time of the simulation), more than 40% of the particles had exited the FOV of $300 \text{ kpc} \times 300 \text{ kpc}$. Thus, simulations were run where mass of the galaxy was then reduced to as small as $1 \times 10^5 M_{\odot}$. After successive simulation runs, it was found that reducing the mass of the galaxy showed improvements of the simulation remaining stable. Within the $1 \times 10^{10} M_{\odot}$ simulation, only 20% of particles were outside of the FOV by 2.5 Gyr.

4.1 The $1 \times 10^8 M_{\odot}$ Galaxy and Adjusting Parameters

Our first simulation where a galaxy remained stable (Fig. 4.1) had the total mass of the galaxy at $1 \times 10^8 M_{\odot}$, a radius of 10 kpc, 10^4 particles in the bulge, 10^4 particles in the disk, and 2×10^4 particles in the halo iterated over 20 billion years.

During the whole simulation time, no particles fly out of the FOV. Spiral arms can be seen forming within this simulation and the galaxy does not dissociate. However, $1 \times 10^8 M_{\odot}$ is relatively small when compared with our MW, making it resemble a dwarf galaxy. Spiral arms also do not form until 20 billion years, which is much too late since it is longer than the age of the universe. However, this was a great breakthrough as it showed that our simulation *could* produce galaxies which remained stable and exhibit spiral arms. This was not achieved with any other mass, when no other parameters were varied.

One simulation was run where the mass was lowered to $1 \times 10^5 M_{\odot}$. However, the simulation no longer produced physical results, and the stellar particles no longer appeared to have any gravitational effect on one another. The galaxy was completely still, and there were no changes in stellar position over the entire run time. However, a galactic mass of $1 \times 10^5 M_{\odot}$ is very low (even for dwarf galaxies¹), so non-physical results are to be expected.

The next step was then adjusting the number of stellar particles within the bulge and disk to make it resemble the MW's stellar ratios. As previously discussed, this was when the halo parameter was not yet well understood, and it was believed that the halo referred to a dark matter halo. Thus, we treated the halo as if it was a dark matter halo, and the number of particles was modified to give a 1-3-6 distribution of particles, where the disk has three times as many particles as the bulge, and the halo has twice as many particles as the disk. The extra particles within the halo are accounted in the bulge as discussed in Section 3.3.

¹<https://astronomy.swin.edu.au/cosmos/D/Dwarf+Galaxy>

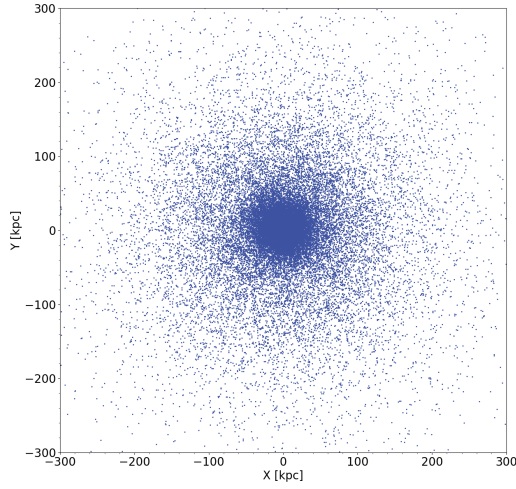
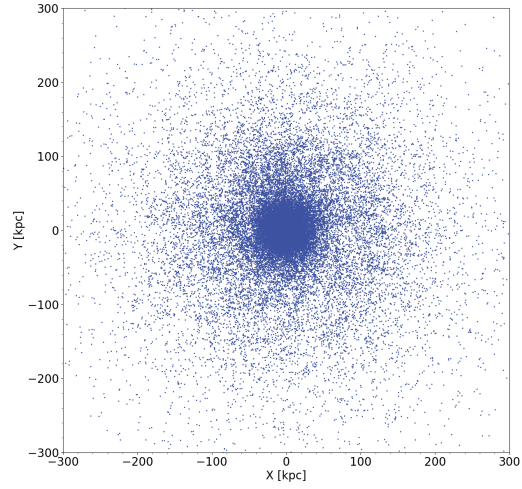
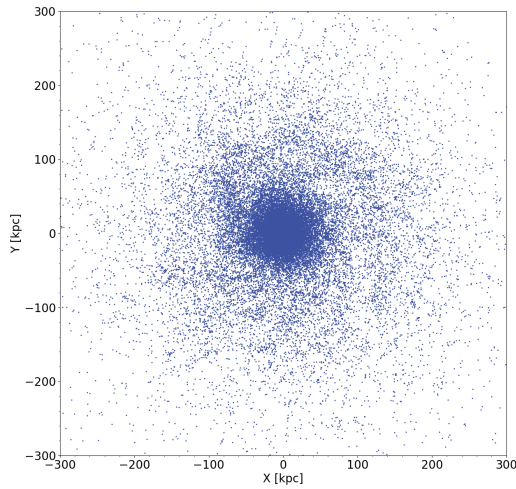
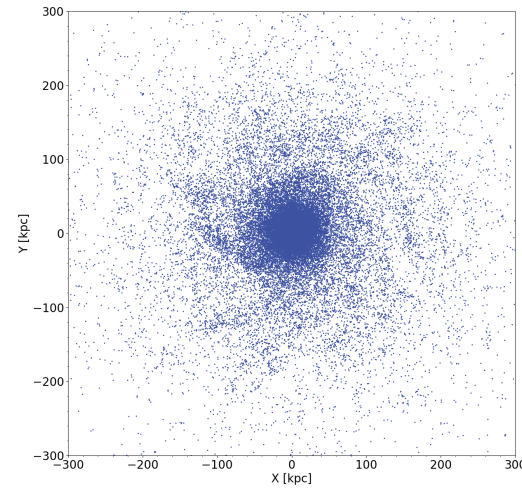
(a) $t = 5$ billion years(b) $t = 10$ billion years(c) $t = 15$ billion years(d) $t = 20$ billion years

Figure 4.1: First simulated galaxy with a total mass of $1 \times 10^8 M_{\odot}$, radius of 10 kpc, 10^4 particles in the bulge, 10^4 particles in the disk, and 2×10^4 particles in the halo. These parameters are emblematic of a dwarf galaxy.

We started with a distribution of 4×10^4 particles in the bulge, 1.3×10^4 in the disk, and 2.3×10^4 within the halo which is equivalent to 9×10^3 in the bulge and 1.3×10^9 in the disk. We also increased the mass to $1 \times 10^{15} M_{\odot}$ and a radius of 40kpc. Even with the confusion regarding the halo, the simulation still approaches a 1-2 distribution of particles which is seen in the MW. This distribution of particles would remain constant throughout the simulations because it has similar proportionality to that of the MW. It was believed that since we were doubling the number of particles within our simulation (from the repulsion force within the $1 \times 10^8 M_{\odot}$ simulation to now 40k), the mass and radius of the galaxy must be increased.

Thus, for an initial simulation with the new proportions, the total mass was increased to $1 \times 10^{15} M_{\odot}$ and the radius to 40 kpc. These values were chosen by roughly doubling the radius of 20 kpc, and setting the mass to the lower limit of the MW's mass. When the mass was set with this new set of parameters, the simulation once again started to dissociate. However, the more physically realistic ratios seemed to hold galactic structure to 7.5 Gyr, before they inevitably dissociated. When compared with the default simulation where the galaxy would dissociate after only a few 100 Myr, this showed that the new ratio of particles appeared to remain in galactic structure. No particles also flew out of the FOV during this simulation either.

From there, the mass and radius were reduced to approach that of the MW's and to exhibit galactic structure. A simulation was run at $1 \times 10^{12} M_{\odot}$ and a radius of 40 kpc. The galaxy once again seemed to hold in galactic structure for about 7.5 Gyr until it dissociated, and no particles flew out of the FOV. This continued the trend that reducing mass held the structure of the galaxy for longer and reduced (or eliminated) the number of particles which flew out of the FOV (see Appendix C).

4.2 The $1 \times 10^{10} M_{\odot}$ Galaxy and MW Parameters

The simulation that held galactic structure had the total mass of the galaxy at $1 \times 10^{10} M_{\odot}$, a radius of 30 kpc, 4×10^3 particles in the bulge, 1.3×10^4 particles in the disk, and 2.3×10^4 particles in the halo iterated over 20 billion years (Fig. 4.2). This simulation contained 10^4 particles in the bulge and 1.3×10^4 in the disk. This simulation is much more loosely bound, and spiral arms are not visible, implying that this galaxy is more lenticular than spiral.

While the radius of the galaxy is comparable to observed measurements of the MW galaxy, the mass is still less than the mass of the MW ($1 \times 10^{12} M_{\odot}$). Thus, more simulations were run with a larger mass in order to approach $1 \times 10^{12} M_{\odot}$. However, none of these simulations ever held galactic structure over the course of the entire simulation, even with a reduction in radius size. Within these simulations, some particles left (in Appendix C, the bottom 3 simulations never lost more than 4% of the total particles), but they still became homogeneous throughout the FOV by 7.5 Gyr. Simulations with masses and radii at $1 \times 10^{12} M_{\odot}$ and 30 kpc; $1 \times 10^{12} M_{\odot}$ and 20 kpc; and $1 \times 10^{12} M_{\odot}$ and 10 kpc all failed to hold galactic structure through the entire simulation.

Attention was then brought back again to the distribution of particles, specifically within the halo. particles were adjusted to a 1-2 ratio due to how the halo was defined. Thus, a new goal was set of making a simulation with parameters of that similar to that of the MW. The total mass of the galaxy should be around $1 \times 10^{12} M_{\odot}$, the radius between 15 - 30kpc, and the particle ratios with a 1-2 ratio of particles in the bulge and disk. As of now, we have not been able to create a galaxy with these new properties which holds galactic structure past 7.5 billion years. It once again has been found that reducing the mass of the galaxy maintained the structure for longer iterations, as well as reducing the radius. However, we can only reduce the mass and radius so far before we approach a dwarf galaxy (as seen in Section 4.1). Increasing the total number of particles has seemed to help maintain structure as well. One possible explanation for this is that more particles increased the

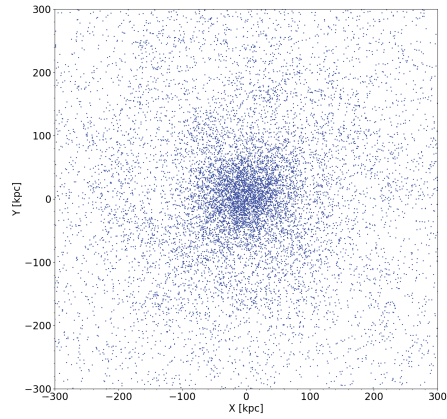
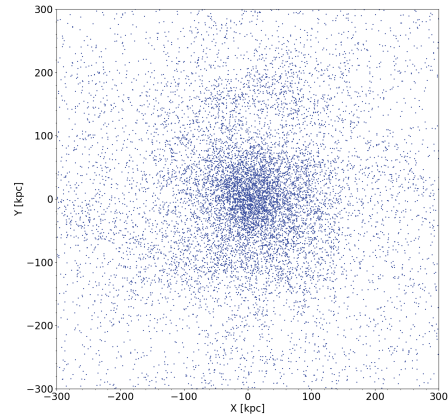
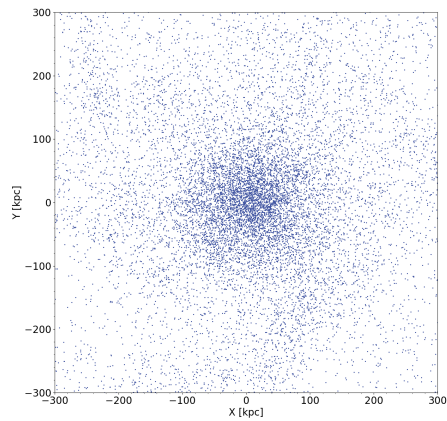
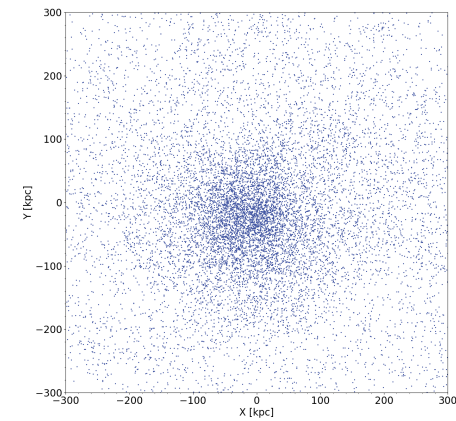
(a) $t = 5$ billion years(b) $t = 10$ billion years(c) $t = 15$ billion years(d) $t = 20$ billion years

Figure 4.2: Galaxy with a mass of $1 \times 10^{10} M_{\odot}$, a radius of 30 kpc, 4×10^3 particles in the bulge, 1.3×10^4 particles in the disk, and 2.3×10^4 particles in the halo iterated over 20 billion years

total gravitational effects of the galaxy, helping it to hold together. However, the repulsion force described earlier in Section 3.1 is the ultimate problem of why our galaxies seem to dissociate.

4.3 The “Repulsion Force” Problem

In simulations, it can be seen that there is what we call a “repulsion force” which appears to seemingly forces particles out of the bulge and disk radially (see Fig. 4.3) within short simulation times (100 Myr). We do not know what causes this but it is clear that this is not physical. What we do know is that this problem is not structure dependent since it occurs in both the bulge and disk of the galaxy.

We do not know if this unknown force is a gravitational force, but there’s evidence that this is a repulsion force problem because of the number of stellar particles which have left the FOV. Appendix C gives results from many simulations run with an evolution of the number of particles. Within smaller mass and smaller radius galaxies, the repulsion force appears to have a greater impact on the simulation, causing more particles to fly out of the FOV. This is counter-intuitive because as discussed previously, the lower mass and radii galaxies showed a trend of holding structure for longer iterations (see sections 4.1 and 4.2). Yet, in our default simulation which has a relatively small mass and radius (mass of $1 \times 10^{10} M_{\odot}$, radius of 10 kpc, 10^4 particles in bulge, 2×10^4 in disk, and 10^4 in halo), 25% of the total number of particles are expelled outside of our FOV in the first 2.5 Gyr (1/4 of the time of the simulation). Similar results are seen within all simulations of masses below $1 \times 10^{10} M_{\odot}$ and radii below 15 kpc (see Appendix C). These particles never returned over the course of the simulation. It should also be noted that particles fly out in all directions, which can be seen in Fig. 4.4.

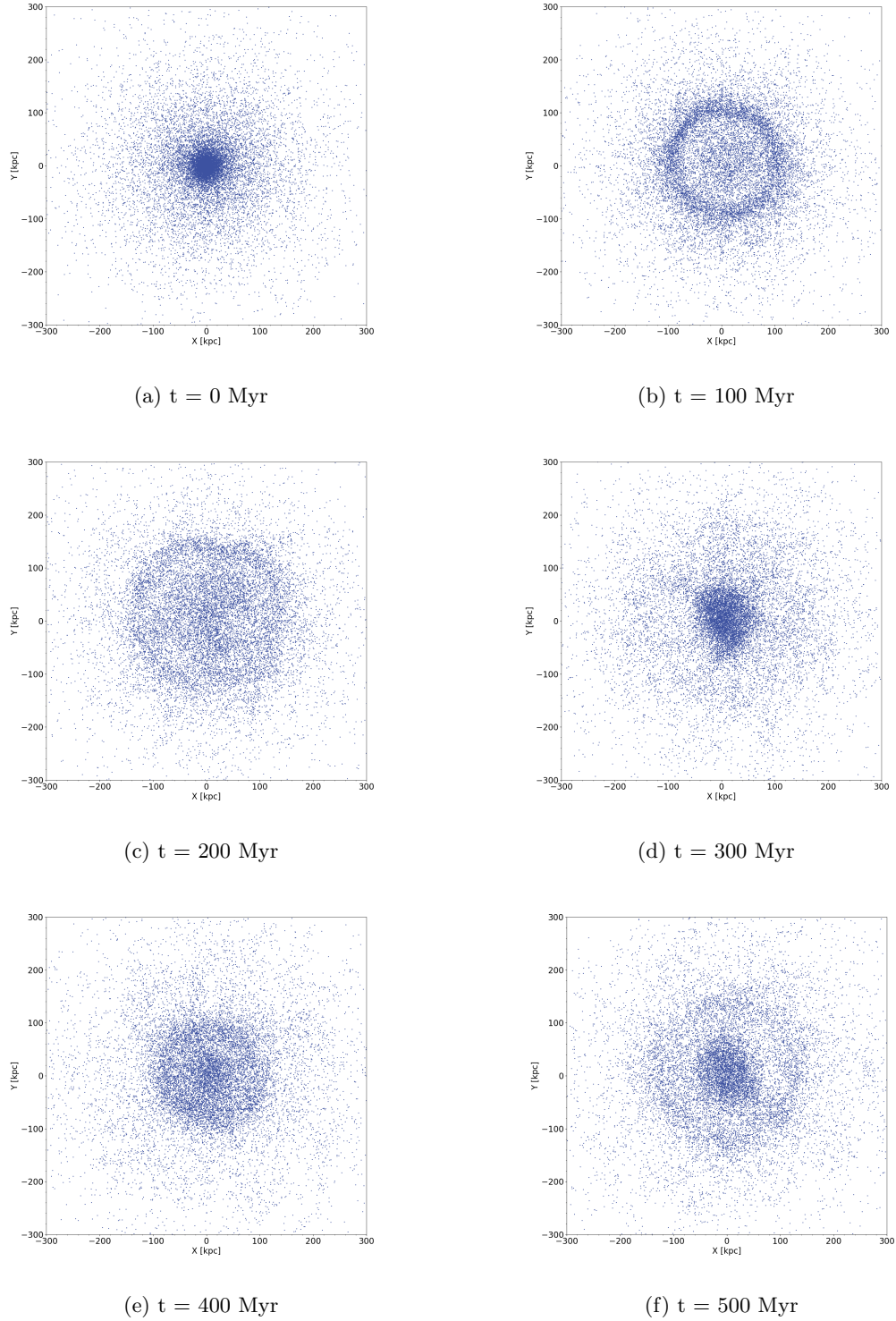


Figure 4.3: The repulsion force effects seen within the default simulation ($1 \times 10^{10} M_{\odot}$, a radius of 10 kpc, 10^3 particles in the bulge, 10^3 particles in the disk, and 2×10^4 particles in the halo) where the radius was relatively small (10 kpc). Between an iteration of 100 Myr, stars within the galaxy can be seen flying radially outward, which doesn’t make physical sense. The galaxy then seemingly recombines before exploding again.

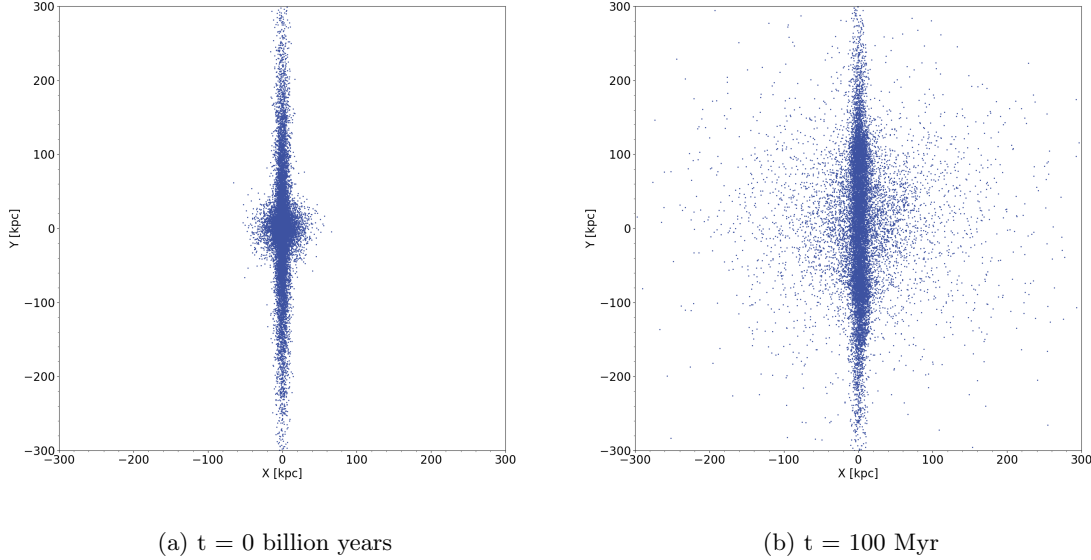


Figure 4.4: The repulsion force seems to force particles within the center of the galaxy to travel out radially in all directions. This is the default simulation viewed along the x-axis by 90° .

However, at higher mass and radii simulations, where the mass exceeded $1 \times 10^{10} M_\odot$ and radii above 15 kpc, minimal repulsion force effects were seen (even though the galaxy did not hold structure). In a simulation run with a mass of $1 \times 10^{10} M_\odot$ and a radius of 20 kpc, by the end of the simulation, 96% of all particles were still within the FOV of $300 \text{ kpc} \times 300 \text{ kpc}$.

Across all simulations, repulsion force effects were seen to expand the galaxy. No simulation was ever run with a radius beyond 50 kpc. Initially at $t = 0$, the galaxy is already beyond the specified radius within the OP. During this time, no dynamics are applied to the simulation, making the $t = 0$ frame act as the initial positions of all stars. However, during the next iteration of 100 Myr, dynamics are applied to the stars, and the center of the galaxy expands outwards in all simulations. The bulge averaged a diameter of 100 - 200 kpc for all simulations, and a 300×300 FOV was needed to include the galaxy. Further analysis should be conducted where the FOV of the simulation is limited to the size of the bulge (10 - 50 kpc) to see how many particles travel out of the specified

```

1 #The converter and new_galactics_model statements in the make_galaxy definition:
2     converter=nbody_system.nbody_to_si(M_galaxy, R_galaxy)
3     galaxy = new_galactics_model(n_halo,
4                                 converter,
5                                 #do_scale = True,
6                                 bulge_number_of_particles=n_bulge,
7                                 disk_number_of_particles=n_disk)
8
9 #The converter and GADGET-2 statement in the simulate definition:
10    converter = nbody_system.nbody_to_si(1.0e12|units.MSun, 100|units.kpc)
11    dynamics = Gadget2(converter, number_of_workers=8)
12    dynamics.parameters.epsilon_squared = (100 | units.parsec)**2

```

Figure 4.5: Both of the converter functions within the *make_galaxy* and *simulate* definitions

radius.

One possible explanation may lie within the dynamics of the simulation itself. Within the simulation, there are two packages responsible for stellar dynamics: the **new_galactics_model** package and the GADGET-2 package. Within the *make_galaxy* definition (as seen in Section 3.2), the **new_galactics_model** package is responsible for the interstellar dynamics of the simulation, including the bulge, when initially creating the galaxy. The source code for this package was not found within the GitHub repository, so more work must be done to understand and discover the source code of this package. As discussed in Section 2.3, GADGET-2 is responsible for the fluid-like dynamics (hydrodynamics) which control disk and spiral arm formation.

There are converter statements within both the *make_galaxy* and *simulate* definitions (Fig. 4.5). However, the *make_galaxy* uses values from the OP to define the parameters of the simulation. The converter in the *simulate* definition has preset values for conversion.

This means that when we create the galaxy, the interstellar dynamics are controlled by the mass and radius of the galaxy, which are set in the OP. These parameters are then used in the **new_galactics_model** package to simulate how the initial dynamics of stars function. However, once the galaxy has been created, the simulation runs through a new converter function, which has parameters independent from the OP. This new converter simulates the galaxy with a mass

of $1 \times 10^{12} M_{\odot}$, and simulates the galaxy at a radius of 100kpc. A smoothing parameter is run (`dynamics.epsilon.squared`) that increases the radius of the simulation until the entire galaxy has been simulated. Within the original script, the radius and smoothing parameter were set at 100 kpc and 100 pc respectively. However, as seen in Fig. 4.3, the repulsion effects can be seen on a scale smaller than 100kpc. Thus, the *simulation* definition had to be edited for a higher resolution. However, at smaller resolutions the simulation would crash, meaning that the solution was not as simple as reducing the parameters. The simulation would not account for the increase in resolution, meaning that it would hit the max time set before it reached `t_end`. Thus, a time-step for GADGET-2 was implemented to control the resolution of the simulation:

```

1 def simulate(galaxy, converter, n_bulge, n_halo, t_end, dt=25 | units.Myr):
2
3     converter = nbody_system.nbody_to_si(1.0e12|units.MSun, 1|units.kpc) #an AMUSE "
    Quantity" where numpy functions also work on quantities
4     dynamics = Gadget2(converter, number_of_workers=8) # represents fluids by means
    of smoothed particle hydrodynamics. number of workers represents the number of
    parallel workers: speed of simulation dependent on amount of RAM allocated
5     dynamics.parameters.epsilon_squared = (1 |units.kpc)**2 # epsilon is value of
    smoothing parameter squared
6
7     # max internal timestep for Gadget
8     dynamics.parameters.max_size_timestep = dt
9     # max possible evolve time for Gadget, 2**k *dt should be > tend. For 20 billion
    years, dt > 1
10    dynamics.parameters.time_max = 2**9*dt

```

The parameter `dt` controls the interval of time that GADGET-2 runs at. Currently, it's set for 25 Myr, which is a quarter of the time-step that the rest of the simulation runs at, giving us the necessary resolution to see the full effects of the repulsion force.

After implementing the GADGET-2 time-step, the full effects of the repulsion force can be seen in Fig. 4.3. Between 0 - 200 Myr, stellar particles seemingly fly out of the bulge and disk. By 300 Myr, the particles then reformed in a concentrated density (see Fig. 4.3 (d)), only to then be forced out again by 500 Myr. There seems to be a period function to the repulsion force, which only repeats itself once.

GADGET-2 has been used in many previous galactic simulations (Volker, 2005) without any

problems, suggesting that the **new_galactics_model** package is at fault. However, it's difficult to narrow the exact problem because we did not find the source code.

Chapter 5

Future Works

It has been shown that our simulation can produce some galaxies which exhibit fairly physical properties that are observed within real galaxies. The future of this project lies in making the simulation comparable to real observations, and creating the framework for mock observations.

5.1 Simulation Problems

First, the problem of `new_galactics_model` package must be resolved. This is the only code of the simulation that hasn't been fully understood, since we don't have the source code for the package.

Another problems is the scale of our galaxies. All of our plots are scaled at a distance of around 600 kpc so that they show most of the galaxy. However, we have set the radius of our galaxies to be at most 30 kpc, meaning that our galaxies are seemingly an order of magnitude larger than they should be. It is not clear why this occurs, but once again the `merge_two_galaxies` may be at fault. This simulation shares the same OP as our code, and the radius is set to 10 kpc for both galaxies. However, as seen in Fig. 3.5, both galaxies extend far beyond 10 kpc. This hints at either a conversion problem, a problem with how the OP functions, or the repulsion force problem.

5.2 Compare with Observed Galaxies

Next, we tackle the problem of comparing our simulations with observed galaxies. One element that could be added to the simulation, which would improve comparisons with observed galaxies is the addition of more complex structure such as a bar. As discussed in Section 2.1, many galaxies have bars within their structure. As of now, there is no easy way of adding a bar to the simulation, so no comparisons can be made with barred galaxies. However, there are documented simulations online which *do* include bars. However, many of these simulations implement the dynamics differently to how AMUSE does, making it difficult to transpose the code from one simulation to another. The addition of a bar would remove the limitations of which galaxies we can observe.

After we account for modifying our galaxy in order to accommodate observed galaxies, we would build the framework for creating mock observations with our simulation. In previously observed galaxies, integral field unit (IFU) spectroscopy has been the technique used to observe galaxies. To perform mock IFU spectroscopy, libraries of synthetic and standard spectra are implemented through MUSE and PHOENIX (Ivanov, Valentin D. et al., 2019; Husser, T.-O. et al., 2013). These libraries would allow us to assign a stellar spectrum to each particle. A mock observation would then integrate these spectra within a spatial pixel. By integrating these spectra along the line of sight for every spatially resolved pixel, we create a mock spectrum from the galaxy at every location. Our simulation should eventually be able to replicate IFU spectroscopy observations previously conducted by Batista et. al. (2017).

Finally, all observed galaxies have visual noise associated with galaxy surveys. Due to the limitations of physics, being able to identify specific stars outside our own galaxy is impossible. While great work has been done to reduce the amount of noise associated with observations (Seljak, Hamaus, & Desjacques, 2009), we must still account for this noise within our simulated galaxy. By applying noise to our simulation, and effectively blurring the edges of our particles, we can replicate

the noise that we see in observations. Gaussian noise has previously been used in other galaxy image simulations (Lanusse et al., 2021) and should suffice for our purposes.

Chapter 6

Conclusion

Simulations solve problems with observational studies of galaxies, notably galactic structure and line of sight problems. The AMUSE computational software was used due to its accessibility to robust and reliable dynamics as well as being able to include other sophisticated cosmological and galactic dynamics, such as GADGET-2. By adjusting the parameters of our simulation, we're able to create galactic simulations that resemble real galaxies while also being able to distinguish structure. Using a script developed by a past student within the program, modifications were made so that a galaxy could remain stable over the lifetime of the Universe. Two simulations were created that remained stable across the whole simulation: the $1 \times 10^8 M_{\odot}$ and $1 \times 10^{10} M_{\odot}$. The $1 \times 10^8 M_{\odot}$ is reminiscent of a dwarf galaxy, while the $1 \times 10^{10} M_{\odot}$ simulation has parameters that are similar to our own MW. However, it was discovered that a repulsion force was present within our simulation, which causes stellar particles to move outward in a non-physical way.

Appendix A

Milky Way Galaxy Simulation

A.1 Original Simulation Script

The code for the original script can be viewed on the next page.

AMUSE/Projects/Simulations/single_galaxy_2.py

```
import numpy
from matplotlib import pyplot
from amuse.lab import *
from amuse.ext.galactics_model import new_galactics_model
from prepare_figure import single_frame
from amuse.community.gadget2.interface import Gadget2

def make_plot(particles, filename):

    # close any open plots
    pyplot.close()
    x_label = "X [kpc]"
    y_label = "Y [kpc]"
    figure = single_frame(x_label, y_label, logy=False, xsize=14, ysize=14)
    pyplot.xlim(-300, 300)
    pyplot.ylim(-300, 300)

    pyplot.scatter(particles.x.value_in(units.kpc), particles.y.value_in(units.kpc),
                  alpha=1, s=5, lw=0)

    pyplot.savefig('Projects/Simulations/plots/' + filename)

def save_particle_positions(particles, filename):

    # save the positions in variables
    x_positions = particles.x.value_in(units.kpc)
    y_positions = particles.y.value_in(units.kpc)

    # rehsape them
    x_positions = x_positions.reshape((x_positions.size,1))
    y_positions = y_positions.reshape((y_positions.size,1))

    # stack them
    np_positions = numpy.hstack((x_positions, y_positions))

    # save this positions as .csv
    numpy.savetxt('Projects/Simulations/positions/' + filename, np_positions,
                 delimiter=',')

    return None

def make_galaxy(M_galaxy, R_galaxy, n_halo, n_bulge, n_disk):
    converter=nbody_system.nbody_to_si(M_galaxy, R_galaxy)
    galaxy = new_galactics_model(n_halo,
                                converter,
                                #do_scale = True,
                                bulge_number_of_particles=n_bulge,
                                disk_number_of_particles=n_disk)

    return galaxy, converter
```

```

def simulate(galaxy, converter, n_bulge, n_halo, t_end):

    converter = nbody_system.nbody_to_si(1.0e12|units.MSun, 100|units.kpc)
    dynamics = Gadget2(converter, number_of_workers=8)
    dynamics.parameters.epsilon_squared = (100 | units.parsec)**2
    set1 = dynamics.particles.add_particles(galaxy)
    dynamics.particles.move_to_center()

    disk = set1[:n_bulge] # This is just the thin and thick disks #:n_halo]
    bulge = set1[n_bulge:n_halo] # This is just the bulge
    disk_and_bulge = set1[:n_halo] # This is everything except dark matter halo

    # make a plot for each subset
    make_plot(disk, "disk_0myr")
    make_plot(bulge, "bulge_0myr")
    make_plot(disk_and_bulge, "galaxy_0myr")

    # save the initial positions
    save_particle_positions(disk, "disk_0myr.csv")
    save_particle_positions(bulge, "bulge_0myr.csv")
    save_particle_positions(disk_and_bulge, "galaxy_0myr.csv")

    sentinel = 1

    # use a while loop to simulate in parts, to get
    # snapshots of different parts of the simulation
    while sentinel*100 ≤ t_end.value_in(units.Myr):

        # evolve another 100 Myr
        dynamics.evolve_model(sentinel*100|units.Myr)

        # make and save the plots
        make_plot(disk, "disk_" + str(100*sentinel) + "myr")
        make_plot(bulge, "bulge_" + str(100*sentinel) + "myr")
        make_plot(disk_and_bulge, "galaxy_" + str(100*sentinel) + "myr")

        # save the initial positions
        save_particle_positions(disk, "disk_" + str(100*sentinel) + "myr.csv")
        save_particle_positions(bulge, "bulge_" + str(100*sentinel) + "myr.csv")
        save_particle_positions(disk_and_bulge, "galaxy_" + str(100*sentinel) + "myr.csv")

        # print to the terminal to see progress
        print('Done with t = ' + str(100*sentinel) + ' Myr')
        print()

        # add one to the sentinel
        sentinel += 1

    dynamics.stop()

def new_option_parser():
    from amuse.units.optparse import OptionParser
    result = OptionParser()
    result.add_option("-M", unit=units.MSun,
                     dest="M_galaxy", default = 1.0e12 | units.MSun,

```

```
        help="Galaxy mass [%default]")
result.add_option("-R", unit=units.kpc,
                  dest="R_galaxy", default = 10 | units.kpc,
                  help="Galaxy size [%default]")
result.add_option("--n_bulge", dest="n_bulge", default = 10000,
                  help="number of stars in the bulge [%default]")
result.add_option("--n_disk", dest="n_disk", default = 10000,
                  help="number of stars in the disk [%default]")
result.add_option("--n_halo", dest="n_halo", default = 20000,
                  help="number of stars in the halo [%default]")
result.add_option("--t_end", unit=units.Myr,
                  dest="t_end", default = 2000|units.Myr,
                  help="End of the simulation [%default]")

return result

if __name__ == '__main__':
    o, arguments = new_option_parser().parse_args()

    print()
    print('Creating galaxy')
    print()

    galaxy1, converter = make_galaxy(o.M_galaxy, o.R_galaxy,
                                     o.n_halo, o.n_bulge, o.n_disk)

    print('Done creating galaxy, starting simulation')
    print()

    simulate(galaxy1, converter, int(o.n_bulge), int(o.n_halo), o.t_end)
```

A.2 Modified Simulation Script

The code for the current iteration of the simulation script can be viewed on the next page.

AMUSE/Projects/Simulations/single_galaxy_2copy.py

```

import numpy
from matplotlib import pyplot
from amuse.lab import *
from amuse.ext.galactics_model import new_galactics_model
from prepare_figure import single_frame
from amuse.community.gadget2.interface import Gadget2
from amuse.datamodel import Particles

def make_plot(particles, filename):

    # close any open plots
    pyplot.close()
    x_label = "X [kpc]"
    y_label = "Y [kpc]"
    figure = single_frame(x_label, y_label, logy=False, xsize=14, ysize=14)
    pyplot.xlim(-300, 300)
    pyplot.ylim(-300, 300)

    pyplot.scatter(particles.x.value_in(units.kpc), particles.y.value_in(units.kpc),
                  alpha=1, s=5, lw=0)

    pyplot.savefig('AMUSE/Projects/Simulations/plots/single_galaxy_2copy/' + filename)

def save_particle_positions(particles, filename):

    # save the positions in variables
    x_positions = particles.x.value_in(units.kpc)
    y_positions = particles.y.value_in(units.kpc)

    # rehsape them
    x_positions = x_positions.reshape((x_positions.size,1)) # Reshape makes sure that
    value can be multiplied with vectors
    y_positions = y_positions.reshape((y_positions.size,1))

    # stack them
    np_positions = numpy.hstack((x_positions, y_positions)) #Stack arrays in sequence
    horizontally (column wise).

    # save this positions as .csv
    numpy.savetxt('AMUSE/Projects/Simulations/positions/single_galaxy_2copy/' + filename,
    np_positions, delimiter=',')

    return None

def make_galaxy(M_galaxy, R_galaxy, n_halo, n_bulge, n_disk): #M_galaxy, R_galaxy,
n_halo, n_bulge, n_disk are dummy variables
    converter=nbody_system.nbody_to_si(M_galaxy, R_galaxy) #converting from amuse units
    to si
    print(M_galaxy)
    print(R_galaxy)
    galaxy = new_galactics_model(n_halo, #creating new galaxy from imported model
    converter,

```

```

#do_scale = True,
bulge_number_of_particles=n_bulge,
disk_number_of_particles=n_disk)

#Change line-of-sight (rotate along x axis, rotate along y axis, rotate along the z
axis)
galaxy.rotate(0.0, 0.0, 0.0)

#galaxy.velocity += [0.0, 0.0, 0.0] | units.km/units.s
return galaxy, converter

def simulate(galaxy, converter, n_bulge, n_halo, t_end, dt=25 | units.Myr):
    converter = nbody_system.nbody_to_si(1.0e12|units.MSun, 1|units.kpc) #an AMUSE
"Quantity" where numpy functions also work on quantities
    dynamics = Gadget2(converter, number_of_workers=8) # represents fluids by means of
smoothed particle hydrodynamics. number of workers represents the number of parallel
workers: speed of simulation dependent on amount of RAM allocated
    dynamics.parameters.epsilon_squared = (1 |units.kpc)**2 # epsilon is value of
smoothing parameter squared

    # max internal timestep for Gadget
    dynamics.parameters.max_size_timestep = dt
    # max possible evolve time for Gadget, 2**k *dt should be > tend. For 20 billion
years, dt > 1
    dynamics.parameters.time_max = 2**9*dt

    set1 = dynamics.particles.add_particles(galaxy)# defines particles to set1
    dynamics.particles.move_to_center() # Shift positions and velocities of the particles
such that their center of mass
                                #(velocity) is centered on the origin.

    disk = set1[:n_bulge] # This is just the thin and thick disks #:n_halo]
    bulge = set1[n_bulge:n_halo] # This is just the bulge
    disk_and_bulge = set1[:n_halo] # This is everything except dark matter halo

    # make a plot for each subset
    make_plot(disk, "disk_0myr")
    make_plot(bulge, "bulge_0myr")
    make_plot(disk_and_bulge, "galaxy_0myr")

    # save the initial positions
    save_particle_positions(disk, "disk_0myr.csv")
    save_particle_positions(bulge, "bulge_0myr.csv")
    save_particle_positions(disk_and_bulge, "galaxy_0myr.csv")

    sentinel = 1 #value used to loop

    # use a while loop to simulate in parts, to get
    # snapshots of different parts of the simulation
    while sentinel*100 <= t_end.value_in(units.Myr):

        # evolve another 100 Myr
        dynamics.evolve_model(sentinel*100|units.Myr)

        # make and save the plots
        make_plot(disk, "disk_" + str(100*sentinel) + "myr")
        make_plot(bulge, "bulge_" + str(100*sentinel) + "myr")

```


A.3 Table of Simulations

A table of some simulations ran can be viewed on the next page

Simulation Description	Total Mass (Solar Masses)	Radius (kpc)	Particles in Bulge	Particles in Disk	% of Particles at $t = 2.5\text{Gyr}$	% of Particles $t = 5.0\text{ Gyr}$	% of Particles $t = 7.5\text{ Gyr}$	% of Particles $t = 10\text{ Gyr}$
The "Default" Simulation	1×10^{12}	10	10^4	10^4	75	72	69	67
Simulation Mass Increased	2×10^{15}	10	10^4	10^4	60	58	52	50
Simulation Mass Decreased	1×10^{10}	10	10^4	10^4	95	93	90	88
Stable Dwarf Galaxy	1×10^8	10	10^4	10^4	100	100	100	100
Simulation Mass Decreased Again	1×10^5	10	10^4	10^4	100	100	100	100
Increasing Radius:	1×10^{10}	20	10^4	10^4	99	98	97	96
Initial MW Distribution	1×10^{15}	50	9×10^3	1.3×10^4	97	97	96	95
Reducing the Mass and Radius for MW Distribution	1×10^{10}	30	9×10^3	1.3×10^4	100	100	100	100
Reducing Radius	1×10^{10}	20	9×10^3	1.3×10^4	99	98	96	94
Increased Mass and Radius	1×10^{12}	30	9×10^3	1.3×10^4	97	98	97	96
Reduce Radius	1×10^{12}	20	9×10^3	1.3×10^4	97	97	96	96

Bibliography

- Adams, F. C., Graff, D. S., & Richstone, D. O. (2003). Formation of Supermassive Black Holes in Galactic Bulges: A Rotating Collapse Model Consistent with the $M_{BH} - \sigma$ Relation. *The Astrophysical Journal*, 591(1).
- Aller, M. C., & Richstone, D. O. (2007, August). Host Galaxy Bulge Predictors of Supermassive Black Hole Mass. , 665(1), 120-156. doi: 10.1086/519298
- Aoyama, S., hou, K.-C., Hirashita, H., Nagamine, K., & Shimizu, I. (2018). Cosmological simulation with dust formation and destruction. *Monthly Notices of the Royal Astronomical Society*, 478(4).
- Batiste, M., Bentz, M. C., Manne-Nicholas, E. R., Onken, C. A., & Bershady, M. (2017). The BRAVE Program. I. Improved Bulge Stellar Velocity Dispersion Estimates for a Sample of Active Galaxies. *The Astrophysical Journal*, 835(271).
- Batiste, M., Bentz, M. C., Raimundo, S. I., & Vestergaard, C. A., Marianne abd Onken. (2017). Recalibration of the $M_{BH}-\sigma_*$ Relation for AGN. *The Astrophysical Journal*, 838(L10).
- Beifiori, A., Courteau, S., M., C. E., & Y., Z. (2012). On the correlations between galaxy properties and supermassive black hole mass. *Monthly Notices of the Royal Astronomical Society*, 419(3).
- Burkert, A., & Silk, J. (2001). Star Formation-Regulated Growth of Black Holes in Protogalactic Spheroids. *The Astrophysical Journal*, 554(2).

- Cappellari, M., Scott, N., Alatalo, K., Blitz, L., Bois, M., Bournaud, F., ... Young, L. M. (2013, 05). The ATLAS3D project – XV. Benchmark for early-type galaxies scaling relations from 260 dynamical models: mass-to-light ratio, dark matter, Fundamental Plane and Mass Plane. *Monthly Notices of the Royal Astronomical Society*, *432*(3), 1709-1741. Retrieved from <https://doi.org/10.1093/mnras/stt562> doi: 10.1093/mnras/stt562
- Crain, R. A., Schaye, J., Bower, R. G., Furlong, M., Schaller, M., Theuns, T., ... Trayford, J. W. (2015). The EAGLE simulations of galaxy formation: calibration of subgrid physics and model variations. *Monthly Notices of the Royal Astronomical Society*, *450*(2).
- Dullo, B. T., Bouquin, A. Y. K., de Paz, A. G., Knapen, J. H., & Gorgas, J. (2020, jul). The black hole mass-color relations for early- and late-type galaxies: Red and blue sequences. *The Astrophysical Journal*, *898*(1), 83. Retrieved from <https://dx.doi.org/10.3847/1538-4357/ab9dff> doi: 10.3847/1538-4357/ab9dff
- Ferrarese, L., & Merritt, D. (2000). A Fundamental Relation between Supermassive Black Holes and Their Host Galaxies. *The Astrophysical Journal*, *539*(L9).
- Gadotti, D. A. (2009, 02). Structural properties of pseudo-bulges, classical bulges and elliptical galaxies: a Sloan Digital Sky Survey perspective. *Monthly Notices of the Royal Astronomical Society*, *393*(4), 1531-1552. Retrieved from <https://doi.org/10.1111/j.1365-2966.2008.14257.x> doi: 10.1111/j.1365-2966.2008.14257.x
- Gebhardt, K., Bender, R., Bower, G., Dressler, A., Faber, S. M., Filippenko, A. V., ... Kormendy, J. (2000). A Relationship between Nuclear Black Hole Mass and Galaxy Velocity Dispersion. *The Astrophysical Journal*, *539*(L13).
- Gültekin, K., Richstone, D. O., Gebhardt, K., Lauer, T. R., Tremaine, S., Aller, M. C., ... Siopis, C. (2009, June). The M- σ and M-L Relations in Galactic Bulges, and Determinations of Their Intrinsic Scatter. , *698*(1), 198-221. doi: 10.1088/0004-637X/698/1/198

- Gültekin, K., Richstone, D. O., Gebhardt, K., Lauer, T. R., Tremaine, S., Aller, M. C., ... Filippenko, A. V. (2009). THE M- σ AND M-L RELATIONS IN GALACTIC BULGES, AND DETERMINATIONS OF THEIR INTRINSIC SCATTER. *The Astrophysical Journal*, 698(198).
- Habing, H. J., Olton, F. M., Chester, T., Gillett, F., & Rowan-Robinson, M. (1985, November). Stars in the bulge of our Galaxy detected by IRAS. , 152(1), L1-L4.
- Husser, T.-O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., & Hauschildt, P. H. (2013). A new extensive library of phoenix stellar atmospheres and synthetic spectra. *A&A*, 553, A6. Retrieved from <https://doi.org/10.1051/0004-6361/201219058> doi: 10.1051/0004-6361/201219058
- Ivanov, Valentin D., Coccatto, Lodovico, Neeser, Mark J., Selman, Fernando, Pizzella, Alessandro, Dalla Bontà, Elena, ... Morelli, Lorenzo (2019). Muse library of stellar spectra. *A&A*, 629, A100. Retrieved from <https://doi.org/10.1051/0004-6361/201936178> doi: 10.1051/0004-6361/201936178
- Kormendy, J., & Richstone, D. (1995). Inward Bound—The Search For Supermassive Black Holes In Galactic Nuclei. *Annual Review of Astronomy and Astrophysics*, 33(581).
- Lanusse, F., Mandelbaum, R., Ravanbakhsh, S., Li, C.-L., Freeman, P., & Póczos, B. (2021, 05). Deep generative models for galaxy image simulations. *Monthly Notices of the Royal Astronomical Society*, 504(4), 5543-5555. Retrieved from <https://doi.org/10.1093/mnras/stab1214> doi: 10.1093/mnras/stab1214
- Matsuda, T., Fujimoto, Y., Shima, E., Sawada, K., & Inaguchi, T. (1989). Numerical Simulations of Interaction between Stellar Wind and Interstellar Medium. *Progress of Theoretical Physics*, 81(4).
- Pelupessy, F., Elteren, A. v., Vries, N. d., McMillan, S. L. W., Drost, N., & Zwart, S. F. (2013). The Astrophysical Multipurpose Software Environment. *Astronomy Astrophysics (AA)*,

557(A84).

Peterson, B. M. (2013). *The Physics of Accretion onto Black Holes*. SpringerLink.

Ramsden, P., Lanning, D., Nicholl, M., & McGee, S. L. (2022, jun). The bulge masses of TDE host galaxies and their scaling with black hole mass. *Monthly Notices of the Royal Astronomical Society*, 515(1), 1146–1157. Retrieved from

Seljak, U. c. v., Hamaus, N., & Desjacques, V. (2009, Aug). How to Suppress the Shot Noise in Galaxy Surveys. *Phys. Rev. Lett.*, 103, 091303. Retrieved from <https://link.aps.org/doi/10.1103/PhysRevLett.103.091303> doi: 10.1103/PhysRevLett.103.091303

Shankar, F., Bernardi, M., Sheth, R. K., Ferrarese, L., Graham, A. W., Savorgnan, G., ... Lapi, A. (2016). Selection bias in dynamically measured supermassive black hole samples: its consequences and the quest for the most fundamental relation. *Monthly Notices of the Royal Astronomical Society*, 460(3).

Sijacki, D., Vogelsberger, M., Genel, S., Springel, V., Torrey, P., Snyder, G. F., ... Hernquist, L. (2015). The Illustris simulation: the evolving population of black holes across cosmic time. *Monthly Notices of the Royal Astronomical Society*, 452(1).

Spiridon, V.-L., & Slusanschi, E.-I. (2013). N-Body Simulations with GADGET-2. IEEE.

Stahler, S. W., & Palla, F. (2004). *The Formation of Stars*. WileyVHC.

Tempel, E., & Tenjes, P. (2006, 08). Line-of-sight velocity dispersions and a mass-distribution model of the Sa galaxy NGC 4594. *Monthly Notices of the Royal Astronomical Society*, 371(3), 1269-1279. Retrieved from <https://doi.org/10.1111/j.1365-2966.2006.10741.x> doi: 10.1111/j.1365-2966.2006.10741.x

Volker, S. (2005). The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364(4).

Zwart, S. P., & McMillan, S. (2018). *Astrophysical Recipes: The art of AMUSE*. The American Astronomical Society.