

USING MILKYWAY@HOME TO MEASURE THE MASS OF THE ORPHAN-CHENAB STREAM PROGENITOR DWARF GALAXY

Eric J. Mendelsohn

Submitted in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:

Dr. Heidi Jo Newberg, Chair

Dr. Ethan Brown

Dr. Gyorgy Korniss

Dr. Malik Magdon-Ismael



Department of Physics, Applied Physics and Astronomy
Rensselaer Polytechnic Institute
Troy, New York

[August 2022]
Submitted July 2022

© Copyright 2022
by
Eric J. Mendelsohn
All Rights Reserved

CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vii
ACKNOWLEDGMENT	x
ABSTRACT	xiii
1. INTRODUCTION	1
1.1 Using Dwarf Galaxies as Gravitational Probes	1
1.2 Overview	3
2. A PRELIMINARY FIT OF THE ORPHAN-CHENAB PROGENITOR	5
2.1 Stellar Data of the OCS	5
2.1.1 SDSS Data	6
2.1.2 DEC Data	7
2.1.3 Distribution of Stream Stars Along Λ_{OCS}	9
2.1.4 Estimation of Total Stellar Mass	10
2.1.4.1 Systematic Error from Using a Redder Globular Cluster	15
2.1.5 Beta Dispersion as a Function of Position Along the Stream	17
2.2 N-Body Simulations	18
2.2.1 Progenitor Creation	19
2.2.2 Accelerations on Each Body	22
2.2.3 Phase Space Evolution Method	22
2.2.4 Likelihood Calculation	23
2.3 Optimization Procedure	25
2.3.1 Differential Evolution Genetic Algorithm	25
2.3.2 Timestep Cut in Parameter Space	26
2.4 Results	27
2.4.1 Using Marginalized Parameter Sweeps to Estimate Error	29
2.4.2 Full Result	34
2.5 Discussion	37
2.5.1 An Extremely Low-Mass Diffuse Progenitor	37
2.5.2 The Location of the Remnant Progenitor	37
2.5.3 The Distribution of Dark Matter Along the Stream	39
2.5.4 The Effect of the LMC on the OCS	41
2.5.5 Sources of Systematic Error	42

3. ADDING THE EFFECTS OF THE LARGE MAGELLANIC CLOUD	44
3.1 Modeling the LMC	44
3.2 Dynamical Friction	45
3.3 Orbit Fitting	46
4. AUTOMATIC DIFFERENTIATION	48
4.1 Derivation of the Sensitivity Matrix	48
4.2 Shortcomings of Numerical Differentiation	50
4.3 Implementation of Automatic Differentiation	52
4.3.1 Testing Basic Automatic Differentiation	54
4.4 Non-Classical Derivative Information	55
4.4.1 Derivatives from Random Distributions	56
4.4.1.1 Testing Derivatives of the Radial Distance	61
4.4.1.2 Testing Derivatives of the Velocity	64
4.4.1.3 Derivatives of Root-Finding Algorithms	68
4.4.2 Derivatives from Binning	72
4.4.3 Derivatives of the Reverse Evolution Time	81
4.4.4 Derivatives from the EMD Calculation	82
5. ANALYSIS OF SYSTEMATIC ERRORS	86
5.1 Complications of a Turbulent Likelihood Surface	86
5.2 Discussion of Results	89
5.3 Improving the Jacobian Measurement	96
5.3.1 Averaging More Jacobians	96
5.3.2 Running with More Bodies and Using Finite Difference	96
6. CONCLUSION	99
6.1 The OCS Fit	99
6.2 Systematic Errors from N-body Automatic Differentiation	100
REFERENCES	101
APPENDICES	105
A. INCOMPLETENESS CORRECTIONS FOR SDSS DATA	105
B. CALCULATING MASS PER TURNOFF STAR USING ISOCHRONES	107
C. ATTEMPTING THE GENERAL EMD DERIVATIVES	111
D. USING LARGE SOFTENING LENGTHS	121

LIST OF TABLES

2.1	The number of F-turnoff stars in the on-field and off-field as a function of Λ . We notice a small gap in the OCS at $\Lambda = 10.5^\circ$, however our optimizations will not be able to properly resolve the gap due to the large error of its associated bin. We split this table in terms of the SDSS data (top) and the DEC data (bottom).	11
2.2	Peak turnoff star color and magnitude range for each of our 11 globular clusters.	13
2.3	Mass per F-Turnoff star using different globular clusters as a reference. The total mass of each globular cluster, with the exception of M13, NGC 5053, and Palomar 5, came from Kimmig et al. (2015). M13's total mass was pulled from Leonard et al. (1992) and Palomar 5's mass came from Odenkirchen et al. (2002).	16
2.4	Fitted values for the Beta dispersion and dispersion error in each Λ range. . . .	20
2.5	The search ranges for our optimizations against the stellar data.	26
2.6	Fitted parameters from MilkyWay@home optimizations. Errors were calculated using Hessian method as described in Section 2.1.4. Note that the primary source of error is from the navigation of the likelihood surface as measured by the differences between individual runs, and not the statistical error calculated from the Hessian.	28
2.7	The likelihood scores calculated for each optimized run. Likelihoods closer to 0 imply a better fit.	28
2.8	Radial profile and mass calculated from fitted parameters. The primary source of error is from the navigation of the likelihood surface as measured by the differences between individual runs, and not the statistical error calculated from the Hessian.	29
2.9	Histograms of stellar density and Beta dispersion σ_B along the stream. Left histogram generated from data. Right histogram shows best simulation fit. The number of counts N in each bin is normalized for the purposes of the Earth Mover Distance calculation in our likelihood score. δ_N and δ_{σ_B} are the errors in the normalized counts and the Beta dispersion respectively.	30
2.10	The current sky position of the progenitor in both Galactic and Equatorial coordinates calculated from each optimization.	38
2.11	The kinetic and gravitational potential energies of the simulated remnant progenitor in each optimization. Note that the kinetic energy is much larger than the gravitational potential energy, indicating that the progenitor is no longer gravitationally bound.	38
5.1	Fitted dwarf parameters determined from MilkyWay@home's differential evolution genetic algorithm. Run 12 is the additional fit found to be consistent with Mendelsohn et al. (2022).	86

5.2	Natural logarithm of the largest value within each run's gradient and Hessian. .	86
5.3	Coefficients of variation of the Jacobian (absolute value). With the exception of $\frac{\partial R_b}{\partial q_z}$, each value is larger than one, the standard deviation of the Jacobian must have a larger magnitude than its average, implying turbulence.	90
5.4	Average Jacobian of the fitted dwarf parameters. We represent the units of length, time, mass, and angle are given in kiloparsecs, gigayears, structure masses ($1SM = 222,288.47M_\odot$), and degrees respectively.	91
5.5	The uncertainties associated with each model parameter. We represent the units of length, time, mass, and angle are given in kiloparsecs, gigayears, structure masses ($1SM = 222,288.47M_\odot$), and degrees respectively. We treat the uncertainty in the LMC mass M_{LMC} as the mass reported in Erkal et al. (2019), $4.5 \times 10^5 SM$	92
5.6	Fractional error contributions from each of the non-fitted model parameters. We also include the fractional errors for the dark matter radius R_d and mass M_d . We also record the total fractional systematic error for each parameter below (totals were added in quadrature). These totals do not include the contributions from the missing LMC as we cannot confidently determine them from this analysis. .	93
5.7	Maximum uncertainty allowed for each model parameter. We list the target precision of each fitted parameter as a percentage next to the parameter name. The red text in the table shows the smallest allowed uncertainty for a model parameter. The smaller the number, the more sensitive the associated fitted parameter is to the associated model parameter.	95

LIST OF FIGURES

2.1	Distribution of stars based on $(g-i)_0$ color. Blue represents stars in the on-field, and green reflects the off-field. The strongest signal can be detected by allowing stars with $0.12 < (g-i)_0 < 0.47$, where we see the largest excess in on-field star counts. We represent the upper and lower bounds of this range with red lines in this plot. Only data from the SDSS is shown in this histogram.	6
2.2	F-turnoff stars from SDSS binned in g_{corr} with $-21^\circ < \Lambda_{OCS} < 21^\circ$, excluding the range $-7^\circ < \Lambda_{OCS} < 10^\circ$ to avoid contamination from the Sagittarius Stream. We see a slight increase in star counts as we approach higher g_{corr} . The rate of increase is slightly higher for the on-field than for the off-field.	7
2.3	Footprint of stellar data from the SDSS and DEC in Galactic longitude (l) and Galactic latitude (b). As can be seen in the DEC data (right), the data is incomplete as much of its off-field (yellow) is unobserved. There are also places where the on-field (blue) is incomplete. Our $(g-i)_0$ and g_0 cuts were used to generate this footprint. The pink line shows the orbit of the OCS as determined from Newberg et al. (2010).	8
2.4	The on-field (red), off-field (green), and excess(blue) of the OCS. Error bars of the excess are shown in the blue brackets.	10
2.5	CMDs of our 11 candidate globular clusters for our OCS color proxy. We plotted this data using TOPCAT (Taylor 2005) and globular cluster data from An et al. (2008). The red lines show the g_0 boundaries of our turnoff range, and the blue line represents the peak $(g-i)_0$ color.	14
2.6	Radial distribution of stars in NGC 5053 that passed our color cuts. The stellar densities (red) in the outer radii of the globular cluster fit relatively well to a Plummer distribution (black). Only data points to the right of the dotted blue line contributed to the fit. We notice a sizeable number of F-turnoff stars missing in the core due to high stellar density in the center.	16
2.7	A plot of the peak turnoff color versus the mass per turnoff star for various globular clusters. Redder globular clusters seem to have a larger mass per F-turnoff star, following an exponential curve. The solid red curve shows the curve of best fit ($\chi^2 = 14.8$) while the dashed red lines are the curves one standard deviation above and below the best fit. The curve has the form $\ln(M/M_\odot) = r(g-i)_0 + \ln(A)$, where $r = 17.7 \pm 4.6$ and $\ln(A) = -3.96 \pm 1.89$	18
2.8	The distribution of stars within each set of Λ bins as a function of B . The open circles represent the number of stars in each B sub-bin, and the black line is the curve of best fit using Equation 2.20. We excluded B sub-bins from the fit if that sub-bin was less than half filled due to the incomplete areal coverage of the DEC data.	19

2.9	Plot of average likelihood score as a function of the number of bodies. The likelihood is a measure of how well an individual simulation matches the comparison simulation; zero likelihood is a perfect reproduction. Each point represents the likelihood score of a simulation against a comparison stream generated using the same dwarf parameters averaged over 12 random seeds. Each run calculates the likelihood using the same comparison stream, which was run using 100,000 bodies. The red dashed line is where the number of bodies equals 40,000. Since this is within the area where the likelihood curve flattens, we choose to run 40,000 bodies in all of our simulations.	21
2.10	Plots illustrating the fraction of our 5D parameter space searched as a function of the baryonic Plummer scale radius (a_B) and the baryonic progenitor mass (M_B). The color of a pixel shows the fraction of the phase space tested for a given (a_B, M_B) . Redder regions in these plots are where a larger percentage of our parameter space is removed. Note that areas where the density of the progenitor galaxy is higher are where we lose most of our phase space. Of the total phase space, 8.92% of our parameter space is cut out.	27
2.11	1-dimensional marginalized parameter sweeps over each dwarf parameter. The blue line shows the negative of the likelihood calculated at each dwarf parameter value while the red line indicates the best fitted value for that parameter (from Run 3). The blue shaded area shows the error in our marginalized likelihood score.	33
2.12	End state histograms showing normalized body counts over Λ_{OCS}	35
2.13	End state histograms showing B dispersion over Λ_{OCS}	36
2.14	Close-up view of simulated progenitor remnant (Run 3). The remnant has been fully disrupted and the progenitor's original structure cannot be easily observed.	39
2.15	Tidal stream produced by Run 3 viewed from different angles. We see that the tidal tails of the stream are thicker near apogalacticon. The green star is the Galactic Center (GC) and the cyan star is the location of the OCS's progenitor dwarf galaxy.	40
2.16	The final state stream (left) compared to the stream while its progenitor is at apogalacticon (right). The thickness of the stream at apogalacticon is comparable in both images regardless of where the progenitor is.	41
2.17	Distribution of baryons and dark matter in Run 3. The tails clearly have a much higher mass-to-light ratio at points that are farther away from the core.	42

5.1	A plot showing how the softening length, which moderates chaotic behavior, affects the gradient and Hessian of our likelihood score. The red line shows the “lnfactor_gradient” of the likelihood score, and the blue line represents the likelihood score’s “lnfactor_hessian”. We observe that smaller softening lengths tend to more extreme derivatives. For each softening length, we averaged each quantity over 6 random seeds to obtain our standard errors. We surprisingly find a maximum at a softening length of 10^{-5} kpc. We know that when the softening length is too small, the system is dominated by strong interactions that cannot be properly resolved in N -body, so the system evaporates. Bodies ejected by such close encounters do not feel any more strong interactions for the rest of the simulation, and thus do not contribute more to the chaotic behavior.	88
5.2	A plot of how the derivative information increases as a function of time. The simulation which generated this plot cam from generating a 500-body 2-component dwarf galaxy and evolving it in a Milky Way potential. The cyan line with red error bars shows the average “lnfactor_gradient” value of each body’s x coordinate, and the yellow line with blue error bars shows the average “lnfactor_hessian” value of each body’s x coordinate. We see that the size of the gradient and Hessian both steadily increase in order of magnitude together almost logarithmically.	89
5.3	A parameter sweep in R_b zoomed into the likelihood peak. The plot on the left shows the parameter sweep run using 40,000 bodies. The plot on the right was run using 80,000 bodies. We see that while running with double the bodies has shifted the optimal radius upwards, we notice that the plot with more bodies has a slightly smoother likelihood surface.	97

ACKNOWLEDGMENT

First and foremost, I would like to thank my research advisor Dr. Heidi Jo Newberg. When I first joined her research group, I had zero computational skills and only vague surface knowledge of astronomy. It is thanks to her patience and guidance that I have been shaped into the researcher I am today. Without her insight and kindness, I would not be writing this dissertation.

I would also like to thank my committee members: Dr. Ethan Brown, Dr. Gyorgy Korniss, and Dr. Malik Magdon-Ismael. Dr. Brown was my first academic advisor and an expert in his field. It is thanks to him that I started walking down the path of dark matter research in my early graduate years. Dr. Korniss was my quantum mechanics professor during my first graduate year, and his classes were the ones I enjoyed the most. Dr. Magdon-Ismael has collaborated on the MilkyWay@home project and numerous occasions, much to the groups appreciation. He is also the one who motivated me to go above and beyond regarding my dissertation topic.

I would like to next thank my beautiful and loving wife Jaime. She always kept an ear open for my physics and coding rants, often leading me to a potential fix with only a few words of encouragement. I am also grateful to her family for offering to let me stay with them after I leave RPI, securing a future foothold and giving me the peace of mind to continue working on my dissertation. I am also grateful to my mother Theresa Mendelsohn for her unconditional love and patience, always keeping a door open for when college life was too stressful. I would also like to thank my younger siblings Alex, Jason, Jacob, and Corinne for offering some much needed distractions and breaks from the toil of debugging.

I also give my thanks Heidi's StarLab group for their help over my graduate life. I'd like to thank Dr. Siddhartha Shelton, MilkyWay@home's previous lead developer, who took the time during his last graduate year to teach me the ins and outs of MilkyWay@home's basic structure. I'd also like to give special thanks to Dr. Jeffery Thompson who has provided much help with MilkyWay@home's servers. I'd also like to thank the cool-headed Thomas Donlon, the most ambitious member of our group, for putting together a user-friendly version of *N*-body (nbody-lite) and a python package to allow future students the ability to more easily parse MilkyWay@home's outputs (mwahpy).

This work was supported by NSF grant AST19-08653; the NASA/NY Space Grant;

contributions made by the Marvin Clan, Babette Josephs, and Manit Limlamai; and the 2015 Crowd Funding Campaign to support Milky Way research. I would also like to thank the 600,000 MilkyWay@home volunteers who donated their computational power to make this research possible.

This dissertation uses 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.

Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the Participating Institutions.

SDSS-IV acknowledges support and resources from the Center for High Performance Computing at the University of Utah. The SDSS website is www.sdss.org.

SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration including the Brazilian Participation Group, the Carnegie Institution for Science, Carnegie Mellon University, Center for Astrophysics — Harvard & Smithsonian, the Chilean Participation Group, the French Participation Group, Instituto de Astrofísica de Canarias, The Johns Hopkins University, Kavli Institute for the Physics and Mathematics of the Universe (IPMU) / University of Tokyo, the Korean Participation

Group, Lawrence Berkeley National Laboratory, Leibniz Institut für Astrophysik Potsdam (AIP), Max-Planck-Institut für Astronomie (MPIA Heidelberg), Max-Planck-Institut für Astrophysik (MPA Garching), Max-Planck-Institut für Extraterrestrische Physik (MPE), National Astronomical Observatories of China, New Mexico State University, New York University, University of Notre Dame, Observatório Nacional / MCTI, The Ohio State University, Pennsylvania State University, Shanghai Astronomical Observatory, United Kingdom Participation Group, Universidad Nacional Autónoma de México, University of Arizona, University of Colorado Boulder, University of Oxford, University of Portsmouth, University of Utah, University of Virginia, University of Washington, University of Wisconsin, Vanderbilt University, and Yale University.

Special thanks must also be given to Dr. Carl Grillmair for giving us his data of the Orphan-Chenab Stream's southern tail, which he got using the Dark Energy Camera (DECam), which was constructed by the Dark Energy Survey (DES) collaboration. Funding for the DES Projects has been provided by the DOE and NSF(USA), MISE (Spain), STFC (UK), HEFCE (UK), NCSA (UIUC), KICP (U. Chicago), CCAPP (Ohio State), MIFPA (Texas A&M), CNPQ, FAPERJ, FINEP (Brazil), MINECO (Spain), DFG (Germany) and the collaborating institutions in the Dark Energy Survey, which are Argonne Lab, UC Santa Cruz, University of Cambridge, CIEMATMadrid, University of Chicago, University College London, DES-Brazil Consortium, University of Edinburgh, ETH Zurich, Fermilab, University of Illinois, ICE (IEECCSIC), IFAE Barcelona, Lawrence Berkeley Lab, LMU Munchen and the associated Excellence Cluster Universe, University of Michigan, NOAO, University of Nottingham, Ohio State University, University of Pennsylvania, University of Portsmouth, SLAC National Lab, Stanford University, University of Sussex, and Texas A&M University.

This research has made use of the NASA/IPAC Infrared Science Archive, which is funded by the National Aeronautics and Space Administration and operated by the California Institute of Technology.

ABSTRACT

We fit the mass and radial profile of the Orphan-Chenab Stream’s (OCS) dwarf galaxy progenitor by using turnoff stars in the Sloan Digital Sky Survey (SDSS) and the Dark Energy Camera (DEC) to constrain N-body simulations of the OCS progenitor falling into the Milky Way on the 1.5 PetaFLOPS MilkyWay@home distributed supercomputer. We infer the internal structure of the OCS’s progenitor under the assumption that it was a spherically symmetric dwarf galaxy comprised of a stellar system embedded in an extended dark matter halo. We optimize the evolution time, the baryonic and dark matter scale radii, and the baryonic and dark matter masses of the progenitor using a differential evolution algorithm. The likelihood score for each set of parameters is determined by comparing the simulated tidal stream to the angular distribution of OCS stars observed in the sky. We fit the total mass of the OCS’s progenitor to $(2.0 \pm 0.3) \times 10^7 M_\odot$ with a mass-to-light ratio of $\gamma = 73.5 \pm 10.6$ and $(1.1 \pm 0.2) \times 10^6 M_\odot$ within 300 pc of its center. Within the progenitor’s half-light radius, we estimate a total mass of $(4.0 \pm 1.0) \times 10^5 M_\odot$. We also fit the current sky position of the progenitor’s remnant to be $(\alpha, \delta) = ((166.0 \pm 0.9)^\circ, (-11.1 \pm 2.5)^\circ)$ and show that it is gravitationally unbound at the present time. The measured progenitor mass is on the low end of previous measurements, and if confirmed lowers the mass range of ultrafaint dwarf galaxies. Our optimization assumes a fixed Milky Way potential, OCS orbit, and radial profile for the progenitor, ignoring the impact of the Large Magellanic Cloud (LMC).

Using second-order forward automatic differentiation, we also attempt to computationally determine the systematic errors introduced from the fixed orbit, gravitational potential, and lack of an LMC. This paper describes the methods employed to implement automatic differentiation in areas of our code where derivative information is not propagated, such as through random number generation and discrete histogram binning. We find that due to the turbulent and chaotic behavior of our searchable likelihood surface, the systematic errors derived through automatic differentiation are severely overestimated. Recommendations for future work to estimate the systematic errors are provided.

CHAPTER 1

INTRODUCTION

1.1 Using Dwarf Galaxies as Gravitational Probes

A few dozen dwarf galaxies are known to orbit around the Milky Way. Over the course of billions of years, these galaxies tidally disrupt and stretch around the Milky Way into tidal streams. The positions and velocities of the stars that make up these streams therefore carry information about the Galaxy’s gravitational field. As such, these dwarf galaxies act as gravitational probes for determining the distribution of gravitating mass in the Milky Way (Ibata et al. 2001; Johnston et al. 2002; Koposov et al. 2010; Newberg et al. 2010; Law & Majewski 2016; Bonaca & Hogg 2018; Ibata et al. 2021). For example, the path of the stream probes the Galactic potential, the transverse motion of the stream probes the time-dependence of the potential, and the width of the stream probes the internal structure of the original progenitor (Willett 2010). This idea of mapping the Galactic potential with stellar streams is not new. However, the focus has mostly been on using streams to constrain the gravitational potential of the Galaxy and hence the structure of its dark halo (Koposov et al. 2010; Bovy et al. 2016; Bonaca & Hogg 2018). However, it is clear that the streams are also affected by the internal structure of their progenitors. Here, we fix the potential and use one well-known halo stream to probe the internal structure of its progenitor.

While examining the Sagittarius Stream, Belokurov et al. (2006) and Grillmair (2006) independently discovered the Orphan Stream, which was so named due to the lack of a visible progenitor. The southern portion of the stream was later named Chenab (Shipp et al. 2018) before it was discovered that both pieces of the stream were generated from the tidal disruption of the same dwarf galaxy. Therefore, we will refer to the stream as the Orphan-Chenab Stream (OCS). Preliminary measurements of the OCS found that it contained old, metal-poor stars possibly from the merger of a satellite galaxy with the Milky Way. Belokurov et al. (2007) placed a lower bound of $10^5 M_\odot$ on the total mass of the OCS’s missing progenitor by studying the interactions of the OCS with the High Velocity

Portions of this chapter previously appeared as: Mendelsohn, E. J., Newberg, H. J., Shelton, S., et al. 2022, *Astrophys. J.*, 926, 106.

Clouds known as Complex A, the dwarf galaxy Ursa Major II (UMa II), and other globular clusters (Segue I, Ruprecht 106, and Palomar 1). However, it was later shown in Sales et al. (2008) that the OCS was kinematically separate from both Complex A and UMa II. In 2010, Newberg et al. (2010) calculated the orbit of the OCS and showed that a progenitor with a total mass of $2.5 \times 10^6 M_\odot$ could be used to fit the tidal stream rather well. This mass is roughly 100 times smaller than most measured Ultra-Faint Dwarf (UFD) galaxies, and other estimates of the OCS’s progenitor suggest a total mass on the order of 10^8 to $10^9 M_\odot$ (Fardal et al. 2019; Hendel et al. 2018).

The disparity in the progenitor mass estimates is interesting because the measured velocity dispersions of stars in ultrafaint galaxies lead many to conclude that dwarf spheroidal galaxies, including ultrafaint dwarf galaxies, have $10^7 M_\odot$ of mass enclosed within the central 300 pc, independent of the dwarf galaxy’s luminosity (Mateo et al. 1993; Gilmore et al. 2007; Strigari et al. 2008). A mass of a few times $10^6 M_\odot$ is less than ultrafaint dwarf galaxies (UFDs) are believed to have. But most measurements of dwarf galaxy masses are derived from velocity dispersions and the assumption that dwarf galaxies are in equilibrium.

While equilibrium is a reasonable assumption for dwarf spheroidal galaxies, it has been suggested that observations should be obtained to look for signs of tidal stretching as a check (Battaglia et al. 2013). UFDs in particular are susceptible to errors in measurement from velocity dispersion due to their small number of bright stars, complications in measuring velocities due to the presence of binary stars, and the possibility that tidal forces could make the assumption of equilibrium invalid. Martin et al. (2008) find that UFDs are elongated, and suggest tidal disruption is the “least problematic” explanation. Objects that are nearly completely disrupted or close to apogalacticon could exhibit velocity dispersions that are systematically an order of magnitude or more higher than equilibrium values due to contamination from extra-tidal stars (Smith et al. 2013; Blańa et al. 2015). Depending on the angle of observation, enforcing dynamical equilibrium on a dwarf spheroidal galaxy undergoing tidal disruption can either overestimate the mass when measuring dispersion along its major axis or underestimate it when measuring on a perpendicular axis (Lokas et al. 2010). As an example, the kinematics of stars in Willman 1 are so far from Gaussian that one cannot even pretend that it is in dynamical equilibrium for the purpose of computing a mass-to-light ratio (Willman et al. 2011). In addition, Triangulum II, previously thought to be the most dark-matter-dominated galaxy known (Kirby et al. 2015), has been downgraded

to possibly “a star cluster or tidally stripped dwarf galaxy” because the originally measured velocity dispersion was calculated including a star that is now known to be in a binary system (Kirby et al. 2017; Buttry et al. 2021). More recently, a new ultra-faint structure named DELVE 2 was discovered whose mass-to-light ratio is sensitive to whether or not the system is undergoing tidal disruption with the Large Magellanic Cloud (LMC), and thus, whether it is classified as a globular cluster or a UFD (Cerny et al. 2020). Clearly, the mass-to-light ratios of UFDs, thought to have the highest mass-to-light ratios of any known objects, are uncertain.

Due to their status as the most dark matter dominated objects in the cosmos, UFDs are popular targets for dark matter indirect detection experiments. The lack of gamma ray signals from the centers of ultrafaint galaxies is being used to place upper limits on the properties of the as yet undetected dark matter particles (Abdallah et al. 2020). Null results from dark matter searches in dwarf galaxies with Fermi LAT data have provided some of the strongest constraints on the dark matter annihilation cross section (Ackermann et al. 2015); more recently, very weak excesses have been found in three ultrafaint dwarf galaxies (Albert et al. 2017), but one is already found to be from a background source (Li et al. 2021). But indirect detection experiments rely heavily on the estimate of the amount of dark matter above background that they are targeting to determine the detection limits. If we find that UFDs are less massive than previously assumed, the constraints these experiments put on dark matter particles would be modified.

1.2 Overview

Using the petaFLOPS-scale MilkyWay@home volunteer supercomputer, Shelton et al. (2021) showed that in a perfect world it would be possible to determine the mass and radial profile of both the stars and the dark matter in a dwarf galaxy progenitor that fell into the Milky Way and was ripped apart into a tidal stream, using only the density distribution of stars in the tidal stream. In this context a perfect world means that the Milky Way potential is known and does not change with time, the orbit of the progenitor in the potential is known, and the dwarf galaxy is known to consist of a Plummer sphere distribution of stars embedded in a Plummer sphere distribution of dark matter, created so that the combination is in stable equilibrium. In this case, only the mass and Plummer radius of the two dwarf galaxy components and the evolution time of the simulation need to be fit.

This method of determining the progenitor satellite’s properties from the tidal stream it produces does not rely on the assumption of dynamical equilibrium. We use MilkyWay@home to generate a large population of simulated dwarf galaxies with varying masses and shapes. Each dwarf galaxy is placed within a static Milky Way potential and evolved for a given amount of time to create a simulated tidal stream. This simulated tidal stream is then compared with the measured distribution of stars in the actual OCS and is assigned a likelihood that a dwarf galaxy with those simulated parameters produced the observed stream. We use differential evolution to evolve the dwarf galaxy parameters until the generated tidal stream closely matches the stellar data.

Our goal is to use MilkyWay@home to measure the mass and shape of the stars and dark matter in the progenitor galaxy that was tidally disrupted to become the OCS. In Chapter 2, we explain how we used stellar data from the Sloan Digital Sky Survey (SDSS) and the Dark Energy Camera (DECam) to fit the progenitor of the OCS using MilkyWay@home. In Chapter 3, we go over some of the new algorithmic changes we implemented to add the Large Magellanic Cloud into the Milky Way gravitational system. Chapter 4 is a review of our implementation of automatic differentiation as a build option for the N-body executable, including the complicated derivations of the equations necessary to propagate derivative information through certain portions of the code where classical derivatives cannot be calculated. Chapter 5 reports the results of our automatic differentiation, and our conclusion is found in Chapter 6.

CHAPTER 2

A PRELIMINARY FIT OF THE ORPHAN-CHENAB PROGENITOR

2.1 Stellar Data of the OCS

To optimize our dwarf progenitor’s parameters, we need accurate stream data to compare against our simulations. To this end, we extract the density of stars along the OCS as well as its width using actual data from the sky. We then parse it into a binned histogram which MilkyWay@home will compare with simulations. We use data from both the Sloan Digital Sky Survey (SDSS; Blanton et al. 2017) and the Dark Energy Camera (DEC; Flaugher et al. 2015) to map the OCS. We use the same Lambda-Beta (Λ, B) coordinate system defined in Newberg et al. (2010) to follow the stream across the sky. Since the OCS does not follow a great circle across the sky and does not maintain the same distance from us as a function of Λ , we apply corrections to the unextincted magnitude in the g-band (g_0) and the B coordinate. The equations for these corrected values are as follows:

$$g_{corr} = g_0 - 0.00022\Lambda^2 + 0.034\Lambda, \quad (2.1)$$

$$B_{corr} = \begin{cases} B + 0.00628\Lambda^2 + 0.42\Lambda + 5.0 & \Lambda \leq -15.0^\circ \\ B & \Lambda > -15.0^\circ \end{cases}, \quad (2.2)$$

as calculated in Newberg et al. (2010). To select F-turnoff stars at the same distance as the OCS, we filter stars such that $20.7 < g_{corr} < 21.7$. Because we removed faint stars (keeping $g_0 < 22.5$) so that we do not need to consider a variable reduction in completeness (Newberg et al. 2002; Weiss et al. 2018), we miss fainter OCS stars at lower Λ_{OCS} , where the stream is further away. We correct for the missing data when generating the binned histogram.

This chapter previously appeared as: Mendelsohn, E. J., Newberg, H. J., Shelton, S., et al. 2022, *Astrophys. J.*, 926, 106.

2.1.1 SDSS Data

From the SDSS DR16 release (Ahumada et al. 2020), we selected all stars with right ascension (α) between 123.75° and 172.0° and declination (δ) between -25.0° and 60.0° . We define our on-field to be the remaining stars within 2 degrees of $B_{corr} = 0^\circ$ and our off-field to be the stars with $2.0^\circ < |B_{corr}| < 4.0^\circ$. The difference between these two fields in a given Λ bin is defined to be the excess in that bin. To avoid overlap between our two data sets, we also apply a Λ cut to the SDSS data, keeping only stars with $\Lambda < 21^\circ$.

We select F-turnoff stars with a $(g - i)_0$ color between 0.12 and 0.47. We select this color range to maximize the signal-to-noise ratio between the on-field and off-field stars (see Figure 2.1). We also filter stars such that $17.0 < g_0 < 22.5$. To remove stars in front of or behind the OCS we enforced a distance cut of $20.7 < g_{corr} < 21.7$, which is the same cut applied in Newberg et al. (2010).

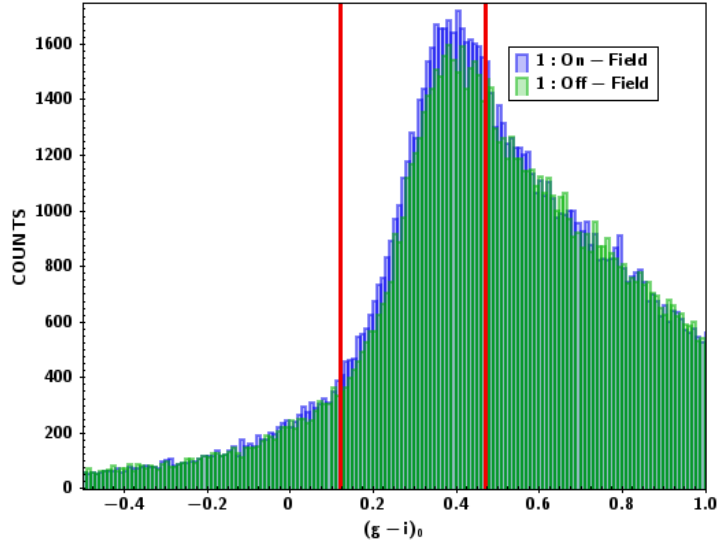


Figure 2.1: Distribution of stars based on $(g - i)_0$ color. Blue represents stars in the on-field, and green reflects the off-field. The strongest signal can be detected by allowing stars with $0.12 < (g - i)_0 < 0.47$, where we see the largest excess in on-field star counts. We represent the upper and lower bounds of this range with red lines in this plot. Only data from the SDSS is shown in this histogram.

The errors in the counts are assumed to follow a Poisson distribution. Thus, the error

of the counts in the i^{th} Λ bin N_i , for both the on-field and the off-field, is simply given by:

$$\sigma_{N_i} = \sqrt{N_i}. \quad (2.3)$$

To account for the incompleteness in our fields at $\Lambda_{OCS} < -21^\circ$, we must calculate what fraction of each Λ bin is filled. Since the incompleteness results from the $g_0 = 22.5$ magnitude limit, we can use Newberg et al. (2010) to exactly determine the boundary $g_b(\Lambda_{OCS})$:

$$g_b(\Lambda_{OCS}) = a_g \Lambda_{OCS}^2 + b_g \Lambda_{OCS} + c_g, \quad (2.4)$$

where $a_g = -0.00022$, $b_g = 0.034$, and $c_g = 22.5$. As can be seen in Figure 2.2, there are more F-turnoff stars at higher magnitudes in both the on-field and the off-field. We therefore correct for the missing stars by fitting a linear model to the star counts as a function of g_{corr} , as described in Appendix A.

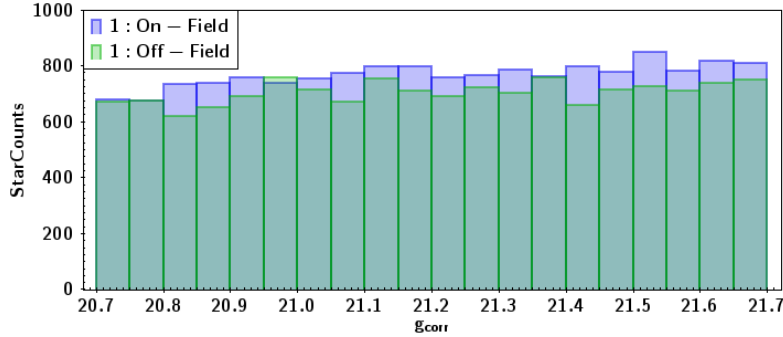


Figure 2.2: F-turnoff stars from SDSS binned in g_{corr} with $-21^\circ < \Lambda_{OCS} < 21^\circ$, excluding the range $-7^\circ < \Lambda_{OCS} < 10^\circ$ to avoid contamination from the Sagittarius Stream. We see a slight increase in star counts as we approach higher g_{corr} . The rate of increase is slightly higher for the on-field than for the off-field.

2.1.2 DEC Data

The DEC data was taken from Grillmair et al. (2015). As the DEC uses the same g-band and i-band filters as the SDSS, we applied the same color and magnitude cuts as in the SDSS data, keeping stars with $21^\circ \leq \Lambda \leq 48^\circ$. We define our on-field and off-field in

the same way as we did for the SDSS data. However, as can be seen in Figure 2.3, the DEC data does not completely fill our on and off-fields.

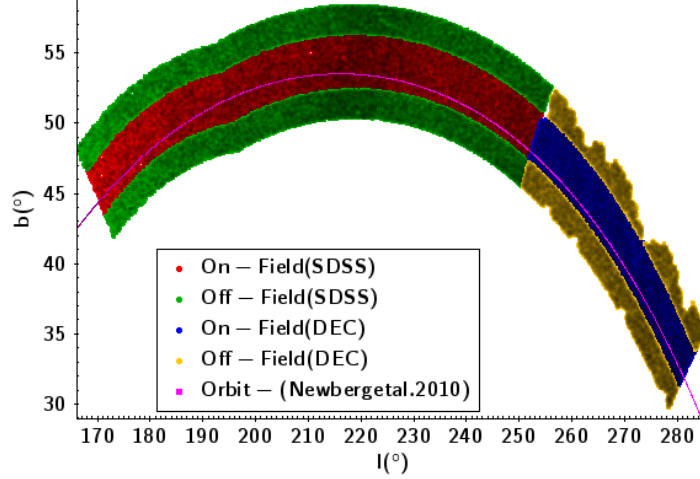


Figure 2.3: Footprint of stellar data from the SDSS and DEC in Galactic longitude (l) and Galactic latitude (b). As can be seen in the DEC data (right), the data is incomplete as much of its off-field (yellow) is unobserved. There are also places where the on-field (blue) is incomplete. Our $(g - i)_0$ and g_0 cuts were used to generate this footprint. The pink line shows the orbit of the OCS as determined from Newberg et al. (2010).

To correct our fields for the lack of data, we apply Monte Carlo approximations to “fill in” the missing patches in the sky. For the i^{th} Λ bin in the on-field, we randomly populate the bin with 8192 (2^{13}) test stars. For each test star, we check whether the test star is within 0.18° of a real star in that bin. We selected 0.18° as our threshold because this was slightly larger than the maximum nearest-neighbor angular distance between two stars in the DEC data. We count the number of test stars within 0.18° of a real star, p_i , and divide it by the total number of test stars, $M = 8192$, to get the “filled” fraction of the i^{th} bin, k_i :

$$k_i = \frac{p_i}{M}. \quad (2.5)$$

Assuming a Poisson distribution for p_i , we derive the following expression for the error in k_i :

$$\sigma_{k_i} = \frac{\sqrt{p_i}}{M}. \quad (2.6)$$

By dividing the number of counts in an on-field bin N_i by the ratio k_i , we can approximate the true number of stars in that bin:

$$N'_i = \frac{N_i}{k_i}. \quad (2.7)$$

This means that the errors in counts in the Λ bin are given by:

$$\sigma_{N'_i} = \frac{1}{k_i} \sqrt{\sigma_{N_i}^2 + \left(\frac{N_i}{k_i}\right)^2 \sigma_{k_i}^2} = \frac{1}{k_i} \sqrt{N_i \left(1 + \frac{N_i}{M k_i}\right)}. \quad (2.8)$$

We apply the same process to each bin in the off-field as well.

2.1.3 Distribution of Stream Stars Along Λ_{OCS}

After combining the stellar data from the SDSS with those from the DEC, we look at each bin and calculate the number of stars within the on-field and off-field. Using the off-field as a background, we subtract it from the on-field and determine the excess within the stream. The excess of the i^{th} bin (E_i) and its error (σ_{E_i}) are calculated using the following formulae:

$$E_i = \begin{cases} N_{\text{on},i} - N_{\text{off},i} & N_{\text{on},i} \geq N_{\text{off},i} \\ 0 & N_{\text{on},i} < N_{\text{off},i} \end{cases} \quad (2.9)$$

$$\sigma_{E_i} = \sqrt{\sigma_{N_{\text{on},i}}^2 + \sigma_{N_{\text{off},i}}^2}. \quad (2.10)$$

The excess and errors of each bin are listed in Table 2.1 along with the number of stars in each on-field and off-field bin. We also present a histogram representing this data in Figure 2.4. Using our cuts, we find an excess of $5,631 \pm 356$ F-turnoff stars in the OCS within the range of $-33^\circ \leq \Lambda \leq 48^\circ$. For the purposes of our likelihood calculation which we will describe later, we also calculate the normalized excess star count (e_i) and its error (σ_{e_i}):

$$e_i = \frac{E_i}{\sum_j E_j} \quad (2.11)$$

$$\sigma_{e_i} = \frac{1}{\sum_j E_j} \sqrt{(1 - 2e_i)\sigma_{E_i}^2 + e_i^2 \sum_j \sigma_{E_j}^2} \quad (2.12)$$

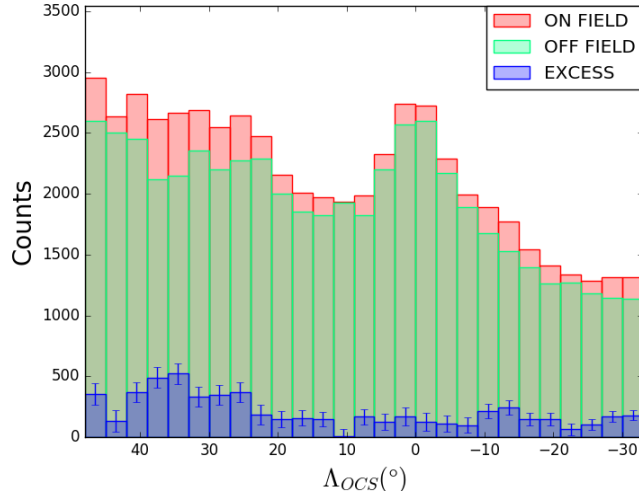


Figure 2.4: The on-field (red), off-field (green), and excess(blue) of the OCS. Error bars of the excess are shown in the blue brackets.

2.1.4 Estimation of Total Stellar Mass

The simulations include bodies that represent both the baryons (stars) and the dark matter. The baryonic simulation bodies represent all of the stellar mass, not just the mass of the turnoff stars. Therefore, to constrain the mass of the OCS, we must relate the number of F-turnoff stars in the sky to an amount of baryonic mass. To accomplish this, we search for a globular cluster whose CMD properties most closely match those of the OCS. Once we find a globular cluster with a turnoff star color that is close enough to that of the OCS, we use that globular cluster to estimate the stellar mass per F-turnoff star in the OCS. Simply multiplying this ratio by the number of F-turnoff stars we detect in the OCS will give us an approximate stellar mass. For our analysis, we look at 11 candidate globular clusters using the globular cluster data from An et al. (2008).

It should be noted that because An et al. (2008) uses the dust map from Schlegel et al. (1998) to calculate their unextincted magnitudes and our stellar data calculates extinctions using Schlafly & Finkbeiner (2011), we need to correct the magnitudes from our globular clusters to adjust for over-reddening. Before we perform our globular cluster analysis, we add the extinction calculated from Schlegel et al. (1998) to each star's magnitude in An et al. (2008) and then subtract the extinction derived from Schlafly & Finkbeiner (2011).

Table 2.1: The number of F-turnoff stars in the on-field and off-field as a function of Λ . We notice a small gap in the OCS at $\Lambda = 10.5^\circ$, however our optimizations will not be able to properly resolve the gap due to the large error of its associated bin. We split this table in terms of the SDSS data (top) and the DEC data (bottom).

Λ Center	On-Field	Off-Field	Excess (E)	Error (σ_E)
-31.5	1314	1136	178	40
-28.5	1315	1144	171	44
-25.5	1283	1178	105	46
-22.5	1335	1271	64	49
-19.5	1412	1262	150	52
-16.5	1542	1395	147	54
-13.5	1774	1530	244	57
-10.5	1891	1676	215	60
-7.5	1990	1893	97	62
-4.5	2286	2173	113	67
-1.5	2722	2596	126	73
1.5	2737	2567	170	73
4.5	2329	2201	128	67
7.5	1989	1822	167	62
10.5	1933	1927	6	62
13.5	1968	1821	147	62
16.5	2006	1850	156	62
19.5	2152	2004	148	64
22.5	2480	2294	186	69
25.5	2646	2269	337	70
28.5	2546	2189	357	69
31.5	2686	2356	330	71
34.5	2668	2134	534	69
37.5	2611	2163	448	69
40.5	2817	2445	372	73
43.5	2637	2491	146	72
46.5	2953	2597	356	74

When selecting globular cluster stars for our F-turnoff color calculation, we must search over a defined magnitude range for each cluster. To select this range, we plot a color-magnitude diagram (CMD) in $(g - i)_0$ and g_0 . We set the minimum (brightest) turnoff magnitude at the intersection of the subgiants with the Red Giant Branch (RGB). We record the $(g - i)_0$ color of the intersection and determine the brightest magnitude for Main Sequence (MS) stars of that color, defining that g_0 as the maximum turnoff magnitude. The ranges are listed Table 2.2, and the CMDs of each candidate can be seen in Figure 2.5.

Using these stars, we generate two one-dimensional histograms with bins in $(g - r)_0$ and $(g - i)_0$ color, respectively. We then fit the histogram to a skew normal distribution with a linear background. The model has the form:

$$f(x) = mx + b + Ae^{-\frac{(x-\xi)^2}{2\omega^2}} \left(1 + \operatorname{erf} \left(\gamma \left(\frac{x-\xi}{\omega\sqrt{2}} \right) \right) \right), \quad (2.13)$$

where m is the linear slope, b is the linear y-intercept, A is the amplitude, ξ is the “unskewed” mean, ω is the “unskewed” standard deviation, and γ is the skew parameter. In our fits, we assume that $\gamma \geq 0$ because we expect the number of stars to drop off more slowly towards the redder end of the distribution. We fit these six parameters using a χ^2 best-fit method and a differential evolution algorithm. We run the algorithm 10 independent times and keep the best fit to ensure our χ^2 fit does not fall into a local minimum. After fitting the six parameters, we calculate the Hessian of the natural logarithm of the χ^2 surface around that minimum. By inverting the Hessian, we derive the covariance matrix of the fitted parameters:

$$(H \ln \chi^2)^{-1} = \begin{pmatrix} \frac{\partial^2 \ln \chi^2}{\partial x_1^2} & \frac{\partial^2 \ln \chi^2}{\partial x_1 \partial x_2} & \dots \\ \frac{\partial^2 \ln \chi^2}{\partial x_2 \partial x_1} & \frac{\partial^2 \ln \chi^2}{\partial x_2^2} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}^{-1} = \begin{pmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2}^2 & \dots \\ \sigma_{x_2 x_1}^2 & \sigma_{x_2}^2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \quad (2.14)$$

Each second derivative is calculated numerically using a finite step size, where each i^{th} parameter is assigned a step size h_i . We approximate each second derivative using the following formula:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \simeq \begin{cases} \frac{f(x_i - h_i, x_j - h_j) - f(x_i - h_i, x_j + h_j) - f(x_i + h_i, x_j - h_j) + f(x_i + h_i, x_j + h_j)}{4h_i h_j} & i \neq j \\ \frac{f(x_i - h_i) - 2f(x_i) + f(x_i + h_i)}{h_i^2} & i = j. \end{cases} \quad (2.15)$$

After calculating the errors of each parameter, we take these errors and set them as the new step size for each of their respective parameters. We then calculate the Hessian again, taking those errors and feeding them back into the Hessian calculation over and over until the errors converge within a significantly small enough threshold (0.0001).

Taking our six fitted parameters and their errors, we calculate the mode of the fitted distribution. While skewed normal distributions do not have an analytical solution for the mode (Mo), there does exist a numerical approximation for it while $\gamma \geq 0$:

$$Mo \simeq \xi + \omega \left(\zeta \sqrt{\frac{2}{\pi}} - \left(\frac{4 - \pi}{4} \right) \frac{\left(\zeta \sqrt{\frac{2}{\pi}} \right)^3}{1 - \frac{2}{\pi} \zeta^2} - \frac{1}{2} e^{-\frac{2\pi}{\gamma}} \right), \quad (2.16)$$

where

$$\zeta = \frac{\gamma}{\sqrt{1 + \gamma^2}}. \quad (2.17)$$

Running our SDSS OCS data through this pipeline, we find that the excess of the OCS has a peak $(g - r)_0$ color of 0.214 ± 0.066 and a $(g - i)_0$ color of 0.286 ± 0.083 . We similarly run our 11 candidate globular clusters through our algorithm to determine which has the closest color peak. Table 2.2 lists the color peaks for each of these globular clusters.

Table 2.2: Peak turnoff star color and magnitude range for each of our 11 globular clusters.

Globular Cluster	g_0 F-Turnoff Range	$(g - r)_0$	$(g - i)_0$
M2	19.0 - 21.0	0.351 ± 0.015	0.457 ± 0.028
M3	18.5 - 20.5	0.309 ± 0.022	0.382 ± 0.047
M5	18.0 - 20.0	0.326 ± 0.024	0.456 ± 0.051
M13	18.0 - 20.0	0.318 ± 0.035	0.425 ± 0.056
M15	18.75 - 20.75	0.338 ± 0.020	0.483 ± 0.033
M53	19.75 - 21.5	0.264 ± 0.034	0.351 ± 0.042
M92	18.0 - 20.0	0.262 ± 0.027	0.359 ± 0.043
NGC 4147	20.0 - 21.5	0.267 ± 0.038	0.353 ± 0.050
NGC 5053	19.5 - 21.5	0.241 ± 0.025	0.326 ± 0.041
NGC 5466	19.25 - 21.25	0.283 ± 0.024	0.351 ± 0.029
Palomar 5	20.5 - 22.25	0.361 ± 0.040	0.505 ± 0.053

The globular cluster whose peak F-turnoff star color is most similar to that of the OCS is NGC 5053.

After determining which globular cluster to use, we count the total number of F-turnoff stars we see in NGC 5053 from the photometric data provided by An et al. (2008), applying the same color cuts we employed in our stream separation using the turnoff range measured for NGC 5053, selecting stars with $0.12 < (g - i)_0 < 0.47$ and $19.5 < g_0 < 21.5$. However,

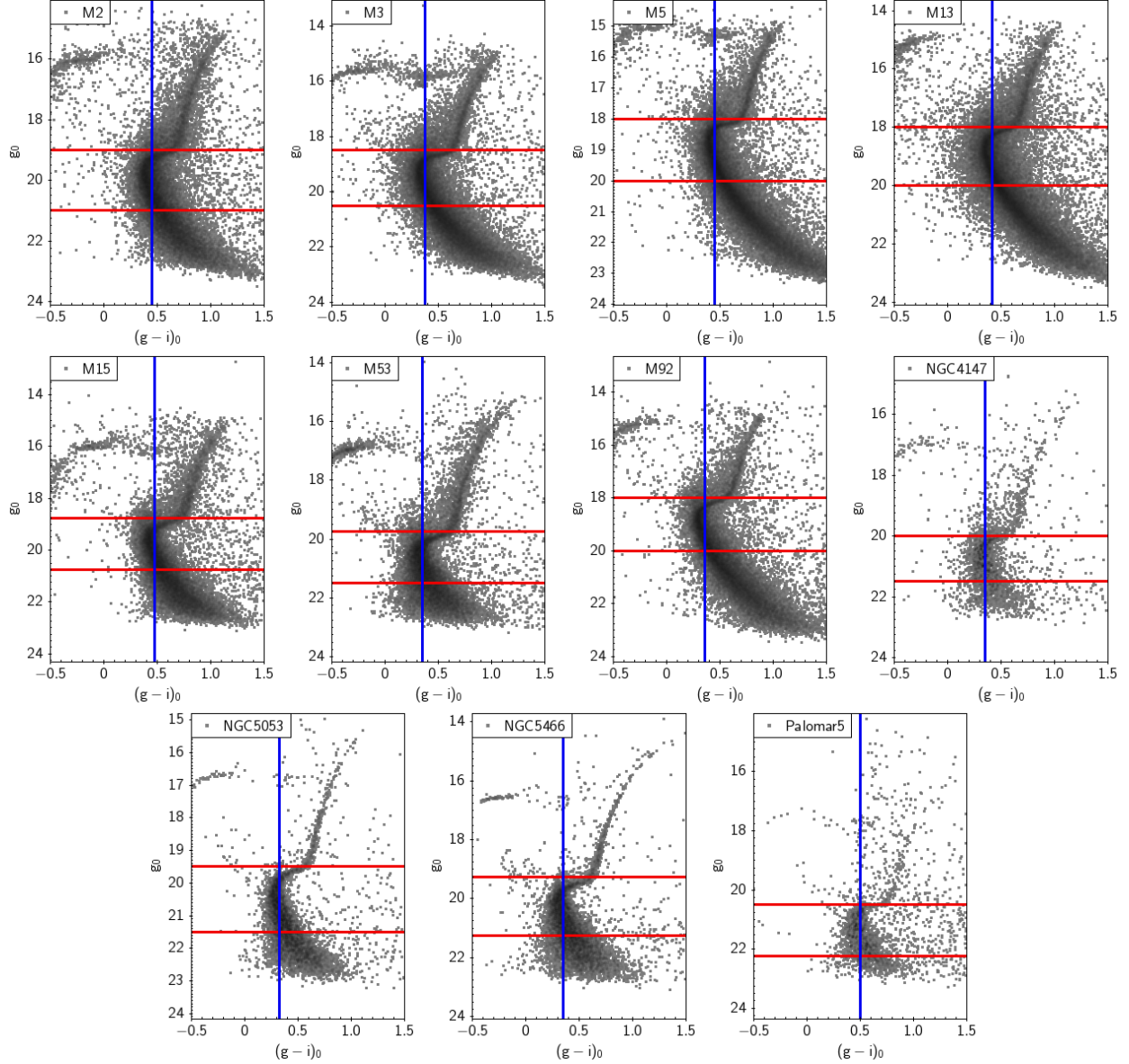


Figure 2.5: CMDs of our 11 candidate globular clusters for our OCS color proxy. We plotted this data using TOPCAT (Taylor 2005) and globular cluster data from An et al. (2008). The red lines show the g_0 boundaries of our turnoff range, and the blue line represents the peak $(g-i)_0$ color.

since globular clusters have a high stellar density in the sky, it is difficult to resolve all of the stars within the center of the cluster. Although An et al. (2008) uses a DAOPHOT pipeline that is specifically designed for crowded-field stellar photometry (Stetson 1987) to parse the SDSS data, we find that there are still stars which are not resolved. To correct for the missing stars, we fit the radial distribution of NGC 5053 to that of a Plummer sphere.

The 2D surface number density of a Plummer sphere is given by:

$$\Sigma(R) = \frac{N_{total}a^2}{\pi(R^2 + a^2)^2}, \quad (2.18)$$

where the number of stars ΔN within ΔR of radius R from the center of the cluster is represented with the formula:

$$\Delta N = 2\pi R \Sigma(R) \Delta R. \quad (2.19)$$

We fit the total number of F-turnoff stars (N_{total}) and the angular scale radius (a) to NGC 5053 using the method of least squares, setting the radius R for a given star as the angular distance between that star and the center of the cluster, which we find to be at $(\alpha, \delta) = (199.107^\circ, 17.6927^\circ)$. To ensure our fit is not heavily impacted by the overcrowding near the center of the globular cluster, we exclude all stars within the central core by requiring $R > \theta_{1/2} \sim 0.035^\circ$, where $\theta_{1/2}$ is the angular distance from the globular cluster's center to where the stellar density drops to half of its maximum value. Figure 2.6 shows the radial distribution of F-turnoff stars in NGC 5053 and the fitted Plummer curve.

Correcting for missing stars near the center of the cluster, we find a total of $3,932 \pm 110$ F-turnoff stars within NGC 5053. Taking the mass of NGC 5053 to be $(5.37 \pm 1.32) \times 10^4 M_\odot$ (Baumgardt 2017), we calculate that each turnoff star represents a stellar mass of roughly $(13.7 \pm 3.4) M_\odot$. A calculation of the mass per turnoff star using isochrones, which finds a similar value, is done in Appendix B. Multiplying this number by the $5,631 \pm 356$ F-turnoff stars we calculate in the OCS gives us a total baryonic mass of $(7.7 \pm 2.0) \times 10^4 M_\odot$ within the range $-33.0^\circ < \Lambda < 48.0^\circ$.

2.1.4.1 Systematic Error from Using a Redder Globular Cluster

While the turnoff color of NGC 5053 matches that of the OCS within errors, the globular cluster does have a somewhat redder measure of turnoff color. To illustrate how a redder color would impact our estimate of the mass per turnoff star, we recalculate this quantity using all of our other globular clusters as the reference. Table 2.3 below shows the mass per F-turnoff star we find if we use each of these redder globular clusters as a reference.

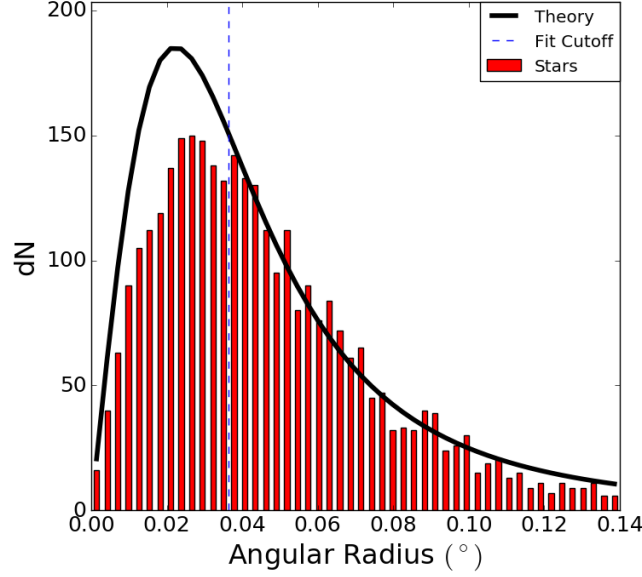


Figure 2.6: Radial distribution of stars in NGC 5053 that passed our color cuts. The stellar densities (red) in the outer radii of the globular cluster fit relatively well to a Plummer distribution (black). Only data points to the right of the dotted blue line contributed to the fit. We notice a sizeable number of F-turnoff stars missing in the core due to high stellar density in the center.

Table 2.3: Mass per F-Turnoff star using different globular clusters as a reference. The total mass of each globular cluster, with the exception of M13, NGC 5053, and Palomar 5, came from Kimmig et al. (2015). M13’s total mass was pulled from Leonard et al. (1992) and Palomar 5’s mass came from Odenkirchen et al. (2002).

GC	Mass (M_{\odot})	F Stars	Mass/Star (M_{\odot})
M2	$(5.75 \pm 2.65) \times 10^5$	$16,701 \pm 1,687$	34.4 ± 15.9
M3	$(4.68 \pm 1.40) \times 10^5$	$15,808 \pm 261$	29.6 ± 8.9
M5	$(3.89 \pm 1.16) \times 10^5$	$12,061 \pm 376$	32.3 ± 9.7
M13	$(6.05 \pm 0.33) \times 10^5$	$17,464 \pm 554$	34.6 ± 2.2
M15	$(5.13 \pm 1.06) \times 10^5$	$6,543 \pm 1,843$	78.4 ± 27.4
M53	$(3.48 \pm 1.04) \times 10^5$	$10,913 \pm 328$	31.9 ± 9.6
M92	$(2.75 \pm 0.57) \times 10^5$	$11,382 \pm 1,107$	24.2 ± 5.5
NGC 4147	$(3.72 \pm 2.82) \times 10^4$	$1,564 \pm 87$	23.8 ± 18.1
NGC 5053	$(5.37 \pm 1.32) \times 10^4$	$3,932 \pm 110$	13.7 ± 3.4
NGC 5466	$(3.80 \pm 2.45) \times 10^4$	$4,425 \pm 126$	8.6 ± 5.5
Palomar 5	$(5.2 \pm 0.7) \times 10^3$	528 ± 37	9.8 ± 1.5

Plotting the mass per turnoff star as a function of the peak color (Figure 2.7), we see that redder globular clusters typically produce a larger calculated mass per turnoff star. Fitting these points to an exponential curve of the form $\ln(M/M_\odot) = r(g-i)_0 + \ln(A)$, we find $r = 17.7 \pm 4.6$ and $\ln(A) = -3.96 \pm 1.89$ ($\chi^2 = 14.8$). This implies that using a globular cluster for the mass estimate of a stellar system with a bluer peak color overestimates its mass. Because we select a fixed (blue) color range to select our turnoff stars ($0.12 < (g-i)_0 < 0.47$), it makes sense that using a redder globular cluster would result in fewer stars falling within this color range, and thus give us a larger stellar mass per turnoff star. If we extrapolate the curve to find the mass per turnoff star using a globular cluster with a color similar to the OCS, we find $3.0 \pm 6.9 M_\odot$ per turnoff star, roughly one quarter of the mass per turnoff star estimated using NGC 5053. It should be noted, however, that we do not use this extrapolated mass per turnoff for analyzing the OCS due to the extrapolation’s poor fit and large errors. We instead use the mass per turnoff derived from NGC5053, keeping in mind that any masses we fit using this data are overestimates.

2.1.5 Beta Dispersion as a Function of Position Along the Stream

To fit the apparent width of the stream as a function of Λ , we measure the Beta dispersion of the OCS by splitting each Λ bin into 20 B bins, covering both the on and off fields. In each Λ bin, we fit the distribution of B values to a Gaussian distribution with a linear background, taking the fitted σ value as the Beta dispersion:

$$f(x) = mx + b + Ae^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.20)$$

where m is the slope of the background, b is the y-intercept, A is the amplitude, μ is the mean, and σ is the standard deviation of the Gaussian distribution.

To ensure a larger sample size, we group each consecutive set of three Λ bins into a single bin for the purposes of the Beta dispersion calculation. We employ the same Monte Carlo method of bin completion used in Section 2.1.2 to determine the number of stars in each Beta bin. However, we exclude any bins whose data coverage is less than half of the bin area from our fit. We use the “curve fit” algorithm from SciPy (Virtanen et al. 2020) to

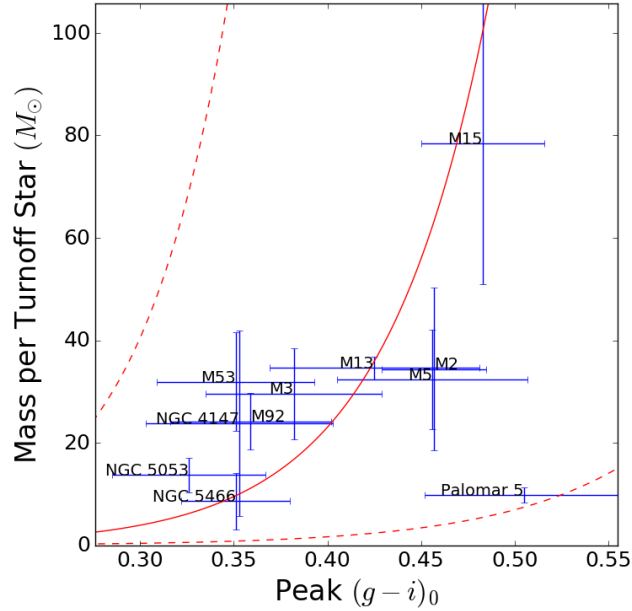


Figure 2.7: A plot of the peak turnoff color versus the mass per turnoff star for various globular clusters. Redder globular clusters seem to have a larger mass per F-turnoff star, following an exponential curve. The solid red curve shows the curve of best fit ($\chi^2 = 14.8$) while the dashed red lines are the curves one standard deviation above and below the best fit. The curve has the form $\ln(M/M_\odot) = r(g - i)_0 + \ln(A)$, where $r = 17.7 \pm 4.6$ and $\ln(A) = -3.96 \pm 1.89$.

fit the beta dispersions and their respective errors. Figure 2.8 shows the fit in each set of Λ bins while Table 2.4 lists the numerical values of the Beta dispersion and their errors.

2.2 N-Body Simulations

For a given set of dwarf galaxy parameters, we calculate the likelihood that those parameters produce the observed data histogram, as constructed from the values calculated in Section 2.1. This section summarizes the methods and algorithms that are laid out in more detail in Shelton et al. (2021).

To calculate the likelihood of a set of progenitor dwarf galaxy parameters, we place the prescribed simulated dwarf galaxy in a static Milky Way potential and let it evolve through

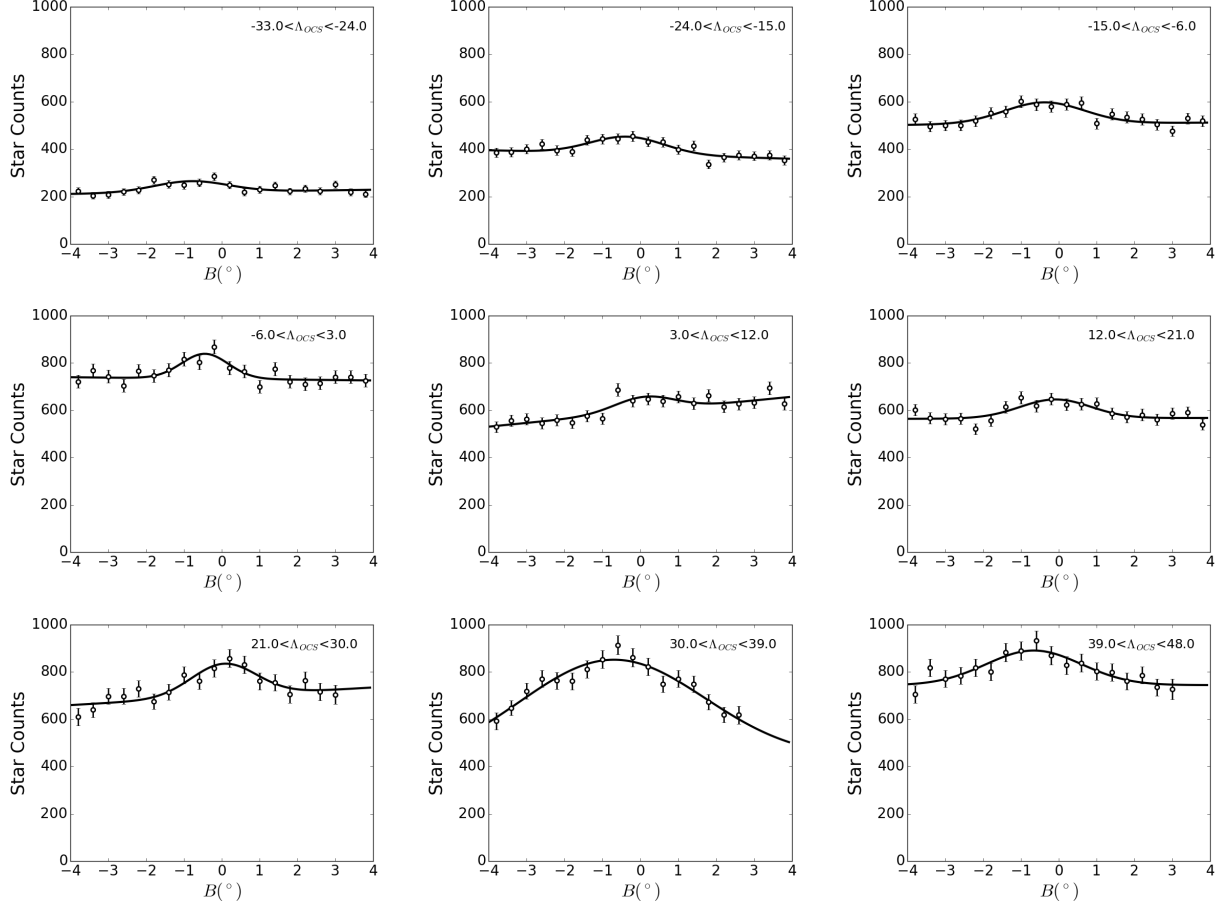


Figure 2.8: The distribution of stars within each set of Λ bins as a function of B . The open circles represent the number of stars in each B sub-bin, and the black line is the curve of best fit using Equation 2.20. We excluded B sub-bins from the fit if that sub-bin was less than half filled due to the incomplete areal coverage of the DEC data.

time, interacting with the gravitational well and its own bodies. We employ multi-threading in our N-body algorithm to speed up our calculations. After running the simulation for a prescribed number of timesteps, we calculate the likelihood that the simulated and observed streams result from the same progenitor.

2.2.1 Progenitor Creation

We represent our simulated dwarf galaxy as a collection of 40,000 bodies; half of the bodies represent visible baryonic matter, and the other half represent dark matter. Regardless of the baryonic and dark matter masses, we employ the same number of bodies, changing

Table 2.4: Fitted values for the Beta dispersion and dispersion error in each Λ range.

Λ Range	Beta Dispersion	Dispersion Error
$[-33.0, -24.0]$	1.003	0.323
$[-24.0, -15.0]$	1.001	0.208
$[-15.0, -6.0]$	1.076	0.218
$[-6.0, 3.0]$	0.612	0.146
$[3.0, 12.0]$	0.791	0.308
$[12.0, 21.0]$	0.947	0.257
$[21.0, 30.0]$	0.886	0.193
$[30.0, 39.0]$	2.263	0.825
$[39.0, 48.0]$	1.230	0.275

the mass per particle to represent the full mass of a component. We choose 40,000 bodies because it is the minimum number of particles needed to consistently generate the same stream over different random seeds (see Figure 2.9).

To model our progenitor dwarf galaxy, we arrange our 40,000 bodies into two nested, concentric Plummer spheres (Plummer 1911). The radial profile of a Plummer sphere is given by:

$$\rho(r) = \frac{3M}{4\pi a^3} \left(1 + \frac{r^2}{a^2}\right)^{-5/2}, \quad (2.21)$$

where a is a scale radius, and M is the total mass of the Plummer sphere. Each progenitor can thus be described by four parameters, the baryonic mass (M_B), the dark matter mass (M_D), the baryon scale radius (a_B), and the dark matter scale radius (a_D). However, since the properties of our dwarf galaxies have a degree of scale invariance, it is more efficient to optimize over the total baryonic component (M_B and a_B) and a two ratios (ξ_R and ξ_M). These ratios are defined as follows:

$$\xi_R = \frac{a_B}{a_B + a_D}, \quad (2.22)$$

$$\xi_M = \frac{M_B}{M_B + M_D}. \quad (2.23)$$

The method we use to randomly generate these progenitor dwarf galaxies is detailed in

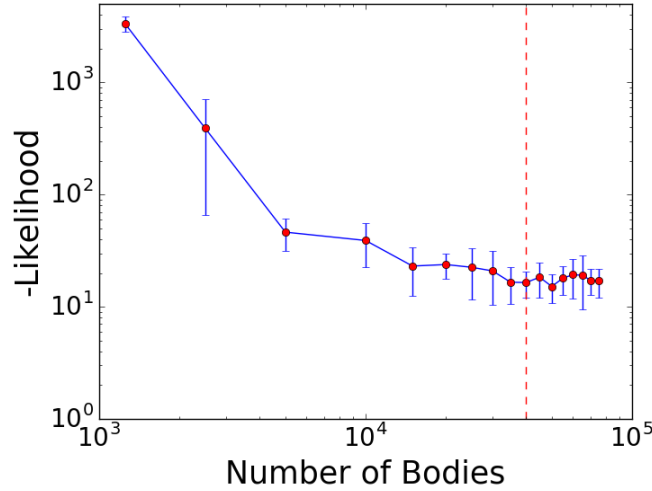


Figure 2.9: Plot of average likelihood score as a function of the number of bodies. The likelihood is a measure of how well an individual simulation matches the comparison simulation; zero likelihood is a perfect reproduction. Each point represents the likelihood score of a simulation against a comparison stream generated using the same dwarf parameters averaged over 12 random seeds. Each run calculates the likelihood using the same comparison stream, which was run using 100,000 bodies. The red dashed line is where the number of bodies equals 40,000. Since this is within the area where the likelihood curve flattens, we choose to run 40,000 bodies in all of our simulations.

Shelton et al. (2021).

We determine the starting location of the progenitor by first specifying its present-day position and velocity, which we calculated using an orbit for the OCS determined from Newberg et al. (2010). Using its present-day position and velocity, we integrate a single-body orbit backwards in time up to the evolution time τ_{evolve} to find the progenitor’s starting position and velocity, centering the 40,000 body simulated progenitor at that point. Although the orbit remains fixed across simulations, we determined it from different information than we will use to measure the dwarf galaxy parameters; the orbit was fit using sky position, line-of-sight velocity, and heliocentric distance.

Our progenitor dwarf galaxy is thus completely characterized with five parameters: the evolution time τ_{evolve} , the baryonic Plummer radius a_B , the radius ratio ξ_R , the baryonic Plummer mass M_B , and the baryonic mass ratio ξ_M . As such, these are the parameters we wish to optimize over using MilkyWay@home.

2.2.2 Accelerations on Each Body

Our Milky Way potential consists of 3 components: a Miyamoto-Nagai disk, a Hernquist bulge, and a logarithmic halo (Miyamoto & Nagai 1975; Hernquist 1990; Sakamoto et al. 2003). Our Spherical Hernquist bulge has a scale radius of 0.7 kpc and a mass of $= 9.9 \times 10^{10} M_{\odot}$ (Law et al. 2005). The Miyamoto-Nagai disk in our simulations has a scale radius of 6.5 kpc, a scale height of 0.26 kpc, and a mass of $3.4 \times 10^{10} M_{\odot}$ (Law et al. 2005). We implement a logarithmic halo with a circular velocity of 73 km s^{-1} ($74.61 \text{ kpc Gyr}^{-1}$) (Newberg et al. 2010) and a scale radius of 12 kpc (Law et al. 2005) to model the dark matter halo. We translate these potentials into accelerations by calculating the negative gradient of each potential ($\mathbf{a} = -\nabla\phi$). Within our simulations, we represent distances in kiloparsecs (kpc), time in gigayears (Gyr), and mass in Structure Masses (SM), where $1 \text{ SM} = 222,288.47 M_{\odot}$. Using these units, $G = 1$.

Our algorithm uses classical Newtonian gravitation to calculate the acceleration between two bodies. We employ a Barnes-Hut tree algorithm (Barnes & Hut 1986) which has a complexity of $O(N \log N)$ to speed up calculations. Our N-body simulations employ a Velocity Verlet method (Verlet 1967) to calculate the new positions and velocities of each particle at the end of each timestep. We use this Velocity Verlet algorithm because it is a symplectic integrator, meaning the area of its phase space is conserved as the system evolves through time. This property is especially attractive for our integrator since it implies that energy is conserved as the dwarf galaxy falls into the Milky Way, an important constraint that must be maintained throughout our N-body simulations. The calculation of our accelerations, timestep, and softening length is further detailed in Shelton et al. (2021).

2.2.3 Phase Space Evolution Method

After establishing the Milky Way potential and creating the progenitor dwarf, we determine the starting point of our simulation by defining approximately where we expect the progenitor’s core to lie in the stream at the present time, and thus where we would like the simulated progenitor core to end up at the end of the simulation. For the OCS, we place this starting point at $(l, b, R) = (258.0^{\circ}, 45.8^{\circ}, 21.5 \text{ kpc})$ with a velocity $(v_x, v_y, v_z) = (-185.5, 54.7, 147.4) \text{ kpc Gyr}^{-1}$. This velocity is given in right-handed Galactic Cartesian coordinates, in which the X-axis is the direction from the Sun to the Galactic Center and the Y-axis is in the direction of the Sun’s motion. This starting point defines the orbit of the

OCS and was calculated using the orbit fit from Newberg et al. (2010). We place a single body at the starting point and evolve that body backwards in time by our evolution time, τ_{evolve} . We replace the body with the simulated dwarf galaxy progenitor, and the N-body integration begins.

2.2.4 Likelihood Calculation

After running a simulation to completion, we need to translate the resulting tidal stream into a histogram to compare with the data histogram. We transform our bodies into a Lambda-Beta (Λ, B) system that follows the OCS along its $B = 0$ great circle. We use the same coordinate system as derived in Newberg et al. (2010). We split the stream into a histogram that consists of 28 Λ bins subtending the range $(-33^\circ, 48^\circ)$ and one Beta bin covering the range $(-15^\circ, 15^\circ)$. Within each bin, we record the number of bodies and normalize the counts with respect to the total number of bodies within the histogram’s range. We also calculate the Beta dispersion at each Λ where we have a measured stream width.

To compare the similarity between two histograms, we employ a metric called the likelihood score. This value represents the natural logarithm of the probability that two histograms match. As such, the likelihood score ranges from negative infinity (worse case set to -9999999.9) to 0, where a higher likelihood is indicative of a closer match. Our likelihood score consists of 3 components: the Earth Mover Distance (EMD) Component, the Cost Component, and the Beta Dispersion Component.

The EMD Component (Rubner et al. 2000) measures how well the shapes of two normalized histograms match each other. Given two histograms, one representing the number of stream stars as a function of Λ and the other representing the number of simulation bodies as a function of Λ , the code calculates the minimum amount of “work” necessary to deform one histogram into the other and translates that work into an EMD score. If the EMD score is greater than some maximum EMD score we define ($EMD_{max} = 50$), the code outputs the worst case likelihood score. Otherwise, we calculate the EMD component to be:

$$\ln(\mathcal{L}_{EMD}) = 300 \ln \left(1 - \frac{EMD}{EMD_{max}} \right) \quad (2.24)$$

The Cost Component compares the total mass within each histogram and adds a penalty commensurate to the mass difference. We utilize this component because the EMD score can only be calculated if both histograms are normalized, so the information about

their mass is lost. The Cost Component is given by:

$$\ln(\mathcal{L}_{Cost}) = -\frac{(M_{sim}N_{sim} - M_{data}N_{data})^2}{2\left(M_{data}^2N_{data} + M_{sim}^2N_{sim}\left(\frac{N_{sim}}{N_{total}}\right)\left(1 - \frac{N_{sim}}{N_{total}}\right)\right)} \quad (2.25)$$

where M_{sim} is the mass per body in the simulation, M_{data} is the mass per turnoff star we calculated in Section 2.1.4, N_{sim} is the number of baryonic bodies within the histogram's range, N_{data} is the number of turnoff stars within the histogram's range (5,631 stars), and N_{total} is the total number of baryonic bodies we ran in the simulation (20,000).

The Beta Dispersion Component compares the width of the streams as a function of Λ . For each Λ bin in the simulation, we calculate the Beta dispersion $\sigma_{\beta,sim,i}$ and its error $\delta_{\sigma_{\beta,sim,i}}$, and then compare them to the values determined from the stellar data, $\sigma_{\beta,data,i}$ and $\delta_{\sigma_{\beta,data,i}}$. The formula for the Beta Dispersion Component is given by:

$$\ln(\mathcal{L}_{Disp}) = -\frac{1}{2} \sum_i^{N_{bins}} \frac{(\sigma_{\beta,sim,i} - \sigma_{\beta,data,i})^2}{\delta_{\sigma_{\beta,sim,i}}^2 + \delta_{\sigma_{\beta,data,i}}^2}. \quad (2.26)$$

The total likelihood score is sum of each of these components:

$$\ln(\mathcal{L}) = \ln(\mathcal{L}_{EMD}) + \ln(\mathcal{L}_{Cost}) + \ln(\mathcal{L}_{Disp}) \quad (2.27)$$

The details of how each component is calculated can be found in Shelton et al. (2021). Only the locations of the baryonic bodies contribute to the calculation of the likelihood score, because we are only able to observe the stars in the tidal stream.

Due to the chaotic behavior of N-body systems, the slightest deviation in the progenitor parameters (or the random seed) can shift the core of our progenitor further ahead or behind in the stream, drastically affecting the likelihood. To mitigate this chaos, we check the likelihood score over a range of evolution times to “match” the cores of the simulated stream to that of the data. Once the simulation reaches 98% of the evolution time, the code calculates the likelihood at each timestep and keeps the snapshot with the best likelihood. This comparison continues until the simulation time reaches 102% of the evolution time. The best likelihood in this range of forward evolution times is returned as the likelihood for the given parameter set.

2.3 Optimization Procedure

Exploring our parameter space requires hundreds of thousands of likelihood calculations, each requiring an N-body simulation. Our N-body simulations on average take 15 minutes to compute on a single processor; it would take several years to run this optimization on an average laptop. Therefore, computing these optimized parameters within a reasonable time-frame demands the use of a much more sophisticated computer network.

MilkyWay@home is a collection of roughly 26,000 volunteered computers connected by the Berkeley Open Infrastructure for Network Computing (BOINC), operating at 1.5 PetaFLOPS of combined computing power. MilkyWay@home uses this massive computing power in conjunction with the Time Asynchronous Optimization (TAO; Desell et al. 2009) package to optimize the parameters which define the size and shape of the progenitor dwarf galaxy. This section explains the differential evolution genetic algorithm we use to optimize our progenitor parameters and describes the parameter space we explore.

2.3.1 Differential Evolution Genetic Algorithm

MilkyWay@home starts an optimization by first generating a random population of 50 parameter sets, each containing the five parameters described in Section 2.2.1, on the main server. For each parameter set in the population, MilkyWay@home sends out a package containing an N-body executable, a Lua file containing important settings for the simulation, the aforementioned parameter set, and a comparison histogram containing the real stellar data to several of the 26,000 volunteered computers. The volunteered computer runs the N-body executable with the chosen parameters and generates a simulated tidal stream, calculates the density of bodies across the stream, and logs the information into a histogram. The newly generated histogram is compared to the data histogram, and the computer generates a likelihood score based on how well the two histograms match. This score and its associated parameter set are sent back to the main server.

After calculating the likelihood scores of the initial population, we employ a differential evolution algorithm to generate new parameter sets to test. For each member of our population \mathbf{x} , we select three other random distinct members (\mathbf{a} , \mathbf{b} , and \mathbf{c}) to act as “genetic donors” to the new parameter set \mathbf{y} . We then generate a random integer p between 1 and n , where n is the number of elements in our parameter set (in this case 5). Also, we generate a random number r_i between 0 and 1 for each element x_i . Given these numbers, the i^{th}

element y_i of the new parameter set is thus defined as:

$$y_i = \begin{cases} a_i + F(b_i - c_i) & r_i < CR \vee i = p \\ x_i & otherwise \end{cases} . \quad (2.28)$$

where CR is the crossover rate (set to 0.9) and F is the differential weight (set to 0.8). These newly created sets are then sent to our volunteers to calculate a likelihood score. Once a parameter set returns with its likelihood score, it is compared against the rest of the population. Only the parameter sets with the highest likelihood scores are kept to act as “genetic donors” for new parameter sets.

Ultimately, the population of parameter sets converges to the phase space point with the highest likelihood, and thus, to a set of parameters that most accurately matches the stellar data. We say an optimization has converged when all members of the population are identical. MilkyWay@home can take between two to four weeks to converge.

2.3.2 Timestep Cut in Parameter Space

Although Shelton et al. (2021) gives us a realistic physical basis for our choice of timestep, there are still regions of our parameter space which could lead to unwieldy run times. In particular, these occur when our generated dwarf progenitor is extraordinarily dense. MilkyWay@home searches over a 5-dimensional parameter space for the parameters with the highest likelihood (search range defined in Table 2.5). However, it would not be wise to explore the full 5D box as the corners of our parameter space with high mass and low radius would produce exceedingly dense progenitors which could take up to 80 days to run. Regions of parameter space that would require more than 1.5 million timesteps in the simulation are excluded from our optimization. Figure 2.10 shows the excised region of parameter space.

Table 2.5: The search ranges for our optimizations against the stellar data.

τ_{evolve} (Gyrs)	R_B (kpc)	$\xi_R = \frac{R_B}{R_B + R_D}$	M_B (SM)	$\xi_M = \frac{M_B}{M_B + M_D}$
[2.0 - 6.0]	[0.01 - 0.50]	[0.1 - 0.6]	[0.1 - 100.0]	[0.001 - 0.95]

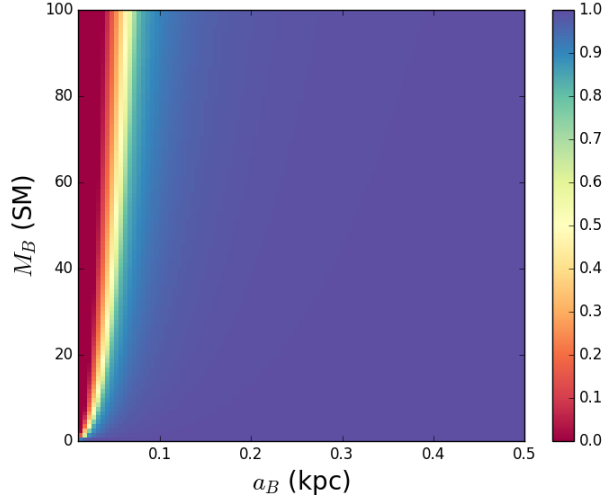


Figure 2.10: Plots illustrating the fraction of our 5D parameter space searched as a function of the baryonic Plummer scale radius (a_B) and the baryonic progenitor mass (M_B). The color of a pixel shows the fraction of the phase space tested for a given (a_B, M_B) . Redder regions in these plots are where a larger percentage of our parameter space is removed. Note that areas where the density of the progenitor galaxy is higher are where we lose most of our phase space. Of the total phase space, 8.92% of our parameter space is cut out.

Placing a non-linear cut on our phase space like this does remove the ability for MilkyWay@home to fit the densest ultra-compact dwarf (UCD) galaxies. For example, simulating a dwarf galaxy like M60-UCD1, one of the densest dwarf galaxies ever measured (Strader et al. 2013), would require roughly 17 million timesteps to evolve the progenitor through our gravitational well for 6.0 gigayears. Fortunately, our fits to the OCS do not suggest a UCD progenitor, but rather a dwarf galaxy with a similar radius but over 100 times less mass.

2.4 Results

We ran six optimizations over our five dwarf parameters using MilkyWay@home’s distributed supercomputer. The raw converged results for each optimization run can be found in Table 2.6. The likelihood score calculated for each run can be found in Table 2.7.

Table 2.6: Fitted parameters from MilkyWay@home optimizations. Errors were calculated using Hessian method as described in Section 2.1.4. Note that the primary source of error is from the navigation of the likelihood surface as measured by the differences between individual runs, and not the statistical error calculated from the Hessian.

Run	τ_{evolve} (Gyrs)	R_B (kpc)	ξ_R	M_B (SM)	ξ_M
1	3.6354 ± 0.0004	0.2560 ± 0.0010	0.3455 ± 0.0014	1.017 ± 0.011	0.0413 ± 0.0005
2	3.6345 ± 0.0004	0.2327 ± 0.0005	0.2543 ± 0.0010	1.146 ± 0.007	0.0179 ± 0.0003
3	3.6333 ± 0.0004	0.1812 ± 0.0007	0.1828 ± 0.0009	1.223 ± 0.020	0.0126 ± 0.0003
4	5.3921 ± 0.0004	0.2060 ± 0.0010	0.1315 ± 0.0015	1.635 ± 0.017	0.1445 ± 0.0006
5	3.6334 ± 0.0003	0.1842 ± 0.0007	0.1820 ± 0.0010	1.251 ± 0.014	0.0119 ± 0.0002
6	5.3956 ± 0.0004	0.1987 ± 0.0004	0.1305 ± 0.0021	1.682 ± 0.009	0.1507 ± 0.0016

Table 2.7: The likelihood scores calculated for each optimized run. Likelihoods closer to 0 imply a better fit.

Run	$\ln(\mathcal{L})$
1	-9.604550
2	-8.809277
3	-7.726065
4	-18.033154
5	-7.997704
6	-18.982624

From these results, we see that the differential evolution genetic algorithm we used did not consistently converge to the same parameter set. We find that for Runs 4 and 6, the algorithm reached a local maximum whose parameters suggest a lower dark matter concentration. Runs 3 and 5 have the highest likelihood scores and seem to reflect the true maximum in the likelihood surface, whereas Runs 1 and 2 got close to this solution but got failed to optimize further.

Table 2.8 gives the same optimized dwarf parameters as in Table 2.6, but measured in physical units. We also directly compare our data histogram against the optimized run with the best likelihood in Table 2.9.

Table 2.8: Radial profile and mass calculated from fitted parameters. The primary source of error is from the navigation of the likelihood surface as measured by the differences between individual runs, and not the statistical error calculated from the Hessian.

Run	τ_{evolve} (Gyrs)	R_B (kpc)	R_D (kpc)	M_B ($10^5 M_\odot$)	M_D ($10^7 M_\odot$)
1	3.6354 ± 0.0004	0.2560 ± 0.0010	0.485 ± 0.004	2.26 ± 0.02	0.524 ± 0.009
2	3.6345 ± 0.0004	0.2327 ± 0.0005	0.682 ± 0.004	2.55 ± 0.02	1.39 ± 0.02
3	3.6333 ± 0.0004	0.1812 ± 0.0007	0.810 ± 0.006	2.72 ± 0.04	2.13 ± 0.05
4	5.3921 ± 0.0004	0.2060 ± 0.0010	1.361 ± 0.019	3.63 ± 0.04	0.215 ± 0.002
5	3.6334 ± 0.0003	0.1842 ± 0.0007	0.828 ± 0.006	2.78 ± 0.03	2.30 ± 0.05
6	5.3956 ± 0.0004	0.1987 ± 0.0004	1.323 ± 0.024	3.74 ± 0.02	0.211 ± 0.003

2.4.1 Using Marginalized Parameter Sweeps to Estimate Error

We use the Hessian method as described in Section 2.1.4 to calculate the errors in each of our fitted parameters. However, to accurately calculate these errors, we need our step size to already be comparable to the errors we wish to calculate. To accomplish this, we perform a marginalized 1-dimensional parameter sweep along each dwarf parameter to visualize an adequate step size. In this section, we describe the methods by which we quickly and efficiently calculate marginalized likelihood scores for these parameter sweeps.

Calculating a marginalized likelihood score from N-body simulations is difficult for two reasons. First, the marginalization computation must be completed within the likelihood space, rather than the $\log(\text{likelihood})$ space we represent in our version of the likelihood score. This requires us to numerically compute integrals over regions of our parameter space where the likelihood is several orders of magnitude lower than machine precision. Second, any marginalization method we implement over our likelihood space is inherently computationally expensive, requiring us to calculate a complicated 4-dimensional integral over a rough likelihood surface. Calculating one likelihood score from a set of parameters requires an N-body simulation. If we were to use a crude 3-point Gaussian quadrature method to calculate each integral, that would require us to run 81 N-body simulations to calculate the marginalized likelihood for one point of our parameter sweep. Assuming we use about 30 points for our parameter sweep, that brings the total number of simulations to 2,430, requiring one to three months for each parameter sweep.

To address the first issue, we define a special convention for adding together likelihood scores. Given two likelihood scores $\ln(\mathcal{L}_>)$ and $\ln(\mathcal{L}_<)$ (where $\mathcal{L}_> > \mathcal{L}_<$), we define their sum $\ln(\mathcal{L}_{sum})$ to follow the following equation:

Table 2.9: Histograms of stellar density and Beta dispersion σ_B along the stream. Left histogram generated from data. Right histogram shows best simulation fit. The number of counts N in each bin is normalized for the purposes of the Earth Mover Distance calculation in our likelihood score. δ_N and δ_{σ_B} are the errors in the normalized counts and the Beta dispersion respectively.

Λ	DATA				SIMULATION (RUN 3)			
	N	δ_N	σ_B	δ_{σ_B}	N	δ_N	σ_B	δ_{σ_B}
-31.5	0.0316	0.0072	0.0264	0.0022	1.543	0.130
-28.5	0.0304	0.0078	1.003	0.323	0.0261	0.0021	1.220	0.104
-25.5	0.0186	0.0081	0.0284	0.0022	1.315	0.106
-22.5	0.0114	0.0089	0.0301	0.0023	1.184	0.092
-19.5	0.0266	0.0091	1.001	0.208	0.0273	0.0022	1.138	0.094
-16.5	0.0261	0.0095	0.0278	0.0022	1.068	0.087
-13.5	0.0433	0.0101	0.0300	0.0023	1.071	0.084
-10.5	0.0382	0.0105	1.076	0.218	0.0284	0.0022	0.994	0.080
-7.5	0.0172	0.0109	0.0216	0.0020	0.969	0.089
-4.5	0.0201	0.0117	0.0216	0.0020	0.924	0.085
-1.5	0.0224	0.0127	0.612	0.146	0.0223	0.0020	0.716	0.066
1.5	0.0302	0.0127	0.0207	0.0019	0.829	0.077
4.5	0.0227	0.0118	0.0190	0.0018	0.764	0.075
7.5	0.0297	0.0108	0.791	0.308	0.0209	0.0019	0.656	0.062
10.5	0.0011	0.0110	0.0190	0.0018	0.707	0.070
13.5	0.0261	0.0108	0.0220	0.0020	0.502	0.046
16.5	0.0277	0.0109	0.947	0.257	0.0262	0.0022	0.702	0.058
19.5	0.0262	0.0113	0.0287	0.0023	0.666	0.053
22.5	0.0325	0.0139	0.0388	0.0026	0.610	0.038
25.5	0.0655	0.0146	0.869	0.195	0.0546	0.0031	0.660	0.038
28.5	0.0618	0.0140	0.0679	0.0035	0.561	0.029
31.5	0.0586	0.0141	0.0745	0.0036	0.710	0.035
34.5	0.0925	0.0145	2.606	1.217	0.0784	0.0037	0.771	0.037
37.5	0.0870	0.0149	0.0800	0.0038	0.784	0.037
40.5	0.0655	0.0144	0.0598	0.0033	0.887	0.049
43.5	0.0240	0.0152	1.325	0.311	0.0527	0.0031	0.863	0.051
46.5	0.0629	0.0150	0.0468	0.0029	0.660	0.042

$$e^{\ln(\mathcal{L}_{sum})} = e^{\ln(\mathcal{L}_{>})} + e^{\ln(\mathcal{L}_{<})}. \quad (2.29)$$

Doing some simple math, we can rewrite this equation as:

$$\ln(\mathcal{L}_{sum}) = \ln(\mathcal{L}_{>}) + \ln(1 + e^{\ln(\mathcal{L}_{<}) - \ln(\mathcal{L}_{>})}) \quad (2.30)$$

Overall, larger likelihood scores dominate the smaller ones. In a double precision computation, if the two likelihood scores differ by more than 64, the smaller likelihood score is completely ignored in the calculation. Therefore, given a list of likelihoods to add, it is most prudent to add together the smaller likelihood scores first.

To calculate the marginalized likelihood score $\ln(\tilde{\mathcal{L}})$, we use a Monte Carlo integration method using N random points in our parameter space of volume V . To determine the minimum number of random points we need for our calculation, we need to know how N affects the error in $\ln(\tilde{\mathcal{L}})$. Given a list of likelihoods $\{\mathcal{L}\}$, the marginalized likelihood $\tilde{\mathcal{L}}$ calculated using Monte Carlo is given by the following equation:

$$\tilde{\mathcal{L}} = \frac{V}{N} \sum_i \mathcal{L}_i = V\mu_{\mathcal{L}}, \quad (2.31)$$

where $\mu_{\mathcal{L}}$ is the average of \mathcal{L} . From this, it is clear that the main source of error in this equation comes from the calculation of $\mu_{\mathcal{L}}$, which is simply the error in the mean:

$$\delta_{\tilde{\mathcal{L}}} = V \frac{\sigma_{\mathcal{L}}}{\sqrt{N}}, \quad (2.32)$$

where $\sigma_{\mathcal{L}}$ is the standard deviation of the population that \mathcal{L} was pulled from. Performing basic error propagation, we find the error in the marginalized likelihood score to be:

$$\delta_{\ln(\tilde{\mathcal{L}})} = \frac{\delta_{\tilde{\mathcal{L}}}}{\tilde{\mathcal{L}}} = \frac{\sigma_{\mathcal{L}}}{\mu_{\mathcal{L}}\sqrt{N}}. \quad (2.33)$$

It can then be shown that:

$$\delta_{\ln(\tilde{\mathcal{L}})} = \frac{1}{\sqrt{N-1}} \sqrt{N \frac{\sum \mathcal{L}_i^2}{(\sum \mathcal{L}_i)^2} - 1}. \quad (2.34)$$

Equation 2.34 is what we use in Figure 2.11 to determine the errors in our parameter sweeps (shaded in blue). However, given this equation, we can generate a rough estimate for what this error looks like on average. We can reorder our list $\{\mathcal{L}\}$ in descending order, where \mathcal{L}_0 is the highest likelihood and \mathcal{L}_1 is the second highest likelihood. Since we anticipate that each pair of adjacent entries in the reordered list differs by tens of orders of magnitude, we can make the following approximations:

$$\sum_i \mathcal{L}_i \approx \mathcal{L}_0 + \mathcal{L}_1 = \mathcal{L}_0 (1 + k), 0 < k \leq 1, \quad (2.35)$$

$$\sum_i \mathcal{L}_i^2 \approx \mathcal{L}_0^2 + \mathcal{L}_1^2 = \mathcal{L}_0^2 (1 + k^2), 0 < k \leq 1, \quad (2.36)$$

where $k = p_1/p_0$. We can make this approximation since we expect only the largest few likelihoods in our list to impact the summations. Substituting these quantities into Equation 2.34 gives us:

$$\delta_{\ln(\tilde{\mathcal{L}})} \approx \frac{1}{\sqrt{N-1}} \sqrt{N \frac{(1+k^2)}{(1+k)^2} - 1} = \sqrt{\frac{1+k^2 - 2k/(N-1)}{(1+k)^2}}. \quad (2.37)$$

Taking the limit where k approaches 0, we find that $\delta_{\tilde{\mathcal{L}}}$ takes the form:

$$\delta_{\ln(\tilde{\mathcal{L}})} \approx \sqrt{1 - \frac{2k}{N-1}} \approx 1. \quad (2.38)$$

Surprisingly, we find that the effect of N on $\delta_{\ln(\tilde{\mathcal{L}})}$ is incredibly muted in our likelihood space. In fact, looking at the parameter sweeps in Figure 2.11, we find that the errors associated with each point are extremely close to 1. It is therefore unreasonable to use a bound in $\delta_{\ln(\tilde{\mathcal{L}})}$ to select a minimum value for N .

Instead, we should set N to be the minimum number of measurements by which the unbiased value of σ_p can be most efficiently estimated to a set percentage. To determine σ_p to within 1%, the unbiased variance σ_p^2 must be determined to within 2%. Also, the relationship between σ_p^2 and the sample variance s_p^2 can be estimated using Bessel's correction:

$$\frac{\sigma_p^2}{s_p^2} = \frac{N}{N-1}. \quad (2.39)$$

Therefore, to know σ_p^2 to within 2%, we measure the likelihoods from 50 random points for each Monte Carlo approximation to calculate our parameter sweeps.

The likelihood surfaces we observe are not smooth, but we still see an overall dip near the optimal parameters. A Gaussian peak in the likelihood surface translates to a parabola in the log(likelihood) parameter sweeps. Using the apparent width of these peaks, we determine the appropriate step size to use when calculating the Hessian for the purposes

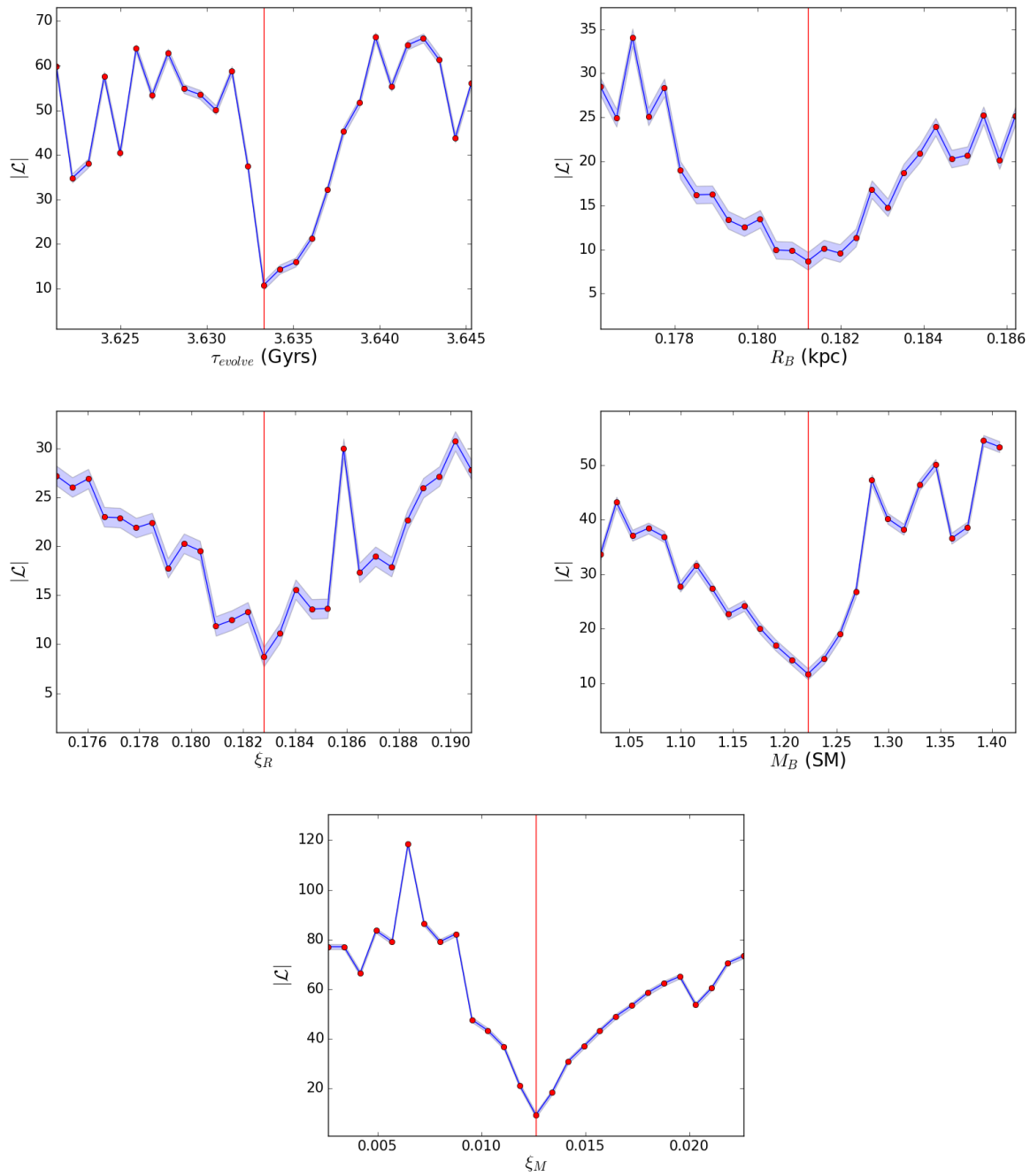


Figure 2.11: 1-dimensional marginalized parameter sweeps over each dwarf parameter. The blue line shows the negative of the likelihood calculated at each dwarf parameter value while the red line indicates the best fitted value for that parameter (from Run 3). The blue shaded area shows the error in our marginalized likelihood score.

of error analysis.

2.4.2 Full Result

After finding the converged values for each of our optimizations, we reran the simulations using each best fit parameter set to check that the generated simulated tidal stream matched the data, as shown in Figures 2.12 and 2.13. In all optimizations, we see a general trend where the B dispersion is relatively well fit for $\Lambda < 10^\circ$, but poorly fit for $\Lambda > 10^\circ$, especially near the core. This is likely due to the larger errors in the B dispersion associated with the region near the core in the DEC data at higher Λ_{OCS} . Most of the optimizations (Runs 2, 3 and 5) converged to similar tidal debris that closely resembles the OCS. Runs 1, 4, and 6 had a poorer likelihood score compared to the other optimizations, and we see that the distribution of bodies near its core is wider across Λ_{OCS} than the data. Due to the generally poor fit of these runs, we exclude them from our analysis of the OCS progenitor.

It should be noted that none of these optimizations reproduce the gap ($\sigma = 1.06$) in the tidal stream at $\Lambda = 10.5^\circ$. This is not only because our data can barely resolve this gap, but also because such a gap would only appear as a result of substructures within the Milky Way’s dark matter halo (Koposov et al. 2019), which is not accounted for in our current halo potential.

Taking the average of the 3 best fitted parameters (Runs 2, 3, and 5), we estimate the following physical properties of the OCS progenitor dwarf galaxy:

$$\tau_{evolve} = 3.6337 \pm 0.0004^{(stdev.)} \pm 0.0004^{(Hess.)} \text{Gyrs} \quad (2.40)$$

$$R_B = 0.1994 \pm 0.0167^{(stdev.)} \pm 0.0006^{(Hess.)} \text{kpc} \quad (2.41)$$

$$R_D = 0.773 \pm 0.046^{(stdev.)} \pm 0.005^{(Hess.)} \text{kpc} \quad (2.42)$$

$$M_B = (2.68 \pm 0.07^{(stdev.)} \pm 0.03^{(Hess.)}) \times 10^5 M_\odot \quad (2.43)$$

$$M_D = (1.94 \pm 0.28^{(stdev.)} \pm 0.04^{(Hess.)}) \times 10^7 M_\odot \quad (2.44)$$

$$M_{total} = (1.97 \pm 0.28^{(stdev.)} \pm 0.04^{(Hess.)}) \times 10^7 M_\odot \quad (2.45)$$

where the first error is the standard deviation of the mean and the second error is the propagated error from each individual optimization’s Hessian error. We see that our largest source of error comes from the optimization process rather than the curvature of the likelihood surface. So, we only propagate those errors. From these parameters we calculate a

fitted mass-to-light ratio of $\gamma = 73.5 \pm 10.6$.

Note that these results and errors are derived under the assumption of a perfect Milky Way potential, orbit, and functional form of the dwarf galaxy and do not include systematic errors due to these assumptions.

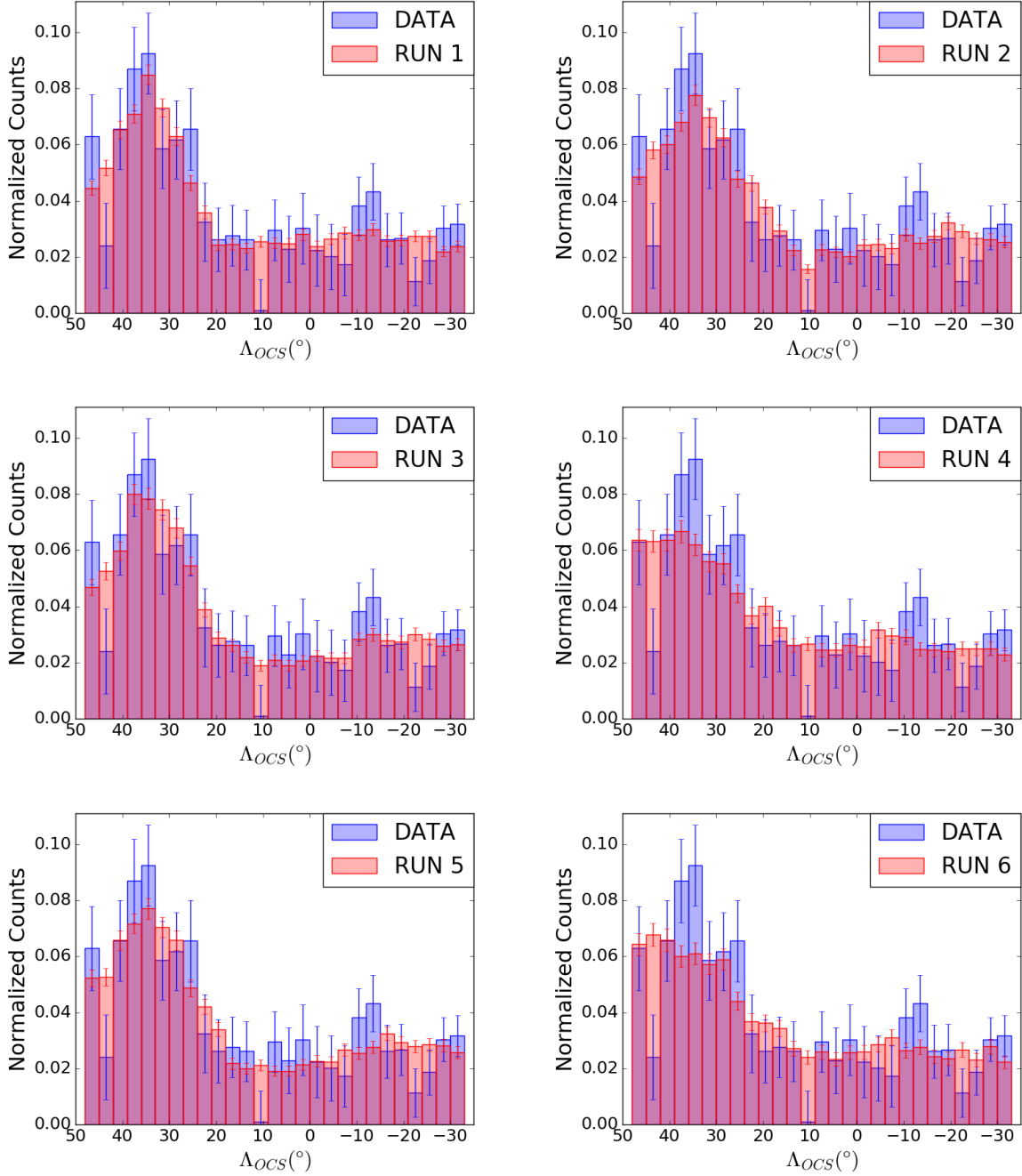


Figure 2.12: End state histograms showing normalized body counts over Λ_{OCS} .

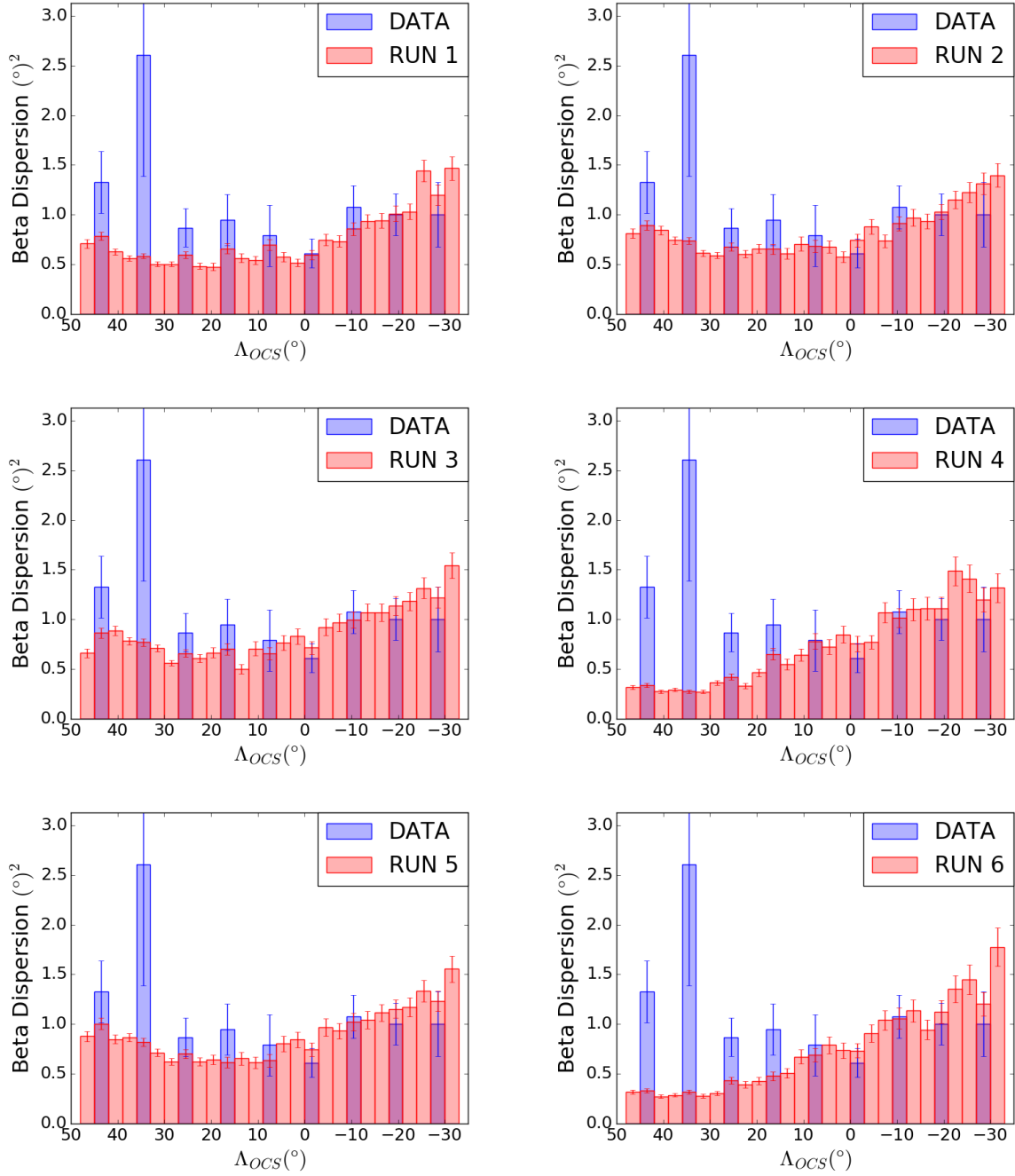


Figure 2.13: End state histograms showing B dispersion over Λ_{OCS} .

2.5 Discussion

2.5.1 An Extremely Low-Mass Diffuse Progenitor

Our estimated mass for the OCS’s progenitor is not consistent with the masses of ultrafaint dwarf galaxies measured from the dispersion of the line-of-sight velocities, which assume spherical symmetry and virial equilibrium. To illustrate this point more clearly, let us use these parameters to calculate the total mass within 300 pc of the progenitor’s center. Within this radius, we calculate $(1.6 \pm 0.1) \times 10^5 M_\odot$ of baryonic matter and $(9.2 \pm 2.2) \times 10^5 M_\odot$ of dark matter. In total, we find a mass of $(1.1 \pm 0.2) \times 10^6 M_\odot$ within 300 pc of the progenitor’s center, roughly one order of magnitude smaller than what is claimed possible in Strigari et al. (2008). However, these results are also about one order of magnitude larger than the mass estimated in Newberg et al. (2010), who estimated the total OCS progenitor mass from N-body simulations to be $2.5 \times 10^6 M_\odot$. We note that this mass calculation was done under the assumption that the dark matter followed a Plummer profile. However, we note that 300 pc is very close to the half-light radius of the best fit dwarf progenitor, and Shelton et al. (2021) showed that our algorithm was most sensitive to the mass within the half-light radius.

2.5.2 The Location of the Remnant Progenitor

We use our fitted simulations to estimate the current location of the OCS’s progenitor. Note that there is a peak in the star counts of both the data and the stream simulation around $\Lambda \sim 35^\circ$ in Figure 2.12. We know this point of high stellar density must be the location of the progenitor remnant because there is no other reason for stars to pile up at this position, so far from apogalacticon. In this section we will determine the position of the progenitor remnant and show that it is no longer gravitationally bound.

For each optimization, we translate the positions of all the bodies into spherical coordinates (r, θ, ϕ) , where the origin is the Galactic Center and the z-axis is perpendicular to the Galactic Plane. We bin the baryonic (stellar) bodies in the polar (θ) and azimuthal (ϕ) angles and select the bin with the highest density of bodies. Using the bodies in the selected bin, we calculate the average Galactocentric distance (r) and its standard deviation. To improve the calculation, we remove all bodies with distances more than 2.5 standard deviations away from the mean and recalculate the average, a process known as sigma-clipping. We perform this sigma-clipping 50 times for the sake of being thorough. After calculating the 3D point in space with the highest density of baryonic bodies, we translate it into Galactic

and Equatorial coordinates. The progenitor position of each run is shown in Table 2.10.

Table 2.10: The current sky position of the progenitor in both Galactic and Equatorial coordinates calculated from each optimization.

Run	l ($^{\circ}$)	b ($^{\circ}$)	α ($^{\circ}$)	δ ($^{\circ}$)
2	264.0	44.4	165.7	-10.1
3	270.4	40.5	167.7	-15.9
5	260.4	46.0	164.6	-7.3

Our simulations predict that the remnant progenitor core is expected to be at around $(l, b) = ((264.9 \pm 2.9)^{\circ}, (43.6 \pm 2.8)^{\circ})$, or $(\alpha, \delta) = ((166.0 \pm 0.9)^{\circ}, (-11.1 \pm 2.5)^{\circ})$. This is close to the sky position where Grillmair et al. (2015) detected the OCS’s possible missing progenitor, $(\alpha, \delta) \approx (167^{\circ}, -14^{\circ})$.

We next check whether or not the simulated remnant progenitor is still gravitationally bound. For each optimization, we calculate the total kinetic energy of the remnant and compare it to its gravitational potential energy. We select bodies that are within 0.5 kpc of the progenitor’s center since that distance is roughly the cylindrical radius of our simulated stream (see Figure 2.14). We transform our bodies into the center-of-momentum reference frame and add together the kinetic energies ($E_K = \frac{1}{2} \sum_i^N m_i v_i^2$) of all the bodies. Similarly, we add together the gravitational energies from each pair of bodies ($E_P = \sum_i^N \sum_{i < j}^N \frac{G m_i m_j}{|r_i - r_j|}$). We record the energies we calculate for each optimization in Table 2.11.

Table 2.11: The kinetic and gravitational potential energies of the simulated remnant progenitor in each optimization. Note that the kinetic energy is much larger than the gravitational potential energy, indicating that the progenitor is no longer gravitationally bound.

Run	E_K ($SM kpc^2 Gyr^{-2}$)	E_P ($SM kpc^2 Gyr^{-2}$)
2	5.977	-0.096
3	7.587	-0.092
5	5.751	-0.093

We find that the simulated remnant progenitor has a kinetic energy of (6.4 ± 0.6) SM kpc² Gyr⁻² and a gravitational potential energy of $-(0.094 \pm 0.001)$ SM kpc² Gyr⁻², and is thus

gravitationally unbound. Looking at the stream directly in Figure 2.14 makes this readily apparent.

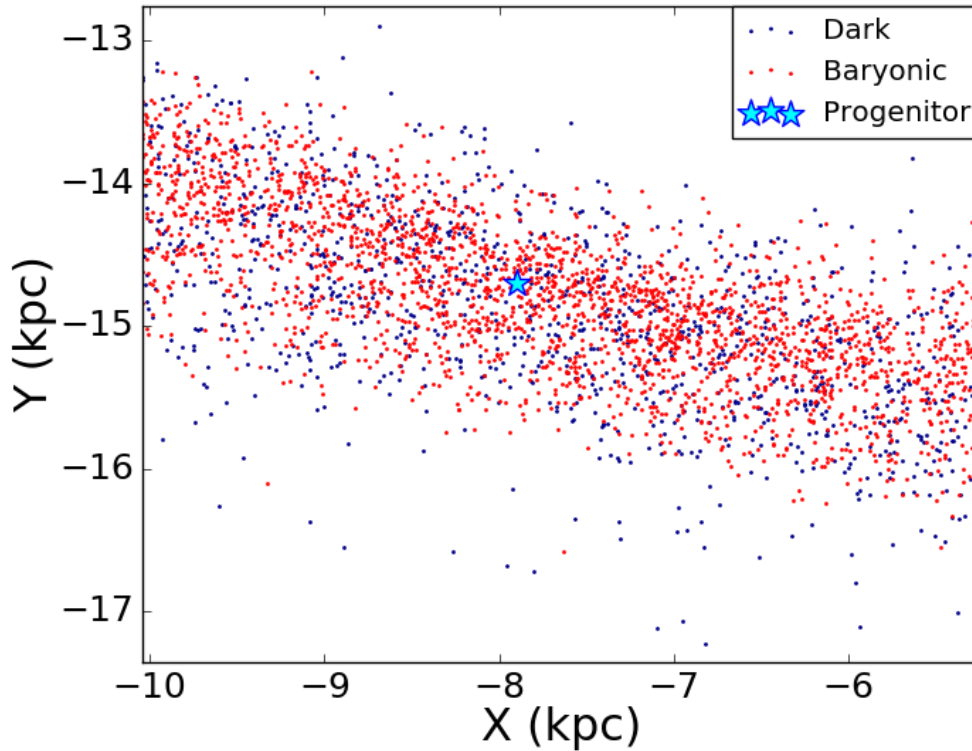


Figure 2.14: Close-up view of simulated progenitor remnant (Run 3). The remnant has been fully disrupted and the progenitor’s original structure cannot be easily observed.

2.5.3 The Distribution of Dark Matter Along the Stream

The tidal debris we observe in our simulations reveals some interesting aspects of the distribution of baryonic and dark matter in the stellar stream. We analyze the distribution of matter within the tails of the simulated OCS. Figure 2.15 shows that the tails of the OCS are thicker than the central region nearer to the progenitor remnant. There are two possible contributions to this thickening: either the tails are naturally fanning out at apogalacticon due to the symplectic (phase space area preserving) property of gravitational systems, or

material with a higher velocity dispersion in the progenitor is being tidally stripped first, populating the regions of the stream further from the core.

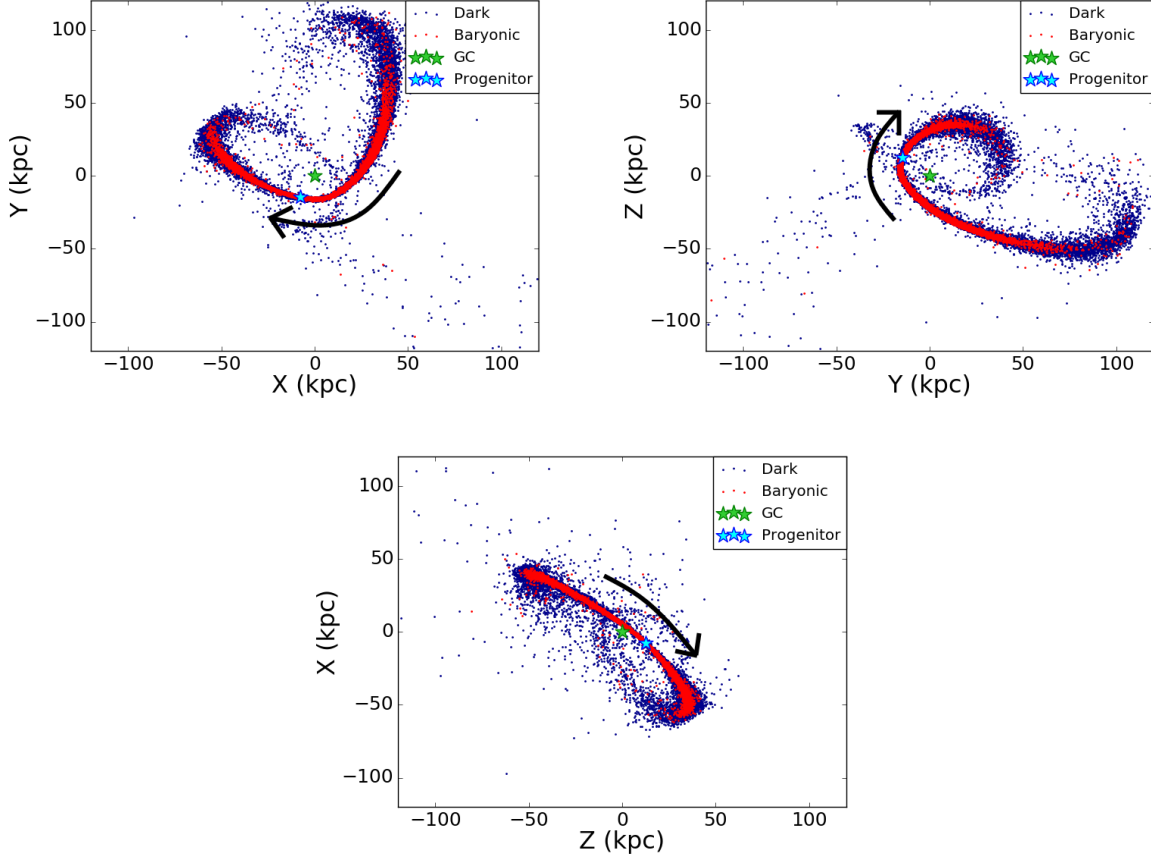


Figure 2.15: Tidal stream produced by Run 3 viewed from different angles. We see that the tidal tails of the stream are thicker near apogalacticon. The green star is the Galactic Center (GC) and the cyan star is the location of the OCS’s progenitor dwarf galaxy.

We can test this by looking at the simulated tidal stream generated using Run3, our best run, when the core is at apogalacticon. If a high velocity dispersion has a stronger impact on the thickness of the tails than the fanning at apogalacticon, then the tails should maintain their thickness even when they are not at apogalacticon. However, as can be clearly seen in Figure 2.16, while the core is at apogalacticon it bears a similar thickness to the tails in the fully evolved stream. Also, the tails of the stream in the core-apogalacticon state share

a similar thickness to the core in the final state. This indicates that the thickening of the tails in the final state is mostly caused by the natural fanning that occurs at apogalacticon.

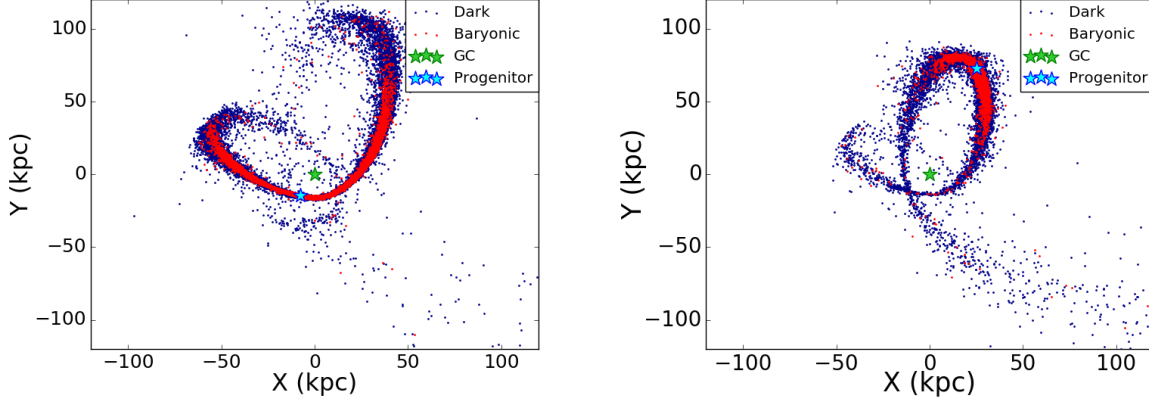


Figure 2.16: The final state stream (left) compared to the stream while its progenitor is at apogalacticon (right). The thickness of the stream at apogalacticon is comparable in both images regardless of where the progenitor is.

In addition to being thicker than the core, the tails also possess a higher mass-to-light ratio. We see this in Figure 2.17, which maps out the density of baryons and dark matter along Λ_{OCS} . While the core of the OCS has a relatively high dark matter density, it is still not as high as in its tails, where the mass-to-light ratio jumps to as high as 142 in Run 3. Due to its high dark matter concentration and low baryonic contamination, the tails of the OCS might serve as a better candidate for indirect dark matter detection than the central core.

2.5.4 The Effect of the LMC on the OCS

When creating our data histogram, we excluded the southern tail of the OCS because our current N-body simulations do not account for the gravitational influence of the LMC. While the OCS's northern tail is not greatly influenced by the presence of the LMC, its southern tail exhibits peculiar perturbations in its stellar velocities which cannot be explained by a static Milky Way potential or a low mass LMC (Erkal et al. 2019). While this

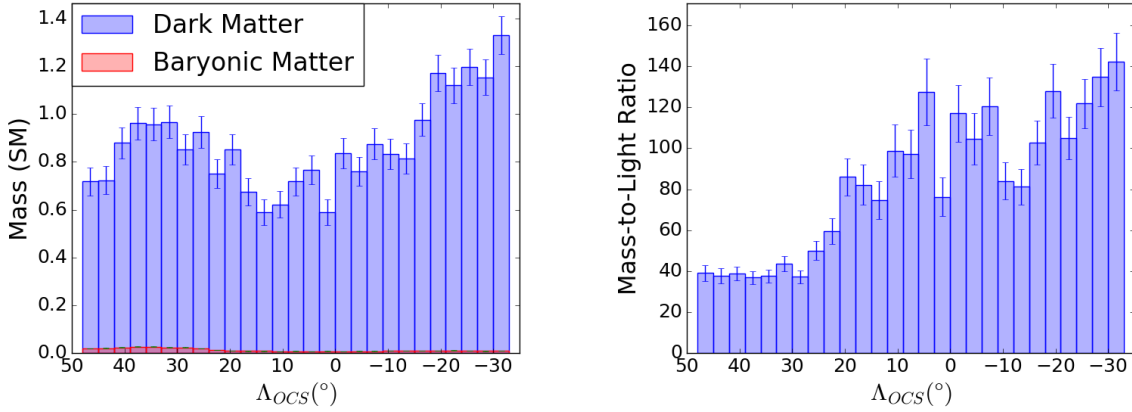


Figure 2.17: Distribution of baryons and dark matter in Run 3. The tails clearly have a much higher mass-to-light ratio at points that are farther away from the core.

dissertation does not fit data from the southern tail, we do plan on including the tail and the gravitational effects of the LMC in a future MilkyWay@home paper.

2.5.5 Sources of Systematic Error

While we measured the OCS progenitor properties without the assumption of dynamical equilibrium, our measurement still relies on a few key presuppositions. First, we assume the Milky Way has a static, axisymmetric Galactic potential for the entirety of the simulation. There currently exists no mechanism in our N-body simulations for our progenitor to perturb the Milky Way to give rise to second-order effects. We do not include any time-dependence in our Milky Way potential. We do not take into account other gravitating systems which could interact with the OCS, such as the LMC or other Milky Way satellites. Adding the LMC to our simulations would not only further perturb the stream, but also induce a reflex motion in the entirety of the Milky Way Galaxy, forcing the dwarf galaxy progenitor to fall into an accelerating potential. It has also been recently discovered that most of the mass in the inner disk of the Milky Way is located in its fast rotating bar (Portail et al. 2017), which is also not included in our Galactic model.

Second, we fix the orbit of our simulated OCS throughout each of our optimizations to the values determined in Newberg et al. (2010). Since the orbit is fit to the northern (leading) tail of the OCS, which should be at lower energy than the orbit of the dwarf galaxy itself, we expect the orbit is not exact. We have not explored how the orbit affects the fit properties of

the dwarf galaxy progenitor. However, we are only using the stellar distributions along and across the stream, which are not thought to be strongly affected by small differences in the orbit, to constrain the stream. Additionally, the stream created in our best fit simulation is similar to the stream observed, which gives us some confidence that our orbit and progenitor properties are reasonable.

Third, we assume that the OCS progenitor has the form of a two component Plummer sphere. We used a Plummer profile for our progenitor because it is a simple model with an analytic energy distribution function. However, it is probably not the best (and certainly not a perfect) model, particularly for the dark matter halo. We are unsure whether using a different profile, such as a Hernquist or Navarro-Frenk-White profile, would significantly affect the optimization results.

Finally, there is the measurement of baryonic mass in the OCS. As discussed in Section 2.1.4.1, using the redder NGC 5053 to estimate the mass of the bluer OCS overestimates the mass per turnoff star. From our analysis, we find that the stellar mass could be reduced by a factor of 4 due to inaccuracies in our calculation of the stellar mass per turnoff. This would affect the mass to light ratio, but would not significantly change the total mass within 300 pc of the center of the progenitor.

CHAPTER 3

ADDING THE EFFECTS OF THE LARGE MAGELLANIC CLOUD

Since Erkal et al. (2019) showed that the LMC only seemed to have an impact on the Southern Tail of the OCS, we originally did not include its effects since we were only fitting the Northern Tail and the progenitor core. However, in order for us to now measure how much the LMC impacts our full fit, it is now necessary for us to implement a LMC within our N-body code. This chapter details all of the major changes we made to accommodate the inclusion of this relatively massive satellite galaxy.

3.1 Modeling the LMC

We model the LMC as a statically defined Plummer sphere (Equation 2.21) that falls into the Milky Way over the course of the simulation. We set the final LMC position and velocity in accordance with Erkal et al. (2019) at $\mathbf{x} = (-1.1, -41.1, -27.9)$ kpc and $\mathbf{v} = (-57, -226, 221)$ kpc Gyr⁻¹, respectively, in right-handed Galactic Cartesian coordinates. While our code performs the reverse orbit calculation for the progenitor dwarf galaxy, it simultaneously calculates the initial position and velocity of the LMC. We allow the dwarf galaxy to feel the gravitational influence of the LMC, but we have the LMC ignore the gravitational impact of the dwarf due to its much smaller mass¹.

One thing that particularly complicates this implementation is that we are anticipating that the LMC has a mass on the order of 10% of the Milky Way. This means we would expect the LMC to exert a non-negligible gravitational acceleration on the Milky Way. Since our simulations are performed with the Galactic Center as the origin, introducing an LMC places our calculations within a non-inertial reference frame. To account for this shift in our Velocity Verlet integrations (Verlet 1967), we subtract the acceleration on the Milky Way’s center by the LMC from every acceleration in the timestep:

$$\mathbf{a}'(t) = \mathbf{a}(t) - \mathbf{a}_{\text{MW}}(t). \quad (3.1)$$

¹It should be noted that within this analysis, to reproduce the fits in Mendelsohn et al. (2022), we set the LMC mass to zero. Regardless, we are still implementing this code to identify how adding mass to the LMC would impact such a fit.

3.2 Dynamical Friction

Since we are modeling the LMC as a large static potential falling into a static Milky Way, the gravitational drag we would expect to see between live bodies does not appear within our simulation. Therefore, to approximate this process, we use Chandrasekhar's formula (Chandrasekhar 1943) to describe the dynamical friction acting on the LMC:

$$\mathbf{F}_{\text{dyn}} = -16\pi^2 \ln(\Lambda) m_s (M_{\text{LMC}} + m_s) \frac{\mathbf{v}_{\text{LMC}}}{v_{\text{LMC}}^3} \int_0^{v_{\text{LMC}}} v^2 f(\mathbf{x}, v) dv, \quad (3.2)$$

where $\ln(\Lambda)$ is the Coulomb Logarithm ($\Lambda = 1.6$), m_s is the average mass of halo stars, M_{LMC} is the mass of the LMC, \mathbf{v}_{LMC} is the velocity of the LMC, and $f(\mathbf{x}, v)$ is the number density distribution of halo stars.

The hardest calculation required in this formula is the distribution function $f(\mathbf{x}, v)$. Since MilkyWay@home has several halos to select from, properly implementing dynamical friction would require us to calculate this distribution function for an arbitrary halo density, a notoriously difficult calculation. It is therefore necessary for us to find ways to approximate this value. The first assumption we make is that the distribution can be locally approximated at \mathbf{x} as a uniform density field with a Maxwellian velocity distribution:

$$f(\mathbf{x}, v) = \frac{n(\mathbf{x})}{(2\pi\sigma^2(\mathbf{x}))^{3/2}} e^{-\frac{v^2}{2\sigma^2(\mathbf{x})}}, \quad (3.3)$$

where $n(\mathbf{x})$ is the number density at \mathbf{x} and $\sigma^2(\mathbf{x})$ is the 1-dimensional velocity dispersion of halo stars at \mathbf{x} . Plugging this distribution into Chandrasekhar's formula yields:

$$\mathbf{F}_{\text{dyn}} = -\frac{4\pi^2 \ln(\Lambda) \rho_{\text{halo}}(\mathbf{x}) M_{\text{LMC}}}{v_{\text{LMC}}^3} \left[\text{erf} \left(\frac{v_{\text{LMC}}}{\sigma(\mathbf{x})\sqrt{2}} \right) - \sqrt{\frac{2}{\pi}} \frac{v_{\text{LMC}}}{\sigma(\mathbf{x})} e^{-\frac{v_{\text{LMC}}^2}{2\sigma^2(\mathbf{x})}} \right] \mathbf{v}_{\text{LMC}}, \quad (3.4)$$

where $\rho_{\text{halo}}(\mathbf{x})$ is the mass density of the halo.

This simplification alone is not enough as we still do not know the velocity dispersion $\sigma^2(\mathbf{x})$. To approximate this quantity we look at the Spherical Jeans Equation:

$$\frac{\partial(\rho_{\text{halo}}(r)\sigma_r^2)}{\partial r} + \frac{2\rho_{\text{halo}}(r)}{r} (\sigma_r^2 - \sigma_\theta^2) + \rho_{\text{halo}}(r) \frac{d\Phi_{\text{halo}}}{dr} = 0, \quad (3.5)$$

where Φ_{halo} is the gravitational potential of the Milky Way halo. Assuming isotropy ($\sigma_r^2 = \sigma_\theta^2$)

simplifies our equation to:

$$\frac{\partial(\rho_{halo}(r)\sigma_r^2)}{\partial r} = -\rho_{halo}(r)\frac{d\Phi_{halo}}{dr}. \quad (3.6)$$

Since $\mathbf{a}_{halo} = -\nabla\Phi_{halo}$, it is easy to show:

$$\frac{\partial(\rho_{halo}(r)\sigma_r^2)}{\partial r} = \rho_{halo}(r)(\mathbf{a}_{halo} \cdot \hat{r})dr. \quad (3.7)$$

We then integrate the above equation from radius where our LMC resides (r_{LMC}) to infinity:

$$\rho_{halo}(r)\sigma_r^2|_{r=\infty} - \rho_{halo}(r)\sigma_r^2|_{r=r_{LMC}} = \int_{r_{LMC}}^{\infty} \rho_{halo}(r)(\mathbf{a}_{halo} \cdot \hat{r})dr. \quad (3.8)$$

As we look at larger radii, the density of our halo naturally drops to zero. In addition, for a gravitationally bound body to exist at larger radii, the velocity of the body must approach zero, implying that its velocity dispersion also approaches zero. Therefore, we find that we can approximate the 1-dimensional velocity dispersion of the halo at r_{LMC} using the following integral:

$$\sigma^2(r_{LMC}) = -\frac{1}{\rho_{halo}(r_{LMC})} \int_{r_{LMC}}^{\infty} \rho_{halo}(r)(\mathbf{a}_{halo}(r) \cdot \hat{r})dr. \quad (3.9)$$

3.3 Orbit Fitting

Adding an LMC to the Milky Way system modifies the gravitational potential to such a degree that the orbit fit in Newberg et al. (2010) would no longer be valid, which would require us to calculate a new orbit before optimizing our dwarf parameters. However, as stated in Chapter 2, there could exist a coupling between the orbital and dwarf parameters. Therefore, it would be in our best interest to try and fit both sets of parameters simultaneously.

To accomplish this, we add three new components to the likelihood score calculation: the VLOS Component, the $\langle B \rangle$ Component, and the Distance Component. The VLOS Component performs a bin-by-bin comparison the average line-of-sight velocities between

the data and simulation histograms, outputting a χ^2 metric:

$$\ln(\mathcal{L}_{VLOS}) = -\frac{1}{2} \sum_i^{N_{bins}} \frac{(\mu_{VLOS,data,i} - \mu_{VLOS,sim,i})^2}{\sigma_{VLOS,data,i}^2 + \sigma_{VLOS,sim,i}^2}. \quad (3.10)$$

The $\langle B \rangle$ Component performs a similar calculation using the average B in each bin:

$$\ln(\mathcal{L}_{\langle B \rangle}) = -\frac{1}{2} \sum_i^{N_{bins}} \frac{(\mu_{B,data,i} - \mu_{B,sim,i})^2}{\sigma_{B,data,i}^2 + \sigma_{B,sim,i}^2}, \quad (3.11)$$

and the Distance Component looks at the average Galactocentric distance R in each bin:

$$\ln(\mathcal{L}_R) = -\frac{1}{2} \sum_i^{N_{bins}} \frac{(\mu_{R,data,i} - \mu_{R,sim,i})^2}{\sigma_{R,data,i}^2 + \sigma_{R,sim,i}^2}. \quad (3.12)$$

It should be noted in this analysis that our likelihood score of interest does not use these additional components, but we anticipate implementing these calculations in future fits involving the LMC.

CHAPTER 4

AUTOMATIC DIFFERENTIATION

MilkyWay@home (MW@home) is a distributed supercomputer tasked with measuring the amount of dark matter within the progenitor dwarf galaxy by optimizing the results of N-body simulations mimicking its tidal disruption. It accomplishes this by using a differential evolution genetic algorithm to find which progenitor properties best fit an observable stream in the sky. Given a set of parameters and an input stream, the N-body simulation generates a simulated tidal stream from those parameters and compares it to the input stream by calculating a likelihood score. The differential evolution algorithm collects these likelihood scores and uses them to find the parameter set that best fits the data.

These fitted parameters, however, are heavily dependent on the accuracy of the model which simulates them. The uncertainties in Galactic parameters (disk thickness, halo mass, etc.) as well as the limited criteria used in the likelihood calculation must provide an upper bound on how accurately the parameters of the dwarf progenitor can be measured. To understand these properties, it is necessary to understand the curvature and texture of the likelihood surface we are optimizing on. Therefore, we require accurate derivative information of the likelihood surface with regards to these Galactic parameters.

In Section 4.1, we derive the sensitivity matrix which we then use to determine how much a local maximum in our likelihood space would change given a shift in the model parameters. In Section 4.2, we explain why using a numerical approach to calculating the derivatives necessary for our sensitivity matrix is computationally difficult, explaining the need for the automatic differentiation process described in Section 4.3. Finally, in Section 4.4, we explore which processes in our N-Body simulation automatic differentiation fail to propagate derivative information, and the subsections within mathematically derive the functions we implement to recover said information.

4.1 Derivation of the Sensitivity Matrix

Let \mathbf{p} be the vector of all parameters optimized through differential evolution, and let \mathbf{x} be the vector of all other parameters kept at a constant value (disk thickness, halo mass, etc.). Also, let $\mathcal{L}(\mathbf{x}, \mathbf{p})$ be the likelihood score evaluated for some \mathbf{x} and \mathbf{p} . We define $\mathbf{p}_c(\mathbf{x})$ as the location of a local maximum of $\mathcal{L}(\mathbf{x}, \mathbf{p})$ in \mathbf{p} -space for a given \mathbf{x} . Assuming $\mathcal{L}(\mathbf{x}, \mathbf{p})$

is twice-differentiable in \mathbf{p} , we can approximate $\mathcal{L}(\mathbf{x}, \mathbf{p})$ for when \mathbf{p} is sufficiently close to $\mathbf{p}_c(\mathbf{x})$:

$$\mathcal{L}(\mathbf{x}, \mathbf{p}) = \sum_i \sum_j a_{ij}(\mathbf{x})(p_i - p_c(\mathbf{x})_i)(p_j - p_c(\mathbf{x})_j) + C(\mathbf{x}), \quad (4.1)$$

where $a_{ij}(\mathbf{x})$ and $C(\mathbf{x})$ are some functions purely dependent on \mathbf{x}^2 . Taking the derivative with respect to one of the optimized parameters p_k gives us:

$$\frac{\partial \mathcal{L}}{\partial p_k} = \sum_j [a_{jk}(\mathbf{x}) + a_{kj}(\mathbf{x})](p_j - p_c(\mathbf{x})_j). \quad (4.2)$$

We can also easily show that:

$$\frac{\partial^2 \mathcal{L}}{\partial p_j \partial p_k} = a_{jk}(\mathbf{x}) + a_{kj}(\mathbf{x}). \quad (4.3)$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial p_k} = \left(\frac{\partial^2 \mathcal{L}}{\partial p_k \partial \mathbf{p}} \right) \cdot (\mathbf{p} - \mathbf{p}_c(\mathbf{x})), \quad (4.4)$$

or more generally,

$$\nabla_{\mathbf{p}} \mathcal{L} = \nabla_{\mathbf{pp}}^2 \mathcal{L} \cdot (\mathbf{p} - \mathbf{p}_c(\mathbf{x})), \quad (4.5)$$

where $\nabla_{\mathbf{p}} \mathcal{L}$ is the gradient of $\mathcal{L}(\mathbf{x}, \mathbf{p})$ with respect to \mathbf{p} and $\nabla_{\mathbf{pp}}^2 \mathcal{L}$ is the associated Hessian matrix of $\mathcal{L}(\mathbf{x}, \mathbf{p})$.

As a side note, Equation 4.5 has some interesting utilities. If we rearrange our terms to solve for $\mathbf{p}_c(\mathbf{x})$, we get:

$$\mathbf{p}_c(\mathbf{x}) = \mathbf{p} - [\nabla_{\mathbf{pp}}^2 \mathcal{L}]^{-1} \cdot \nabla_{\mathbf{p}} \mathcal{L}. \quad (4.6)$$

If we change \mathbf{p} to be a guess of the optimal value \mathbf{p}_n and $\mathbf{p}_c(\mathbf{x})$ to be our next guess \mathbf{p}_{n+1} , then we get a formula that looks strikingly similar to the Newton-Raphson algorithm for finding the roots of functions:

$$\mathbf{p}_{n+1} = \mathbf{p}_n - [\nabla_{\mathbf{pp}}^2 \mathcal{L}]^{-1} \cdot \nabla_{\mathbf{p}} \mathcal{L}, \quad (4.7)$$

though in this case, we're finding the root of the function $\nabla_{\mathbf{p}} \mathcal{L}$. If we can directly compute the derivative and gradient of our likelihood surface, Equation 4.7 allows us to take a rough

²It should be noted that $\mathcal{L}(\mathbf{x}, \mathbf{p})$ is actually the logarithm of the likelihood, so a quadratic form of $\mathcal{L}(\mathbf{x}, \mathbf{p})$ corresponds to a Gaussian form in the likelihood.

estimate of a local maximum (provided from a differential evolution algorithm) and recursively “polish” the parameter set until it most closely represents a local maximum in the likelihood surface.

Taking the derivative of Equation 4.5 with respect to one of our non-optimized parameters x_k on both sides yields:

$$\frac{\partial}{\partial x_k} \nabla_{\mathbf{p}} \mathcal{L} = \frac{\partial}{\partial x_k} [\nabla_{\mathbf{pp}}^2 \mathcal{L} \cdot (\mathbf{p} - \mathbf{p}_c(\mathbf{x}))] = \left[\frac{\partial}{\partial x_k} \nabla_{\mathbf{pp}}^2 \mathcal{L} \right] \cdot (\mathbf{p} - \mathbf{p}_c(\mathbf{x})) - \nabla_{\mathbf{pp}}^2 \mathcal{L} \cdot \frac{\partial \mathbf{p}_c}{\partial x_k}. \quad (4.8)$$

Evaluating the expression at $\mathbf{p} = \mathbf{p}_c$ gives us:

$$\left. \frac{\partial}{\partial x_k} \nabla_{\mathbf{p}} \mathcal{L} \right|_{\mathbf{p}=\mathbf{p}_c} = - \left. \nabla_{\mathbf{pp}}^2 \mathcal{L} \right|_{\mathbf{p}=\mathbf{p}_c} \cdot \frac{\partial \mathbf{p}_c}{\partial x_k}. \quad (4.9)$$

Reorganizing our equation and generalizing to the entire \mathbf{x} vector finally gives us our desired equation:

$$\nabla_{\mathbf{x}} \mathbf{p}_c = - \left[\left. \nabla_{\mathbf{pp}}^2 \mathcal{L} \right|_{\mathbf{p}=\mathbf{p}_c} \right]^{-1} \left. \nabla_{\mathbf{px}}^2 \mathcal{L} \right|_{\mathbf{p}=\mathbf{p}_c}, \quad (4.10)$$

where

$$(\nabla_{\mathbf{px}}^2 \mathcal{L})_{ij} = \frac{\partial^2 \mathcal{L}}{\partial p_i \partial x_j} \quad (4.11)$$

and $\nabla_{\mathbf{x}} \mathbf{p}_c$ is the Jacobian matrix of the optimal fitted parameters, which we define as the sensitivity matrix:

$$(\nabla_{\mathbf{x}} \mathbf{p}_c)_{ij} = \frac{\partial p_c(\mathbf{x})_i}{\partial x_j}. \quad (4.12)$$

The sensitivity matrix is the derivative object consisting of all the derivatives between the optimal parameter set (the dwarf parameters which yield the highest likelihood score) and the non-fitted parameters. By knowing this matrix, we can determine the direction and magnitude a fitted parameter will shift given a shift in the model parameters, giving us a clear method by which to propagate the uncertainties in our Galactic parameters.

4.2 Shortcomings of Numerical Differentiation

The most common and straightforward method of computing the derivative of a function is to take a numerical approach, using a predetermined step size and actively computing that function at various points (stencils) separated by integer step sizes. The simplest (and crudest) implementation of numerical differentiation on a function $f(x)$ uses the most basic

limit definition of the derivative:

$$\frac{\partial f}{\partial x} \simeq \frac{f(x+h) - f(x)}{h}, \quad (4.13)$$

where h is the step size. Since the derivative is defined as the limit where h approaches zero, it makes sense to assume that using a smaller step size would yield a more accurate result. However, due to limited machine precision, computers become more inaccurate when they have to divide by small numbers or subtract two large but very similar numbers, both crucial operations for a numerical derivative computation. For one to calculate an accurate derivative, one must therefore select a step size small enough to accurately approximate the function as a linear curve, but also large enough to avoid reaching the machine precision of the computer.

For simple well-known functions, finding an appropriate step size is relatively trivial. However, for more complicated and chaotic functions (say an N-body simulation), there is no clearly apparent choice to make without performing a deep analysis. If the function is particularly rough and noisy, the step size must be carefully chosen such that it can resolve the macroscopic features you wish to observe while also ignoring the noisy behavior within the function, placing a lower limit on the size of the optimal step size. However, a step size that is too large not only is inaccurate, but also runs the risk of leaving the features you wish to analyze unresolved. These two opposing features of taking the numerical derivative of a noisy function make finding an optimal step size a delicate balancing act, where larger step sizes must be selected to avoid noise at the cost of accuracy.

One could use a higher order stencil formula to reduce the error term by additional orders of magnitude, but this comes at the expense of requiring more stencil point calculations. This predicament is made readily apparent when trying to estimate the errors in fitted parameters from its Hessian. To estimate the error in a fitted parameter, we use a Cramér-Rao lower bound, which requires us to calculate and invert the Hessian matrix of our likelihood score evaluated at the fitted parameter. Each second-order derivative in the Hessian requires a separate specialized step size for each fitted parameter that must first be loosely estimated beforehand using a marginalized parameter sweep in each dimension (which also requires a tedious $(N-1)$ -dimensional integral to compute). The step size in this context is less an optimally selected value and more an initial guess that will not nec-

essarily provide an accurate Hessian. One could improve the step size recursively by using the estimated errors as the new step size for the error estimation; however, this method is not feasible for our N-body simulations. Using the crudest stencils, an $(N \times N)$ Hessian requires a minimum of $(2N^2 + 1)$ N-body simulations to compute. Depending on the input parameters, our N-body simulations on average take about 15 minutes to generate a likelihood score, though some can take as long as 50 hours. Due to the inherently long run-times required to calculate each likelihood score, optimizing the step size outside of eyeballing the widths of parameters sweeps becomes infeasible, and thus human error invariably impacts our derivative calculations and fitted errors.

4.3 Implementation of Automatic Differentiation

To overcome the ambiguity introduced with the step size, we sidestep the entire issue by implementing automatic differentiation (Wengert 1964) to instead calculate analytic instantaneous derivatives. Just as numerical differentiation draws its basis from the limit definition of the derivative, automatic differentiation draws its basis from a more nuanced property, the chain rule. In a forward accumulation method, each classical float/double object stored in memory also stores its gradient and Hessian information. When we input such an object into an elementary function, we use the analytic derivatives coded into the function, the derivatives of the inputs, and chain rule arithmetic to calculate the gradient and Hessian of the the output float/double object. Since most computer calculations consist of a series of elementary functions, computing the derivatives of the full program boils down to propagating derivative information from function to function and observing the last set of derivatives stored within the output.

We define a structure within the MW@home code called “real”. An object x of type “real” has three attributes: “value”, “gradient”, and “hessian”. The “value” attribute contains a float or double precision number representing the value of x . Defining N as the total number of parameters we wish to differentiate over, the “gradient” attribute is an N -dimensional vector which stores the object’s propagated gradient ∇x , and the “hessian” attribute is an $(N \times N)$ matrix which stores the object’s propagated Hessian $\nabla^2 x$. Within this dissertation, we use the following notation when describing a “real” object z : $z = \langle z, \nabla z, \nabla^2 z \rangle$.

In light of the discoveries regarding the nature of our likelihood surface, which will be

discussed in greater detail in Section 5.1, we added two more attributes to our “real” objects: “Infactor_gradient” and “Infactor_hessian”. These changes were motivated by the fact that our derivative information grew to values that were too large to store linearly within our “real” objects. To tackle this problem, “Infactor_gradient” and “Infactor_hessian” store the natural logarithm of the value with the largest magnitude within the gradient and Hessian respectively, dividing the remaining values by this largest value to more efficiently preserve our derivative information. Therefore, when calling true gradient in our code, we need to ensure that we multiply our gradient by the exponent of “Infactor_gradient”.

For our N-body simulations, if we wanted to calculate the Hessian of the fitted and non-fitted parameters, we would need to differentiate over 5 dwarf progenitor parameters and 5 dwarf orbital parameters. If we also wanted to differentiate over the Galactic parameters, we would need to include 2 parameters for the bulge, 3 for the disk, and another 3 for the halo, bringing the total number of parameters to 18. Including a simplistic Large Magellanic Cloud gives us another 2 parameters (radius and mass), bringing our final tally to $N = 20$. This means every single “real” object would need to store a total of 231 scalars³ and require the same number of calculations to completely update the values of one object, increasing the run-time of our N-body simulation by such a factor. While this seems like a lot, it is still much faster than calculating a Hessian of the same size using a numerical approach, which would require a minimum of 801 N-body simulations. Due to the incredible computational cost of implementing automatic differentiation, running our differential evolution algorithm on an N-body simulation with automatic differentiation would not be feasible. Fortunately though, the derivative information we wish to obtain is not necessary for the optimization process, only for the error analysis of the optimal fitted parameters. Thus, we added a build option that switches the implementation of automatic differentiation on or off.

We define special mathematical functions to take in and transform the attributes of these “real” objects. When “real” objects $x = \langle x, \nabla x, \nabla^2 x \rangle$ and $y = \langle y, \nabla y, \nabla^2 y \rangle$ are passed into a function $z(x, y)$, we evaluate the resulting “real” object $z = \langle z, \nabla z, \nabla^2 z \rangle$

³The actual sum of scalars from the value, gradient, and Hessian is 421, but the symmetric nature of the Hessian allows us to reduce this number to 231 scalars.

using the following forward accumulation formulation from Abate et al. (1997):

$$\begin{aligned}
z &= z(x, y) \\
\nabla z &= \frac{\partial z}{\partial x} \nabla x + \frac{\partial z}{\partial y} \nabla y \\
\nabla^2 z &= \frac{\partial z}{\partial x} \nabla^2 x + \frac{\partial z}{\partial y} \nabla^2 y + \frac{\partial^2 z}{\partial x^2} [\nabla x \otimes \nabla x] + \frac{\partial^2 z}{\partial y^2} [\nabla y \otimes \nabla y] + 2 \frac{\partial^2 z}{\partial x \partial y} [\nabla x \odot \nabla y],
\end{aligned} \tag{4.14}$$

where the derivatives of z are formulas hard-coded into the newly defined function. The first functions we define are the arithmetic operators ($+$, $-$, \times , \div). Since C++ code cannot apply these operators on objects that are incapable of being converted into floats, it is necessary for us to explicitly code in functions that handle these operations⁴. We then include the single-input basic elementary functions (e^x , $\ln(x)$, $\sin(x)$, $\cos(x)$, $\tan(x)$, *etc.*) and their inverses. We also code in the double-valued elementary functions (x^y , $\log_y(x)$, *etc.*) and several other important functions ($|x|$, $\min(x, y)$, $\max(x, y)$, $\lfloor x \rfloor$, $\lceil x \rceil$, *etc.*).

Whenever we declare a constant number in our code (i.e. `real variable = <some number>;`), we store it as a “real” object with a zero gradient and a zero Hessian ($c = \langle c, 0, 0 \rangle$). If the number we define is the i^{th} parameter we wish to differentiate over (like a dwarf or Galactic parameter), then we declare it the same way as a constant number, but we set the i^{th} entry in the object’s gradient vector to 1 ($p = \langle p, e_i, 0 \rangle$). The i^{th} entry within the gradient vector is thus assigned to keep track of the derivatives with respect to that parameter as the code runs. Similarly, the i^{th} row and column in the Hessian thus keep track of all the derivative information for that parameter as well.

4.3.1 Testing Basic Automatic Differentiation

To ensure the derivatives propagated forward were accurate, we created a special test within MilkyWay@home’s testing suite called `autodiff_test`. The test checks each basic function in the mathematical library (addition, multiplication, trigonometric functions, logarithms, *etc.*) by comparing the derivative calculated via automatic differentiation with a derivative calculated using finite difference formulas. When testing first order derivatives,

⁴The functions for adding, subtracting, multiplying, and dividing “real” objects are `mw_add()`, `mw_sub()`, `mw_mul()`, and `mw_div()` respectively.

we compare automatic differentiation's gradient to the following finite difference:

$$\frac{d}{dx}f(x) \approx \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}. \quad (4.15)$$

We use a step size of $h = 0.0001$ for all our numerically computed finite differences. For second order derivatives of the same independent variable, we use the formula:

$$\frac{d^2}{dx^2}f(x) \approx \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2}, \quad (4.16)$$

and for second order derivatives over two different variables, we use the formula:

$$\begin{aligned} \frac{d^2}{dxdy}f(x, y) \approx \frac{1}{144h^2} & (f(x-2h, y-2h) - 8f(x-2h, y-h) + 8f(x-2h, y+h) \\ & - f(x-2h, y+2h) - 8f(x-h, y-2h) + 64f(x-h, y-h) \\ & - 64f(x-h, y+h) + 8f(x-h, y+2h) + 8f(x+h, y-2h) \\ & - 64f(x+h, y-h) + 64f(x+h, y+h) - 8f(x+h, y+2h) \\ & - f(x+2h, y-2h) + 8f(x+2h, y-h) - 8f(x+2h, y+h) \\ & + f(x+2h, y+2h)). \end{aligned} \quad (4.17)$$

If the two calculated derivatives are within 5% of each other, the check passes. For each function, the test checks the equality of each derivative pair over 100,000 randomly selected test points. The test only passes if all checks pass.

Over the course of designing this test, we encountered unique difficulties in determining whether the two derivatives were similar enough. The problem did not arise from improperly implementing the automatic derivatives, but rather from ensuring the numerical derivatives were accurate. In cases where the derivatives were high, such as evaluating the derivative of $\tan(x)$ for x close to $\pi/2$, the formulas for finite difference would suffer in their accuracy. Thus, we were required our test to pull random values only from specially selected ranges that depended on the tested function.

4.4 Non-Classical Derivative Information

Unfortunately for us, implementing automatic differentiation for our N-body simulations is not as simple as redefining our numerical objects and functions. To explain how, we

must first understand “derivative information” and what it means for such information to be “classical” or “non-classical”. We define “derivative information” as the ability to predictably measure the rate of change in the output of a function given a known infinitesimal change in its inputs. If changing the input of a function causes a measurable and predictable change in its output, we say that the function must propagate derivative information.

In order for automatic differentiation to properly calculate the derivative information propagated through a function, that function must have clearly defined derivatives. The elementary functions within our mathematical library have derivatives that are also elementary, so we say that the derivative information propagated through such functions is “classical”.

However, several processes within our code can cause us to lose our derivative information or not properly convey the derivatives in the first place. This in particular occurs when we call a function that does not behave like a traditional mathematical function, such as a random number generator or a Boolean operation. While the derivatives of such functions are either zero or undefined, we know they still propagate finite non-zero derivative information as they have a tangible and measurable effect on the final output state of the simulation. Such derivative information is thus classified as “non-classical”. As these problematic operations are mostly discrete in nature, approximating the propagated derivatives of each one will require us to analyze its continuous analog.

4.4.1 Derivatives from Random Distributions

In order to generate a progenitor dwarf galaxy in our simulations, we randomly generate the position and velocity of each body in our progenitor using a process called rejection-sampling. For each body, we generate a random number to represent the distance between the body and the center of the progenitor (its radius). This radius contains no classical derivative information since it was randomly generated from a uniform distribution. Our code then takes this radius and compares it to a radial probability distribution function by generating another random number between 0 and 1. If this random number is less than the likelihood of the body having its radius, that radius is logged for the body. Otherwise, the code repeats testing a new radius. The velocities of each body are also calculated using rejection-sampling, but we use an energy distribution function to determine their speeds.

The radial distribution function we use is dependent solely on the scale radius of the progenitor. Yet, no classical derivative information reaches the position data. If we were to

slightly increase the scale radius, we would expect to see the distribution of bodies shift to higher radii within the progenitor, which would naturally have a sizeable impact in the final stream shape. Therefore, we must derive a way to estimate the propagated gradient and Hessian of each body's initial position and velocity.

We look at an arbitrary probability distribution function $f(\mathbf{x}, \epsilon)$, where \mathbf{x} is a list of parameters which define the shape of the function, ϵ is the independent variable, and f is both normalized and continuous. We also specify some additional properties of f within this analysis as the random distributions we are interested in also bear these properties:

$$D(f(\mathbf{x}, \epsilon)) : \{\epsilon | 0 \leq \epsilon < \infty\}, \quad (4.18)$$

$$(\forall \epsilon) f(\mathbf{x}, \epsilon) \geq 0, \quad (4.19)$$

$$f(\mathbf{x}, \epsilon) = 0 \implies \epsilon = 0. \quad (4.20)$$

We generate a random distribution of indistinguishable bodies using f at $\mathbf{x} = \mathbf{x}_0$. Since f is a normalized function, it is possible to shift all the bodies distributed under $f(\mathbf{x}_0, \epsilon)$ to follow a new distribution $f(\mathbf{x}_1, \epsilon)$ generated with a different list of parameters $\mathbf{x} = \mathbf{x}_1$. Since the only distinguishing feature of each body in this distribution is its value of ϵ , its transformed ϵ must be a function of its initial ϵ . Therefore, we can define a function $\varepsilon_0(\epsilon, \mathbf{x})$ to describe the body's new location under the new distribution given its old position. Specifically, we define this function such that a body with $\epsilon = \epsilon_0$ in the distribution $f(\mathbf{x}_0, \epsilon)$ transforms to a body with $\epsilon = \varepsilon_0(\epsilon_0, \mathbf{x})$ under the new distribution $f(\mathbf{x}, \epsilon)$. It naturally follows that:

$$\varepsilon_0(\epsilon_0, \mathbf{x}_0) = \epsilon_0. \quad (4.21)$$

While there are a plethora of ways to reorganize the bodies to follow the shifted distribution, we select the way that minimizes the total absolute shift in ϵ for each body. We do this to enforce the uniqueness, continuity, and smoothness of the transformation. Under this restriction, it also follows that bodies that transform in this way never cross each other along the ϵ -axis. That is because if a transformation did cause two bodies to cross each other between the initial and final distributions, there would exist another transformation with a lower total absolute shift in ϵ , which would be a mapping that reverses the final ϵ values of

the two bodies. Therefore,

$$(\forall \epsilon_1, \epsilon_2)(\epsilon_1 < \epsilon_2) \implies (\varepsilon_0(\epsilon_1, \mathbf{x}) < \varepsilon_0(\epsilon_2, \mathbf{x})). \quad (4.22)$$

From this property, we can make one more key observation. Since bodies do not move past each other under this transformation, the total number of bodies with $\epsilon_1 < \epsilon < \epsilon_2$ in the initial distribution $f(\mathbf{x}_0, \epsilon)$ must be the same as the number of bodies with $\varepsilon_0(\epsilon_1, \mathbf{x}) < \epsilon < \varepsilon_0(\epsilon_2, \mathbf{x})$ in the final distribution $f(\mathbf{x}, \epsilon)$. In other words, the area under the distribution curve is invariant under this transformation:

$$\int_{\epsilon_1}^{\epsilon_2} f(\mathbf{x}_0, \epsilon) d\epsilon = \int_{\varepsilon_0(\epsilon_1, \mathbf{x})}^{\varepsilon_0(\epsilon_2, \mathbf{x})} f(\mathbf{x}, \epsilon) d\epsilon. \quad (4.23)$$

Setting $\epsilon_1 = \epsilon$ and $\epsilon_2 = \epsilon + d\epsilon$, we get:

$$\int_{\epsilon}^{\epsilon+d\epsilon} f(\mathbf{x}_0, \epsilon') d\epsilon' = \int_{\varepsilon_0(\epsilon, \mathbf{x})}^{\varepsilon_0(\epsilon+d\epsilon, \mathbf{x})} f(\mathbf{x}, \epsilon') d\epsilon'. \quad (4.24)$$

Evaluating the integrals on both sides and dividing by $d\epsilon$ gives:

$$f(\mathbf{x}_0, \epsilon) = f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}). \quad (4.25)$$

Taking the total derivative on both sides with respect to x_i , where $x_i \in \mathbf{x}$, yields:

$$0 = \frac{d}{dx_i} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) + f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}). \quad (4.26)$$

Evaluating the total derivatives on the right-hand side then gives us:

$$\begin{aligned} 0 &= \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) \\ &\quad + \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) \\ &\quad + f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}). \end{aligned} \quad (4.27)$$

By dividing both sides by $f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))$ and reorganizing our terms, we find:

$$\begin{aligned} \frac{\partial}{\partial \epsilon} \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) + \frac{\partial}{\partial \epsilon} \ln(f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) \\ = - \frac{\partial}{\partial x_i} \ln(f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}). \end{aligned} \quad (4.28)$$

While Equation 4.28 looks unwieldy, we can actually see that it is a first-order linear non-homogeneous differential equation of the form:

$$y'(t) + p(t)y(t) = g(t), \quad (4.29)$$

with solutions of the form:

$$y(t) = e^{-\int p(t)dt} \left[\int e^{\int p(t)dt} g(t)dt + C \right]. \quad (4.30)$$

To use Equation 4.30 as a template for solving Equation 4.28, must first solve for the integrating factor:

$$\int p(t)dt = \int \frac{\partial}{\partial \epsilon} \ln(f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) d\epsilon. \quad (4.31)$$

By performing the substitution $\varepsilon = \varepsilon_0(\epsilon, \mathbf{x})$, we greatly simplify the integral:

$$\int p(t)dt = \int \frac{\partial}{\partial \epsilon} \ln(f(\mathbf{x}, \varepsilon)) d\varepsilon = \ln(f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))). \quad (4.32)$$

Placing this integrating factor into our solution in Equation 4.30, we find:

$$\frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) = - \frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \left[\int \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{\partial}{\partial \epsilon} \varepsilon_0(\epsilon, \mathbf{x}) d\epsilon + C(\mathbf{x}) \right], \quad (4.33)$$

where $C(\mathbf{x})$ is the constant of integration.

In order to solve for this constant, we must supply the differential equation with an initial condition. Consider taking the area under $f(\mathbf{x}_0, \epsilon)$ from $\epsilon = 0$ to $\epsilon = a$. By Equation 4.23, we know that this area is preserved when we change \mathbf{x} :

$$\int_0^a f(\mathbf{x}_0, \epsilon) d\epsilon = \int_{\varepsilon_0(0, \mathbf{x})}^{\varepsilon_0(a, \mathbf{x})} f(\mathbf{x}, \epsilon) d\epsilon. \quad (4.34)$$

Taking the limit as a approaches infinity on both sides yields:

$$\int_0^\infty f(\mathbf{x}_0, \epsilon) d\epsilon = \int_{\varepsilon_0(0, \mathbf{x})}^{\varepsilon_0(\infty, \mathbf{x})} f(\mathbf{x}, \epsilon) d\epsilon. \quad (4.35)$$

Note that because f is a normalized function, the left-hand side of the equation must be 1. However, we also find that $\varepsilon_0(0, \mathbf{x}) = 0$ and $\varepsilon_0(\infty, \mathbf{x}) = \infty$ because if these limits were anything else, then the right-hand side of the equation would be less than 1. Therefore,

$$\frac{d}{dx_i} \varepsilon_0(0, \mathbf{x}) = 0. \quad (4.36)$$

After applying this initial condition to Equation 4.33 and simplifying the integral, we determine:

$$\frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) = -\frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon) d\varepsilon. \quad (4.37)$$

Setting $\mathbf{x} = \mathbf{x}_0$ and generalizing for all derivatives, we find the gradient of a generated body's ϵ value to be:

$$\frac{\partial}{\partial \mathbf{x}} \varepsilon_0(\epsilon, \mathbf{x}_0) = -\frac{1}{f(\mathbf{x}_0, \epsilon)} \int_0^\epsilon \nabla_{\mathbf{x}} f(\mathbf{x}_0, \varepsilon) d\varepsilon. \quad (4.38)$$

To calculate the second-order derivatives of $\varepsilon_0(\epsilon, \mathbf{x})$, we take the total derivative with respect to x_j on both sides of Equation 4.37:

$$\frac{d^2}{dx_i dx_j} \varepsilon_0(\epsilon, \mathbf{x}) = \frac{d}{dx_j} \left[\frac{-1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon) d\varepsilon \right]. \quad (4.39)$$

Applying the product rule to the right-hand-side of the equation yields:

$$\begin{aligned} \frac{d^2}{dx_i dx_j} \varepsilon_0(\epsilon, \mathbf{x}) &= \frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))^2} \left(\frac{\partial}{\partial x_j} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) + \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_j} \varepsilon_0(\epsilon, \mathbf{x}) \right) \\ &\times \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon) d\varepsilon - \frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \frac{d}{dx_j} \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon) d\varepsilon. \end{aligned} \quad (4.40)$$

The second term on the right-hand-side is the derivative of a definite integral whose limits and integrand both depend on x_j . Using some fundamental calculus, it is easy to prove the following statement for any general functions $a(x)$, $b(x)$, and $z(x, y)$:

$$\frac{d}{dx} \int_{a(x)}^{b(x)} z(x, y) dy = \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} z(x, y) dy + z(x, b(x)) \frac{\partial b}{\partial x} - z(x, a(x)) \frac{\partial a}{\partial x}. \quad (4.41)$$

Applying Equation 4.41 to the second term of Equation 4.40, we derive:

$$\begin{aligned}
\frac{d^2}{dx_i dx_j} \varepsilon_0(\epsilon, \mathbf{x}) &= \frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))^2} \left(\frac{\partial}{\partial x_j} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) + \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_j} \varepsilon_0(\epsilon, \mathbf{x}) \right) \\
&\times \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon) d\varepsilon - \frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \\
&\times \left(\int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}, \varepsilon) d\varepsilon + \frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_j} \varepsilon_0(\epsilon, \mathbf{x}) \right). \tag{4.42}
\end{aligned}$$

Substituting Equation 4.37 into Equation 4.42 gives us:

$$\begin{aligned}
\frac{d^2}{dx_i dx_j} \varepsilon_0(\epsilon, \mathbf{x}) &= -\frac{1}{f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x}))} \left(\frac{\partial}{\partial x_i} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_j} \varepsilon_0(\epsilon, \mathbf{x}) \right. \\
&\quad + \frac{\partial}{\partial x_j} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) \\
&\quad + \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \varepsilon_0(\epsilon, \mathbf{x})) \frac{d}{dx_i} \varepsilon_0(\epsilon, \mathbf{x}) \frac{d}{dx_j} \varepsilon_0(\epsilon, \mathbf{x}) \\
&\quad \left. + \int_0^{\varepsilon_0(\epsilon, \mathbf{x})} \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}, \varepsilon) d\varepsilon \right). \tag{4.43}
\end{aligned}$$

Setting $\mathbf{x} = \mathbf{x}_0$ and generalizing for all derivatives, we derive our sought-after Hessian to be:

$$\begin{aligned}
\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}} \varepsilon_0(\epsilon, \mathbf{x}_0) &= -\frac{1}{f(\mathbf{x}_0, \epsilon)} \left(2 \times \nabla_{\mathbf{x}} f(\mathbf{x}_0, \epsilon) \odot \frac{\partial}{\partial \mathbf{x}} \varepsilon_0(\epsilon, \mathbf{x}_0) \right. \\
&\quad + \frac{\partial}{\partial \epsilon} f(\mathbf{x}_0, \epsilon) \times \frac{\partial}{\partial \mathbf{x}} \varepsilon_0(\epsilon, \mathbf{x}_0) \otimes \frac{\partial}{\partial \mathbf{x}} \varepsilon_0(\epsilon, \mathbf{x}_0) \\
&\quad \left. + \int_0^{\epsilon} \nabla_{\mathbf{x}\mathbf{x}}^2 f(\mathbf{x}_0, \varepsilon) d\varepsilon \right). \tag{4.44}
\end{aligned}$$

4.4.1.1 Testing Derivatives of the Radial Distance

MilkyWay@home has the capacity to simulate three types of progenitor dwarf galaxy density profiles: Plummer, Hernquist, and Navarro-Frenk-White (NFW). MilkyWay@home can generate single component progenitors using one of these profiles or pair any two together to form a double component dwarf galaxy, where one profile is used for the distribution of baryonic matter and the other acts as the distribution for dark matter. Because of this complexity and the fact that NFW profiles are inherently divergent, we directly implement the integrals in Equations 4.38 and 4.44 within our code to calculate the necessary derivative information from our dwarf parameters.

Within this subsection and the next, we will analytically compute these derivatives for a simple dwarf progenitor in order to test whether or not our computations are accurate. In particular, we will be using the Plummer distribution because its form and distribution functions are mathematically friendly. When using rejection-sampling to determine radial distance of each body, our code uses the following normalized density distribution:

$$f(r, a) = r^2 \rho(r, a) = \frac{3r^2}{a^3} \left(1 + \frac{r^2}{a^2}\right)^{-5/2}. \quad (4.45)$$

Since the only parameter that defines our distribution is the Plummer scale radius a , we only have one first order derivative of the randomly generated radius (r_s) that we need to compute:

$$\frac{dr_s}{da} = \frac{-1}{f(r_s, a)} \int_0^{r_s} \frac{d}{da} f(r, a) dr. \quad (4.46)$$

Calculating the derivative within the integral, we find:

$$\frac{dr_s}{da} = \frac{-1}{f(r_s, a)} \int_0^{r_s} \frac{3r^2}{a^4} \left(1 + \frac{r^2}{a^2}\right)^{-7/2} \left(2\frac{r^2}{a^2} - 3\right) dr. \quad (4.47)$$

By applying the trigonometric substitution $r = a \tan(\theta)$, our integral becomes:

$$\frac{dr_s}{da} = \frac{-3}{af(r_s, a)} \int_0^{\tan^{-1}(r_s/a)} \tan^2(\theta) \cos^5(\theta) (2 \tan^2(\theta) - 3) d\theta. \quad (4.48)$$

By reorganizing our terms and using some trigonometric identities, we can write our integral as:

$$\frac{dr_s}{da} = \frac{-3}{af(r_s, a)} \int_0^{\tan^{-1}(r_s/a)} (5 \sin^4(\theta) - 3 \sin^2(\theta)) \cos(\theta) d\theta. \quad (4.49)$$

Applying the substitution $u = \sin(\theta)$ gives us:

$$\frac{dr_s}{da} = \frac{-3}{af(r_s, a)} \int_0^{r_s/\sqrt{r_s^2+a^2}} (5u^4 - 3u^2) du, \quad (4.50)$$

and evaluating this integral yields:

$$\frac{dr_s}{da} = \frac{3r_s^3 a}{f(r_s, a) (r_s^2 + a^2)^{5/2}}. \quad (4.51)$$

Substituting in Equation 4.45, we find that our derivative takes on a very simple form:

$$\frac{dr_s}{da} = \frac{r_s}{a}. \quad (4.52)$$

This surprisingly simple answer makes intuitive sense. If the scale radius of the progenitor were to increase, we would expect each body to shift outward at a rate proportional to its distance from the center, similar in principle to cosmic inflation. However, unlike cosmic inflation, the growth is inversely proportional to the scale radius itself. This is because larger scale radii lead to flatter distribution functions since they are normalized.

Since our distribution function is only dependent on a , the Hessian of r_s also only has one term. Using Equation 4.44, we see that the second derivative of r_s has the form:

$$\frac{d^2 r_s}{da^2} = \frac{-1}{f(r_s, a)} \left(2 \frac{\partial}{\partial a} f(r_s, a) \frac{dr_s}{da} + \frac{\partial}{\partial r} f(r_s, a) \left(\frac{dr_s}{da} \right)^2 + \int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr \right). \quad (4.53)$$

Substituting in the derivatives we have already calculated, the second derivative becomes:

$$\begin{aligned} \frac{d^2 r_s}{da^2} = \frac{-1}{f(r_s, a)} & \left(\frac{3r_s^3}{a^5} \left(1 + \frac{r_s^2}{a^2} \right)^{-7/2} \left(4 \frac{r_s^2}{a^2} - 6 \right) \right. \\ & \left. + \frac{\partial}{\partial r} f(r_s, a) \left(\frac{r_s}{a} \right)^2 + \int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr \right). \end{aligned} \quad (4.54)$$

For the other two derivatives, we find:

$$\frac{\partial}{\partial r} f(r_s, a) = \frac{3r_s}{a^3} \left(1 + \frac{r_s^2}{a^2} \right)^{-7/2} \left(2 - 3 \frac{r_s^2}{a^2} \right), \quad (4.55)$$

and

$$\frac{d^2}{da^2} f(r, a) = \frac{3r^2}{a^5} \left(1 + \frac{r^2}{a^2} \right)^{-9/2} \left(2 \frac{r^4}{a^4} - 21 \frac{r^2}{a^2} + 12 \right). \quad (4.56)$$

To solve the integral term within the Hessian, we apply the substitution $r = a \tan(\theta)$:

$$\int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr = \frac{3}{a^2} \int_0^{\tan^{-1}(r_s/a)} \tan^2(\theta) \cos^7(\theta) (2 \tan^4(\theta) - 21 \tan^2(\theta) + 12) d\theta. \quad (4.57)$$

Reorganizing our terms and applying some trigonometric identities, we find:

$$\int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr = \frac{3}{a^2} \int_0^{\tan^{-1}(r_s/a)} (35 \sin^6(\theta) - 45 \sin^4(\theta) + 12 \sin^2(\theta)) \cos(\theta) d\theta. \quad (4.58)$$

Applying another substitution $u = \sin(\theta)$ simplifies the integral to:

$$\int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr = \frac{3}{a^2} \int_0^{r_s/\sqrt{r_s^2+a^2}} (35u^6 - 45u^4 + 12u^2) du. \quad (4.59)$$

Evaluating the integral yields:

$$\int_0^{r_s} \frac{d^2}{da^2} f(r, a) dr = \frac{3r_s^3}{a^5} \left(1 + \frac{r_s^2}{a^2}\right)^{-7/2} \left(4 - \frac{r_s^2}{a^2}\right). \quad (4.60)$$

Plugging Equations 4.55 and 4.60 into Equation 4.53 reveals a striking yet simple solution:

$$\frac{d^2 r_s}{da^2} = \frac{-1}{f(r_s, a)} \frac{3r_s^3}{a^5} \left(1 + \frac{r_s^2}{a^2}\right)^{-7/2} \left(4 \frac{r_s^2}{a^2} - 6 + 2 - 3 \frac{r_s^2}{a^2} + 4 - \frac{r_s^2}{a^2}\right) = 0. \quad (4.61)$$

This fact can also be easily seen by taking the derivative of Equation 4.52:

$$\frac{d^2 r_s}{da^2} = \frac{d}{da} \frac{dr_s}{da} = \frac{d}{da} \left(\frac{r_s}{a}\right) = \frac{1}{a} \frac{dr_s}{da} - \frac{r_s}{a^2} = \frac{r_s}{a^2} - \frac{r_s}{a^2} = 0. \quad (4.62)$$

4.4.1.2 *Testing Derivatives of the Velocity*

The velocity distribution for most progenitors is, in most cases, too complicated for us to model analytically, even for systems with spherical symmetry. The energy distribution for such systems is given by Binney & Tremaine (2008):

$$f(\epsilon) = \frac{1}{\sqrt{8\pi^2}} \int_0^\epsilon \frac{d\Psi}{\sqrt{\epsilon - \Psi}} \frac{d^2 \rho}{d\Psi^2}, \quad (4.63)$$

where $\epsilon = -\frac{1}{2}v^2 - \Phi(r)$, $\Psi = \Phi(r)$, and $\Phi(r)$ is the gravitational potential of the spherically symmetric system. The single component Plummer sphere, however, has a very nice and simple analytic form:

$$f(\epsilon) = \frac{24\sqrt{2}}{7\pi^3} \frac{a^2}{M^4} \epsilon^{7/2}. \quad (4.64)$$

Therefore, for a given radial distance r_s , the velocity distribution we apply rejection-sampling to has the form:

$$f(v) = 4\pi v^2 \times \frac{24\sqrt{2}}{7\pi^3} \frac{a^2}{M^4} \epsilon(v)^{7/2} = \frac{96\sqrt{2}}{7\pi^2} \frac{a^2}{M^4} v^2 \left(\frac{M}{\sqrt{r_s^2 + a^2}} - \frac{1}{2}v^2 \right)^{7/2}. \quad (4.65)$$

However, since the gradient and Hessian calculations of our randomly selected velocity v_s require normalized distributions, we must first calculate the normalization constant A :

$$A = \int_0^{v_{esc}} \frac{96\sqrt{2}}{7\pi^2} \frac{a^2}{M^4} v^2 \left(\frac{M}{\sqrt{r_s^2 + a^2}} - \frac{1}{2}v^2 \right)^{7/2} dv, \quad (4.66)$$

where $v_{esc} = \sqrt{\frac{2M}{\sqrt{r_s^2 + a^2}}}$ is the escape velocity at a given radius r . Reorganizing terms, we see:

$$A = \int_0^{v_{esc}} \frac{96\sqrt{2}}{7\pi^2} \frac{a^2}{\sqrt{M}} (r_s^2 + a^2)^{-7/4} v^2 \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2 \right)^{7/2} dv. \quad (4.67)$$

Applying the substitution $v = \sqrt{\frac{2M}{\sqrt{r_s^2 + a^2}}} \sin(\theta)$, we get:

$$A = \frac{384}{7\pi^2} a^2 M (r_s^2 + a^2)^{-5/2} \cdot \int_0^{\pi/2} \sin^2(\theta) \cos^8(\theta) d\theta. \quad (4.68)$$

Evaluating this integral gives us the normalization of our velocity distribution:

$$A = \frac{3}{4\pi} a^2 M (r_s^2 + a^2)^{-5/2}, \quad (4.69)$$

which turns out exactly to be our Plummer sphere density profile.

It is no coincidence that our normalization constant became this familiar density profile, and it hints at a much deeper relationship. The formula provided by Binney & Tremaine (2008) in Equation 4.63 is the 6D phase space distribution of positions and velocities in a spherical, gravitationally stable system. If one were to integrate the distribution function over all possible velocities, as we have done in calculating our normalization constant, we would end up with a function that represents the distribution of positions in the system, its spatial density profile. Therefore, for a given radial distance, we find that the normalized

velocity distribution our rejection-sampling sees is:

$$\tilde{f}(v, r_s, a, M) = \frac{128\sqrt{2}}{7\pi} \left(\frac{\sqrt{r_s^2 + a^2}}{M} \right)^{3/2} v^2 \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2 \right)^{7/2}. \quad (4.70)$$

Our velocity distribution is dependent on two variables, the Plummer scale radius a and the total mass M . This gives us a total of five derivatives we can calculate and test with. We use Equation 4.38 to set up the first derivative of v_s with respect to a :

$$\frac{dv_s}{da} = \frac{-1}{\tilde{f}(v_s, r_s, a, M)} \int_0^{v_s} \frac{d}{da} \tilde{f}(v, r_s, a, M) dv. \quad (4.71)$$

One important thing we have to remember is that v_s is also a function of r_s , which we know is dependent on a . Therefore, we need to take this into consideration when computing our derivative:

$$\frac{d}{da} \tilde{f}(v, r_s, a, M) = \frac{\partial}{\partial a} \tilde{f}(v, r_s, a, M) + \frac{\partial}{\partial r} \tilde{f}(v, r_s, a, M) \frac{dr_s}{da}. \quad (4.72)$$

Since

$$\begin{aligned} \frac{\partial}{\partial r} \tilde{f}(v, r_s, a, M) &= \frac{64\sqrt{2}}{7\pi} \frac{r_s v^2}{M^{3/2} (r_s^2 + a^2)^{1/4}} \\ &\times \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2 \right)^{5/2} \left(3 - 5 \frac{\sqrt{r_s^2 + a^2}}{M} v^2 \right), \end{aligned} \quad (4.73)$$

and

$$\begin{aligned} \frac{\partial}{\partial a} \tilde{f}(v, r_s, a, M) &= \frac{64\sqrt{2}}{7\pi} \frac{av^2}{M^{3/2} (r_s^2 + a^2)^{1/4}} \\ &\times \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2 \right)^{5/2} \left(3 - 5 \frac{\sqrt{r_s^2 + a^2}}{M} v^2 \right), \end{aligned} \quad (4.74)$$

we can show using Equation 4.52 that:

$$\begin{aligned} \frac{d}{da} \tilde{f}(v, r_s, a, M) &= \frac{64\sqrt{2}}{7\pi} \frac{(r_s^2 + a^2)^{3/4} v^2}{aM^{3/2}} \\ &\times \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2 \right)^{5/2} \left(3 - 5 \frac{\sqrt{r_s^2 + a^2}}{M} v^2 \right). \end{aligned} \quad (4.75)$$

To further simplify this equation, we substitute in the escape velocity $v_{esc} = \sqrt{\frac{2M}{\sqrt{r_s^2 + a^2}}}$:

$$\frac{d}{da} \tilde{f}(v, r_s, a, M) = \frac{256}{7\pi a v_{esc}^3} \left(1 - \frac{v^2}{v_{esc}^2}\right)^{5/2} \left(3 - 10 \frac{v^2}{v_{esc}^2}\right). \quad (4.76)$$

Plugging this into our integral gives:

$$\frac{dv_s}{da} = -\frac{256}{7\pi a v_{esc}^3 \tilde{f}(v_s, r_s, a, M)} \int_0^{v_s} v^2 \left(1 - \frac{v^2}{v_{esc}^2}\right)^{5/2} \left(3 - 10 \frac{v^2}{v_{esc}^2}\right) dv. \quad (4.77)$$

By using the substitution $v = v_{esc} \sin(\theta)$, we can simplify our integral to:

$$\frac{dv_s}{da} = -\frac{256}{7\pi a \tilde{f}(v_s, r_s, a, M)} \int_0^{\sin^{-1}(v_s/v_{esc})} \sin^2(\theta) \cos^6(\theta) (3 - 10 \sin^2(\theta)) d\theta. \quad (4.78)$$

Evaluating this integral, we find:

$$\frac{dv_s}{da} = -\frac{256}{7\pi a \tilde{f}(v_s, r_s, a, M)} \left(\frac{v_s}{v_{esc}}\right)^3 \left(1 - \frac{v_s^2}{v_{esc}^2}\right)^{7/2}. \quad (4.79)$$

Plugging in our normalized distribution function into the equation, we find a surprisingly simple answer for our derivative:

$$\frac{dv_s}{da} = -\frac{v_s}{2a}. \quad (4.80)$$

The next derivative we will calculate is the the first derivative of v_s with respect to M :

$$\frac{dv_s}{dM} = \frac{-1}{\tilde{f}(v_s, r_s, a, M)} \int_0^{v_s} \frac{d}{dM} \tilde{f}(v, r_s, a, M) dv. \quad (4.81)$$

Thankfully, r_s is not a function of M for this case, so we can more easily evaluate our derivatives. We can write our derivative with respect to the mass M as:

$$\begin{aligned} \frac{d}{dM} \tilde{f}(v, r_s, a, M) &= -\frac{64\sqrt{2} (r_s^2 + a^2)^{3/4} v^2}{7\pi M^{5/2}} \\ &\times \left(1 - \frac{\sqrt{r_s^2 + a^2}}{2M} v^2\right)^{5/2} \left(3 - 5 \frac{\sqrt{r_s^2 + a^2}}{M} v^2\right), \end{aligned} \quad (4.82)$$

or in terms of the escape velocity v_{esc} :

$$\frac{d}{dM} \tilde{f}(v, r_s, a, M) = -\frac{256}{7\pi M} \frac{v^2}{v_{esc}^3} \left(1 - \frac{v^2}{v_{esc}^2}\right)^{5/2} \left(3 - 10 \frac{v^2}{v_{esc}^2}\right), \quad (4.83)$$

Plugging this derivative into our integral yields:

$$\frac{dv_s}{dM} = \frac{256}{7\pi M v_{esc}^3} \tilde{f}(v_s, r_s, a, M) \int_0^{v_s} v^2 \left(1 - \frac{v^2}{v_{esc}^2}\right)^{5/2} \left(3 - 10 \frac{v^2}{v_{esc}^2}\right) dv. \quad (4.84)$$

We see that the integral above is the same integral in Equation 4.77, which we have already solved. We can therefore show that our mass derivative of v_s also has a very simple form:

$$\frac{dv_s}{dM} = \frac{v_s}{2M}. \quad (4.85)$$

Knowing these two derivatives, we can very easily calculate all the necessary second-order derivatives for our tests using rudimentary calculus:

$$\frac{d^2 v_s}{da^2} = \frac{d}{da} \frac{dv_s}{da} = \frac{d}{da} \left[-\frac{v_s}{2a} \right] = -\left[\frac{dv_s}{da} \frac{1}{2a} - \frac{v_s}{2a^2} \right] = \frac{v_s}{4a^2} + \frac{v_s}{2a^2}, \quad (4.86)$$

$$\frac{d^2 v_s}{dadM} = \frac{d}{da} \frac{dv_s}{dM} = \frac{d}{da} \left[\frac{v_s}{2M} \right] = \frac{1}{2M} \frac{dv_s}{da}, \quad (4.87)$$

$$\frac{d^2 v_s}{dM^2} = \frac{d}{dM} \frac{dv_s}{dM} = \frac{d}{dM} \left[\frac{v_s}{2M} \right] = \frac{dv_s}{dM} \frac{1}{2M} - \frac{v_s}{2M^2} = \frac{v_s}{4M^2} - \frac{v_s}{2M^2}. \quad (4.88)$$

We therefore show that our the Hessian of v_s takes the following values:

$$\frac{d^2 v_s}{da^2} = \frac{3v_s}{4a^2}, \quad (4.89)$$

$$\frac{d^2 v_s}{dadM} = -\frac{v_s}{4Ma}, \quad (4.90)$$

$$\frac{d^2 v_s}{dM^2} = -\frac{v_s}{4M^2}. \quad (4.91)$$

4.4.1.3 Derivatives of Root-Finding Algorithms

In order to properly and accurately calculate the derivatives above using Equations 4.38 and 4.44, we are required to calculate the gradient and Hessian of the normalized distribution function. We were intending to use simple numerical derivatives to quickly calculate

these derivatives, but we soon discovered that this method was not accurate enough to pass the basic tests listed in the last two sections, requiring several step sizes to be optimized simultaneously. To circumvent this, we decided to implement automatic differentiation for this purpose as well. For the radial positions, there were no complications. However, the velocities distributions once again proved to be difficult to work with.

The main source of complication arises from its distribution function. As it requires a complicated integration for anything not a Plummer sphere, we must numerically approximate it, necessitating the accurate determination of its upper and lower bounds of integration. While its lower bound is a straightforward calculation, the upper bound requires that we find the radius at which the total energy of a gravitationally bound test body is equal to its gravitational potential energy (i.e. its kinetic energy is zero). To solve this, our code employs a bisection algorithm to quickly find the root of the function. However, this type of algorithm does not propagate derivative information, and since the root we calculate must contain information about the mass and radius of the dwarf galaxy, we must again find a way to artificially insert to correct derivative information into the code.

Assume we have an analytic function $f(\mathbf{x}, \epsilon)$ with a root $\epsilon_0(\mathbf{x})$, where \mathbf{x} is the list of parameters that define the function. We know that such a root must have an inherent dependence on \mathbf{x} . We therefore know that for values of ϵ close to $\epsilon_0(\mathbf{x})$, our function can be written as:

$$f(\mathbf{x}, \epsilon) = \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x}))(\epsilon - \epsilon_0(\mathbf{x})) + \frac{1}{2} \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x}))(\epsilon - \epsilon_0(\mathbf{x}))^2 + \dots \quad (4.92)$$

taking the derivative of both sides with respect to one of the parameters x_i , we find:

$$\begin{aligned} \frac{d}{dx_i} f(\mathbf{x}, \epsilon) &= (\epsilon - \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\ &\quad - \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \\ &\quad + \frac{1}{2} (\epsilon - \epsilon_0(\mathbf{x}))^2 \frac{d}{dx_i} \left[\frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\ &\quad - (\epsilon - \epsilon_0(\mathbf{x})) \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) + \dots \end{aligned} \quad (4.93)$$

By evaluating both sides of the equation at $\epsilon = \epsilon_0(\mathbf{x})$ gives us:

$$\frac{\partial}{\partial x_i} f(\mathbf{x}, \epsilon_0(\mathbf{x})) = -\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}). \quad (4.94)$$

Rearranging terms, we get the gradient of the root:

$$\frac{d}{dx_i} \epsilon_0(\mathbf{x}) = -\frac{\frac{\partial}{\partial x_i} f(\mathbf{x}, \epsilon_0(\mathbf{x}))}{\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x}))} \quad (4.95)$$

The Hessian can be determined from taking the derivative of Equation 4.93 with respect to the parameter x_j :

$$\begin{aligned} \frac{d^2}{dx_i dx_j} f(\mathbf{x}, \epsilon) = \frac{d}{dx_j} \left[(\epsilon - \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \right. \\ \left. - \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \right. \\ \left. + \frac{1}{2} (\epsilon - \epsilon_0(\mathbf{x}))^2 \frac{d}{dx_i} \left[\frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \right. \\ \left. - (\epsilon - \epsilon_0(\mathbf{x})) \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) + \dots \right]. \end{aligned} \quad (4.96)$$

By evaluating this derivative into each term, we expand the equation as such: The Hessian can be determined from taking the derivative of Equation 4.93 with respect to the parameter

x_j :

$$\begin{aligned}
\frac{d^2}{dx_i dx_j} f(\mathbf{x}, \epsilon) &= -\frac{d}{dx_j} \epsilon_0(\mathbf{x}) \frac{d}{dx_i} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\
&+ (\epsilon - \epsilon_0(\mathbf{x})) \frac{d^2}{dx_i dx_j} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\
&- \frac{d}{dx_j} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \frac{d}{dx_i} \epsilon_0(\mathbf{x}) - \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d^2}{dx_i dx_j} \epsilon_0(\mathbf{x}) \\
&- (\epsilon - \epsilon_0(\mathbf{x})) \frac{d}{dx_j} \epsilon_0(\mathbf{x}) \frac{d}{dx_i} \left[\frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\
&+ \frac{1}{2} (\epsilon - \epsilon_0(\mathbf{x}))^2 \frac{d^2}{dx_i dx_j} \left[\frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \\
&+ \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \frac{d}{dx_j} \epsilon_0(\mathbf{x}) \\
&- (\epsilon - \epsilon_0(\mathbf{x})) \frac{d}{dx_j} \left[\frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \\
&- (\epsilon - \epsilon_0(\mathbf{x})) \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d^2}{dx_i dx_j} \epsilon_0(\mathbf{x}) + \dots .
\end{aligned} \tag{4.97}$$

Setting $\epsilon = \epsilon_0(\mathbf{x})$, we find:

$$\begin{aligned}
\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}, \epsilon_0(\mathbf{x})) &= -\frac{d}{dx_j} \epsilon_0(\mathbf{x}) \frac{d}{dx_i} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] - \frac{d}{dx_j} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \\
&- \frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d^2}{dx_i dx_j} \epsilon_0(\mathbf{x}) + \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \frac{d}{dx_j} \epsilon_0(\mathbf{x}).
\end{aligned} \tag{4.98}$$

It is easy to show that:

$$\frac{d}{dx_i} \left[\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right] = \frac{\partial^2}{\partial \epsilon \partial x_i} f(\mathbf{x}, \epsilon_0(\mathbf{x})) + \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}). \tag{4.99}$$

We can therefore write the Hessian of our root as:

$$\begin{aligned}
\frac{d^2}{dx_i dx_j} \epsilon_0(\mathbf{x}) &= \frac{-1}{\frac{\partial}{\partial \epsilon} f(\mathbf{x}, \epsilon_0(\mathbf{x}))} \left(\frac{\partial^2}{\partial \epsilon \partial x_i} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_j} \epsilon_0(\mathbf{x}) \right. \\
&+ \frac{\partial^2}{\partial \epsilon \partial x_j} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \\
&+ \frac{\partial^2}{\partial \epsilon^2} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \frac{d}{dx_i} \epsilon_0(\mathbf{x}) \frac{d}{dx_j} \epsilon_0(\mathbf{x}) \\
&\left. + \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}, \epsilon_0(\mathbf{x})) \right).
\end{aligned} \tag{4.100}$$

4.4.2 Derivatives from Binning

Another place where our derivative information is classically unaccounted for is in the case of counting and binning. To calculate our likelihood, we perform an Earth Mover Distance calculation to see how similar the shape of our simulated tidal stream matches to the input data stream. The shape of the simulated stream is measured by creating a histogram, binning the simulated bodies in Λ and B . However, this process destroys the classical derivative information propagated by each body. To illustrate this, imagine a body with (Λ, B) coordinates of (Λ_0, B_0) , which contain derivative information. To account for this body in our likelihood calculation, our code must determine which (Λ, B) -bin must be incremented by one count. Calculating the increase in the counts for a (Λ, B) -bin from this body is equivalent to passing (Λ_0, B_0) through a finite linear combination of Heaviside functions. However, since the derivative of the Heaviside function is the Dirac delta function, no meaningful derivative information can be efficiently computed or propagated.

To better approximate the derivatives in the counts, we forgo an integer representation. Instead of assigning a body to a single bin, we “split” the body into fractional pieces, placing larger pieces in bins where the body is more likely to be placed. For example, if a body’s coordinates would place it very close to a bin’s boundary, the largest piece would be placed in the bin it occupies, but a similarly large piece would be added to the adjacent bin. The most intuitive way to assign the fractional pieces to bins is to transform the point-like coordinates of the body into a probability distribution across the entire sky. One sadly cannot determine the shape of a probability distribution from the location of its mean value alone. However, the sky positions of each body in our N-body simulations have an inherent resolution limit determined by two factors: the softening length and the timestep.

Since the bodies in an N-body simulation do not represent individual stars, but rather enormous clusters of mass, there exists a scale length at which the close interactions of bodies would no longer reasonably reflect reality. This scale length is implemented as the softening length (l_s), the distance by which the force between two bodies is deadened. Therefore, we would expect the positions of each body in our simulation to have an inherent uncertainty on the order of l_s . If a body has a heliocentric radius of r_H , the inherent isotropic angular uncertainty in the sky position from the softening length (σ_{l_s}) is:

$$\sigma_{l_s} = \frac{l_s}{r_h}. \quad (4.101)$$

The timestep also imposes a limit on how well we can know a body's position. The timestep (τ) is the smallest unit of time we can resolve when performing an N-body simulation. If a body has a velocity, then the uncertainty of the position in the direction of motion of the body is given by the velocity multiplied by the timestep. Translating to sky positions, the uncertainty in the sky position from the timestep σ_τ is:

$$\sigma_\tau = |\boldsymbol{\mu}|\tau, \quad (4.102)$$

where $\boldsymbol{\mu}$ is the proper motion of the body. It should be noted, though, that unlike σ_{l_s} , σ_τ is not isotropic, only following the direction of $\boldsymbol{\mu}$. We can therefore characterize the total uncertainty in the sky position as an ellipse in the sky with a semi-major axis (a) and semi-minor axis (b) given by:

$$a = \sqrt{|\boldsymbol{\mu}|^2\tau^2 + \left(\frac{l_s}{r_h}\right)^2}, \quad (4.103)$$

$$b = \frac{l_s}{r_h}. \quad (4.104)$$

While it would be simple to translate this ellipse into a bivariate Gaussian distribution, such a distribution is unfortunately not appropriate for the system. This is because a bivariate Gaussian distribution has a domain that covers \mathbb{R}^2 , whereas the distribution we need must be completely normalized over the surface of a unit sphere. Therefore, we instead require a 5-parameter Fisher-Bingham distribution (FB₅), also known as the Kent distribution (Kent 1982). FB₅ has the form:

$$f(\mathbf{x}) = \frac{1}{c(\kappa, \beta)} e^{(\kappa\boldsymbol{\gamma}_1 \cdot \mathbf{x}) + \beta[(\boldsymbol{\gamma}_2 \cdot \mathbf{x})^2 - (\boldsymbol{\gamma}_3 \cdot \mathbf{x})^2]}, \quad (4.105)$$

where \mathbf{x} is a vector on the unit sphere, κ and β are constants that describe the shape of the distribution, $c(\kappa, \beta)$ is the normalization constant, and the three vectors ($\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$) are 3-dimensional orthonormal vectors that describe the position and orientation of the distribution. Since we want a distribution mapped in (Λ, B) coordinates, we define our Cartesian system in \mathbf{x} such that:

$$\mathbf{x} = \cos(B) \cos(\Lambda) \hat{i} + \cos(B) \sin(\Lambda) \hat{j} + \sin(B) \hat{k}. \quad (4.106)$$

According to Kent (1982), γ_1 is the unit vector that points to the mean value of the distribution. If we set the position of our body (Λ_0, B_0) as the mean, we trivially find:

$$\gamma_1 = \cos(B_0) \cos(\Lambda_0) \hat{i} + \cos(B_0) \sin(\Lambda_0) \hat{j} + \sin(B_0) \hat{k}. \quad (4.107)$$

γ_2 is the unit vector that points in the same direction as the semi-major axis of the distribution. Since this direction is the same as the proper motion of the body $\boldsymbol{\mu}$, we derive:

$$\begin{aligned} \gamma_2 = \frac{1}{|\boldsymbol{\mu}|} \Big[& (-\mu_\Lambda \sin(\Lambda_0) - \mu_B \cos(\Lambda_0) \sin(B_0)) \hat{i} \\ & + (\mu_\Lambda \cos(\Lambda_0) - \mu_B \sin(\Lambda_0) \sin(B_0)) \hat{j} \\ & + \mu_B \cos(B_0) \hat{k} \Big], \end{aligned} \quad (4.108)$$

where $\mu_\Lambda = \boldsymbol{\mu} \cdot \hat{\Lambda}$ and $\mu_B = \boldsymbol{\mu} \cdot \hat{B}$. γ_3 is the unit vector that points in the same direction as the semi-minor axis and can be easily calculated by taking the cross product between γ_1 and γ_2 :

$$\gamma_3 = \gamma_1 \times \gamma_2. \quad (4.109)$$

Determining the constants κ and β for our distribution is a much more difficult task. According to Kent (1982), κ represents the “concentration” of the distribution near the mean while β is a measure of the distribution’s “ovalness”. However, there exists no analytic way to derive these constants even when given a population to fit. Thankfully for us, there exists a way to closely approximate these parameters in the case of large κ (Kent 1982). We should satisfy this condition since we expect our ellipse of uncertainty to subtend only a very small solid angle on the unit sphere. Assuming large κ , the moment estimates of $\tilde{\kappa}$ and $\tilde{\beta}$ of a population set $\{\mathbf{y}\}$ are given by:

$$\tilde{\kappa} \cong \frac{1}{2 - 2r_1 - r_2} + \frac{1}{2 - 2r_1 + r_2}, \quad (4.110)$$

$$\tilde{\beta} \cong \frac{1}{2} \left(\frac{1}{2 - 2r_1 - r_2} - \frac{1}{2 - 2r_1 + r_2} \right), \quad (4.111)$$

where

$$r_1 = E[\gamma_1 \cdot \mathbf{y}], \quad (4.112)$$

$$r_2 = E[(\gamma_2 \cdot \mathbf{y})^2 - (\gamma_3 \cdot \mathbf{y})^2], \quad (4.113)$$

and $E[g(\mathbf{y})]$ is the expectation value of $g(\mathbf{y})$.

To calculate the moment estimates of $\tilde{\kappa}$ and $\tilde{\beta}$, we assume our population set $\{\mathbf{y}\}$ to be uniform ellipse on the unit sphere whose semi-major (a) and semi-minor (b) axes follow those in Equations 4.103 and 4.104, respectively. We also further assume $0 \leq b \leq a < \pi/2$. We rotate our unit sphere in Cartesian space such that the center of our ellipse is on the z-axis and the semi-major axis follows the x-axis, transforming $\{\mathbf{y}\}$ to $\{\mathbf{y}'\}$. For easier computation, we parameterize $\{\mathbf{y}'\}$ in spherical coordinates (θ, ϕ) :

$$\mathbf{y}'(\theta, \phi) = \sin(\theta) \cos(\phi) \hat{i} + \sin(\theta) \sin(\phi) \hat{j} + \cos(\theta) \hat{k}. \quad (4.114)$$

The focal points of this ellipse lie at $\mathbf{y}'(d, 0)$ and $\mathbf{y}'(d, \pi)$, where:

$$\cos(d) = \frac{\cos(a)}{\cos(b)}. \quad (4.115)$$

We can therefore say that $\mathbf{y}'(\theta, \phi)$ lies on the boundary if it satisfies the following equation:

$$\cos^{-1}(\mathbf{y}'(\theta, \phi) \cdot \mathbf{y}'(d, 0)) + \cos^{-1}(\mathbf{y}'(\theta, \phi) \cdot \mathbf{y}'(d, \pi)) = 2a. \quad (4.116)$$

Evaluating the dot products yields:

$$\begin{aligned} & \cos^{-1}(\cos(d) \cos(\theta) + \sin(d) \sin(\theta) \cos(\phi)) \\ & + \cos^{-1}(\cos(d) \cos(\theta) - \sin(d) \sin(\theta) \cos(\phi)) = 2a. \end{aligned} \quad (4.117)$$

To simplify the math, we substitute $s = \cos(d) \cos(\theta)$ and $t = \sin(d) \sin(\theta) \cos(\phi)$ into the equation:

$$\cos^{-1}(s + t) + \cos^{-1}(s - t) = 2a. \quad (4.118)$$

Taking the cosine of both sides and using the cosine angle sum identity yields:

$$\cos(\cos^{-1}(s + t)) \cos(\cos^{-1}(s - t)) - \sin(\cos^{-1}(s + t)) \sin(\cos^{-1}(s - t)) = \cos(2a), \quad (4.119)$$

which simplifies to:

$$s^2 - t^2 - \sqrt{1 - 2(s^2 + t^2) + (s^2 - t^2)^2} = \cos(2a). \quad (4.120)$$

Reorganizing the terms and simplifying further gets us:

$$\cos^2(2a) - 2(s^2 - t^2) \cos(2a) = 1 - 2(s^2 + t^2). \quad (4.121)$$

Our equation for the boundary of the ellipse is now represented in terms of the sum and difference of s^2 and t^2 . It can easily be shown using trigonometric identities that:

$$s^2 \pm t^2 = [\cos^2(d) \mp \sin^2(d) \cos^2(\phi)] \cos^2(\theta) \pm \sin^2(d) \cos^2(\phi). \quad (4.122)$$

By substituting Equation 4.122 into Equation 4.121, we find:

$$\begin{aligned} \cos^2(2a) - 2([\cos^2(d) + \sin^2(d) \cos^2(\phi)] \cos^2(\theta) - \sin^2(d) \cos^2(\phi)) \cos(2a) \\ = 1 - 2([\cos^2(d) - \sin^2(d) \cos^2(\phi)] \cos^2(\theta) + \sin^2(d) \cos^2(\phi)). \end{aligned} \quad (4.123)$$

Solving for $\cos^2(\theta)$ we find:

$$\cos^2(\theta) = \frac{\sin^2(2a) - 2\sin^2(d) \cos^2(\phi)(1 + \cos(2a))}{2[(1 - \cos(2a)) \cos^2(d) - (1 + \cos(2a)) \sin^2(d) \cos^2(\phi)]}. \quad (4.124)$$

After applying the double angle identities for sine and cosine, we derive the parameterization of the ellipse's boundary on the unit sphere:

$$\theta(\phi) = \cos^{-1} \left(\sqrt{\frac{\sin^2(a) \csc^2(d) - \cos^2(\phi)}{\tan^2(a) \cot^2(d) - \cos^2(\phi)}} \right). \quad (4.125)$$

We substitute $p = \sin(a) \csc(d)$ and $q = \tan(a) \cot(d)$ to further simplify our equation:

$$\theta(\phi) = \cos^{-1} \left(\sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} \right). \quad (4.126)$$

Given this equation of the ellipse, the expectation value of $g(\mathbf{y}')$ takes the following form:

$$E[g(\mathbf{y}')] = \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} g(\mathbf{y}'(\theta, \phi)) \sin(\theta) d\theta d\phi, \quad (4.127)$$

where Ω is the solid angle subtended by the ellipse $\{\mathbf{y}'\}$. The formula for Ω is given by:

$$\Omega = \int_0^{2\pi} \int_0^{\theta(\phi)} \sin(\theta) d\theta d\phi = \int_0^{2\pi} \int_{\cos(\theta(\phi))}^1 d(\cos(\theta)) d\phi. \quad (4.128)$$

Evaluating the first integral and substituting in Equation 4.126, we find:

$$\Omega = 2\pi - \int_0^{2\pi} \sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} d\phi. \quad (4.129)$$

Since $\cos(\phi)$ is symmetric about $\phi = \pi$, we can simplify our formula to:

$$\Omega = 2\pi - 2 \int_0^\pi \sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} d\phi. \quad (4.130)$$

Substituting $x = \cos(\phi)$, our solid angle becomes:

$$\Omega = 2\pi - 2 \int_{-1}^1 \sqrt{\frac{p^2 - x^2}{(q^2 - x^2)(1 - x^2)}} dx. \quad (4.131)$$

Since our integrand is even, we can further simplify our integral to:

$$\Omega = 2\pi - 4 \int_0^1 \sqrt{\frac{p^2 - x^2}{(q^2 - x^2)(1 - x^2)}} dx. \quad (4.132)$$

Unfortunately, this integral does not have an elementary solution. However, we can represent this integral as an elliptic integral. It can be shown that our solid angle formula has the solution:

$$\Omega = 2\pi - 4 \frac{p^2}{q\sqrt{p^2 - 1}} \Pi \left(\frac{1}{1 - p^2} \middle| \frac{q^2 - p^2}{q^2(1 - p^2)} \right), \quad (4.133)$$

where $\Pi(n|m)$ is the complete elliptic integral of the third kind. It can also be shown that:

$$p^2 = \frac{\sin^2(a) \cos^2(b)}{\sin^2(a) - \sin^2(b)} \quad (4.134)$$

and

$$q^2 = \frac{\sin^2(a)}{\sin^2(a) - \sin^2(b)}. \quad (4.135)$$

Therefore,

$$\Omega = 2\pi - 4 \frac{\sin(a) \cos^2(b)}{\sin(b) \cos(a)} \Pi \left(\frac{\sin^2(b) - \sin^2(a)}{\sin^2(b) \cos^2(a)} \middle| \frac{\sin^2(b) - \sin^2(a)}{\cos^2(a)} \right). \quad (4.136)$$

Elliptic integrals are tedious to compute, especially those of the third kind. However, since we anticipate a and b to be small angles, we can try to approximate the value of this elliptic integral. In our small angle limit, we see that the m term in $\Pi(n|m)$ approaches zero. For $|m| \ll 1$,

$$\Pi(n|m) \approx \frac{\pi}{2\sqrt{1-n}} + \frac{\pi m}{4n} \left(\frac{1}{\sqrt{1-n}} - 1 \right). \quad (4.137)$$

Therefore, we can quickly and easily approximate our solid angle Ω to be:

$$\Omega \approx 2\pi \left(1 - \cos(b) \left(1 - \frac{1}{2} \sin^2(b) \left(\frac{\tan(a)}{\tan(b)} - 1 \right) \right) \right). \quad (4.138)$$

Taking the small angle approximation of Equation 4.138 shows us that $\Omega \approx \pi ab$, the area of an ellipse in a flat geometry.

We calculate r_1 using the following formula:

$$\begin{aligned} r_1 &= E[\boldsymbol{\gamma}_1 \cdot \mathbf{y}] = E[\boldsymbol{\gamma}'_1 \cdot \mathbf{y}'] = \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} (\boldsymbol{\gamma}'_1 \cdot \mathbf{y}') \sin(\theta) d\theta d\phi \\ &= \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} \cos(\theta) \sin(\theta) d\theta d\phi \end{aligned} \quad (4.139)$$

After integrating with respect to θ , we find:

$$\begin{aligned} r_1 &= \frac{1}{2\Omega} \int_0^{2\pi} (1 - \cos^2(\theta(\phi))) d\phi = \frac{2}{\Omega} \int_0^{\pi/2} \frac{q^2 - p^2}{q^2 - \cos^2(\phi)} d\phi \\ &= \frac{2}{\Omega} \frac{q^2 - p^2}{q^2 - 1} \int_0^{\pi/2} \frac{1}{1 + \frac{1}{q^2-1} \sin^2(\phi)} d\phi. \end{aligned} \quad (4.140)$$

We see that our integral once again takes the form of an elliptic integral of the third kind:

$$r_1 = \frac{2}{\Omega} \frac{q^2 - p^2}{q^2 - 1} \Pi \left(\frac{1}{1 - q^2} \middle| 0 \right). \quad (4.141)$$

This nicely simplifies down to:

$$r_1 = \frac{\pi}{\Omega} \frac{q^2 - p^2}{q\sqrt{q^2 - 1}}. \quad (4.142)$$

Substituting in p and q , we get the following formulation:

$$r_1 = \frac{\pi}{\Omega} \sin(a) \sin(b). \quad (4.143)$$

Taking the limit as a and b become very small, we see that r_1 approaches 1.

To calculate r_2 , we need to calculate two expectation values:

$$r_2 = E[(\boldsymbol{\gamma}_2 \cdot \mathbf{y})^2] - E[(\boldsymbol{\gamma}_3 \cdot \mathbf{y})^2]. \quad (4.144)$$

We therefore need to solve the following integrals:

$$E[(\boldsymbol{\gamma}_2 \cdot \mathbf{y})^2] = \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} \sin^2(\theta) \cos^2(\phi) \sin(\theta) d\theta d\phi, \quad (4.145)$$

$$E[(\boldsymbol{\gamma}_3 \cdot \mathbf{y})^2] = \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} \sin^2(\theta) \sin^2(\phi) \sin(\theta) d\theta d\phi. \quad (4.146)$$

Subtracting the two integrals from each other gives us:

$$r_2 = \frac{1}{\Omega} \int_0^{2\pi} \int_0^{\theta(\phi)} \sin^2(\theta) \cos(2\phi) \sin(\theta) d\theta d\phi. \quad (4.147)$$

Evaluating the integral over θ , we find:

$$r_2 = \frac{1}{\Omega} \int_0^{2\pi} \left(\frac{2}{3} - \cos(\theta(\phi)) + \frac{1}{3} \cos^3(\theta(\phi)) \right) \cos(2\phi) d\phi. \quad (4.148)$$

Substituting in Equation 4.126 yields:

$$r_2 = \frac{1}{\Omega} \int_0^{2\pi} \left(\frac{2}{3} - \sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} + \frac{1}{3} \left(\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)} \right)^{3/2} \right) \cos(2\phi) d\phi. \quad (4.149)$$

The first term in this integral naturally cancels out, leaving us with:

$$r_2 = \frac{1}{\Omega} \int_0^{2\pi} \left(\frac{1}{3} \left(\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)} \right)^{3/2} - \sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} \right) \cos(2\phi) d\phi. \quad (4.150)$$

Once again, we find ourselves with a rather tedious integral with no elementary solution.

So, we will need to find a way to approximate this integral for our exigencies. Since $p^2 =$

$q^2 \cos^2(b)$, we can show that:

$$\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)} = 1 - \frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} \quad (4.151)$$

For very small b , we can therefore make the following approximations:

$$\sqrt{\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)}} \approx 1 - \frac{1}{2} \frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} - \frac{1}{8} \left(\frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} \right)^2, \quad (4.152)$$

$$\left(\frac{p^2 - \cos^2(\phi)}{q^2 - \cos^2(\phi)} \right)^{3/2} \approx 1 - \frac{3}{2} \frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} + \frac{3}{8} \left(\frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} \right)^2. \quad (4.153)$$

Applying these approximations to our original integral simplifies to:

$$r_2 \approx \frac{1}{4\Omega} \int_0^{2\pi} \left(\frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} \right)^2 \cos(2\phi) d\phi = \frac{1}{\Omega} \int_0^{\pi/2} \left(\frac{q^2 \sin^2(b)}{q^2 - \cos^2(\phi)} \right)^2 \cos(2\phi) d\phi. \quad (4.154)$$

Evaluating this integral gives us:

$$r_2 \approx \frac{\pi q \sin^4(b)}{4(q^2 - 1)^{3/2} \Omega}. \quad (4.155)$$

Substituting in q finally yields our approximation of r_2 :

$$r_2 \approx \frac{\pi}{4\Omega} \sin(a) \sin(b) (\sin^2(a) - \sin^2(b)) = \frac{r_1}{4} (\sin^2(a) - \sin^2(b)). \quad (4.156)$$

When binning a body, we use its sky position, heliocentric distance, and proper motion to determine the parameters of its FB_5 surface distribution. For each bin, we then numerically calculate the volume under the surface distribution in that bin by taking the double integral:

$$V = \int_{\Lambda_{min}}^{\Lambda_{max}} \int_{B_{min}}^{B_{max}} \frac{1}{c(\kappa, \beta)} e^{(\kappa \gamma_1 \cdot \mathbf{x}) + \beta[(\gamma_2 \cdot \mathbf{x})^2 - (\gamma_3 \cdot \mathbf{x})^2]} \cos(B) d\Lambda dB, \quad (4.157)$$

where $c(\kappa, \beta)$ is approximated using:

$$c(\kappa, \beta) \approx \frac{2\pi e^\kappa}{\sqrt{\kappa^2 - 4\beta^2}}. \quad (4.158)$$

The amount by which each bin count is incremented is still either 0 or 1, depending on whether the body falls within the bin. However, we set the derivative information of the

incremented amount equal to the derivative information stored within the calculated volume. Therefore, after assigning all bodies to their appropriate bins and propagating their derivatives forward, we derive an adequate numerical approximation of how the counts in each bin will shift given a change in the input parameters. This formulation also allows us to derive derivative information for the total number of bodies that fall within the range of our histogram, an important quantity needed to determine the total mass within the observed stream, a key component of the output likelihood score.

4.4.3 Derivatives of the Reverse Evolution Time

Unlike most of our input parameters, the reverse evolution time is never passed into a classical continuous function in the simulation code. Rather, it is implemented in a “while” loop to tell the code when to stop performing the reverse orbit integration. Since this is the only way we use the evolution time, no derivatives with respect to such times can be calculated using classical automatic differentiation.

Fortunately for us, inserting the reverse evolution time derivatives into the code is a much simpler endeavor than what was required in the previous sections. The reverse evolution time τ_{evolve} debuts in the code when determining the initial starting position (\mathbf{x}_0) and velocity (\mathbf{v}_0) of the dwarf progenitor using a reverse orbit integrator. In order for us to propagate the τ_b derivatives, we must calculate the τ_b derivatives of \mathbf{x}_0 and \mathbf{v}_0 and artificially insert them into the derivative information of \mathbf{x}_0 and \mathbf{v}_0 . Increasing τ_{evolve} is equivalent to adding more timesteps to the reverse orbit integration. Therefore, the derivatives of the time-reversed positions and velocities is easy to derive:

$$\frac{\partial \mathbf{x}_0}{\partial \tau_b} = \frac{\partial \mathbf{x}_{0,T}}{\partial \tau_b} = \mathbf{v}_{0,T} = -\mathbf{v}_0, \quad (4.159)$$

$$\frac{\partial \mathbf{v}_0}{\partial \tau_b} = -\frac{\partial \mathbf{v}_{0,T}}{\partial \tau_b} = -\mathbf{a}_{0,T}, \quad (4.160)$$

where $\mathbf{a}_{0,T}$ is the initial *time-reversed* external acceleration felt by the progenitor. The accelerations in our system are not eigenstates of the time-reversal operator since our simulations since our simulations apply dynamical friction to the LMC, a drag vector acting in the opposite direction of its velocity. Therefore, we cannot state $\mathbf{a}_{0,T} = \mathbf{a}_0$.

Using Equations 4.159 and 4.160, we can artificially add the τ_b derivatives to the “gradient” attribute in the \mathbf{x}_0 and \mathbf{v}_0 objects. Introducing the off-diagonal τ_b elements

of the Hessian matrix to the \mathbf{x}_0 and \mathbf{v}_0 objects is also simple, implementing the gradient information of \mathbf{v}_0 and \mathbf{a}_0 :

$$\frac{\partial^2 \mathbf{x}_0}{\partial \tau_b \partial p} = -\frac{\partial \mathbf{v}_0}{\partial p}, \quad (4.161)$$

$$\frac{\partial^2 \mathbf{v}_0}{\partial \tau_b \partial p} = -\frac{\partial \mathbf{a}_{0,\mathbf{T}}}{\partial p}, \quad (4.162)$$

where p is an input parameter other than τ_b . For the diagonal term of the \mathbf{x}_0 Hessian, we easily derive:

$$\frac{\partial^2 \mathbf{x}_0}{\partial \tau_b^2} = -\frac{\partial \mathbf{v}_0}{\partial \tau_b} = \mathbf{a}_{0,\mathbf{T}}. \quad (4.163)$$

However, the diagonal term of the \mathbf{v}_0 Hessian requires a quantity that is not used or stored in our original N-body simulation:

$$\frac{\partial^2 \mathbf{v}_0}{\partial \tau_b^2} = -\frac{\partial^2 \mathbf{v}_{0,\mathbf{T}}}{\partial \tau_b^2} = -\frac{\partial \mathbf{a}_{0,\mathbf{T}}}{\partial \tau_b} = -\mathbf{j}_{0,\mathbf{T}}, \quad (4.164)$$

where $\mathbf{j}_{0,\mathbf{T}}$ is the initial time-reversed external jerk felt by the progenitor. As this quantity can have a rather complicated derivation, we decided to approximate this quantity using a numerical second derivative of the time-reversed velocity. Since we are performing a reverse orbit calculation, we naturally calculate this velocity at several past timesteps. Using a backward finite difference formula for the second derivative and the velocities from the previous three timesteps, we can quickly and accurately calculate the time-reversed jerk:

$$\mathbf{j}_{0,\mathbf{T}} \approx \frac{2\mathbf{v}_{0,\mathbf{T}}^{(0)} - 5\mathbf{v}_{0,\mathbf{T}}^{(1)} + 4\mathbf{v}_{0,\mathbf{T}}^{(2)} - \mathbf{v}_{0,\mathbf{T}}^{(3)}}{\Delta \tau_b^2}, \quad (4.165)$$

where $\mathbf{v}_{0,\mathbf{T}}^{(i)}$ is the time-reversed velocity calculated i steps in the past. Substituting into Equation 4.164 and simplifying yields:

$$\frac{\partial^2 \mathbf{v}_0}{\partial \tau_b^2} \approx \frac{2\mathbf{v}_0^{(0)} - 5\mathbf{v}_0^{(1)} + 4\mathbf{v}_0^{(2)} - \mathbf{v}_0^{(3)}}{\Delta \tau_b^2}. \quad (4.166)$$

4.4.4 Derivatives from the EMD Calculation

When comparing the angular distribution of two tidal streams, we implement an Earth Mover's Distance (EMD) calculation rather than a χ^2 calculation. This formulation introduces an additional complexity to our automatic differentiation algorithms. To explain why, we must first elaborate on how this calculation is computed in the first place. We describe

a histogram containing the binned information of the simulated tidal stream in n bins as:

$$P = \{((\Lambda_0, B_0), N_{p,0}), \dots, ((\Lambda_{n-1}, B_{n-1}), N_{p,n-1})\}, \quad (4.167)$$

where (Λ_i, B_i) is the Λ - B coordinate of the i th bin and $N_{p,i}$ is the number of bodies binned in the i th bin. We similarly describe the data histogram containing the real tidal stream we wish to compare against as:

$$Q = \{((\Lambda_0, B_0), N_{q,0}), \dots, ((\Lambda_{n-1}, B_{n-1}), N_{q,n-1})\}. \quad (4.168)$$

Before performing our EMD calculation, we first normalize both histograms:

$$\tilde{P} = \left\{ \left((\Lambda_0, B_0), w_{p,0} = \frac{N_{p,0}}{\sum_k N_{p,k}} \right), \dots, \left((\Lambda_{n-1}, B_{n-1}), w_{p,n-1} = \frac{N_{p,n-1}}{\sum_k N_{p,k}} \right) \right\}, \quad (4.169)$$

$$\tilde{Q} = \left\{ \left((\Lambda_0, B_0), w_{q,0} = \frac{N_{q,0}}{\sum_k N_{q,k}} \right), \dots, \left((\Lambda_{n-1}, B_{n-1}), w_{q,n-1} = \frac{N_{q,n-1}}{\sum_k N_{q,k}} \right) \right\}, \quad (4.170)$$

We therefore define the EMD between \tilde{P} and \tilde{Q} as⁵:

$$EMD(\tilde{P}, \tilde{Q}) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f(\tilde{P}, \tilde{Q})_{ij}, \quad (4.171)$$

where d_{ij} are the ground distances between the bins as described below:

$$d_{ij} = \sqrt{(\Lambda_i - \Lambda_j)^2 + (B_i - B_j)^2}, \quad (4.172)$$

and $f(\tilde{P}, \tilde{Q})_{ij}$ is the flow matrix between \tilde{P} and \tilde{Q} that minimizes the work W :

$$W = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{ij}. \quad (4.173)$$

The flow matrix $f(\tilde{P}, \tilde{Q})_{ij}$ represents a transformation of \tilde{P} into \tilde{Q} where each element shows how much of the weight in the i th bin must be shifted into the j th bin for such

⁵Technically, the EMD is supposed to be normalized with respect to the flow matrix ($EMD(\tilde{P}, \tilde{Q}) = \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij} f(\tilde{P}, \tilde{Q})_{ij}}{\sum_{i=1}^n \sum_{j=1}^n f(\tilde{P}, \tilde{Q})_{ij}}$), but since we have normalized P and Q beforehand, the denominator of this quantity will always be 1.

a transformation to occur. Calculating an EMD boils down to determining which flow matrix that transforms \tilde{P} into \tilde{Q} minimizes the total work, and this is where automatic differentiation encounters a problem. Similar to the root-finding algorithm in Section 4.4.1.3, the algorithm we use to determine this optimal flow matrix does not fully propagate the derivative information we are interested in. Therefore, in order to determine the derivatives of the EMD, we must find a way to approximate the derivatives of the optimized flow matrix.

Fortunately for us, since we are using simple 1-dimensional histograms, the EMD can be easily calculated using the following algorithm: we define a sequence Δ such that from two histograms P and Q such that:

$$\Delta_0 = 0, \quad (4.174)$$

and

$$\Delta_{i+1} = N_{p,i} + \Delta_i - N_{q,i}. \quad (4.175)$$

Using this sequence, we can derive the EMD using the following analytical formula:

$$EMD = \sum_i^n |\Delta_i|. \quad (4.176)$$

This algorithm gives us an easy and direct way to calculate the EMD from a pair of histograms P and Q . Since this algorithm uses simple functions, we can easily apply automatic differentiation to this set of formulas. It should be noted, though, that the absolute value function is not differentiable everywhere, which does complicate the differentiation. To make these derivatives continuous, we approximate the derivative of the absolute value function as a steep sigmoid function:

$$\frac{d|x|}{dx} \approx \frac{x}{\sqrt{x^2 + \epsilon^2}}, \quad (4.177)$$

where ϵ is the machine precision of the computer.

While it is not particularly needed for our analysis, it would be nice if automatic differentiation could also calculate the derivatives of the EMD for the case of the 2-dimensional histogram for potential future projects. In such cases, the code numerically calculates the first and second order derivatives of the EMD with respect to the number of *non-normalized* counts in each bin, using those derivatives to propagate the derivatives from the bin counts in automatic differentiation. We originally tried to approximate the gradient and hessian using solely the properties of EMD optimization, however, the method was both too com-

putationally expensive and inaccurate to be properly implemented. A detailed synopsis of this attempt can be found in Appendix C.

CHAPTER 5

ANALYSIS OF SYSTEMATIC ERRORS

After acquiring our best fit dwarf parameters from Chapter 2 and creating a version of N -body compatible with automatic differentiation in Chapter 4, we ran those same fitted parameters on N -body with automatic differentiation active. Over the course of this coding project, we found one additional fit of the OCS progenitor that was consistent with Mendelsohn et al. (2022) (Run 12). The parameter sets we plugged into automatic differentiation can be seen in Table 5.1.

5.1 Complications of a Turbulent Likelihood Surface

Something that became painfully apparent when calculating the gradient and Hessian of the likelihood surface was its inherently turbulent and chaotic behavior. We quickly found that the derivatives of such a surface become so obscenely large that performing derivative propagations in linear space produces values that potentially cannot be stored within the computer’s memory. For example, Run 3 produced gradients whose orders of magnitudes hovered around 10^{41} and Hessians with an average order of magnitude of 10^{86} . Table 5.2 shows the natural logarithm of the largest derivative within each run’s calculated gradient and Hessian.

Table 5.1: Fitted dwarf parameters determined from MilkyWay@home’s differential evolution genetic algorithm. Run 12 is the additional fit found to be consistent with Mendelsohn et al. (2022).

Run	τ_b	R_b	ξ_R	M_b	ξ_M	$\ln(\mathcal{L})$
2	3.6345	0.2327	0.2543	1.146	0.0179	-8.809277
3	3.6333	0.1812	0.1828	1.223	0.0126	-7.726065
5	3.6334	0.1842	0.1820	1.251	0.0119	-7.997704
12	3.6322	0.2195	0.2467	1.145	0.0200	-8.388980

Table 5.2: Natural logarithm of the largest value within each run’s gradient and Hessian.

Run	$\ln(\nabla \ln(\mathcal{L})_{max})$	$\ln(\nabla^2 \ln(\mathcal{L})_{max})$
2	94.2	209.2
3	105.8	221.3
5	49.6	109.6
12	96.2	199.9

The main source of these high derivatives comes from the chaotic behavior that arises from body-body gravitational interactions. It is well known that gravitational systems with three or more bodies exhibit chaotic behavior, where any slight change in the initial conditions of the system potentially changes its final state drastically. Such systems naturally imply very large first and second order derivatives. One way to observe how these body-body interactions naturally propagate large derivatives is to modify the softening length. When the softening length is large, the most dominant gravitational force between two nearby bodies, the source of our observed chaos, is deadened. In Figure 5.1, we see that for smaller softening lengths, which are indicative of more chaotic behavior, the first and second order derivatives of our likelihood score grow to extraordinarily large values. For each softening length in this plot, we generated a simulated tidal stream as our N -body executable input and compared it to tidal streams with the same dwarf parameters, but generated using different random seeds. We averaged over six seeds to get Figure 5.1.

In chaotic systems, we also expect to see the system to become even more chaotic (exhibiting more extreme derivatives) as the system evolves over time. Figure 5.2 shows how the derivative information of a single body grows rather quickly as the simulation progresses. The large derivatives produced from this turbulence were what motivated the addition of the “`lnfactor_gradient`” and “`lnfactor_hessian`” attributes.

While running with a larger softening length would allow us to circumvent the need to store the large derivatives, the size we would need to significantly reduce our derivatives by orders of magnitude would end up sacrificing too much accuracy in our acceleration computations. A more in depth analysis of this can be found in Appendix D.

Surprisingly, the large factors we observe within the gradient and Hessian do not necessarily imply a similarly large Jacobian. Recalling Equation 4.10, we see that our desired Jacobian is essentially the matrix product of two matrices, where one is a sub-matrix of our Hessian and the other is the inverse of another sub-matrix from the same Hessian. Looking back at our example Run 3, our Hessian has an overall factor of 10^{86} . Our non-inverted sub-matrix would therefore contribute a factor of 10^{86} to the Jacobian whereas our inverted

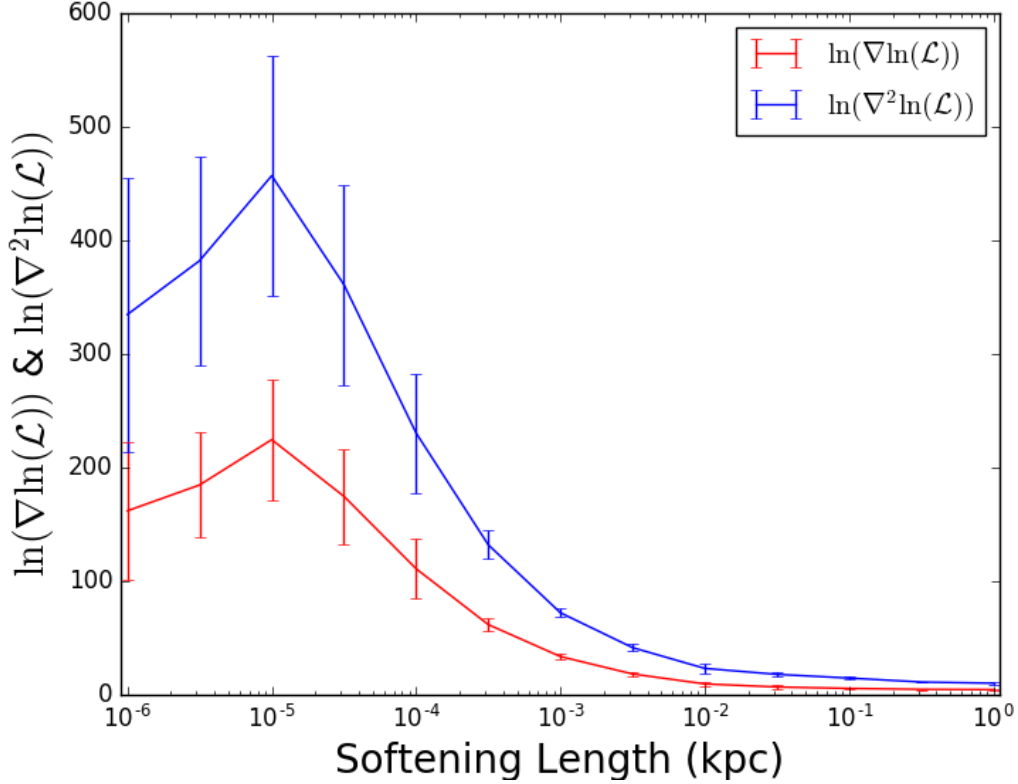


Figure 5.1: A plot showing how the softening length, which moderates chaotic behavior, affects the gradient and Hessian of our likelihood score. The red line shows the “lnfactor_gradient” of the likelihood score, and the blue line represents the likelihood score’s “lnfactor_hessian”. We observe that smaller softening lengths tend to more extreme derivatives. For each softening length, we averaged each quantity over 6 random seeds to obtain our standard errors. We surprisingly find a maximum at a softening length of 10^{-5} kpc. We know that when the softening length is too small, the system is dominated by strong interactions that cannot be properly resolved in N -body, so the system evaporates. Bodies ejected by such close encounters do not feel any more strong interactions for the rest of the simulation, and thus do not contribute more to the chaotic behavior.

portion contributes a cancelling factor of 10^{-86} , providing a Jacobian of a much lower and manageable order of magnitude.

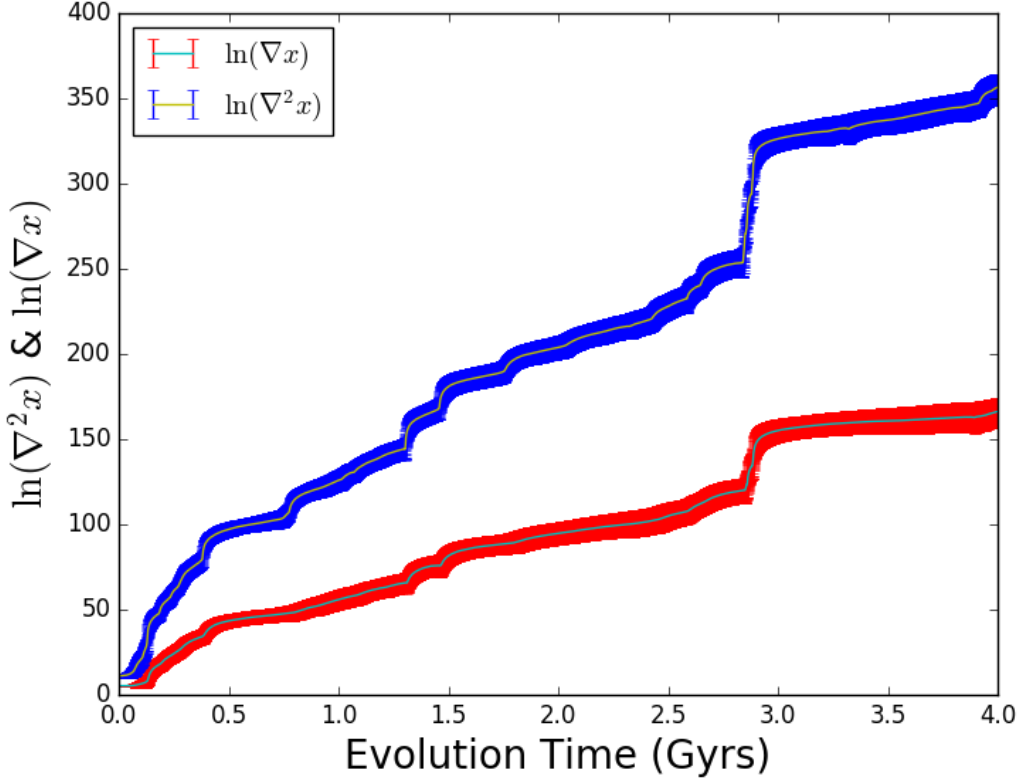


Figure 5.2: A plot of how the derivative information increases as a function of time. The simulation which generated this plot came from generating a 500-body 2-component dwarf galaxy and evolving it in a Milky Way potential. The cyan line with red error bars shows the average “lnfactor_gradient” value of each body’s x coordinate, and the yellow line with blue error bars shows the average “lnfactor_hessian” value of each body’s x coordinate. We see that the size of the gradient and Hessian both steadily increase in order of magnitude together almost logarithmically.

5.2 Discussion of Results

Despite our Jacobian’s insensitivity to the overall size of the Hessian, the Jacobian calculation is still dominated by noise. We observe and quantify this noise in Table 5.3, where we record the coefficient of variation (standard deviation divided by mean) of the Jacobian. Just as in a turbulent river it is impossible to estimate the river’s speed from measuring the velocity of a single water molecule, we cannot accurately determine how our likelihood peak shifts given a single Jacobian calculation. However, just as we can average the velocities of several water molecules to infer the river’s speed, so too can we average

Table 5.3: Coefficients of variation of the Jacobian (absolute value). With the exception of $\frac{\partial R_b}{\partial q_z}$, each value is larger than one, the standard deviation of the Jacobian must have a larger magnitude than its average, implying turbulence.

$\frac{\partial}{\partial x}$	τ_b	R_b	ξ_R	M_b	ξ_M
b	1.5	34	1.6	1.7	13
r	2.3	1.5	2.4	2.6	1.8
v_x	2.3	1.5	2.4	2.6	1.4
v_y	1.9	2.6	1.2	1.6	2.2
v_z	1.9	11	1.8	1.2	4.1
M_{bulge}	7.3	1.2	4.0	7.8	1.2
a_{bulge}	2.3	1.2	2.1	4.0	1.2
M_{disk}	2.4	1.2	2.3	51	1.2
a_{disk}	2.4	2.2	2.3	2.6	1.3
b_{disk}	1.6	1.4	1.8	7.8	1.2
v_{halo}	2.5	1.2	2.3	19	1.2
a_{halo}	2.5	3.8	1.7	1.6	2.9
q_z	2.3	0.86	1.4	2.0	1.1
M_{LMC}	2.4	1.4	2.3	2.7	1.4

several Jacobians to approximate the true sensitivity matrix.

Bearing in mind the complications and limitations that arise from a chaotic likelihood surface, we calculate the sensitivity matrices for each of the runs listed in Table 5.1. In Table 5.4, we report the calculated Jacobian averaged over all four runs. Although the Jacobian alone is not sufficient to draw conclusions about the systematic errors of our optimizations, we do note that the scale radius of the LMC (a_{LMC}) is exactly zero, implying no relationship between the size of the LMC and any of the fitted parameters. As striking as this observation appears, this result is not surprising since our optimizations did not originally include an LMC, setting its mass to zero. If we were to slightly shift the scale radius of the LMC while keeping its mass zero, then we would expect to see no change in the system. The same could be said if we were to calculate the derivatives of the likelihood score with respect to any of the other LMC parameters, such as its initial position and velocity. As such, our analysis will not consider the systematic errors from the uncertainty in the LMC radius. While the Jacobian terms regarding the LMC mass are non-zero, it is still not truly possible to accurately measure its systematic error on the system without significant and severe

Table 5.4: Average Jacobian of the fitted dwarf parameters. We represent the units of length, time, mass, and angle are given in kiloparsecs, gigayears, structure masses ($1SM = 222, 288.47M_\odot$), and degrees respectively.

$\frac{\partial}{\partial x}$	τ_b	R_b	ξ_R	M_b	ξ_M
b	4.8×10^{-2}	-9.2×10^{-5}	1.2×10^{-2}	-6.4×10^{-1}	-1.1×10^{-3}
r	-3.4×10^0	9.1×10^{-2}	-8.1×10^{-1}	2.9×10^1	-3.4×10^{-1}
v_x	8.1×10^{-1}	-2.9×10^{-2}	2.1×10^{-1}	-7.2×10^0	9.7×10^{-2}
v_y	-3.8×10^{-2}	-5.1×10^{-3}	-1.5×10^{-2}	1.9×10^0	3.1×10^{-2}
v_z	-3.7×10^{-1}	-3.3×10^{-3}	-1.1×10^{-1}	6.8×10^0	4.1×10^{-2}
M_{bulge}	6.3×10^{-6}	-2.5×10^{-6}	3.9×10^{-6}	1.0×10^{-4}	3.6×10^{-6}
a_{bulge}	1.3×10^0	-4.3×10^{-2}	3.4×10^{-1}	-7.8×10^0	2.2×10^{-1}
M_{disk}	1.1×10^{-4}	-6.5×10^{-6}	3.2×10^{-5}	-7.7×10^{-5}	2.1×10^{-5}
a_{disk}	-1.0×10^0	3.4×10^{-2}	-2.9×10^{-1}	9.2×10^0	-1.3×10^{-1}
b_{disk}	-1.0×10^{-1}	2.4×10^{-3}	-2.4×10^{-2}	-4.6×10^{-1}	-2.0×10^{-2}
v_{halo}	1.1×10^0	-6.6×10^{-2}	3.0×10^{-1}	2.3×10^0	2.7×10^{-1}
a_{halo}	3.0×10^{-1}	-4.9×10^{-3}	1.0×10^{-1}	-3.8×10^0	2.2×10^{-2}
q_z	3.7×10^{-1}	-3.9×10^{-2}	1.3×10^{-1}	2.1×10^1	3.1×10^{-1}
M_{LMC}	-4.2×10^{-5}	1.3×10^{-6}	-1.1×10^{-5}	3.7×10^{-4}	-5.4×10^{-6}
a_{LMC}	0	0	0	0	0

extrapolation.

To calculate the amount of error contributed to each fitted parameter, we must first determine the uncertainty of each of our model parameters. The uncertainties in the orbital parameters b , r , v_x , v_y , and v_z , as well as the halo velocity v_{halo} , were pulled directly from Newberg et al. (2010). The bulge and disk parameter uncertainties were obtained by determining the standard deviation of parameters used in Law et al. (2005), Wang et al. (2022), Lilleengen et al. (2022), and the commonly used galpy potential MWPotential2014 described in Bovy (2015). We determined the uncertainty in the halo scale radius a_{halo} from analyzing Newberg et al. (2010), Wang et al. (2022), Lilleengen et al. (2022), and Bovy (2015). Since our original Milky Way potential had no halo flattening ($q_z = 1$), we took the average oblate flattening from Johnston et al. (2005) and Xue et al. (2015), recording the uncertainty as one minus that average. Since we also ran with no LMC, we treat the uncertainty in its mass M_{LMC} as the mass reported from Erkal et al. (2019) ($4.5 \times 10^5 SM$), though using such an error is a clear and blatant extrapolation beyond what one would consider reasonable for

Table 5.5: The uncertainties associated with each model parameter. We represent the units of length, time, mass, and angle are given in kiloparsecs, gigayears, structure masses ($1SM = 222,288.47M_\odot$), and degrees respectively. We treat the uncertainty in the LMC mass M_{LMC} as the mass reported in Erkal et al. (2019), $4.5 \times 10^5 SM$.

b	r	v_x	v_y	v_z	M_{bulge}	a_{bulge}	M_{disk}	a_{disk}	b_{disk}	v_{halo}	a_{halo}	q_z
1.2	1.8	10	1.0	9.2	62000	0.42	120000	1.9	0.019	25	2.8	0.19

use in error propagation. The uncertainties determined from analyzing the current literature are listed in Table 5.5.

Using the uncertainties in Table 5.5, the Jacobian in Table 5.4, and basic error propagation, we find the fractional error contribution of each model parameter for each of our five fitted parameters, which are recorded in Table 5.6. The fractional errors provided within this table are extraordinarily large. Ignoring the extrapolated errors from the lack of a LMC, we find that the measured systematic errors on each of our fitted parameters range from ten times to almost 600 times the fitted values. With the exception of the scale height of the disk b_{disk} , every model parameter listed in Table 5.6 adds an incredibly significant systematic error to the fitted baryonic mass M_b of the progenitor dwarf, a parameter that has otherwise been shown to be fit extremely well in MilkyWay@home (Shelton 2018). Despite this inconsistency, we do notice that the systematic errors in the dark matter mass place an error range that covers the ranges proposed by Newberg et al. (2010), Hendel et al. (2018), and Fardal et al. (2019), suggesting that further resolution of either our starting errors or the Jacobian could definitively narrow down the mass ranges of the OCS progenitor.

It is still possible that performing the analysis with smaller uncertainties in the model parameters could produce more reasonable systematic errors. Shelton (2018) showed in his thesis that when MW@home was given the tidal debris from a known simulated stream, the distributed supercomputer was able to consistently pull out the parameters that produced the simulated stream with reasonable accuracy. More specifically, it was able to determine the evolution time τ_b to a precision of 3%, the baryonic mass M_b and scale radius a_b to a precision of 1%, and the dark matter mass M_d and scale radius a_d to a precision of 10%. This level of precision is what we should expect if all of the Galactic model parameters are known exactly. For the purpose of this analysis, we determine the minimum errors needed to determine τ_b to 50%, R_b to 10%, M_b to 10%, R_d to 20%, and M_d to within an order

Table 5.6: Fractional error contributions from each of the non-fitted model parameters. We also include the fractional errors for the dark matter radius R_d and mass M_d . We also record the total fractional systematic error for each parameter below (totals were added in quadrature). These totals do not include the contributions from the missing LMC as we cannot confidently determine them from this analysis.

	τ_b	R_b	ξ_R	M_b	ξ_M	R_d	M_d
b	1.6×10^{-2}	5.5×10^{-4}	7.3×10^{-2}	6.3×10^{-1}	9.9×10^{-2}	9.2×10^{-2}	6.4×10^{-1}
r	1.7×10^0	8.3×10^{-1}	7.2×10^0	4.4×10^1	4.5×10^1	9.1×10^0	6.3×10^1
v_x	2.2×10^0	1.5×10^0	1.0×10^1	6.0×10^1	7.1×10^1	1.3×10^1	9.4×10^1
v_y	1.0×10^{-2}	2.5×10^{-2}	7.5×10^{-2}	1.6×10^0	2.3×10^0	9.8×10^{-2}	2.8×10^0
v_z	9.3×10^{-1}	1.5×10^{-1}	4.9×10^0	5.2×10^1	2.8×10^1	6.2×10^0	5.9×10^1
M_{bulge}	1.1×10^{-1}	7.7×10^{-1}	1.2×10^0	5.2×10^0	1.7×10^1	1.7×10^0	1.8×10^1
a_{bulge}	1.5×10^{-1}	9.2×10^{-2}	7.1×10^{-1}	2.7×10^0	6.8×10^0	9.0×10^{-1}	7.4×10^0
M_{disk}	3.7×10^0	3.9×10^0	1.9×10^1	7.7×10^0	1.9×10^2	2.4×10^1	1.9×10^2
a_{disk}	5.3×10^{-1}	3.3×10^{-1}	2.7×10^0	1.4×10^1	1.8×10^1	3.4×10^0	2.3×10^1
b_{disk}	5.4×10^{-4}	2.3×10^{-4}	2.2×10^{-3}	7.2×10^{-3}	2.7×10^{-2}	2.8×10^{-3}	2.8×10^{-2}
v_{halo}	7.6×10^0	8.3×10^0	3.6×10^1	4.7×10^1	5.0×10^2	4.6×10^1	5.1×10^2
a_{halo}	2.3×10^{-1}	6.9×10^{-2}	1.4×10^0	8.8×10^0	4.4×10^0	1.8×10^0	9.9×10^0
q_z	2.0×10^{-2}	3.7×10^{-2}	1.2×10^{-1}	3.3×10^0	4.3×10^0	1.6×10^{-1}	5.5×10^0
M_{LMC}	5.2×10^0	2.9×10^0	2.4×10^1	1.4×10^2	1.8×10^2	3.0×10^1	2.3×10^2
TOTAL	1.0×10^1	8.3×10^0	4.3×10^1	1.0×10^2	5.4×10^2	5.5×10^1	5.6×10^2

of magnitude (100%). For the parameters p_b that have a direct reference in the Jacobian (where p_b can be either τ_b , R_b , or M_b), calculating the target uncertainty σ_x of a parameter x is very simple:

$$\sigma_x = \frac{p_b \delta_{p_b}}{\left| \frac{\partial p_b}{\partial x} \right|}, \quad (5.1)$$

where δ_{p_b} is the target precision we wish to achieve for the fitted parameter p_b and $\frac{\partial p_b}{\partial x}$ is the relevant Jacobian value. On the other hand, the fitted dark matter parameters require a more complex analysis, since the fitted parameters describing the dark matter are represented by ratios in relation to the baryonic components. The relation between the baryonic mass M_b , the dark matter mass M_d , and the fitted mass ratio ξ_M is given by the following equation:

$$M_d = M_b \left(\frac{1}{\xi_M} - 1 \right). \quad (5.2)$$

The error propagation of this relation is therefore given as follows:

$$\sigma_{M_d}^2 = \left(\frac{1}{\xi_M} - 1 \right)^2 \sigma_{M_b}^2 + \frac{M_b^2}{\xi_M^4} \sigma_{\xi_M}^2. \quad (5.3)$$

By dividing both sides by M_d , we can then derive the fractional error propagation:

$$\delta_{M_d}^2 = \delta_{M_b}^2 + \frac{\delta_{\xi_M}^2}{(1 - \xi_M)^2}. \quad (5.4)$$

Since:

$$\delta_{\xi_M} = \left| \frac{\partial \xi_M}{\partial x} \right| \frac{\sigma_x}{\xi_M} \quad (5.5)$$

we can rearrange Equation 5.4 as:

$$\sigma_x = \xi_M (1 - \xi_M) \left| \frac{\partial \xi_M}{\partial x} \right|^{-1} \sqrt{\delta_{M_d}^2 - \delta_{M_b}^2}. \quad (5.6)$$

The same math applies for the dark matter scale radius R_d and the fitted radius ratio ξ_R . Using Equations 5.1 and 5.6, we calculated the maximum uncertainty allowed for each parameter to produce systematic errors on par with our target baseline, which we record in Table 5.7.

According to this table, we find that the baryonic mass M_b is the fitted parameter that generally demands the highest sensitivity from all the model parameters with the exception of the disk mass and halo velocity which is more constrained by the dark matter mass, though not by much. This observation is surprising since Shelton (2018) showed that MilkyWay@home's optimization algorithm is most sensitive to the baryonic mass, and thus should have smaller systematic errors.

Ignoring the baryonic mass, we observe from Table 5.7 that there are some model parameters whose required level of certainty is currently attainable. The Galactic latitude (b) of the OCS progenitor only needs to be known down to roughly ± 12 degrees (reasonable) to fit the total dark matter mass of the OCS progenitor to within 100%, and the heliocentric distance needs to be known to ± 39 pc, a precision of roughly 0.2%. While this might seem small, let us assume that the OCS's progenitor has a diameter of about 1 kpc (consistent with what we see in Figure 2.14). If we were to average the heliocentric distances of stars near the progenitor, we would calculate an standard deviation of roughly 0.5 kpc. Following

Table 5.7: Maximum uncertainty allowed for each model parameter. We list the target precision of each fitted parameter as a percentage next to the parameter name. The red text in the table shows the smallest allowed uncertainty for a model parameter. The smaller the number, the more sensitive the associated fitted parameter is to the associated model parameter.

	$\tau_b(50\%)$	$R_b(10\%)$	$M_b(10\%)$	$R_d(20\%)$	$M_d(100\%)$
b ($^\circ$)	3.8×10^1	2.2×10^2	1.9×10^{-1}	2.3×10^0	1.2×10^1
r (kpc)	5.3×10^{-1}	2.2×10^{-1}	4.1×10^{-3}	3.5×10^{-2}	3.9×10^{-2}
v_x (kpc Gyr $^{-1}$)	2.2×10^0	6.9×10^{-1}	1.7×10^{-2}	1.3×10^{-1}	1.4×10^{-1}
v_y (kpc Gyr $^{-1}$)	4.8×10^1	3.9×10^0	6.3×10^{-2}	1.8×10^0	4.2×10^{-1}
v_z (kpc Gyr $^{-1}$)	4.9×10^0	6.0×10^0	1.8×10^{-2}	2.6×10^{-1}	3.2×10^{-1}
M_{bulge} (SM)	2.9×10^5	8.1×10^3	1.2×10^3	7.2×10^3	3.7×10^3
a_{bulge} (kpc)	1.4×10^0	4.6×10^{-1}	1.5×10^{-2}	8.2×10^{-2}	6.1×10^{-2}
M_{disk} (SM)	1.6×10^4	3.1×10^3	1.6×10^3	8.8×10^2	6.3×10^2
a_{disk} (kpc)	1.8×10^0	5.8×10^{-1}	1.3×10^{-2}	9.9×10^{-2}	1.0×10^{-1}
b_{disk} (kpc)	1.7×10^1	8.4×10^0	2.6×10^{-1}	1.2×10^0	6.8×10^{-1}
v_{halo} (kpc Gyr $^{-1}$)	1.7×10^0	3.0×10^{-1}	5.3×10^{-2}	9.5×10^{-2}	4.9×10^{-2}
a_{halo} (kpc)	6.1×10^0	4.1×10^0	3.2×10^{-2}	2.7×10^{-1}	6.2×10^{-1}
q_z	4.9×10^0	5.2×10^{-1}	5.8×10^{-3}	2.2×10^{-1}	4.4×10^{-2}
M_{LMC} (SM)	4.3×10^4	1.6×10^4	3.3×10^2	2.6×10^3	2.5×10^3

the formula for the standard deviation of the mean ($\sigma_\mu = \frac{\sigma}{\sqrt{N}}$), we would need to know the distances of at least 160 stars that were guaranteed to be within the stream, a number we are able to detect near the progenitor’s core.

The velocities of the OCS progenitor, however, cannot be reasonably determined to an adequate precision. Assuming a heliocentric distance of 21.5 kpc, our analysis would require at most an uncertainty in proper motion on the order of ± 0.025 rad Gyr $^{-1}$, or ± 0.0052 mas yr $^{-1}$. Even *Gaia*, a project well-known for having the most accurate proper motion data to date, only has systematic errors in proper motion on the order of ± 0.1 mas yr $^{-1}$ (Luri et al. 2018), not nearly sensitive enough to resolve the necessary uncertainties. The same can also be said regarding the Galactic model parameters. Most papers that estimate the mass and scale lengths of the Milky Way Galaxy can typically only do so down to an error of 10% (Valenti et al. 2016; Licquia & Newman 2015; McMillan 2011, 2016; Kafle et al. 2014). However, the required uncertainties in our model parameters must be two or even

three orders of magnitude lower to produce our target precisions.

5.3 Improving the Jacobian Measurement

The results we acquired in Chapter 2 are very close to what we would expect in a dwarf galaxy, with exception of its total mass and mass within 300 pc. We would therefore expect the true systematic errors of the result to be much smaller than what our analysis with automatic differentiation would otherwise suggest. However, even a hypothetical analysis with impractically small uncertainties in the model parameters would fail to produce such expected systematic errors. It is therefore clear that our evaluation of the sensitivity matrix is far too large to produce a reasonable estimate of our systematic errors. The following subsections list potential ways we can feasibly improve the accuracy of this measurement.

5.3.1 Averaging More Jacobians

One reason we could be finding a large Jacobian could be because we only have 4 best-fit data points to average over. If the distribution of Jacobians is generally random in most directions within our search space, averaging too few points increases the likelihood of having all of our Jacobians pointing in a similar direction, overestimating the average. According to Table 5.3, the average coefficient of variation across each component of the Jacobian is roughly 3.6. This means in order to measure an average Jacobian to within 50% (using the formula for standard deviation of the mean $\sigma_\mu = \frac{\sigma}{\sqrt{N}}$), we would need at least 52 data points within our likelihood peak to resolve an accurate Jacobian and adequately quantify the systematic errors of our fitted OCS progenitor.

5.3.2 Running with More Bodies and Using Finite Difference

One way we might be able to reduce the turbulence and size of the Jacobians may be to increase the number of bodies we run within an optimization. Within Figure 5.3, we performed a zoomed in parameter sweep over the baryonic radius near its local maximum, similar to what was done in Figure 2.11. The figure on the left was created using a simulations of 40,000 bodies, whereas the figure on the right used 80,000 bodies.

From Figure 5.3, we notice that the distribution of likelihood scores becomes narrower and smoother when we double the number of bodies in the simulations. We do notice,

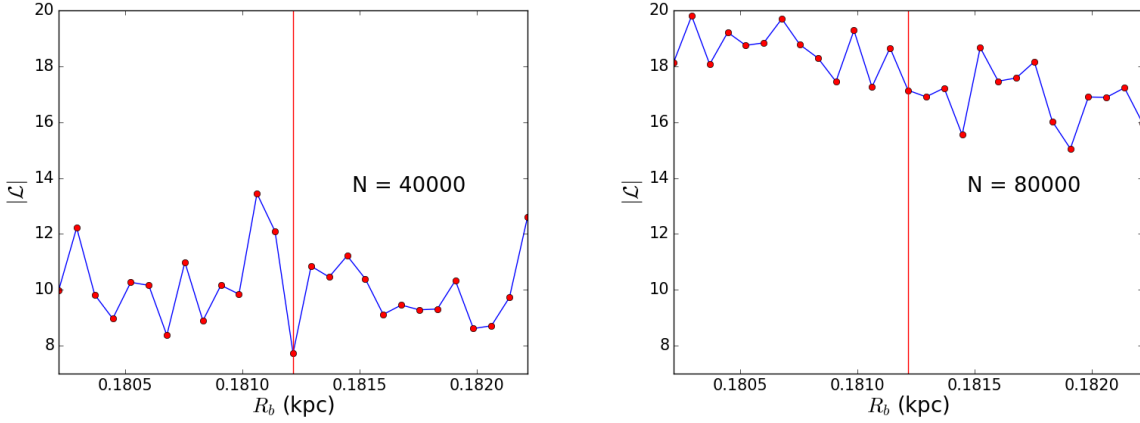


Figure 5.3: A parameter sweep in R_b zoomed into the likelihood peak. The plot on the left shows the parameter sweep run using 40,000 bodies. The plot on the right was run using 80,000 bodies. We see that while running with double the bodies has shifted the optimal radius upwards, we notice that the plot with more bodies has a slightly smoother likelihood surface.

however, that a slight slope in the likelihood surface emerges that would push our baryonic radius R_b slightly higher in an optimization. This smoothing we observe indicates that running with many more bodies, say 250,000 bodies, could sufficiently allow us to more accurately calculate the derivatives of our likelihood surface without needing to worry about noise. Unfortunately, running with that many bodies in a differential evolution algorithm would take an unacceptable amount of time on MilkyWay@home. So, we could instead initially optimize our results on MilkyWay@home using 40,000 bodies and polish our results on a local supercomputer with 250,000 using a method like conjugate gradient ascent.

Although increasing the number of bodies does seem to reduce the noise in our likelihood surface, we find that automatic differentiation instead detects derivatives of even higher magnitudes. Using the fitted parameters from Run 3, we found that running with 80,000 bodies instead of 40,000 brings our gradient to an order of magnitude of 10^{64} from 10^{41} and our Hessian to an order of 10^{139} from 10^{86} . This is because running with more bodies increases the number of strong interactions within the simulation, thus making the system more chaotic on its most infinitesimal scales. Adding more bodies in this sense is akin to tightening a loose spring. While relaxed, the spring has wide coils of a large amplitude, similar to the left plot in Figure 5.3. When we add more bodies and tighten the spring, the coils become more taut, and the spring appears to smooth out. However, the ups and downs

of the coil still exist on the springs smallest scales, exhibiting much more extreme slopes than when it was loose. With such behavior in the likelihood surface, it may be wise to use finite difference derivatives to calculate the requires derivatives for our sensitivity matrix.

CHAPTER 6

CONCLUSION

6.1 The OCS Fit

We made the first estimate of the mass and radial profile of the stars and dark matter in the dwarf galaxy progenitor of a tidal stream. To do this, we developed a procedure for characterizing the stellar distribution along and across the OCS, using turnoff stars in the OCS observed in the SDSS and the DEC. We used turnoff stars in NGC 5053 to approximate the stellar mass of the OCS. Using this information for a real observed tidal stream as input to MilkyWay@home, which uses the N-body optimization method developed in Shelton et al. (2021), we calculate the following properties for the dwarf galaxy progenitor of the OCS:

$$\tau_{evolve} = (3.6337 \pm 0.0004)\text{Gyrs} \quad (6.1)$$

$$R_B = (0.20 \pm 0.02)\text{kpc} \quad (6.2)$$

$$R_D = (0.77 \pm 0.05)\text{kpc} \quad (6.3)$$

$$M_B = (2.68 \pm 0.07) \times 10^5 M_\odot \quad (6.4)$$

$$M_D = (1.9 \pm 0.3) \times 10^7 M_\odot \quad (6.5)$$

$$M_{total} = (2.0 \pm 0.3) \times 10^7 M_\odot \quad (6.6)$$

From these numbers, we calculated a mass-to-light ratio of $\gamma = 73.5 \pm 10.6$. The implied $(1.1 \pm 0.2) \times 10^6 M_\odot$ within 300 pc of the progenitor's center is one order of magnitude smaller than the presumed minimum mass of ultrafaint dwarf galaxies.

Our simulations of the OCS's tidal debris show an unbounded and heavily disrupted progenitor remnant at a current sky position of around $(l, b) = ((264.9 \pm 2.9)^\circ, ((43.6 \pm 2.8)^\circ)$, or $(\alpha, \delta) = ((166.0 \pm 0.9)^\circ, (-11.1 \pm 2.5)^\circ)$. Further studies of the simulated tidal stream suggest that most of the mass of the OCS (especially its dark matter) may reside in its tails, making them optimal candidates for indirect dark matter detection experiments.

Portions of this chapter previously appeared as: Mendelsohn, E. J., Newberg, H. J., Shelton, S., et al. 2022, *Astrophys. J.*, 926, 106..

6.2 Systematic Errors from N-body Automatic Differentiation

We implemented automatic differentiation in an attempt to quantify the systematic error that arises from our uncertainty within our Galactic model parameters, a method that had never been attempted before. We however learned that due to the chaotic body-body gravitational interactions moderated by the softening length, any likelihood surface produced from an N-body simulation becomes dominated by turbulence, limiting the amount of useful information that can be gleaned from its derivatives.

We did not succeed in calculating useful estimates of the systematic errors in our measurement of the properties of the progenitor of the Orphan Stream. However, we continue to find that there is a tremendous power in the information contained in the distribution of tidal stream stars. It is possible that by increasing the number of runs (52 or more) we will be able to reduce the uncertainty in the derivatives. It is possible that running optimizations over more bodies and calculating the requisite Hessian matrix may allow us to calculate a more accurate sensitivity matrix, though we may need to use finite difference derivatives for us to properly determine it. The measurement of the systematic errors in the measurement of the progenitor of the Orphan-Chenab Stream remains an open question.

REFERENCES

- Abate, J., Bischof, C., Roh, L., & Carle, A. 1997, in Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, ISSAC '97 (New York, NY, USA: Association for Computing Machinery), 149–155
- Abdallah, H., Adam, R., Aharonian, F., et al. 2020, *Phys. Rev. D*, 102, 062001
- Ackermann, M., Albert, A., Anderson, B., et al. 2015, *Phys. Rev. Lett.*, 115, 231301
- Ahumada, R., Allende Prieto, C., Almeida, A., et al. 2020, *Astrophys. J., Suppl. Ser.*, 249, 3
- Albert, A., André, M., Anghinolfi, M., et al. 2017, *Physics Letters B*, 769, 249
- An, D., Johnson, J. A., Clem, J. L., et al. 2008, *ApJS*, 179, 326
- Arellano Ferro, A., Giridhar, S., & Bramich, D. M. 2010, *Mon. Not. R. Astron. Soc.*, 402, 226
- Barnes, J., & Hut, P. 1986, *Nature*, 324, 446
- Battaglia, G., Helmi, A., & Breddels, M. 2013, *NewAR*, 57, 52
- Baumgardt, H. 2017, *Mon. Not. R. Astron. Soc.*, 464, 2174
- Belokurov, V., Zucker, D. B., Evans, N. W., et al. 2006, *Astrophys. J.*, 642, L137
- Belokurov, V., Evans, N. W., Irwin, M. J., et al. 2007, *Astrophys. J.*, 658, 337
- Binney, J., & Tremaine, S. 2008, *Galactic Dynamics* (2nd ed., Princeton, NJ: Princeton University Press)
- Blaña, M., Fellhauer, M., Smith, R., et al. 2015, *MNRAS*, 446, 144
- Blanton, M. R., Bershadsky, M. A., Abolfathi, B., et al. 2017, *Astron. J.*, 154, 28
- Bonaca, A., & Hogg, D. W. 2018, *Astrophys. J.*, 867, 101
- Bovy, J. 2015, *Astrophys. J., Suppl. Ser.*, 216, 29
- Bovy, J., Bahmanyar, A., Fritz, T. K., & Kallivayalil, N. 2016, *Astrophys. J.*, 833, 31
- Buttry, R., Pace, A. B., Koposov, S. E., et al. 2021, arXiv e-prints, arXiv:2108.10867
- Cerny, W., Pace, A. B., Drlica-Wagner, A., et al. 2020, arXiv e-prints, arXiv:2009.08550
- Chandrasekhar, S. 1943, *Astrophys. J.*, 97, 255
- Cummings, J. D., Kalirai, J. S., Tremblay, P. E., Ramirez-Ruiz, E., & Choi, J. 2018, *Astrophys. J.*, 866, 21
- Desell, T., Magdon-Ismael, M., Szymanski, B., et al. 2009, in Fifth IEEE International Conference on e-Science, 263–270

- Dotter, A., Chaboyer, B., Jevremović, D., et al. 2008, *Astrophys. J., Suppl. Ser.*, 178, 89
- Erkal, D., Belokurov, V., Laporte, C. F. P., et al. 2019, *MNRAS*, 487, 2685
- Fardal, M. A., van der Marel, R. P., Sohn, S. T., & del Pino Molina, A. 2019, *MNRAS*, 486, 936
- Flaugher, B., Diehl, H. T., Honscheid, K., et al. 2015, *Astron. J.*, 150, 150
- Fryer, C. L., Belczynski, K., Wiktorowicz, G., et al. 2012, *Astrophys. J.*, 749, 91
- Gilmore, G., Wilkenson, M., Kleyna, J., et al. 2007, *NuPhS*, 173, 15
- Grillmair, C. J. 2006, *Astrophys. J., Lett.*, 645, L37
- Grillmair, C. J., Hetherington, L., Carlberg, R. G., & Willman, B. 2015, *ApJL*, 812, L26
- Harris, W. E. 1996, *Astron. J.*, 112, 1487
- Hendel, D., Scowcroft, V., Johnston, K. V., et al. 2018, *MNRAS*, 479, 570
- Hernquist, L. 1990, *Astrophys. J.*, 356, 359
- Hovey, L., Hughes, J. P., & Eriksen, K. 2015, *Astrophys. J.*, 809, 119
- Ibata, R., Diakogiannis, F. I., Famaey, B., & Monari, G. 2021, *Astrophys. J.*, 915, 5
- Ibata, R., Lewis, G. F., Irwin, M., Totten, E., & Quinn, T. 2001, *Astrophys. J.*, 551, 294
- Johnston, K. V., Law, D. R., & Majewski, S. R. 2005, *Astrophys. J.*, 619, 800
- Johnston, K. V., Spergel, D. N., & Haydn, C. 2002, *Astrophys. J.*, 570, 656
- Kafle, P. R., Sharma, S., Lewis, G. F., & Bland-Hawthorn, J. 2014, *Astrophys. J.*, 794, 59
- Kent, J. T. 1982, *J. R. Stat. Soc. Series B (Methodological)*, 44, 71
- Kimmig, B., Seth, A., Ivans, I. I., et al. 2015, *Astron. J.*, 149, 53
- Kirby, E. N., Cohen, J. G., Simon, J. D., & Guhathakurta, P. 2015, *ApJL*, 814, L7
- Kirby, E. N., Cohen, J. G., Simon, J. D., et al. 2017, *Astrophys. J.*, 838, 14
- Koposov, S. E., Rix, H.-W., & Hogg, D. W. 2010, *Astrophys. J.*, 712, 260
- Koposov, S. E., Belokurov, V., Li, T. S., et al. 2019, *MNRAS*, 485, 4726
- Kroupa, P. 2001, *Mon. Not. R. Astron. Soc.*, 322, 231
- Law, D. R., Johnston, K. V., & Majewski, S. R. 2005, *Astrophys. J.*, 619, 807
- Law, D. R., & Majewski, S. R. 2016, *The Sagittarius Dwarf Tidal Stream(s)*, ed. H. Newberg & J. Carlin, Vol. 420 (Springer, Cham), 31–62

- Leonard, P. J. T., Richer, H. B., & Fahlman, G. G. 1992, *Astron. J.*, 104, 2104
- Li, S., Liang, Y.-F., & Fan, Y.-Z. 2021, *Phys. Rev. D*, 104, 083037
- Licquia, T. C., & Newman, J. A. 2015, *The Astrophysical Journal*, 806, 96
- Lilleengen, S., Petersen, M. S., Erkal, D., et al. 2022, arXiv e-prints, arXiv:2205.01688
- Lokas, E. L., Kazantzidis, S., Klimentowski, J., & Mayer, L. 2010, in *American Institute of Physics Conference Series*, Vol. 1240, American Institute of Physics Conference Series, ed. V. P. Debattista & C. C. Popescu, 363–366
- Luri, X., Brown, A. G. A., Sarro, L. M., et al. 2018, *Astron. Astrophys.*, 616, A9
- Martin, N. F., de Jong, J. T. A., & Rix, H. 2008, *Astrophys. J.*, 684, 1075
- Mateo, M., Olszewski, E. W., Pryor, C., Welch, D. L., & Fischer, P. 1993, *Astron. J.*, 105, 510
- McMillan, P. J. 2011, *MNRAS*, 414, 2446
- . 2016, *MNRAS*, 465, 76
- Mendelsohn, E. J., Newberg, H. J., Shelton, S., et al. 2022, *Astrophys. J.*, 926, 106
- Miyamoto, M., & Nagai, R. 1975, *PASJ*, 27, 533
- Newberg, H. J., Willett, B. A., Yanny, B., & Xu, Y. 2010, *Astrophys. J.*, 711, 32
- Newberg, H. J., Yanny, B., Rockosi, C., et al. 2002, *Astrophys. J.*, 569, 245
- Odenkirchen, M., Grebel, E. K., Dehnen, W., Rix, H.-W., & Cudworth, K. M. 2002, *Astron. J.*, 124, 1497
- Plummer, H. C. 1911, *MNRAS*, 71, 460
- Portail, M., Gerhard, O., Wegg, C., & Ness, M. 2017, *Mon. Not. R. Astron. Soc.*, 465, 1621
- Rubner, Y., Tomasi, C., & Guibas, L. J. 2000, *IJCV*, 40, 99
- Sakamoto, T., Chiba, M., & Beers, T. C. 2003, *Astron. Astrophys.*, 397, 899
- Sales, L. V., Helmi, A., Starkenburg, E., et al. 2008, *MNRAS*, 389, 1391
- Schlafly, E. F., & Finkbeiner, D. P. 2011, *Astrophys. J.*, 737, 103
- Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, *Astrophys. J.*, 500, 525
- Schönberner, D., Jacob, R., Lehmann, H., et al. 2014, *Astronomische Nachrichten*, 335, 378
- Shelton, S. 2018, PhD thesis, Rensselaer Polytechnic Institute
- Shelton, S., Newberg, H. J., Weiss, J., et al. 2021, arXiv e-prints, arXiv:2102.07257

- Shipp, N., Drlica-Wagner, A., Balbinot, E., et al. 2018, *Astrophys. J.*, 862, 114
- Smith, R., Fellhauer, M., Candlish, G. N., et al. 2013, *MNRAS*, 433, 2529
- Stetson, P. B. 1987, *Publ. Astron. Soc. Pac.*, 99, 191
- Strader, J., Seth, A. C., Forbes, D. A., et al. 2013, *ApJL*, 775-1, L6
- Strigari, L. E., Bullock, J. S., Kaplinghat, M., et al. 2008, *Natur*, 454, 1096
- Tang, B., Fernández-Trincado, J. G., Geisler, D., et al. 2018, *Astrophys. J.*, 855, 38
- Taylor, M. B. 2005, in *Astronomical Society of the Pacific Conference Series*, Vol. 347, TOP-CAT & STIL: Starlink Table/VOTable Processing Software, *Astronomical Data Analysis Software and Systems XIV*, ed. P. Shopbell, M. Britton, & R. Ebert, 29
- Valenti, E., Zoccali, M., Gonzalez, O. A., et al. 2016, *Astron. Astrophys.*, 587, L6
- Verlet, L. 1967, *PhRv*, 159, 98–103
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *Nature Methods*, 17, 261
- Wang, H.-F., Yang, Y.-B., Hammer, F., & Wang, J.-L. 2022, arXiv e-prints, arXiv:2204.08542
- Weiss, J., Newberg, H. J., Newby, M., & Desell, T. 2018, *Astrophys. J., Suppl. Ser.*, 238, 17
- Wengert, R. E. 1964, *Commun. ACM*, 7, 463
- Willett, B. A. 2010, PhD thesis, Rensselaer Polytechnic Institute
- Willman, B., Geha, M., Strader, J., et al. 2011, *Astron. J.*, 142, 16
- Xue, X.-X., Rix, H.-W., Ma, Z., et al. 2015, *Astrophys. J.*, 809, 144

APPENDIX A

INCOMPLETENESS CORRECTIONS FOR SDSS DATA

Using stars in the range $-21^\circ < \Lambda_{OCS} < -7^\circ$ and $10^\circ < \Lambda_{OCS} < 21^\circ$ with $20.7 < g_{corr} < 21.7$, we fit the stellar density of F-turnoff stars $s(g_{corr})$ to a linear model:

$$s(g_{corr}) = m_s g_{corr} + b_s. \quad (\text{A.1})$$

We exclude stars with $-7^\circ < \Lambda_{OCS} < 10^\circ$ to avoid contamination from the Sagittarius Stream. As seen in Figure 2.2, the line that fits the on-field is different from that of the off-field, so we must fit them separately. Binning over 20 g_{corr} bins ($[g_1, g_2, \dots, g_{20}]$), we calculate the number of stars that fall within each bin, differentiating between the on-field and off-field. After normalizing these star counts, we use the “curve fit” algorithm from SciPy (Virtanen et al. 2020) to calculate the slope ($m_{\text{on/off}}$) and intercept ($b_{\text{on/off}}$) and their respective errors ($\delta m_{\text{on/off}}, \delta b_{\text{on/off}}$) for the on-field and off-field:

$$m_{\text{on}} \pm \delta m_{\text{on}} = (8.16 \pm 1.62) \times 10^{-3}, \quad (\text{A.2})$$

$$m_{\text{off}} \pm \delta m_{\text{off}} = (4.91 \pm 1.68) \times 10^{-3}, \quad (\text{A.3})$$

$$b_{\text{on}} \pm \delta b_{\text{on}} = -0.123 \pm 0.034, \quad (\text{A.4})$$

$$b_{\text{off}} \pm \delta b_{\text{off}} = -0.054 \pm 0.036. \quad (\text{A.5})$$

For each i^{th} Λ_{OCS} bin in the on-field subtending the Λ_{OCS} range $\Lambda_{\text{min},i}$ to $\Lambda_{\text{max},i}$, we assume each completed bin in (Λ, g_{corr}) -space has the form of a trapezoidal prism bounded by the fitted planes described in Equations A.2, A.3, A.4, and A.5. This makes the total volume of such a bin equal to:

$$V_i = (21.4m_s + b_s)(\Lambda_{\text{max},i} - \Lambda_{\text{min},i}). \quad (\text{A.6})$$

To calculate the volume of the bin that is actually filled, we slice this volume using the curve described in Equation 2.4. The infinitesimal volume dv of a trapezoidal cross-section with width $d\Lambda$ is given by:

$$dv = \frac{1}{2}(m_s(g_b(\Lambda) + 20.7) + 2b_s)(g_b(\Lambda) - 20.7)d\Lambda. \quad (\text{A.7})$$

Substituting in $g_b(\Lambda)$, we find that the filled volume of the i^{th} bin is:

$$\begin{aligned} v_i = & \frac{m_s a_g^2}{10} (\Lambda_{max,i}^5 - \Lambda_{min,i}^5) + \frac{m_s a_g b_g}{4} (\Lambda_{max,i}^4 - \Lambda_{min,i}^4) \\ & + \frac{1}{3} \left(m_s a_g c_g + \frac{m_s b_g^2}{2} + b_s a_g \right) (\Lambda_{max,i}^3 - \Lambda_{min,i}^3) \\ & + \frac{1}{2} (m_s b_g c_g + b_s b_g) (\Lambda_{max,i}^2 - \Lambda_{min,i}^2) \\ & + \left(m_s \frac{c_g^2 - 20.7^2}{2} + b_s (c_g - 20.7) \right) (\Lambda_{max,i} - \Lambda_{min,i}). \end{aligned} \quad (\text{A.8})$$

Dividing v_i by V_i gives us the filled ratio k_i of the bin:

$$k_i = \frac{v_i}{V_i}. \quad (\text{A.9})$$

Performing error propagation, we get the error δ_{k_i} :

$$\begin{aligned} \delta_{k_i}^2 = & \frac{1}{V_i^4} \left(\left(\frac{a_g^2}{10} (\Lambda_{max,i}^5 - \Lambda_{min,i}^5) + \frac{a_g b_g}{4} (\Lambda_{max,i}^4 - \Lambda_{min,i}^4) \right. \right. \\ & + \frac{1}{3} \left(a_g c_g + \frac{b_g^2}{2} \right) (\Lambda_{max,i}^3 - \Lambda_{min,i}^3) + \frac{b_g c_g}{2} (\Lambda_{max,i}^2 - \Lambda_{min,i}^2) \\ & + \left. \left. \frac{c_g^2 - 20.7^2}{2} (\Lambda_{max,i} - \Lambda_{min,i}) \right) V_i - 21.4 (\Lambda_{max,i} - \Lambda_{min,i}) v_i \right)^2 \delta_m^2 \\ & + \frac{1}{V_i^4} \left(\left(\frac{a_g}{3} (\Lambda_{max,i}^3 - \Lambda_{min,i}^3) + \frac{b_g}{2} (\Lambda_{max,i}^2 - \Lambda_{min,i}^2) \right. \right. \\ & + \left. \left. (c_g - 20.7) (\Lambda_{max,i} - \Lambda_{min,i}) \right) V_i - (\Lambda_{max,i} - \Lambda_{min,i}) v_i \right)^2 \delta_s^2 \end{aligned} \quad (\text{A.10})$$

After calculating the filled ratio, we approximate the corrected star count (N'_i) and errors ($\sigma_{N'_i}$) using the following formulas:

$$N'_i = \frac{N_i}{k_i}, \quad (\text{A.11})$$

$$\sigma_{N'_i} = \frac{1}{k_i} \sqrt{N_i + \left(\frac{N_i}{k_i} \right)^2 \delta_{k_i}^2}. \quad (\text{A.12})$$

This algorithm is repeated for each bin in the off-field as well.

APPENDIX B

CALCULATING MASS PER TURNOFF STAR USING ISOCHRONES

As a sanity check for our previous calculation, we recalculate the baryonic mass per F-turnoff star using theoretical isochrones of NGC 5053. To perform this calculation, we first need to know three things about NGC 5053: its age, its metallicity ($[\text{Fe}/\text{H}]$), and its alpha abundance ($[\alpha/\text{Fe}]$). From the literature, we find NGC 5053's age to be 12.5 ± 2.0 Gyrs (Arellano Ferro et al. 2010), its $[\text{Fe}/\text{H}]$ to be -2.27 dex (Harris 1996, 2010 edition), and its $[\alpha/\text{Fe}]$ to be 0.2 (Tang et al. 2018). We use isochrone data from the Dartmouth Stellar Evolution Database (DSED; Dotter et al. 2008), selecting the isochrones that best fit the globular cluster's age and chemical abundances. For our calculations, we use the isochrone with $[\text{Fe}/\text{H}] = -2.49$ dex as it was the closest metallicity. However, we will later demonstrate that this deviation does not greatly impact our result or errors by redoing the calculation assuming $[\text{Fe}/\text{H}] = -1.98$ dex.

The initial mass function (IMF) we implement comes from Kroupa (2001), and has the form:

$$\varepsilon(m) = \begin{cases} Am^{-\alpha_0} & 0.01 \leq m < 0.08 \\ A(0.08)^{\alpha_1-\alpha_0} m^{-\alpha_1} & 0.08 \leq m < 0.5 \\ A(0.5)^{\alpha_2-\alpha_1} (0.08)^{\alpha_1-\alpha_0} m^{-\alpha_2} & 0.5 \leq m < 1.0 \\ A(1.0)^{\alpha_3-\alpha_2} (0.5)^{\alpha_2-\alpha_1} (0.08)^{\alpha_1-\alpha_0} m^{-\alpha_3} & m > 1.0, \end{cases} \quad (\text{B.1})$$

where m is the mass of the initial progenitor star in solar masses, A is the normalization constant, $\alpha_0 = 0.3 \pm 0.4$, $\alpha_1 = 1.3 \pm 0.3$, $\alpha_2 = 2.3 \pm 0.1$, and $\alpha_3 = 2.3 \pm 0.2$. The uncertainties in these powers are one-standard-deviation errors.

The formula we use to calculate the stellar mass per turnoff star is fairly straightforward. We take the total stellar mass within the cluster and divide it by the number of F-turnoff stars we find in our isochrone:

$$m_{FT} = \frac{M_{stars}}{N_{FT}} = \frac{\int m \varepsilon(m) dm}{\int \varepsilon(m) dm}. \quad (\text{B.2})$$

Note that since the IMF is in both our numerator and denominator, the normalization

constant A cancels out of our formula, making the normalization we select arbitrary. The easiest quantity to calculate is N_{FT} , the number of F-turnoff stars in our isochrone. We say a star is an F-turnoff star if its $(g - i)_0$ color falls between 0.12 and 0.47, and its g_0 magnitude within the g_0 F-Turnoff Range of the isochrone, using the same convention developed in Section 2.1.4 to define the range.

We define a function we call the F-check function ($F_C(m)$) which outputs 1 if the input initial mass m falls within our F-turnoff range as a result of our isochrone model and 0 otherwise:

$$F_C(m) = \begin{cases} 1 & \text{m is F-turnoff star} \\ 0 & \text{otherwise.} \end{cases} \quad (\text{B.3})$$

Given this function and a list of initial masses from the isochrone model ($[m_1, m_2, \dots, m_N]$), the formula for counting the number of F-turnoff stars becomes straightforward:

$$N_{FT} = \int F_C(m) \varepsilon(m) dm \simeq \sum_{i=1}^N F_C(m_i) \varepsilon(m_i) \Delta m_i, \quad (\text{B.4})$$

where Δm_i is the width of the i^{th} mass bin defined below:

$$\Delta m_i = \begin{cases} m_2 - m_1 & i = 1 \\ m_N - m_{N-1} & i = N \\ \frac{m_{i+1} - m_{i-1}}{2} & \text{otherwise.} \end{cases} \quad (\text{B.5})$$

Calculating the total stellar mass, on the other hand, is a bit more complicated. For stars whose initial mass is larger than the largest mass in the isochrone model ($m > m_N$), their current mass is only but a small fraction of their original mass. This is because such stars have turned into white dwarfs, neutron stars, or black holes, ejecting most of their original mass through their planetary nebulae or supernovae. This ejected mass does not necessarily disappear from the globular cluster, but could be accelerated with enough energy to push it past the cluster's escape velocity. We know the mass of NGC 5053 is $(5.37 \pm 1.32) \times 10^4 M_\odot$, and from our previous fit, we find the scale Plummer radius of NGC 5053 to be 11.7 ± 0.5 pc. Performing a simple back-of-the-envelope calculation of the cluster's highest escape velocity ($v_{e,max} = \sqrt{\frac{2GM}{a}}$) gives an escape velocity of ~ 6.4 km s⁻¹. Since planetary nebulae expand

from their center stars with speeds between 20 and 40 km s⁻¹ (Schönberner et al. 2014), and the shocks of supernovae reach speeds of several thousand km s⁻¹ (Hovey et al. 2015), it is safe to assume that all ejecta of stellar transitions is not gravitationally bound by the cluster and can effectively be ignored in the mass calculation.

Under these assumptions, the total stellar mass can be calculated using the following formula:

$$M_{stars} = \int_{0.01}^{m_N + \frac{\Delta m_N}{2}} m \varepsilon(m) dm + \int_{m_N + \frac{\Delta m_N}{2}}^{100.0} M_R(m, [Fe/H]) \varepsilon(m) dm, \quad (\text{B.6})$$

where $M_R(m, [Fe/H])$ is the remnant mass function, a function that outputs the stellar remnant mass of a star given its initial progenitor mass. We use the models fitted in Cummings et al. (2018) to calculate the remnant mass of white dwarfs and the models from Fryer et al. (2012) to account for the masses in neutron stars and black holes. We cut off our mass calculation at an initial mass of 100 solar masses as the remnant masses past that point become more uncertain due to mass loss and pair-instability supernovae which leave behind no remnant (Fryer et al. 2012). Combining these models, we find the mass of the remnant in solar masses is given by the following formula:

$$M_R(m, [Fe/H]) = \begin{cases} (0.080 \pm 0.016)m + (0.489 \pm 0.030) & m < 2.85 \\ (0.187 \pm 0.061)m + (0.184 \pm 0.199) & 2.85 \leq m < 3.60 \\ (0.107 \pm 0.016)m + (0.471 \pm 0.077) & 3.60 \leq m < 9.0 + 0.9[Fe/H] \\ 1.36 & 9.0 + 0.9[Fe/H] \leq m < 11.0 \\ 1.1 + 0.2e^{(m-11.0)/4.0} - (2.0 + 10^{[Fe/H]})e^{0.4(m-26.0)} & 11.0 \leq m < 30.0 \\ \Theta(m, [Fe/H]) & 30.0 \leq m < 50.0 \\ \max(1.8 + 0.04(90 - m), \Theta(m, [Fe/H])) & 50.0 \leq m < 90.0 \\ \max(1.8 + \log_{10}(m - 89), \Theta(m, [Fe/H])) & m > 90.0, \end{cases} \quad (\text{B.7})$$

where

$$\Theta(m, [Fe/H]) = \min(33.35 + (4.75 + 1.25 \times 10^{[Fe/H]})(m - 34), m - 10^{[Fe/H]/2}(1.3m - 18.35)). \quad (\text{B.8})$$

To propagate the errors in this calculation, we recompute the stellar mass per turnoff star several times, changing the parameters with errors by their respective error. We have 11 quantities with errors in this calculation: the age, the four powers in the IMF, and the six parameters from the white dwarf remnant formula derived by Cummings et al. (2018). For each parameter, we either add its error, subtract its error, or leave the parameter unchanged. We calculate the mass per turnoff star for each permutation (3^{11} total) and treat each one as a separate data point. We then take the average value and calculate the standard deviation of all the points as the error. Using this method, we find that the stellar mass per F-turnoff star we expect to measure in NGC 5053 to be $13.5 \pm 2.7 M_{\odot}$ per F-turnoff star. Using an isochrone with $[Fe/H]=-1.98$ dex, we get an answer of $14.9 \pm 3.5 M_{\odot}$ per turnoff star. Both of these numbers are exceptionally close to the value we measure in Section 2.1.4 and well within the expected errors.

APPENDIX C

ATTEMPTING THE GENERAL EMD DERIVATIVES

While minimizing the EMD, the flow matrix is subjected to the following constraints⁶:

$$f_{ij} \geq 0, \tag{C.1}$$

$$\sum_j f_{ij} = w_{p,i}, \tag{C.2}$$

$$\sum_i f_{ij} = w_{q,j}. \tag{C.3}$$

One property of the optimal flow matrix we know is that opposite off-diagonal elements cannot both be positive since such a matrix would describe a transformation that sends some weight from one bin to another and back again, a clearly non-optimal transformation. Therefore, to simplify the EMD calculation, we redefine the flow matrix f_{ij} in the following way:

$$f_{ij} = \delta_{ij}h_i + \frac{g_{ij} + |g_{ij}|}{2}, \tag{C.4}$$

where the vector \mathbf{h} is the vector containing the diagonal elements of the flow matrix and g_{ij} is defined as the *pseudo*-flow matrix. If the pseudo-flow matrix is antisymmetric, then our flow matrix both satisfies the aforementioned property of optimized flow matrices and the constraint in Equation C.1. Substituting this redefinition into our constraints gives us:

$$g_{ij} = -g_{ji}, \tag{C.5}$$

$$h_k + \sum_j \frac{g_{kj} + |g_{kj}|}{2} = w_{p,k}, \tag{C.6}$$

$$h_k + \sum_i \frac{g_{ik} + |g_{ik}|}{2} = w_{q,k}. \tag{C.7}$$

⁶Conventionally, the second and third constraints here are represented as inequalities. However, for the purposes of this analysis, we treat the diagonal elements of the flow matrix as the amount of weight that is not shifted in the transformation. Since the EMD is independent of the diagonal elements of the flow matrix, this redefinition is valid.

Using the antisymmetric nature of g_{ij} , we can rewrite the last two constraints as:

$$\sum_i g_{ij} = w_{q,j} - w_{p,j} = \Delta_j. \quad (\text{C.8})$$

Since the ground distance matrix d_{ij} is symmetric and g_{ij} is antisymmetric, our optimizing work function becomes:

$$W = \sum_{i>j} d_{ij} |g_{ij}|. \quad (\text{C.9})$$

Let us assume that for two histograms P and Q we already have the optimized flow matrix $f(\tilde{P}, \tilde{Q})_{ij}$. Its corresponding pseudo-flow matrix can be determined from the following formula:

$$g(\tilde{P}, \tilde{Q})_{ij} = f(\tilde{P}, \tilde{Q})_{ij} - f(\tilde{P}, \tilde{Q})_{ji}. \quad (\text{C.10})$$

Now let us apply a perturbation to the histogram P by increasing the bodies in the α th bin by $dN_{p,\alpha}$. The normalized histogram then becomes $\tilde{P} + d\tilde{P}_\alpha$ where:

$$d\tilde{P}_\alpha = \left\{ \left((\Lambda_i, B_i), \frac{1}{\sum_k N_{p,k}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_k N_{p,k}} \right) dN_{p,\alpha} \right) \right\}. \quad (\text{C.11})$$

We expect that a flow matrix that transforms $\tilde{P} + d\tilde{P}_\alpha$ into \tilde{Q} would be very similar to the flow matrix $f(\tilde{P}, \tilde{Q})$. However, if we were to apply matrix $f(\tilde{P}, \tilde{Q})$ onto $\tilde{P} + d\tilde{P}_\alpha$, the resulting transformed histogram would be $\tilde{Q} + d\tilde{P}_\alpha$. Therefore, to find a flow matrix that transforms $\tilde{P} + d\tilde{P}_\alpha$ to \tilde{Q} , we must add a flow matrix $X(d\tilde{P}_\alpha)_{ij}$ that transforms $d\tilde{P}_\alpha$ to the zero histogram. Since $d\tilde{P}_\alpha$ contains only one bin with a positive value (the α th bin) while all the others are negative, one simple flow matrix would transfer all the weight from that bin to all the other bins:

$$X(d\tilde{P}_\alpha)_{ij} = \delta_{i\alpha}(1 - \delta_{ij}) \frac{N_{p,j}}{(\sum_k N_{p,k})^2} dN_{p,\alpha}. \quad (\text{C.12})$$

Therefore, we can describe a flow matrix that transforms $\tilde{P} + d\tilde{P}_\alpha$ to \tilde{Q} as:

$$f(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{ij} = f(\tilde{P}, \tilde{Q})_{ij} + \delta_{i\alpha}(1 - \delta_{ij}) \frac{N_{p,j}}{(\sum_k N_{p,k})^2} dN_{p,\alpha}. \quad (\text{C.13})$$

It should be noted that $f(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{ij}$ is not an *optimal* flow matrix. It merely satisfies the

constraints of the flow matrix as describe in Equations C.1, C.2, and C.3. It does, however, provide us with a good initial guess of the true optimal flow matrix between $\tilde{P} + d\tilde{P}_\alpha$ and \tilde{Q} . Inserting this relation into Equation C.10 nets us our initial guess for the perturbed optimal pseudo-flow matrix:

$$g(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{ij} = g(\tilde{P}, \tilde{Q})_{ij} + \frac{\delta_{i\alpha}N_{p,j} - \delta_{j\alpha}N_{p,i}}{(\sum_k N_{p,k})^2}dN_{p,\alpha}. \quad (\text{C.14})$$

In order to optimize a pseudo-flow matrix, it is necessary for us to understand which parameters of the matrix are independent or dependent on each other. To do this, we must first solve the linear system of equations presented in Equation C.8. To illustrate the process, we will examine the case of a histogram with five bins ($n = 5$) and generalize from there. In the $n = 5$ case, our pseudo-flow matrix has the following form:

$$\mathbf{g} = \begin{pmatrix} 0 & -g_{10} & -g_{20} & -g_{30} & -g_{40} \\ g_{10} & 0 & -g_{21} & -g_{31} & -g_{41} \\ g_{20} & g_{21} & 0 & -g_{32} & -g_{42} \\ g_{30} & g_{31} & g_{32} & 0 & -g_{43} \\ g_{40} & g_{41} & g_{42} & g_{43} & 0 \end{pmatrix} \quad (\text{C.15})$$

Due to its antisymmetry constraint, the pseudo-flow matrix can completely characterized by the elements it has below the zero-diagonal. Before applying the second constraint, we vectorize these elements using the following mapping:

$$g_{ij} = g_k \text{ where } k = \frac{i(i-1)}{2} + j. \quad (\text{C.16})$$

We then set up an augmented matrix to represent the system of linear equations constraining these values:

$$\left[\begin{array}{ccccccccc|c} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta_0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \Delta_1 \\ 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \Delta_2 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & \Delta_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & \Delta_4 \end{array} \right]. \quad (\text{C.17})$$

Within this system of linear equations, we can already identify a fractal pattern emerging

from the augmented matrix. Finding the row-echelon form of this matrix gives us:

$$\left[\begin{array}{cccccccccc|c} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta_0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \Delta_0 + \Delta_1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & \Delta_0 + \Delta_1 + \Delta_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 \end{array} \right]. \quad (\text{C.18})$$

Very clearly, we see a row of zeros on the bottom row of the matrix. Fortunately for us, since \tilde{P} and \tilde{Q} are normalized, the sum of all Δ_i must be zero, meaning we can freely remove that row from the system of equations. Calculating the reduced row-echelon form of the matrix therefore gives us:

$$\left[\begin{array}{cccccccccc|c} 1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -\Delta_1 \\ 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -\Delta_2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & -\Delta_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \sum_{i=0}^3 \Delta_i \end{array} \right]. \quad (\text{C.19})$$

The pivots of this matrix correspond to the elements of the leftmost column of the pseudo-flow matrix (g_{i0}), showing that g_{ij} can be entirely characterized by the inner lower triangular piece of the matrix and the two histograms \tilde{P} and \tilde{Q} . In general, we find that for a histogram length of n bins, the value g_{i0} can be calculated from the following formula:

$$g_{i0} = \delta_{i,n-1} \sum_{k=0}^{n-2} \Delta_k - (1 - \delta_{i,n-1}) \Delta_i - \sum_{j=1}^{i-1} g_{ij} + \sum_{j=i+1}^{n-2} g_{ji}. \quad (\text{C.20})$$

If we were to perturb \tilde{P} by increasing the α th bin by $dN_{p,\alpha}$, we would observe the above equation transforming as such:

$$g'_{i0} = g_{i0} + \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left[(1 - \delta_{i,n-1}) \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) - \delta_{i,n-1} \sum_{\beta=0}^{n-2} \left(\delta_{\beta\alpha} - \frac{N_{p,\beta}}{\sum_{\gamma} N_{p,\gamma}} \right) \right]. \quad (\text{C.21})$$

Evaluating the rightmost summation gives us:

$$g'_{i0} = g_{i0} + \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left[(1 - \delta_{i,n-1}) \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) + \delta_{i,n-1} \left(\delta_{a,n-1} - \frac{N_{p,n-1}}{\sum_{\gamma} N_{p,\gamma}} \right) \right]. \quad (\text{C.22})$$

This further simplifies down to:

$$g'_{i0} = g_{i0} + \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right). \quad (\text{C.23})$$

To find an approximate form for our perturbed *optimal* pseudo-flow matrix $g'(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q})_{ij}$, we plan on taking our original guess $g(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q})$ and applying it to a Newton-Raphson-esque optimization. Specifically, we will be using the “polishing” method we partially covered in Section 4.1 (Equation 4.7). To perform this polishing, we must first calculate the derivatives of the optimization function W with respect to the free parameters of the pseudo-flow matrix. For a free parameter $g_{ij}(i > j > 0)$, we find exactly three terms within W containing that parameter. The partial of W with respect to g_{ij} is therefore give by:

$$\frac{\partial W}{\partial g_{ij}} = d_{ij} \text{sgn}(g_{ij}) - d_{i0} \text{sgn}(g'_{i0}) + d_{j0} \text{sgn}(g'_{j0}). \quad (\text{C.24})$$

Unfortunately, these derivatives are entirely composed of sign functions, whose derivatives are Dirac delta functions, which are not continuous derivatives, complicating our derivative analysis. To properly propagate our derivatives, we replace our sign functions with a steep sigmoid function:

$$\text{sgn}(x) \longrightarrow \lim_{a \rightarrow 0} \frac{x}{\sqrt{x^2 + a^2}} = S_a(x), \quad (\text{C.25})$$

where a is a sufficiently small number (i.e. machine precision). Using this substitution, our first-order derivatives become:

$$\frac{\partial W}{\partial g_{ij}} = d_{ij} S_a(g_{ij}) - d_{i0} S_a(g'_{i0}) + d_{j0} S_a(g'_{j0}). \quad (\text{C.26})$$

Plugging in Equation C.21, we find:

$$\begin{aligned} \frac{\partial W}{\partial g_{ij}} = & d_{ij} S_a(g_{ij}) - d_{i0} S_a \left(g_{i0} + \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) \right) \\ & + d_{j0} S_a \left(g_{j0} + \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{j\alpha} - \frac{N_{p,j}}{\sum_{\gamma} N_{p,\gamma}} \right) \right). \end{aligned} \quad (\text{C.27})$$

Expanding out the infinitesimal terms, we get:

$$\begin{aligned} \frac{\partial W}{\partial g_{ij}} = & d_{ij} \frac{g_{ij}}{\sqrt{g_{ij}^2 + a^2}} - d_{i0} \frac{g_{i0}}{\sqrt{g_{i0}^2 + a^2}} + d_{j0} \frac{g_{j0}}{\sqrt{g_{j0}^2 + a^2}} \\ & - \frac{a^2}{(g_{i0}^2 + a^2)^{3/2}} d_{i0} \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) \\ & + \frac{a^2}{(g_{j0}^2 + a^2)^{3/2}} d_{j0} \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{j\alpha} - \frac{N_{p,j}}{\sum_{\gamma} N_{p,\gamma}} \right). \end{aligned} \quad (\text{C.28})$$

Taking another derivative with respect to the free parameter g_{kl} yields:

$$\begin{aligned} \frac{\partial^2 W}{\partial g_{ij} \partial g_{kl}} = & \delta_{ik} \delta_{jl} d_{ij} \frac{a^2}{(g_{ij}^2 + a^2)^{3/2}} \\ & - (\delta_{il} - \delta_{ik}) d_{i0} \frac{a^2}{(g_{i0}^2 + a^2)^{3/2}} \\ & + (\delta_{jl} - \delta_{jk}) d_{j0} \frac{a^2}{(g_{j0}^2 + a^2)^{3/2}} \\ & + (\delta_{il} - \delta_{ik}) d_{i0} \frac{3a^2 g_{i0}}{(g_{i0}^2 + a^2)^{5/2}} \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) \\ & - (\delta_{jl} - \delta_{jk}) d_{j0} \frac{3a^2 g_{j0}}{(g_{j0}^2 + a^2)^{5/2}} \frac{dN_{p,\alpha}}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{j\alpha} - \frac{N_{p,j}}{\sum_{\gamma} N_{p,\gamma}} \right). \end{aligned} \quad (\text{C.29})$$

We can approximate the true optimal pseudo-flow matrix $g'(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q})_{ij}$ by applying our initial guess $g(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q})_{ij}$ to Equation 4.7:

$$\mathbf{g}'(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q}) \approx \mathbf{g}(\tilde{P} + d\tilde{P}_{\alpha}, \tilde{Q}) - \left(\frac{\partial^2 W}{\partial \mathbf{g} \partial \mathbf{g}} \right)^{-1} \bigg|_{\mathbf{g}(\tilde{P} + d\tilde{P}, \tilde{Q})} \cdot \frac{\partial W}{\partial \mathbf{g}} \bigg|_{\mathbf{g}(\tilde{P} + d\tilde{P}, \tilde{Q})}, \quad (\text{C.30})$$

where the free parameters of $\mathbf{g}(\tilde{P} + d\tilde{P}, \tilde{Q})$ are vectorized using the following mapping:

$$g(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{ij} = g(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_m \text{ where } m = \frac{(i-2)(i-1)}{2} + j - 1. \quad (\text{C.31})$$

We calculate the derivative of our work function evaluated at $\mathbf{g}(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})$, ignoring infinitesimal terms of order two or higher:

$$\begin{aligned} \left. \frac{\partial W}{\partial g_{ij}} \right|_{\mathbf{g}(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})} &= d_{ij} \frac{g(\tilde{P}, \tilde{Q})_{ij}}{\sqrt{g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2}} - d_{i0} \frac{g(\tilde{P}, \tilde{Q})_{i0}}{\sqrt{g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2}} + d_{j0} \frac{g(\tilde{P}, \tilde{Q})_{j0}}{\sqrt{g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2}} \\ &\quad - \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} d_{i0} \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right) \\ &\quad + \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}} d_{j0} \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \left(\delta_{j\alpha} - \frac{N_{p,j}}{\sum_\gamma N_{p,\gamma}} \right) \\ &\quad + \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2\right)^{3/2}} d_{ij} \frac{\delta_{i\alpha} N_{p,j} - \delta_{j\alpha} N_{p,i}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha} \\ &\quad - \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} d_{i0} \frac{\delta_{i\alpha} N_{p,0} - \delta_{0\alpha} N_{p,i}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha} \\ &\quad + \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}} d_{j0} \frac{\delta_{j\alpha} N_{p,0} - \delta_{0\alpha} N_{p,j}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha}. \end{aligned} \quad (\text{C.32})$$

Since $\mathbf{g}(\tilde{P}, \tilde{Q})$ is an optimized pseudo-flow matrix, then by definition, its gradient with respect to its free parameters must be zero. Therefore, the first three terms in Equation C.32 must total to zero as well. This makes our gradient directly proportional to the infinitesimal $dN_{p,\alpha}$:

$$\begin{aligned} \left. \frac{\partial W}{\partial g_{ij}} \right|_{\mathbf{g}(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})} &= \frac{a^2 dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \left[\frac{d_{ij}}{\left(g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2\right)^{3/2}} \frac{\delta_{i\alpha} N_{p,j} - \delta_{j\alpha} N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right. \\ &\quad - \frac{d_{i0}}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} \left[\frac{\delta_{i\alpha} N_{p,0} - \delta_{0\alpha} N_{p,i}}{\sum_\gamma N_{p,\gamma}} + \delta_{i\alpha} - \frac{N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right] \\ &\quad \left. + \frac{d_{j0}}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}} \left[\frac{\delta_{j\alpha} N_{p,0} - \delta_{0\alpha} N_{p,j}}{\sum_\gamma N_{p,\gamma}} + \delta_{j\alpha} - \frac{N_{p,j}}{\sum_\gamma N_{p,\gamma}} \right] \right]. \end{aligned} \quad (\text{C.33})$$

Because of this, our evaluated Hessian is allowed to take on a much simpler form. Since we want Equation C.30 to only have terms that are first-order or lower in $dN_{p,\alpha}$, the only terms in our Hessian that will contribute to our equation are the terms that are of zeroth order in $dN_{p,\alpha}$. Therefore, our evaluated Hessian has the form:

$$\begin{aligned} \left. \frac{\partial^2 W}{\partial g_{ij} \partial g_{kl}} \right|_{\mathbf{g}(\tilde{P}, \tilde{Q})} &= \delta_{ik} \delta_{jl} d_{ij} \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2\right)^{3/2}} \\ &\quad - (\delta_{il} - \delta_{ik}) d_{i0} \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} \\ &\quad + (\delta_{jl} - \delta_{jk}) d_{j0} \frac{a^2}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}}. \end{aligned} \quad (\text{C.34})$$

Using this Hessian, we can show:

$$\begin{aligned} &\left(\left. \frac{\partial^2 W}{\partial g_{ij} \partial g_{kl}} \right|_{\mathbf{g}(\tilde{P}, \tilde{Q})}^{-1} \cdot \left. \frac{\partial W}{\partial g_{kl}} \right|_{\mathbf{g}(\tilde{P}+d\tilde{P}_\alpha, \tilde{Q})} \right)_{ij} = \\ &\frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \sum_{k>l>0} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl}, \end{aligned} \quad (\text{C.35})$$

where

$$\begin{aligned} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij} &= \frac{d_{ij}}{\left(g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2\right)^{3/2}} \frac{\delta_{i\alpha} N_{p,j} - \delta_{j\alpha} N_{p,i}}{\sum_\gamma N_{p,\gamma}} \\ &\quad - \frac{d_{i0}}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} \left[\frac{\delta_{i\alpha} N_{p,0} - \delta_{0\alpha} N_{p,i}}{\sum_\gamma N_{p,\gamma}} + \delta_{i\alpha} - \frac{N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right] \\ &\quad + \frac{d_{j0}}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}} \left[\frac{\delta_{j\alpha} N_{p,0} - \delta_{0\alpha} N_{p,j}}{\sum_\gamma N_{p,\gamma}} + \delta_{j\alpha} - \frac{N_{p,j}}{\sum_\gamma N_{p,\gamma}} \right], \end{aligned} \quad (\text{C.36})$$

and

$$\begin{aligned}
H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl} &= \delta_{ik}\delta_{jl} \frac{d_{ij}}{\left(g(\tilde{P}, \tilde{Q})_{ij}^2 + a^2\right)^{3/2}} \\
&\quad - (\delta_{il} - \delta_{ik}) \frac{d_{i0}}{\left(g(\tilde{P}, \tilde{Q})_{i0}^2 + a^2\right)^{3/2}} \\
&\quad + (\delta_{jl} - \delta_{jk}) \frac{d_{j0}}{\left(g(\tilde{P}, \tilde{Q})_{j0}^2 + a^2\right)^{3/2}}.
\end{aligned} \tag{C.37}$$

The inverse of this matrix can be computationally evaluated.

For $i > j > 0$, we can write the perturbed optimal pseudo-flow matrix as:

$$\begin{aligned}
g'(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{ij} &= g(\tilde{P}, \tilde{Q})_{ij} + \frac{\delta_{i\alpha}N_{p,j} - \delta_{j\alpha}N_{p,i}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha} \\
&\quad - \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \sum_{k>l>0} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl}.
\end{aligned} \tag{C.38}$$

It can then be shown using Equation C.20 for $i > 0$ that the form of $g'(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{i0}$ is:

$$\begin{aligned}
g'(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{i0} &= g(\tilde{P}, \tilde{Q})_{i0} + \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \left(\delta_{i\alpha} - \frac{N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right) \\
&\quad - \sum_{j=1}^{i-1} \left[\frac{\delta_{i\alpha}N_{p,j} - \delta_{j\alpha}N_{p,i}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha} - \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \sum_{k>l>0} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl} \right] \\
&\quad + \sum_{j=i+1}^{n-2} \left[\frac{\delta_{j\alpha}N_{p,i} - \delta_{i\alpha}N_{p,j}}{(\sum_\gamma N_{p,\gamma})^2} dN_{p,\alpha} - \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \sum_{k>l>0} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ji,kl}^{-1} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl} \right].
\end{aligned} \tag{C.39}$$

Simplifying the summations, we find:

$$\begin{aligned}
g'(\tilde{P} + d\tilde{P}_\alpha, \tilde{Q})_{i0} &= g(\tilde{P}, \tilde{Q})_{i0} \\
&\quad + \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \left(\delta_{i\alpha} \frac{N_{p,0} + N_{p,n-1}}{\sum_\gamma N_{p,\gamma}} - (\delta_{\alpha,0} + \delta_{\alpha,n-1}) \frac{N_{p,i}}{\sum_\gamma N_{p,\gamma}} \right) \\
&\quad + \frac{dN_{p,\alpha}}{\sum_\gamma N_{p,\gamma}} \sum_{k>l>0} G_\alpha(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl} \left[\sum_{j=1}^{i-1} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} - \sum_{j=i-1}^{n-2} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ji,kl}^{-1} \right].
\end{aligned} \tag{C.40}$$

Using these equations, we can finally determine the first-order derivatives of the pseudo-flow

matrix for $i > j > 0$:

$$\frac{dg(\tilde{P}, \tilde{Q})_{ij}}{dN_{p,\alpha}} = \frac{\delta_{i\alpha}N_{p,j} - \delta_{j\alpha}N_{p,i}}{(\sum_{\gamma} N_{p,\gamma})^2} - \frac{1}{\sum_{\gamma} N_{p,\gamma}} \sum_{k>l>0} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} G_{\alpha}(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl}, \quad (\text{C.41})$$

and for $i > j = 0$:

$$\begin{aligned} \frac{dg(\tilde{P}, \tilde{Q})_{i0}}{dN_{p,\alpha}} &= \frac{1}{\sum_{\gamma} N_{p,\gamma}} \left(\delta_{i\alpha} \frac{N_{p,0} + N_{p,n-1}}{\sum_{\gamma} N_{p,\gamma}} - (\delta_{\alpha,0} + \delta_{\alpha,n-1}) \frac{N_{p,i}}{\sum_{\gamma} N_{p,\gamma}} \right) \\ &+ \frac{1}{\sum_{\gamma} N_{p,\gamma}} \sum_{k>l>0} G_{\alpha}(\mathbf{g}(\tilde{P}, \tilde{Q}))_{kl} \left[\sum_{j=1}^{i-1} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ij,kl}^{-1} - \sum_{j=i-1}^{n-2} H(\mathbf{g}(\tilde{P}, \tilde{Q}))_{ji,kl}^{-1} \right]. \end{aligned} \quad (\text{C.42})$$

If we only needed the first-order derivatives, this method would provide a decent enough approximation. However, to calculate the Hessian, we would need to compute the derivative of these formulas, which is where we encounter a problem. If we were to take the exact derivative of this *approximate* derivative, we would find that the resulting Hessian would not be symmetric. The symmetry of Hessians is a property we heavily exploit in the implementation of second-order automatic differentiation. To properly calculate this Hessian would require us to take the limiting behavior of iterating the true optimal flow matrix $g'(\tilde{P} + d\tilde{P}, \tilde{Q})$ into Equation C.30 infinitely many times, where calculating one iteration is already computationally intensive.

APPENDIX D

USING LARGE SOFTENING LENGTHS

Since small softening lengths seem to be the cause of the high turbulence, a larger softening length would smooth out our likelihood surface, making it more navigable. Using a larger softening length, however, necessarily sacrifices the accuracy of our gravitational forces within the core of the progenitor, a keystone feature of N -body. An optimal softening length in this context would therefore need to be large enough to produce smooth likelihood surfaces, but small enough as to not sacrifice too many of the close body-body interactions within the progenitor's core. Looking at Figure 5.1, we note that the turbulence of the system starts to grow rapidly when the softening length becomes smaller than 0.01 kpc for a progenitor with a baryonic scale radius of 0.2 kpc.

To quantify how inaccurate our gravitational forces would become using such a softening length, let us look at a body at the center of a sample Plummer sphere with scale radius a and mass M . We can approximately represent the magnitude of the total acceleration felt on the body $a_G(R)$ from bodies within radius R of it using the following formula:

$$a_G(R) = 4\pi Q \int_0^R r^2 \frac{\rho(r)}{r^2} dr = 4\pi Q \int_0^R \rho(r) dr = \frac{3MQ}{a^3} \int_0^R \left(1 + \frac{r^2}{a^2}\right)^{-5/2} dr. \quad (\text{D.1})$$

Q in this equation is a form factor that represents the angular perturbation in the distribution of bodies arranged in a Plummer sphere. If we had a perfectly isotropic distribution of bodies, Q would equal zero. Integrating Equation D.1, we find:

$$a_G(R) = \frac{MQ(3a^2R + 2R^3)}{a^2(a^2 + R^2)^{3/2}}. \quad (\text{D.2})$$

We treat the softening length l_s essentially as the distance by which two bodies no longer exert any force on each other. We can therefore approximate the fraction of the total force on a body in the center of the Plummer sphere using the following equation:

$$\frac{a_G(l_s)}{a_G(\infty)} = \frac{l_s(3a^2 + 2l_s^2)}{2(a^2 + l_s^2)^{3/2}}. \quad (\text{D.3})$$

Plugging in $l_s = 0.01\text{kpc}$ and $a = 0.2\text{kpc}$, we find that such a softening length would reduce the forces within the center of the progenitor by about 7.5%. While this level of accuracy

might appear reasonable, we must remember that uncertainties within our acceleration calculations quickly grow and propagate due to the vast number of acceleration computations that occur within an N-body simulation. For reference, our simulations require double precision floating points to avoid the uncertainties that crop up when computing with single precision numbers. Therefore, the 7.5% reduction in accuracy is far too steep a cost to pay for lower turbulence in the likelihood score.