CONSTRAINING DWARF GALAXY PROPERTIES USING TIDAL STREAMS

Siddhartha Shelton

Submitted in Partial Fullfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:
Dr. Heidi Jo Newberg
Dr. Ethan Brown
Dr. Vincent Meunier
Dr. Malik Magdon-Ismail

Department of Physics
Rensselaer Polytechnic Institute
Troy, New York

[December 2018]
Submitted December 2018

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would first and foremost like to thank my advisor and mentor, Dr. Heidi Jo Newberg. She has provide invaluable guidance in shaping me into a successful researcher and physicist. She accepted me into her group even though I had very limited astronomy and computational experience, yet never wavered in her belief that I would prosper as a graduate student. Her intellect and command of her field has been a driving influence on my life and graduate experience. I will forever be grateful for her extraordinary patience and kindness.

I would also like to thank my committee members, Dr. Ethan Brown, Dr. Vincent Meunier, Dr. Malik Magdon-Ismail. Dr. Brown is a leader in his field, and his knowledge and experience about dark matter have provided very valuable insight into my own research. Dr. Meunier was my Professor in Computational Physics and Advanced Computational Physics which were critical in my ability to learn the tools I needed to be successful as a computational researcher. Dr Magdon-Ismail has been a long time collaborator for the MilkyWay@home project, and has provided numerous contributions to its success.

I would like to thank the love of my life, Alyssa Leung, my girlfriend and partner in crime. Her love and support have kept me going during the most stressful of times. I would like to thank my family: my Mother, Lily Shelton, for her unwaivering support and belief that I could do anything, and her countless sacrifices to ensure I would have the opportunity to try; My sisters Saina Shelton and Selma Stoddard, who have always been there for me. Thank you for always having an ear open and a drink ready. I would not be the person I am today without you all.

Thank you to all of my Starlab friends and colleagues: Jake Weiss, Paul Amy, Charles Martin, Clayton Rayment, Thomas Donlon II, Jeffery Thompson, Keighley Rockcliffe, and many others. Not only have you been amazingly helpful as colleagues and collaborators, but through our discussions and general buffoonery, have provided much needed levity from the day to day stress and demands of graduate school and research. Also, thank you for putting up with my terrible jokes! Thank you to my roommate Michael Agiorgousis, you have always been a good friend, whether it was giving advice, or listening to me complain.

I would like to thank my long time friends who have always stuck by my side. It can be many months between seeing each other, but it always somehow feels like we haven't missed

a beat.

I would like to thank the more than 200,000 MilkyWay@home volunteers for their countless hours of donated computing time, their constant community involvement with debugging and development, and their financial support over the past decade.

I would also like to thank and acknowledge contributions made by The Marvin Clan, Babette Josephs, Manit Limlamai, and the 2015 Crowd Funding Campaign to Support Milky Way Research. This publication is based on work supported by the National Science Foundation under grant No. AST 16-15688 and the NASA/NY Space Grant fellowship.

I use data from the Sloan Digital Sky Survey. Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS Web Site is `http://www.sdss.org/`.

The SDSS is managed by the Astrophysical Research Consortium for the Participating Institutions. The Participating Institutions are the American Museum of Natural History, Astrophysical Institute Potsdam, University of Basel, University of Cambridge, Case Western Reserve University, University of Chicago, Drexel University, Fermilab, the Institute for Advanced Study, the Japan Participation Group, Johns Hopkins University, the Joint Institute for Nuclear Astrophysics, the Kavli Institute for Particle Astrophysics and Cosmology, the Korean Scientist Group, the Chinese Academy of Sciences (LAMOST), Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Ohio State University, University of Pittsburgh, University of Portsmouth, Princeton University, the United States Naval Observatory, and the University of Washington.

ABSTRACT

We have developed a method for estimating the properties of the progenitor dwarf galaxy from the tidal stream of stars that were ripped from it as it fell into the Milky Way. In particular, we show that the mass and radial profile of the progenitor dwarf galaxy of the Orphan Stream, including the stellar and dark matter components, can be reconstructed from the distribution of stars in the tidal tail it produced. This method can be used to estimate the dark matter content in dwarf galaxies without the assumption of virial equilibrium that is required to estimate the mass using line-of-sight velocities.

We use N-body simulations to create a two component dwarf galaxy, initially in virial equilibrium, and place it in orbit around a published static model of the Milky Way Galaxy. The distribution of stars in the resulting tidal debris stream is compared to the actual distribution of stars along the Orphan Stream, using a comparison metric specifically designed for this problem. For purposes of testing the algorithm, we simulate the "actual" distribution of stars along the Orphan Stream using reasonable dwarf galaxy parameters and a different random seed, so that we know whether our algorithm is able to reconstruct the dwarf galaxy given only observable information about the resulting tidal stream. The algorithm fits the dark matter mass, dark matter radius, stellar mass, radial profile of stars, and orbital time. Our simulations assumed that the Milky Way potential, dwarf galaxy orbit, and the form of the density model for the dwarf galaxy are known exactly.

We use MilkyWay@home, a 0.8 PetaFLOPS distributed supercomputer, to optimize our dwarf galaxy parameters until we arrive at a best-fit to the Orphan Stream data. We show that we have been able to recover the parameters used in the creation of a simulated tidal debris stream for both the baryonic and dark matter components. This is accomplished even when the dark matter component extends well past the half light radius of the dwarf galaxy progenitor, proving that we are able to extract information about the dark matter halos of dwarf galaxies.

CHAPTER 1
Introduction

1.1 Dark Matter in Dwarf Galaxies

The first concrete evidence of dark matter came from Vera Rubin's study of galactic rotation curves. Instead of finding the expected Keplerian decline in the rotation speeds of stars far from the center of the host galaxy, she observed that the rotation curves were fairly flat. This indicated that the stars being observed were embedded in an extended halo of unseen matter. Her research showed that large amounts of this 'dark matter' was needed to keep spiral galaxies from flying apart due to their high rotation speeds. Further evidence came from gravitational lensing measurements, which looks at the the way light is bent around large concentrations of mass. Gravitational lensing measurements of galaxy mergers such as the Bullet Cluster, among others, show that the majority of the mass is not in the areas inhabited by stellar mass (Paraficz et al., 2016; Bradač et al., 2008). It is now known that dark matter accounts for the majority of the mass in the Universe.

Since the discovery of dark matter, scientists have searched for the elusive constituent particles. These searches generally fall under two categories: direct and indirect searches. Direct dark matter detection searches look for any possible interactions between theorized dark matter candidates and ordinary matter. Such experiments include The Large Underground Xenon experiment (LUX), XENON100, and XENON1T (Akerib et al., 2017; Xenon100 Collaboration et al., 2012; Aprile et al., 2016), which look for nuclear recoil interactions with xenon atoms caused by the popular dark matter candidate, the Weakly Interacting Massive Particle (WIMP); and DM-Ice, which uses sodium iodide crystals to look for an annual modulation of WIMP signals caused by the Earth moving with respect to the Galaxy (Jo & DM-Ice Collaboration, 2016), among others.

Indirect detection searches look for the byproducts of interactions between dark matter and ordinary matter, such as the products of annihilation or decay. One possible product of decay or self-annihilation is neutrinos. Excess neutrino production is expected to occur in areas where dark matter may exist with a high density, and thus may readily interact with

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, "Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing," ApJ.

itself. These areas include the center of the Sun, the Galactic center, the Galactic halo, and dwarf galaxies (Aartsen et al., 2015). One of the goals of the The IceCube Neutrino Observatory is to find excess muon neutrino production from the Galactic center (Aartsen et al., 2015). Another common search is for excess gamma rays, also theorized to be a product of dark matter interactions and annihilation. A goal of The Fermi Gamma-ray Space Telescope (GLAST) is to look for excess gamma rays from the Galactic center. In fact, such a signal was detected, and is known as the Galactic center extended source (Hooper & Goodenough, 2011; Abazajian et al., 2014). However, many attribute the excess gamma rays to a large population of millisecond pulsars (Arca-Sedda et al., 2018), cosmic ray bursts (Cholis et al., 2015), young gamma-ray pulsars (O'Leary et al., 2015), or the stellar population of the central Galactic bulge (Macias et al., 2018).

As we have mentioned, prime targets for dark matter indirect detection experiments include dwarf galaxies. The Milky Way halo is made up of a large number of dwarf galaxies, globular clusters, tidal streams, and other substructures (Newberg & Carlin (2016), p. 87). Dwarf galaxies in particular are thought to have large amounts of dark matter while containing few sources of background gamma rays. Therefore, gamma ray production attributed to dark matter interactions may be easily observable. Some dwarf galaxies are so-called ultrafaint dwarf galaxies, which have the largest mass to light ratio, the ratio of the estimated total mass to the luminous mass, of any known structures. This makes them especially prime targets for indirect dark matter detection experiments, and have been used in recent years as probes to place limits on dark matter particle properties (Bartels et al., 2016; Lee et al., 2016).

Ultrafaint dwarf galaxies are also important to cosmological model simulations, which simulate the formation of the Universe. These simulations often have trouble reconciling their predictions with observations at small scales, known as the 'small scale' problems (Bull et al., 2016). For example, these simulations predict a universal radial distribution that is centrally peaked, or cuspy, where observations tend to find shallower, or cored, distributions (Bull et al., 2016). Furthermore, if dwarf galaxies are found to have very little or no dark matter at all then they may be reclassified as tidal dwarfs, structures that were 'kicked out' by galaxy mergers. This would explain why many dwarf galaxies are found to orbit in planes around both the Milky Way and Andromeda, providing a possible solution to the satellite planes problem Bull et al. (2016). This would, in turn, worsen the missing satellite problem

– the overabundance of predicted Milky Way satellites by these simulations as compared with the number observed (Bull et al., 2016). Therefore, it is equally vital knowing both the mass content and radial distribution of ultrafaint dwarf galaxies.

Estimates of the total mass of ultrafaint galaxies have been shown many times to be greatly overestimated. The total mass of these galaxies are often determined using measures of their velocity dispersion, and assuming that they are in equilibrium. If only the line-of-sight dispersion is known, then the mass estimates can be inaccurate (Gilmore et al., 2007; Strigari et al., 2007). Masses can also be derived by radial velocities (Wolf et al., 2010). However, the usefulness of this method is limited to cases where mass follows light. The velocity dispersion measures only the mass within the radius where the stars are measured, and thus depends on the radial distribution of the dark matter. Mass estimates from line-of-sight velocities will systematically overestimate the total mass of a galaxy if it is in the process of tidally disrupting. If a dwarf galaxy is undergoing tidal disruption, it is inherently not in virial equilibrium, and thus assuming equilibrium can lead to erroneous mass estimates. Furthermore, measures of the velocity dispersions follow the assumption that spectroscopically studied member stars are not part of binary systems. Such binary stars can artificially inflate the measured velocity dispersion of dwarf galaxies because they can be misinterpreted as single stars (McConnachie & Côté, 2010). When this occurs, the entire orbital velocity of the star, instead of just a component, is attributed to its rotation around the host galaxy, indicating a much higher mass enclosed within its orbit. Also, studies have shown that the measured velocity dispersions of objects that are almost fully disrupted, or close to apogalacticon, could be an order of magnitude or higher than equilibrium values (Smith et al., 2013; Blaña et al., 2015). This could be due to stars that have been tidally stripped from the progenitor but remain close enough to the remnant core to be mistaken as bound stars by observers.

There have been several cases where the estimated masses of ultrafaint galaxies were shown to be inaccurate because the target galaxy is in the process of tidally disrupting, or stars used to measure the velocity dispersion were found to be part of a binary system. Segue 1, initially identified as a globular cluster (Belokurov et al., 2007), was originally found to have an estimated mass to light ratio of 1000:1 (Geha et al., 2009). However, its stars were found to have similar properties in kinematics and photometry as the stars of the Sagittarius dwarf tidal stream, making it likely that it was a dwarf galaxy satellite of the Sagittarius dwarf galaxy, and in the process of tidally disrupting (Niederste-Ostholt et al., 2009). The claim of a high mass-to-light ratio has been called into questions by Domínguez et al. (2016), who found a possible orbit for Segue 1 in which a satellite with no dark matter at all will produce the observed line-of-sight velocity dispersion.

Similarly, Triangulum II was estimated to be the most dark matter dominated galaxy known (Kirby et al., 2015), but a star used in the velocity dispersion calculation was later found to be in a binary system, greatly reducing the mass estimate and changing its classification to a star cluster or tidally stripped dwarf galaxy (Kirby et al., 2017). Further studies by (Martin et al., 2008) showed that ultrafaint galaxies are elongated, suggesting tidal disruption as a likely explanation. These cases illustrate the need for a new method for determining the dark matter content of dwarf galaxies without making any of the assumptions discussed above.

We present an algorithm that can measure the dark matter content of dwarf galaxies without the need to assume equilibrium, and is independent of the presence of binary stars. The algorithm uses N-body simulations to model current stellar density distributions of a tidal stream, recovering parameters of the progenitor that inform us of the matter composition and radial distribution of both baryonic and dark matter. This method requires only (l,b) sky coordinate data of tracer stars from the tidal stream. As we will show, we have been able to recover parameters used to create simulated tidal streams. This includes the remarkable ability to recover the dark matter properties outside of the half-light radius.

1.2 Overview

We developed a method to probe the mass and radial extent of the dark matter content of dwarf galaxies. Instead of assuming equilibrium conditions in these galaxies, we look at dwarf galaxies we know to be in the process of tidal disruption. These galaxies are currently in the process of merging with the Milky Way galaxy, producing tidal streams of stars. Tidal streams show promise for being very useful as probes of the Milky Way potential, because we know that all of the stars in the stream were once located within the same progenitor. Therefore, using tracer stars along the stream can help characterize the orbits they have traversed.

Using a previously published Milky Way Galaxy model whose form and parameters lead to a best fit orbit to the Orphan Stream, we optimize the parameters of the dwarf galaxy progenitor until we create a simulated tidal stream similar to that of the Orphan Stream. This is done using N-body simulations to constrain the dark matter content within dwarf galaxies by comparing a simulated tidal stream distribution to data for a particular tidal stream. The simulated tidal stream is created by first generating a dwarf galaxy in equilibrium and then evolving it through an orbit around the Milky Way. Although we plan to relax these limitations in the future, we currently assume a fixed Milky Way potential, and a dwarf galaxy orbit that was fit to the position and line-of-sight velocities along the tidal stream. The simulated progenitor dwarf galaxy is generated with two separate Plummer model density profiles for the stars and the dark matter and comprises of N self-interacting bodies. The dwarf galaxy is then evolved through a fixed Milky Way potential and the resulting simulated tidal stream is compared with an actual tidal stream of stars in the Milky Way halo. The properties of the dwarf galaxy from which the tidal stream formed are determined by varying the simulation parameters until the simulated and observed streams match.

We have chosen the Orphan Stream because estimates of the luminosity of its progenitor galaxy indicate that it may have been an ultrafaint dwarf galaxy (Belokurov et al., 2007; Sales et al., 2008; Grillmair et al., 2015), making it an ideal choice. Furthermore, we have access to stellar data, the Milky Way model, and orbit from Newberg et al. (2010). Together, using the Orphan Stream allows us to begin with an idealized case for optimizing the dwarf galaxy parameters, in which the Milky Way galaxy potential and dwarf galaxy orbit are known exactly.

The theoretical model used for our dwarf galaxies is comprised of two components, one to represent stars and the other to represent dark matter. When the model is turned into a simulated dwarf galaxy comprised of bodies, the bodies of each component are allowed to interact with each other and with the fixed Milky Way potential gravitationally.

At the end of the simulation we create a histogram of the simulated stellar density as a function of angle along the orbit that will be compared with the observed density distribution of stars. Since dark matter cannot be directly observed, it cannot be compared directly with observations. However, the ‘unseen’ dark matter component influences the evolution of the visible matter and the formation of the tidal debris stream. Therefore, properties of the dark matter is inherently encoded in the evolution of the baryonic matter.

Using stellar stream data, a similar histogram can be made to represent the actual tidal stream. These histograms can then be compared to determine the 'likelihood' that they are the same. We have developed a sophisticated comparison metric which measures how good a match the simulated histogram is to the data. We optimize the orbital time and parameters in the dwarf galaxy model to obtain the most similar histograms. The parameters used to create the simulated dwarf galaxy can be optimized to maximize the likelihood measure. This optimization is done using a Differential Evolution algorithm, run on a powerful distributed supercomputer, MilkyWay@home.

In this thesis, we will begin with an overview of our simulations, describing the general setting. We then give a brief description of the previously published Milky Way galaxy model, and progenitor dwarf galaxy orbit used. We follow this with a theoretical break down of the dwarf galaxy model, how we implemented this theoretical framework in generating a simulated dwarf galaxy, and the tests we performed on our simulated dwarf galaxy implementation. Next, we describe the other various parameters needed to make the simulations work. We then move onto the comparison algorithm, with a description of how we create the histograms from simulated data, and each of the three part metric used for comparison, how they work and their role in the over all comparison likelihood. We conclude this chapter with how this is used in practice. We then describe the tidal stream under investigation, the estimations we use, and how we convert stellar data into a form usable in our optimizations. Finally, we present the results from our optimizations, followed by our concluding remarks and plans for future work. In the appendices are detailed descriptions of the parameter files used for the N-body code package and instructions for usage, several derivations behind equations presented throughout this text, descriptions of previous methods, and results from using the velocity dispersion instead of stream width to constrain the mass of the progenitor satellite.

CHAPTER 2

N-body Simulations - External Potential and Progenitor Orbit

As mentioned in the introduction, we perform N-body simulations in order to constrain the dark matter content of dwarf galaxies. We describe here what an N-body simulation is, and the setting in which our simulations are performed. In describing the setting, we will give a complete description of our Milky Way model, our choice of dwarf galaxy orbit and the input parameters for our simulation.

In a general sense, an N-body simulation is a method for finding the numerical solution to a many-body problem. The objects here are given some initial set of positions and velocities in standard Cartesian space. The simulation then calculates the final positions and velocities of each object. The objects can interact with each other and with their surroundings. In this way, the 'simulation' is the numerical solution to the integrals of motion for these objects. In our case, the N objects are used to model a dwarf galaxy, and interact with each other and with their environment gravitationally, adhering to Newtonian physics. The environment is a large central gravitational potential well, representing the Milky Way Galaxy. The environment is also made up of the much smaller potential wells of the other objects. Running the simulation involves placing the dwarf galaxy in orbit around the Milky Way potential. From there, the interactions between each body and its environment are calculated and integrated for some set amount of simulation time. Running the simulation is the subject of an upcoming chapter.

It is also possible to represent the Milky Way Galaxy using bodies as other authors have done (Chakrabarti & Blitz, 2009; de la Vega et al., 2015). This would require a large number of bodies, possibly in the millions, consuming a large amount of computational power for every simulation. Since we are performing an optimization, which require thousands of simulations, representing the Milky Way with bodies is not computationally feasible. We therefore choose a static potential model for the Milky Way Galaxy, which has a fixed analytic form.

The dwarf galaxy model requires several input parameters to determine the mass and

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, "Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing," ApJ.

radial distribution of the baryonic (stellar) and dark matter components: the total mass and a scale radius of each component. For the baryonic component, we specify the mass (M_B) and scale length (R_B). We also specify two ratio parameters used to determine the mass (M_D) and scale length (R_D) of the dark matter component. Since these input parameters are altered by MilkyWay@home's search algorithm during an optimization, ratio values are used to determine the dark matter component properties in order to explore a larger range of mass and scale length values with a narrower search range. The radius ratio (ξ_R) and mass ratio (ξ_M) values are related to the dark matter parameters by:

$$\begin{aligned} R_D &= \frac{R_B}{\xi_R}(1-\xi_R), \\ M_D &= \frac{M_B}{\xi_M}(1-\xi_M). \end{aligned} \tag{2.1}$$

In terms of the baryon and dark matter parameters, the ratios can also be written:

$$\begin{aligned} \xi_R &= \frac{R_B}{R_B+R_D}, \\ \xi_M &= \frac{M_B}{M_B+M_D}. \end{aligned} \tag{2.2}$$

In order to represent the Milky Way Galaxy, we use a theoretical model comprised of three components representing the Milky Way's central bulge, disk and halo. Similar to the dwarf galaxy, each of the three components requires several input parameters that determine the mass and shape of the respective component. Furthermore, the orbit on which the dwarf galaxy is placed is determined by a set of position and velocity coordinates that, along with the Milky Way potential, fully characterize the orbit. We could simultaneously constrain the dwarf galaxy, the Milky Way potential model, and the orbital parameters. However, for now, we choose to use previously published parameters for the Milky Way model and orbit.

In the future when we adapt this algorithm to fit the Milky Way and orbit parameters, we will need to fit additional data constraints such as line-of-sight velocity and stream position, and include multiple tidal streams. These values are set in the parameter file, which will be described in a later chapter, meaning these values can also easily be set as input parameters to be altered in the optimization.

The Milky Way model parameters were chosen to be the same as those used to fit the orbit for the Orphan Stream in Newberg et al. (2010). We also adopt the corresponding best fit orbital parameters given by Model 5 of Table 3 in Newberg et al. (2010). Therefore, the only parameters being constrained are those of the dwarf galaxy and the time of evolution along the orbit.

2.1 Galactic Potential

As stated above, the Milky Way Galaxy is represented using a static potential. Specifically, the model chosen to represent the Milky Way is comprised of three components. Each component is meant to represent a different part of the Milky Way, some of which are shown in Figure 2.1.

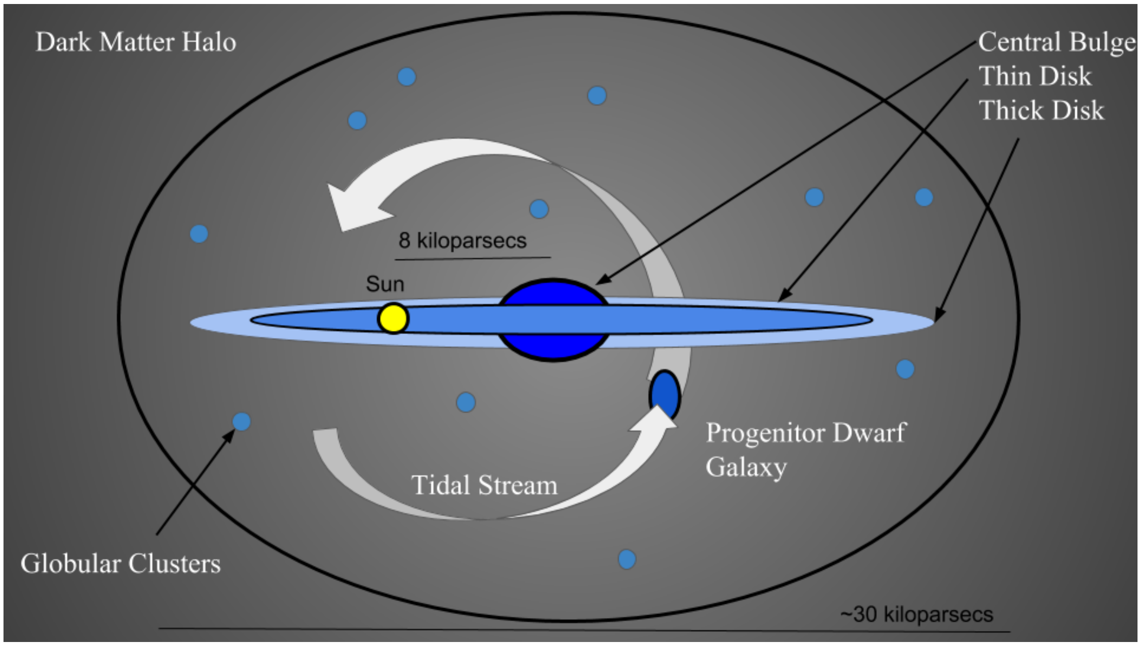

Figure 2.1: A basic schematic of the Milky Way Galaxy highlighting the main components adapted from Figure 1.1 of Newby (2013). It includes the central bulge, the thin and thick disks, and the relative position of the sun. It also points out the dark matter halo which surrounds the galaxy, an example of a tidal stream (not to scale) and the many embedded globular clusters present in the halo.

The model used to represent the Milky Way does not have this many details, however. The first component represents the bulge; the second component represents the entire disk, combining together the thin and thick disk in one model. The final component represents the dark matter halo using a smooth potential. The many embedded globular clusters are not individually represented, but such representation is possible. This will be further discussed in Section 9.1.

To describe the bulge, a spherically symmetric Hernquist model is used (Hernquist, 1990). This potential is given by:

$$\Phi_{bulge}(r) = -\frac{GM_{bulge}}{r + d}, \tag{2.3}$$

where d = 0.7 kpc is a scale length, and the mass $M_{bulge} = 1.0 \times 10^{11}\ M_{\odot}$(Law et al., 2005). As previously stated, the thin and thick disk are represented by a single potential. However, the disk is not spherically symmetric. Therefore, a cylindrical potential, the Miyamoto-Nagai potential, was chosen to represent the disk. This is given by:

$$\Phi_{disk}(r) = -\frac{GM_{disk}}{\sqrt{R^2 + (b + \sqrt{z^2 + c^2})^2}}, \tag{2.4}$$

where b = 6.5 kpc and c = 0.26 kpc are scale parameters, R is the cylindrical radius ($R = x^2 + y^2$), and the mass $M_{disk} = 3.4 \times 10^{10}\ M_{\odot}$ (Law et al., 2005).

The exact mass of the dark matter halo is not known. Therefore, a potential model with a parameter representing the mass is required. The logarithmic halo potential is such a model, given by:

$$\Phi_{halo}(r) = v_0^2 \ln\left(1 + \frac{r^2}{a^2}\right), \tag{2.5}$$

where v_0 = 73 km/s (Newberg et al., 2010) is a scale velocity, and a = 12 kpc is a scale parameter (Law et al., 2005). The scale velocity, as with many other halos with a similar parameter, is determined from the rotation speed of stars at large radii with respect to the Galactic center (Binney & Tremaine, 2006). This parameter is set so that the sum of the three components of this Milky Way galactic potential yields those rotation speeds. The potentials are in units of $[\Phi] = \mathrm{kpc}^2/\mathrm{Gyr}^2$; a description of these units is given in Section 5.2. When implemented in the simulation, the acceleration due to the above potentials are used, descriptions of which are in Section 2.1.1. The final potential model for the Milky Way is then the sum of these potentials:

$$\Phi_{MW}(r) = \Phi_{bulge}(r) + \Phi_{disk}(r) + \Phi_{halo}(r), \tag{2.6}$$

An important note here is that each potential component, and thus the combined model, is dependent only on the distance from the center of the Milky Way.

2.1.1 Accelerations Due to Milky Way Potentials

During the force calculations the potential equations are not used, but rather their derivatives: the accelerations due to their respective potentials. The potentials are either spherical or cylindrical but the positions and velocities are in Cartesian coordinates. Therefore, when calculating the accelerations, they are calculated for each component in Cartesian space. The parameters used below are the same as the parameters involved in the respective potentials.

For the spherical Hernquist model, the accelerations have a simple form and are the same for each component. The accelerations for each component are given by:

$$\vec{a} = -\frac{\vec{x}M}{r(r+d)^2}. \tag{2.7}$$

The Miyamoto-Nagai disk is not spherically symmetric; the z-component of the acceleration is different from the others. For the Miyamoto-Nagai disk, the accelerations are given by:

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = -\frac{M}{\left[x^2+y^2+\left(b+\sqrt{z^2+c^2}\right)^2\right]^{\frac{3}{2}}} \begin{pmatrix} x \\ y \\ z\frac{b+\sqrt{z^2+c^2}}{\sqrt{z^2+c^2}} \end{pmatrix}. \tag{2.8}$$

The logarithmic halo is spherically symmetric. Therefore the components of the acceleration are all the same. However, there are some models which prefer an oblate or prolate halo (Cole & Lacey, 1996; Jurić et al., 2008; Law & Majewski, 2010). Therefore, a parameter is introduced into the z-component that can modify the shape to accomplish either model choice. For the log halo the components of the acceleration are given by:

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = -\frac{2v_0^2\gamma^2}{\gamma^2(x^2+y^2+a^2)+z^2} \begin{pmatrix} x \\ y \\ z/\gamma^2 \end{pmatrix}, \tag{2.9}$$

where γ is the ‘flattening’ parameter. Since we want a spherically symmetric halo, this parameter is set to 1. Since the accelerations are vectors, each Cartesian component is added separately to the total acceleration of each body.

2.2 Choice of Orbit

In general an orbit is defined by a position in space (3 parameters), a velocity within that space (3 more parameters), and a potential through which the particle will travel. Since the particle can be at any place along the one-dimensional orbit, the orbit can be defined by 5 parameters. Rather than start the dwarf galaxy at all possible positions, velocities, and times in the past, we would like to choose only those positions and orbital times that return the debris to the location we currently see it, on the orbit it is currently traveling. To achieve this, at the beginning of the simulation we perform a 'reverse' orbit calculation: a single body is placed at the current debris coordinates, but with the sign of the velocities reversed. The orbit is then integrated for a variable evolution time (this is one of the five adjustable parameters that are fit) and the final positions and velocities, with their sign again reversed, recorded. The phase space coordinates of this body is then taken as the center of mass position and momentum of the dwarf galaxy. The integration is done using the same method of integration as with the N-body simulations and will be described in Section 5.1.

The dwarf galaxy is then placed at this position on the backwards orbit. The dwarf galaxy then evolves along the forward orbital path. With an initial position given by $(l, b, R) = (218^\circ, 53.5^\circ, 28.6\text{ kpc})$ with velocity $(v_x, v_y, v_z) = (-156, 79, 107)$ kpc Gyr^{-1}, similar to that given by Model 5 of Table 3 in Newberg et al. (2010), the orbit and the approximate ending position of the tidal debris is fixed. As we will see, the integration time in the forward direction will need to be adjusted slightly compared to the reverse orbit in order to put the debris in exactly the observed location.

CHAPTER 3
Dwarf Progenitor Model

3.1 Theoretical Model

We choose a dwarf galaxy model that can represent the full spectrum of possible mass-to-light ratios, i.e., from little to no dark matter as in globular clusters to heavily dark matter dominated as expected in ultrafaint galaxies. The model must be able to represent both situations by adjustment of the model parameters. This allows us the freedom to fit the data without previous knowledge of how much dark matter should be present in the progenitor.

For our model, we adopt a double Plummer potential profile - one representing dark matter and one representing baryonic matter. The Plummer model is not the most realistic profile for modeling dwarf galaxies, and is often used to model globular clusters (Heggie & Hut, 2003); it was chosen for its simplicity and ease of implementation. As we will describe further later, the probability distribution function for the Plummer model is known analytically. Furthermore, the model lacks a singularity at $r = 0$ and the mass enclosed quickly falls off for larger radii (Heggie & Hut, 2003). In future work, we plan to implement a wider range of dwarf galaxy models.

To derive this model, we use the single Plummer density distribution and potential (Plummer, 1911) as a starting point. This is a spherically symmetric model often used in N-body simulations (e.g. Newberg et al. 2010, Law & Majewski 2010,). The Plummer density distribution and potential (Plummer, 1911) is given by:

$$\rho(r) = \frac{3M}{4\pi a^3}\left(1 + \frac{r^2}{a^2}\right)^{-5/2}, \tag{3.1}$$

$$\Phi(r) = -\frac{GM}{a}\left(1 + \frac{r^2}{a^2}\right)^{-\frac{1}{2}}, \tag{3.2}$$

where a is a scale length, M is the total mass, and r is some radius. This model is robust

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, “Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing,” ApJ.

enough to be extended to a two-component model. This is done by adding another term to the single Plummer potential:

$$\Phi(r) = -\frac{GM_d}{a_d}\left(1+\frac{r^2}{a_d^2}\right)^{-\frac{1}{2}} - \frac{GM_b}{a_b}\left(1+\frac{r^2}{a_b^2}\right)^{-\frac{1}{2}}, \tag{3.3}$$

where the subscripts represent the dark matter (d) and baryonic matter (b) components and a_d and a_b are the dark matter and baryonic matter scale lengths respectively. As a result of the linearity of Equation 3.3, the spatial density distribution follows from Poisson's equation for gravity: $\nabla \cdot \Psi = -4\pi G\rho$:

$$\rho(r) = \frac{3M_d}{4\pi a_d^3}\left(1+\frac{r^2}{a_d^2}\right)^{-\frac{5}{2}} + \frac{3M_b}{4\pi a_b^3}\left(1+\frac{r^2}{a_b^2}\right)^{-\frac{5}{2}}. \tag{3.4}$$

These two equations completely characterize the dwarf model being used. The parameters for the dark matter and baryonic matter component allow for easy alteration of each component's mass and radial extent. Furthermore, the combined model maintains spherical symmetry.

This model represents the initial dwarf galaxy in our simulations, before any disruption. The bodies in the simulation must be assigned positions and velocities in a way as to completely trace this initial potential model. In other words, the combined distribution of the bodies in the initial simulated dwarf galaxy must be characterized by this potential model.

Monte Carlo rejection sampling (von Neumann, 1951) is used to assign the body's coordinates in phase space. The positions are sampled and assigned first. These are then used to assign velocities. The positions are sampled so that the overall radial density distribution follows the correct theoretical form, and will be described in Section 3.2. The velocities are assigned so that the structure of the dwarf galaxy is stable and in equilibrium as will be described in Section 3.3.

3.2 Implementation – Assigning Radii

The positions are assigned by sampling the two components of Equation 3.4 separately using their single Plummer density distribution. This is allowed because the density distribution is a linear combination of the two components. Sampling separately ensures a proper spatial distribution for each component; the two components are stacked together with a common center of mass. They not only follow their individual theoretical forms, but, together, follow the form for the combined density distribution, as will be demonstrated in Chapter 4 where we discuss the testing of our algorithm.

The density distribution is spherically symmetric and a function of radius (Binney & Tremaine, 2006). This means that sampling need only be done in one dimension. Cartesian positions can be determined by first sampling for a radial magnitude, assigning each a randomly oriented unit vector, and then taking the components. The density is not the function actually sampled. The mass enclosed function, the integral of the density distribution, is used to determine a radial magnitude (Binney & Tremaine, 2006). That is, the sampling is performed under the curve

$$\frac{\mathrm{d}M}{\mathrm{d}r} = 4\pi r^2 \rho. \tag{3.5}$$

First a random radial magnitude, r, and a random number u from a uniform distribution between [0,1] are chosen. If the condition $r^2\rho/(r^2\rho)_{max} > u$ is met, the radial magnitude is accepted (von Neumann, 1951). Since the two components are sampled separately, the value $(r^2\rho)_{max}$ can be found analytically for each component. This value occurs at:

$$r_{max} = \sqrt{\frac{2}{3}}a, \tag{3.6}$$

where a is, again, the scale length of the respective component. This method allows us to sample underneath the curve created by dM/dr, ensuring the values found are well distributed and representative of the function being sampled (von Neumann, 1951). Therefore, at the end of the sampling routine, we have two sets of radii, one for each component, representative of their respective density distributions. An example of this routine's ability to sample dM/dr is shown below. The accepted points are shown in green and the rejected points in red. As the plot shows, all accepted points are below the curve, while all rejected points are above the curve. This plot also highlights the main drawback of this method: many points are rejected, which can severely slow the sampling routine. Depending on the shape of the curve, the number of rejected points can far outnumber the number of accepted points.

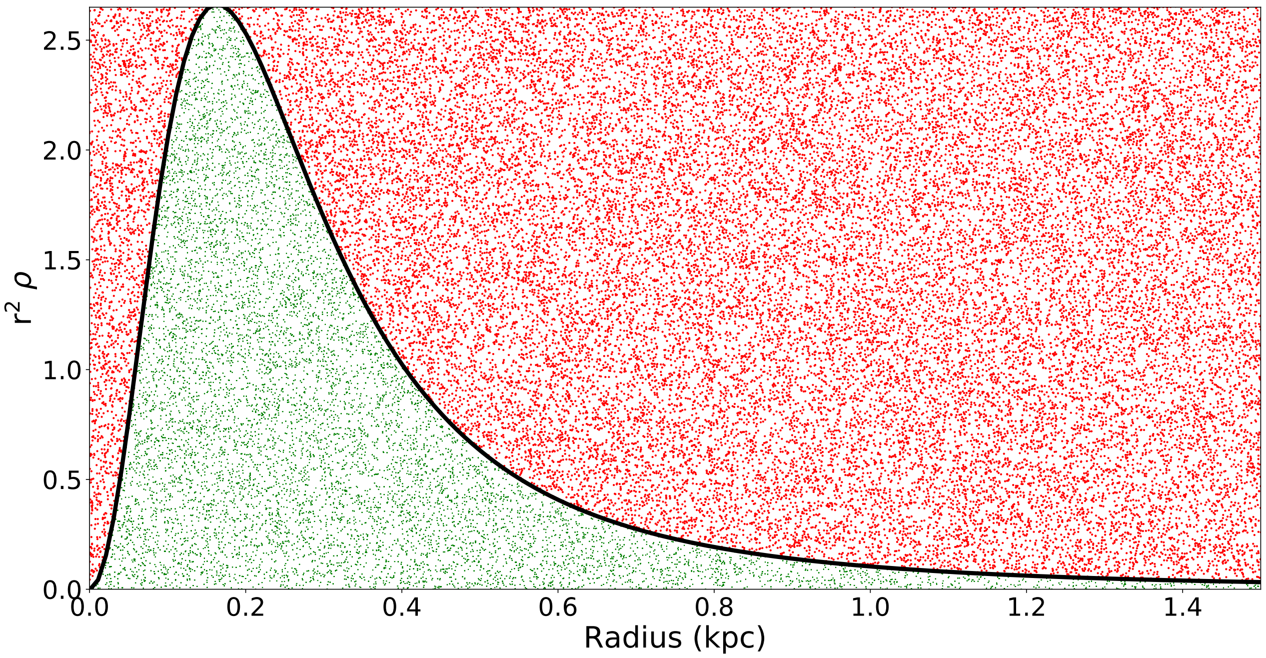

Figure 3.1: An example of the results of rejection sampling. The black line shows the function $r^2\rho(r)$, where $\rho(r)$ is a single Plummer density function, that is being sampled. The green dots underneath the curve are the values that have been accepted and the red dots above the curve are the values that have been rejected. As the plot shows, the accepted values are a good sampling of the function.

The method for determining positions from the radius is the same as that used for determining velocities from orbital speeds. Therefore, it will be discussed at the end of this chapter.

3.3 Implementation – Assigning Velocity

Each body must also be assigned an orbital speed. This is a significantly more complicated process. The stability of the simulated dwarf galaxy relies heavily on the velocity distribution. If the assigned speeds are too high the bodies will fly off. If too low, the bodies will collapse to the center. In either case, the simulated dwarf galaxy will quickly decay. Instead, we want an object that maintains its general shape and distribution if it is left to evolve in empty space, i.e., no external potentials. For this to occur, the bodies must have stable orbits around the dwarf's center of mass.

We cannot assign velocities to each component separately as we did with positions. The velocities are determined using a probability distribution (Binney & Tremaine, 2006; Aarseth et al., 1974), described in Section 3.3.1. The two component Plummer model probability distribution function is not a simple sum of the single components' probability distributions as the combined density distribution is a simple sum of the one component density distributions. Furthermore, as we will see in Section 3.3.1, the probability distribution is a function of the density and potential, and cannot be separated into a linear combination of the single component densities and potentials.

In fact, if we were to attempt to assign velocities to each component separately, when the two components are brought together, the combined structure would be inherently unstable. This is because there will suddenly be more mass located within the orbital radius for each body. The orbits for these bodies would decay leading to a runaway process as the entire structure rapidly decays, or even violently blows apart. This was seen in previous attempts at creating a two component structure.

3.3.1 Distribution Function

A probability distribution function, when integrated in phase space, produces the probability an object will be found in that area of phase space (Binney & Tremaine, 2006). The distribution can be reduced to a function of the body energy, model density and potential. The spherically symmetric probability distribution we use is (as taken from Binney & Tremaine (2006))

$$f(\epsilon) = \frac{1}{\sqrt{8}\pi^2} \int_0^{\epsilon} \frac{d\Psi}{\sqrt{\epsilon - \Psi}} \frac{\mathrm{d}^2 \rho}{\mathrm{d}\Psi^2}, \tag{3.7}$$

where $\Psi(r) = -\Phi(r)$, is the combined model potential, $\rho(r)$ is the model density distribution, and the energy, ϵ, is defined as $\epsilon = -\frac{1}{2}v^2 - \Phi(r)$. For a single component Plummer model, the integral in Equation 3.7 has an analytic solution, given by:

$$f(\epsilon) = \frac{24\sqrt{2}}{7\pi^3} \frac{a^2}{M^4} \epsilon^{7/2}, \tag{3.8}$$

where M is the total mass, and a is the scale radius of the model. The simplicity of this equation is another reason the Plummer model is so widely used. Sampling the distribution function for a single component Plummer sphere is often very fast and the algorithm is simple to implement. For a two component model, however, the integral in Equation 3.7 must be solved numerically, which is a decisively slower process. A theoretical description and the implementation is to follow.

3.3.2 Calculating Derivatives

Since the density, $\rho(r)$, and potential, $\Psi(r)$, equations are both one dimensional functions of radius, the integral of Equation 3.7 is converted to integrate over radius. This is done by expanding the derivative in the integrand in terms of r. The first derivative is expanded as:

$$\frac{\mathrm{d}\rho}{\mathrm{d}\Psi} = \frac{\mathrm{d}\rho}{\mathrm{d}r}\frac{\mathrm{d}r}{\mathrm{d}\Psi} = \frac{\mathrm{d}\rho}{\mathrm{d}r}\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}. \tag{3.9}$$

The first derivative is simple enough, and provides us with footing to expand the second derivative. The derivative with respect to potential, $\frac{d}{d\Psi}$, is treated as an operator that, for a function of radius:

$$\frac{\mathrm{d}f(r)}{\mathrm{d}\Psi} \rightarrow \frac{\mathrm{d}f(r)}{\mathrm{d}r}\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}. \tag{3.10}$$

Using this operator on the first derivative we find,

$$\frac{\mathrm{d}}{\mathrm{d}\Psi}\left[\frac{\mathrm{d}\rho}{\mathrm{d}r}\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}\right] = \frac{\mathrm{d}}{\mathrm{d}r}\left[\frac{\mathrm{d}\rho}{\mathrm{d}r}\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}\right]\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}. \tag{3.11}$$

This becomes:

$$\frac{\mathrm{d}^2\rho}{\mathrm{d}\Psi^2} = \left[\frac{\mathrm{d}^2\rho}{\mathrm{d}r^2}\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1} - \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-2}\frac{\mathrm{d}^2\Psi}{\mathrm{d}r^2}\frac{\mathrm{d}\rho}{\mathrm{d}r}\right]\left(\frac{\mathrm{d}\Psi}{\mathrm{d}r}\right)^{-1}. \tag{3.12}$$

The differential $\mathrm{d}\Psi$ is also changed to $\mathrm{d}r$: $\mathrm{d}\Psi \rightarrow g(r)\mathrm{d}r$, where $g(r)$ is the derivative of Ψ with respect to r. This is done by multiplying the integrand by $d\Psi/dr$. While this may seem incorrect, when applied, the differential, '$\mathrm{d}r$', is not actually present, only $g(r)$. Therefore, multiplying by $g(r)$ is the same as changing the differential from $\mathrm{d}\Psi$ to $\mathrm{d}r$. The final integral is then given by:

$$f(\epsilon) = \frac{1}{\sqrt{8}\pi^2} \int_0^{\epsilon} \frac{g(r)dr}{\sqrt{\epsilon - \Psi(r)}} \left[\frac{\mathrm{d}^2\rho}{\mathrm{d}r^2} \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-1} - \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-2} \frac{\mathrm{d}^2\Psi}{\mathrm{d}r^2} \frac{\mathrm{d}\rho}{\mathrm{d}r} \right] \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-1}. \tag{3.13}$$

3.3.3 Calculating Limits of Integration

The limits of integration must also be converted to radial limits, r'_{lower} and r'_{upper}, by solving for the radius, r', at which the potential is equal to the energy: $\Psi(r') = \epsilon'$, where ϵ' is either of the current limits of integration, $\epsilon'_{lower} = 0$ and $\epsilon'_{upper} = \epsilon$.

At the lower limit, $\Psi(r'_{lower}) = 0$ when $r'_{lower} = \infty$. However, the integrand has a well defined peak and quickly decreases to zero, allowing for an easy approximation for this limit. This is shown in Figure 3.2.

At the upper limit, $\Psi(r'_{upper}) = \epsilon$, and r'_{upper} is determined by finding the root of $\Psi(r'_{upper}) - \epsilon = 0$. The root is found numerically using a bisection root finding algorithm implemented specifically for this case. This turns the integral into:

$$f(\epsilon) = \frac{1}{\sqrt{8}\pi^2} \int_{\infty}^{r'_{upper}} \frac{g(r)dr}{\sqrt{\epsilon - \Psi(r)}} \left[\frac{\mathrm{d}^2\rho}{\mathrm{d}r^2} \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-1} - \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-2} \frac{\mathrm{d}^2\Psi}{\mathrm{d}r^2} \frac{\mathrm{d}\rho}{\mathrm{d}r} \right] \left(\frac{\mathrm{d}\Psi}{\mathrm{d}r} \right)^{-1}. \tag{3.14}$$

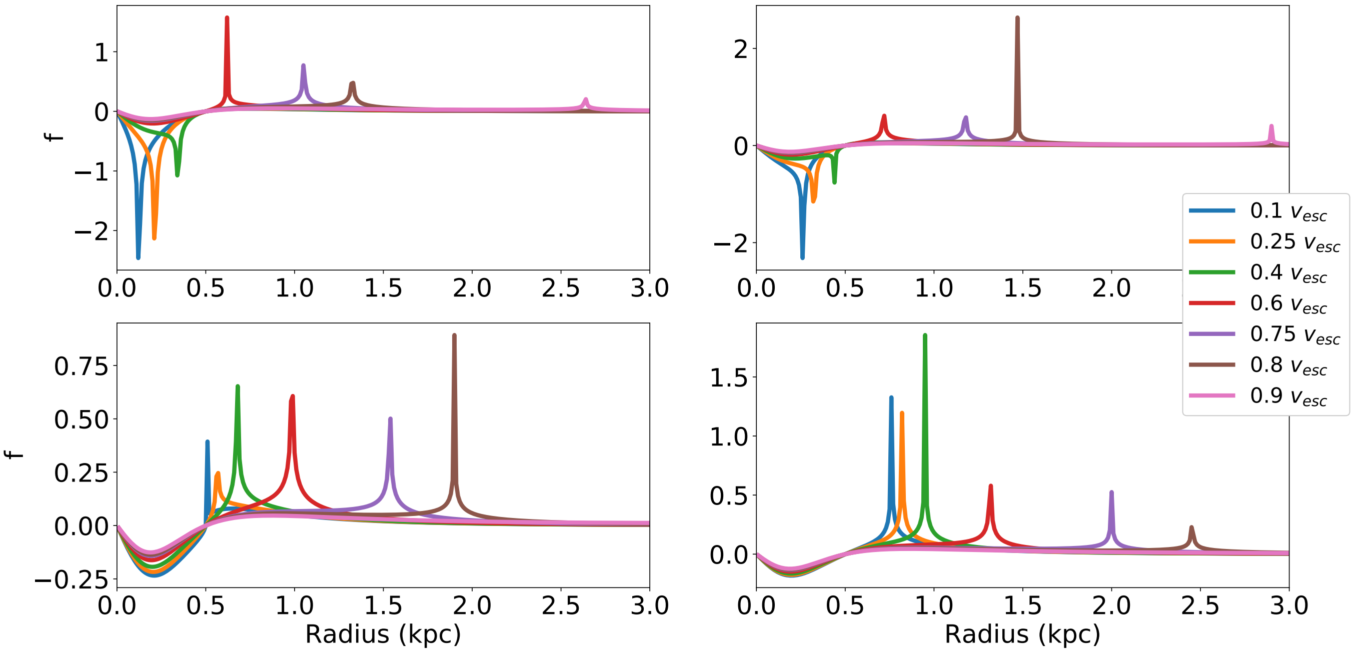

Figure 3.2: Plots of the integrand of the distribution function for different energies, $\epsilon = \Psi(r) - 1/2v^2$, as a function of r', the value integrated over. Each plot shows a different radius, r, each a fraction of the sum of the scale radii, $(a_b + a_d)$, of the dwarf galaxy components. The upper left is $0.1(a_b + a_d)$, the upper right is $0.25(a_b + a_d)$, the lower left is $0.5(a_b + a_d)$, the lower right is $0.75(a_b + a_d)$. Each curve shows a different fraction of the escape velocity for that r. In each case, the function quickly goes to zero.

3.3.4 Numerical Formulas

There are several numerical methods used in calculating and sampling the distribution function. They will not be derived because they are all well established numerical methods. However, they are important to the replication of this algorithm and thus will be listed here.

As mentioned above, the upper limit of integration is found using a bisection root finding algorithm. This method searches for a root within an interval provided to the algorithm. The search interval was determined in a way to ensure there existed a root within. Again, we are looking for the r' that satisfies $\Psi(r') = \epsilon$. The search interval, (r'_{lower}, r'_{upper}), should satisfy the requirement $\Psi(r'_{lower}) > \epsilon$ and $\Psi(r'_{upper}) < \epsilon$. Since $\epsilon = \Psi(r) - 1/2v^2$, and $\Psi(r)$ is a positive quantity, it follows that $\epsilon \leq \Psi(r)$. It then also follows that lower energies correspond to smaller values of $\Psi(r')$, and thus larger values of r'. Therefore, the lower bound is set to $r'_{lower} = 0$ where the first inequality is definitely satisfied. The upper bound is first given an initial value and is iteratively increased until $\Psi(r'_{upper}) < \epsilon$.

For the first and second derivatives, a five point central difference approximation is used in order to ensure accuracy. For the first derivative, this is given by:

$$f'(x) \approx \frac{f(x-2h) - 8f(x-h) - f(x+2h) + 8f(x+h)}{12h}, \tag{3.15}$$

and the second derivative is given by

$$f''(x) \approx \frac{-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)}{12h^2}, \tag{3.16}$$

where h is the step size and is set to $h = 0.001$ in both cases. This value was determined to work well for the functions involved: the density and potential. The integral is performed using a 3-point Gaussian quadrature technique. This technique is based on the rule:

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \sum_{i=1}^{n} w_i f\left(\frac{b-a}{2} x_i + \frac{a+b}{2}\right), \tag{3.17}$$

where w_i are weights and x_i are specific points, both of which can be looked up in tables, and n=3. For our purposes, this formula is not applied over the entire integration range, but over small increments. Therefore, if $v^2 f(r) = f(x)$, what is actually implemented is given by:

$$\begin{aligned}
\int_a^b f(x)dx = \sum_{i=0} \int_{\alpha_i}^{\beta_i} f(x)dx &\approx \sum_i c_1 \sum_{j=1}^{3} w_j f\left(c_1 x_j + c_2\right), \\
c_1 &= \frac{\beta_i - \alpha_i}{2}, \\
c_2 &= \frac{\beta_i + \alpha_i}{2}, \\
\beta_i - \alpha_i = h &= \begin{cases} (1.5a - a)/100 & \text{if } \alpha_i < 1.5a \\ (b - 1.5a)/10 & \text{if } \alpha_i > 1.5a \end{cases},
\end{aligned} \tag{3.18}$$

where h is an adaptive step size used to give better resolution to more significant parts of the integrand, like the peaks in Figure 3.2. This normally occurs within $1.5a$. The limits of integration, α and β, are iterated in a way that allows for alteration of h during the integration. The i sum is taken up to where the lower bound, α, exceeds b.

Finally, a max finder based on a golden section search is used to determine $(v^2 f)_{max}$ for a given radius (Press et al., 2007).

3.3.5 Sampling Velocities

As we have shown, the distribution function can be reduced to a function of radius, r, and velocity, v, by expansion of the derivatives and through the calculation of energy. The dimensionality of the distribution function is further reduced by using the density distributions to first assign radius before assigning velocities. Since the spatial location has already been determined, we need only generate the magnitude of the particle velocity and three random numbers specifying the direction of the velocity unit vector as will be discussed in the following section. The distribution function then becomes a one dimensional function of velocity magnitude. Similar to the mass enclosed function for the positions, we choose the magnitude of the velocity using the spherically symmetric distribution (Binney & Tremaine, 2006),

$$\frac{dN(v)}{dv} = 4\pi v^2 f(v), \tag{3.19}$$

where dN is the number of particles between v and $v + dv$, and the same rejection sampling technique as before. In this case, $(v^2 f(v))_{max}$ for the given radius is found numerically.

3.4 Implementation – Determining Positions

We choose to make the dwarf galaxy isotropic in space. Therefore, the particle position vector, $\vec{x}$, and velocity vector, $\vec{v}$, is determined by multiplying the magnitudes by a different but random Cartesian unit vector on the unit sphere:

$$\vec{x} = |r|\hat{u}_1, \tag{3.20}$$

$$\vec{v} = |v|\hat{u}_2. \tag{3.21}$$

The unit vector is chosen by first selecting a vector, $\vec{u} = (u_x, u_y, u_z)$, created from three random numbers between [-1,1], such that

$$u_x^2 + u_y^2 + u_z^2 < 1. \tag{3.22}$$

Then,

$$\hat{u} = \frac{\vec{u}}{\sqrt{u_x^2 + u_y^2 + u_z^2}}, \tag{3.23}$$

is a unit vector that is randomly sampled over the surface of a unit sphere. The components of the product vector are then taken as the Cartesian coordinates for the positions/velocities of that body: by drawing the radial and velocity magnitudes from density and probability distributions respectively, and the directions randomly, we achieve a system with the desired density and velocity distributions but with random motions. Furthermore, such a method allows for various types of body orbits.

3.5 Implementation – Assigning Body Masses

The number of baryonic and dark matter bodies is fixed for all simulation parameters: half of the bodies are assigned to baryons and the other half as dark matter. Currently, we use 20,000 bodies in the simulation, with 10,000 assigned to each component. The mass of each component is split evenly between their respective bodies. The ratio between the dark matter and baryonic matter component masses is altered by changing the mass assigned to each component and thus their bodies.

Specifically, if $m_{b,i}$ is the mass of the ith baryonic body, $m_{b,j}$ is the mass of the jth baryonic body, such that $i \neq j$ (and similarly for the dark matter component),

$$m_{b,i} = m_{b,j \neq i}, \tag{3.24}$$

$$m_{d,i} = m_{d,j \neq i}. \tag{3.25}$$

Every baryonic body has the same mass as every other baryonic body and every dark matter body has the same mass as every other dark matter body. But, the mass of each components' bodies, $m_{b,i}$ may not be the same as the masses of the other components' bodies, $m_{d,i}$. The only time they are the same is if the total mass of each component is the same. So, again, if m is the mass of a body and M is the total mass of a component:

$$m_{b,i} \begin{cases} \neq m_{d,j} & \text{if } M_b \neq M_d \\ = m_{d,j} & \text{if } M_b = M_d \end{cases}, \tag{3.26}$$

meaning, again, that the mass of a baryonic body has the same mass as a dark matter body only if the total mass of the two components are the same. Therefore, a dwarf galaxy, for example, with twice as much dark matter as baryonic matter would have $M_d/M_b = 2$ and $m_d/m_b = 2$. This is our current method for assigning body masses and type in our algorithm.

Alternatively, the body masses could be set uniformly, each the same fraction of the total mass of the dwarf galaxy, $m = (M_b + M_d)/N$, where N is the total number of bodies in the simulation. In this case, the combined density distribution, instead of the individual densities, would be sampled. Each body, initially tagged as baryonic, would then be given a chance to be tagged as dark matter. The ratio between the amount of dark matter and baryonic matter would be realized by how many bodies are assigned as dark matter bodies and how many are assigned as baryonic bodies. The bodies would be iterated over and randomly tagged as dark matter until this was achieved. Therefore, for dwarf galaxies with some dark to light mass ratio $M_d/M_b = \xi$, the body masses would be equal, $m_d = m_b$, and the number of each type of body would be $N_d/N_b = \xi$. This method would, then, require an entirely separate sampling process. In fact, this was the previous method of assigning body masses and type.

The main problem with this method is highlighted when the ratio between dark matter and baryonic matter becomes extreme. In those cases there will be very few bodies of one type in the simulation. The bodies are still meant to represent the potential of that component, which require a fairly large amount of bodies, upwards of a few thousand to ten thousand. Therefore, to account for extreme dark to light ratios, the total number of bodies in the simulation would have to be extremely large, possibly in the hundreds of thousands, so that when these situations occur, the number of bodies assigned to the less massive of the components would still be enough to accurately represent the potential well. Even then, there is still no guarantee that there would be enough bodies to properly represent the potential well of that component. The amount of bodies needed would slow the simulation considerably, making it unfeasible to run on client computers. It is for this reason that this method was discarded and our current method adopted.

Our current method for assigning bodies allows us to fix the number of bodies to a computationally reasonable value while still maintaining enough bodies in each component to properly represent the potential well of that component and of the entire model. As we have said, we currently use 20,000 bodies, with 10,000 for each component, as compared with the over 100,000 bodies needed with the previous method. Furthermore, our current method does not require the several iterations over the entire body list to randomly assign dark matter. Also, the previous method requires us to sample the combined density distribution. In the rejection sampling technique, as was previously discussed, the calculation of $r^2\rho(r)_{max}$ is required and would need to be found numerically as opposed to the analytic values currently used. Therefore, our current method also reduces the number of calculations required in our sampling, and simplifies the process. For these reasons, our current method is much more feasible and all around more attractive.

CHAPTER 4
Testing the Dwarf Model

As the previous chapter shows, the dwarf galaxy creation algorithm is quite complex. Therefore, we must test that the dwarf galaxy creation algorithm works correctly. We test our dwarf galaxy creation algorithm by testing the physical properties of the simulated dwarf galaxy in empty space. This means making sure the algorithm produces the correct distributions in phase space, testing the stability of the dwarf galaxy, and checking whether or not it is created in and maintains virial equilibrium when evolved in empty space.

First, the dwarf's spatial and velocity distributions are compared to theoretical norms when the dwarf galaxy is first created. This is repeated after a number of years of evolution in empty space to test that the dwarf galaxy retains these distributions. The virial ratio, the ratio of the total kinetic energy to total potential energy of the bodies in the dwarf galaxy, is also measured. The virial theorem states that in order for the dwarf galaxy to be in virial equilibrium, the virial ratio must be $2T/|U| = 1$, where T is the total kinetic energy, and U is the total potential energy. The virial ratio is calculated using two methods. This measurement is done when the dwarf galaxy is first created, and after a number of years of evolution in empty space, to test that the dwarf galaxy retains virial equilibrium. These tests, together, show that the dwarf galaxy is created correctly. Repeating them over time tests the stability. By stability, we mean that it retains, without significant deviation, the initial spatial and velocity distributions, and equilibrium status.

We test two different models for the progenitor, one where mass follows light and one with an extended dark matter profile. Since this thesis focuses on the Orphan Stream in particular, in both cases, the parameters for the dwarf galaxy are adapted from Newberg et al. (2010) for the best fit Plummer model, scale radius of $r_s = 0.2$ kpc and total mass of $M_{total} \sim 2.5 \times 10^6 M_\odot$, to the orbit of the Orphan Stream found in the same. In the first dwarf model, the two components are given the same parameters. Their scale radii are both set to 0.2 kpc, and the mass of each is one half of M_{total}, corresponding to a mass follows light model of the dwarf galaxy as in Newberg et al. (2010). The second dwarf galaxy uses

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, “Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing,” ApJ.

the same values for the baryonic component. For the dark matter component of this model, a scale radius of $r_s = 0.8$ kpc and a mass of $M_{DM} \sim 1 \times 10^7 M_\odot$ is used. This corresponds to a dark matter dominated dwarf galaxy with an extended dark matter halo as predicted by some models.

While we present here only two dwarf models, the tests to be described have been repeated over a large range of progenitor parameters. The results for each model was found to closely agree with those shown here.

4.1 Phase Space Distribution

In order to test that the spatial distribution is assigned correctly, we can compare the distribution created by the dwarf creation algorithm to theoretical expectations. First, a dwarf galaxy is created with some test model parameters. The parameters used are not of the most importance as long as they are realistic. The dwarf galaxy is allowed to evolve in empty space for several evolution times, and the final body positions and velocities are recorded. Using the positions and velocities of each body, the magnitude and angular components of the positions and velocities are calculated and made into histograms. The theoretical expectations come from the sampling function. As described in the previous chapter, the mass enclosed function is sampled to assign positions. Dividing by the mass per particle, m, yields the expected counts per histogram bin as a function of r, θ, and ϕ:

$$dN = 4\pi r^2 \frac{\rho(r)}{m} \text{sin}\theta dr d\theta d\phi, \tag{4.1}$$

where $\rho(r)$ is the combined model as given by Equation 3.4.

4.1.1 Testing Radial Distribution

The radial component of the position can be seen in Figures 4.1 and 4.2. The first figure, Figure 4.1, shows the radial profile of a dwarf galaxy whose components have the same parameters, and shows the radial distribution of each component over time. The second, Figure 8.4, shows an example where the parameters of the two components are different. In each, the theoretical distribution, $r^2\rho(r)$, is also plotted. As the figures show, the radial distribution matches well with the theoretical expectation. After 4 Gyr of evolution, there is a significant shift in the radial profile. This is due to relaxation of the bodies in the structure, reaching an equilibrium state. As we will show in Section 4.2, the dwarf galaxy is created very close to equilibrium. As the dwarf galaxy evolves in empty space, the virial ratio approaches one. However, as seen in Figures 4.1 and 4.2, the distribution maintains its general shape after 4 gigayears of evolution.

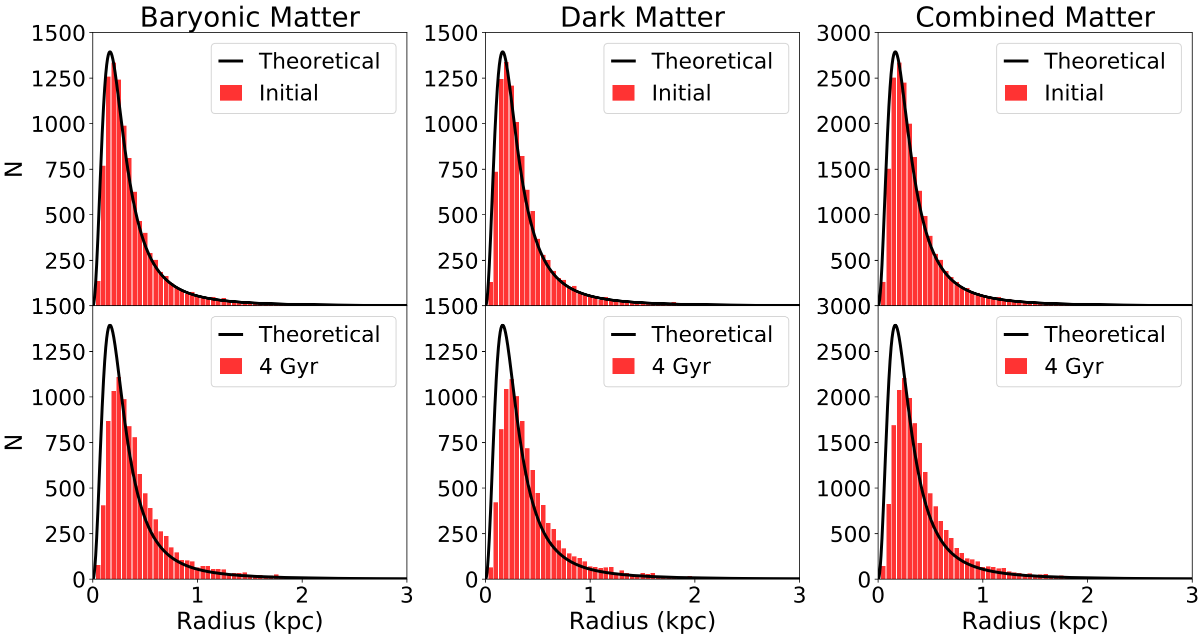

Figure 4.1: Plot of the radial distribution of each component for a dwarf galaxy with a mass follows light model. The black line is the theoretical distribution expected for each component. The dwarf is evolved in empty space. The top row shows each component and the combined distribution initially. The second row shows the distributions after 4 Gyr of evolution. After this evolution there is some deviation from the initial and theoretical distributions but the general shape remains.

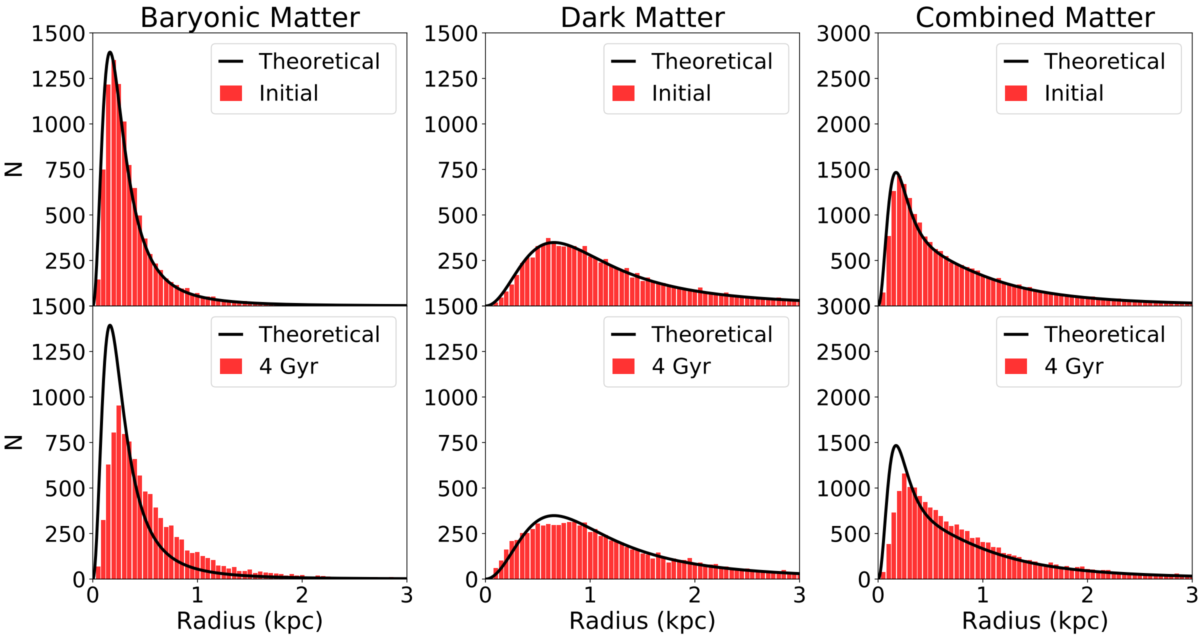

Figure 4.2: Plot of the radial distribution of each component for a dwarf galaxy with an extended dark matter halo. The black line is the theoretical distribution expected for each component. The dwarf is evolved in empty space. The top row shows each component and the combined distribution initially. The second row shows the distributions after 4 Gyr of evolution. After this evolution there is also some deviation from the initial and theoretical distributions but the general shape remains.

4.1.2 Testing Velocity Distribution

As stated in the previous chapter, the theoretical form of the velocity distribution is not known, as the probability distribution function cannot be solved analytically for the two component case. The single Plummer model does have an analytical solution (Aarseth et al., 1974), given by Equation 3.8, which can be used to create a velocity distribution. This is also done by rejection sampling, as described in Section 3.3.5, but with use of the analytic equation for the distribution function.

From the linearity of Equations 3.3 and 3.4, it can be seen that if the masses and scale radii of the two components were identical, the equations would reduce to a one component model given by Equation 3.1. Doing this with the two component algorithm then creates what would be a one component velocity distribution. This can be compared with the distribution created from the analytic solution. Figure 4.3 shows a histogram of a velocity distribution from the two component model with another histogram of a velocity distribution created from the analytic solution for a single Plummer model (represented as a curve) over time. As the plot shows, the velocity distribution of each component matches well with that from the analytic equation. The distribution also retains its shape after 4 gigayears of evolution.

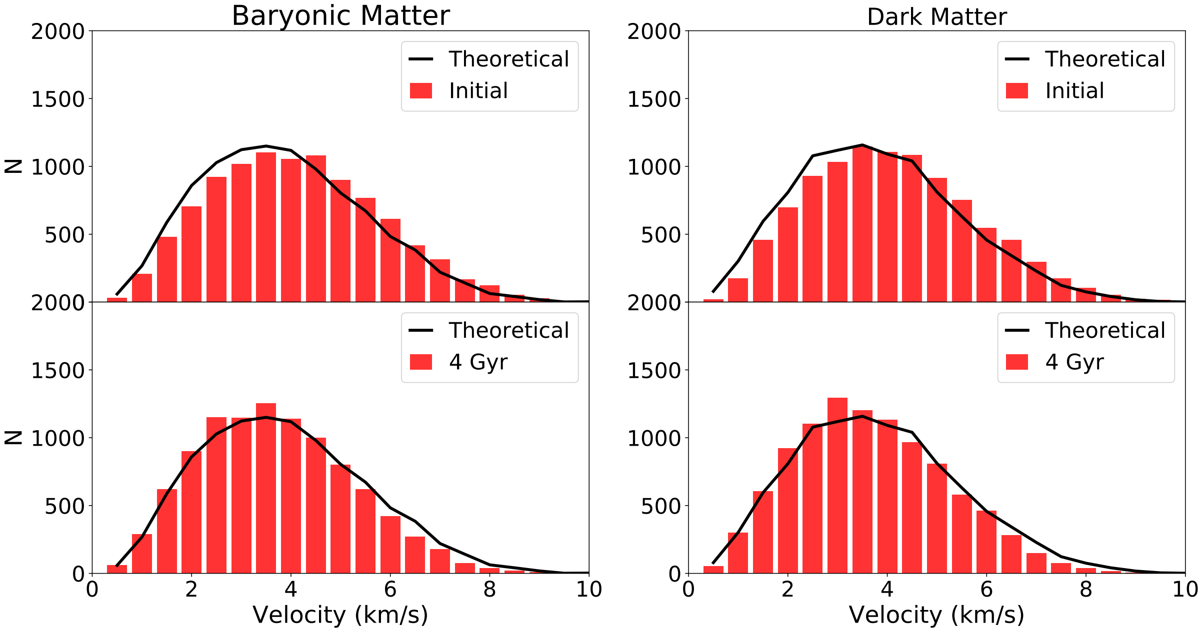

Figure 4.3: The velocity distribution of the dwarf galaxy with a mass follows light model. When the two components have the same parameters, it creates a 'one component' dwarf galaxy made from our two component algorithm. The black line is a theoretical line generated from sampling the analytically known single Plummer distribution function. The first row is the initial velocity distribution of each component. The second row is the velocity distribution of each row after 4 Gyr of evolution. There is no significant deviation over time or from the 'theoretical' distribution.

4.1.3 Angular

We also required that the dwarf galaxy be spherically symmetric. To test that this is indeed the case and to make sure the model retains spherical symmetry over time, the angular components of the positions and velocities are plotted initially and over time in the same fashion as the radii and velocity distributions. As per Equation 4.1, the θ distribution should follow a sine curve distribution and the ϕ distribution should be uniform over the range.

For the mass follows light model, the spatial angular components are shown in Figure 4.4 and the velocity angular components are shown in Figure 4.5. For the model with an extended dark matter halo, the spatial angular components are shown in Figure 4.6 and the velocity angular components are shown in Figure 4.7. As both sets of plots show, the angular components fit the theoretical expectations closely; the model is in fact spherically symmetric. The plots also show that the model retains this symmetry after 4 gigayears of evolution.

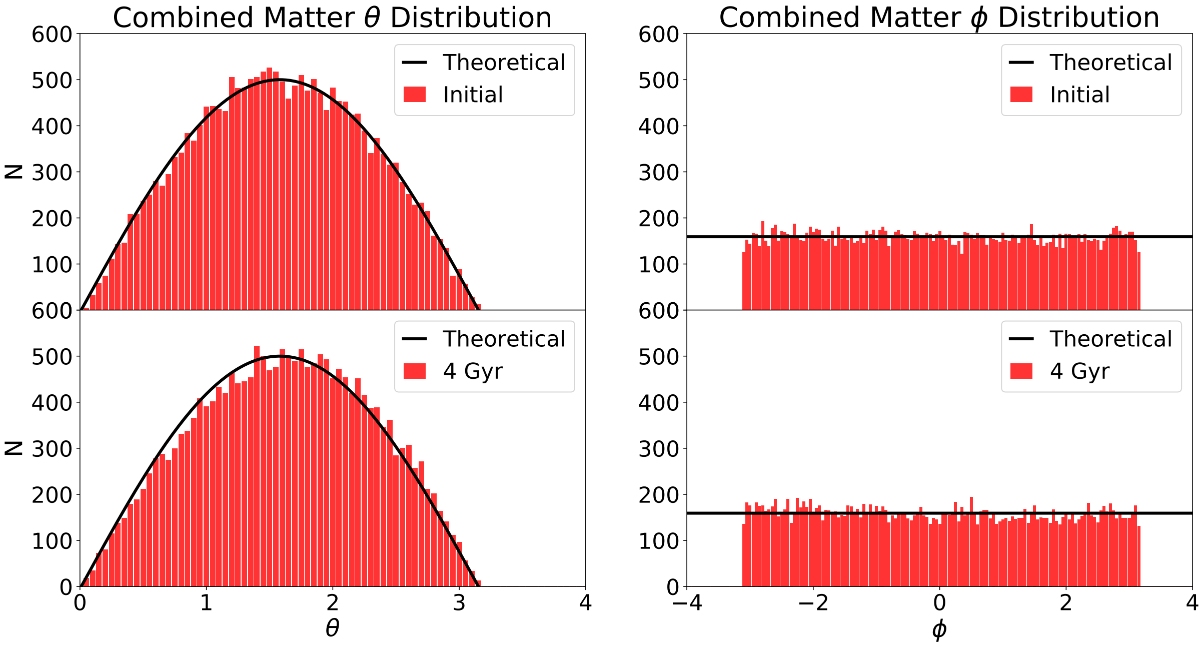

Figure 4.4: The angular components of the spatial distribution initially and after 4 Gyr for a dwarf galaxy with a mass follows light model.

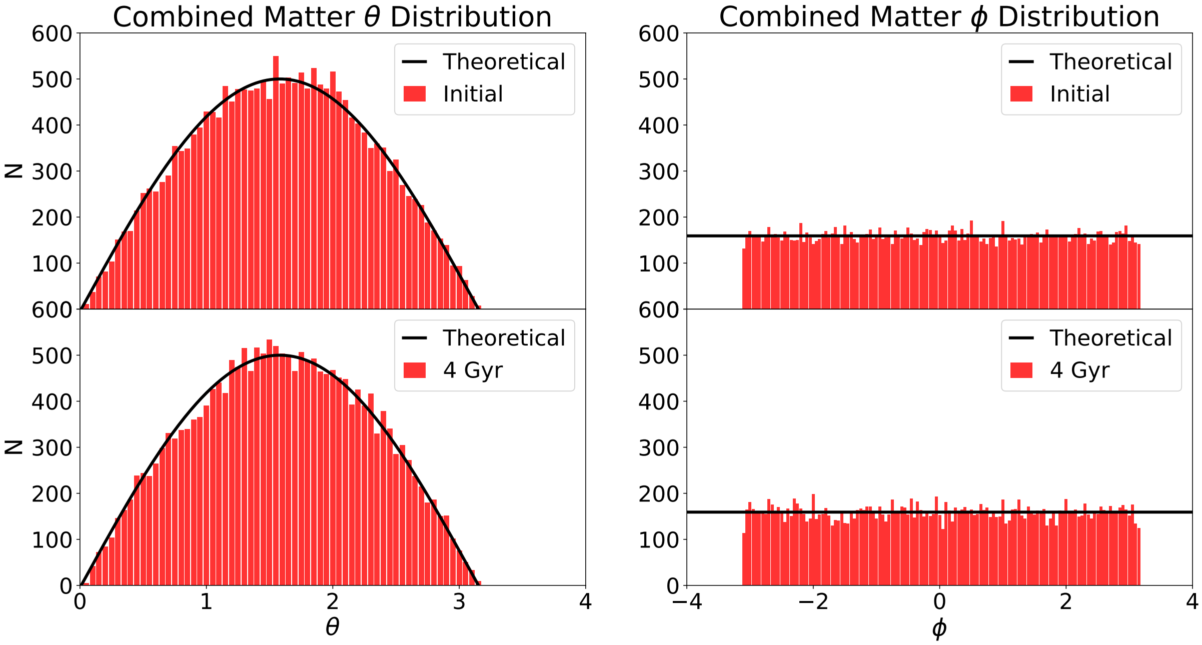

Figure 4.5: The angular components of the velocity initially and after 4 Gyr for a dwarf galaxy with a mass follows light model.

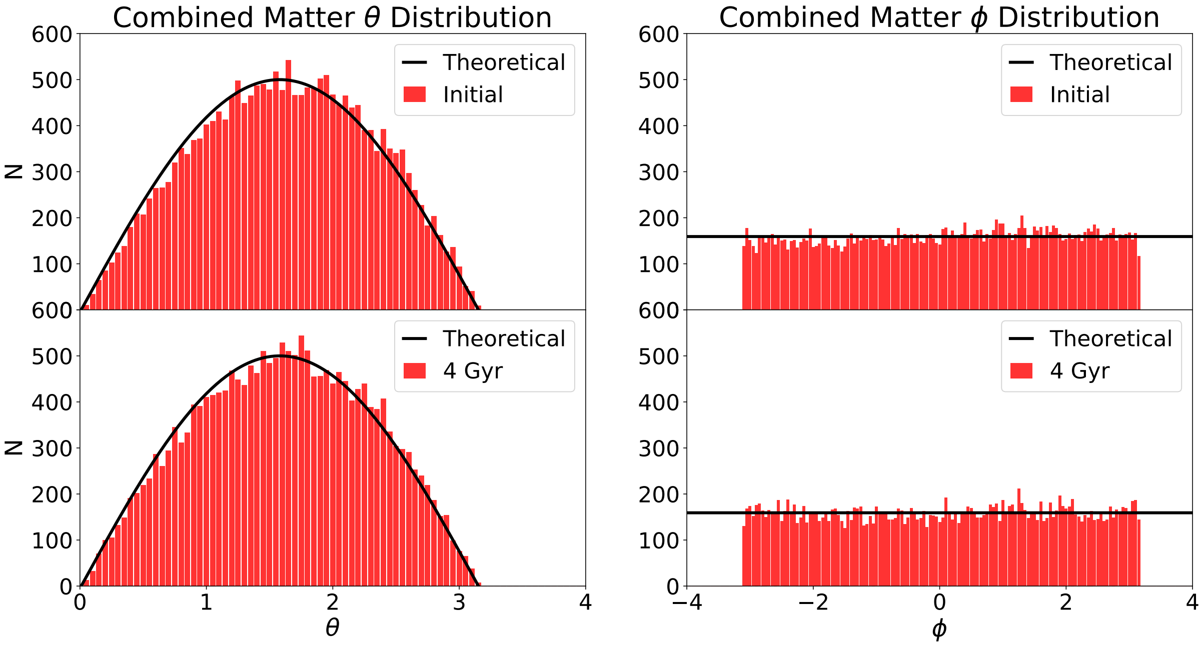

Figure 4.6: The angular components of the spatial distribution initially and after 4 Gyr for a dwarf galaxy with an extended dark matter halo.

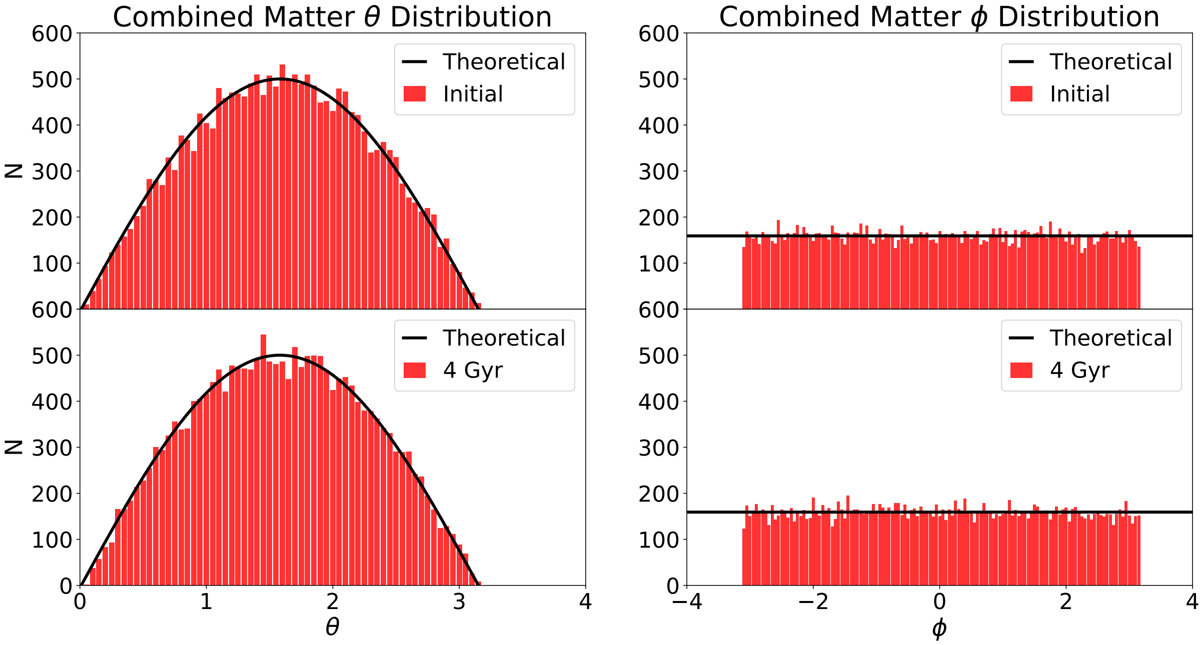

Figure 4.7: The angular components of the velocity initially and after 4 Gyr for a dwarf galaxy with an extended dark matter halo.

4.2 Virial Equilibrium

The previous tests show that the configuration created is in fact stable. We also want to test that the initial structure is in virial equilibrium, and, with no external potential, it remains so over time. Note, though that virial equilibrium is not maintained in the presence of an external potential, in this case that of the Milky Way galaxy. In this setting, as with the other tests in this chapter, the dwarf galaxy is placed in empty space.

An object comprised of N gravitating bodies in virial equilibrium must have a total potential energy that is twice its total kinetic energy (Binney & Tremaine, 2006):

$$|U| = 2T, \tag{4.2}$$

where U is the total potential energy and T is the total kinetic energy. The virial ratio, $U/2T$, then, for a system in virial equilibrium, should be 1. Again, using the spatial and velocity coordinates for each body at the end of the simulation, the total kinetic energy is found by summing the kinetic energy of each body:

$$T = \sum_{i}^{N} \frac{1}{2} m_i (v_x^2 + v_y^2 + v_z^2)_i, \tag{4.3}$$

where m is the mass of a body, the v's are its velocity components in Cartesian space, and N is the number of bodies.

The total potential energy can be calculated in two ways. The first method uses the body positions and the combined theoretical potential of Equation 3.4. Using the Cartesian coordinates to find the radius, the potential energy is given by:

$$U_1 = \sum_{i}^{N} m_i \Phi(r_i). \tag{4.4}$$

The second method uses the classical Newtonian equation, calculating the energy between each individual body:

$$U_2 = \frac{1}{2} \sum_{i \neq j}^{N} \sum_{j \neq i}^{N-1} \frac{m_i m_j}{r_{ij}}. \tag{4.5}$$

The second method is more robust as it does not rely on the form of the potential. Calculating the virial ratio in this way works for any arrangement of bodies. If the system is in equilibrium, then both methods should yield a virial ratio of approximately 1.

The reason the first method is also used is that it allows for a test of whether the bodies are properly tracing the theoretical potential model. If the bodies have been improperly allocated then the ratio calculated from the first method would be vastly different from that of the second method. This can be easily seen if one were to use both methods on a simulated dwarf galaxy, in virial equilibrium, made with some set of parameters, but use a different set of parameters when calculating the theoretical potential in the sum for U_1. The bodies would obviously not trace the theoretical potential calculated with the different set of parameters used for U_1, and thus the virial ratio would be very different from 1. While, the second method, not reliant on the form of the potential would yield a virial ratio of 1. This helps test that the algorithm is producing a proper distribution, while at the same time testing for virial equilibrium.

Table 4.1 shows the results of calculating the virial ratio using both methods, for two simulated dwarf galaxies initially, after 2 gigayears, and after 4 gigayears. The left table was calculated for the simulated dwarf galaxy with both components the same (corresponding to the same simulated dwarf galaxy used in Figures 4.1, 4.3, 4.4, and 4.5). This dwarf galaxy is effectively a one component Plummer model, which is known to be stable and in virial equilibrium (Binney & Tremaine, 2006; Aarseth et al., 1974). The left panel in Table 4.1 shows that the dwarf is created very nearly in virial equilibrium and remains so over 4 gigayears in our simulation as expected.

The right panel in Table 4.1 shows the virial ratio for a dwarf galaxy where the two components have different parameters. In this case the dwarf is also created very nearly in virial equilibrium and also remains so over 4 gigayears. In both cases, the two methods are in very close agreement, further indicating that the sampling is correct and the theoretical potentials are realized upon generation of the dwarf model.

In both panels of Table 4.1, the first method shows a slight decrease of the virial ratio over time. The decrease is not very significant and can be explained by relaxation in the dwarf galaxy distribution. Relaxation in the distribution can change the overall dwarf potential from the theoretical potential used in the first method. This is seen in the radial distribution plots; after 4 Gyr of evolution the radial distribution is shifted from the initial distribution. Relaxation also explains why the second method yielded a ratio that maintained equilibrium values. However, because there is not a large deviation in the radial distribution after 4 gigayears, and relaxation is so slight and occurs on such a large time scale, the dwarf galaxy can be considered stable and in equilibrium.

Table 4.1: Virial ratio of our simulated dwarf galaxy initially, after 2 Gyr and 4 Gyr of evolution.

Dwarf With Identical Components		
Ratio	Method 1	Method 2
Initial	0.960	0.963
2 Gyr	0.958	1.006
4 Gyr	0.935	1.019

Dwarf With Different Components		
Ratio	Method 1	Method 2
Initial	1.007	1.005
2 Gyr	0.996	1.004
4 Gyr	0.984	1.003

CHAPTER 5
Running the Simulation

We have so far described the setting of our simulations, including the model for the Milky Way, and the description of the orbit. We have also described the theoretical dwarf model and how it was implemented in the algorithm. These are the main components of the simulation. There are several other aspects that are necessary in order to run a simulation that we now detail. These include the integrator itself, the units used, the algorithm used to approximate the body-body interactions, determining the timestep and softening parameter, the random number seed, and the outputs of the simulation.

5.1 N-body Integrator

The equations of motion for each body must be solved in order to determine their final positions in phase space. The only force present in the simulation is gravity. The force calculations between the bodies use a Barnes-Hut tree algorithm which will be discussed later. The accelerations of a body due to the Milky Way potential use analytic equations which are described in Section 2.1.1. Therefore, we will skip the step of calculating the accelerations for each body, and just state that they have been updated. A Velocity Verlet algorithm (Verlet, 1967) is used to integrate the bodies in the simulation. The implementation of this method is described here.

For each body, i, we calculate the initial array of Cartesian accelerations, $a_{0,i}$, using the initial positions, $x_{0,i}$, and velocities, $v_{0,i}$, from rejection sampling as described in Chapter 3. Thereafter, for every timestep t, starting with $t = 0$:

$$\begin{aligned} v_{t+\frac{1}{2},i} &= v_{t,i} + a_{t,i}\left(\frac{\tau}{2}\right), \\ x_{t+1,i} &= x_{t,i} + v_{t+\frac{1}{2},i}\tau, \end{aligned} \tag{5.1}$$

where $x_{t+1,i}$ are the new positions for the next timestep. The $t + \frac{1}{2}$ terms are ‘half-timestep’ values, and are used to denote that the value in question is not the final calculated value for the next timestep. The value τ is the timestep length. The determination of the timestep

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, “Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing,” ApJ.

length is described in Section 5.4. Next the accelerations are updated using the new positions, x_{t+1}, and the half-timestep velocities, $v_{t+\frac{1}{2}}$. From that acceleration, the velocity at the next time step, $t+1$, is calculated:

$$\begin{aligned} a_{t,i} &\rightarrow a_{t+1,i}(x_{t+1,i}, v_{t+\frac{1}{2},i}), \\ v_{t+1,i} &= v_{t+\frac{1}{2},i} + a_{t+1,i}\left(\frac{\tau}{2}\right). \end{aligned} \tag{5.2}$$

The final positions, x_{t+1}, and velocities, v_{t+1}, are then used during the next time step as x_t and v_t respectively. The algorithm is implemented this way for computational simplicity. Eliminating the $t+\frac{1}{2}$ terms between the equations, we arrive at:

$$\begin{aligned} x_{t+1,i} &= x_{t,i} + v_{t,i}\tau + a_{t,i}\left(\frac{\tau^2}{2}\right), \\ v_{t+1,i} &= v_{t,i} + (a_{t,i} + a_{t+1,i})\left(\frac{\tau}{2}\right). \end{aligned} \tag{5.3}$$

This is the original form of the Velocity Verlet algorithm (Verlet, 1967). It is seen that the velocity for the next timestep, v_{t+1} is determined using the average of the accelerations of the previous step and the next step. The Velocity Verlet equations are essentially the projectile motion equations of an object in a gravitational field solved every timestep.

5.2 Simulation Units

As seen in Section 2.1, the potentials have units of $[\Phi]$ =kpc^2/Gyr^2, and not the normal units of $[\Phi]$= m^2/s^2. We also use non-standard units for the other quantities in the simulation.

We simulate dwarf galaxies orbiting and tidally disrupting in the Milky Way. The simulations indubitably describe large mass and length scales. The bodies in the simulation can represent many solar masses with millions of kilometers between them, evolving over billions of years. Their interactions use the gravitational constant which is extremely small. To avoid floating point error, or even overflow error on some machines, we adapt a system of units that can easily accommodate these masses, distances, and time scales.

We use kiloparsecs (kpc) for distance, gigayears (Gyr) for time, and kiloparsecs per gigayear (kpc/Gyr) for velocity. The velocity in km/s is a factor of 0.978 different from the velocity in kpc/Gyr. For mass, we use simulation or structural units which are derived from units where we set the gravitational constant equal to 1. Each mass unit, then has ‘units’ of:

$$[m] = \frac{[\mathrm{kpc}]^3}{[\mathrm{Gyr}]^2} = 222288.47 M_\odot. \tag{5.4}$$

Using these units for mass, we are able to represent a very massive population using small numerical values. We are not attempting to simulate individual stars that make up the dwarf galaxy; doing so would not be computationally feasible as it would require millions of bodies. We are instead representing the potential well that is formed by our initial dwarf galaxy model. The bodies represent large population of stars, acting as mass proxies, that trace the potential form. The N-body simulation, then, is a way of determining how the initial analytic potential model we use for the dwarf galaxy evolves over time as it interacts with the much larger Milky Way Galaxy potential well.

5.3 TreeCode

The total gravitational force on a body, i, due to all the other bodies, j, is given by Newton's law:

$$F_i = \sum_{j \neq i}^{N-1} \frac{G m_i m_j}{r_{ij}}. \tag{5.5}$$

The algorithmic implementation of this equation, using a brute force direct summation, scales as $\mathrm{O}(N^2)$, where N is the number of bodies in the simulation. For a large number of bodies this becomes very computationally intensive. We use, instead, an approximation of the direct summation; the gravitational force calculations between bodies are performed using a Barnes-Hut tree algorithm. This algorithm scales as $N\log(N)$ (Barnes & Hut, 1986), which is a lot less computationally intensive for large N.

The force on each body still follows Newton's law, but uses a multipole expansion. When calculating the force on body i, instead of summing over all the other bodies, j, groups of bodies are combined into one depending on their distance to i. The simulation space is divided into octant cells. If there is more than one body in an octant cell it is again divided into further 'daughter' octant cells. This continues until there is one or no bodies in the smallest octant cell of this 'tree'.

To calculate the force on body i, you start with the outermost cell, the root, and

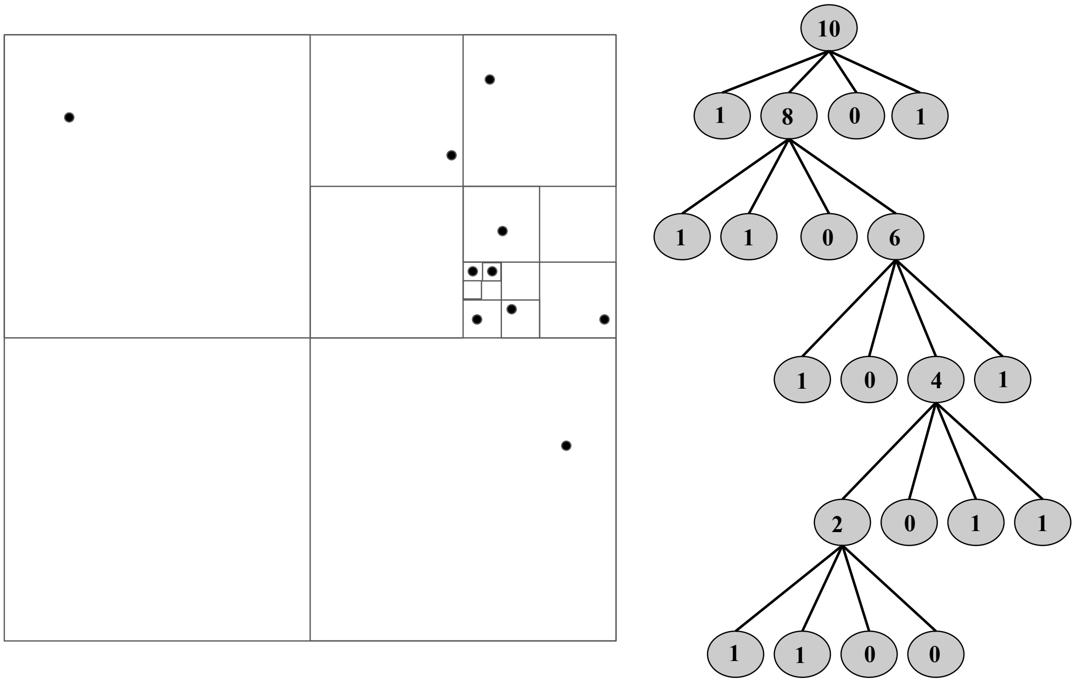

Figure 5.1: A simple two dimensional representation of a quadtree. In this case, the space is divided in quadrants, and further subdivided until there remains one or no bodies in each cell. In our simulations, the three dimensional space is divided into octants.

continue to go to smaller and smaller daughter cells. For each daughter cell, the distance, d, the body i is to the center of mass of that daughter cell is compared with the length, l, of the cell side. If the ratio of l to d is below a threshold value θ, i.e., $l/d < \theta$, then the bodies of that daughter cell are treated as a single body in the force calculation by using the total mass of the daughter cell and the center of mass location as the position. After this condition is met, the process is repeated for the other bodies. A graphical representation of a simple two dimensional quadtree is given in Figure 5.1. The formalism of the tree algorithm is given in Barnes & Hut (1986) and a description of its implementation is given Barnes (2001).

The tree code has some input parameters that determine the structure and behavior. These are ‘hard-coded’ values. The first is whether or not to use quadrupole moment terms in the force calculation. This increases the accuracy in the force calculations and is set to ‘on’. The other is the threshold value, θ, seen above, that determines how far way a group of bodies has to be before being combined. The lower the θ value, the more the algorithm resembles an exact force calculation. This value is set to $\theta = 1$ for our simulations as recommended in Barnes (2001). With this value, the brute force calculation is performed between bodies of the same cell and the approximation used for bodies of different cells if the above condition is met. The tree code is implemented as a multithreaded CPU application allowing for an even greater speed up in computation time.

5.4 Timestep Length, τ

As described in Section 5.1, the equations of motion are integrated numerically. In order to perform this integration the value of the time differential, 'dt', must be determined. This value is the timestep length, τ. During every iteration of the integration the positions and velocities for the next iteration are determined. If the moment in time for the current positions and velocities is t_i and the next moment is t_{i+1}, then the timestep length is the time between these moments, $\tau = t_{i+1} - t_i$. This is the resolution of the integration.

The forces a body experiences during the simulation can be large depending on its location in the overall potential well. The timestep length must be small enough to resolve the motion of the body under the influences of such forces, but must be large enough to not introduce errors due to numerical precision, and for the simulation to be computationally feasible; the smaller the timestep length the more calculations that are needed. An ad hoc rule is used to estimate approximately how small the timestep length would need to be in order to resolve the strongest forces for the initial dwarf progenitor based on the parameters used in its creation. For a one component model, such as a single Plummer sphere, the rule is given by

$$\tau = \frac{1}{\gamma}\sqrt{\frac{1}{\rho}} = \frac{1}{\gamma}\sqrt{\frac{4\pi}{3}\frac{a^3}{M}}, \tag{5.6}$$

where a is the scale radius, M the total mass in simulation units, and γ is a tunable parameter greater or equal to 1, which can be used to make the timestep even smaller. The value under the square root is already an overestimation of $1/\rho$, where ρ is the spherical density: it treats the dwarf as if it where homogeneous and the entire mass were within the scale radius. It is further overestimated by the tuning factor: we use $\gamma = 1000$ in our simulations, which was found to give good simulation precision without making the computation unreasonably long. Higher density means the particles will have higher forces between them and, therefore, larger accelerations. This requires smaller time-steps to resolve the particle trajectory.

Using this rule as a starting point, we adapt it to be compatible with a dwarf model that has two components, each of which has a scale radius and total mass. Unless the two components have the same scale radius, the dwarf galaxy will have some semblance of a core and a halo: a more compact inner part with a more diffuse outer part. We perform a similar overestimation as in Equation 5.6 for each possibility, the core being made up of baryons or dark matter. We calculate two parameters, τ_1 and τ_2:

$$\tau_1 = \frac{1}{\gamma}\sqrt{\frac{4\pi}{3}\frac{a_b^3}{(M_B + M_{encl,d})}}, \tag{5.7}$$

and,

$$\tau_2 = \frac{1}{\gamma}\sqrt{\frac{4\pi}{3}\frac{a_d^3}{(M_D + M_{encl,b})}}, \tag{5.8}$$

where a_b and M_b are the scale radius and total mass of the baryonic component, and $M_{encl,d}$ is the theoretical dark matter mass enclosed within the baryonic scale radius. Similarly, a_d and M_d are the scale radius and total mass of the dark matter component, and $M_{encl,b}$ is the theoretical baryonic mass enclosed within the dark matter scale radius. The mass enclosed for a given radius (r), scale length (a), and total mass (M) is given by:

$$M_{encl} = \frac{Mr^3}{(a^2 + r^2)^{-\frac{3}{2}}}. \tag{5.9}$$

The value of γ is the same as with the single component equation. The smaller of the two parameters is used as the timestep length. Although Equations 5.7 and 5.8 were developed with a core and halo in mind, they still provide a good overestimation of $1/\rho$ when the two components have the same scale radius and/or if they have the same mass. The values are equal, $\tau_1 = \tau_2$, when the two components have the same parameters, in which case the first is used.

The timestep length should have units of time. This is not clear when looking at the equations. However, when using the units for mass as described in Section 5.2, the values have units of gigayears:

$$[\tau] = \sqrt{\frac{[\mathrm{kpc}]^3}{[\mathrm{sim\ units}]}} = \sqrt{[\mathrm{kpc}]^3\frac{[\mathrm{Gyr}]^2}{[\mathrm{kpc}]^3}} = [\mathrm{Gyr}]. \tag{5.10}$$

5.5 Softening Parameter, ϵ^2

The bodies in the simulation each have the gravitational potential well of a point particle. When the bodies are given positions and velocities, it is the overlapping of the individual potential wells that create the potential well of our dwarf model.

The individual potential wells can lead to strong interactions when bodies get close to each other. This could lead to rapid decay of the simulated dwarf galaxy as particles fly off at extremely high, and unrealistic, velocities. These strong interactions would occur with much lower frequency in a real dwarf galaxy, which has many more, much smaller particles. In addition, the strong interactions might not be resolved by the timestep in the simulation. To reduce the effects of strong interactions, the force is 'softened' by removing the singularity at $r = 0$ of Newton's law of gravity, thus truncating the bodies' individual potential wells. This is done by adding a 'softening parameter,' ϵ, to the distance between bodies. The force between two bodies is then

$$F_{ij} = \frac{GM_iM_j}{r_{ij}^2 + \epsilon^2}. \tag{5.11}$$

Therefore, at a distance of $r_{ij} = 0$ the force remains finite. A simple way of interpreting this method is that it makes the bodies into hard spheres. All that remains to do, then, is to determine an appropriate value of ϵ^2. We require the softening parameter to be smaller for denser arrangements of bodies, making them 'smaller spheres', and allowing for more accurate, but still finite, force calculations. Removing the singularity in each individual's potential well also has the overall effect of smoothing out the theoretical potential well being represented as all of the singularities are removed.

For a single component model, we calculate the following value:

$$\epsilon^2 = \frac{a^2}{\beta N}, \tag{5.12}$$

where β is a tunable parameter (we use $\beta = 100$) used to reduce the softening length, N is the number of bodies in the simulation, and a is the scale length. For a two component model we have two scale lengths. So we use a 'center of mass' scale length determined by:

$$a_{cm} = \frac{M_b a_b + M_d a_d}{M_b + M_d}, \tag{5.13}$$

where M_b, a_b, M_d, and a_d are as defined in Section 3.1. This method is remarkably simple and easy to implement, and does not require that we constantly check whether bodies are too close in every iteration.

5.6 Random Number Seed

As a result of the rejection sampling technique used, a large quantity of random numbers is needed when creating the dwarf galaxy. These numbers are determined using a random number generator (RNG). The RNG requires a seed to begin. This is normally either provided by the user or manually set in the parameter files. If one is not provided then the system time of the computer being used is taken as the random number seed.

Currently, the random number seed is set in the parameter file, and is fixed for all parameters for a given optimization. This is necessary because of the distributed nature of MilkyWay@home. We require all simulations with identical parameters to create identical dwarf galaxies and thus produce identical results. This is especially necessary because of the validation technique used by MilkyWay@home described in Chapter 7 of Desell (2009): results that improve the search are repeated, sometimes by other users. If the seed was not fixed, the system times of the different volunteer machines would be used. For the same set of parameters, different results would be sent back to the server and results would not be properly validated. In that case, there would be no guarantee that the optimization algorithm would ever converge.

Using the same seed does not reduce the random nature of the simulations. While the same list of random numbers is produced, these numbers could be used to sample different quantities depending on the parameters for the dwarf galaxy. It is true, though, that the simulation would begin with identical dwarf galaxies for the special case where the simulation time is the only thing being altered.

5.7 Outputs

The dwarf galaxy tidally disrupts during the orbit, leading to a tidal stream of bodies centered at the approximate location of the starting coordinates. There are two possible outputs given by the simulation. The first is the complete list of each body's characteristics. These include the phase space coordinates, the mass, the line-of-sight velocity relative to the Sun, and whether or not the body is baryonic or dark matter. There is also a unique body identifier number for each body to allow for the tracing of individual bodies throughout the simulation. By using different simulation input flags, the user can choose the output coordinates for phase space. The choices are (l,b,R), which are Solar centered, Cartesian, which are Galactic centered, or both. The flags associated with these will be described in Appendix A. These are most useful when performing simulations in a local setting, meaning not on MilkyWay@home. It allows us the means to make plots of the bodies, and perform various calculations and tests.

The other output is the histogram of the body distribution at the end of the simulation. The main histogram is one of normalized counts, and the associated error, along the stream. The other components are the β dispersions and the line-of-sight velocity dispersions in each bin, with their associated error. The creation of this histogram, the calculation of the dispersions, and errors will be described in Chapter 6.

CHAPTER 6
Metric for Comparing Results of Simulation to Data

As described in Section 5.7, there are two types of simulation outputs: the phase space coordinates of each body and the histogram. The histogram output characterizes the stream properties as a function of angle along the simulated stream, and provides a basis for comparison with the data drawn from astronomical surveys. This output is produced when clients run the simulations for a particular set of parameters on MilkyWay@home. Since the dark matter in tidal streams cannot be observed, we compare the observed stars with only the stars in the simulated tidal stream. The *likelihood* that these two distributions are the same guides our search for dwarf galaxy parameters; each set of initial parameters produces a different density distribution along the stream and thus a different likelihood. Our comparison method is described in this chapter.

The metric used for comparison is comprised of three components, each of which account for different physical aspects of the stream. The first component focuses strictly on the density of stars as a function of angle along the stream. The second focuses on the stellar mass that each data set represents. The final component looks at a histogram of the width of the stream as a function of angle along the stream.

6.1 Creating Histograms of Density and Stream Width Along the Stream

Before a comparison can be made, we must make a histogram from the simulated tidal data. As we have said, only the baryonic bodies are included in the histogram, though they are influenced in the simulation by the unseen dark matter. At the end of the simulation, each body is represented in Galactocentric Cartesian coordinates and must be converted, first to Solar-centered Cartesian coordinates, and then into angular coordinates with the equator of the system aligned along the tidal debris stream under investigation. This type of coordinate system is known as (Λ, β), or in this case (Λ_{Orphan}, β_{Orphan}). They are a rotation of the standard (l, b) coordinate system, with transformation rules for the Orphan Stream

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, "Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing," ApJ.

defined in Newberg et al. (2010). The rotation angles, unique to the stream in question, are given by: $(\phi, \theta, \psi) = (128.79^o,\ 54.39^o,\ 90.70^o)$ (Newberg et al., 2010). In this system, Λ is parallel to the stream and β is perpendicular to the stream. For a complete description of this transformation see Appendix D.

A region of the sky is selected in which the bodies will be analyzed. This region can be narrow or encompass the entire sky. Normally, the region is kept narrow in β to avoid capturing multiple wraps of the stream, and to avoid particles that have been kicked out of the dwarf galaxy and are not associated with the stream. The region in Λ is kept wide for simulated data, encompassing almost the entire sky. Depending on the availability of stellar data, the range is narrowed to avoid including bins without any observational information.

In this region, data from both the observations and the simulation are binned in Λ, to represent the density of stars as a function of angular position along the stream. The total number of bodies that fall within the entire region, N_T, is also noted. The count and error in each bin is given by $N_i \pm \sqrt{N}$. After the binning is finished, the count in each bin is normalized so that the sum of all the histogram bins is one:

$$\frac{N_i}{N_T} \pm \begin{cases} \frac{\sqrt{N_i}}{N_T} & \text{if } N_i \neq 0 \\ \frac{1}{N_T} & \text{if } N_i = 0 \end{cases}. \tag{6.1}$$

We also characterize the width of the stream as a function of Λ. We have tried two different widths, the line-of-sight velocity dispersion and the dispersion in the β coordinate. The larger the dispersion, the wider that part of the stream. We will show that either is successful for constraining the progenitor mass, but we currently use the β dispersion because the observational data for this quantity is more readily available. The algorithm can handle either, along with their respective errors. Both values are included in the histograms, the line-of-sight velocity dispersion and the β dispersion, with their respective errors. The line-of-sight velocity is determined from the positions and velocities of a body:

$$v_{LOS} = \frac{\vec{x} \cdot \vec{v}}{|\vec{x}|}, \tag{6.2}$$

where the positions are given in Solar-centered Cartesian. The dispersion in some quantity, α, whether it be line-of-sight velocity or β, is given by:

$$\sigma^2 = \frac{1}{N-1}\sum_{i}^{N}(\alpha_i - \bar{\alpha})^2, \tag{6.3}$$

where α_i is the quantity for body i in a given Λ bin, $\bar{\alpha}$ is the average of that quantity for the entire bin, and N is the number of bodies in the bin. The normalization factor is $N-1$ and not N because the average is calculated from the same population. This can be written as (for a derivation, see Appendix B):

$$\sigma_\alpha^2 = \sum_i \frac{\alpha_i^2}{N-1} - \left(\frac{N}{N-1}\right)\left(\sum_i \frac{\alpha_i}{N}\right)^2. \tag{6.4}$$

Using this equation allows for greater computational efficiency; instead of having to calculate the average in each bin, which would involve iterating over the entire body list several times, we instead keep a running sum of α and α^2.

These sums are first calculated for a bin and then updated by several rounds of outlier rejection. Outliers are rejected until all bodies greater than 2.5σ from the average quantity, $\bar{\alpha}$, for that bin are removed. The average is then recalculated by subtracting the values of α and α^2 for the rejected bodies from the running sum. We also reduce N, as each outlier is removed.

The value 2.5σ was chosen as the cutoff because it cut out the most extreme outliers without affecting the characterization of the stream width. Before outlier rejection was implemented, a handful of outliers would greatly inflate the dispersion. This cutoff was found to be enough to eliminate those outliers while still including the main portion of the stream. As will also be described in Appendix A, this value can be adjusted at runtime.

Performing outlier rejection reduces the calculated dispersion of a distribution even if it is a perfect normal distribution, so we multiply σ_α^2 by a correction factor after each iteration. The correction factor is calculated assuming the underlying distribution is Gaussian:

$$\sigma_{data}^2 \cdot \left(\frac{\sigma_{guassian}^2}{\sigma_{truncated}^2}\right) = \sigma_{data}^2 \frac{\int_{-\infty}^{\infty} x^2 e^{-\frac{x^2}{2\sigma^2}} dx}{\int_{-2.5\sigma}^{2.5\sigma} x^2 e^{-\frac{x^2}{2\sigma^2}} dx} = \frac{\sqrt{2\pi}}{2.25581}\sigma_{data}^2 = 1.11\sigma_{data}^2. \tag{6.5}$$

The correction factor depends on the cutoff (in this case 2.5), but does not depend on σ_{data}. This correction factor is applied after every round of outlier rejection. Currently, the algorithm performs six rounds of outlier rejection, which was found to be enough to remove the most extreme outliers for both methods; it is computationally quicker to reject outliers six times rather than testing to determine whether the algorithm has converged. This value can also be altered from the parameter files (see Appendix A) at runtime.

After this process is complete, the error in the final dispersion value is also calculated. For the simulation, this value can be derived analytically and is given by

$$\sigma \pm \delta_\sigma = \sigma \pm \sigma \sqrt{\frac{N+1}{N-1}\frac{1}{N}}, \tag{6.6}$$

where N is the remaining number of bodies not rejected as outliers. The derivation of this equation can be found in Appendix C. For each bin, the histogram contains the line-of-sight velocity dispersion and β dispersion, both with outliers rejected, and with their respective theoretical errors. The user can choose to use both methods, one of the two, or neither, from the parameter files. For the simulation histogram, the dispersions and errors are calculated for each bin. For the stellar data histogram (to be discussed in Section 7.2) we may not have enough data to calculate dispersions in each bin. The bins without dispersions are given a default value to denote the absence of data. When performing the histogram comparison, these bins are skipped.

6.2 Stellar Density Component of the Likelihood

We compare two stellar density distributions using an Earth Mover Distance (EMD, Rubner et al. 2000) technique. This method is preferable to a bin-by-bin comparison such as the χ^2 method because it avoids situations where small differences between the histograms result in huge differences in the goodness of fit. The χ^2 method is given by

$$\chi^2 = \sum_{i=1}^{n} \frac{(\rho_{data,i} - \rho_{sim,i})^2}{\rho_{sim,i}}, \tag{6.7}$$

where $\rho_{sim,i}$ are the simulation histogram bins, $\rho_{data,i}$ are the observed data histogram bins, and χ^2 is the value being minimized. The chi-squared method compares each bin separately and then sums their difference. This does not provide a good comparison metric because the histograms represent stellar density along the stream in a fixed area of the sky, for a certain evolution time.

The location of the progenitor core will naturally have a higher peak than the surrounding stream. Consider two streams, one evolved for some time, T, and another evolved for slightly longer, $T+\Delta t$, where Δt is just large enough for the progenitor core (in this case much more densely populated than the surrounding stream) to be counted in a neighboring bin. Although these two streams would be overall very similar, performing a bin by bin comparison would yield a poor likelihood value because there is no benefit to the highly populated bins being close to each other. In fact, in the unrealistic case of the core location shifting to a neighboring bin being the only change between the two histograms, if the core was comprised of N_1 bodies initially, the neighboring bin had N_2 bodies initially, and ϵ bodies from the core bin shifted to the neighboring bin, the χ^2 value as in Equation 6.7 would change from exactly χ^2=0 initially, to χ^2= $\frac{(N_1+N_2)}{N_1 N_2}\epsilon_i^2$. For large values of N_1, which would be expected for the location of the progenitor core, and for large values of ϵ which would also be expected if the entire peak is shifted to the neighboring bin, the χ^2 value is large. Furthermore, the reported χ^2 is the same if this shift were to occur between bins that were further apart (all other bins remaining the same). Therefore, this method would not weigh a slight shift to a neighboring bin with a better likelihood than a shift to a bin further away. More realistically, many more bins would have bodies shift into neighboring bins, meaning an even larger χ^2 value, and a poorer likelihood. Such a huge comparison value for such a small overall change makes the likelihood surface difficult to traverse.

An Earth Mover's Distance (EMD) calculation resolves this issue by providing a method that compares the overall shape of the two histograms instead of a bin by bin comparison. The method essentially asks how much deformation has to be applied to one histogram to arrive at the other. It is calculated as a transportation problem, determining the minimum amount of work needed to perform this deformation (Rubner et al., 2000). If $\rho_{sim} = \{(\Lambda_1, N_1), ..., (\Lambda_m, N_m)\}$ are the bin centers and counts in one histogram and $\rho_{data} = \{(\Lambda_1, N_1), ..., (\Lambda_n, N_n)\}$ are the bin centers and counts in the other histogram, the 'work' to perform the deformation is given by:

$$W(\rho_{sim}, \rho_{data}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}, \tag{6.8}$$

where f_{ij} is a flow matrix and d_{ij} is a matrix of the ground distances between bins (Rubner et al., 2000). This value, $W(\rho_{sim}, \rho_{data})$, is minimized until an optimal flow matrix is found. The EMD value is defined in terms of the optimal flow matrix and the ground distance matrix as follows:

$$EMD(\rho_{sim,i}, \rho_{data,i}) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}. \tag{6.9}$$

This method is explained in detail in Rubner et al. (2000). The probability returned from the EMD calculation is scaled and returned as the stellar density portion of the likelihood value.

$$P_{shape} = \begin{cases} 1 - \frac{EMD}{EMD_{max}} & 0 < EMD < EMD_{max} \\ 0 & otherwise \end{cases}, \tag{6.10}$$

where $EMD_{max} = 50$ is a scaling factor so that for reasonable configurations of bodies EMD/EMD_{max} ranges from zero to one.

The two histograms being compared must have the same normalization so that it is not necessary to create or destroy bodies to turn one histogram into the other, which would make us dependent on the placement of a "source" or "sink" of bodies. We normalize each histogram so that the sum over all bins is one. However, by doing this, information on the number of bodies, and subsequently the stellar mass represented in each histogram, is lost. To penalize the likelihood when the total mass of stars does not match the total mass of the stellar bodies, we use a mass cost function. While EMD compares the shapes of the two histograms, the mass cost compares the total mass each histogram represents

6.3 Mass Cost Component of the Likelihood

The amount of mass represented in each histogram is determined by the mass per baryonic body and the number of bodies included in the histogram. Unless the entire sky is used when making the histogram, something that is generally not done, the number of bodies included in the histogram can vary between simulations with different parameters; some bodies may be excluded because they fall outside of the (Λ, β) bounds of the histogram.

We introduce a penalty to the likelihood if the mass represented by the data histogram differs from the mass represented by the simulation histogram. Note that each body in the simulation represents a fixed mass of stars, M_{sim}, equal to the total mass of the baryonic component divided by the number of baryonic bodies. Similarly, each observed star in the data histogram represents some mass of stars, as traced by the type of star that is observed. The mass of each "star" in the data histogram is inflated by how many solar masses each tracer star represents, in solar masses. The total mass in the simulated histogram is $M_{sim}N_{sim}$, and the total stellar mass represented by the data histogram is $M_{data}N_{data}$. We then determine the number of standard deviations, σ, by which these two integrated mass estimates differ. N_σ is found by dividing the difference between the two masses by the sum of the two errors, added in quadrature. The N_σ that two values with error, $x_1 \pm \sigma_{x_1}$ and $x_2 \pm \sigma_{x_2}$, are different is given by:

$$N_{\sigma_{x_1-x_2}} = \frac{x_1 - x_2}{\sqrt{\sigma_{x_1}^2 + \sigma_{x_2}^2}}. \tag{6.11}$$

The denominator is derived using the standard method of error propagation when adding or subtracting two values with error.

The errors in the masses represented by each histogram can be determined by considering how the two histograms are made. While they are both counting experiments with two outcomes, an object is either included or not, they differ in that the simulation has a finite number of objects that can be included in the histogram. The total mass in the simulation histogram is determined by the mass represented by each object and the total number of objects. Since we assign the mass per object in each histogram, we consider it to be a constant with no error. For the simulation histogram, then, the error in the total object count is the primary source of error in this calculation.

For the stellar histogram, the number of stars in the population associated with the stellar stream is not exactly known but is known to be very large. The objects in the stellar histogram are tracer stars, each of which represent a population of stars not included in the data or histogram. The measured values are, again, the masses in each histogram, determined by the population mass represented by each object and the total number of objects. The stellar population mass represented by each object in the histogram is assumed to be the same. This means that a constant, the population mass represented by each tracer star, will need to be applied to scale the two histograms; the number of tracer stars included in the stellar histogram is multiplied by this constant. The determination of this constant will be discussed in Section 7.1 when discussing the creation of a stellar histogram from available data. For the stellar histogram, the errors in the total object count is again the primary source of error.

Because the number of bodies in the simulation is fixed, the $N/2$ baryonic bodies in the simulation, and the number in the part of the sky that is "observed" is selected from this maximum, the error in the counts for a simulation histogram is given by binomial statistics:

$$N_{sim} \pm \sqrt{N_{sim}\left(\frac{N_{sim}}{N_{total}}\right)\left(1-\frac{N_{sim}}{N_{total}}\right)}, \tag{6.12}$$

where N_{sim} is the number of bodies in the simulation histogram, and N_{total} is the number of bodies in the simulation (baryonic component), i.e., the limit in the number of bodies that can be in the histogram.

Unlike the simulation histogram, there is no upper limit on the number of stars that can appear in the data histogram. The stars included in the data histogram is limited only by the data available from astronomical surveys. While there are a finite number of stars that are associated with the tidal stream, this number is so large compared to the number in our simulations that it can be approximated as a continuous domain. The error in the counts for a histogram made from stellar data is thus given by Poisson statistics, and will depend on the method for producing the data histogram as well as the background, if any. For a simple case without background, the error is given by,

$$N_{data} \pm \sqrt{N_{data}}. \tag{6.13}$$

Putting the error values into Equation 6.11, and multiplying the counts and their respective error by the mass per object, the N_σ difference in the mass is given by:

$$N_\sigma = \frac{M_{sim}N_{sim} - M_{data}N_{data}}{\sqrt{M_{data}^2 N_{data} + M_{sim}^2 N_{sim}\left(\frac{N_{sim}}{N_{total}}\right)\left(1-\frac{N_{sim}}{N_{total}}\right)}}. \tag{6.14}$$

This equation is then plugged into a probability density describing whether the mass within the two histograms are different by chance, given by:

$$P_{mass} = e^{-\frac{N_\sigma^2}{2}}, \tag{6.15}$$

where N_σ is the number of σ the mass between the two histograms are different. This gives the probability that the two histograms represent the same mass. The "most similar" that the two histograms can be is identical, with a $N_\sigma = 0$. The maximum probability is one, giving it the same scale as the EMD component, by construction. This can then be combined with the stellar density component. Together, these two methods characterize the density distribution along the stream and the mass each represent.

6.4 β Dispersion Component of the Likelihood

In addition to using the density of stars along the stream to constrain the model parameters, we also use the stream width. Historically, streams have been separated by likely dwarf galaxy and likely globular cluster progenitors by the width of the tidal stream produced, suggesting this width might be a powerful discriminant of progenitor mass.

At first, we tried using a 2D Earth Mover Distance method to compare the density distribution in the Λ and β directions, but this was time-consuming and did not work when the parameters caused the stream to shift in the β direction. Shifts in the β direction could indicate that the orbital parameters need to be adjusted, but we do not currently optimize the properties of the progenitor and the orbit of the progenitor simultaneously.

We then tried a histogram of the line-of-sight velocity dispersion, which worked well for simulated data but failed on the real data because the velocity errors were too large and the number of stars with spectra too small to measure the parameters of the Orphan Stream progenitor. Therefore, a histogram of the spatial dispersion in the β coordinate, binned in the same Λ bins as the number density used for the EMD, is used to compare the data with the simulations.

Similar to the cost component, we calculate a σ difference between the width histograms. We calculate the N_σ between individual bins between the two histograms, using a similar probability density to that given by Equation 6.15.

The σ difference is between the bins' dispersion value. The total probability that the two histograms have the same stream widths is given by the multiplicative series of each bin's probability,

$$P = \prod_i P_i = \prod_i e^{-\frac{N_{\sigma,i}^2}{2}} = e^{-\frac{1}{2}\left(N_{\sigma,1}^2 + N_{\sigma,2}^2 + \ldots + N_{\sigma,N_{bins}}^2\right)} = e^{-\frac{N_\sigma^2}{2}}, \tag{6.16}$$

where the N_σ^2 in the exponent is the sum written as:

$$N_\sigma^2 = \sum_i^{N_{bins}} \frac{(\sigma_{\beta,data,i} - \sigma_{\beta,sim,i})^2}{(\delta^2_{\sigma_{\beta,data,i}} + \delta^2_{\sigma_{\beta,sim,i}})}. \tag{6.17}$$

where $\delta_{\sigma_{\beta,sim}}$ is the error from Equation 6.6. This error value is also used for $\delta_{\sigma_{\beta,data}}$, when comparing the simulation against a simulated data set. For actual data, the error value is determined using a method to be described in Section 7.2. For the simulated histograms, we have width information for every bin. This is not the case for the stellar histogram. Therefore, only bins with width information are included in this sum. The probability density for the width is then,

$$P_{width} = e^{-\frac{N_\sigma^2}{2}}. \tag{6.18}$$

This is the same probability density distribution used for the mass cost component.

We previously used a χ^2 probability distribution for this component of the likelihood, but abandoned it in favor of the previously described method. For a full description of the previous method, see Appendix E.

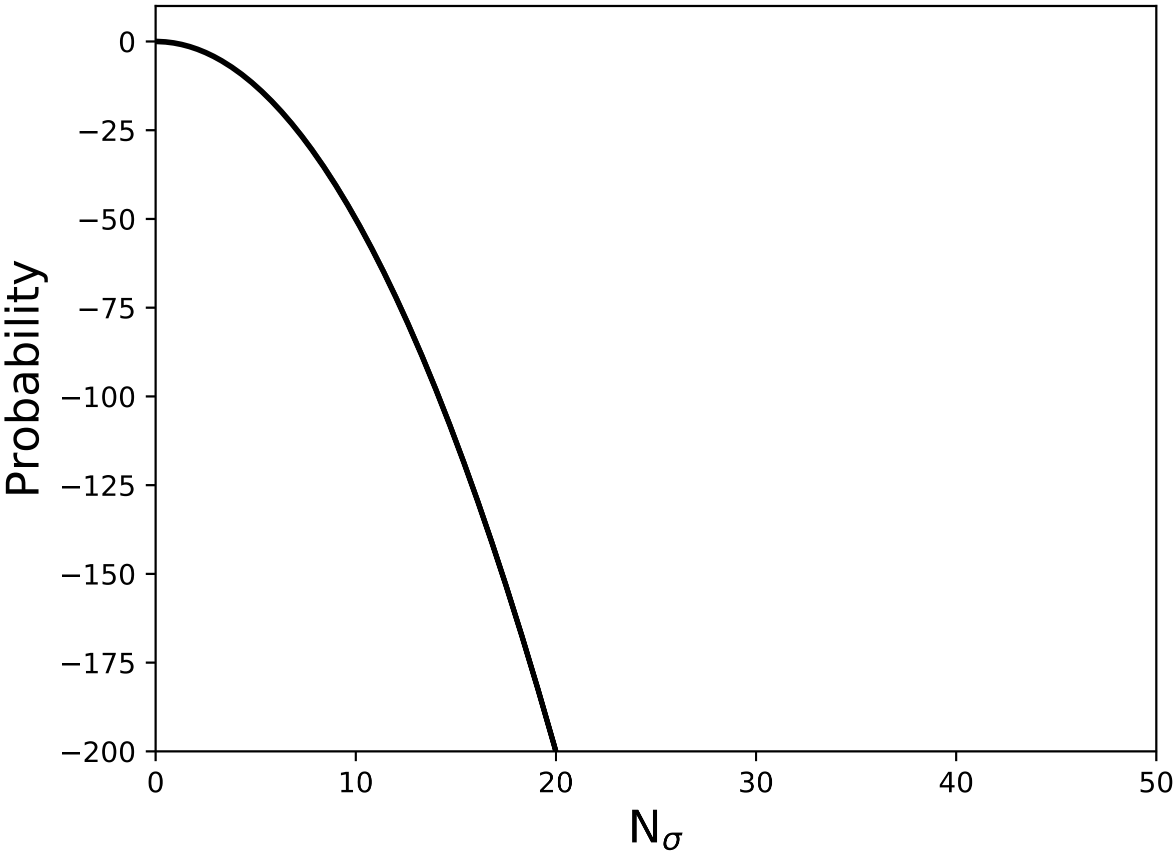

Figure 6.1: The probability function used for both the mass cost and the dispersion as a function of the number of σ difference, N_σ. For the mass cost, this is the σ difference between the mass represented by each histogram. For the dispersion, it is the square root of the cumulative σ difference between the dispersion of each corresponding bin between the two histograms given by Equation 6.17.

6.5 Final Likelihood

Together, the three components described above allow for a comparison of the tidal debris stream represented by two sets of histograms. Each component is combined into a final probability given by their product:

$$\mathcal{L} = \prod_{i} P_i. \tag{6.19}$$

Or, in log space:

$$\ln\left[\mathcal{L}\right] = \sum_{i} \ln(P_i) = \ln(P_{geometry}) + \ln(P_{cost}) + \ln(P_{dispersion}). \tag{6.20}$$

We note that we have normalized the individual terms in the probability density distribution so that they range from zero to one. The maximum probability for each component in log space is zero; for identical histograms, $\ln[\mathcal{L}] = 0$. Any different normalization would produce additive constants in log space. Therefore, normalizing the component probability densities amounts to a shift in the likelihood surface and not a change in the shape. We prefer to keep the maximum likelihood at zero.

The calculation of $\ln[\mathcal{L}]$ provides a metric with which to measure the similarity of the two histograms and thus the stellar distributions they represent. Our model parameters are constrained by finding the parameter set for which the likelihood, $\ln[\mathcal{L}]$, is a maximum. Depending on what combination of the β and line-of-sight velocity dispersion is used, the above equation could have a fourth term.

6.5.1 Best Likelihood Determination: Reducing Chaotic Behavior

Our method for navigating the likelihood surface to find the best-fit parameters relies on the assumption that small changes in the initial parameters produce small changes in the resulting histograms, so that the likelihood surface is fairly smooth. Unfortunately, chaos can contradict this assumption, and in particular we found that small changes to the initial conditions, including changing the random seed, can change the final position of the dwarf galaxy along the orbit, and therefore cause an unacceptably large change in the likelihood. If the forward evolve time is held constant, the location of the peak of the distribution of stars would fluctuate back and forth along Λ as each dwarf galaxy parameter is changed, making it difficult to find the peak likelihood.

As described in previous sections, the optimizer currently fits five parameters: four dwarf parameters and the evolution time. One particle is moved backwards along the orbit for the evolution time, and we run the simulation in the forward direction for as long as it takes for the progenitor to get to the current position, as determined by the time to get to the maximum likelihood. At the beginning of the simulation, the number of time steps in the simulation is determined based on the timestep length and the simulation time. At some percentage of the total number of time steps, and for every subsequent time step, the simulation will create a histogram from the current state of the simulation during that time step, compare with the input histogram and calculate a likelihood. We save the best likelihood value from all of the steps tested, along with the associated histogram. The percentage value is an input parameter, and is currently set to 98 percent, so the program searches for the maximum likelihood for a window of forward evolution times between 98% and 100% of the evolution time parameter.

CHAPTER 7
The Progenitor of the Orphan Stream

7.1 Stellar Mass Represented by Orphan Stream Tracer Stars

The Orphan Stream has been chosen as an example due to the interesting property of its unknown progenitor, and the access to stellar density data from Newberg et al. (2010). Multiple possibilities for the progenitor have been proposed, including SEGUE-1 (Newberg et al., 2010), Complex A (Jin & Lynden-Bell, 2007), UMa II (Belokurov et al., 2007). However, arguments in Newberg et al. (2010) are convincing in rejecting these possibilities and they suggest that an overdensity around (l,b) = (270,30) at the edge of the SDSS data in the region could be the remains of the progenitor.

We have turnoff star data from Newberg et al. (2010) which allow us to trace the stream density across the sky. Each turnoff star represents a population of stars with a wide range of masses that we do not see because they are too dim, or because they have evolved away from the main sequence. To begin creating a histogram of Orphan stream data, we must determine the mass represented by the turnoff stars we observe. Our method of doing this is to use a globular cluster with similar stellar population. Using turnoff star data from that cluster and the total mass of that cluster, we can estimate how much mass each turnoff star represents. We then make a similar assumption as in Savage et al. (2006), that since the stellar population is similar to the Orphan Stream, the turnoff stars should represent similar masses. The Orphan stream has a turnoff color of $(g\text{-}r)_0 = 0.22$. Palomar 5 has a similar turnoff magnitude, so we make it our globular cluster of choice. The total mass of Pal 5 is taken as approximately $5{\times}10^3 M_\odot$ (Savage et al., 2006). We select a total of 3253 stars in the sky location of Pal 5 from SDSS DR14 (Abolfathi et al., 2018), using the CasJob query shown below. We make a color cut in the range $0.12 < (g\text{-}r) < 0.26$, as this was the color cut used in Newberg et al. (2010) for the Orphan Stream turnoff star selection. We also apply a magnitude cut of $19 < g < 23$.

Portions of this chapter are in preparation to be submitted to: S. Shelton, “Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing,” ApJ.

Table 7.1: CasJob query used to select Palomar 5 stars.

CasJob Query
select star.objid, star.ra, star.dec, star.dered_u, star.dered_g, star.dered_r, star.dered_i, star.dered_z into mydb.MyTable_7 from star, dbo.fGetNearbyObjEq(229.0256,-0.12415,10) n where star.objid = n.objid

After the cuts, 352 turnoff stars were selected. The full data set is shown in Figure 7.1, with the selected turnoff stars shown in blue.

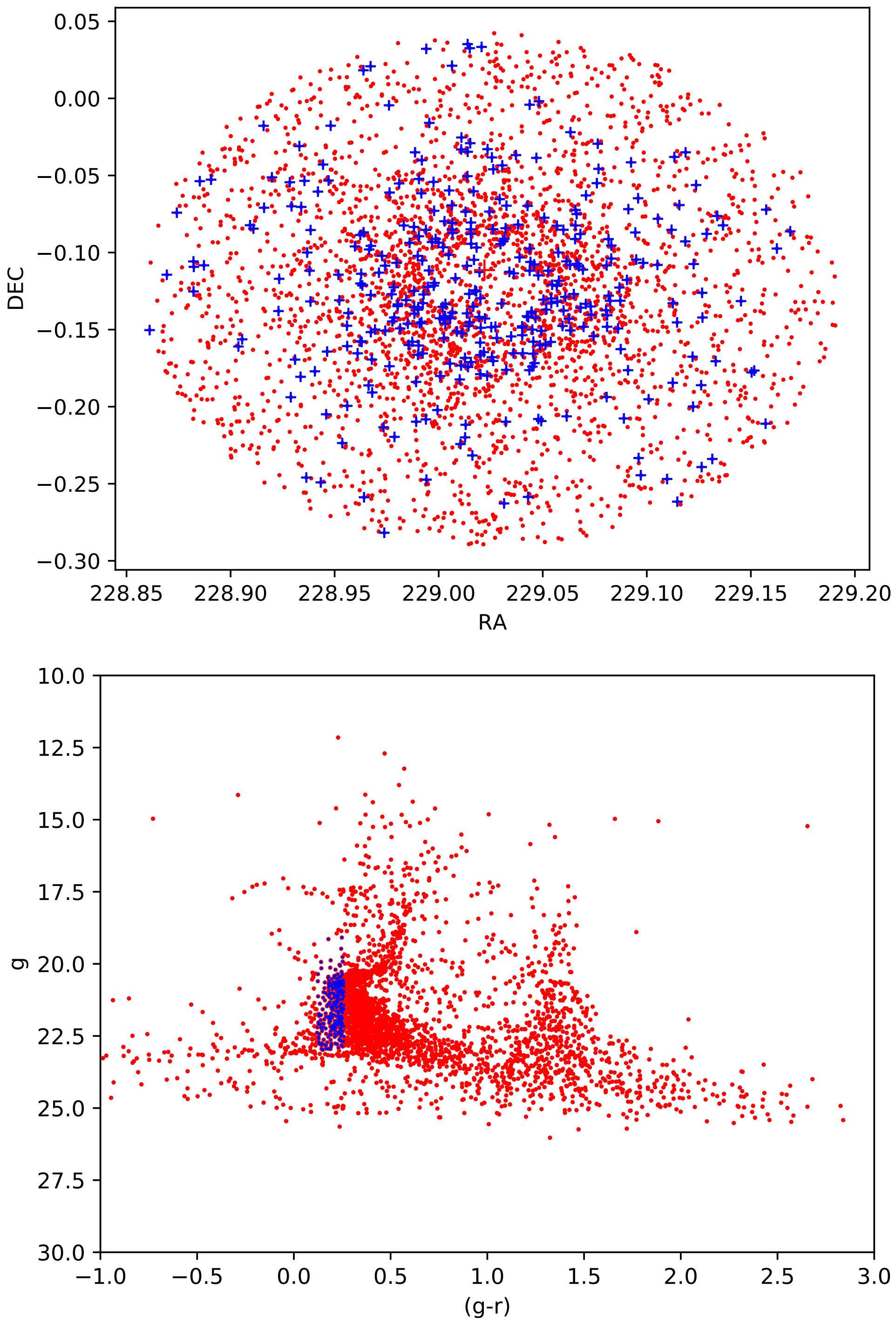

Figure 7.1: Top: Right ascension (RA) and declination (DEC) of the full data selection. The selected turnoff stars are shown in blue. Bottom: Color magnitude diagram of the full data selection. The selected turnoff stars are again shown in blue.

As the plot of RA and DEC show, there is an apparent underdensity in the center of the data set. This is because the cores of globular clusters are too densely populated with stars for the deblending algorithm to pick out each individual star. Before estimating the mass per turnoff star, we first estimate how many stars are missing from the data set. We fit a Plummer model to the data, with the radius given by the distance to the cluster center, taken as (RA,DEC)= (229.022^o, $-.11139^o$). The data is a two dimensional projection of the globular cluster on the night sky, so we use a Plummer surface mass density model. This is given by:

$$\Sigma(R) = \frac{Ma^2}{\pi(a^2 + R^2)^2}, \tag{7.1}$$

where M is the total mass, and a is the scale length. Since this is a surface mass density, the scale length is converted from the core radius, not the half-mass radius, for Pal 5. The core radius is 2.29 arcmin, or 0.0382^o. The mass enclosed in this model is given by:

$$dM_{enclosed} = 2\pi\Sigma(R)RdR. \tag{7.2}$$

Integrated from the center to radius R, this becomes,

$$M_{enclosed}(R) = \frac{MR^2}{a^2 + R^2}. \tag{7.3}$$

Since the color selection was similar to Savage et al. (2006), we use the same expected background of 32 stars, which is added to the total turnoff star count. These equations are plotted in Figure 7.2, along with the corresponding stellar values.

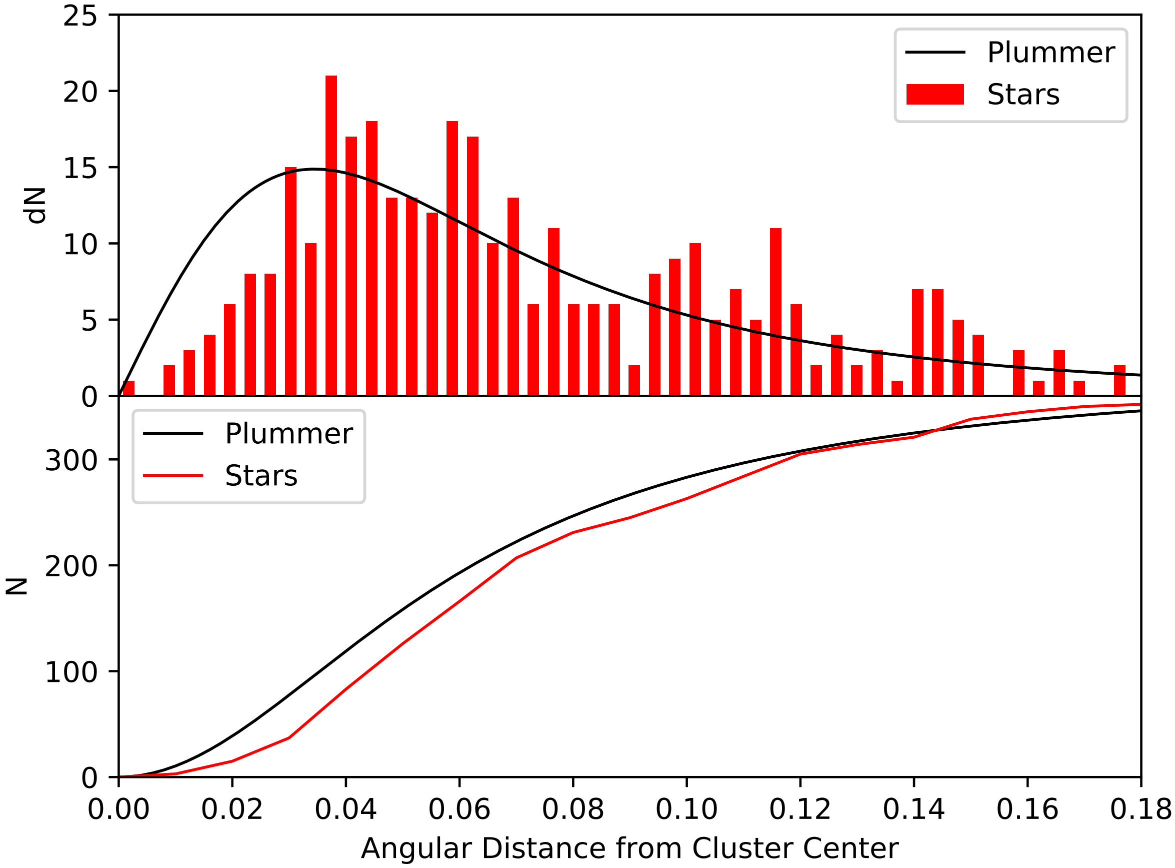

Figure 7.2: In the upper panel is a plot of the mass enclosed function of the Plummer surface density, with a histogram of the turnoff star counts. Below is a plot of the total mass enclosed as a function of distance from the cluster center of the Plummer model and the turnoff stars. This is with an added expected background of 32 stars. As both panels show, there is a count discrepancy near the core of the cluster.

Looking at the count difference between the stellar data and the model, it was determined that there were approximately 30 turnoff stars missing from the core. Dividing the total mass by the then 382 turnoff stars gives that each turnoff stars represent about 13$M_\odot$.

7.2 Creation of a Data Histogram

We use detections from the Sloan Digitial Sky Survey (SDSS) of turnoff stars in the region of the sky corresponding to the location of the Orphan stream. The $(\Lambda,\ \beta)_{Orphan}$ coordinates for the stars in both the On and Off fields, as described in the Newberg et al. (2010), were compiled. We then created a density distribution histogram of counts along the stream, and determined the associated error in those counts using standard error propagation. After some extensive analysis, we derive β dispersions and their associated errors for four points along the stream.

As stated above, the software allows either line-of-sight velocities or β dispersions to be used in combination with the density histogram to constrain the properties of the dwarf galaxy progenitor. Initially, the line-of-sight velocity dispersion was used successfully with simulated data. However, the line-of-sight velocity errors in the available data for the Orphan Stream were too large, and the number of stars with measured velocities was too small to get useful results with the real data. Instead, we use β dispersions, for which we have much more data. As will be described in Appendix A, the user can choose to use one or both of the dispersion methods in the parameter files for the simulation. This was done with an eye towards more complete data that will become available from future astronomical surveys. Currently, data from the Gaia mission is becoming available and could possibly allow the use of the line-of-sight velocity dispersion method in conjunction with the β dispersion method. Using both would add an extra constraint on the stream width and may aid in optimization.

7.2.1 Recreating the Data

We first recreate the histogram data from Newberg et al. (2010). The β coordinates are corrected as described in Section 6 of Newberg et al. (2010). This shifts the β coordinates of the data so that the stream center better aligns with $\beta = 0$. The On field is located within $-2^o < \beta_{Orphan} < 2^o$ from the stream center, and the Off field is located within $2^o < |\beta_{Orphan}| < 4^o$ away from the stream on both sides. The On field, Off field and their difference is binned in Λ, as shown in Figure 7.3 and tabulated in Table 7.2. In our figure we use the same bin boundaries and widths as in Figure 5 of Newberg et al. (2010). The error in the counts in each bin is calculated as $\sqrt{N_i}$, where N_i is the number of star counts in a single bin of either the On or Off field histogram. The error for each bin in the difference histogram is calculated by summing the two errors in quadrature, as shown in Equation 7.4:

$$\sigma_{\text{diff},i} = \sqrt{N_{\text{On},i} + N_{\text{Off},i}}. \tag{7.4}$$

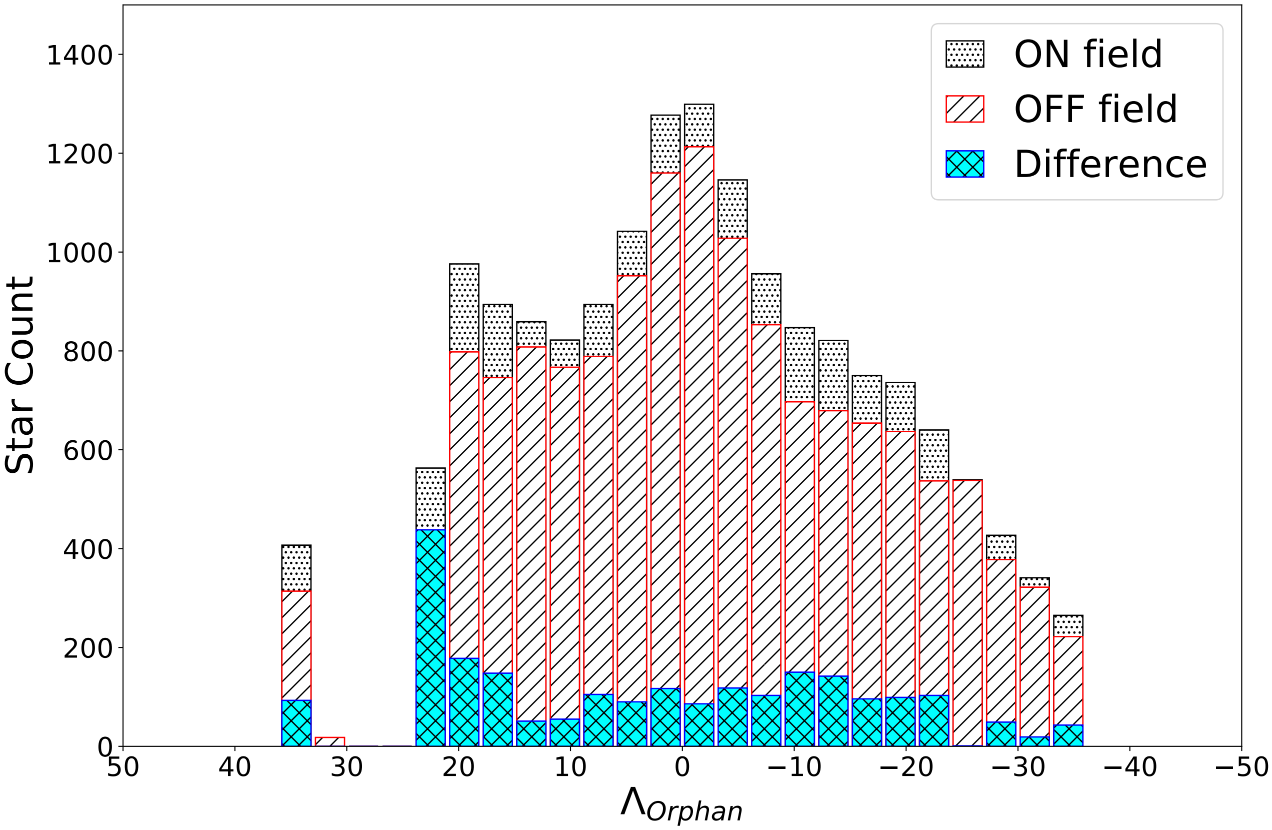

Figure 7.3: Histogram of the uncombined Λ bins. This figure reproduced Figure 5 of Newberg et al. (2010). The black histogram depicts the On field, the red the Off field, and the blue histogram the difference.

Table 7.2: On-field and Off-field star count data with their corresponding difference as seen in Figure 7.3. There are three bins for which we have no or incomplete data.

Bin Center (Λ)	On-field	Off-field	Difference	Error
-34.5	265	222	43	22.07
-31.5	341	322	19	25.75
-28.5	427	378	49	28.37
-25.5	539	538	1	32.82
-22.5	640	537	103	34.31
-19.5	736	637	99	37.05
-16.5	750	654	96	37.47
-13.5	821	679	142	38.73
-10.5	847	697	150	39.29
-7.5	956	853	103	42.53
-4.5	1146	1028	118	46.63
-1.5	1299	1213	86	50.12
1.5	1277	1160	117	49.37
4.5	1042	952	90	44.65
7.5	894	789	105	41.02
10.5	822	767	55	39.86
13.5	859	808	51	40.83
16.5	894	746	148	40.50
19.5	976	798	178	42.12
22.5	563	125	438	26.23
25.5	-	-	-	-
28.5	-	-	-	-
31.5	-	18	-	-
34.5	407	314	93	26.85

As can be seen in Table 7.2, there are three bins for which we have no data. This presents a problem with our current comparison algorithms. For the EMD calculation, we can easily skip these bins with a specific choice of flags. However, the cost component will need to be adjusted to remove the counts in the corresponding bins of the simulation histogram from the total histogram count. Since we have no data in that region of the sky, we cannot use simulation data in that region to constrain the mass in the histogram.

Next, the σ_β for each bin must be determined. Ideally, we would calculate a value for each bin in Table 7.2. However, this was not possible because there were too few stars and the errors in the star counts were too large to trace a cross section through each 3^o-wide stream segment. In many bins it was impossible to see any difference between the On and Off fields.

7.2.2 Modeling the Stream Width

Several Λ bins were combined in order to measure the excess stream stars in four separate segments along the stream, as shown as the histograms in Figure 7.4. We performed a numerical fit to the entire Λ bin using a model for the background and the Stream.

The model used a combination of a linear model for the background and a Gaussian model for the stream. It was fit to the data of the combined On and Off field counts as a function of β, for each Λ bin. The data are shown as open circles in the top panels and bar graphs in the lower panels of each plot in Figure 7.4. The model is given by:

$$f(x) = mx + b + Ae^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{7.5}$$

The parameters are the linear slope (m), the linear y-intercept (b), the Gaussian amplitude (A), the Gaussian offset (μ), and the Gaussian width (σ). The data is fit using a differential evolution algorithm to minimize the sum of square residuals between the model and data. The differential evolution algorithm used here is different from the one used in MilkyWay@home and was written specifically for this problem. This algorithm was tested on several mock datasets with random noise, and was able to recover the parameters that produced them.

The best fit curves to the data are shown as the black curves in the top panels of the plots in Figure 7.4. The best fit σ of the Gaussian component is taken as σ_β, and the error in the fit σ, as calculated in the next section, is taken as the error in σ_β.

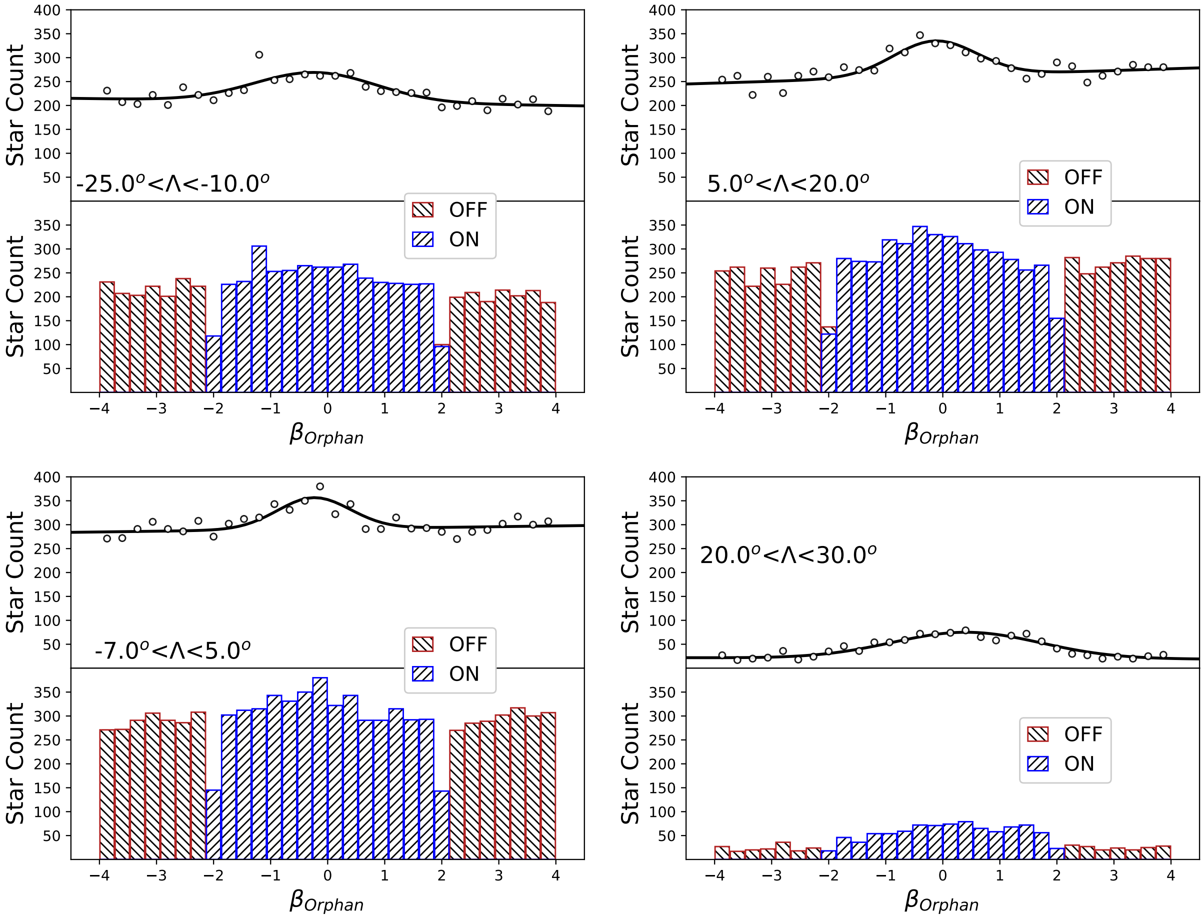

Figure 7.4: Plots of the combined Λ_{Orphan} bins. The On field (blue histograms) and Off field (red histogram) are shown separately. The combined counts of the two fields are shown in the top parts of each panel as black circles. The best fit to the combined distribution is given by a black curve. The fitted σ values and errors, which corresponds to the width of the On field, are given in Table 7.3.

Table 7.3: Fitted Gaussian widths and corresponding errors for four segments of the stream.

Bin Range	Fitted σ	Fit Error
-25 < Λ < -10	1.07	0.454
-7 < Λ < 5	0.64	0.33
5 < Λ < 20	0.74	0.32
20 < Λ < 30	1.3	0.40

7.2.3 Calculating the Error in σ_β

The error in the fit width of the Gaussian is determined using a Hessian matrix method. The Hessian matrix is a matrix of the partial derivatives of the cost function (sum of squared residuals, P) as a function of each parameter. It is therefore a 5×5 matrix:

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{bmatrix} \frac{\partial^2 f}{\partial m^2} & \frac{\partial^2 f}{\partial m \partial b} & \cdots & \frac{\partial^2 f}{\partial m \partial \sigma} \\ \frac{\partial^2 f}{\partial b \partial m} & \ddots & \cdots & \frac{\partial^2 f}{\partial b \partial \sigma} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \sigma \partial m} & \frac{\partial^2 f}{\partial \sigma \partial b} & \cdots & \frac{\partial^2 f}{\partial \sigma^2} \end{bmatrix}. \tag{7.6}$$

The partial derivatives are determined numerically. We keep the order of the numerical derivatives the same. Therefore, for the diagonal elements the derivatives are given by a numerical second derivative:

$$f''(x) \approx \frac{P(\alpha + h) - 2P(\alpha) + P(\alpha - h)}{h^2}, \tag{7.7}$$

and the off diagonal elements are given by a numerical partial derivative:

$$f''(x) \approx \frac{P(\alpha + h_1, \gamma + h_2) - P(\alpha + h_1, \gamma - h_2) - P(\alpha - h_1, \gamma + h_2) + f(\alpha - h_1, \gamma - h_2)}{4h_1 h_2}, \tag{7.8}$$

where α and γ are any of the parameters such that $\alpha \neq \gamma$, and h's are step sizes for a given parameter. The inverse of this matrix is the variance matrix, $\Sigma_{ij} = H_{ij}^{-1}$, the diagonal values of which are related to the errors in the fit of each parameter by:

$$\sigma_{fit,i} = \Sigma_{ii} = \sqrt{H_{ii}^{-1}}. \tag{7.9}$$

Since the step sizes for each parameter are generally different, we must choose ideal values. We first use some initial value for the step sizes for each parameter. We then use an iterative method of calculating the errors in each parameter and replacing the step size for a parameter with its error. This is repeated until the errors no longer change between iteration, i.e., they converge. This is also done with several different initial values, though the convergence error values are normally the same for any reasonable initial step size.

The $\sigma_{\beta,i} \pm \sigma_{fit,i}$ for each of the combined bins are assigned to the bin in Figure 7.3 whose bin center corresponds to the bin center of the combined bin. Because the background counts are much larger than the counts of turnoff stars in the stream, the errors in the σ_β's are fairly large. The error could be reduced, and the number of measured bins increased, if the stream could be better separated and the background reduced. Proper motion data from future Gaia data releases will be very helpful for this.

CHAPTER 8
Optimization and Results

8.1 MilkyWay@home

Our method of probing the dark matter content of dwarf galaxies requires significant computational resources from MilkyWay@home. MilkyWay@home is an approximately 800 TeraFLOPS volunteer supercomputing platform that is specially designed to optimize model parameters, given a dataset with which to compare the model, and a function that measures the goodness of fit. It uses the Berkeley Open Infrastructure for Network Computing (BOINC), which manages the sending of "work units" to volunteered computers in all countries in the world. Each work unit computes the goodness-of-fit for one set of model parameters. These work units run in the background when the volunteered computers are not being used for other tasks; when they are finished, the results are sent back to our local server.

This platform was originally designed to use statistical photometric parallax to fit a density model to turnoff stars in the Milky Way's stellar halo (Cole et al., 2008; Newby et al., 2013; Newberg, 2013). For this case we use a maximum likelihood algorithm to define the best fit, 20 parameter density model to the observed spatial distribution of stars. We have adapted MilkyWay@home to find the best model fit to observed properties of tidal debris, including stream density and dispersion.

Optimization methods come from the Toolkit for Asynchronous Optimization (TAO) (Desell, 2009), which was originally designed specifically for MilkyWay@home. The optimization methods include particle swarm, differential evolution, and genetic algorithms that work well in our highly asynchronous, heterogeneous environment (Desell et al., 2008, 2009).

For the N-body application, MilkyWay@home optimizes over five parameters. It uses a differential evolution algorithm specially modified to work on a distributed supercomputer (Desell, 2009; Weiss, 2018). This optimizer creates and maintains a population of the best simulation likelihoods. It replaces members of the population with parameter sets with likelihoods that are better, as these better population members are returned from volunteer

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, "Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing," ApJ.

machines. At any given time, the current population is used to determine what values to send out to volunteers to evaluate. A complete description of the method and algorithm can be found in Chapters 2 and 5 of Desell (2009), and Section 4 of Weiss et al. (2018). Chapter 2 of Desell (2009) describes the basic algorithm in its normal form. Chapter 5 of Desell (2009) and Section 4 of Weiss et al. (2018) describe the implementation of this algorithm in our highly asynchronous environment.

We use 20,000 bodies in our simulations, of which 10,000 represent the stars and 10,000 represent the dark matter. Simulation times vary from a few minutes to a few days depending on the model parameters being simulated and the speed of the volunteer machine. Optimization on Milkway@home tends to take two to three weeks until convergence. Full convergence is reached when returned values no longer differ, and the entire population is identical.

The differential evolution algorithm has several parameters which affect how quickly our search converges. We have tried different optimization parameters and were able to speed up the optimization from what was previously about three months for convergence. This involved changing the population size, cross over rate and the differential scaling factor. The algorithm uses a population of parents made up of a parameter set and an associated likelihood value. For N-body, we use a population size of 50. Children are generated from two randomly selected parents. The crossover probability is set to 0.9 and the differential scaling factor is set to 0.8. For a detailed explanation of the roles of these parameters in the algorithm see Section 4 of Weiss et al. (2018). The current values for these parameters are given in Table 8.1.

Table 8.1: Current values for the differential evolution optimization algorithm. Using these parameters has led to a N-body optimization convergence time of approximately 3 weeks.

Current Differential Evolution Parameters	
Parameter	Current Value
Population Size	50
crossover rate	0.9
Differential Scaling Factor	0.8

8.2 Simulated Data

To test our ability to recover parameters for our model, we created a simulated data set with known parameters, and attempted to recover the parameters used in its creation using MilkyWay@home. Table 8.2 shows the parameters used to generate the simulated ("correct") data histogram, as well as the range ("search range") over which the optimization algorithm was allowed to search. The search range for the mass ratio parameter (ξ_M) was set as wide as possible, with a mass to light ratio ranging between 1000:1 to 1:1, representing dwarf galaxies that are strongly dark matter dominated to dwarf galaxies with no dark matter at all. The baryonic mass (M_B) search range was also set fairly wide compared with the dwarf galaxy mass used in Newberg et al. (2010) of 2.5×10^6 $M_\odot$. We again use the static Milky Way potential and orbit from Newberg et al. (2010), as described in Section 2.1 and 2.2, respectively.

The baryonic mass search range of 1 to 100 structural units corresponds to a mass of 2.2×10^5 $M_\odot$ to 2.2×10^7 $M_\odot$, respectively. The search range for the baryonic scale radius (R_B) was narrower, between 0.05 kpc to 0.5 kpc. The smaller range was chosen because any lower would mean a very dense object, requiring a very small timestep and thus an unfeasibly long computational time on the average client computer. The upper end of the range limits the scale radius to a relatively large Milky Way dwarf galaxy, though the LMC, SMC, Sgr, Crater II and Canis Major are all larger. The radius ratio (ξ_R) search range goes from 0.1 to 0.6, corresponding to a dark matter halo that is 100 times more extensive as the baryonic component to mass follows light.

The simulation time was also given a fairly wide search range corresponding to the dwarf galaxy core remaining undisrupted to it being mostly disrupted as seen in Figure 8.1. A (Λ, β) plot of the simulated dwarf galaxy used in our final optimizations, with histograms of both components is shown in Figure 8.2. As seen in the figure, the dark matter component follows the evolution of the baryonic component, with alignment of the peaks in their distributions. We can extract information about the unseen dark matter by exploring how the tidal stream is formed in its presence.

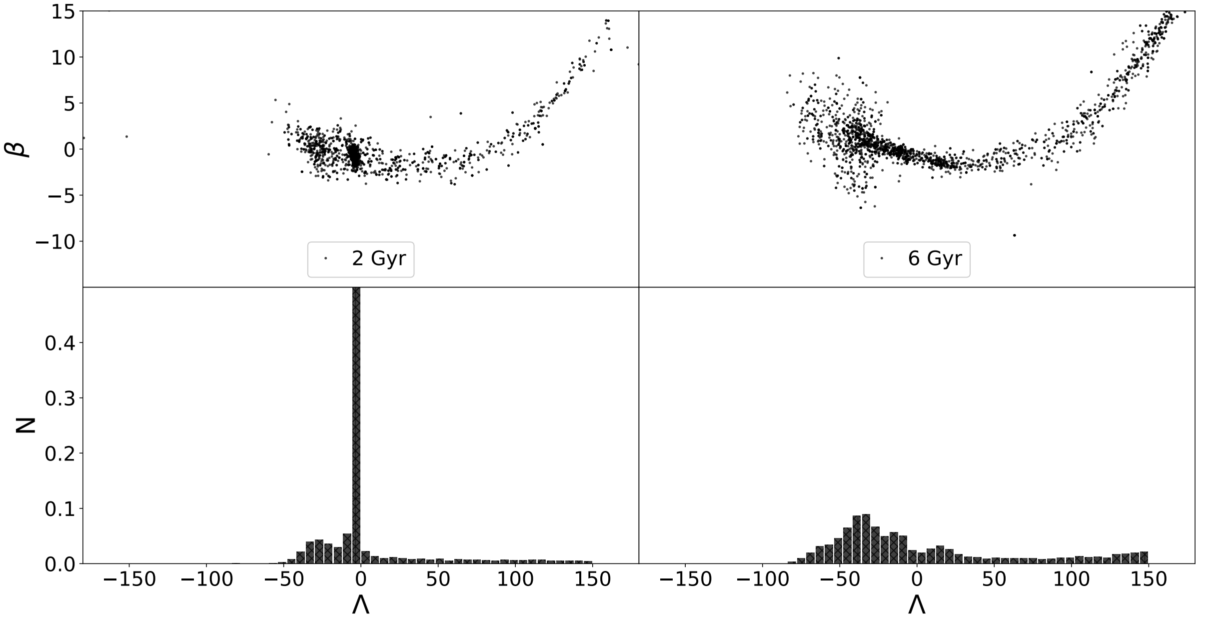

Figure 8.1: The stellar portion of a tidally disrupted dwarf galaxy in stream coordinates (Λ, β). The N-body simulation was evolved for 2 Gyr (left) and 6 Gyr (right), with their respective histograms of stellar density as a function of angle along the stream shown in the lower panels.

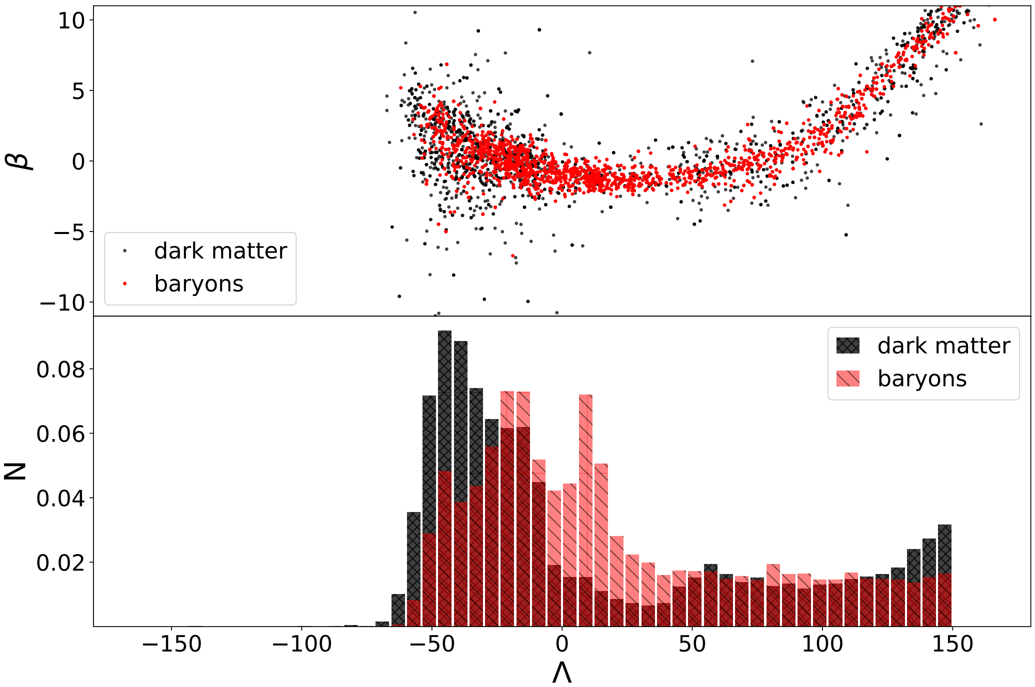

Figure 8.2: **The simulated data histogram. Above is a sub-sampled Λ, β plot of the two components. Below are the corresponding density distribution histograms. As can be seen in either panel, the two components have similar density distributions. The main difference is the baryonic core present at $\Lambda \sim 10$. Both panels illustrate that the dark matter component closely mirrors the evolution of the baryonic component.**

Evidence that the properties of the dark matter in the progenitor can be ascertained from the distribution of stars in the stellar stream it produces is shown in Figure 8.3. This figure shows a two-dimensional likelihood surface, generated by varying the two ratio parameters that determine the dark matter properties, while fixing the other parameters. The simulated "data" histogram is generated with a different random seed than was used in generating the streams that were generated to calculate the likelihoods. A ridge is clearly visible in the left panel of Figure 8.3, where the recorded likelihood is much better than the surrounding parameter space. On the right is the same plot but with the correct answer indicated by a green 'x', and the fitted values from our optimizations shown as black dots. In addition, the right panel includes an overlay that indicates the region of parameter space in which the dark matter mass within the half-light radius is approximately constant. Note that this region aligns with the ridge in the likelihood surface, but that the likelihood surface is not constant along the ridge. The likelihood is higher near the mass and radial profile for which the simulated data was created. If the algorithm was sensitive exclusively to the mass within the half light radius, the fitted values would have been randomly distributed along the shaded region. The fitted values fall within the ridge, indicating an implied relationship between the values that the algorithm is fitting and the mass outside of the half-light radius.

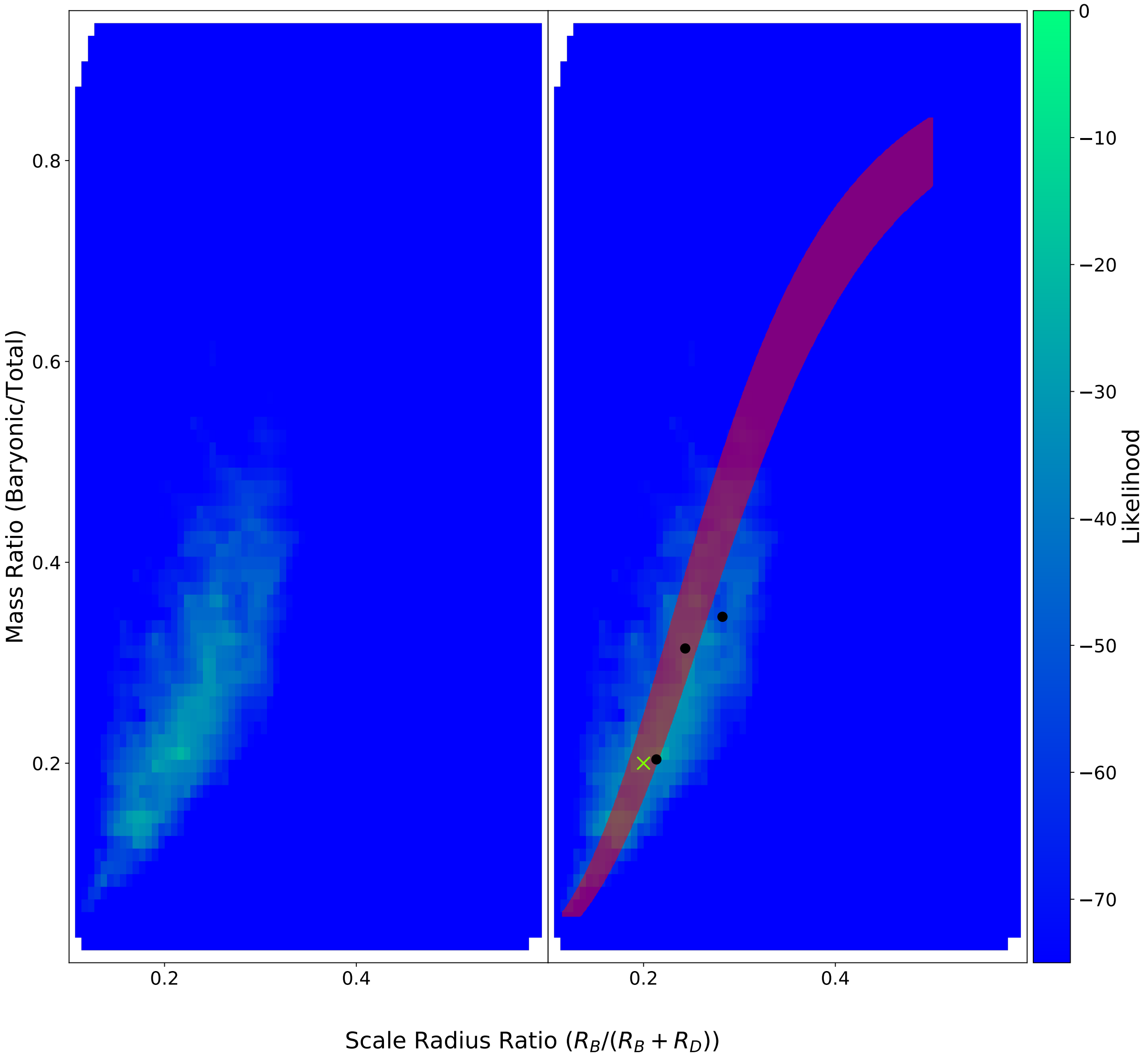

Figure 8.3: Likelihood surface of the two ratio parameters, the radius ratio and mass ratio, that are used to determine the dark matter scale radius and mass. The right plot is the same as the left but also shows the fitted values as black dots, and the correct answer as a green x. Also plotted is the region, displayed in red, in these two parameters where the dark matter mass within the half light radius is approximately the same as the simulated progenitor. As the heat map shows, there is a ridge in the surface that corresponds with the red region. However, the ridge is peaked at the dark matter mass and scale length of the simulated progenitor, indicating that information about these two quantities is present in the resulting stream.

We use a different seed in the creation of our simulated data set from that used for the optimization runs. For the baryonic component parameters, we use similar parameters as those of the single component Plummer model used in the N-body simulations in Newberg et al. (2010), which is approximately 12 simulation units, or 2.7 $\times 10^6$ $M_\odot$, with a scale length of 0.2 kpc. For the dark matter component, we use a mass to light ratio of 4:1, with the dark matter radial extent four times as large as the baryonic component. This represents a dwarf galaxy whose stellar component is embedded in an extended dark matter halo. This can be seen in Figure 8.4, which shows the radial distribution of both components and the entire dwarf galaxy as a whole. The mass ratio was also set to 0.2, which makes the dark matter mass four times as massive as the baryonic component, at 48 simulation units, or 1.1×10^7 $M_\odot$. These parameters were found by fixing the baryonic component and altering the dark matter component until the final distribution, shown in Figure 8.2, is reminiscent of the Orphan stream stellar distribution given in Figure 5 of Newberg et al. (2010). The likelihood value obtained from comparing a histogram made from these parameters with a simulation also run with these parameters but with a different random number seed used in the optimization (different from the one used in the creation of the histogram) is given along with the "correct" parameters in Table 8.2.

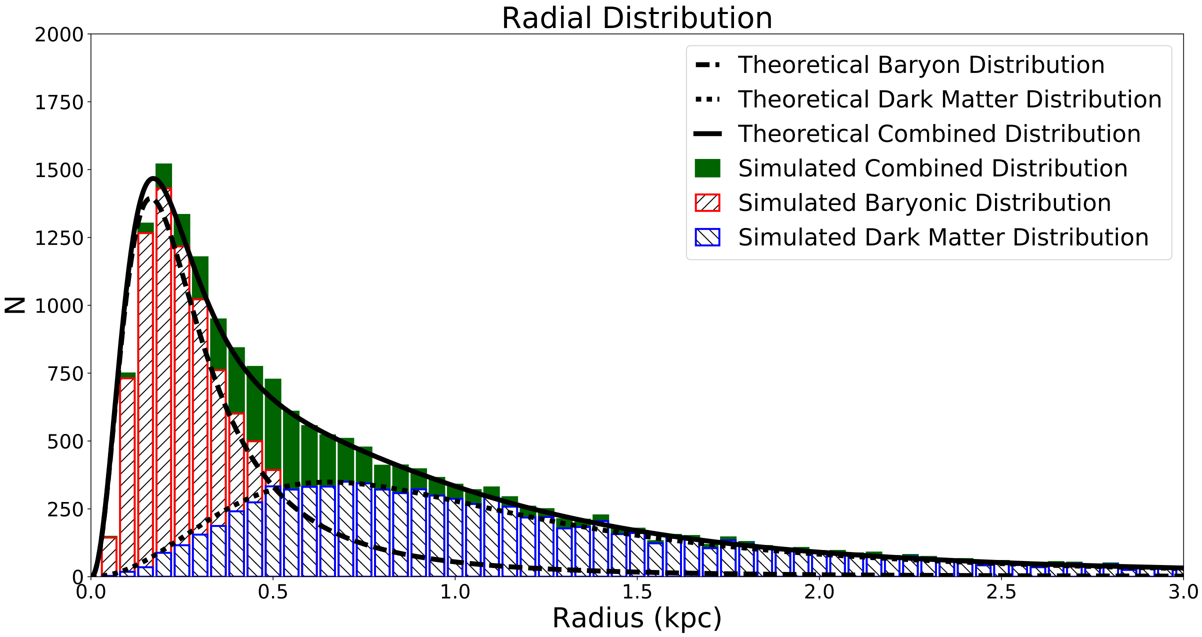

Figure 8.4: An example of the initial distribution of the two component dwarf galaxy with the parameters set to the values as described above. As the figure shows, the dark matter component extends well past the baryonic component.

8.3 Results

To create the histogram used in our optimizations, the dwarf galaxy is integrated from 4 Gyr in the past forward for 3.95 Gyr. This places the progenitor core at (l,b) $\sim$ (250^o, 50^o) as in Figure 17 of Newberg et al. (2010). As we have described previously, in the simulations performed on MilkyWay@home, the dwarf galaxy is placed in orbit at some time in the past and integrated forward for the same amount of time, returning the best likelihood value between a search range of 98% and 100% of the evolution time. For this histogram, the forward evolution time is 98.75% of the time in the past from which it was evolved, placing the progenitor core within the search range for the best likelihood calculation. In other words, doing this places the progenitor core just before where it would end up if the forward evolution time was also set to 4 Gyr. This is done to give the optimizer more freedom in the placement of the progenitor core during the optimization as we have described in Section 6.5.1. Since the core is not placed at 100% of the evolution time, fluctuations on both sides of the progenitor core can be taken into account. Examples of these fluctuations are shown in Figures 8.5 and 8.6. These plots show histograms of tidal streams created from a dwarf galaxy with the same parameters used in our optimizations, but each formed with a different random number seed. As seen in those figures, the peak fluctuates on either side of $\Lambda_{Orphan} \sim 25$ and $l \sim 250$, respectively.

Three identical but independent optimizations were performed for our simulated tidal stream. The results of the three optimization trials are shown in Tables 8.2. We have converted the parameters to values that are easier to understand, and are given in Table 8.3. We were able to recover the baryonic mass very well, with all the fitted values within 2.3% from the correct answer. Similarly, we are able to recover the baryonic scale radius, with all the fitted values within 4% from the correct answer. The evolution time was not as accurately recovered, with the fitted results all within 8.5% of the correct answer. We were not able to recover the dark matter values accurately, with the fitted mass ratios as far as 73% from the correct answer, and the radius ratios as far as 41% from the correct answer.

Table 8.2: Best fit values for optimizations fit to a histogram made from a different seed from that used in the simulations run on MilkyWay@home. These were fit using the comparison method described in Chapter 6. The simulated data histogram was created by integrating a dwarf galaxy from 4 Gyr in the past for 3.95 Gyr. The parameters are the evolution time, the scale length of the baryonic component (R_B), the radius ratio (ξ_R), the baryonic mass (M_B), and the mass ratio (ξ_M).

Parameters	Evolve Time (Gyr)	R_B (kpc)	ξ_R	M_B (sim units)	ξ_M	Likelihood
Correct	4.0	0.2	0.2	12.0	0.2	-41.127
Search Range	[2.0 - 6.0]	[0.05 - 0.5]	[0.1 - 0.6]	[1.0 - 100.0]	[0.001 - 0.95]	
Trial 1	4.045 ± 0.031	0.208 ± 0.003	0.282 ± 0.025	11.719 ± 0.025	0.346 ± 0.068	-17.226
Trial 2	4.340 ± 0.030	0.205 ± 0.005	0.213 ± 0.010	12.061± 0.036	0.204 ± 0.013	-18.582
Trial 3	4.134 ± 0.026	0.208 ± 0.005	0.243 ± 0.024	11.871 ± 0.042	0.314 ± 0.045	-18.954

Table 8.3: The fitted values as in Table 8.2, but converted to values that are easier to understand. In addition to those parameters from Table 8.2, we have the scale length of the dark matter component (R_D), and the mass of the dark matter component (M_D).

Parameters	Evolve Time (Gyr)	R_B (kpc)	R_D (kpc)	M_B ($M_\odot$)	M_D ($M_\odot$)
Correct	4.0	0.2	0.8	2.667×10^6	1.067×10^7
Trial 1	4.045 ± 0.031	0.208 ± 0.003	0.530 ± 0.071	$2.605 \pm .001 \times10^6$	$.493 \pm .166\times10^7$
Trial 2	4.340 ± 0.030	0.205 ± 0.005	0.755 ± 0.052	$2.681 \pm .001\times10^6$	$1.048 \pm .086\times10^7$
Trial 3	4.134± 0.026	0.208 ± 0.005	0.646 ± 0.091	$2.639 \pm .001\times10^6$	$.576 \pm 0.132\times10^7$

However, the mass ratio had a wide search range of $\xi_M = [0.001, 0.95]$. The dark matter mass represented by this ratio is dependent on the baryonic mass, as given by Equation 2.1, which has a search range of $M_B = [1, 100]$ simulation units, or [2.2×10^5 $M_\odot$, 2.2×10^7 $M_\odot$]. Looking at all possible configurations of baryon mass and dark matter mass accessible during the optimization, the mass of the dark matter can be set anywhere between 10^4 $M_\odot$ and 2.2×10^{10} $M_\odot$. Looking at the narrowest search range possible, that which would occur if the mass of the baryons was fixed at the correct value, the accessible values of dark matter masses would still range from 1.4×10^5 $M_\odot$ to 2.67×10^9 $M_\odot$. This is an incredibly large range of masses. Yet, in each trial, we were able to fit the dark matter mass within a factor of approximately 2. Therefore, although the mass ratio was not fit very accurately, considering the size of the parameter space, the fit to the total dark matter mass is remarkable.

We can perform a similar analysis for the radius ratio, which ranges from $\xi_R = [0.1, 0.6]$. The dark matter scale radius represented by this ratio is dependent on the baryonic scale radius, which has a range of $R_B = [0.05, 0.5]$ kpc. Again looking at the possible configurations of baryonic scale radius and dark matter scale radius accessible during the optimization, the scale length of the dark matter can be set anywhere between 0.033 kpc to 4.5 kpc. Looking at the narrowest range possible, if the baryonic scale radius was fixed to the correct answer, the accessible values of dark matter scale lengths would range from 0.33 kpc to 1.8 kpc. This, too, is a wide range of possible scale lengths. Yet, we were able to fit the dark matter scale radius within a factor 1.5.

Furthermore, the fact that we are able to recover these values to this degree is incredible as the radial extent of the dark matter component extends well beyond the half light radius (see Figure 8.4). In fact, with these parameters, most of the dark matter mass is located outside of the half light radius of 0.261 kpc, with 1.43 simulation mass units (3.2×10^5 $M_\odot$), of dark matter mass enclosed within. This means our algorithm was able to find the total dark matter mass within a factor $\sim$2 with less than 3% of the dark matter mass within the half light radius.

We performed a preliminary error analysis using a Hessian matrix method, as described in Section 7.2.3. The error values are small; none of the correct values lay within the errors of the fit values. In fact, even the fitted parameters that were quite close to the correct values are many σ's away. A table of the number of σ each fitted value is away from the correct value is given in Table 8.4.

Table 8.4: The number of σ, each fitted value was away from the correct answer according to the calculated errors.

Parameters	Evolve Time (Gyr)	N_σ
Correct	4.0	
Trial 1	4.045 ± 0.031	1.45
Trial 2	4.340 ± 0.030	11.33
Trial 3	4.134 ± 0.026	5.15

Parameters	R_B (kpc)	N_σ
Correct	0.2	
Trial 1	0.208 ± 0.003	2.67
Trial 2	0.205 ± 0.005	1.0
Trial 3	0.208 ± 0.005	1.60

Parameters	ξ_R	N_σ
Correct	0.2	
Trial 1	0.282 ± 0.025	3.28
Trial 2	0.213 ± 0.010	1.30
Trial 3	0.243 ± 0.024	1.79

Parameters	M_B (sim units)	N_σ
Correct	12.0	
Trial 1	11.719 ± 0.025	11.24
Trial 2	12.061± 0.036	1.69
Trial 3	11.871 ± 0.042	3.07

Parameters	ξ_M	N_σ
Correct	0.2	
Trial 1	0.346 ± 0.068	2.15
Trial 2	0.204 ± 0.013	.31
Trial 3	0.314 ± 0.045	2.53

Although it is possible that the optimizer could be tuned to better traverse the likelihood surface, we suspect that the poorness of the fit is most likely due to our choice of random seed used to create the simulated data set. For some choices of random number seed, some of the distinctive characteristics of the provided histogram, such as peaks, can disappear. The simulated data histogram used in the above optimizations, seen in the first panel in each figure, was one such case. Our particular choice of random seed produces a tidal stream that is unfortunately not very similar to the real data in that it does not have a strong peak (see Figures 8.5 and 8.6). The best fit parameters from MilkyWay@home, using a different seed, therefore do not reproduce the data. Although other random seeds would have produced a closer match to the fit parameters, it may not be possible in practice to fit much closer than we show here due to chaotic differences between the simulation results. This process deserves a more thorough analysis in the future.

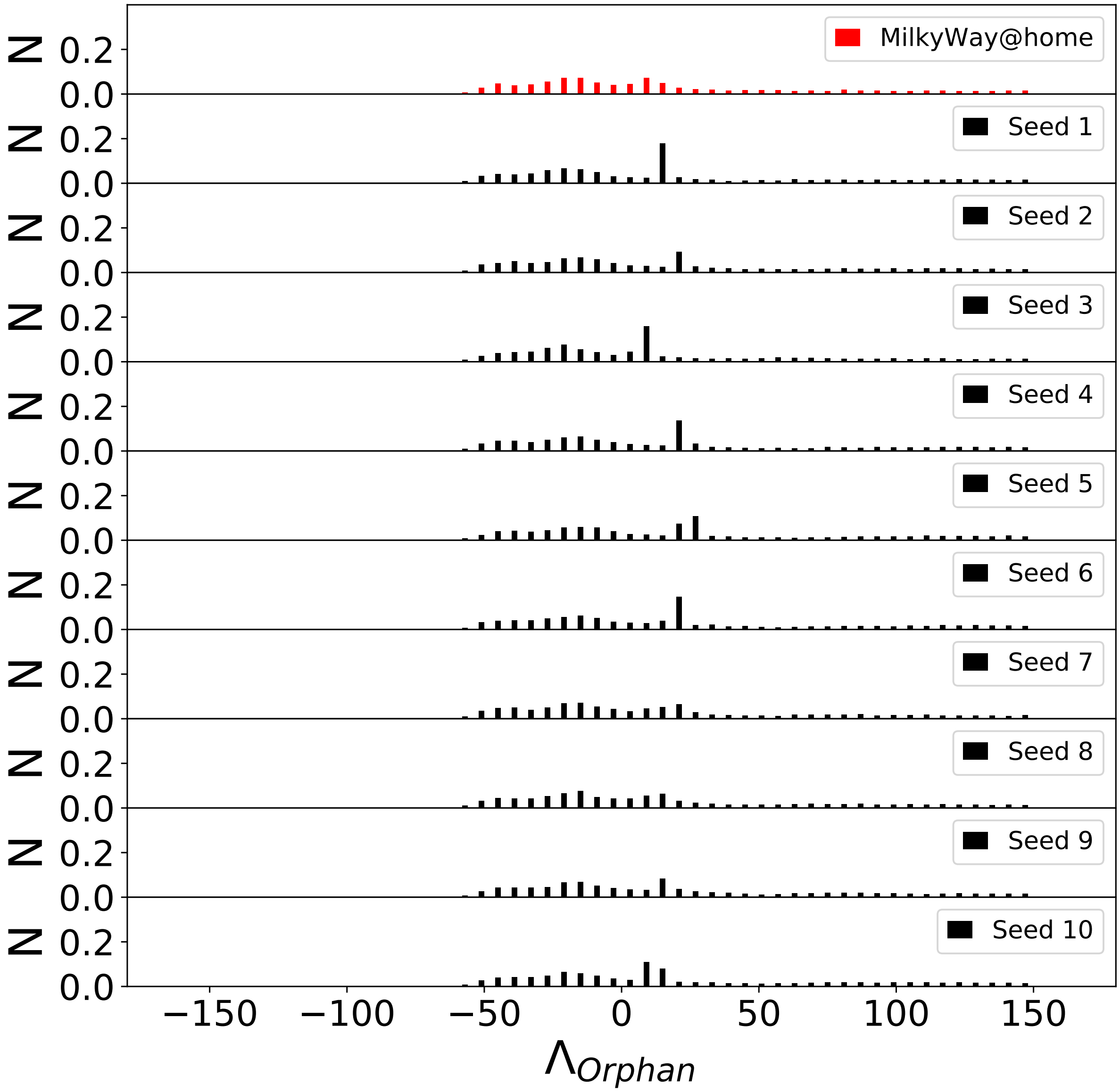

Figure 8.5: Plots of the simulation histogram used in the optimization, made with 10 different seeds. As the plot shows, there is fluctuations in the location and shape of the peak seen at $\Lambda_{Orphan} \sim 25^o$.

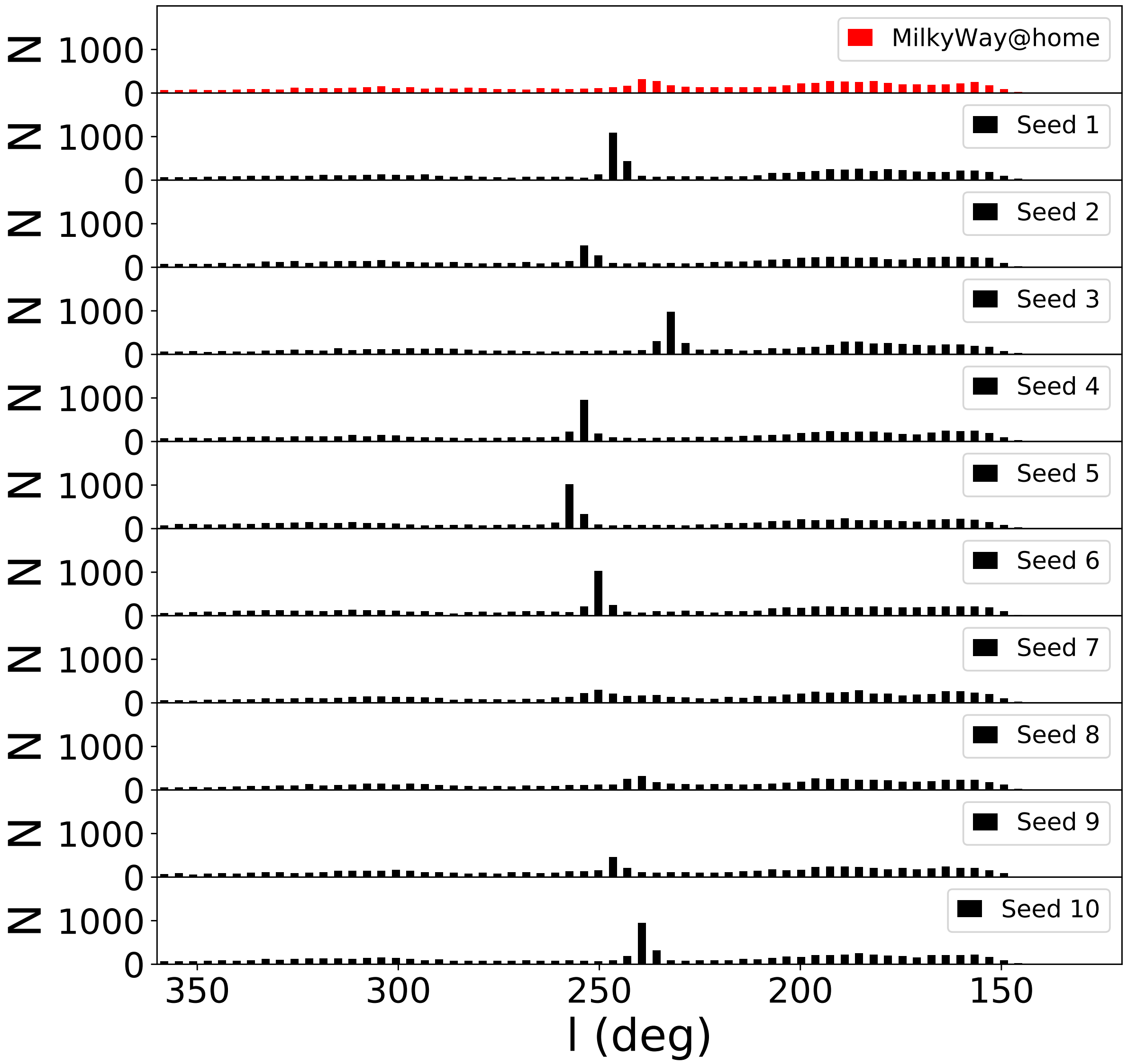

Figure 8.6: **The same dwarf galaxies shown in Figure 8.5, but binned in l. As the plot shows, there is a fluctuation in the location and size of the peak at $l \sim 250^o$.**

These results show a proof of concept: our distributed supercomputer, MilkyWay@home, using the described optimization algorithm, can recover the best fit parameters of a progenitor dwarf model, given sufficient photometric data for tracer stars in the Orphan Stream, and that the model for the dwarf galaxy, the Milky Way potential, and the orbital parameters are known exactly. Tests will be required to determine how sensitive we are to errors in these quantities. In principle, the algorithm can be extended to fit these as well, with additional data required to constrain the additional parameters. Importantly, using our N-body method, we are able to extract information about the dark matter distribution well outside of the half light radius. This can be compared to other methods of fitting tidal stream properties, such as streakline (Küpper et al., 2012), which are limited to information contained within the tidal radius. These results give us confidence that we can fit accurate dwarf galaxy parameters to a histogram of stellar stream data: the parameters that MilkyWay@home fit are reasonable estimates of the parameters of the dwarf galaxy progenitor for the Orphan Stream.

CHAPTER 9
Conclusions

We have developed a method for constraining the shape and mass of the progenitor dwarf galaxy from the distribution of stars in the tidal stream that it created. Our model provides a straightforward way to probe the dark matter content of the Milky Way satellites without the assumption of virial equilibrium required to tie line-of-sight velocities to dwarf galaxy mass. It also avoids the pitfalls of using a small number of spectra to determine the line-of-sight velocity dispersion.

We have also derived a metric of comparison between our simulated tidal stream distribution and that compiled from stellar data. This metric has proven to work well as seen when comparing our simulations to simulated data sets. The comparison metric is comprised of three components, each constraining a different aspect of the tidal stream: the general shape, the mass present in each stream, and the width of the stream as a function of stream length. To utilize these comparison metrics we create a normalized histogram of the simulated tidal stream along the stream length. We derive expressions for the stream's angular dispersion along with a theoretical error.

Using stellar positional data, we are able to create a histogram compatible with our algorithm that allows us to compare our simulations directly to stellar data. We estimate the amount of mass each tracer star represents by comparing the Orphan Stream to a globular cluster comprised of a similar population of stars. We use stellar data from Newberg et al. (2010) to create a density distribution histogram along the stream. We determine the dispersions along the stream by fitting a model to the data in each histogram bin, using a similar optimization algorithm as that used in MilkyWay@home. We use a Hessian matrix technique to determine the errors in our fitted values, one of which is the error in the dispersion. In order to estimate the parameters of the real Orphan Stream progenitor, we need to run the algorithm in a way that recognizes the missing data in some Λ bins in the real data. Here, we only show that it is is possible to create the required inputs for MilkyWay@home from real stellar data. After implementing these changes, our next step is

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, "Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing," ApJ.

to perform an optimization over the Orphan Stream data histogram. We may need to use several seeds in the optimization in order to find typical fitted values.

We demonstrate the ability to recover the parameters used in creating a simulated tidal stream both when using the same and a different seed as the simulations. We are able to recover the parameters for the baryonic component very well, with the fitted masses all within 2.3% of the correct value and the baryonic scale radius all fit within 4% of the correct value. We are able to recover the mass of the dark matter component within a factor ~2 of the correct value, and the dark matter scale length within a factor of 1.5 of the correct value, even though, for our simulated tidal stream progenitor, 97% of the dark matter mass lies outside the half-light radius. This gives us confidence that we are able to extract information about the dark matter component of the progenitor of the Orphan Stream.

9.1 Discussion and Future Work

We have been able to recover parameters used in the creation of a simulated tidal stream. This demonstrates our algorithm's ability to traverse the complicated likelihood surface of our five parameters. Furthermore, we are able to recover these parameters when we have used a different seed for the creation of our simulated data set. This further illustrates the versatility of our optimization algorithm in finding the best fit values and indicates our ability to fit actual stellar data.

The optimizations performed so far have used a number of idealizations. First, the dwarf galaxy is described by an analytical model whose parameters are altered. The progenitor of the Orphan Stream might not be well represented by a Plummer model, and the stellar distribution and/or dark matter distribution might not be spherically symmetric. Secondly, we assume the Milky Way galaxy potential is known exactly. We use the axisymmetric time-independent model that led to the best orbit fit to the Orphan Stream. There may be other models which lead to a better fit which were not tested by Newberg et al. (2010), or are a more realistic representation of the Milky Way. Finally, we place the dwarf galaxy on a fixed, predefined orbit, which is also uncertain.

Fitting the Milky Way galaxy model parameters and the orbital parameters at the same time as the dwarf galaxy parameters would be advantageous. The algorithm can be expanded to accommodate the new parameters straightforwardly. New metrics of comparison will be required to constrain the goodness of fit with the additional parameters. The histograms will also need to be expanded to accommodate the new information. In addition to fitting the Milky Way model parameters, different potential models can be implemented to study the effects on the fit. We would also like to implement more models for the dwarf galaxy such as the generalized Hernquist model (Zhao, H., 1995), the Navarro-Frank-White model (NFW, Navarro et al. (1997)), and the Einasto profile (Retana-Montenegro et al., 2012).

A full error analysis must also be performed for the fitted values to the simulated and stellar data. We have described the use of a Hessian matrix in our determination of the error in the fitted values to our model for the stellar data. We have used this method to determine the error in the fitted dwarf galaxy parameters. We also need to evaluate the influence of systematic error on the accuracy of our fitted values. In determining the systematic error to our fits, we will also need to explore the effect of each fixed parameter used throughout the algorithm. These include the range of forward times scanned to find the best likelihood, the number of iterations used in rejection sampling, the effect of inexact orbital parameters, the Milky Way potential, the timestep size, and the softening parameter, among others. We will need to alter each parameter to determine how they affect the likelihood values.

We place the dwarf galaxy in orbit around a static representation of the Milky Way galaxy. Such a placement assumes the evolution of the Milky Way galaxy and the Universe has already taken place. In reality, dwarf galaxies form and disrupt in a universe that is constantly evolving. However, the Milky Way is thought to have fully formed over 10 Gyr ago (Krauss & Chaboyer, 2003; Cowan et al., 2002; del Peloso et al., 2005), long before the disruption of the Orphan Stream progenitor began. We do make the assumption that the Milky Way potential has not significantly changed since its formation, other than acquiring satellites. The largest structures and subhalos, such as the LMC, may have an effect on the orbit of the dwarf galaxy and even on the motion of the Milky Way. The effects of subhalos on the dwarf galaxy disruption is currently being explored using N-body realizations of several subhalo models. We are also investigating the impact of having the Milky Way move in the presence of large structures, such as the LMC, while the dwarf galaxy is disrupting. We would eventually like to simultaneously fit multiple dwarf galaxies to several tidal streams. This would be especially useful in constraining the form of the Milky Way Galaxy model, and its associated parameters.

We have also considered the possibility that the sudden placement of the dwarf galaxy in the presence of the Milky Way potential can cause tidal shocks in the dwarf galaxy. Such shocks could lead to evaporation of the dwarf galaxy, as particles fly off at high speeds. To study this, the Milky Way potential was ramped with the dwarf galaxy's center of mass and momentum held fixed. This was meant to allow the dwarf galaxy to relax in place, acclimating to the comparably large Milky Way potential. This method worked best when the ramping period was small compared to the simulation time; long ramping times often resulted in tidal stripping of the dwarf galaxy. Surprisingly, ramping did not significantly change the N-body simulations so it was not included in our algorithm.

We have also extensively studied the effects of dynamical friction on the dwarf galaxy disruption. It was discovered that because of the predicted low mass of the Orphan Stream progenitor, 10^7 $M_\odot$, dynamical friction does not have a notable impact on the disruption and orbit of the dwarf galaxy. We found that dynamical friction becomes more significant when the mass of the dwarf galaxy exceeds 10^9 $M_\odot$, at which point the dwarf galaxy experiences drag, resulting in orbital decay.

9.1.1 Future Algorithmic Endeavors

We plan to extend the MilkyWay@home algorithm so that the orbit of the Orphan Stream and the properties of the Milky Way potential can be fit simultaneously with the properties of the progenitor dwarf galaxy. In order to accomplish this, we are in the process of refactorizing the algorithm which generates the simulated histograms, and the layout of the histograms themselves. The data structure currently makes a single histogram, and for each bin derives all the information needed for the likelihood calculation. Instead, we plan to make several instances of a histogram, each with their designated values that will be used for the corresponding likelihood function. These changes will not only allow us to expand the number of fit parameters, but will also allow us to use different binning arrangements for the density distribution information, the dispersion information, and any other new values added to the histogram. This may become important as the project is expanded to fit more parameters involved with the orbit, the Milky Way potential, and even simultaneously fitting other tidal streams.

We are currently building and expanding the test suite of the N-body algorithm. This is especially important for the continuity of the N-body project. As more students are beginning to develop production algorithms, there needs to be a systematic way of making sure each component of N-body behaves in the expected manner.

A large portion of MilkyWay@home volunteers use Graphic Processing Units (GPUs) to crunch workunits. Currently, only the Separation application has a working GPU version. We would like to finish work on a N-body GPU application. This would vastly increase the number of users that can work on N-body workunits, and thus the computational power devoted to the N-body application. This will become more and more necessary as we expand the number of parameters that are fit. More parameters means more results from volunteers are needed for the optimization.

BIBLIOGRAPHY

Aarseth, S. J. 2003, Gravitational N-Body Simulations: Tools and Algorithms, (1st ed.; Cambridge, UK: Cambridge University Press)

Aarseth, S. J., Henon, M., & Wielen, R. 1974, A&A, 37, 183

Aarseth, S. J., & Hoyle, F. 1963, MNRAS, 126, 223

Aartsen, M. G., Abraham, K., Ackermann, M., et al. 2015, EPJC, 75, 492

Abazajian, K. N., Canac, N., Horiuchi, S., & Kaplinghat, M. 2014, PhRvD, 90, 023526

Abolfathi, B., Aguado, D. S., Aguilar, G., et al. 2018, ApJS, 235, 42

Akerib, D. S., Alsum, S., Araújo, H. M., et al. 2017, PhRvL, 118, 021303

Aprile, E., Aalbers, J., Agostini, F., et al. 2016, JCAP, 4, 027

Arca-Sedda, M., Kocsis, B., & Brandt, T. D. 2018, MNRAS, 479, 900

Barnes, J., & Hut, P. 1986, Natur, 324, 446

Barnes, J. E. 2001, Treecode Guide

Bartels, R., Krishnamurthy, S., & Weniger, C. 2016, PhRvL, 116, 051102

Belokurov, V., Zucker, D. B., Evans, N. W., et al. 2006, ApJ, 642, L137

Belokurov, V., Zucker, D. B., Evans, N. W., et al. 2007, ApJ, 654, 897

Belokurov, V., Evans, N. W., Irwin, M. J., et al. 2007, ApJ, 658, 337

Binney, J., & Tremaine, S. 2006, Galactic Dynamics, (2nd ed.; Princeton, NJ: Princeton University Press)

Blaña, M., Fellhauer, M., Smith, R., et al. 2015, MNRAS, 446, 144

Bradač, M., Allen, S. W., Treu, T., et al. 2008, ApJ, 687, 959

Bull, P., Akrami, Y., Adamek, J., et al. 2016, PDU, 12, 56

Carlberg, R. G. 2012, ApJ, 748, 20

Carlberg, R. G., Grillmair, C. J., & Hetherington, N. 2012, ApJ, 760, 75

Chakrabarti, S., & Blitz, L. 2009, MNRAS, 399, L118

Cholis, I., Evoli, C., Calore, F., et al. 2015, JCAP, 12, 005

Cole, N., Newberg, H., Magdon-Ismail, M., et al. 2008, ApJ, 683

Cole, S., & Lacey, C. 1996, MNRAS, 281, 716

Cowan, J. J., Sneden, C., Burles, S., et al. 2002, ApJ, 572, 861

de la Vega, A., Quillen, A. C., Carlin, J. L., Chakrabarti, S., & D'Onghia, E. 2015, MNRAS, 454, 933

Debattista, V. P., Roškar, R., Valluri, M., et al. 2013, MNRAS, 434, 2971

Deg, N., & Widrow, L. 2013, MNRAS, 428, 912

del Peloso, E. F., da Silva, L., Porto de Mello, G. F., & Arany-Prado, L. I. 2005, A&A, 440, 1153

Desell, T. 2009, PhD thesis, Rensselaer Polytechnic Institute, Troy, New York

Desell, T. 2011, in International Conference on Computational Intelligence and Software Engineering (CiSE) (Wuhan, China: SCIRP)

Desell, T., Magdon-Ismail, M., Szymanski, B., et al. 2009, in In the 5th IEEE International Conference on e-Science (San Diego, USA: IEEE)

Desell, T., Szymanski, B., & Varela, C. A. 2008, in Proceedings of the 17th International Heterogeneity in Computing Workshop (Miami, Fl: IEEE)

Domínguez, R., Fellhauer, M., Blaña, M., et al. 2016, MNRAS, 461, 3630

Drlica-Wagner, A., Albert, A., Bechtol, K., et al. 2015, ApJ, 809

Eyre, A., & Binney, J. 2009a, MNRAS, 399, L160

—. 2009b, MNRAS, 400, 548

Fellhauer, M., Evans, N. W., Belokurov, V., Wilkinson, M. I., & Gilmore, G. 2007, MNRAS, 380, 749

Fellhauer, M., Belokurov, V., Evans, N. W., et al. 2006, ApJ, 651, 167

Geha, M., Willman, B., Simon, J. D., et al. 2009, ApJ, 692, 1464

Gilmore, G., Wilkinson, M. I., Wyse, R. F. G., et al. 2007, ApJ, 663, 948

Gnedin, O. Y., & Primack, J. R. 2004, PhRvL, 93, 061302

Grillmair, C. J. 2006, ApJ, 645

Grillmair, C. J., Hetherington, L., Carlberg, R. G., & Willman, B. 2015, ApJ, 812, L26

Hayashi, E., Navarro, J. F., & Springel, V. 2007, MNRAS, 377, 50

Heggie, D., & Hut, P. 2003, The Gravitational Million-Body Problem: A Multidisciplinary Approach to Star Cluster Dynamics, (1st ed.; Cambridge, UK: Cambridge University Press)

Helmi, A. 2004, PASA, 21, 212

Hernquist, L. 1990, ApJ, 356, 359

Høimyr, N., Blomer, J., Buncic, P., et al. 2012, in Journal of Physics Conference Series, Vol. 396, Journal of Physics Conference Series (Washington, DC: IOP), 032057

Hooper, D., & Goodenough, L. 2011, PhLB, 697, 412

Ibata, R., Lewis, G. F., Irwin, M., Totten, E., & Quinn, T. 2001, ApJ, 551, 294

Jin, S., & Lynden-Bell, D. 2007, MNRAS, 378, L64

Jo, J., & DM-Ice Collaboration. 2016, in Proceedings of the 38th International Conference on High Energy Physics (ICHEP2016). (Chicago, USA: PoS), 1223

Johnston, K. V., Law, D. R., & Majewski, S. R. 2005, ApJ, 619, 800

Jurić, M., Ivezić, Ž., Brooks, A., et al. 2008, ApJ, 673, 864

Kennedy, G. F. 2014, MNRAS, 445, 4446

Kirby, E. N., Cohen, J. G., Simon, J. D., & Guhathakurta, P. 2015, ApJ, 814, L7

Kirby, E. N., Cohen, J. G., Simon, J. D., et al. 2017, ApJ, 838, 83

Koposov, S. E., Rix, H.-W., & Hogg, D. W. 2010, ApJ, 712, 260

Koposov, S. E., Belokurov, V., Evans, N. W., et al. 2012, ApJ, 750, 80

Krauss, L. M., & Chaboyer, B. 2003, Sci, 299, 65

Küpper, A. H. W., Lane, R. R., & Heggie, D. C. 2012, MNRAS, 420, 2700

Law, D. R., Johnston, K. V., & Majewski, S. R. 2005, ApJ, 619, 807

Law, D. R., & Majewski, S. R. 2010, ApJ, 714, 229

Law, D. R., Majewski, S. R., & Johnston, K. V. 2009, ApJ, 703, L67

Lee, S. K., Lisanti, M., Safdi, B. R., Slatyer, T. R., & Xue, W. 2016, PhRvL, 116, 051103

Macias, O., Gordon, C., Crocker, R. M., et al. 2018, NatAs, 2, 387

Majewski, S. R., Skrutskie, M. F., Weinberg, M. D., & Ostheimer, J. C. 2003, ApJ, 599, 1082

Martin, C., Amy, P. M., Newberg, H. J., et al. 2018, MNRAS, 477, 2419

Martin, N. F., de Jong, J. T. A., & Rix, H.-W. 2008, ApJ, 684, 1075

Mateu, C., Read, J. I., & Kawata, D. 2018, MNRAS, 474, 4112

McConnachie, A. W., & Côté, P. 2010, ApJ, 722, L209

Mendelson, B. 1990, Introduction to Topology, (3rd ed.; Mineola, NY: Dover Publications, Inc.)

Miyamoto, M., & Nagai, R. 1975, PASJ, 27, 533

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563

—. 1997, ApJ, 490, 493

Newberg, H. 2013, in Advancing the Physics of Cosmic Distances, Proceedings of the International Astronomical Union, IAU Symposium, Vol. 289 (Beijing, China: IAU), 74–81

Newberg, H. J., & Carlin, J. L. 2016, Tidal Streams in the Local Group and Beyond: Observations and Implications, (1st ed.; New York City, NY: Springer)

Newberg, H. J., Willett, B. A., Yanny, B., & Xu, Y. 2010, ApJ, 711, 32

Newberg, H. J., Yanny, B., Cole, N., et al. 2007, ApJ, 668, 221

Newberg, H. J., Yanny, B., Rockosi, C., et al. 2002, ApJ, 569, 245

Newby, M. 2013, PhD thesis, Rensselaer Polytechnic Institute, Troy, NY

Newby, M., Cole, N., Newberg, H., et al. 2013, AJ, 145

Niederste-Ostholt, M., Belokurov, V., Evans, N. W., et al. 2009, MNRAS, 398, 1771

Odenkirchen, M., Grebel, E. K., Kayser, A., Rix, H.-W., & Dehnen, W. 2009, AJ, 137, 3378

O'Leary, R. M., Kistler, M. D., Kerr, M., & Dexter, J. 2015, arXiv:1504.02477

Paraficz, D., Kneib, J.-P., Richard, J., et al. 2016, A&A, 594, A121

Peñarrubia, J., Koposov, S. E., & Walker, M. G. 2012, ApJ, 760, 2

Plummer, H. C. 1911, MNRAS, 71, 460

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007, Numerical Recipes 3rd Edition: The Art of Scientific Computing, (3rd ed.; Cambridge, UK: Cambridge University Press)

Price-Whelan, A. M., & Johnston, K. V. 2013, ApJ, 778, L12

Prior, S. L., Da Costa, G. S., & Keller, S. C. 2009, ApJ, 704, 1327

Quinn, D. P., Wilkinson, M. I., & Irwin, M. J. 2008, AN, 329, 1071

Retana-Montenegro, E., van Hese, E., Gentile, G., Baes, M., & Frutos-Alfaro, F. 2012, A&A, 540, A70

Rubner, Y., Tomasi, C., & Guibas, L. J. 2000, IJCV, 40, 99

Sales, L. V., Helmi, A., Starkenburg, E., et al. 2008, MNRAS, 389, 1391

Sanderson, R. E., & Helmi, A. 2013, MNRAS, 435, 378

Savage, C., Newberg, H. J., Freese, K., & Gondolo, P. 2006, JCAP, 7, 003

Sellwood, J. A. 2003, ApJ, 587, 638

Sesar, B., Ivezić, Ž., Stuart, J. S., et al. 2013, AJ, 146, 21

Shelton, S., Weiss, J., Rayment, C., et al. In prep.

Siegal-Gaskins, J. M., & Valluri, M. 2008, ApJ, 681, 40

Smith, R., Sánchez-Janssen, R., Fellhauer, M., et al. 2013, MNRAS, 429, 1066

Strigari, L. E., Bullock, J. S., & Kaplinghat, M. 2007, ApJ, 657

Tonini, C., Lapi, A., & Salucci, P. 2006, ApJ, 649, 591

Vera-Ciro, C., & Helmi, A. 2013, ApJ, 773, L4

Verlet, L. 1967, PhRv, 159, 98–103

Vivas, A. K., Zinn, R., & Gallart, C. 2005, ApJ, 129, 189

von Neumann, J. 1951, Various Techniques Used in Connection with Random Digits, Vol. 12 Washington, DC: DOC)

Wall, J. V., & Jenkins, C. R. 2012, Practical Statistics for Astronomers (Cambridge Observing Handbooks for Research Astronomers), (2nd ed.; Cambridge, UK: Cambridge University Press)

Weiss, J. 2018, PhD thesis, Rensselaer Polytechnic Institute, Troy, New York

Weiss, J., Newberg, H. J., Newby, M., & Desell, T. 2018, ApJS, 238, 17

Willett, B. A. 2010, PhD thesis, Rensselaer Polytechnic Institute, Troy, NY

Willett, B. A., Newberg, H. J., Zhang, H., Yanny, B., & Beers, T. C. 2009, ApJ, 697, 207

Wolf, J., Martinez, G. D., Bullock, J. S., et al. 2010, MNRAS, 406, 1220

Xenon100 Collaboration, Aprile, E., Arisaka, K., et al. 2012, APh, 35, 573

Yoon, J. H., Johnston, K. V., & Hogg, D. W. 2011, ApJ, 731, 58

Zhao, H. 1995, MNRAS, 278, 488

A Algorithmic Options

This section describes the N-body algorithmic options present, which may be instrumental for future researchers using the N-body code package. In this chapter we describe the parameter file which is sent to users and acts as a front end to our simulations. We also describe the many developer options which were implemented by the author that can and have been utilized for various purposes. These options ‘open up’ the algorithm package, allowing for many different types of study apart from the main goal of determining the dark matter content of dwarf galaxies.

While many of these options are utilized through the parameter files, some are determined from the command line. There are the numerous flags that can be set, both at the moment of compilation of the algorithm and at run time. Some of these flags will be described here.

We will also give the command line requirements for the compilation and running of the N-body application of the MilkyWay@home code package.

A.1 Parameter File

The parameter files provide a sort of front end through which a user can set various options for the simulation. The parameter files are scripts written in the LUA scripting language. They are interpreted by the main N-body algorithm which is written mostly in C. The ‘LUA files’ allow for a convenient way of changing algorithm options without having to recompile the MilkyWay@home algorithm package. There are several types of LUA files which can be used. The two main classes are the client side LUA files and the developer LUA files. The client side LUA file provides the most basic of options. Only the options absolutely necessary for running a simulation are present. The developer LUA files provide far more options, only useful to N-body developers and users using the algorithm locally (not on MilkyWay@home). In order to use the client side LUA files, the algorithm must be built with developer options off. The client side LUA files have the general naming scheme of “EMD_v172_■.lua” where “172” is the version number, which is changed when a new version of the algorithm is released, and the ■ indicate any addition information about that specific LUA file. The developer LUA files have the naming scheme “for_developers_■.lua”,

---

Portions of this chapter are in preparation to be submitted to: S. Shelton, “Reconstructing the Orphan Stream Progenitor with MilkyWay@home Volunteer Computing,” ApJ.

where the ■ is again any other information about that specific LUA file.

Below, we describe the various options present in the developer LUA files. Since all options in the client side LUA files are also present in the developer LUA files, this provides a complete list.

A.1.1 Standard Model Settings

The standard settings are given in Table A.1 below. They control the version of the algorithm to use, the integration method, and the setting of the simulations. It also provides options for the type of model being utilized.

Table A.1: Simulation model options available in the LUA files.

LUA File - Model Settings	
Setting	Description
nbodyMinVersion	The lowest version number of the algorithm that will be accepted for use with this parameter file.
use_tree_code	true; false. Use the Barnes-Hut tree algorithm for force calculations. False uses the exact force calculations.
run_null_potential	true; false. True runs the simulation without the Milky Way potential, placing the simulation at the origin, in empty space.
totalBodies	any integer. The number of bodies used in the simulation when using a dwarf galaxy
ModelComponents	2;1;0. The number of components the model has. 2 for the two component Plummer model. 1 for the single component Plummer model. 0 turns off the dwarf model.
manual_bodies	true; false. Uses a user inputted list of bodies in the simulation. Can be used with a dwarf model or by itself.

A.1.2 Histogram Options

There are several options and adjustments that can be made that alter the creation of the histogram. They include the area in the sky in which the histogram can be made, the degree to which outliers are rejected, and the transformation angles for the stream being investigated. These are given in Table A.2 below.

Table A.2: Histogram settings available in the LUA files.

LUA File - Histogram Settings	
Setting	Description
lda_bins	Number of Λ bins to use
lda_lower_range; lda_upper_range	The Λ bin interval over which to make the histogram. Any interval between [-180,180]
bta_bins	Number of β bins to use. Usually 1
bta_lower_range; bta_upper_range	The β bin interval over which to make the histogram. Any interval between [-180,180]. Usually a small interval centered around the origin
SigmaCutoff	The number of σ used for outlier rejection in calculating dispersions
Correction	The correction factor required for correcting the probability distribution due to outlier rejection. Value is determined by the σ cutoff used and must be calculated by the user. Note: used to correct σ^2.
phi;theta;psi	Found in the makeHistogram() function. Transformation angles for the Orphan Stream as taken from Newberg et al. (2010)

A.1.3 Likelihood Options

There are also options that can be adjusted for the comparison algorithm. They include the method used for the geometry component of the likelihood, which dispersion components to use, β, line-of-sight velocity, both or neither, and controls over the best likelihood calculation. These are listed in Table A.3 below.

A.1.4 Other Developer Options

There are a number of other options that are primarily used by and are most useful to developers. They have proven extraordinarily useful in debugging and/or testing new algorithms. They can also be used when running a simulation for purposes other than an optimization. For example, using some of these options, the algorithm has been used to study the effects of dynamic friction on dwarfs over time, simulating the orbit of a group of stars in the Milky Way disk, among others. These are listed in Table A.4 below.

Table A.3: Comparison algorithm options available in the LUA files

LUA File - Likelihood Settings	
Setting	Description
nbodyLikelihood_Method	Histogram geometry component comparison method. Only EMD should be used. For a complete list of alternatives see the comparison algrotihm. ex: ‘EMD’; ‘Poisson’; ‘ChisqAlt’; ‘KullbackLeibler’
use_best_likelihood	true; false. Whether or not to use the best likelihood algorithm as described in Chapter 6.
best_like_start	The fraction of the entire simulation after which the best likelihood algorithm starts. Decimal between [0;1]
use_beta_disps	true; false. Whether or not to use the β dispersion component of the comparison algorithm.
use_vel_disps	true; false. Whether or not to use the line-of-sight velocity dispersion component of the comparison algorithm. Can be used in conjunction with the β dispersion component.

Table A.4: Other developer options available in the LUA files.

LUA File - Developer Options	
Setting	Description
print_reverse_orbit	true; false. prints the initial reverse orbit integration to a file
print_out_parameters	true; false. prints out the input parameters in simulation and solar units
useMultiOutputs	true; false. Algorithm can produce multiple coordinate outputs. This turns that option on
freqOfOutputs	The frequency of how often the simulation writes a coordinate output. Number represents the number of timesteps between outputs.
timestep control	true; false. User can manually control the number of timesteps the simulations runs. This option should be used with care
Ntime_steps	The number of timesteps the simulation will run. Any integer between 0 and the number of timesteps the simulation should have run. Can be set to 0, which will not run any integration. Any outputs will be of initial body arrangement

A.1.5 Milky Way Potential Struct

The makePotential() function in the lua files determines the Milky Way potential model used. It creates a struct object which requires three components, one for each component of the Milky Way potential. The three listed components are ‘spherical’, ‘disk’, and ‘halo’. The ‘spherical’ component is used for the central bulge, the ‘disk’ component is used for the stellar disk, and the ‘halo’ component is used for the dark matter halo. For each component a model is chosen and the required model parameter values are listed. An example of how this struct object is created is given below:

Table A.5: Struct to determine the Milky Way potential.

Milky Way Potential Struct

```
return Potential.create{
        spherical = Spherical.spherical{ mass = ■, scale = ■ },
        disk = Disk.miyamotoNagai{ mass = ■, scaleLength = ■, scaleHeight = ■ },
        halo = Halo.logarithmic{ vhalo = ■, scaleLength = ■, flattenZ = ■ } }
```

Where the gray boxes indicate where input values should be placed. Instead of this struct, a nil value can be returned, which would then mean the simulation will be performed in empty space.

A.1.6 The Context Struct

The makeContext() function returns a struct which sets the many required variables needed in the simulation. Many of the developer options available are sent into the simulation using this struct. Other than the previously listed developer options, there are two other options that are sent into the simulation using this struct that are not generally altered. The first is 'useQuad', which determines whether or not to use quadrapole moments in the gravitational interactions between bodies. This option is normally set to true. The other is 'theta', which sets the θ parameter used by the Barnes-hut tree algorithm as previously described is Section 5.3. This value is normally set to 1. The struct is of the form as shown below. It is shown in its entirety because of the specific formatting of the struct members.

Table A.6: Struct to determine the Context.

Simulation Context Struct

```
return NBodyCtx.create {
          timeEvolve = evolveTime,
          timestep = get_timestep(),
          eps2 = calculateEps2(totalBodies, soften_length ),
          criterion = criterion,
          useQuad = true,
          useBestLike = use_best_likelihood,
          BestLikeStart = best_like_start,
          useVelDisp = use_vel_disps,
          useBetaDisp = use_beta_disps,
          Nstep_control = timestep_control,
          Ntsteps = Ntime_steps,
          BetaSigma = SigmaCutoff,
          VelSigma = SigmaCutoff,
          BetaCorrect = Correction,
          VelCorrect = Correction,
          MultiOutput = useMultiOutputs,
          OutputFreq = freqOfOutputs,
          theta = 1.0 }
```

The function determines the ‘soften_length’ variable, the ‘center of mass’ scale length, before sending it into the calculateEps2() function, which calculates the softening parameter as described in 5.5. The get_timestep() function is also in the parameter files. It determines the timestep as described in Section 5.4. The variable ‘criterion’ is determined by the developer option ‘use_tree_code’ and is equal to either ‘TreeCode’ or ‘Exact’.

A.1.7 Setting the Initial Bodies

The makeBodies() function is responsible for the reverse orbit calculation and the dwarf galaxy initialization. The phase space location where the initial dwarf model is placed at the beginning of the simulation as described in Section 2.2 is determined by the reverseOrbit() function in the N-body algorithm. The function call is similar to the structs described in the preceding sections. It takes in the Milky Way potential model, the Galactocentric Cartesian position and velocities of the starting location of the reverse orbit, the evolution time, and the timestep.

Table A.7: Function to determine the reverse orbit position.

Reverse Orbit Function

```
finalPosition, finalVelocity = reverseOrbit {
            potential = potential,
            position = lbrToCartesian(ctx, Vector.create(l, b, r)),
            velocity = Vector.create(v_x, v_y, v_z),
            tstop = revOrbTime,
            dt = ctx.timestep / 10.0 }
```

The variables 'finalPosition' and 'finalVelocity' are the vectors storing the phase space coordinates for where the dwarf model should be placed. The values for (l,b,r) and (v_x, v_y, v_z) above are listed as the variables 'orbit_parameter_■' in the lua files and the values are taken from Newberg et al. (2010). The function lbrToCartesian() is a function written in the source code which converts the (l,b,r) coordinates to Galactocentric Cartesian coordinates. In the case that the simulation is taking place in empty space, the 'finalPosition' and 'finalVelocity' variables are set to (0,0,0), the origin in phase space.

The dwarf model is also determined by a function call. Depending on the model chosen, the function called and parameters sent differ. For the single component Plummer model, the function call is as follows:

Table A.8: Function that generates a single component Plummer model.

Single Component Plummer Model

```
firstModel = predefinedModels.plummer {
            nbody = totalBodies,
            prng = prng,
            position = finalPosition,
            velocity = finalVelocity,
            mass = mass_l,
            scaleRadius = rscale_l,
            ignore = false }
```

The dwarf model is stored in the variable firstModel. As this indicates, multiple models can be made, each using this type of function call. All the models are returned by the makeBodies() and are subsequently sent into the simulation. The 'prng' is the link to the random number generator. The 'ignore' parameter is a flag used to mark the matter type of the bodies. For single component models, the baryonic input parameters are used for the dwarf model. For the two component model, the function call differs slightly as shown below:

Table A.9: Function that generates a two component Plummer Model.

Two Component Model

```
firstModel = predefinedModels.mixeddwarf {
            nbody = totalBodies,
            prng = prng,
            position = finalPosition,
            velocity = finalVelocity,
            comp1 = Dwarf.plummer{mass = mass_l, scaleLength = rscale_l},
            comp2 = Dwarf.plummer{mass = mass_d, scaleLength = rscale_d},
            ignore = true }
```

As the above shows, the model used for each component of the dwarf has to be specified. Currently, only the Plummer model has been tested in the way described in Chapter 4. The makeBodies() function also allows for a user inputted list of bodies that can be used in the simulation by itself or in conjunction with a dwarf model. The function call to use a user inputted list of bodies is as follows:

Table A.10: Function for generating a user inputted model.

User Inputted Model
manualModel = predefinedModels.manual_bodies { body_file = manual_body_file }

Where 'manual_body_file' is the file name of the body list. The manual model option is only available to developers. It provides a way to solve the equations of motion in very specific cases that may not be related to the main project goals. For example, this option can and has been used to simulate the motion of groups of stars around the Milky Way, or to create halo substructure to influence the orbit of a dwarf model.

In makeBodies(), depending on the chosen options from Section A.1.1, the returned value may be a single plummer model or a two component model. Furthermore, each model can be returned by itself or with a user inputted list of bodies. This can easily be expanded to create several models for the simulation to run simultaneously by calculating the appropriate reverse orbit values, creating new model function calls, and returning them.

A.1.8 Taking in Parameters

The developer parameter files expect a minimum of 6 input parameters. The first is the backward evolution time in gigayears, and the second is a deprecated parameter, which can be set to any value. The second value was a time ratio used to determine the forward evolution time, which was made defunct by the best likelihood calculation. However, this parameter must still be included because it is expected by the MilkyWay@home server. The third parameter is the baryonic scale radius, in kiloparsecs, and the fourth is the radius ratio. The fifth parameter is the baryonic mass, in structural units, and the sixth is the mass ratio. These values are each rounded to 9 decimal places when taken into the parameter files. This is done because the command line precision differs between some machines, leading to slightly different parameters and thus different likelihood values. The seventh parameter is optional and is the name and path to the user inputted body list. This can only be included in the developer parameter file.

The random number seed can also be sent in by command line, as will be described in the following section. Whether it is sent in or not, the seed is hardcoded in the parameter file where the input parameters are read in.

A.2 Usage

The MilkyWay@home algorithm is a free and open source package. It is publicly available on Github at `https://github.com/Milkyway-at-home`. There are several repositories. The 'milkywayathome client' repository contains all of the code required to run N-body on a single computer. The 'milkywayathome workunits' repository contains sample parameter files used in MilkyWay@home. The 'milkyway server' repository contains the code that runs the server which sends out and receive workunits. The 'tao' repository contains a version of the Toolkit for Asynchronous Optimization (TAO), which are the various optimization algorithms. The 'boinc' repositories contain versions of the Berkeley Open Infrastructure for Network Computing (BOINC) algorithm that enables us to utilize volunteer computing. We describe how to download, compile and run the N-body application contained in the 'milkywayathome client' repository using a linux terminal.

A.2.1 Downloading the Package

In order to download the client repository, 'git' package manager must be installed. After copying the url of the repository, in a linux terminal, the repository can be downloaded by running:

Table A.11: Command for downloading the MilkyWay@home client code package.

Download Command
$git clone `https://github.com/Milkyway-at-home/milkywayathome_client.git`

Downloading the other repositories can be done in a similar fashion, using the url for that repository. After the repository has been downloaded, navigate into the directory. The initial clone downloads only our base algorithm. The repository also contains links to several other repositories as submodules which also need to be downloaded. This is done by running the commands:

The first command initializes the links to the submodules. The second performs the download, similar to the git clone command.

Table A.12: Command for downloading submodules contained within the MilkyWay@home client code package.

Update Command
$git submodule init
$git submodule update –recursive

A.2.2 Compiling

It is recommended to compile the MilkyWay@home algorithm in a separate directory than the one in which it was downloaded. In order to compile the algorithm CMAKE must be installed on the machine being used. CMAKE automatically generates the makefiles necessary to build the algorithm. To build the binaries needed for running the algorithm first navigate into the directory in which the build will take place. Running cmake, followed by compilation flags, and then the path to the algorithm directory will create the makefiles. After that is complete, running make in that same directory will build the algorithm binaries. There are many compilation flags. The primary flags are given in Table A.13. A complete listing of the options and associated flags can be found in the various CMakeLists files in the algorithm.

Table A.13: Important flags used in the compilation of the N-body algorithm.

Important Compilation Flags	
Flag	Description
-DNBODY_DEV_OPTIONS	ON;OFF. Set to On in order to access developer options. Off to use the client side parameter files
-DNBODY_OPENMP	ON;OFF. Build the algorithm single threaded (OFF) or multithreaded (ON).
-DNBODY_STATIC	ON;OFF. Build the binary using static libraries. This should be set to ON to reduce machine library dependencies.
-DNBODY_GL	ON;OFF. Builds the visualizer. Additional package installations required, including open gl.
-DSEPARATION	ON;OFF. Option for building the Separation code. Can be excluded. Defaults to ON.
-DNBODY	ON;OFF. Option for building the N-body code. Can be excluded. Defaults to ON.
-DCMAKE_BUILD_TYPE	Debug; Release; RelWithDebInfo; MinSizeRel. Set to Release for a normal build which can be sent to clients. The other options builds the binary with debugging information.

An example compilation command which builds only the multithreaded N-body application statically, with the developer options on is as follows:

Table A.14: Command for compiling the MilkyWay@home code package.

Compilation Command

```
$cmake -DCMAKE_BUILD_TYPE=Release -DNBODY_DEV_OPTIONS=ON
        -DNBODY_GL=OFF -DNBODY_STATIC=ON -DSEPARATION=OFF
        -DNBODY_OPENMP=ON ../milkywayathome_client/
$make
```

The make commands builds the algorithm using the makefiles generated by CMAKE. There is an optional flag '-j' that can be added to the 'make' command which runs the build multithreaded. This builds the code much faster. It is recommended that this be excluded during the first build of the algorithm.

A.2.3 Running a Simulation

Once the algorithm is compiled, a simulation can be run. There are also a number of flags which can be used when running the simulation. Some are required and some provide different options to the user. Below, in Table A.15, is a list of the main flags used in the simulation:

Table A.15: Important flags used in the running of the N-body algorithm.

Important Runtime Flags	
Flag	Description
–help	Lists all the possible runtime flags and descriptions
-f	Required. Path and file name of the LUA file being used.
-o	Path and file name instruction for the simulation to place the coordinate output file. Not required if other output selected.
-z	Path and file name instruction for the simulation to place the histogram file. Not required if other output selected.
-h	Path and file name of the input histogram file used in comparing to simulation. Not required.
-n	Number of threads to use in the simulation (If compiled with OPENMP). Not required. If not included, system sets number of threads
-x; -b;	Only one option can be used. Instructions on coordinate output format. Not required. No flag means lbr positional output. -x for Cartesian positional output. -b for both.
-P	Print the percentage progress of the simulation to standard output. Not required.
-u	Runs the visualizer. May need additional package installation and compilation with OPENGL. Not required.
-s; -S; -V; -D	Used when comparing two histograms without simulation. Flag for the second histogram. Only one option can be used. -s matches the histogram without dispersion components. -S for β dispersions only. -V for velocity dispersions only. -D for both β and velocity dispersions.

The flags are listed with their required inputs after calling the binary. After the flags, a list of the require input parameters should be given. The required input parameters can changed depending on the parameter (LUA) file selected, so care must be used when listing the input parameters. At least one output option must always be selected when running a simulation. If a input histogram is not given, the simulation will run and produce the selected outputs without running the comparison algorithms. The coordinate outputs are only used by developers. We have found it useful to provide options on changing the format of those outputs. With the implementation of the dispersion comparison algorithms, options on which dispersion algorithm to use were implemented. Two example run commands are given below.

Table A.16: Command for running an N-body simulation.

Run Command

```
$ ./PATH TO BINARY/BINARY NAME
          -f LUA FILE
          -h INPUT HISTOGRAM FILE
          -z OUTPUT HISTOGRAM FILE
          -o COORDINATE OUTPUT FILE
          -b -p -n NUMBER OF THREADS
          INPUT PARAMETER LIST
```

This command will take in a histogram file provided by the user, run a simulation and compare the input histogram with the simulation. Both the simulation histogram and the coordinate output file will be written to the user machine. The coordinate output will have both lbr and Cartesian positional outputs. As the simulation runs, the percent progress will be written to standard out. Note, the entire command should be put into the terminal without line breaks, and with no spaces in the file names. It is written this way here for readability. Another command is given below.

Table A.17: Command for comparing two histograms without running a simulation.

Histogram Comparison Command

```
$ ./PATH TO BINARY/BINARY NAME
          -h INPUT HISTOGRAM FILE 1
          -S INPUT HISTOGRAM FILE 2
```

This command will take in two histograms and compare them. The comparison will use β dispersions only. The same note as the previous command applies here as well.

B Computationally Simplified Dispersion

Here we derive the computationally simplified version, Equation 6.4, of the dispersion, Equation 6.3. By using the former equation, the average in each bin does not need to be first calculated in an initial loop, greatly reducing the amount of computation needed. Instead, the sum can be performed directly. We perform this derivation for a general parameter, α. In our algorithm this can be either line-of-sight velocity or β. We begin with the dispersion in the parameter:

$$\sigma^2 = \frac{1}{N-1}\sum_i^N (\alpha_i - \bar{\alpha})^2. \tag{B.1}$$

This can be expanded to:

$$\sigma_\alpha^2 = \sum_i^N \frac{\alpha_i^2}{N-1} - \frac{2\bar{\alpha}}{N-1}\sum_i^N \alpha_i + \sum_i^N \frac{\bar{\alpha}^2}{N-1}, \tag{B.2}$$

where $\bar{\alpha}$ is simply a number, and can be moved out of the summations. Therefore, the final summation equates to N sums of 1, and is thus equal to N:

$$\sigma_\alpha^2 = \sum_i^N \frac{\alpha_i^2}{N-1} - \frac{2N\bar{\alpha}}{N-1}\sum_i^N \frac{\alpha_i}{N} + \frac{N\bar{\alpha}^2}{N-1}. \tag{B.3}$$

Note, we have also inserted a factor of N/N in the second summation. With this insertion, the middle summation is equal to $\bar{\alpha}$ since $\bar{\alpha} = \sum_i^N \alpha_i/N$. We then have:

$$\sigma_\alpha^2 = \sum_i^N \frac{\alpha_i^2}{N-1} - \frac{2N\bar{\alpha}^2}{N-1} + \frac{N\bar{\alpha}^2}{N-1}. \tag{B.4}$$

Thus, we arrive at Equation 6.4:

$$\sigma_\alpha^2 = \sum_i \frac{\alpha_i^2}{N-1} - \left(\frac{N}{N-1}\right)\left(\sum_i \frac{\alpha_i}{N}\right)^2. \tag{B.5}$$

C Error in Dispersion

We derive here the theoretical error in the dispersion equation. We begin first by defining the two rules of which we shall make primary usage. The first is the propagation of error when squaring a function with error:

$$(f \pm \delta_f)^2 = f^2 \pm 2f\delta_f. \tag{C.6}$$

The second is the propagation of error when adding two values with error:

$$(A \pm \delta_A) + (B \pm \delta_B) + ... = (A + B + ...) \pm \sqrt{\delta_A^2 + \delta_B^2 + ...}. \tag{C.7}$$

We perform this derivation for a general parameter, α. In our algorithm this can be either line-of-sight velocity or β. Starting with the equation for dispersion:

$$\sigma^2 = \frac{1}{N-1}\sum_i^N (\alpha_i - \bar{\alpha})^2, \tag{C.8}$$

which is rewritten to include the errors:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\sum_i^N \left[(\alpha_i \pm \delta_{\alpha_i}) - (\bar{\alpha} \pm \delta_{\bar{\alpha}})\right]^2. \tag{C.9}$$

Using the addition rule, this can then be written as:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\sum_i^N \left[(\alpha_i - \bar{\alpha}) \pm \sqrt{\delta_{\alpha_i}^2 + \delta_{\bar{\alpha}}^2}\right]^2. \tag{C.10}$$

By definition, the error in the coordinate is $\alpha_i \pm \sigma$, so $\delta_{\alpha_i} = \sigma$. The error in the mean is $\bar{\alpha} \pm \frac{\sigma}{\sqrt{N}}$, so $\delta_{\bar{\alpha}} = \frac{\sigma}{\sqrt{N}}$. Inserting these values and simplifying:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\sum_i^N \left[(\alpha_i - \bar{\alpha}) \pm \sigma\sqrt{\frac{N+1}{N}}\right]^2. \tag{C.11}$$

Using the squaring rule:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\sum_i^N \left[(\alpha_i - \bar{\alpha})^2 \pm 2(\alpha_i - \bar{\alpha})\sigma\sqrt{\frac{N+1}{N}}\right]. \tag{C.12}$$

This summation is now just a sum of (A + B + ...). Therefore, the equation becomes:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\left[\sum_i^N (\alpha_i - \bar{\alpha})^2\right] \pm \frac{1}{N-1}\sqrt{\sum_i^N \left(2(\alpha_i - \bar{\alpha})\sigma\sqrt{\frac{N+1}{N}}\right)^2}. \tag{C.13}$$

This then can be written:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\left[\sum_i^N (\alpha_i - \bar{\alpha})^2\right] \pm \frac{2\sigma}{N-1}\sqrt{\frac{N+1}{N}}\sqrt{\sum_i^N (\alpha_i - \bar{\alpha})^2}. \tag{C.14}$$

The summation in the error value is a simple rearrangement of the dispersion, Equation C.8:

$$\sigma^2 \pm \delta_{\sigma^2} = \frac{1}{N-1}\left[\sum_i^N (\alpha_i - \bar{\alpha})^2\right] \pm \frac{2\sigma}{N-1}\sqrt{\frac{N+1}{N}}\sqrt{\sigma^2(N-1)}. \tag{C.15}$$

Simplifying:

$$\sigma^2 \pm \delta_{\sigma^2} = \sigma^2 \pm 2\sigma^2\sqrt{\frac{N+1}{N-1}\frac{1}{N}}. \tag{C.16}$$

However, this is the error in σ^2. Using the squaring rule, we have $\sigma^2 \pm \delta_{\sigma^2} = \sigma^2 \pm 2\delta_\sigma\sigma$ where $\sigma \pm \delta_\sigma$. Therefore, the equation above becomes:

$$\sigma^2 \pm 2\delta_\sigma\sigma = \sigma^2 \pm 2\sigma^2\sqrt{\frac{N+1}{N-1}\frac{1}{N}}. \tag{C.17}$$

Comparing the left and right sides of the equation, it is apparent that:

$$\delta_\sigma = \sigma\sqrt{\frac{N+1}{N-1}\frac{1}{N}}. \tag{C.18}$$

Finally, the error in the theoretical σ is given by:

$$\sigma \pm \sigma\sqrt{\frac{N+1}{N-1}\frac{1}{N}}. \tag{C.19}$$

D Coordinate Transformations

D.1 Reverse Orbit

The (l,b,r) coordinates of the Orphan Stream position must be converted to Galactocentric Cartesian in order to perform the reverse orbit calculations. The conversions from (l,b,r) to heliocentric Cartesian is given by:

$$
\begin{aligned}
x_\odot &= \mathrm{R}\cos l \sin b, \\
y &= \mathrm{R}\sin l \sin b, \\
z &= \mathrm{R}\cos b.
\end{aligned}
\tag{D.20}
$$

The Galactocentric coordinates are then found by shifting the x-coordinate, $x = x_\odot - 8\mathrm{kpc}$.

D.2 Histogram

To create a histogram, the Galactocentric coordinates used in the simulation must be transformed to Orphan (Λ, β). The Cartesian coordinates are shifted to be centered around the Sun by shifting the x-coordinate: $x_\odot = x + 8\mathrm{kpc}$. These coordinates are then transformed to (l, b, r):

$$
\begin{aligned}
l &= \tan^{-1}\left(\frac{y}{x_\odot}\right), \\
b &= \tan^{-1}\left(\frac{z}{\sqrt{x_\odot^2 + y^2}}\right), \\
r &= \sqrt{x_\odot^2 + y^2 + z^2}.
\end{aligned}
\tag{D.21}
$$

Finally, these coordinates are rotated to align with the Orphan Stream using a rotation matrix given by Newberg et al. (2010):

$$
\begin{pmatrix} \cos\beta\cos\Lambda \\ \cos\beta\sin\Lambda \\ \sin\beta \end{pmatrix} = M \begin{pmatrix} \cos b\cos l \\ \sin b\sin l \\ \sin b \end{pmatrix}, \tag{D.22}
$$

where M is given by:

$$
M = \begin{pmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & \sin\psi\sin\theta \\ -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & \cos\psi\sin\theta \\ \sin\theta\sin\phi & -\sin\theta\cos\phi & \cos\theta \end{pmatrix}.
$$

The rotation angles are unique to a tidal stream. Since we are currently investigating the Orphan Stream, these angles are given by $(\phi, \theta, \psi) = (128^\circ\!.79,\ 54^\circ\!.39,\ 90^\circ\!.70)$ (Newberg et al., 2010).

When a histogram is created, it is bound between a range in (Λ, β). Therefore, only bodies in this range will be included in the histogram. The histograms are usually one dimensional, binning the density along the stream, Λ.

E χ^2 Probability Metric for the Dispersion

We briefly mentioned in our description of the dispersion component of the likelihood that we previously used a different probability function to grade the fitness of the stream width. This method also led to Equation 6.17, but its derivation and interpretation differ. We calculated a σ difference between the dispersion in each corresponding bin of the two sets of histograms, in a similar way as with the mass cost component. We then summed the difference to calculate a χ^2 value. The interpretation here was that we were comparing the two histograms as a whole, and calculating a probability from the sum of the differences, instead of finding the probability that each bin were the same dispersion and multiplying the probabilities. Using the former interpretation required use of the χ^2 probability distribution, given by

$$f(x,k) = \begin{cases} \frac{x^{\frac{k}{2}-1}e^{-\frac{x}{2}}}{2^{k/2}\Gamma(k/2)} & x > 1 \\ 0 & x < 1 \end{cases}, \tag{E.23}$$

where x is the N_σ^2 calculated previously, and k is number of bins in our histograms. In practice, this function was scaled. In order to utilize the χ^2 distribution for our purpose, we first write it in log space:

$$\ln(f(x,k)) = \begin{cases} (\frac{k}{2} - 1)\ln(N_\sigma^2) - \frac{N_\sigma^2}{2} & N_\sigma^2 > 0 \\ 0 & N_\sigma^2 < 0 \end{cases}. \tag{E.24}$$

Note, we have dropped all leading multiplicative coefficients and replaced x with N_σ^2. The coefficients scale the distribution values; dropping the coefficients is allowed because the shape of the distribution is retained. The χ^2 distribution as written here is shown as a solid black line in Figure E.1. The plots show the probability that two histograms are the same given a measured N_σ as a function of N_σ^2 and N_σ.

The logarithm of the other two probability distributions described so far, that produced by the EMD and the mass cost function, have a peak of zero. In fact, the EMD was scaled for this to occur. This is not the case with the χ^2 probability distribution as so far written. Therefore, we must scale the χ^2 distribution to agree with the other components of the likelihood. Therefore, the $\ln(\chi^2)$ distribution is shifted down by the peak value. The peak occurs at $N^2_{\sigma,peak} = 2(\frac{k}{2} - 1)$. Shifting the distribution then augments Equation E.24 by adding to it the function value at $N^2_{\sigma,peak}$, yielding the following:

$$\ln(f(x,k)) = \begin{cases} (\frac{k}{2} - 1)\ln(N^2_\sigma) - \frac{N^2_\sigma}{2} - (\frac{k}{2} - 1)\left[\ln(k-2) - 1\right] & N^2_\sigma > 0 \\ 0 & N^2_\sigma < 0 \end{cases} . \tag{E.25}$$

This is shown as a dashed blue line in Figure E.1. Another issue with this distribution lies in the shape. As the N^2_σ decreases, the probability increases until it reaches the peak, after which the probability quickly decreases. This seems counterintuitive because a smaller value of N^2_σ corresponds to less of a accumulative difference between the two histograms and should correspond to a higher probability that they are the same. The peak of the distribution occurs at k - 2, which is the degrees of freedom in our histograms. According to the distribution, a N^2_σ less than this is indicates the two histograms are more similar than could be expected, given the errors. However, when using test data created with the same seed as the simulations, we can, in theory, recover the exact answer, with a N^2_σ of zero. Since we do not want the likelihood to be worse for histograms that are more similar, we fix the probability for $N^2_\sigma < (k-2)$. We replace it with a piecewise constant value set to the peak of the probability distribution. This is shown by the dotted red line in Figure E.1. Therefore, instead of this component of the likelihood reporting a worse and worse match for more and more similar distributions, it will not contribute at all. In effect, this component of the likelihood helps guide the optimization algorithm through the likelihood surface while the simulated distribution is “very” different from the comparison distribution. The EMD and mass cost component will help “fine tune” the distribution once it is similar enough to the comparison distribution.

The final version of the the χ^2 distribution used is then given by a piecewise function:

$$\ln(f(x,k)) = \begin{cases} (\frac{k}{2}-1)\ln(N_\sigma^2) - \frac{N_\sigma^2}{2} - (\frac{k}{2}-1)\left[\ln(k-2)-1\right] & N_\sigma^2 > k-2 \\ 0 & N_\sigma^2 < k-2 \end{cases} . \quad \text{(E.26)}$$

From this description, it can already be seen that the method described in Section 6.4 is far simpler, and does not require changing of the actual shape of the probability distribution. It is also more consistent with the mass component, in that we find the probability that two values are different, the dispersions in each bin, and then multiply the probabilities together. In that way, the final likelihood can also be interpreted as a 2 + N component likelihood function, where N is the number of bins in our histograms.

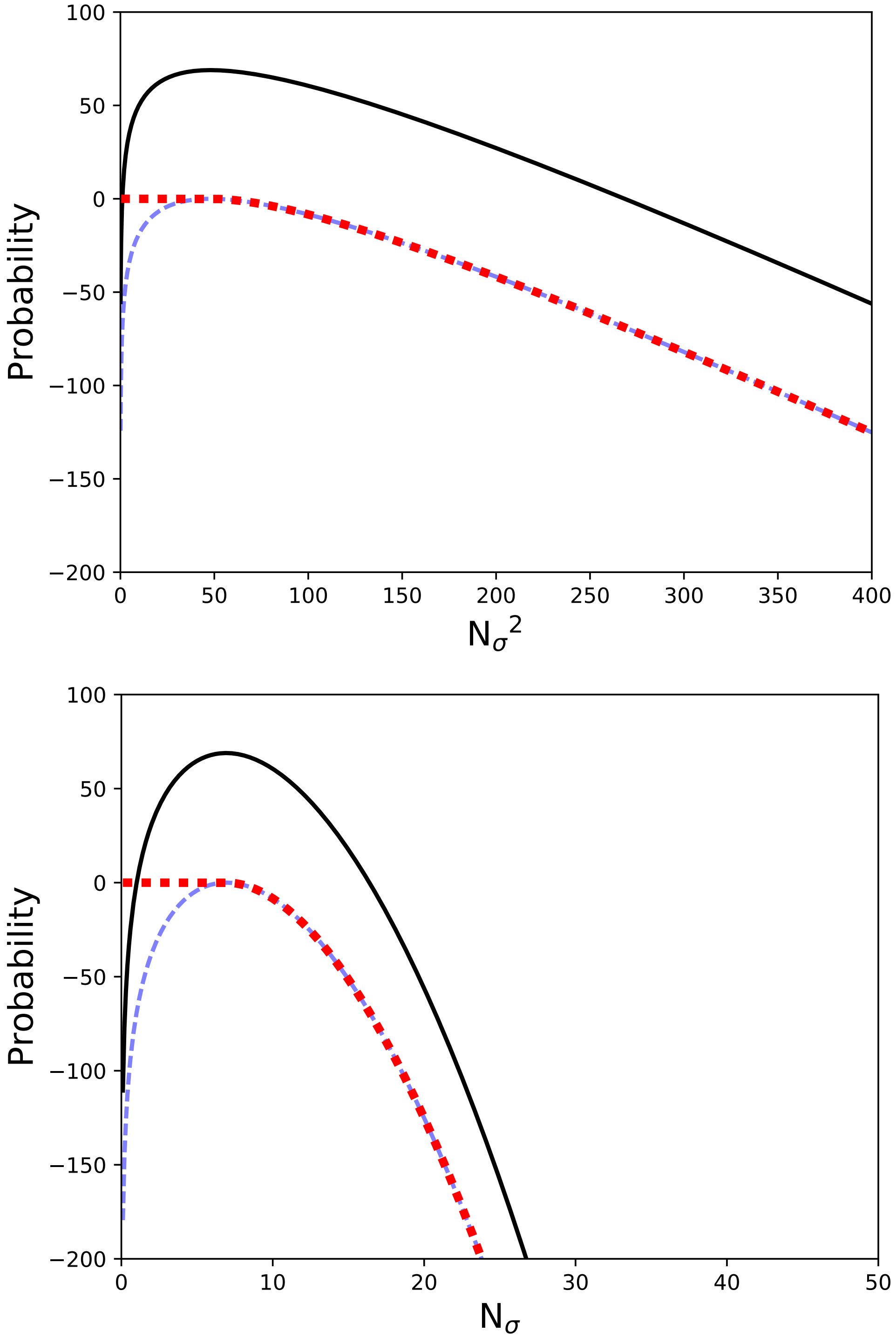

Figure E.1: Modification of the χ^2 distribution with a k value of 50. Solid Black: The original χ^2 distribution. Dashed Blue: χ^2 distribution as with the solid black but shifted down so the peak is at a probability of zero. Dotted Red: χ^2 distribution as with the dashed blue but with the probabilities set to a value of zero for $N_\sigma^2 < k - 2$.

F Line-of-Sight Velocity Dispersion with χ^2 Distribution

We present here results fitted using the line-of-sight velocity dispersion. This was done with the previous version of the dispersion likelihood value, utilizing the χ^2 probability distribution, a description of which can by found in Appendix E.

The first test was to recover the parameters used to create a test histogram where the histogram was made using the same random number seed as the simulations. The results of these runs are given in Table F.18 which show the values used to create the simulated data set, the search range, and the values recovered from MilkyWay@home for 5 identical runs. As the table shows, we were able to recover all of the parameters quite well.

Table F.18: Tables of best fit values found by the differential evolution search algorithm on MilkyWay@home. Each instance has identical search parameters but are independent of each other. The provided data set was made with the same seed as the simulations. These were all fit using the χ^2 probability distribution to compare the line-of-sight velocity dispersions. The parameters are the evolution time, the scale length of the baryonic component (R_B), the radius ratio (ξ_R), the baryonic mass (M_B), and the mass ratio (ξ_M).

Parameters	Evolve Time (Gyr)	R_B (kpc)	ξ_R
Correct	3.95	0.2	0.2
Search Range	[3.0 - 5.0]	[0.1 - 0.5]	[0.1 - 0.5]
Trial 1	3.961	0.196	0.203
Trial 2	3.975	0.193	0.189
Trial 3	3.975	0.205	0.223
Trial 4	3.965	0.208	0.233
Trial 5	3.956	0.199	0.2

Parameters	M_B (sim units)	ξ_M	Likelihood
Correct	12.0	0.2	
Search Range	[1.0 - 50.0]	[0.01 - 0.95]	
Trial 1	12.009	0.204	-1.552
Trial 2	12.06	0.172	-1.591
Trial 3	12.134	0.256	-1.344
Trial 4	12.134	0.275	-1.345
Trial 5	12.042	0.204	-1.296

Our next step is to use a *different* random number seed for the optimization as for the simulated data. These runs were more realistic as we would not know the 'seed' with which the actual data was made. These are shown in Table F.19. As can be seen with this fitted results, the line-of-sight velocity dispersion worked well in fitting the simulated data. It was only when applying the algorithm to actual data that we encountered difficulty.

Table F.19: Tables of best fit values found by the differential evolution search algorithm on MilkyWay@home. Each instance has identical search parameters but are independent of each other. The provided data set was made with a *different* seed from that of the simulations. These were all fit using the χ^2 probability distribution to compare the line-of-sight velocity dispersions. The parameters are the evolution time, the scale length of the baryonic component (R_B), the radius ratio (ξ_R), the baryonic mass (M_B), and the mass ratio (ξ_M).

Parameters	Evolve Time (Gyr)	R_B (kpc)	ξ_R
Correct	3.95	0.2	0.2
Search Range	[3.0 - 5.0]	[0.1 - 0.5]	[0.1 - 0.5]
Trial 1	3.951	0.210	0.181
Trial 2	3.936	0.204	0.182
Trial 3	3.939	0.202	0.183

Parameters	M_B (sim units)	ξ_M	Likelihood
Correct	12.0	0.2	
Search Range	[1.0 - 50.0]	[0.01 - 0.95]	
Trial 1	12.133	0.154	-1.548
Trial 2	12.085	0.142	-1.633
Trial 3	12.025	0.146	-1.348