

On fitting the morphology of simulations of interacting galaxies to synthetic data

by

Graham West

A dissertation submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Computational Science

Middle Tennessee State University
December, 2021

Dissertation Committee:

John Wallin, Chair

Zachariah Sinkala

Abdul Khaliq

William Robertson

ABSTRACT

Gravitational interactions between galaxies represent a fundamental cosmological process. These interactions are responsible for numerous aspects of the formation and evolution of galaxies, such as enhanced and suppressed star formation rates, the development of tidal features, and the feeding active galactic nuclei. Given observational data from systems of interacting galaxies, we seek to determine the values of various dynamical parameters through the optimization of numerical models via genetic algorithms. However, fitting these models can be quite difficult. The core challenges include 1) developing an objective fitness function for quantifying the similarity between model and target images, 2) understanding the inherent symmetries of the dynamical system which promote morphological degeneracies and impede optimization, 3) determining the optimal genetic algorithm operators for the problem.

In this dissertation, we show how naive implementations of fitness functions can yield unintuitive results. We then propose a novel fitness function which was developed by utilizing data from the *Galaxy Zoo: Mergers* project (GZM). The human-scored models obtained from GZM were used to validate our fitness function and led to the adoption of a tidal distortion term which dramatically improved results. We also give a characterization of various geometric and dynamical symmetries inherent within the system and show how the knowledge of these symmetries can be used to reduce the volume of the parameter search space when performing optimization. Lastly, we implement a real-coded genetic algorithm with features designed to address these symmetries. Using simulated target systems with known parameters as a surrogate for observational data, we test our fitness function and genetic algorithm for robustness, accuracy, and convergence. We discuss the link between the degree of tidal distortion present in a target image and the constraints on the dynamical parameters using three different target systems with varying morphology.

As an offshoot of our development of our work on the galaxy optimization problem, we also present a kernel mixing strategy which can be applied in both stochastic optimization and adaptive Markov chain Monte Carlo contexts. The method is flexible and robust enough to handle parameter spaces that are highly multimodal. We provide results from several benchmark problems, incorporating the method into simulated annealing, real-coded genetic algorithm, and adaptive Markov chain Monte Carlo contexts. Results show a significant increase in performance in variants of these methods which incorporate the mixing strategy over those which do not.

ACKNOWLEDGMENTS

First, I would like to thank my advisor John Wallin for his counsel and support on this project as well as his friendship. I am very grateful to my dissertation committee members Zachariah Sinkala, Abdul Khaliq, and William Robertson for their input on how to best present this research. I would especially like to thank Dr. Sinkala for his guidance on our kernel mixing paper. I would like to thank my research collaborator and fellow computational science student Matthew Ogden for his invaluable input on the project. I also extend this gratitude to William (Pete) Smith and Ethan Lawing, both of whom were undergraduate students in physics who chose to work with us for their senior thesis projects. Most of all, I would like to thank my parents Kerry and Kenna West for their unyielding love, sacrifice, and commitment to my academic success.

Table of Contents

List of Figures	xi
List of Tables	1
I Introduction	2
1 Motivation	2
2 Project Goal and Objectives	4
2.1 Project Goals	4
2.2 Project Objectives	5
3 Galaxies	6
3.1 Historical development	6
3.2 Anatomy and classification	8
3.3 Interactions and mergers	10
3.4 Simulation	13
4 Previous Work: SPAM and Galaxy Zoo: Mergers	15
4.1 SPAM: Stellar Particle Animation Module	16
4.2 Galaxy Zoo: Mergers	18
II Current Work	21
5 A robust fitness function and genetic algorithm to morphologically constrain the dynamics of interacting galaxies	22
5.1 Introduction	22
5.2 Quantifying Image Similarity with Fitness Functions	24

5.2.1	Variance of model quality	24
5.2.2	Challenges of similarity scoring	26
5.2.3	Quantifying tidal distortion	27
5.2.4	Complete multi-factor fitness function	29
5.3	Analysis of Morphological Symmetries	31
5.3.1	Causes and effects of symmetry	31
5.3.2	Types of symmetry	32
5.3.3	Geometrical symmetries	33
5.3.4	Dynamical symmetries	39
5.4	Optimization of Fitness Functions	40
5.4.1	Variable Transformations	41
5.4.2	Genetic algorithm for optimizing image similarity	43
5.4.3	Experimental testing of the GA	50
5.4.4	Discussion of GA Results	59
5.5	Conclusions	60
6	A kernel mixing strategy for use in stochastic optimization and adaptive Markov chain Monte Carlo contexts	62
6.1	Introduction	62
6.1.1	The Metropolis method	63
6.1.2	Simulated Annealing	67
6.1.3	Real-coded Genetic Algorithms (GA)	68
6.2	Kernel Mixing Method	70
6.2.1	Motivation and advantages	70
6.2.2	The method	71
6.2.3	Implementation	72
6.3	Numerical Experiments and Results	74

6.3.1	Benchmark 1: Ackley function	76
6.3.2	Benchmark 2: thermal isomerization of α -pinene	81
6.4	Conclusions	87
III Conclusions and Future Work		88
REFERENCES		92

List of Figures

Figure 1	Hubble’s “tuning fork” diagram (from [1]).	8
Figure 2	Left: Spiral NGC 1566. Right: Elliptical NGC 3610. Source: ESA/Hubble & NASA.	9
Figure 3	Hubble Telescope image of the NGC-2623 merger (left) and NGC-4038/39 merger (right). Source: ESA/Hubble & NASA.	11
Figure 4	Left: Hubble Telescope image of the ring galaxy Hoag’s Object. Right: Toomre’s simulations showing the effect on morphology of varying the impact parameter [spiral]. The impact parameter decreases downward while simulation time increases rightward. As can be seen, spirals are created by high-impact collisions while rings are created by low-impact collisions. . . .	13
Figure 5	Depiction of the hierarchical theory of galaxies formation via a merger tree [26].	14
Figure 6	Left: an observational target image. Right: <i>Galaxy Zoo: Mergers</i> models of varying quality. Numbers above the model images are fitness scores between that image and the larger image on the left.	24
Figure 7	Plot of the naive fitness function scores versus the citizen scientist scores. Their Pearson correlation value is $r = -0.189$ (very poor).	25

Figure 8	Demonstration of unperturbed model images. Note that the top model is quite perturbed (as illustrated by the difference plot on the right) while the bottom model barely differs from its unperturbed twin. The right column shows the image difference which allows clear identification of the regions of tidal distortion.	28
Figure 9	Left: citizen scientist scores versus naive fitness scores w/ coloring based on tidal distortion ($r = -0.189$). Right: clone of left plot with low distortion models filtered out ($r = 0.589$). .	29
Figure 10	Left: citizen scientist scores versus complete fitness scores w/ coloring based on tidal distortion ($r = 0.799$). Right: clone of left plot with low distortion models filtered out ($r = 0.799$). .	31
Figure 11	Model plots of 32 unique SPAM transformations. See labels above each image and relative fitness w.r.t. the top left image on the left of each image.	37
Figure 12	Synthetic target 1. Target images are on the top row and the GA's best-fit model images are on the bottom row. The density plots on the left are plotted with a log scale. In the scatter plots, the primary galaxy is blue and the secondary is red. The fitness of the best model image was 0.937.	52

Figure 13	Parameter convergence plots for synthetic target 1 using the GA. The solid lines are the true parameter values while the dotted lines are the best-fit parameter values. The first 11 plots display the transformed SPAM parameters, the next 7 plots display the orbital elements, and the final plot displays the fitness values. Note that t_{min} is in the convention that a value of 10 is interpreted as 10 time units <i>prior</i> to the final state shown in the plots. Also, $\beta = M_{tot}/(v_{min}r_{min}^2)$ is a measure of the impulse applied to the primary by the secondary.	53
Figure 14	Synthetic target 2. Target images are on the top row and the GA's best-fit model images are on the bottom row. The density plots on the left are plotted with a log scale. In the scatter plots, the primary galaxy is blue and the secondary is red. The fitness of the best model was 0.946.	54
Figure 15	Parameter convergence plots for synthetic target 2 using the GA.	55
Figure 16	Synthetic target 3. Left: target/model images from 4,000 particles per galaxy run (fitness: 0.74). Right: target/model images from 10,000 particles per galaxy run (fitness: 0.88). . . .	57
Figure 17	Parameter convergence plots for synthetic target 3. Left: plots for model 1. Right: plots for model 2.	57
Figure 18	Left: standard Gaussian proposal. Right: mixed proposal. . .	72

Figure 19	Results of SA test on the 5D Ackley function. Top left: standard SA. Top right: mixed SA. Bottom left: adaptive SA. Bottom right: mixed/adaptive SA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 50 runs of each method.	77
Figure 20	Results of GA test on 5D Ackley function. Top left: standard GA. Top right: mixed GA. Bottom left: adaptive GA. Bottom right: mixed/adaptive GA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 20 runs of each method.	79
Figure 21	IAC results of MCMC test on the 5D Ackley function. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the average lower bound of the IAC.	80
Figure 22	SRF results of MCMC test on the 5D Ackley function. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the SRF (logarithmic scale).	80
Figure 23	Results of SA test on the α -pinene application. Top left: standard SA. Top right: mixed SA. Bottom left: adaptive SA. Bottom right: mixed/adaptive SA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 100 runs of each method.	83

Figure 24	Results of GA test on the α -pinene application. Top left: standard GA. Top right: mixed GA. Bottom left: adaptive GA. Bottom right: mixed/adaptive GA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 25 runs of each method.	85
Figure 25	IAC results of MCMC test on the α -pinene problem. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the average lower bound of the IAC. . . .	86
Figure 26	SRF results of MCMC test on the α -pinene problem. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the SRF (logarithmic scale).	86

List of Tables

Table 1	Possible symmetries for each parameter (including the auxiliary ψ variables). Integers on top are transformation labels and are used for brevity's sake.	34
Table 2	Shown here are the transformation labels for the geometric symmetries. We omit showing the model images since all 8 are identical.	36
Table 3	List of unique transformation labels.	37
Table 4	Basis of symmetry subspace \mathcal{S}	39
Table 5	Table of parameter values for synthetic target 1 and its best-fit GA model.	53
Table 6	Table of parameter values for synthetic target 2 and its best-fit GA model.	56
Table 7	Table of parameter values for synthetic target 3 and its best-fit GA models.	58
Table 8	Summary of peak average fitness between all four SA and GA cases on the 5D Ackley benchmark problem.	79
Table 9	Data for α -pinene concentrations from Fuguitt and Hawkins [data].	82
Table 10	Summary of peak average fitness between all four SA and GA cases on the α -pinene application.	84

Part I

Introduction

1 Motivation

Gravitational interactions between galaxies represent a fundamental cosmological process. These interactions are responsible for numerous aspects of the formation and evolution of galaxies, such as enhanced and suppressed star formation rates, the development of tidal features, and the feeding active galactic nuclei [16, 17, 20, 31]. The effective impulse applied by an interaction serves to perturb the galaxies out of their equilibrium state, allowing us to probe their structure. In particular, the merger of two galaxies is the central process for hierarchical theories of cosmological structure. They also help us to better understand dark matter and its distribution [**darkmap**].

Due to very strict spatio-temporal limitations on observations, we cannot observe these interactions or mergers over any meaning span of time or from multiple viewing angles. As such, simulation is a necessary and invaluable tool for studying these phenomena. As computing power, software efficiency, and knowledge of extragalactic dynamics have increased in the past decades, simulational research on interacting systems of galaxies has become a significant and successful field of computational astronomy. A particular problem of interest is that of reproducing the morphology and dynamics of observed interacting systems (referred to as target systems). This is done by finding simulation parameters which optimize a fitness function that compares model images with images of the target system. Images of these target systems come in two varieties: 1) brightness maps or (rarely) 2) redshift maps (allowing for the determination of recessional velocity). In the first case, one is only able to fit to the morphology of the target system. This promotes a great deal of degeneracy in the

best-fit solutions since multiple disparate trajectories can produce similar morphologies. In the second case, one can fit to the target system’s morphology *and* dynamics. Though this does not completely eliminate all degeneracy, it drastically reduces it and allows for the estimation of absolute masses and interaction time (instead of merely mass ratios and dynamical time). Unfortunately, this type of data is far more rare than the typical brightness maps.

This dissertation will detail my contributions to the solution of this fitting problem. The next section will provide a more detailed description of the specific goal the project and the key objectives met in order to achieve it. The remainder of the introduction will be comprised of relevant background material in the form of 1) a brief historical survey of extragalactic astronomy, 2) a discussion of the anatomy, classification, and simulation of galaxies and their interactions, and 3) detailed descriptions of our simulation code (SPAM) and available dataset (Galaxy Zoo: Mergers).

After the introduction, I include two papers which detail all of my contributions to the project. The first paper, entitled “- - -,” describes our fitness function for quantifying image similarity, our efforts to reduce morphological degeneracy through the analysis of parameter transformations which preserve morphologies (called symmetries), and our real-coded GA for optimizing the fitness function. The second paper, entitled “- - -,” is unrelated to the main galaxy project. However, during our research, we developed a kernel mixing strategy for optimization and parameter estimation. This was initially only applied to our own problem, but upon realizing its potential, we decided to generalize it and devote an entire paper to it.

Finally, I present some concluding thoughts concerning the state of our project and my contributions presented in this dissertation. I also discuss the future of this project and what next steps we plan to pursue.

2 Project Goal and Objectives

2.1 Project Goals

This dissertation represents the latest step in a long-running project, the goal of which has been to develop an automated pipeline for the fitting of models of interacting systems of galaxies to observational data. A major step in this project was completed by Holincheck et al. [14]. In that paper, they present a semi-automated pipeline which produces a set of best-fit models given an observational image of a target system. This pipeline functions by generating a very large number of models (on the order of 10^5) which are then assessed for fitness through several increasingly strict means. Some of the more obviously erroneous models can be eliminated through a simple calculation of the impulse approximation (a low impulse implies little tidal distortion and is thus likely a bad morphological match). Models which pass this test are then examined by citizen scientists who can approve or reject them (for more information on this process, see Section 4.2). The best models are finally assigned a fitness score based on their frequency of approval by the citizen scientists.

While this method was very successful at finding best-fit models of the target systems tested, it was very time-consuming due to the amount of human labor required by the citizen scientists. Our goal in this dissertation is to automate the citizen scientists' role in the pipeline through the development of various fitness and optimization algorithms. The completion of this goal requires several key objectives to be met (some of which were completed prior to this dissertation, but will be included here for completeness). These objectives come in the form of functional units of software and mathematics which perform specific tasks.

2.2 Project Objectives

Given the goals of the project, we have four specific objectives for this work::

1. Develop a simulation code efficient enough to be used iteratively in an optimization context. This first objective was completed by Wallin et al. [SPAM] via the SPAM code and it was used by Holincheck et al. [14]. As a restricted three-body code, it is $O(n)$ and is therefore efficient enough to justify use in an optimization context where it must be repeatedly called a large number of times.

2. Develop a method for visualizing simulation output to create an accurate model image. Since SPAM outputs a file containing particle positions and velocities, it must be processed appropriately in order to construct a recognizable image of a galaxy interaction. Currently, we apply a simple 2D histogram to obtain this image (with additional logarithmic scaling depending on the context). Assuming that the particle density and the binning resolution are in the proper proportion, this has been a sufficient strategy for our uses with synthetic target images thus far. Another member of our team is developing a more realistic imaging system which utilizes Gaussian blurring and a realistic brightness profile for the galaxies. We will likely adopt this version when moving on to fitting observational targets.

3. Develop a robust fitness function for quantifying the similarity between given target and model images. Previous research has been somewhat restricted by the lack of quality fitness functions. As we will show in Section 5, naive functions such as a pixel brightness RMS error calculation are insufficient to even perform the simple task of distinguishing between significantly tidally perturbed systems and unperturbed disk systems. We will present our new function that incorporates an additional tidal distortion term to achieve superior performance. This function was developed, tested, and validated by checking it against the citizen scientists' scores from the Galaxy Zoo: Mergers dataset.

4. Develop robust methods for optimizing fitness and estimating the various model parameters for a given target image. Due to the many degeneracies and ambiguities created by the restricted 3-body simplifications, the 3D-to-2D projection, and complex nonlinear dynamics of the system, many common optimization and parameter estimation methods are either inefficient or do not converge to the proper solution. For this reason, we developed our several methods specially suited for this task. This includes our kernel mixing strategy and our work on morphological symmetries (allowing us to drastically reduce the volume of parameter space).

The first objective will be discussed in detail in Section 4.1. Our current solution to the second objective is quite simple and so it will not receive very much attention. Objectives three and four will comprise the main body of this dissertation.

3 Galaxies

This section will provide relevant and supplemental background information on galaxies. We will begin with an overview of the historical origin and development of galactic astronomy. Then, we provide discussion of the anatomy and classification of galaxies. Next, we will go over the evolution of galaxies via mergers and interactions between them. Lastly, we will cover methods used to simulate the physics of galaxies and their interactions. We will draw heavily from Mo *et al.* [26] as well as Struck's review [34].

3.1 Historical development

The fields of galactic and extragalactic astronomy are quite recent (approximately 100 years old). This is due largely to the fact that the objects which we now recognize as galaxies were once thought to be merely nebulae residing in our own Milky Way galaxy. In 1781, Charles Messier published his eponymous *Messier Catalog* [23].

Containing 103 different so-called “Messier objects,” this list mis-identified 34 galaxies as one of several types of nebulae or star clusters. In 1888, John Dreyer released his own *New General Catalogue* (NGC) containing over 7,000 objects (with two additional supplements containing over 5,000 objects) [8]. Like Messier and several others before, a multitude of galaxies were mis-identified.

The progression towards the modern understanding of galaxies began in the early 1900’s with several significant observations. In 1912, Slipher observed that the “Andromeda nebula” had a high redshift relative to other known nebulae, implying that it was receding at a high velocity. In 1917, Curtis deduced from the faint novae in the “Andromeda nebula” that it was at the extreme distance of 150,000 parsecs¹ (1 parsec = 3.26 light years) away from Earth [6], much farther than Shapley’s estimated size of the Milky Way ($\sim 30,000$ parsecs) [33]. This resulted in the famed Shapley-Curtis debate over whether Andromeda actually resided in the Milky Way. This debate was resolved by Hubble’s observation of Cepheid variable stars (a common gauge for cosmic distances) in the Andromeda, proving that Andromeda was in fact an entirely separate galaxy.

Soon after the scientific community acknowledged the existence of new, separate galaxies, one of the firsts tasks undertaken was to determine a classification scheme for them and to understand their evolution. We will discuss the classification scheme which arose in more detail in the following section. For now, we will simply state that there are two main classes of galaxies: ellipticals and spirals. In an attempt to articulate the then-current theory of their evolution (it has since been proven incorrect), Hubble developed his famed “tuning fork diagram [18] (see Figure 1). It was believed that ellipticals in the handle of the fork evolved into one of two types of spiral galaxies by flattening due to rotation (it is now believed that ellipticals are

¹Note that this estimate was incorrect. The true distance to the Andromeda galaxy is actually 2,500,000 light years

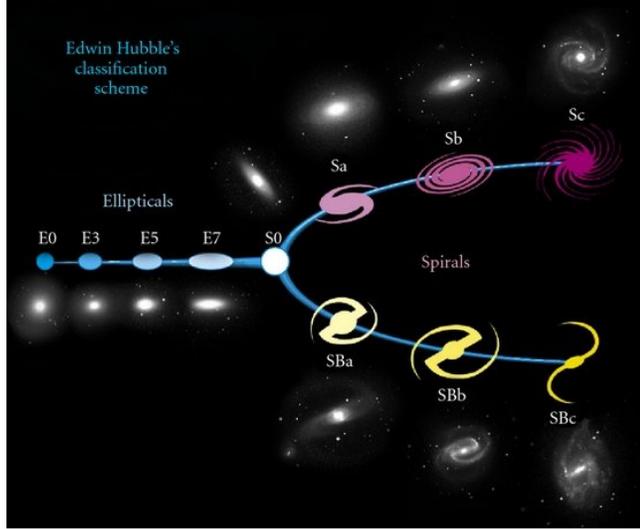


Figure 1: Hubble’s “tuning fork” diagram (from [1]).

the results of a merger of two galaxies [38]). For this reason, ellipticals are often referred to as *early-type* galaxies and spirals as *late-type* (though this is a misnomer, the terminology persists to this day).

3.2 Anatomy and classification

Galaxies are some of the most complex, diverse, and majestic objects in the cosmos. As such, their characterization is dependent on a multitude of properties and parameters. These include luminosity, surface brightness, the gas to stellar mass ratio, redshift, and many others. Since our research is more focused on computational aspects rather than astrophysical and observational aspects, a discussion of a small selection of such terms will suffice. Perhaps the most important for our purposes is *morphology*. Broadly speaking, morphology refers to the shape and structure of a galaxy or interacting system of galaxies. This can include the presence of tidal features such as tails and bridges (these are discussed in the following section) as well as the class to which a given galaxy belongs.



Figure 2: Left: Spiral NGC 1566. Right: Elliptical NGC 3610.
Source: ESA/Hubble & NASA.

Single galaxies may be classified into 3 main types of morphology. *Elliptical* galaxies “are mildly flattened, ellipsoidal systems that are mainly supported by the random motions of their stars” [26]. Ellipticals have little interstellar medium to drive the creation of new stars, causing their star population to be dominated by older stars. Ellipticals are typically the result of a merger of two spiral galaxies (or, rather, any type)².

The second main class of galaxies, *spiral* galaxies, “have highly flattened disks that are mainly supported by rotation” [26]. Spiral anatomy can be divided into three distinct regions. First, the *disk* is their predominant feature. This is the plane in which the majority of the visible galactic mass can be found and is often divided into two or more arms. Second, the *galactic bulge* is found at the center of the galaxy. This is the brightest and densest region of the galaxy and is where the majority of new star formation takes place. Third, the *halo* is the outermost part of the galaxy where stars with only the most highly-eccentric orbits reside. However, despite the apparent lack of baryonic matter, there is still a large amount of dark matter at this distance.

²As an interesting historical note, elliptical galaxies were once thought to be young galaxies, with the justification being that they had not yet evolved into the highly flattened shape of spirals. In reality, the opposite is true.

This *dark matter halo*—which extends to the interior of the galaxy as well—is what is usually referred to by the term halo.

The third class of galaxies contains the *irregular* galaxies. This class is comprised a morphologically diverse cast of galaxies which do not fit neatly into either of the first two categories. Such galaxies are often the result of interactions or mergers which have not yet fully evolved into an elliptical shape.

A common feature in almost all galaxies is a supermassive black hole located in the bulge which consumes and distributes a great deal of matter. These black holes can have masses that range from a few million times more massive than the Sun to more than a billion times more massive than the Sun. The mass of the bulge in spiral galaxies or the mass of the entire elliptical galaxy is correlated with the mass of the central black holes. When black holes are actively accreting material from their neighborhood, they are termed *active galactic nuclei*.

Another important concept is that of the *dynamics* (or *dynamical history*) of a galaxy or system of galaxies. This term (which is fairly general and can be used in several ways) can refer to the either the trajectory of the system through phase space, the evolution of a galaxy’s morphology over time, or specific descriptive parameters (i.e., the time, distance, and velocity at the point of closest approach during an interaction). Though this term is overloaded with connotations, we will try to make our use of it clear from context.

3.3 Interactions and mergers

Like their indwelling stellar systems, galaxies interact via long-range gravitational forces. Though galaxies are large extended bodies, when separated by large distances (many times larger than the galactic radii), they can be treated as point masses since the gravitational field gradient across their surfaces is negligible. However, when one

galaxy nears another, the gravitational field gradient becomes substantial, causing a large imbalance in the attractive force across their disks. This results in perturbations and distortions in their morphologies known as *tidal features*. Common tidal features include tails and bridges. Tails are elongated structures formed by stripping stars and gas from the outer regions of the galaxies. Bridges are are strips of stars and gas which form a connection (or, bridge) between the two galaxies.

A common type of interaction between galaxies is their merger, or collision. This process is initiated when a system of galaxies becomes gravitationally bound. This causes a series of close orbits which induce intense tidal distortions, resulting in the galaxies' eventual coalescence into a single, larger galaxy.



Figure 3: Hubble Telescope image of the NGC-2623 merger (left) and NGC-4038/39 merger (right). Source: ESA/Hubble & NASA.

There are many different types of mergers which may occur:

- *Binary merger*: occurs between only two galaxies
- *Multiple merger*: occurs between 3 or greater.
- *Minor merger*: occurs when there is a significant size difference between the

galaxies

- *Major merger*: occurs when the galaxies are roughly the same size

There also many different types of morphology which can result from mergers. We will describe several:

- *Ring galaxy*: these are defined by a very prominent circular density wave and can result from a nearly on-axis collision between two disk galaxies of similar mass [22]
- *Banana galaxy*: closely related to ring galaxies, these are defined by asymmetric density waves which resemble a banana shape and are formed in much the same way as their ring counterparts but with an increased impact parameter (between the center and edge) [2]
- *Spiral galaxy*: spiral galaxies can be created in much the same way as ring and banana galaxies, but with an even larger impact parameter (near the edge) [spiral]

Mergers also play a vital role in the evolution of star populations within galaxies. As two galaxies collide, their constituent gas and dust are funneled to the inner regions of the galaxies. This seeds regions with sufficient material for star formation to be strongly enhanced (while other regions are conversely suppressed). Mergers can rearrange and enhance star formation across galaxies.

Also, mergers are also critical for modern theories galaxy formation and evolution. In the cold dark matter model (CDM), it is postulated that most galaxies extant today are the result of a sequence mergers of smaller progenitors [34]. This is due to initial fluctuations in the dark matter distribution soon after the Big Bang [7]. Denser regions cluster together into large structures (clumps, filaments, etc.) and attract galaxies which eventually merge.

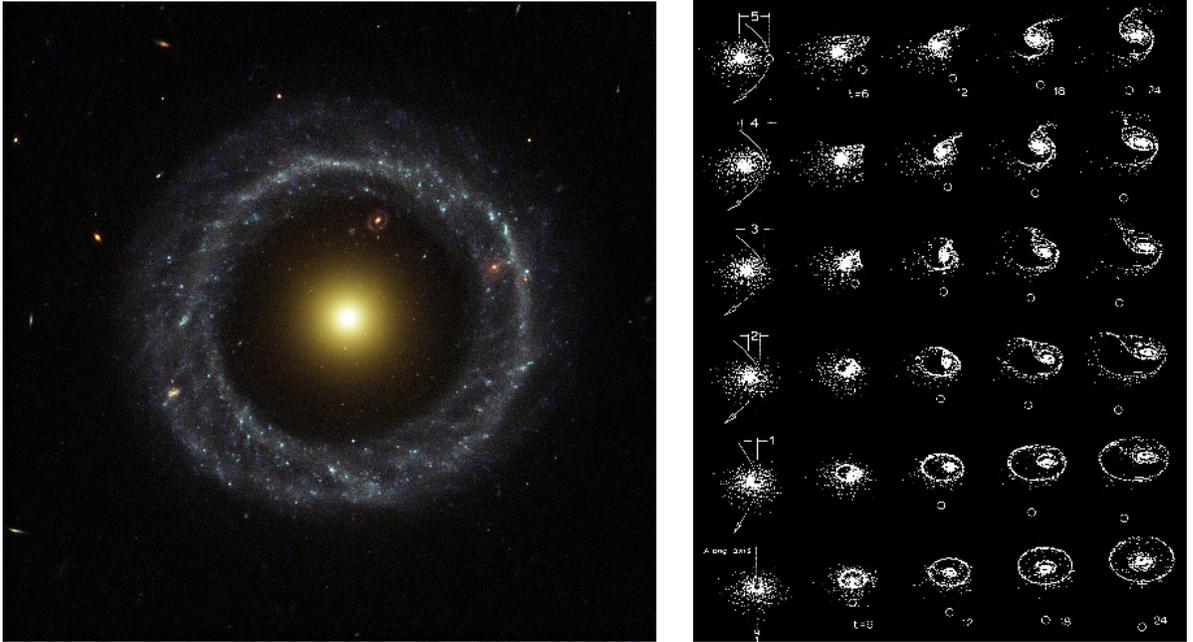


Figure 4: Left: Hubble Telescope image of the ring galaxy Hoag's Object. Right: Toomre's simulations showing the effect on morphology of varying the impact parameter [**spiral**]. The impact parameter decreases downward while simulation time increases rightward. As can be seen, spirals are created by high-impact collisions while rings are created by low-impact collisions.

3.4 Simulation

Astronomical observations of interacting galaxies face many significant challenges. Two which are insurmountable are 1) *the brevity of the human lifespan relative to cosmological timescales* and 2) *the availability of only a single spatial vantage point for any given object*. Given that the timescale of interactions is on the order of 10^8 yr [34], there is no noticeable change in morphology over the lifespan of a single human observer (or even over the entire history of astronomy!). Also, given that the distance to observed interacting systems is on the order of 10^6 ps or greater, we are too far from them for parallax to provide a sufficiently different viewing angle. For these reasons, simulations play an invaluable role in astronomical research, since researchers can

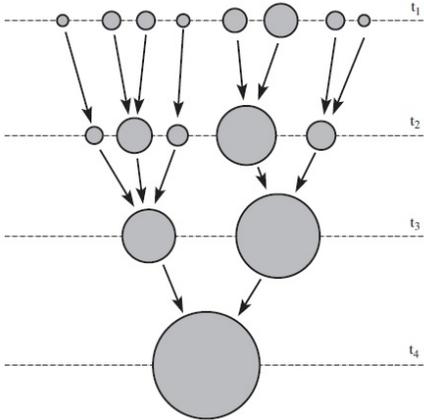


Figure 5: Depiction of the hierarchical theory of galaxies formation via a merger tree [26].

simply construct models of interactions in computer systems³.

Numerous classes of methods for simulating galaxies and their interactions exist in the literature. Due to the nature of gravitation as a pair-wise interaction and the structure of galaxies being primarily a cloud of massive, interacting particles, the most obvious method would be to set up a full n -body code which solves the ODE system

$$\frac{d^2 \vec{r}_i}{dt^2} = \sum_{j \neq i} G \frac{m_j}{\|\vec{r}_i - \vec{r}_j\|^2} \hat{r}_{i \rightarrow j} + \vec{F}_i^{ext} \quad (1)$$

where $i, j = 1, \dots, n$, $\hat{r}_{j \rightarrow i}$ is a unit vector pointing from particle j to particle i , $\|\cdot\|$ is the L^2 norm, G is the universal gravitational constant, and F_i^{ext} represents any additional external forces from, say, dark matter, gas hydrodynamics, etc. While this method is the most accurate and well-suited for high-resolution runs, it is also the most time-consuming since it is $O(n^2)$. Thus, depending on the application and computer hardware, runs with large numbers can take prohibitively long.

Hierarchical tree codes are an alternative to full n -body codes which make some reasonable approximations for the sake of efficiency. Proposed by Barnes and Hut

³This fact was recognized relatively early on in the development of the field of extragalactic astronomy. In 1941, Holmberg performed the first ever n -body simulation in his experiments on the tidal distortions caused by close passage of disk nebulae [16, 17].

[4], these methods work by recursively partitioning the simulation space until every partition has a single particle in it. Upon calculation of the gravitational force for a given particle i , other particles which are far from i but are mutually close can be approximated by a new particle which has their total mass and is located at their center of mass. This provides a significant speedup, granting this method an efficiency of $O(n \log(n))$.

For some applications, even hierarchical tree codes are too slow. In this case, one can use the restricted 3-body method. First developed by Toomre and Toomre [39], this method makes further simplifications for the sake of efficiency. The particles which comprise the galaxy are massless, with the mass being supplied via an additional center of mass point (CoM) with a given mass distribution. The CoM imparts a gravitational acceleration to the massless particles. However, since the particles are massless, the particles do not impart an acceleration to the CoM. In simulations of interactions in which there is more than one galaxy, the CoM's of each galaxy will interact and the massless particles will be attracted to both CoM's. The simplifications made by the restricted 3-body method give it an impressive efficiency of $O(n)$. As such, it is quite useful for running large batches of simulations (making it very suitable for our purposes).

4 Previous Work: SPAM and Galaxy Zoo: Mergers

This section will cover the main resources which were developed prior to my involvement in the project: the SPAM simulation code and the Galaxy Zoo: Mergers dataset. My contributions to the project heavily utilized each of these resources so we will cover them in some detail.

4.1 SPAM: Stellar Particle Animation Module

For our simulation code, we used the restricted three-body SPAM code (Stellar Particle Animation Module) developed by Wallin et al. [41]. The restricted 3-body technique—first developed by Toomre and Toomre [39]—is able to drastically reduce the complexity and runtime of the algorithm with minimal loss of accuracy. As such, it is quite useful for running large batches of simulations (making it very suitable for our purposes). The SPAM code makes several significant simplifications for the sake of runtime, but since the galaxies are not simulated over a long time interval, accuracy is retained. A detailed description of the code is contained in [41], so we will provide only a brief overview of its functionality.

The reference frame of the simulation is set so that one galaxy (called the primary) is initially at the origin with zero velocity and the other galaxy (called the secondary) is given a relative position and velocity. Generally, the larger of the two galaxies is considered the primary. The galaxies themselves are comprised of two parts: a center of mass and a swarm of massless particles. The massless particles only interact with the masses of the two galaxies while the centers of mass interact with both each other and the particles. Initially these particles are placed in a thin disk of circular orbits in specified orbital plane around the center of mass of their respective galaxy.

The list of dynamical parameters which describe the galaxies include the secondary galaxy's positions (x, y, z) and velocities (v_x, v_y, v_z) , both galaxies' masses (M_p, M_s) , their radii (r_p, r_s) , and their orientations in terms of altitude (θ_p, θ_s) and azimuth (ϕ_p, ϕ_s) w.r.t. the z -axis. This gives a total of 14 parameters. Since the x, y position of the galaxies are can be estimated by simple visual comparison with the observational image, there are 12 free parameters in the model.

The gravitational acceleration of the centers of mass and massless particles are

given by

$$\frac{d^2 \vec{R}_p}{dt^2} = G \frac{M_s (|\vec{R}_p - \vec{R}_s|)}{|\vec{R}_p - \vec{R}_s|^2} \hat{r}_{p \rightarrow s} + F_{Dyn,p} \quad (2)$$

$$\frac{d^2 \vec{R}_s}{dt^2} = G \frac{M_p (|\vec{R}_s - \vec{R}_p|)}{|\vec{R}_s - \vec{R}_p|^2} \hat{r}_{s \rightarrow p} + F_{Dyn,s} \quad (3)$$

$$\frac{d^2 \vec{r}_i}{dt^2} = G \frac{M_p (|\vec{r}_i - \vec{R}_p|)}{|\vec{r}_i - \vec{R}_p|^2} \hat{r}_{i \rightarrow p} + G \frac{M_s (|\vec{r}_i - \vec{R}_s|)}{|\vec{r}_i - \vec{R}_s|^2} \hat{r}_{i \rightarrow s} \quad (4)$$

for $i = 1, \dots, N_{particles}$, where R_p and R_s are the center of mass positions, the \vec{r}_i are the positions of the massless particles, and $M_p(r)$ and $M_s(r)$ are the cumulative mass distributions of the two galaxies. These mass distributions are modeled after the bulge, disk, and halo potential described by Hernquist et al. [13]. They are given by

$$M_{bulge}(r) = M_{bulge} \frac{4}{\sqrt{\pi}} \int_0^{r/h_{bulge}} \exp(-x^2) x^2 dx \quad (5)$$

$$M_{disk}(r) = \frac{M_{disk}}{2} \int_0^{r/h_{disk}} \exp(-x) x^2 dx \quad (6)$$

$$M_{halo}(r) = M_{halo} \frac{2\alpha_{halo}}{\sqrt{\pi}} \int_0^{r/r_c} \frac{\exp(-x^2)}{x^2 + q_{halo}^2} x^2 dx \quad (7)$$

$$M(r) = M_{bulge}(r) + M_{disk}(r) + M_{halo}(r) \quad (8)$$

where h_{bulge}, h_{disk}, r_c represent the radii of the bulge, disk, halo (respectively), and q_{halo}, α_{halo} are constants. Note that these functions depend only on the radius r from the galactic center of mass and are, thus, spherically symmetric. Because of this, the distributions are inaccurate in regions above or below the plane of the disk. Since these functions are computationally complex to evaluate and must be called many times throughout a simulation, they are pre-calculated at a set of fixed radial distances and stored. During simulations, their value in between these radii is calculated via linear interpolation.

In addition to gravitational effects, SPAM also includes acceleration due to Chandrasekhar’s dynamical friction [5] (the expression is quite complicated, so we simply refer to it as F_{Dyn} in the above equations). This phenomenon slows objects traveling through a medium of gravitating particles by causing an increase in particle density behind the object, thus increasing the gravitational force in the retrograde direction. Due to the fact that restricted 3-body systems are incapable of replicating this behavior naturally, it must be added artificially. The acceleration due to dynamical friction on the centers of mass is given by

$$\frac{d^2 \vec{R}}{dt^2} = \frac{4\pi G^2 M \ln(\Lambda)}{|v|^3} \left(\operatorname{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right) \vec{v} \quad (9)$$

where $X = v/\sigma$, σ is the velocity dispersion (assuming a Maxwellian velocity distribution), and Λ is a complex function of the distribution of gravitating particles.

Concerning the execution of SPAM, it is quite unique among simulation algorithms. Given a list of **final** parameters, the centers of mass are placed at their respective positions and integrated backward in time, past the point of closest approach to a fixed starting time. Then, the massless particle positions and velocities are initialized. Finally, the system is integrated forward in time back to the original center of mass positions. This initial backward time-stepping combined with the particular reference frame used were chosen in order to maintain alignment between the simulated model and the observational image.

4.2 Galaxy Zoo: Mergers

From a computational perspective, the problem of finding an appropriate function for comparing morphologies is a real challenge. It is difficult to find a method that is robust enough to perform on a level equivalent to that of a human. By nature,

humans are excellent at pattern recognition; and where it can be difficult to “train” an algorithm to spot similarities between objects, humans can do it almost instinctively. For this reason, Holincheck et al. [14] employed the help of thousands of Citizen Scientists to assist the Galaxy Zoo: Mergers project in which they volunteered their pattern recognition abilities to determine best-fit simulations for 62 observational targets.

The Citizen Scientists’ activity was divided into two phases. After roughly 15,000-200,000 simulations were performed for each of the targets, the Citizen Scientists began their work, eventually scoring roughly 200-3,000 simulations per target. Software was specially developed by the Galaxy Zoo: Mergers team to allow the Citizen Scientists to do their work, which was divided into multiple phases. Phase one was the identification of possible morphology matches out of the many thousands of simulations. This phase was divided into three steps, the first of which (called Explore) displayed the target image along with eight simulation images surrounding it in a 3×3 grid. Individuals click on any simulation images they determine to share morphological aspects with the target. The second step (called Enhance) allowed users to alter the values of the SPAM parameters in order to turn an image with relatively similar morphology into one with a greater level of similarity. The third step (called Evaluate) allowed individuals to rank the simulations they chose in order of morphological similarity. Few Citizen Scientists performed this step, so the data gathered from it was not used (phase two essentially replaces the Evaluate step, so there was no loss).

Phase two was the ranking of possible matches determined from phase one. The Citizen Scientists analyzed the morphologies of the simulations via a vote-based tournament scheme called the Merger Wars algorithm. In a typical session, an individual is shown an image of the target (the actual merger) along with two simulations. The

individual then chooses which simulation is most similar to the target, which counts as a “win” for one and a “loss” for the other. (They can also select neither image, thereby affording both images a loss.) Since it is entirely possible for images to be judged poorly, every simulation participates in multiple rounds of the tournament so that a poor round does not have a large impact on the results of the competition. The fitness score of a particular simulation is then calculated as the percentage of “wins” it achieved, with the maximum being 1.

Part II

Current Work

This section is comprised of material to be published in forthcoming papers. The first of these—entitled “A robust fitness function and genetic algorithm to morphologically constrain the dynamics of interacting galaxies”—is a collaborative effort between Matthew Ogden, John Wallin, and myself. We intend to submit it to the journal *Astronomy and Computing*. It presents our efforts to create a fitness function and real-coded genetic algorithm which together can fit simulations of interacting galaxies to data.

The second paper—entitled “A kernel mixing strategy for use in stochastic optimization and adaptive Markov chain Monte Carlo contexts”—is a collaborative effort between Zachariah Sinkala, John Wallin, and myself. We intend to submit it to the journal *Frontiers in Applied Mathematics and Statistics*. It presents and tests a Gaussian kernel mixing strategy which can be applied in both stochastic optimization and Markov chain Monte Carlo contexts.

5 A robust fitness function and genetic algorithm to morphologically constrain the dynamics of interacting galaxies

The material from following section was taken directly from its associated paper with minimal changes. However, due to the fact that we covered much of the relevant introductory material in Section I, we eliminated that material from in order to avoid repetition. Also, since this project has seen many contributors over its lifespan, a brief overview delineating their contributions and my own will be useful. The main resources I have used throughout this project which were developed by others would include 1) the SPAM simulation code and 2) the Galaxy Zoo: Mergers models and scores. Contributions which are purely mine would include 1) the real-coded GA and 2) the symmetry analysis. Our fitness function was a result of group effort, with all members of our team taking part in its development in various ways.

5.1 Introduction

Mergers and interactions are the primary means through which galaxies undergo significant evolutionary change [31]. They produce spectacular tidal features [16, 17] and drive the formation of new stars [20]. However, due to the transience of the human lifespan relative to cosmological time-scales, it is impossible to observe interactions from more than a single cosmological time. Additionally, there is morphological degeneracy due to information loss from projecting a 3-dimensional object onto a two-dimensional image. Because of these limitations, revealing the dynamic and morphological history of a particular interacting system can be very difficult.

The basic physics of gravitational interactions between galaxies can easily be simulated on a computer. However, reproducing the specific morphology and dynamics

of an interacting system can be difficult. Constraining the dynamics purely based on morphology can lead to ambiguity in the optimal solution due to various degeneracies in the dynamical system (there is no unique solution).

The task we face is constraining the dynamical parameters of an interacting system by fitting a simulated model to an observational target image. This problem has seen much attention historically [9, 10, 20, 28, 30, 31, 35–37], but somewhat less in recent years. This is due in part to a lack of data which allows for the development and validation of improved fitness functions to compare a model with a target image [14]. Fortunately, thanks to the *Galaxy Zoo: Mergers* project, human-ranked best-fit models are available for 62 different interacting target pairs. With this dataset, we were able to develop a fitness function which mimics the human rankings from *Galaxy Zoo: Mergers*.

We take a similar approach to Wahde [40], using a genetic algorithm (GA) to fit synthetic target images. Our GA uses a real-coded scheme instead of the binary-coded scheme used by Wahde. We apply our fitness function and GA to several synthetic target systems with varying morphologies and degrees of tidal distortion. We explore how well the dynamics can be reproduced without additional kinematic measurements such as radial velocity profiles across the system.

This paper is formatted in the following fashion. First, we derive our fitness function for quantifying image similarity. We demonstrate the advantages of our function over a naive pixel-to-pixel comparison function. Next, we derive a set of transformations on the simulation parameters which describe the degeneracies inherent in the restricted 3-body formalism and take steps to reduce their negative effects on our GA’s performance. Then, we describe our real-coded GA which we used to optimize morphological similarity. Lastly, we provide results of fitting models of several different synthetic target systems.

5.2 Quantifying Image Similarity with Fitness Functions

We now continue on to discuss the quantification of image similarity. Our goal was to develop an automated method which can perform at a level comparable to the citizen scientists’ pattern recognition displayed in Merger Wars.

5.2.1 Variance of model quality

For any given target image, the quality of models varies greatly. Figure 6 displays a particular target image along with several model images which were obtained from *Galaxy Zoo: Mergers*. The figure displays both the citizen scientist scores and a

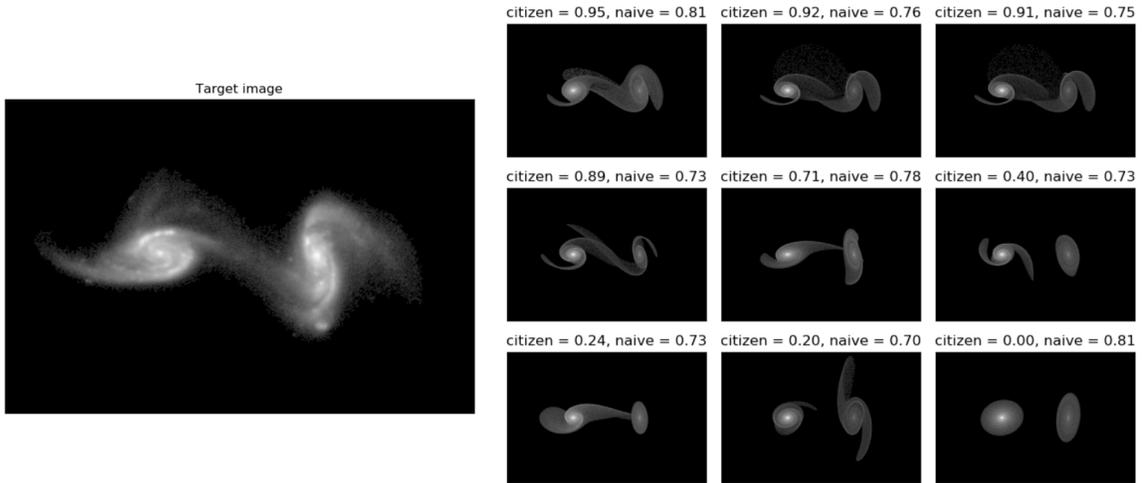


Figure 6: Left: an observational target image. Right: *Galaxy Zoo: Mergers* models of varying quality. Numbers above the model images are fitness scores between that image and the larger image on the left.

naive fitness function score which performs a simplistic, two-dimensional correlation between the target image T and model image M

$$\text{Corr}(T, M) = \frac{\text{Cov}(T, M)}{\sqrt{\text{Var}(T)\text{Var}(M)}} \quad (10)$$

where

$$\text{Cov}(T, M) = \frac{N_x N_y}{N_x N_y - 1} \left(\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} (T_{i,j} - \mu_T)(M_{i,j} - \mu_M) \right) \quad (11)$$

Notice the scores above each model image. The citizen scientist scores consistently assign reasonable and intuitive scores to the models. On the other hand, while some naive fitness function scores seem reasonable—for instance, the top left model (which has the highest citizen scientist score) is better than nearly all other images—there are some clear problems. The bottom right model (to which the citizen scientists gave a score of 0.0) is scored more highly than the top center and top right models despite the fact that it has no tidal features to speak of.

For a more holistic view of the data, Figure 7 displays a plot of the citizen scientist scores vs. the naive fitness scores. In theory, we would like to see a trend which

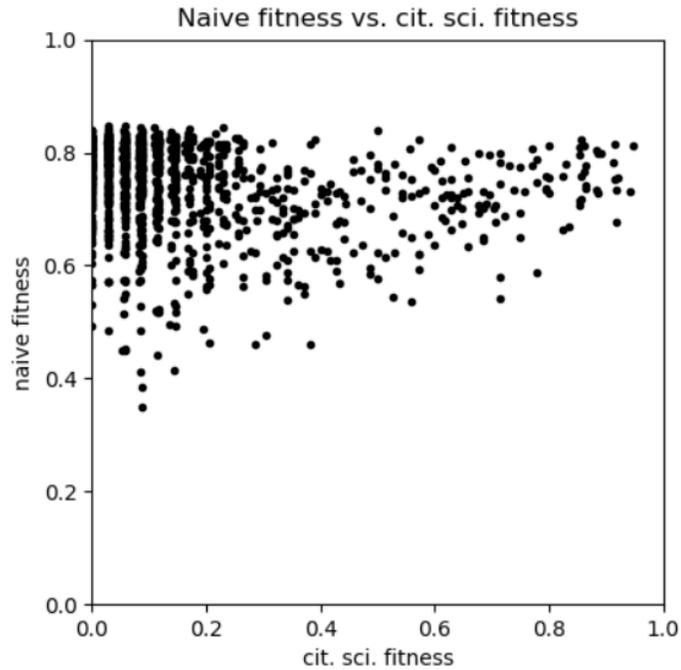


Figure 7: Plot of the naive fitness function scores versus the citizen scientist scores. Their Pearson correlation value is $r = -0.189$ (very poor).

displays a significant, positive correlation between the two scores. However, there is

no clear trend and there is instead a small negative correlation of $r = -0.189$.

Clearly, this simple function is insufficient to serve a robust fitness function. However, after trying roughly dozen other similarity metrics (including pixel RMS error, pixel overlap fraction, and image moments to name a few), we discovered that they all produce the same poor results when compared with the Merger Wars data. This necessitated an analysis of the root cause of the failure of these functions.

5.2.2 Challenges of similarity scoring

Upon close examination of the *Galaxy Zoo: Mergers* data, we discovered several factors which seem to be responsible for the lack of correlation between the human scores and fitness scores:

1: *The majority of models display a low degree of tidal distortion.* Perhaps the most basic challenge is the fact that the overwhelming majority of solutions in SPAM parameter space (and consequently in the data) lead to models which display a little to no tidal distortion. Thus, sampling a model at random from SPAM’s parameter space (which is how *Galaxy Zoo: Mergers* operated) will most likely generate a model with few tidal features.

2: *Fitness scores of models with a small degree of tidal distortion are biased too high.* Compounding the difficulties created by the first challenge, the fitness values of these low-distortion models are consistently higher than would be expected from basic human pattern recognition abilities (see the example discussed above). This is due to two factors. First, by nature of the alignment convention of fixing the primary and secondary galaxies’ x, y position, any possible model is guaranteed to overlap the target to some degree. Second, these regions where the model and target are guaranteed to overlap are the galactic bulges, where there is a high concentration of stars and very little morphological information. Thus, the majority of the score’s

weight comes from regions which have the least impact on the similarity.

3: *Scores of models with significant distortion tend to be biased too low if the distortions are not aligned properly.* Similarly, models which do have a similar morphology but, say, the bridges and tails are rotated by several degrees are scored lower than would be expected from basic human pattern recognition. This is because small misalignments have a tremendous impact on naive pixel-to-pixel comparisons.

4: *There is a a considerable amount of degeneracy within the parameter space such that different values of the input parameters can produce nearly or even identical images.* This is due to a variety of factors and we will devote an entire section to this idea later.

In light of these challenges, a key observation we made concerning the behavior of our fitness functions was that a single function which performs a per pixel comparison of the target and model images is insufficient for realistic performance. Combining this with the fact that several of the challenges listed arise due to a particular model's degree of tidal distortion, our breakthrough occurred when we realized that the best solution would be to simply quantify the degree of tidal distortion present in a model and include that as a term in our fitness functions.

5.2.3 Quantifying tidal distortion

The common thread connecting each of these challenges is the degree of tidal distortion present in a model image. Thus, we decided to incorporate a measure of the distortion into our scoring function. Since this solution involves multiple terms, let us take a moment to distinguish between the two. The *TM-score* (target-model) is the function which calculates a base similarity score between the target and model images. The naive correlation mentioned earlier is an example of a TM-score. The *MU-score* (model-unperturbed) is the function which quantifies the degree of tidal

distortion in the model. This is done via comparison between the model image and a third image called the *unperturbed* model image. This new image is created from SPAM output by taking the unperturbed disk galaxies and translating them to the final position defined by the SPAM parameters. These galaxies will be in alignment with their perturbed model counterparts, but will not display any tidal features (see Figure 8 for examples of unperturbed images). It is then possible to use the same

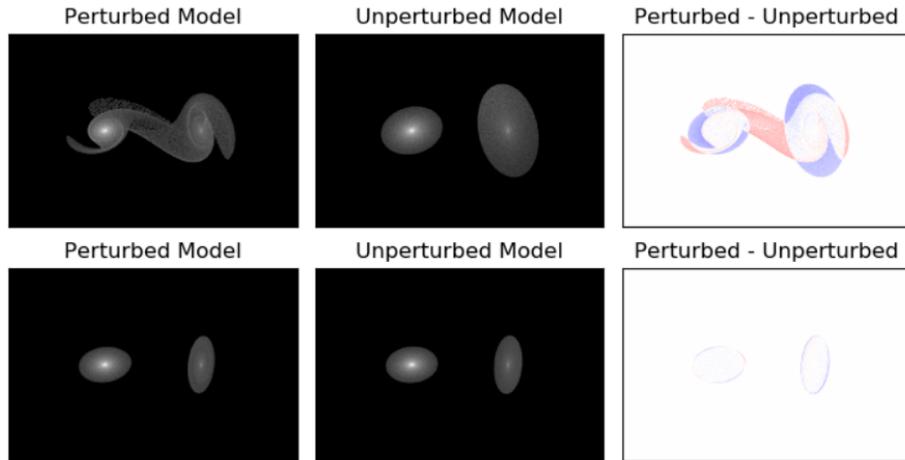


Figure 8: Demonstration of unperturbed model images. Note that the top model is quite perturbed (as illustrated by the difference plot on the right) while the bottom model barely differs from its unperturbed twin. The right column shows the image difference which allows clear identification of the regions of tidal distortion.

function which was used in the TM-score to calculate the degree of tidal distortion. A high score (high similarity) would represent a low degree of distortion since the model and unperturbed images are similar.

We can now reproduce the plot comparing the Merger Wars scores and the naive correlation, but now including the quantified tidal distortion (also calculated via the naive correlation) as an extra color axis (see Figure 9). It is now clear that the lack of a trend is due to a large number of low-distortion models which the citizen scientists correctly scored low but our naive fitness function incorrectly scored high. If we filter out models with a naive correlation with their unperturbed counterpart of 0.85 or

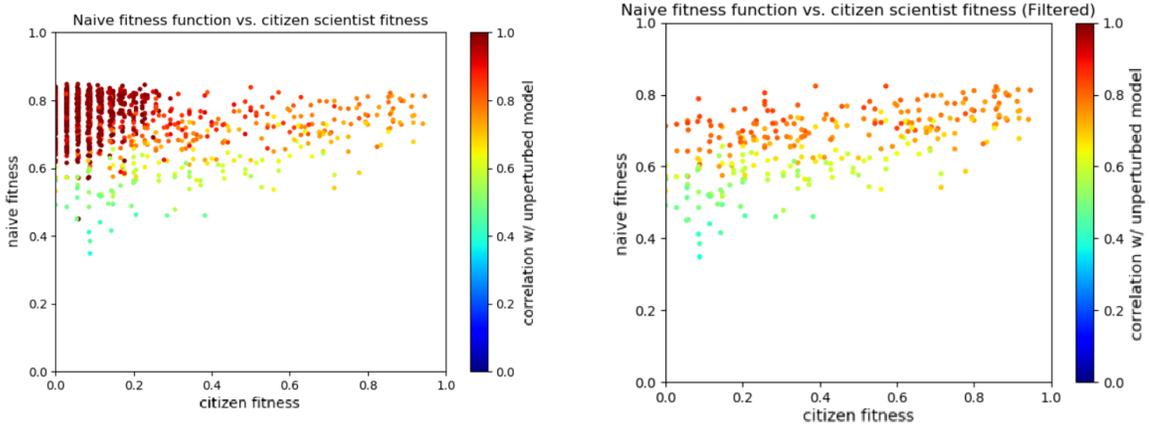


Figure 9: Left: citizen scientist scores versus naive fitness scores w/ coloring based on tidal distortion ($r = -0.189$). Right: clone of left plot with low distortion models filtered out ($r = 0.589$).

greater (the very low-distortion models), the trend is clearly visible and the initial correlation of $r = -0.189$ between the Merger Wars data and the naive score improves to $r = 0.589$.

5.2.4 Complete multi-factor fitness function

We are now in a position to discuss the final formulation of our fitness function. First, we replace the naive two-dimensional correlation with a weighted correlation using logarithmic intensity scaling

$$F_1(T, M, w) = \text{Corr}(\log(1 + T), \log(1 + M), w) \quad (12)$$

where the weights are defined by

$$w_{i,j} = |\log(1+T_{i,j})-\log(1+M_{i,j})|+|\log(1+T_{i,j})-\log(1+U_{i,j})|+|\log(1+M_{i,j})-\log(1+U_{i,j})|+c \quad (13)$$

These changes are designed to lower the significance of regions which are either too bright or display little tidal distortion, respectively. We then multiply the tidal dis-

tortion term to get our complete fitness function

$$f(T, M, U_M, w) = F_1(T, M, w) \cdot F_2\left(F_1(\hat{M}, \hat{U}_M, \mathbb{1}), F_1(\hat{T}, \hat{U}_M, \mathbb{1})\right) \quad (14)$$

$$F_C(T, M, U_M, w) = \begin{cases} f(T, M, U_M, w) & f(T, M, U_M, w) > h \\ h & f(T, M, U_M, w) \leq h \end{cases} \quad (15)$$

where $\mathbb{1}$ represents a weighting of all 1's and

$$F_2(a, b) = \begin{cases} a & a \leq b \\ (1 - a)/(1 - b) & a > b \end{cases} \quad (16)$$

Here, $F_1(\hat{M}, \hat{U}_M, \mathbb{1})$ is the model's degree of tidal distortion and $F_1(\hat{T}, \hat{U}_M, \mathbb{1})$ is an approximation of the target's distortion. Since there is no unperturbed target image U_T , we use the model's in its place. We pass the fitness function through a threshold given by Equation 15 with typical values being $h = 0.01$. This is done so that 1) the correlation never returns a negative value—which would ruin the roulette selection step in the GA—and so that 2) the population doesn't contain individuals which have a prohibitively low selection probability. Note that hatted image variables (e.g., \hat{M}) are truncated binary images where any pixels above a set threshold are set to 1 while any below are set to 0 (this is to greater emphasize tidal features when quantifying the degree of tidal distortion). Also, for the TM-score, we use a fixed image window, but for the MU- and TU-scores, we allow the image window to vary in size with the model so that all tidal features may be captured.

We can again recreate the correlation plots with the complete fitness function F_2 . Using the complete fitness function with tidal distortion term, we now see a very visible trend and achieve a significant positive correlation of $r = 0.799$ for both the

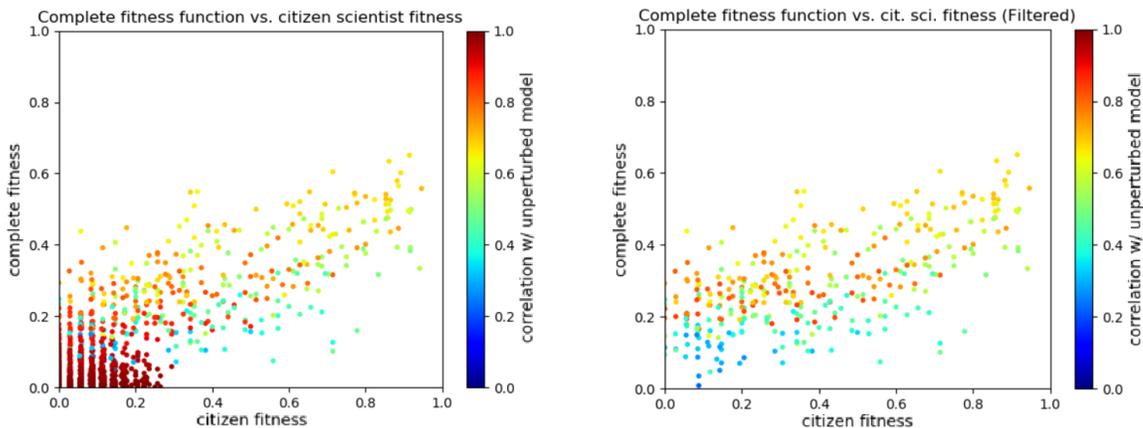


Figure 10: Left: citizen scientist scores versus complete fitness scores w/ coloring based on tidal distortion ($r = 0.799$). Right: clone of left plot with low distortion models filtered out ($r = 0.799$).

filtered and unfiltered data.

5.3 Analysis of Morphological Symmetries

We now move on to discuss the various transformations which, when applied to the SPAM parameters, produce similar or identical morphologies (or symmetries). We will begin with an overview of symmetry, discussing what factors cause it to occur, the challenges it creates, and some definitions of different types of symmetry. We will then derive the transformations which represent the symmetries and discuss how to mitigate the effects of these symmetries on optimization.

5.3.1 Causes and effects of symmetry

The presence of parameter degeneracy and morphological symmetry in the modeling of systems of interacting galaxies causes the following major problem: given a target image of an interacting system that one wishes to fit, there is not a single unique solution which describes it. This is due to 1) degeneracy in the representation of solutions due to the parameters themselves and 2) the fact that target images are two-

dimensional projections (x, y) of a six-dimensional space (x, y, z, v_x, v_y, v_z) , causing otherwise distinct images to look identical. At times, we have additional v_z images (adding a third dimension) which can reduce the degeneracy, but this is rarely the case. Therefore, we will assume that only the x, y projection is included in any given data.

From an optimization perspective, the problems above affect the parameter space in such a way that it can be described by a single unique region which is mirrored into other regions across several different parameter axes. Thus, to use the full parameter space (including these mirrored copies) is vastly inefficient. As we will see, one of the consequences of our investigation of these symmetries is our ability to limit the search to a single unique region, thus increasing efficiency.

5.3.2 Types of symmetry

Symmetries may be grouped into several classes depending on their properties and how they are created. Let us provide some useful terminology so that we may distinguish between them.

Symmetry. By the term symmetry, we mean a transformation between sets of SPAM parameters which preserves an interacting system's morphology.

Projective symmetry. This is a transformation which appears to be a symmetry when only viewing the two-dimensional x, y image projection. In reality, the transformation may not be a true symmetry, but due to the projection, it is impossible to determine without additional information.

Geometric symmetry. Geometric symmetries are transformations which are achieved via the composition of rotations or reflections of the interacting system about various axes. One easily discernible property of them is that they preserve both the kinetic and potential energy.

Dynamic symmetry. Dynamic symmetries are created by altering the parameters in such a way that both the time-scale and interaction strength are altered but the net effect on morphology cancels out. These do not preserve the energy, but they preserve the ratio of kinetic to potential energy (K/U). This has the effect of scaling the Lagrangian $\mathcal{L} = K - U$ and thus preserving the path of least action (or, orbital trajectory).

We will now devote a section each to geometric and dynamic symmetries.

5.3.3 Geometrical symmetries

We will begin our investigation of symmetries with geometric symmetries. We will describe our convention for labeling the various symmetries, derive the geometric symmetries, and adjust the parameter space to account for them.

Creating and labeling symmetries The first matter to consider is identifying which SPAM parameters contribute to the construction of geometric symmetries. It is clear that the masses and radii cannot contribute to any geometric symmetries since mass is not a geometric quantity and galaxies with different radii cannot overlap perfectly. This leaves the positions, velocities, and orientation angles. It is also clear that the secondary x, y positions cannot contribute since they must remain fixed in order to maintain image alignment. As a consequence, v_x, v_y cannot contribute either since changing their values without also changing the corresponding position values would drastically alter the time, distance, and position of closest approach, thus changing the degree of tidal distortion and the location of tidal features. This leaves only 6 SPAM parameters which contribute to the construction of geometric symmetries: $z, v_z, \phi_p, \phi_s, \theta_p, \theta_s$. It is not clear which combinations of transformations of these parameters will, in fact, result in symmetries, so we will consider the set of all possible symmetries for each parameter.

Before doing so, it should be stated that (unlike typical spherical coordinates) the θ vales range from $[0^\circ, 360^\circ]$ as opposed to $[-90^\circ, 90^\circ]$. This convention was chosen because the rotation direction of the galaxies in SPAM is always counter-clockwise with respect to the normal direction defined by ϕ, θ . Because of this, the θ 's have twice the number of possible symmetries as the rest of the parameters. In order to have an equal number of symmetries for each parameter (which makes the following construction significantly simpler and convenient to use), we introduce two auxiliary variables ψ_p, ψ_s which define the direction of rotation for each galaxy. A value of $+1$ will represent counter-clockwise rotation while a value of -1 will represent clockwise rotation. These parameters will be incorporated into SPAM by simply adding 180° to their respective θ values if $\psi = -1$ and by doing nothing to the θ values if $\psi = +1$. Given the auxiliary ψ variables, we have the set of all possible symmetries given in Table 1. Interpreting these transformations is quite simple. For example,

Table 1: Possible symmetries for each parameter (including the auxiliary ψ variables). Integers on top are transformation labels and are used for brevity's sake.

0	1
z	$-z$
v_z	$-v_z$
ϕ_p	$\phi_p + 180$
ϕ_s	$\phi_s + 180$
θ_p	$-\theta_p$
θ_s	$-\theta_s$
ψ_p	$-\psi_p$
ψ_s	$-\psi_s$

$[1, 0, 0, 0, 0, 0, 0, 0]$ (or $[-z, v_z, \phi_p, \phi_s, \theta_p, \theta_s, \phi_p, \phi_s]$) is shorthand for a transformation which negates the z parameter. Individual singleton transformations like this one do not necessarily produce symmetries, but combinations of them do. Together, this combinations produce a total of $2^8 = 256$ different transformations.

Several key observations to make about these transformations are 1) there are

only 2 states for each parameter (0 representing the non-transformed value and 1 representing the transformed value), 2) applying the 1 transformation twice will result in the 0 transformation (since they are all derived from either negating the parameter value or adding 180°) and 3) all of these transformations commute. Furthermore, we can represent this space of transformations as an 8-dimensional vector space V over the binary field $K = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$ of integers mod 2. The proof that this is indeed a vector space is trivial and will therefore be omitted.

Testing and analysis of symmetries In order to test all of these transformations for symmetry, we took the SPAM parameters from a particular *Galaxy Zoo: Mergers* model which had clear tidal features, applied all 256 transformations to it, generated a model for each, and calculated the relative similarity of each of the models w.r.t. the untransformed model. Models whose similarity was ≈ 1 were checked visually for symmetry (note that SPAM’s random seeding of the massless particles makes it impossible to get an exact match). After checking the models, we determined that there were 8 models (one of these being the untransformed model itself) whose mutual relative fitness was ≈ 1 followed by a significant drop in fitness for the remaining models. Thus, there is an 8-fold degeneracy due to geometric symmetries. Table 2 gives the labels for these transformations.

Let us examine these transformations. The first transformation is the identity transformation and is a trivial symmetry. The next two transformations are equivalent to a 180° rotation of the galaxies about their rotation axes (there is a separate symmetry for each galaxy). Since the galaxies are perfectly circular disks, this transformation produces no difference in morphology. The fourth transformation is a projective symmetry which performs a mirroring of the entire system in the z direction. Although this transformation preserves morphology up to projection, if v_z data is known, then this transformation produces a distinct system since all v_z values are

Table 2: Shown here are the transformation labels for the geometric symmetries. We omit showing the model images since all 8 are identical.

z	v_z	ϕ_p	ϕ_s	θ_p	θ_s	ψ_p	ψ_s
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	0	1	0	1	0	0
1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0
1	1	0	1	1	0	0	0
1	1	1	0	0	1	0	0
1	1	0	0	1	1	0	0

negated. The remaining four transformations are produced via compositions of the previous (note that a composition of symmetries is itself a symmetry). This implies that this set of 8 symmetries (let's call it S) is a subspace of V . This is also trivial to show and will be omitted. Note that while the original space V is 8-dimensional, this subspace S is 3-dimensional with basis vectors given by the second, third, and fourth vectors in Table 2.

Now, since these 8 transformations are symmetries, there should be $256/8 = 32$ truly unique models out of all that were tested. To search for the 32 unique models, we ran all comparisons of the 256 models and grouped them into subsets which all mutually had relative fitnesses of ≈ 1 , obtaining the images in Figure 11 and the labels in Table 3.

There are several points to make about these transformations. First, there is a clearly visible pattern, that being the fact that this set of transformations includes all possible combinations of transformations of $v_z, \theta_p, \theta_s, \psi_p, \psi_s$. This implies that this set (call it U) is also a 5-dimensional subspace of V (again, we will omit the proof). Note that due to the 8 symmetries, each of these unique transformations have 8 different SPAM parameter representations. Second, many of these unique models are what we might call *half symmetries*, i.e., transformations which only affect one galaxy (e.g.,

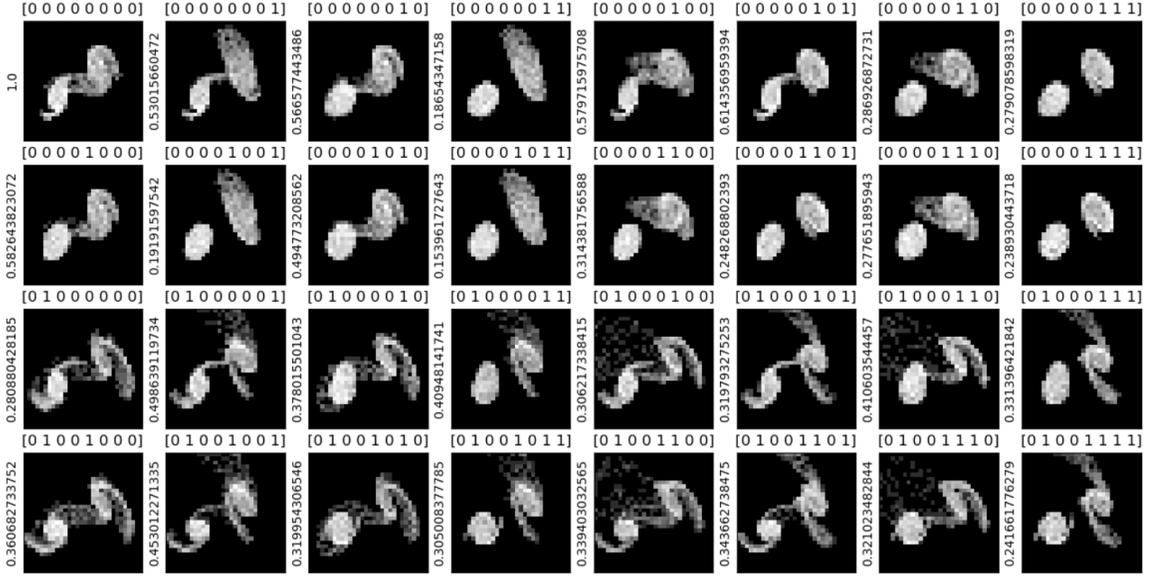


Figure 11: Model plots of 32 unique SPAM transformations. See labels above each image and relative fitness w.r.t. the top left image on the left of each image.

Table 3: List of unique transformation labels.

z	v_z	ϕ_p	ϕ_s	θ_p	θ_s	ψ_p	ψ_s
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1
0	0	0	0	1	0	1	0
0	0	0	0	1	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	1
0	0	0	0	1	1	1	0
0	0	0	0	1	1	1	1
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	0	1	0	0	0
0	1	0	0	1	0	0	1
0	1	0	0	1	0	1	0
0	1	0	0	1	0	1	1
0	1	0	0	1	1	0	0
0	1	0	0	1	1	0	1
0	1	0	0	1	1	1	0
0	1	0	0	1	1	1	1

see the first two images in Figure 11). Many of these are due to the dynamics of the restricted three-body formalism. Third, we also have examples of transformations

which appear to be symmetries in some cases and not in others. Observe Figure 11 again and note the model on the top right (with label $[0,0,0,0,0,1,1,1]$) and the model beneath it (with label $[0,0,0,0,1,1,1,1]$). These two models look nearly identical, which would lead one to believe that $[0,0,0,0,1,0,0,0]$ (the relative transformation between them) is a symmetry. However, now compare the model on the top left (with label $[0,0,0,0,0,0,0,0]$) and the model beneath it (with label $[0,0,0,0,1,0,0,0]$, which is the transformed model). Although these two models have the same relative transformation between them as the previous two, this transformation is clearly just a half symmetry. So, depending on the SPAM parameter values, some transformations can appear to be symmetries when they are not. This further increases the degeneracy in an unpredictable way.

Implications of geometric symmetries for SPAM parameter search ranges

A fundamental consequence of the existence of these symmetries is that there are separate regions of the SPAM parameter space which are identical copies of each other, and therefore can be safely ignored during optimization. How do these symmetries specifically affect the SPAM parameter space and create these copied regions? Let us first look at some simplified, contrived examples.

Suppose we are optimizing a function defined on a 2-dimensional space with a symmetrical transformation defined by $[x, -y]$ or $[0, 1]$ (so that negating y gives the same function value). Then this means that each point in quadrant I is identical to a point in quadrant IV (similarly for II and III). Therefore, we can safely ignore either all $y < 0$ or all $y > 0$ (the choice does not matter since they are identical). Now, suppose we are given another function with a symmetrical transformation defined by $[-x, -y]$ or $[1, 1]$. Then this means that each point in quadrant I is identical to a point in quadrant III (similarly for II and IV). In this case, we can still only ignore a single half-plane, but we have more choices of which to ignore. We could either ignore

$x > 0, x < 0, y > 0$, or $y < 0$. Now, suppose we are given another function with two symmetrical transformations defined by $[1, 0]$ and $[0, 1]$. Then this means that each point in quadrant I is identical to a point in quadrant II (via the first transformation) *and* is also identical to a point in IV (via the second transformation). *Moreover*, each point in quadrant I is identical to a point in quadrant III via the composition of both transformations ($[1, 1]$). In this case, we can ignore multiple regions—say, $x < 0$ *and* $y < 0$ —and thereby retain only a single unique quadrant (in this case, quadrant I).

What we have learned here is that *for each linearly independent symmetrical transformation, we are able to restrict one of the parameters involved in that transformation to strictly its positive values (or negative values)*. Now, since the subspace of symmetries S is 3-dimensional, we can reduce the range of 3 SPAM parameters (note that a parameter’s range cannot be restricted twice). See Table 4 for a recap of the basis of S . As an example, suppose we chose to restrict z, θ_p, θ_s . This would reduce

Table 4: Basis of symmetry subspace S .

z	v_z	ϕ_p	ϕ_s	θ_p	θ_s	ψ_p	ψ_s
0	0	1	0	1	0	0	0
0	0	0	1	0	1	0	0
1	1	1	1	0	0	0	0

the volume of the parameter search space by up to a factor of $2^3 = 8$. This reduction has a tremendous impact on search efficiency during optimization.

5.3.4 Dynamical symmetries

We will begin our discussion of dynamic symmetries just as we did for geometric symmetries: by identifying which parameters are involved. First, galactic radii and orientation angles are irrelevant to the concept of dynamic symmetries (at least for the restricted 3-body case), so they may be ignored. Also, as before, x, y must remain fixed for image alignment. In addition, changing z would not preserve the

orbital trajectory (a necessary condition for dynamic symmetries) or tidal features. This leaves v_x, v_y, v_z, m_p, m_s . Now since the orbit must be preserved, the velocity vector \vec{v} can only be scaled, not rotated. However, in order to preserve the orbit, we must also scale the masses m_p, m_s . This scaling must follow the following form: $[v_x, v_y, v_z, m_p, m_s, t] \rightarrow [\sqrt{\alpha}v_x, \sqrt{\alpha}v_y, \sqrt{\alpha}v_z, \alpha m_p, \alpha m_s, t/\sqrt{\alpha}]$ for any $\alpha > 0$. Note that the t is neither a SPAM parameter nor explicitly involved in the actual transformation but is nonetheless affected by the transformation, so we include it anyway. To show that this transformation is in fact a symmetry, we will apply Newton's Second Law to the center of mass of the secondary galaxy (we will use primed variables to represent transformed quantities):

$$\vec{F}'_s = \frac{d\vec{p}'_s}{dt'} \quad (17)$$

$$\frac{Gm'_p m'_s}{r_{sp}^2} \hat{r}_{sp} = m'_s \frac{d\vec{v}'_s}{dt'} \quad (18)$$

$$\frac{G(\alpha m_p m_s)}{r_{sp}^2} \hat{r}_{sp} = m_s \frac{d(\sqrt{\alpha}\vec{v}_s)}{d(t/\sqrt{\alpha})} \quad (19)$$

Note that in the final equation, all α 's cancel and we are left with the first equation with the primes removed, thus showing that trajectories are preserved up to the time-scale.

The effect of this symmetry on optimization is that when there is no v_z data, the total mass of the system may be held fixed, thus removing it from the list of SPAM parameters which need to be fit.

5.4 Optimization of Fitness Functions

We spent a great deal of time determining which optimization strategy would be the best for our project. Prior to our transition to genetic algorithms (GAs), we used adaptive Markov chain Monte Carlo (MCMC) methods. While these performed

moderately well, they did not meet our desired standards. For this reason, we switched over to GAs which have to be superior. However, despite moving on from MCMC methods, we decided to carry over some features of adaptive MCMC methods and incorporate them into the new GA context (some of which contributed to an increase performance). In the following section, we will detail each aspects of our algorithm and then provide some experimental results.

5.4.1 Variable Transformations

A realization that we had early on was that many of the default SPAM parameters have poor convergence properties. For this reason, we performed several variable transformations in order to obtain a parameter representation which would be more suited to optimization.

Mass fraction and total mass: The first transformation converts the primary and secondary masses m_p, m_s into the mass fraction and total mass m_f, m_T :

$$m_f = \frac{m_p}{m_p + m_s} \tag{20}$$

$$m_T = m_p + m_s \tag{21}$$

This is advantageous for two reasons. First, it utilizes the dynamic symmetry which relates the total mass and velocity scaling (allowing us to easily hold m_T fixed). Also, when viewing citizen fitness scores vs. parameter value plots, m_f has a much more significant peak than either m_p, m_s . This is likely because it determines the relative degrees of tidal distortion between the two galaxies.

Spherical velocity coordinates: Since both galaxies' orientations are represented in terms of spherical coordinates, we also chose to convert the secondary ve-

locity to spherical coordinates $|v|, \phi_v, \theta_v$:

$$|v| = (v_x^2 + v_y^2 + v_z^2)^{1/2} \quad (22)$$

$$\phi_v = \text{atan2}(v_y, v_x) \quad (23)$$

$$\theta_v = \text{asin}(v_z/(v_x^2 + v_y^2 + v_z^2)^{1/2}) \quad (24)$$

With this transformation, the kinetic energy is dependent on fewer parameters. Also, it allows us to apply the spherical random walk used for the orientation angles to the velocity coordinates as well.

Energy ratio: Since dynamic symmetries preserve the morphology by preserving the orbit, we wanted a transformation which took advantage of this. Two consequences of preserving the orbit is that 1) the ratio of kinetic and potential energy (K/U) is preserved and 2) the Lagrangian $L = K - U$ is preserved up to a scale factor. For this reason, we also replace the velocity magnitude $|v|$ with

$$R = \frac{K + U}{K - U} \quad (25)$$

where

$$K = \frac{1}{2}m_s|v|^2 \quad (26)$$

$$U = -G \frac{m_p m_s}{|r|} \quad (27)$$

This transformation also interacts nicely with the dynamic symmetries by directly changing the energy ratios, which are preserved by the transformation. One issue to consider with this symmetry is that when trying to convert back into the SPAM parameters, solving for z requires evaluating a square root, leaving it unclear whether to use the positive or negative solution. We navigate around this issue by representing

all of our SPAM solution in the form of a symmetry which always has a non-zero z value, thus allowing us to always take the positive solution.

Projected area: The final transformations replace the galactic radii with its projected area:

$$A_p = r_p^2 |\cos(\theta_p)| \quad (28)$$

$$A_s = r_s^2 |\cos(\theta_s)| \quad (29)$$

This transformation was chosen primarily because of a frequent convergence problem encountered during early testing. Often, the best-fit solution's r and θ values were incorrect by a small but significant amount. These errors were such that the galactic disks and tidal tails would slightly misalign but in such a way as to roughly preserve the projected area.

Combining all of these transformations, we get the following transformation from the default SPAM parameter set to the new set:

$$\begin{aligned} & [x, y, z, v_x, v_y, v_z, m_p, m_s, r_p, r_s, \phi_p, \phi_s, \theta_p, \theta_s] \rightarrow \\ & [x, y, z, R, \phi_v, \theta_v, m_f, m_\Gamma, A_p, A_s, \phi_p, \phi_s, \theta_p, \theta_s] \end{aligned}$$

5.4.2 Genetic algorithm for optimizing image similarity

We present our GA in a general form in Algorithm 7 and then discuss each step in more detail afterward.

There are numerous parameters which must be set at the beginning of a GA run. These include the population size N_{pop} , the number of generations N_{gen} , and the number of GA phases N_{ph} (explained below). Also, it is important to note that we do not fit the rotation direction during the GA, but instead leave it fixed to either

Algorithm 1: Real-coded genetic algorithm for image similarity optimization

```

1: Initialize:  $N_{pop}, N_{gen}, N_{ph}, \psi_p, \psi_s$ 
2: Initialize population:  $\bar{P}^{j,0}, j = 1, \dots, N_{pop}$ 
3: Evaluate population to obtain fitness scores:  $F(\bar{P}^{j,0}), j = 1, \dots, N_{pop}$ 
4: for  $n = 1$  to  $N_{gen}$  do
5:   if  $n \neq 1$  then
6:     Reseed bottom fraction of population
7:   end if
8:   Perform selection
9:   Perform crossover
10:  Perform mutation
11:  Evaluate population to obtain fitness scores:  $F(\bar{P}^{j,n}), j = 1, \dots, N_{pop}$ 
12: end for
13: for  $m = 1$  to  $N_{ph}$  do
14:  Alternate b/t fixing parameters  $(\phi_v, \phi_p, \phi_s, \theta_v, \theta_p, \theta_s)$  and  $(z, R, m_f, A_s, A_p)$ 
15:  Initialize population:  $\bar{P}^{j,0}, j = 1, \dots, N_{pop}$ 
16:  Evaluate population to obtain fitness scores:  $F(\bar{P}^{j,0}), j = 1, \dots, N_{pop}$ 
17:  for  $n = mN_{gen} + 1$  to  $(m + 1)N_{gen}$  do
18:    if  $n \neq mN_{gen} + 1$  then
19:      Reseed bottom fraction of population
20:    end if
21:    Perform selection
22:    Perform crossover
23:    Perform mutation
24:    Evaluate population to obtain fitness scores:  $F(\bar{P}^{j,n}), j = 1, \dots, N_{pop}$ 
25:  end for
26: end for

```

counterclockwise or clockwise. Since there are only four combinations of values (the two galaxies each have two possible rotation directions), one can simply run a separate GA for each case and determine which performed best. Many additional parameters which allow for fine control over functionality will either be discussed below in the following subsections detailing each aspect of the algorithm.

We use the notation $\bar{P}^{j,n}$ to refer to the SPAM parameter vector of the j -th individual in the population at the n -th generation (specific parameter values can be indexed with a subscript i : $P_i^{j,n}$). We define $F(\bar{P})$ as the fitness score of the model

produced from a given set of SPAM parameters \bar{P} (where F is the complete fitness function discussed earlier).

GA phases: Before describing the steps in the GA, let us discuss the idea of the GA phases. In the initial phase (the first loop in Algorithm 7), we allow all parameters to be varied in order to find the general region of the optimal solution. We then proceed in an alternating sequence of phases in which some parameters are fixed and others are varied. In odd phases, we fix all parameters to their current best-fit values except $\phi_v, \phi_p, \phi_s, \theta_v, \theta_p, \theta_s$. These parameters are highly susceptible to geometric symmetries, so only varying them allows for a more fine-grained search. In even phases, we exchange all of the fixed and varying parameters, so that we now are varying z, R, m_f, A_s, A_p while the angles are fixed at their current best-fit values. This alternation of phases can be repeated as many times as desired. The purpose of this scheme is to take a half symmetry which might be found in the initial phase and (via the alternation) find the true values in a later phase.

Initialize/reseed population: Being robust optimization methods, GAs are intended to be minimally dependent on starting conditions. However, when working with 11 parameters which are all interrelated via complex non-linear dynamics, even GAs can struggle when given a poor initialization. To give our GA the best possible starting conditions, we generate an initial population of linearly-spaced sequence of values for each parameter from its minimum to its maximum. A 2-dimensional example of this where both parameters have a range of $[0,1]$ and the population is size 5 would be given by $\{(0, 0), (0.25, 0.25), (0.5, 0.5), (0.75, 0.75), (1.0, 1.0)\}$. Note that this population is essentially a perfectly correlated diagonal line through the center of the space. Clearly, this would be a terrible initial population if no further steps were taken. However, when a sufficiently large population is used, it has one useful property in that the entire range of values for all parameters is covered. To make use

of this property, we repeatedly apply a random shuffle to the parameters. For the resulting initial population, we use the shuffling with the highest determinant of its correlation matrix of all the parameters (since this would imply that it is the shuffling which is the most “spread out” or, more precisely, the closest approximation to a true uniform distribution).

In addition to initializing the population in the beginning of a GA run, we also use the above method to reseed a portion of the population at each step in order to maintain a high degree of genetic diversity. This is done by replacing a preset fraction of lowest fit solutions with newly generated solutions from the initialization process. We typically use a fraction of 1/8.

Evaluate population: The population fitness is evaluated by running a SPAM simulation for each parameter vector, creating model and unperturbed images, and calculating the fitness score via the complete fitness function discussed above. This is done in parallel via Python’s `Threads` package, allowing for a tremendous speedup since population evaluation is trivially parallel.

Selection: For selection, we implemented fitness-proportionate selection, or roulette wheel selection, where the probability of selection is proportional to the fitness score. The probabilities are obtained by normalizing the fitnesses across the current population. We also implemented elitism, where N_{keep} individuals are kept in the population until the next generation (these solutions are not mutated). We select N_{pop} pairs of parents and let each pair produce one child. Our selection process is laid out in Algorithm 2. Here, we define p^j as the selection probability distribution derived from normalized the fitnesses of the j -th generation. From this distribution, $N_{pop} - N_{keep}$ pairs of parents (labelled \bar{A}^j, \bar{B}^j) are selected. After the random parent selection is done, we perform the elitism step, where the N_{keep} best individuals in the population are saved as duplicate parent pairs (they are both parents \bar{A}^j and \bar{B}^j so that crossover

Algorithm 2: Fitness-proportionate selection

- 1: Let \bar{A}^j, \bar{B}^j where $j = 1, \dots, N_{pop}$ be the parent solution vectors of the next generation
 - 2: Get selection probabilities: $p^j \propto F(\bar{P}^{j,n})^2$
 - 3: **for** $j = 1$ **to** $N_{pop} - N_{keep}$ **do**
 - 4: Select parents \bar{A}^j, \bar{B}^j from the distribution p^j
 - 5: **end for**
 - 6: Keep N_{keep} best solutions as remaining parents
-

between them results in themselves).

Crossover: Our crossover method is essentially the real-coded equivalent of a weighted randomized binary gene exchange crossover. In this scheme, child solutions are given the parameter values of their parents with a probability based on the parents' fitness (higher fit parents have a higher probability of given the child their genes). Algorithm 3 describes our scheme.

Algorithm 3: Weighted randomized gene exchange crossover

- 1: Let \bar{C}^j where $j = 1, \dots, N_{pop}$ be the child solution vectors
 - 2: **for** $j = 1$ **to** N_{pop} **do**
 - 3: **for** $i = 1$ **to** N_{param} **do**
 - 4: Randomly select a child's parameter value C_i^j from (A_i^j, B_i^j) proportional to their fitnesses $(F(\bar{A}^j), F(\bar{B}^j))$
 - 5: **end for**
 - 6: **end for**
-

We tested several other methods of crossover, which did not make their way into the revision of the GA presented in this paper. Weighted mean crossover converged fairly quickly for parameters which were less susceptible to false optima due to symmetries. However, parameters with significant symmetries (θ_p, θ_s , for example) were difficult to converge on due to the fact that mean crossover would attempt to converge in between the two symmetries. Another crossover technique which was quite promising, but proved difficult to incorporate into our final scheme is a variation on weighted randomized crossover. The variation is one of the features which was

brought over from our previous work in adaptive MCMC methods. Using Haario’s covariance matrix from the adaptive Metropolis method (AM) [11], after N_{burn} burn-in generations, we transform the entire population to the principal component basis before and after performing the weighted randomized gene exchange crossover (so that crossover is done in the principal component basis). This method is advantageous because it allows crossover to be performed along (hopefully) more relevant parameter axes. Though this method performed well, it requires a burn-in time that is a sizable percentage of the full runtime. This conflicts with our scheme of several GA short phases where some parameters are held fixed.

Mutation: Our mutation step did retain an idea from our MCMC research (that of kernel mixing). For most parameters, mutation is done by applying a random Gaussian perturbation to the child solutions (see the following paragraph for exceptions). This perturbation is described by a diagonal covariance matrix whose values are based on the parameter range and a scaling factor. Since the optimal scaling factor is not known a priori, we implement a kernel mixing strategy in which each diagonal element of the matrix may be scaled down (thinning the Gaussian in that direction), held fixed, or scaled up (widening the Gaussian in that direction). This scaling is applied randomly based on a presupplied set of mixing probabilities and amplitudes. It is also applied separately for each parameter so that in a single step, one parameter may be scaled up while another may be scaled down.

For the orientation and velocity angles, mutation is done by performing a random walk over the unit sphere. In this case, each step in the walk is done along a particular great circle. The direction is chosen from a uniform distribution while the distance is chosen from the positive side of a Gaussian distribution (which is mixed in the same fashion as the other parameters). The pseudocode for our mutation step is given in Algorithm 4. In each step, we generate a new candidate individual \bar{Q}^j from each

Algorithm 4: Mutation with parameter bounds

```
1: Initialize:  $P_i^{min}, P_i^{max}$  for  $i = 1, \dots, N_{param}$ 
2: for  $j = 1$  to  $N_{pop}$  do
3:   if perturbing  $\phi$ 's or  $\theta$ 's then
4:     Perform great circle transport for orientation and velocity angles
5:   else
6:     Perform kernel mixing on  $\Sigma$  to obtain  $\hat{\Sigma}$ 
7:     Set:  $\bar{Q}^j \sim N(\bar{C}^j, \hat{\Sigma})$ 
8:   end if
9:   for  $i = 1$  to  $N_{param}$  do
10:    if  $Q_i^j < P_i^{min}$  then
11:      Set:  $P_i^{j,n+1} = P_i^{min} + \epsilon$ 
12:    else if  $P_i^{max} < Q_i^j$  then
13:      Set:  $P_i^{j,n+1} = P_i^{max} - \epsilon$ 
14:    else
15:      Set:  $P_i^{j,n+1} = Q_i^j$ 
16:    end if
17:  end for
18: end for
```

child \bar{C}^j . We then check each of its components to ensure that it is within the allowed search space $[P_i^{min}, P_i^{max}]$. If any parameter lies beyond its limits, it is corrected via truncation by replacing its value with one which is slightly within the search space (a distance ϵ away from the boundary).

There are several additional mutation strategies which did not make the final revision of the algorithm. Like with crossover, we experimented with using Haarios's AM covariance matrix. However, it was rejected this for the same reason we rejected its use in crossover. We also, experimented with adapting parameter ranges which were intended to narrow the search onto the optimal region. This strategy was rejected because there was no way to guarantee that the optimal region was not excluded in the narrowing. In an attempt to combat the symmetries, we tested another technique where we periodically apply random flips to the parameters susceptible to symmetries. This technique was rejected because it did not produce results of any better quality. We also tested a kernel scaling idea from simulated annealing in which (due

to cooling), the Gaussian perturbations get asymptotically smaller over runtime. Like the adaptive ranges, this was incorporated in an attempt to narrow in on the optimal region. However, this significantly decreases genetic diversity in the later generations and was eliminated, as well.

5.4.3 Experimental testing of the GA

We performed extensive testing on our fitness function and GA. In this section, we will give an overview of our testing methodology and the results of several example tests to show the effectiveness of our methods.

Experimental methodology and GA hyper-parameter values Our primary method of testing was to use a SPAM model with known parameters as a synthetic target. Doing this allowed for the ability to quantify the accuracy of our fit by 1) visual comparison of the best-fit solution and target images, 2) convergence and clustering of the various SPAM and additional dynamical parameters, and 3) direct comparison of the best-fit and target parameter values.

In the testing of our GA, we used several different synthetic targets systems, but for the sake of conciseness, we will present results from only three different systems. The first system displays tidal features in both galaxies, the second displays no tidal features in either galaxy, and the third displays tidal features in only one galaxy.

For all three target systems, we used a resolution of 4,000 particles per galaxy (ppg) and 35×35 pixel images. For the third target, we performed an additional 10,000 ppg test (see below). The range of search space for each parameter was either obtained via statistics from the Galaxy Zoo: Mergers data files or was manually set. The ranges can be seen on the GA convergences plots below. We limited the search space to only include one of the eight symmetries ($z \geq 0$, $0 \leq \phi_p \leq 180$, and $0 \leq \phi_s \leq 180$). Also, we fixed the rotation direction ψ of the galaxies in the models to

be the same as that of the target since we preferred to keep our GA real-coded. In a case where one does not know the direction, one would simply need to run 4 instances of the GA (one for each combination of directions). Concerning the GA’s parameters, we use a population size of 2^8 (via elitism, we keep the top 4 best-fit solutions). For the initial phase of the GA, we use 2^7 generations (plus the initial step) followed by 2^6 generations (plus the initial step) for each of the subsequent phases of parameter alternation (for a total of 259 generations). We reseed the bottom 1/8 (lowest fitness) of the population at each step using the method discussed earlier. Also, concerning the mixing probabilities and amplitudes for the Gaussian perturbation in the mutation step, we use a uniform 1/3 probability for either thinning, fixing, or widening the Gaussian for each parameter. We also set the thinning and widening amplitudes to be 1/3 and 3, respectively.

We did the majority of our work on a Linux machine equipped with two Intel® Xeon® Gold 6254 CPU @ 3.10 GHz (72 threads total) and 768 GB of RAM. With the parameters described above (4,000 ppg, 256 population size, 259 generations), this system took roughly 12 hours to complete an entire GA run. Since SPAM is linear in time, the 10,000 ppg run took a proportionately longer amount of time. Storage requirements were minimal since SPAM particle files were processed and deleted over the course of the GA.

Synthetic target 1 For our first test, we wanted a target which displayed significant tidal features from both galaxies (tails and a connecting bridge). Such a model is a best-case scenario for the GA to converge since there is a substantial amount of morphological information present in the system (which serves to decrease the level of degeneracy in the parameters).

Figure 12 displays our target image (which is Galaxy Zoo: Merger’s best-fit model of SSIDS ID 587722984435351614) and the GA’s best-fit model. We display these in

two separate formats: a density plot (w/ log scaling) and scatter plot. We show the density plot since it more closely models an observational image. We show the scatter plot to more clearly visualize regions of low density and distinguish between the primary (blue) and secondary (red) galaxies.

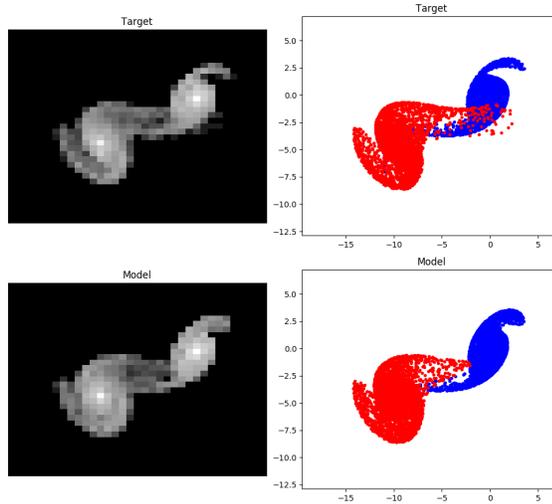


Figure 12: Synthetic target 1. Target images are on the top row and the GA’s best-fit model images are on the bottom row. The density plots on the left are plotted with a log scale. In the scatter plots, the primary galaxy is blue and the secondary is red. The fitness of the best model image was 0.937.

As can be seen from the figure, the GA achieved a high fitness (0.937). There are three notable differences in morphology between the target and model images. First, the primary galaxy’s top tidal tail is slightly thinner in the target than the model. Second, the secondary’s rightward-pointing tail is slightly longer in the target than the model. Third, there is a slightly larger gap between secondary’s leftward-pointing tail and the galactic bulge in the target than the corresponding region of the model.

Moving on to Figure 13 and 5, we see the GA’s parameter convergence history and best-fit values. From the plots and table, we see that most of the SPAM parameters converged well (recall that x, y, m_T are held fixed at the correct value). The z value may appear inaccurate; however, the search range spanned $[0, \sim 25]$, so we do not find

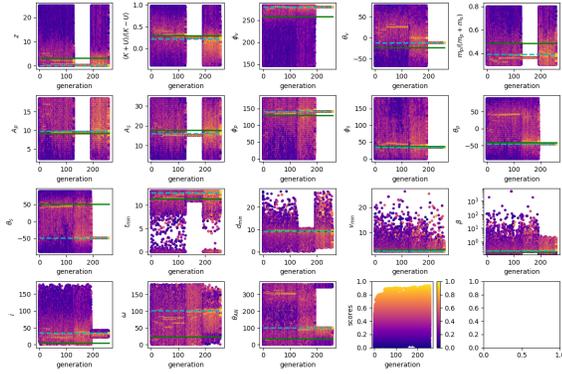


Figure 13: Parameter convergence plots for synthetic target 1 using the GA. The solid lines are the true parameter values while the dotted lines are the best-fit parameter values. The first 11 plots display the transformed SPAM parameters, the next 7 plots display the orbital elements, and the final plot displays the fitness values. Note that t_{min} is in the convention that a value of 10 is interpreted as 10 time units *prior* to the final state shown in the plots. Also, $\beta = M_{tot}/(v_{min}r_{min}^2)$ is a measure of the impulse applied to the primary by the secondary.

Table 5: Table of parameter values for synthetic target 1 and its best-fit GA model.

	x	y	z	R	ϕ_v	θ_v	m_f
target	-9.939	-4.581	3.274	0.291	258.489	-23.275	0.486
model	-9.939	-4.581	0.017	0.212	280.307	-12.235	0.389
	m_T	A_p	A_s	ϕ_p	ϕ_s	θ_p	θ_s
target	48.024	9.125	17.65	129.692	36.812	-41.785	51.429
model	48.024	9.742	16.022	140.538	34.092	-45.42	-48.451
	t_{min}	d_{min}	v_{min}	β	i	ω	Ω
target	11.356	9.134	3.116	0.187	5.163	22.975	35.014
model	12.712	9.215	2.64	0.214	34.34	101.531	99.582

a difference of 3 to be significantly in error. Also, the model’s mass fraction m_f is 20% lower than the target’s, but this parameter doesn’t have a very large impact on the morphology. At first glance, the model’s secondary altitude θ_s appears significantly in error; however, it is merely the (approximate) negative of the target value. This is due to one of the half symmetries discussed earlier. Flipping the altitude of the

secondary does not affect the primary at all (given the restricted 3-body formalism) and the effect on the secondary itself is minor. Concerning the orbital parameters, the first four (none of which are angles) all fit quite well. However, the argument of periapsis ω is off by 80° . This is unrelated to the problem of θ_s since ω is defined by the primary’s plane of rotation. Also, this error is unlikely to significantly affect the morphology since a difference in ω merely amounts to rotating the point of closest approach to a different angular location, leaving $t_{min}, d_{min}, v_{min}$ unchanged. A similar argument applies to Ω .

Synthetic target 2 For our second target, we wanted essentially the opposite of the first: a target with minimal tidal features, making prone to high degrees of parameter degeneracy. We were interested in how the GA would perform under such poor conditions. Figure 14 displays the target (Galaxy Zoo: Merger’s best-fit model of SDSS ID 587736941981466667) and the GA’s best-fit model.

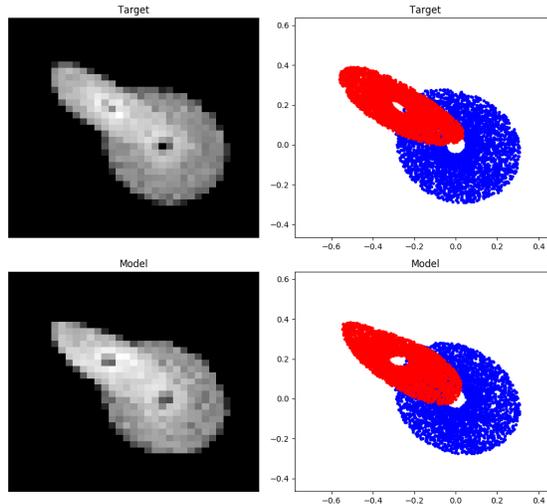


Figure 14: Synthetic target 2. Target images are on the top row and the GA’s best-fit model images are on the bottom row. The density plots on the left are plotted with a log scale. In the scatter plots, the primary galaxy is blue and the secondary is red. The fitness of the best model was 0.946.

Looking at the density plots, it is difficult to discern any morphological difference between the target and model. However, looking at the scatter plot, we see that the right half of the model’s secondary galaxy is significantly warped. This warping is obscured in the density plot and so the model achieved a high fitness of 0.946.

Looking at the GA convergence plots in Figure 15, we see several interesting features. As expected, the parameters responsible for the system’s geometry were fit quite well while the parameters which govern the system’s dynamics display a large amount of error. The reason for this is that the geometry of the system is the only significant constraint (since there are no tidal features). The geometric parameters (the four orientation angles and the areas) were all fit well. Moving to the dynamical parameters, we see a very significant error in the energy ratio R and the velocity angles ϕ_v, θ_v . Surprisingly, the z parameter converged fairly well with visible clustering in the GA run near the target value. This is unexpected since (given the low distortion) large values are consistent with high fitness, and there shouldn’t be any reason to favor lower values which might promote an increase in tidal distortion.

Similarly, the mass fraction m_f fit better than expected since (as stated above) it

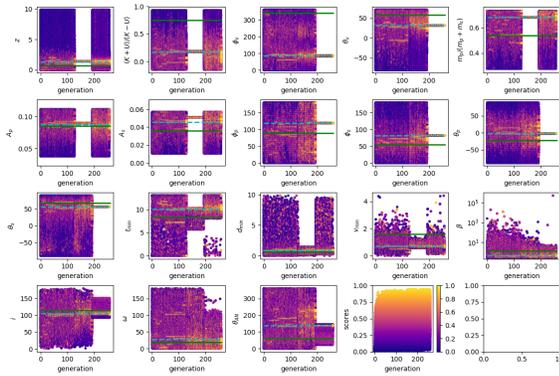


Figure 15: Parameter convergence plots for synthetic target 2 using the GA.

doesn’t have a large impact on fitness.

Concerning the orbital parameters, i, ω are fit well but Ω is not (like due to

Table 6: Table of parameter values for synthetic target 2 and its best-fit GA model.

	x	y	z	R	ϕ_v	θ_v	m_f
target	-0.273	0.19	0.669	0.755	340.999	57.573	0.539
model	-0.273	0.19	1.362	0.176	85.028	31.849	0.684
	m_T	A_p	A_s	ϕ_p	ϕ_s	θ_p	θ_s
target	0.239	0.086	0.036	89.048	54.761	-21.867	66.515
model	0.239	0.087	0.046	118.982	82.289	-1.962	55.589
	t_{min}	d_{min}	v_{min}	β	i	ω	Ω
target	8.27	0.625	1.6	1.245	114.051	18.899	62.544
model	10.102	0.908	0.709	0.409	104.1	28.248	138.288

the velocity angles being incorrect). Concerning the point of closest approach, the model’s t_{min} was slightly larger (which might explain the additional tidal warping in the secondary galaxy). However, the model’s d_{min} was approximately 50% larger than the target while its v_{min} was less than 50% of that of the target. This led to the model having a β value of approximately 1/3 of the target’s value.

Synthetic target 3 For the third target, we chose a system which had tidal features present in only one galaxy. In our initial runs with 4,000 particles per galaxy, we noticed that its high degree of tidal distortion caused the primary galaxy to become quite diffuse. This made achieving a high fitness difficult due to a deceptively low overlap between model and target pixels. For this reason, we increased the resolution to 10,000 particles per galaxy.

Figure 16 displays the target (Galaxy Zoo: Merger’s best-fit model of SSDS ID 587726033843585146) and the GA’s best-fit models for both the 4,000 ppg (model 1) and 10,000 ppg (model 2) runs.

As expected, the fitness for the 10,000 ppg run is higher (0.88) than the 4,000 ppg run (0.74). This is due to both an improved match in morphology and a higher pixel overlap due to the increased density. Looking at the images, we see that since the

Table 7: Table of parameter values for synthetic target 3 and its best-fit GA models.

	x	y	z	R	ϕ_v	θ_v	m_f
target	-0.493	-1.438	1.552	0.074	261.172	-8.735	0.467
model 1	-0.493	-1.438	0.539	-0.129	289.538	-51.039	0.4
model 2	-0.493	-1.438	0.822	-0.158	277.892	-34.453	0.369
	m_T	A_p	A_s	ϕ_p	ϕ_s	θ_p	θ_s
target	2.463	0.201	0.13	56.943	56.708	-83.172	0.0
model 1	2.463	0.721	0.157	72.268	64.868	-69.611	-15.389
model 2	2.463	0.273	0.124	76.612	95.072	-74.684	17.314
	t_{min}	d_{min}	v_{min}	β	i	ω	Ω
target	9.696	1.18	1.921	1.096	111.386	37.074	11.951
model 1	2.356	0.275	2.83	11.537	88.61	154.285	23.141
model 2	2.463	0.032	4.212	561.548	90.27	63.959	21.592

were all nearer to the target value while model 1’s values of $R, m_f, \phi_p, \phi_s, \theta_s$ were nearer. Notice that the two models’ values of θ_s are separated by an approximate half symmetry (they are also both very small, so the symmetry might not be a relevant factor). Comparing the disk radii and distance of closest approach, we find something interesting. Converting units back into SPAM units, we find that the primary radius r_p of the target, model 1, and model 2 are 1.28, 1.45, and 1.02, respectively. Also, the d_{min} ’s of these systems are 1.18, 0.275, and 0.032, respectively. So, for the target, d_{min} is slightly smaller than r_p (i.e., the collision was on the edge), while for the models, it is significantly smaller (i.e., the collision was near the center).

In addition, both models also display bound orbits (since $R < 0$) while the target is very slightly parabolic ($R > 0$). For these reasons, the first four orbital parameters show significant error (especially t_{min} and β). Both models’ t_{min} shows a much more recent collision than is the case for the target but with a significantly larger β . Also, looking at d_{min}, v_{min} , both models had a much faster and close-range flyby than the target. These factors combine to produce highly erroneous β values, particularly for model 2.

5.4.4 Discussion of GA Results

After analyzing the results from the above three tests, there are two principal insights we have drawn: *1) there is an optimal range of degree of tidal distortion within which fitting is easier and beyond which fitting is more difficult and 2) fitting is easier when both galaxies are tidally distorted.*

The first of these insights is demonstrated by the difference in success of fitting targets 1 and 2. Target 1 had a substantial degree of tidal distortion, providing plenty of morphological features. In this case, the parameters fit quite well. In target 2, it was easy to find many models which matched the morphology and geometry; however, we did not find the true values for many of the parameters. This is due to the lack of discernible features in the system. We did not test a target with an extreme degree of tidal distortion since such targets were not included in the Galaxy Zoo: Mergers dataset. These targets are challenging to fit since they are often bound orbits which have undergone multiple approaches. Such systems are highly sensitive to changes in dynamical parameters and create chaotic morphologies.

The second of these insights is demonstrated by the difference in success of fitting targets 1 and 3. In the case of target 1, both galaxies had roughly equal degrees of tidal distortion. Conversely, all of the tidal distortion in target 3 was concentrated in the primary galaxy. After adjusting the resolution, we were able to fit some parameters well and recover the morphology to an extent, but our best model's dynamics were erroneous (see the orbital parameters). We suspect that this is primarily due to a lack of tidal features in the secondary galaxy. Tidal features are created by translating the initially circular orbits of particles (stars and gas) into elliptical orbits, making them highly dependent on the dynamical time since the interaction occurred. When both galaxies have tidal distortions, we have two independent measurements of dynamical time, placing much stronger constraints on the parameters.

5.5 Conclusions

In this paper, we have developed and tested a suite of methods for fitting models of interacting galaxies to target systems. Our fitness function is able to reproduce the citizen scientists' rankings of the Galaxy Zoo: Mergers models to a significant degree of accuracy. It is also able to identify low-distortion models and penalize their score. Our analysis of the various types of symmetries inherent in the simulation code's representation of the galaxies also allows for 1) more efficient exploration of parameter space by elimination of degenerate regions and 2) helpful analysis of GA results to determine if a given parameter is truly inaccurate or merely a transformation away from the true value.

Our experiments suggest that our real-coded GA achieves accurate results on systems where both galaxies are tidally distorted. Given the true value of the total mass, the best-fit model of target 1 gives a close approximation of the absolute time, mass ratios, and orbital parameters, with one degeneracy present in θ_s . This and other degeneracies can be eliminated with the inclusion of spectroscopic information in the fitness function. This would also allow for determination of the absolute time and total mass of the system.

Targets 2 and 3 show the limitations of using morphology alone to fit a system's dynamical parameters (given that they display less significant tidal features than target 1) when spectroscopic data is unavailable. Though our best-fit models reproduce the morphology, they cannot reliably reproduce the dynamics of the system. For systems without significant tidal distortion in both galaxies, the degeneracies present in this dynamical system make it impossible to disambiguate the underlying orbital dynamics.

In future work, we plan to 1) continue iterating on the fitness function through the inclusion of various image transformations and machine learning techniques, 2)

use our GA to fit a model to a true observational image of an interacting system, and
3) expand the number of test systems to further explore how well these systems can be constrained.

6 A kernel mixing strategy for use in stochastic optimization and adaptive Markov chain Monte Carlo contexts

Like before, the material from the following section was taken directly from its associated paper. However, we leave the introductory section unaltered since the material contained within was not discussed in the dissertation’s introduction.

It is also worth noting that the origin of the method discussed in this section is an old attempt to solve the galaxy fitting problem using MCMC methods. At that point, the method was quite different. Although we eventually migrated to genetic algorithms for the galaxy fitting problem, we realized that the mixing method still had potential, so we decided to write this paper.

6.1 Introduction

Mathematical models can be used to reproduce and study the behavior of many physical systems in the world. In most cases, however, these models contain many parameters which must be tuned in order for the model to accurately reproduce the behavior of the system in a specific instance. This problem of fitting the parameters of a mathematical model to data—as well as the related problem of uncertainty quantification of said model parameters—is a ubiquitous problem in applied mathematics. We can represent it as an optimization problem. Suppose we are given model $\bar{f}(\bar{\theta})$ (with parameters $\bar{\theta}$) and data y . We must solve the problem

$$\min_{\bar{\theta}} \|\bar{f}(\bar{\theta}) - \bar{y}\| \tag{30}$$

The value of $\bar{\theta}$ which minimizes this error can be found via various global optimization methods and with subsequent analysis of uncertainty via sampling methods. However, the efficiency of these methods depends largely on various parameters which must be set. For this reason, methods with robust adaptive features which are able to tune themselves depending on the problem are preferred.

In this paper, we will present a new kernel mixing method which can be implemented in various global optimization methods and sampling methods. This method mixes Gaussian distributions with varying covariance matrices so that an overall underestimation or overestimation of the covariance matrix for the particular applied problem does not have as significant of an impact on performance. We specifically test this mixing method’s performance in a simulated annealing (SA), real-coded genetic algorithm (GA), and Markov chain Monte Carlo context. All three of these methods have a similar underlying scheme for generating new solutions from Gaussian perturbations and are thus prime candidates for testing our method.

We will begin with a brief overview of the MCMC, SA, and GA algorithms. We will then discuss the kernel mixing method, its properties, and how it can be implemented. We then test the three algorithms performance with and without our kernel mixing scheme on two popular benchmark problems: the Ackley function and the thermal isomerization of α -pinene.

6.1.1 The Metropolis method

First developed by Metropolis et al. [24] and later generalized by Hastings [12], is the original MCMC method and is the foundation for all subsequent MCMC methods. Given data y , Metropolis samples model parameters’ *posterior* distribution $\pi(\theta|y)$. This distribution describes the probability of the model parameters given the data. This distribution will have a peak at the values of the model parameters which best

approximate the data.

However, as its name would imply, we do not have a priori knowledge of the posterior. Therefore, we must express it in terms of distributions which we do know. From Bayes's Theorem, we obtain

$$\pi(\theta|y) \propto \ell(y|\theta)p(\theta)$$

where ℓ is the *likelihood* distribution and p is the *prior* distribution. The likelihood $\ell(y|\theta)$ tells us the probability of the data given the parameters and is usually calculated from the model-data error

$$\ell(y|\theta) \propto \exp\left(-\|\bar{f}(\bar{\theta}) - \bar{y}\|^2/(2\sigma^2)\right) \quad (31)$$

where σ is the measurement error. This can be easily calculated via a single model evaluation with the given parameters. We may also have additional information about what parameter values are most likely to be realized in the actual system. For example, perhaps the parameters are uniformly distributed over a bounded space. This information is incorporated via the prior distribution p . Together, the likelihood and prior help us to reconstruct the posterior distribution up to a scale factor.

Let us look at how Metropolis performs its sampling. Given an initial state θ^0 (superscript indices will refer to the time step), Metropolis generates new candidate states and probabilistically accepts or rejects them based on relative posterior gains or losses with respect to that of current state. Candidate state generation is performed via sampling of the *proposal* distribution $q(\theta'|\theta^n)$, defined as the probability that a new state θ' will be selected as a candidate for acceptance given the current state θ^n . The most common choice of proposal distribution is the standard multivariate Gaussian distribution. Whether acceptance or rejection is performed is determined by

the *acceptance probability* $\alpha(\theta'|\theta^n)$, defined as the probability that the candidate state θ' will be accepted given the current step θ^n (see Algorithm ?? for the calculation of the acceptance probability). The unification of these candidate generation and acceptance steps is the *kernel* of the method, which defines the transition probabilities between states. Over time, the distribution created from the samples will conform to the posterior, at which point the method is said to have *converged*. Algorithm 5 shows a pseudocode implementation of Metropolis.

Algorithm 5: The Metropolis method

- 1: Initialize: θ^0, N
 - 2: **for** $n = 1$ **to** N **do**
 - 3: Generate $\theta' \sim q(\theta'|\theta^{n-1})$
 - 4: Compute $\alpha(\theta'|\theta^{n-1}) = \min\left(1, \frac{\pi(\theta'|y)}{\pi(\theta^{n-1}|y)}\right)$
 $\qquad\qquad\qquad = \min\left(1, \frac{\ell(y|\theta')p(\theta')}{\ell(y|\theta^{n-1})p(\theta^{n-1})}\right)$
 - 5: Set $\theta^n = \theta'$ with probability α , else $\theta^n = \theta^{n-1}$
 - 6: **end for**
-

It is a known fact that Metropolis is ergodic and is therefore guaranteed to converge to the correct stationary distribution if given sufficient time. However, the time to converge depends largely on the proposal distribution used. As stated, the preferred proposal distribution is Gaussian, whose width is defined by a covariance matrix (in fact, the term *proposal width* is often used in its place). If one chooses too thin of a Gaussian, the chain may have difficulty escaping suboptimal modes within an reasonable amount of time. On the other hand, too wide and the chain will have an excess of rejections. Fortunately, this problem of the unknown proposal can be largely alleviated through clever means such as kernel mixing and adaptive proposals.

Kernel mixing (or more precisely proposal mixing, since kernel would refer to the composition of the proposal and acceptance steps) is a technique where the proposal

used at each step in the chain may be chosen stochastically from a set. Consider an example where we have two proposals Q_1, Q_2 and we have a probability 0.5 of choosing either at any given step. Then, the proposal used at each step is simply $0.5Q_1 + 0.5Q_2$ —the linear combination of the kernels and their respective probabilities. This is implemented by choosing either Q_1 or Q_2 at each step.

Another technique for improving the performance of MCMC is to use so-called *adaptive* methods which allow the proposal to adapt to the state space as more and more samples are obtained. Likely the most widely used adaptive MCMC method is Haario’s *Adaptive Metropolis* (AM) [11] method. The core principle of adaptation is the use of a continuously adapting Gaussian distribution as the proposal for the standard Metropolis method. After beginning with an initial phase of non-adaptation called the *burn-in*, the proposal becomes

$$q^n(\theta'|\theta^n) = N\left(\theta^n, \left(\frac{2.38^2}{M}\right)\Sigma^n + \varepsilon I\right) \quad (32)$$

where M is the dimension of the space (i.e., number of parameters), Σ^n is the covariance matrix computed from samples generated over past steps, and ε is a small factor multiplied by an $M \times M$ identity matrix included for regularization to ensure positive-definiteness. This allows the Gaussian to continually re-scale and re-orient itself as it “learns” the state space.

The AM proposal can also be modified via kernel mixing. A good example of this is the version developed by Roberts and Rosenthal [**AM^{RR}**]

$$q^n(\theta'|\theta^n) = \beta N\left(\theta^n, \left(\frac{2.38^2}{M}\right)\Sigma^n\right) + (1 - \beta)N\left(\theta^n, \hat{\Sigma}\right) \quad (33)$$

where $\hat{\Sigma}$ is a fixed covariance matrix and $0 \leq \beta \leq 1$ (note that our convention is different from theirs in that we have swapped the β and $(1 - \beta)$ terms).

6.1.2 Simulated Annealing

Simulated annealing is a global optimization method with many similar properties to the Metropolis method. Both use proposal distributions to generate candidate solutions followed by an acceptance criterion. The main difference is the cooling aspect of SA which causes it to be more restrictive and therefore less likely to accept candidates worse than the current state. In lieu of the likelihood function, SA converts the model-data error into a fitness value which is then maximized (instead of minimizing the error). This is given by

$$F(\bar{\theta}) = \exp(-\|\bar{f}(\bar{\theta}) - \bar{y}\|/T^n) \quad (34)$$

where T^n is the *temperature* which is cooled over time. Various cooling schedules may be used, but for this paper we use an exponential cooling schedule

$$T^n = T^0 \exp(-n/\tau) \quad (35)$$

where τ is the cooling factor. SA can be described by Algorithm 6 Note the directing

Algorithm 6: Simulated Annealing

- 1: Initialize: θ^0, N
 - 2: **for** $n = 1$ **to** N **do**
 - 3: Generate $\theta' \sim q(\theta'|\theta^{n-1})$
 - 4: Compute $\alpha(\theta'|\theta^{n-1}) = \min\left(1, \frac{F(\theta')}{F(\theta^{n-1})}\right)$
 - 5: Set $\theta^n = \theta'$ with probability α , else $\theta^n = \theta^{n-1}$
 - 6: **end for**
-

analogy between this and Algorithm 5.

6.1.3 Real-coded Genetic Algorithms (GA)

For some complex optimization problems, simpler methods such as gradient descent and even simulated annealing are not robust enough to find the global optimum. In these cases, a popular alternative is the genetic algorithm (GA) (see the original paper by Holland [15]). Taking inspiration from biological processes, GAs solve problems by tackling it with an entire population of solutions which evolve and improve over time in a survival-of-the-fittest fashion. Though the original GA proposed by Holland used a binary encoding, many other researchers have used real-coded or real-valued GAs (see [25, 27, 29]). We will be using the real-coded scheme.

Algorithm 7 shows the architecture of a canonical GA, which is very modular by nature. Let us look more closely at each of these steps.

Algorithm 7: Real-coded genetic algorithm for image similarity optimization

- 1: Initialize: $N_{pop}, N_{gen}, N_{ph}, \psi_p, \psi_s$
 - 2: Initialize population: $\bar{P}^{j,0}, \forall j = 1, \dots, N_{pop}$
 - 3: Evaluate population to obtain fitness scores: $F(\bar{P}^{j,0}), \forall j$
 - 4: **for** $n = 1$ **to** N_{gen} **do**
 - 5: Perform selection
 - 6: Perform crossover
 - 7: Perform mutation (obtaining new population $\bar{P}^{j,n}, \forall j$)
 - 8: Evaluate population to obtain fitness scores: $F(\bar{P}^{j,n}), \forall j$
 - 9: **end for**
-

The GA's *representation* is simply how solutions to the problem are encoded in the individuals in the population. In the case of real-coded GAs, the representation is simply a vector of the model parameters.

Selection is the process by which individuals are paired off as parents for mating. The most popular techniques for accomplishing this are roulette selection [15] (where an individual's selection probability is proportional to its fitness) and rank selection [3] (where an individual's selection probability is proportional to its rank). We used

roulette selection in all of our GA tests.

Having selected the parent pairs, breeding of child solutions occurs through the *crossover* operator. This operator creates children by combining the genetic material of the parents. Depending on preference, one can create a single child from each parent pair or two which replace both parents in the population (in second case, half as many parent pairs are selected). For the case of real-coded GAs, all of the binary coded crossover operators are available (such as single-point, multi-point, and uniform crossover) with additional options due to the nature of the encoding. A common one is the *mean crossover*

$$C^1 = \alpha P^1 + (1 - \alpha)P^2 \quad (36)$$

$$C^2 = (1 - \alpha)P^1 + \alpha P^2 \quad (37)$$

where $\alpha \in [0, 1]$ is uniformly distributed. Many different versions of this scheme exist with various types of weighted means and distributions from which α may be drawn. For our purposes, we use a version of mean crossover with the parents weighted by their fitnesses:

$$C_i = \hat{F}(P^1)P^1 + \hat{F}(P^2)P^2 \quad (38)$$

Here, $\hat{F}(P^1) = F(P^1)/(F(P^1) + F(P^2))$ and $\hat{F}(P^2) = F(P^2)/(F(P^1) + F(P^2))$.

After crossover, the *mutation* operator is applied to the resulting child solutions. This is in order to maintain sufficient genetic diversity in the population throughout the evolution of the population. Mutation is done by stochastically perturbing the child solutions via a chosen kernel distribution. Various choices are available for this kernel [19, 21]. We use a multivariate Gaussian distribution.

6.2 Kernel Mixing Method

We now discuss our kernel mixing method. We will begin with a discussion of the motivation and advantages of using this method. We then discuss the mathematics and implementation of the method in case of both diagonal and non-diagonal covariance matrices.

6.2.1 Motivation and advantages

Incorporation of our kernel mixing method into either stochastic optimization or MCMC contexts results in several advantages. First, as we have discussed, it isn't possible in general to know a priori what the optimal proposal width should be for a given problem. This makes adaptive methods attractive. However, even if one were to choose the optimal "global" proposal width, this by no means guarantees that certain regions of parameter space might not have a superior "local" optimal proposal width. By applying our kernel mixing method, the proposal can cover a range of widths for different parameters.

A second advantage is that when used in an MCMC context, no initial greedy search is required to find a good starting point in parameter space. This is because our mixing method provides an increase in performance in optimization contexts. This performance gain is inherited by MCMC implementations of the method.

Additional advantages that our method has over some more complex methods are its trivial implementation and the minimal overhead it requires. The only significant computational overhead encountered is the diagonalization of non-diagonal covariance matrices. This overhead is only relevant when there is a very large number of parameters and the covariance matrices are adaptive, requiring constant diagonalization.

6.2.2 The method

Our mixing strategy allows for the selection of one of three proposals: 1) the original, fixed proposal, 2) a thinned proposal, and 3) a widened proposal. We included both the thinner and wider proposals since overestimation and underestimation the proposal width are both possible. Additionally, the thin proposal allows the chain to squeeze into thinner modes while the wide proposal allows the chain to escape suboptimal modes and traverse parameter space quickly. The resulting proposal for a simple problem with one parameter has the form of the linear combination

$$q(\theta'|\theta^n) = p_t N(\theta^n, (A_t \sigma)^2) + p_f N(\theta^n, (\sigma)^2) + p_w N(\theta, (A_w \sigma)^2) \quad (39)$$

where σ is the fixed proposal width, $0 < A_t < 1 < A_w$ are the mixing amplitudes, and p_t, p_f, p_w are the mixing probabilities. This resulting mixture of Gaussians (each with a different variance) will not necessarily have the same variance as the original fixed variance σ^2 . However, given the thinning and widening amplitudes A_t, A_w , the mixing probabilities can be set so that the mixed proposal will have identical variance as the fixed proposal. This is useful if one desires to have control over the variance of the mixed proposal. The probabilities that accomplish this are given by

$$(p_t, p_f, p_w) = \left(\frac{(A_w^2 - 1)(1 - p_f)}{(A_w^2 - A_t^2)}, p_f, \frac{(1 - A_t^2)(1 - p_f)}{(A_w^2 - A_t^2)} \right) \quad (40)$$

where the fixed probability p_f is a free parameter, i.e., any value $0 \leq p_f \leq 1$ will result in the same fixed variance.

Figure 18 illustrates a standard 2D Gaussian distribution and a mixture of Gaussians according to the method described above. We have mixing probabilities of $(1/3, 1/3, 1/3)$ and mixing amplitudes of $(1/3, 1, 3)$. There are two main features to notice in the mixture plot. First, is the overall cross shape of the distribution, with a

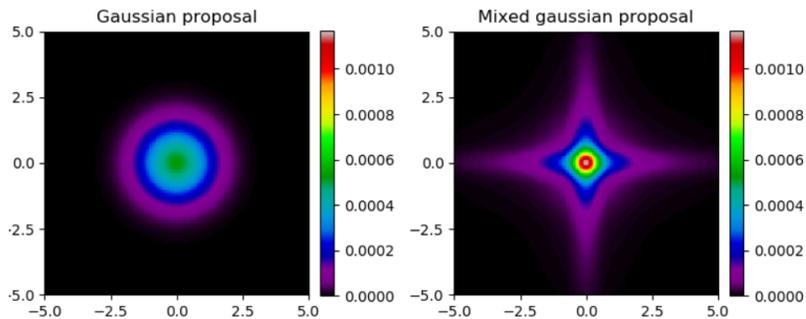


Figure 18: Left: standard Gaussian proposal. Right: mixed proposal.

high probability density along and near the parameter axes. Second is that the peak density in the center is higher in the mixed kernel than the standard Gaussian.

In the case of M parameters, each one is mixed independently (i.e., one parameter could be thinned while another is widened). Therefore, one must account for all combinations of each parameter being mixing in each of the three ways:

$$q(\theta'|\theta^n) = \sum_{i_1 \in \{t,f,w\}} \cdots \sum_{i_M \in \{t,f,w\}} p_{i_1} \cdots p_{i_M} N(\theta^n, (A_{i_1} \sigma_1)^2, \cdots, (A_{i_M} \sigma_M)^2) \quad (41)$$

For convenience in the above, we define $A_f = 1$ and $N(\mu, \sigma_1^2, \cdots, \sigma_M^2)$ as an M -dimensional normal distribution with diagonal covariance matrix given by $(\sigma_1^2, \cdots, \sigma_M^2)$.

6.2.3 Implementation

Regarding implementation, mixing for diagonal covariance matrices is applied according to Algorithm 8. Given a fixed diagonal covariance matrix Σ , a new covariance matrix Σ' (initially the identity matrix I) is constructed by multiplying the diagonal elements of Σ by the square of the chosen mixing amplitude. Note that this is equivalent to multiplying the eigenvalues by the amplitudes.

For general, non-diagonal covariance matrices, Algorithm 9 is used. The significant differences are the diagonalization steps (2, 13) to obtain the eigenvalues. In the

Algorithm 8: Mixing implementation for diagonal covariance matrices

```
1: Initialize:  $p_t, p_f, p_w, A_t, A_w, \Sigma, \Sigma' = I$ 
2: for  $m = 1$  to  $M$  do
3:   Choose (thinning, fixing, widening) from the distribution  $(p_t, p_f, p_w)$ 
4:   if thinning then
5:      $\Sigma'_{m,m} = \Sigma_{m,m} A_t^2$ 
6:   else if fixing then
7:      $\Sigma'_{m,m} = \Sigma_{m,m}$ 
8:   else if widening then
9:      $\Sigma'_{m,m} = \Sigma_{m,m} A_w^2$ 
10:  end if
11: end for
```

Algorithm 9: Mixing implementation for non-diagonal covariance matrices

```
1: Initialize:  $p_t, p_f, p_w, A_t, A_w, \Sigma, D' = I$ 
2: Diagonalize:  $D = V \Sigma V^{-1}$ 
3: for  $m = 1$  to  $M$  do
4:   Choose (thinning, fixing, widening) from the distribution  $(p_t, p_f, p_w)$ 
5:   if thinning then
6:      $D'_{m,m} = D_{m,m} A_t^2$ 
7:   else if fixing then
8:      $D'_{m,m} = D_{m,m}$ 
9:   else if widening then
10:     $D'_{m,m} = D_{m,m} A_w^2$ 
11:  end if
12: end for
13:  $\Sigma' = V^{-1} D' V$ 
```

diagonal case, these were already given, but in the non-diagonal case, they must be calculated. The ability to apply this mixing procedure to non-diagonal matrices implies that it may be used in optimization and MCMC algorithms with adaptive covariance matrices, such as Haario's Adaptive Metropolis algorithm [11]. This merely requires the diagonalization of the covariance matrix at each step in the algorithm.

6.3 Numerical Experiments and Results

We now present tests of our mixing technique in SA, GA, and MCMC contexts. In the SA case, we compare both standard SA and mixed SA with modified versions using an adaptive proposal (following the pattern of AM). Solonen [SA·AM] illustrates the difficulties in weighting the samples when applying the AM technique in the SA context (the changing temperature affects a change in the posterior). They propose multiple possible solutions to this challenge. Since our goal was to propose a mixing method rather than an adaptive method, we take a very simplistic approach to adaptation. Like AM, we accumulate samples over a burn-in period, after which their covariance matrix is calculated. However, we do not continue to adapt the matrix any further. We leave it to others to decide what the proper adaptive method for their application should be. In the GA case, we compare a real-coded GA with a Gaussian mutation operator to one which uses our mixing technique, as well as their adaptive counterparts. Like the SA implementation of proposal adaptation, we accumulate samples (the entire population at each generation) until the burn-in time is reached, at which point the covariance matrix is evaluated and subsequently held fixed. In the MCMC case, we compare Metropolis and Adaptive Metropolis with versions which use our mixing technique. Tests on all three methods are done via two benchmark problems, which will be discussed below.

We perform one test in the SA and GA cases and two in the MCMC case, each of which uses an ensemble of runs over a range of proposal widths. This will allow us to check for both best-case and average performance for each method. The test for the SA and GA cases is a simple averaging of the best fitness at each time step over an ensemble of runs. By “best” fitness, we mean to say we keep track of the highest fitness achieved by the method at each step, rendering the sequence of fitnesses non-decreasing. This allows us to examine the average rate at which the fitness improves

across a broad range of proposal widths. What we will see is that each method has an optimal range of proposal widths at which fitness increases most quickly. Deviations from this optimal region slows performance. The mixing and adaptation techniques implemented serve to extend this range so that performance is less sensitive to one's choice of proposal width.

The first MCMC test calculates a lower bound of the *integrated auto-correlation time* (IAC) for the chain. The IAC is a measure of the length of time (in steps) required for later steps in the chain to become de-correlated with earlier steps. It is given by the formula

$$\tau_{int} = 1 + 2 \sum_{\ell=0}^{N'} r_{\ell} \quad (42)$$

where r_{ℓ} is the autocorrelation of the chain at lag ℓ (an index offset) and N' is some number of steps in the chain (usually the step at which r_{ℓ} reaches zero). This only measures a lower bound since—in many cases— N' is prohibitively large. For this reason, we simply choose the value of the IAC at some preset value of N' (usually $N_{step}/8$). By generating an ensemble of chains, we obtain statistics on this lower bound for the IAC across a range of proposal widths.

The second test uses the *scale reduction factor* (SRF) (cite gelman and rubin). This tests utilizes an ensemble of chains initialized at random locations and compares the values of the between-chain variance B and the within-chain variance W . Given M chains of length N , the SRF is given by $R = \sqrt{\hat{V}/W}$ where

$$B = \frac{N}{M-1} \sum_{m=1}^M (\hat{\theta}_m - \hat{\theta})^2 \quad W = \frac{1}{M} \sum_{m=1}^M \hat{\sigma}_m \quad (43)$$

$$\hat{V} = \frac{N-1}{N} W + \frac{M+1}{MN} B \quad (44)$$

where $\hat{\theta}_m, \hat{\sigma}_m$ are the sample mean and variance of the m -th chain and $\hat{\theta}$ is the overall

sample mean across all m chains. Since the test itself requires an ensemble of chains, we do not obtain statistics for the SRF. However, we do still calculate it over a range of proposal widths.

6.3.1 Benchmark 1: Ackley function

For our first benchmark, we use the Ackley function, given by

$$f(\bar{\theta}) = a \left(1 - \exp \left(-0.2 \sqrt{\sum_{m=1}^M \theta_m^2 / M} \right) \right) + b \left(e - \exp \left(\sum_{m=1}^M \cos(2\pi\theta_m) / M \right) \right) \quad (45)$$

where $a = 20$, $b = 4$, and $M = 5$ is the number of dimensions. This is a common benchmark function in optimization contexts since it has a large number of false minima in which chains may become stuck. For the SA and GA cases, we calculate fitness F via

$$F(\bar{\theta}) = \frac{1}{1 + f(\bar{\theta})^2} \quad (46)$$

For the MCMC case, we used a uniform prior distribution over the entire search space and used the likelihood function

$$\ell(\bar{y}|\bar{\theta}) = \exp \left(-\frac{f(\bar{\theta})^2}{2\sigma^2} \right) \quad (47)$$

where $\sigma = 5$ (this value was chosen since it resulted in an acceptance rate of between 0.25 and 0.5).

Simulated Annealing Now, we present the results of the SA test on the 5D Ackley function (see Figure 19). We have standard SA, mixed SA, adaptive SA, and mixed/adaptive SA. The parameter space is set to $\theta_m \in [-10, 10]$ for $m = 1, \dots, 5$. In each case, we ran 50 randomly initialized instances of the particular SA implementation over 40 evenly-spaced proposal widths. The mixing amplitudes

used were $(A_t, A_w) = (1/3, 3)$ and the mixing probabilities used were $(p_t, p_f, p_w) = (3/5, 1/3, 1/15)$. The choice of amplitudes spans roughly an order of magnitude while the probabilities were derived by setting $p_f = 1/3$ and deriving the other two from Equation 40. Also, both the cooling constant τ and the burn-in time were set to be half the total number of time steps.

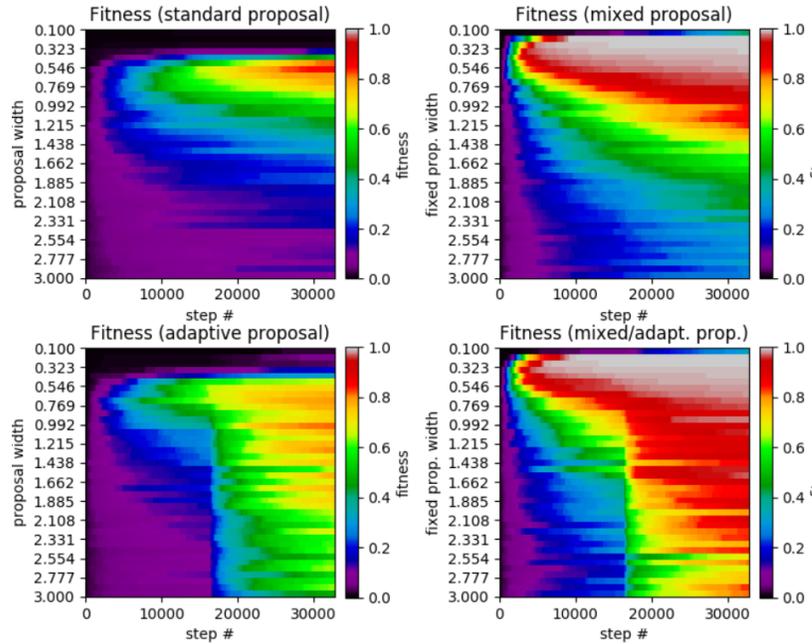


Figure 19: Results of SA test on the 5D Ackley function. Top left: standard SA. Top right: mixed SA. Bottom left: adaptive SA. Bottom right: mixed/adaptive SA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 50 runs of each method.

Comparing the results from the test, we see that standard SA achieves its highest average fitness (0.8592) at a proposal width of 0.5462, with performance dropping as the width is either increased or decreased. In the mixed SA case, we have a higher average fitness (0.9990) at a width of 0.1744. The mixed case also has the added benefit that it achieves a higher average fitness over the entire range of tested widths. This is especially true of the smallest widths under which standard SA made no

progress while mixed SA achieved high fitness. Moving on to the adaptive case, the method achieved a highest average fitness of 0.8174 at a width of 0.9923, which is lower than that of the standard case. However, it had a significantly higher performance for proposal widths larger than the standard SA’s optimal value, making it much more robust. Finally, the mixed/adaptive case seems to inherit the best aspects of both the mixed case (high peak performance) and adaptive case (robust performance across a range of widths). It achieved a highest average fitness of 0.9986 (marginally lower than the mixed case) at a width of 0.1746.

Genetic Algorithm Moving on to the results of the application of GAs to the 5D Ackley function, we have standard GA, mixed GA, adaptive GA, and mixed/adaptive GA. The parameter space was again set to $\theta_m \in [-10, 10]$ for $m = 1, \dots, 5$ with 30 randomly initialized instances over 25 different proposal widths. Identical values of the mixing amplitudes, probabilities, and burn-in time were used.

We saw a similar increase in performance from the standard case to the mixed case as was seen in the SA case. The standard GA case achieved a highest average fitness of 0.9698 at a width of 0.1520 while the mixed case achieved a highest average fitness of 0.9707 at a width of 0.0971. Unlike the SA cases, adaptation did not make significant improvement in performance. Though the adaptive case did achieve a marginally higher highest average fitness (0.9832) than the standard GA, it did not have the effect of achieving a robust level of performance across a range of widths. Similarly, the mixed/adaptive case merely performs marginally better with a highest average fitness of 0.9764. See Table 8 for a comparison of the peak performance of the various SA and GA methods.

Markov Chain Monte Carlo We now present the results of the IAC and SRF tests on the MCMC implementation of our mixing method in the context of the Ackley

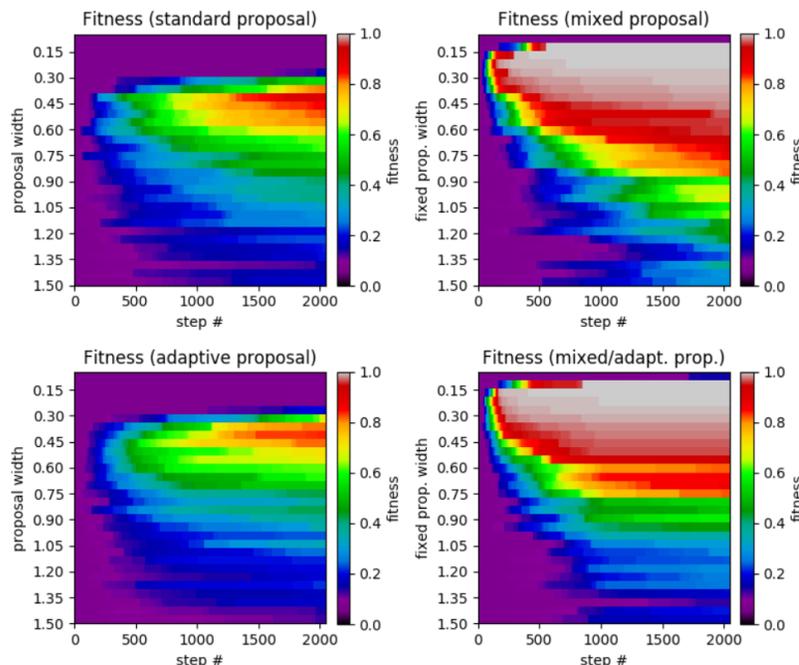


Figure 20: Results of GA test on 5D Ackley function. Top left: standard GA. Top right: mixed GA. Bottom left: adaptive GA. Bottom right: mixed/adaptive GA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 20 runs of each method.

Table 8: Summary of peak average fitness between all four SA and GA cases on the 5D Ackley benchmark problem.

	standard	mixed	adaptive	mixed/adaptive
SA	0.8592	0.9990	0.8174	0.9986
GA	0.9698	0.9707	0.9832	0.9764

benchmark. We compare results from standard Metropolis (MH), mixed Metropolis (MX), adaptive Metropolis (AM), and adaptive-mixed Metropolis (AX). We used identical parameter ranges, mixing amplitudes, and mixing probabilities. The number of step was set to $N_{step} = 2^{11}$ with $N_{burn} = N_{step}/2$. We tested a range of 20 proposal widths with 100 chains generated for each in order to obtain reliable averages. Also, recall that in the Ackley function, all parameters are identical, so we should not see a significant difference in any of the plots of each parameter.

Figure 21 shows the results of the IAC test. Examining any of the five plots,

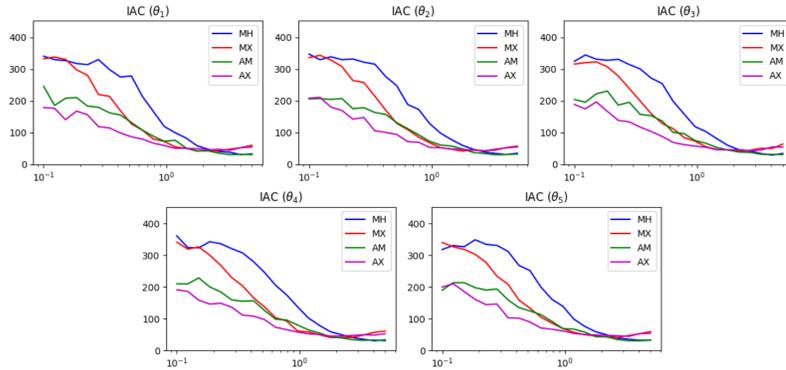


Figure 21: IAC results of MCMC test on the 5D Ackley function. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the average lower bound of the IAC.

we see that for proposal widths smaller than 2, there is a clear ordering of method performance. In decreasing order we have AX, AM, MX, then MH. For these widths, we see that the mixed methods are superior to their unmixed counterpart. For width values larger than 2, we see that the two mixed methods become poorer than their unmixed counterparts. Both of these phenomena can be explained by the presence of the widening option in the mixing scheme.

Figure 22 shows the results of the SRF test. We see similar results in this test

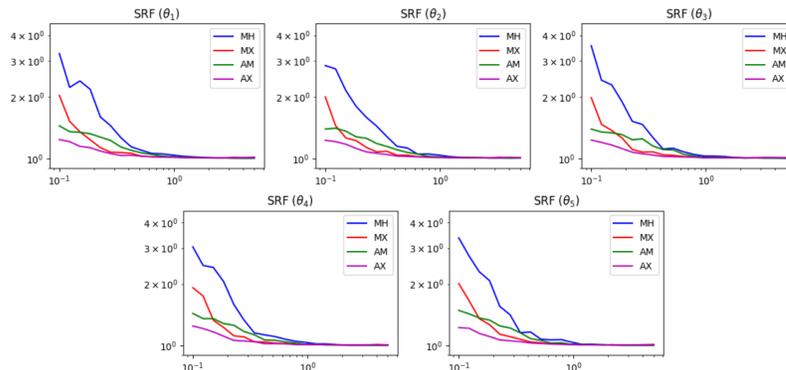


Figure 22: SRF results of MCMC test on the 5D Ackley function. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the SRF (logarithmic scale).

as the previous one. For widths below 1, the mixed methods are superior. This is reasonable, since without the widening option, the chain has difficulties escaping the local minima of the Ackley function. Also, for widths greater than 1, all methods achieve roughly the same SRF. At these widths, the chain can already escape the local minima and does not need help from the widened proposal.

6.3.2 Benchmark 2: thermal isomerization of α -pinene

For our second benchmark test, we apply our method to the thermal isomerization of α -pinene (y_1) into dipentene (y_2) and alloocimene (y_3), which then yields α - and β -pyronene (y_4) and a dimer (y_5). Assuming first-order kinetics, the ODE's for the system are given by

$$\begin{aligned}
 \dot{y}_1 &= -(\theta_1 + \theta_2)y_1 \\
 \dot{y}_2 &= \theta_1 y_1 \\
 \dot{y}_3 &= \theta_2 y_1 - (\theta_3 + \theta_4)y_3 + \theta_5 y_5 \\
 \dot{y}_4 &= \theta_3 y_3 \\
 \dot{y}_5 &= \theta_4 y_3 - \theta_5 y_5
 \end{aligned} \tag{48}$$

with analytical solutions available in [**multiresp**]. The data to which the above model must be fit is shown in Table 9 and was obtained by Fuguitt and Hawkins [**data**], who reported concentrations for the reactant and four products at 8 different times. Despite being a system of first order linear ODEs, this is a popular benchmark problem. This first two parameters are easily identifiable since they contribute the most to any error calculation. This is because are coefficients of the two variables with the largest values (see the table). However, the remaining three parameters are more challenging. The best known solution [] is $\bar{\theta}^* = ()$.

The most straightforward way to calculate a form of error for this problem would

Table 9: Data for α -pinene concentrations from Fuguitt and Hawkins [data].

t	y_1	y_2	y_3	y_4	y_5
1230	88.35	7.3	2.3	0.4	1.75
3060	76.4	15.6	4.5	0.7	2.8
4920	65.1	23.1	5.3	1.1	5.8
7800	50.4	32.9	6.0	1.5	9.3
10680	37.5	42.7	6.0	1.9	12.0
15030	25.9	49.1	5.9	2.2	17.0
22620	14.0	57.4	5.1	2.6	21.0
36420	4.5	63.1	3.8	2.9	25.7

be to simply calculate the sum of squared differences (between the data and the model) in concentration of the chemical species at the given times. Initial testing with this error calculation led to poor convergence, likely due to parameter space being littered with local maxima. Therefore, we instead choose to calculate the sum of squared differences in the derivatives

$$\epsilon(\bar{\theta})^2 = \sum_{i=2}^{N_{int}-1} \|\bar{f}(\bar{y}_i, \bar{\theta}) - (\bar{y}_{i+1} - \bar{y}_{i-1})/(2dt)\|^2 \quad (49)$$

where dt is the timestep, \bar{f} is the vectorized right-hand side of Equation 48 and the \bar{y}_i are linearly interpolated from the data in Table 9. For the SA and GA cases, we calculate fitness via

$$F(\bar{\theta}) = \left(\frac{1}{1 + \epsilon(\bar{\theta})^2} \right) \quad (50)$$

For the MCMC case, we again used a uniform prior distribution over the entire search space and used the likelihood function

$$\ell(\bar{y}|\bar{\theta}) = \exp\left(-\frac{\epsilon(\bar{\theta})^2}{2\sigma^2}\right) \quad (51)$$

where $\sigma = 15$ (again chosen to achieve an acceptance rate of between 0.25 and 0.5).

Simulated Annealing Now, we present the results of the SA test on the α -pinene application (see Figure 23). We again use the 100 samples and 30 proposal width values as in the previous problem. The parameter space is set to $\theta_m \in [0, 0.2]$ for $m = 1, \dots, 5$. The mixing amplitudes used were $(A_t, A_w) = (1/10, 2)$ and the mixing probabilities used were $(p_t, p_f, p_w) = (0.501253, 0.333333, 0.165414)$. Since the reaction coefficients which must be estimated are very small, we choose smaller amplitudes than in the Ackley case. Again, both the cooling constant τ and the burn-in time were set to be half the total number of time steps.

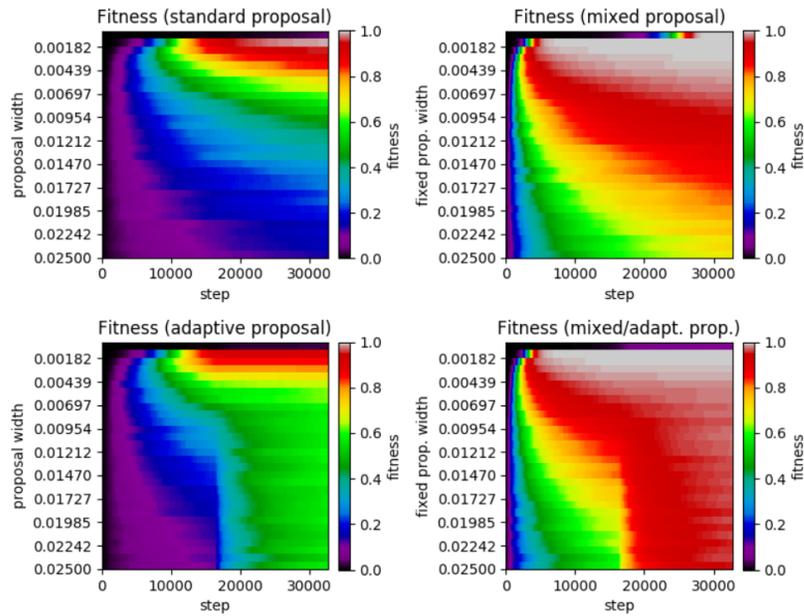


Figure 23: Results of SA test on the α -pinene application. Top left: standard SA. Top right: mixed SA. Bottom left: adaptive SA. Bottom right: mixed/adaptive SA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 100 runs of each method.

Reviewing the plot, we see very similar results to that of the Ackley test. We see that the standard SA achieved a maximum average fitness of 0.9891 at a width of 0.0010 with a significant drop in performance for larger widths. Moving on to the mixed case, the highest average fitness achieved was 0.9999 at a fixed width of

0.0001 (no smaller widths were tested). Like in the previous Ackley test, the mixed method’s performance does not decrease as rapidly when widths far from the optimal are chosen. In the adaptive case, the highest average fitness (a value of 0.9136 at a width of 0.0010) was lower than in the standard case, yet it was more robust in that it did not perform as poorly with proposal widths far from the optimal. Lastly, the mixed/adaptive case has a marginally lower highest average fitness (a value of 0.9994 at a width of 0.0010) than the mixed case. However, it was the most robust by far, consistently achieving high fitness across the entire range of tested proposal widths.

Genetic Algorithm Moving on to the results of the application of GAs to the α -pinene problem, we again have standard GA, mixed GA, adaptive GA, and mixed/adaptive GA. The parameter space was again set to $\theta_m \in [0, 1]$ for $m = 1, \dots, 5$ with 30 randomly initialized instances over 25 different proposal widths. Identical values of the mixing amplitudes, probabilities, and burn-in time were used.

The results of this test are unique across all of the experiments we performed since there is very little performance difference between the four different methods (with all methods achieving a fitness of 0.9989 or greater). Perhaps the most noticeable feature of the plots is the superior performance for the adaptive and mixed/adaptive cases (over their non-adaptive counterparts) for very small widths. Table 10 compares the peak performance of the various SA and GA methods on the α -pinene application.

Table 10: Summary of peak average fitness between all four SA and GA cases on the α -pinene application.

	standard	mixed	adaptive	mixed/adaptive
SA	0.9891	0.9999	0.9136	0.9994
GA	0.9989	0.9994	0.9995	0.9994

Markov Chain Monte Carlo We now present the results of the IAC and SRF tests on the MCMC implementation of our mixing method in the context of the α -pinene

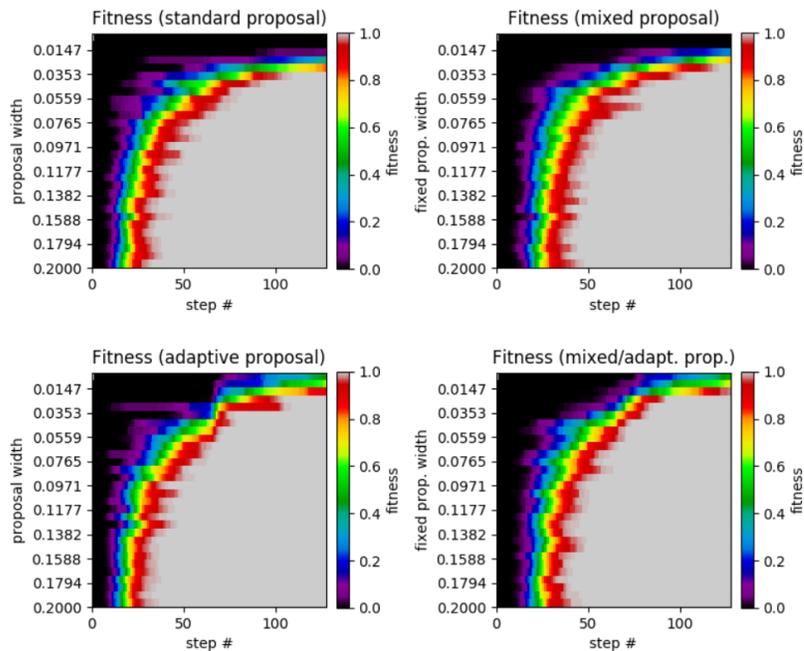


Figure 24: Results of GA test on the α -pinene application. Top left: standard GA. Top right: mixed GA. Bottom left: adaptive GA. Bottom right: mixed/adaptive GA. The horizontal axis displays the step number while the vertical axis represents the the proposal width (increases downward). The color axis displays the average fitness across the 25 runs of each method.

problem. We compare results from standard Metropolis (MH), mixed Metropolis (MX), adaptive Metropolis (AM), and adaptive-mixed Metropolis (AX). We used identical parameter ranges, mixing amplitudes, and mixing probabilities. The number of steps was set to $N_{step} = 2^{11}$ with $N_{burn} = N_{step}/2$. We tested a range of 20 proposal widths with 100 chains generated for each in order to obtain reliable averages. Also, recall that in the Ackley function, all parameters are identical, so we should not see a significant difference in any of the plots of each parameter.

Figure 25 shows the results of the IAC test. Examining any of the five plots, we see that for proposal widths smaller than 2, there is a clear ordering of method performance. In decreasing order we have AX, AM, MX, then MH. For these widths, we see that the mixed methods are superior to their unmixed counterpart. For width

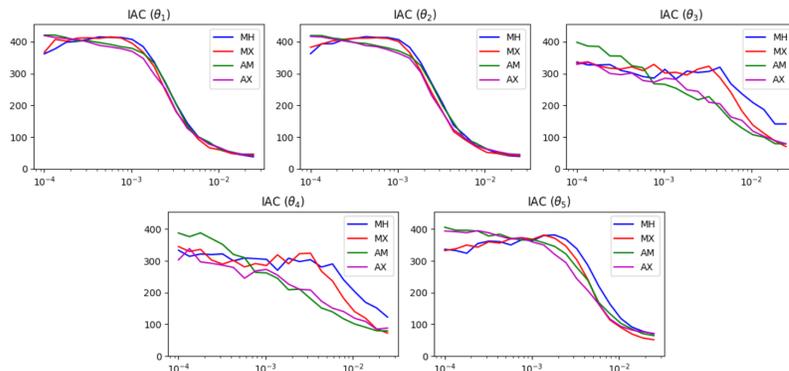


Figure 25: IAC results of MCMC test on the α -pinene problem. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the average lower bound of the IAC.

values larger than 2, we see that the two mixed methods become poorer than their unmixed counterparts. Both of these phenomena can be explained by the presence of the widening option in the mixing scheme.

Figure 26 shows the results of the SRF test. We see similar results in this test

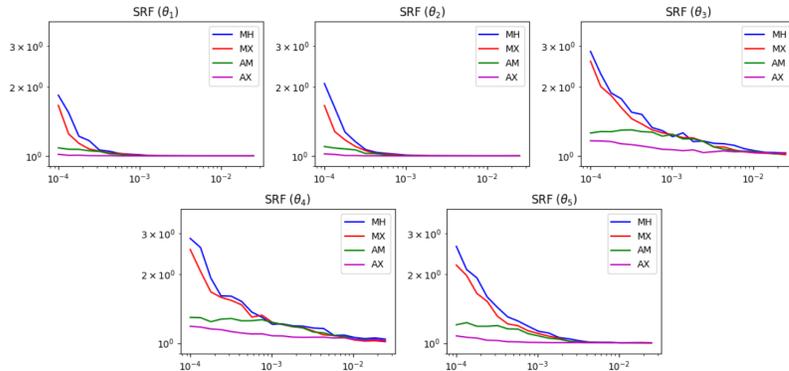


Figure 26: SRF results of MCMC test on the α -pinene problem. The horizontal axis displays the proposal width (logarithmic scale). The vertical axis is the SRF (logarithmic scale).

as the previous one. For widths below 1, the mixed methods are superior. This is reasonable, since without the widening option, the chain has difficulties escaping the local minima of the Ackley function. Also, for widths greater than 1, all methods achieve roughly the same SRF. At these widths, the chain can already escape the local minima and does not need help from the widened proposal.

6.4 Conclusions

In this paper, we have proposed and test a Gaussian kernel mixing method that can be easily implemented in a variety of stochastic optimization and Markov chain Monte Carlo contexts. In the majority of test cases, the method served to provide a substantial increase in performance when combined with the chosen base method. Regarding its use in SA, significant improvement was seen in both test problems. Similarly, in the MCMC implementation, our mixed method performed either as well or slightly better than standard Metropolis or AM in both the IAC and SRF tests. For the GA case, improvement was seen in the Ackley test problem. However, in the α -pinene problem, a small reduction in performance was seen. Further testing on more varied types of benchmark functions would be beneficial for determining for which types of problems our mixing method is best suited. Additionally, an extension of the method to beyond three discrete possibilities for mixing should be investigated. More possible mixing amplitudes could be added. Also, a continuous mixing of the form

$$q(\theta'|\theta^n) = \int_a^b \rho(A)N(\theta^n, (A\sigma)^2)dA \quad (52)$$

(with $a < 1 < b$) which preserves the variance σ^2 would be a reasonable option.

Part III

Conclusions and Future Work

Galaxies and their interactions are a fundamental part of the evolution of our cosmos. Their study grants us a deeper understanding of nature and her manifold complexity. Simulations are an indispensable tool for the modern astronomer in this endeavor, particularly when applied to the problem of fitting simulations of interacting galaxies to observational data. The contributions in this dissertation provide several new methods for accomplishing this task in an efficient and robust manner. Through utilization of the SPAM restricted three-body simulation code and Galaxy Zoo: Mergers' human scores, we were able to complete our main objectives. As discussed in Section 2, these objectives were to *develop a robust fitness function for quantifying the similarity between given target and model images* and to *develop robust methods for optimizing fitness and estimating the various model parameters for a given target image*. The first of these was completed through the quantification of tidal distortion via unperturbed model images. By incorporating this information into the fitness function, we were able to construct a function which reliably penalized models with significantly higher or lower degrees of distortion than the target. The second of these objectives was completed through the development of our real-coded genetic algorithm which was used to optimize our fitness function. Through the testing of several synthetic targets with varying morphology, we were able to 1) demonstrate the method's ability to accurately reproduce morphological features and 2) identify some principles which determine how successful the method will be given different morphologies.

In addition to these two intended goals, we also achieved several unplanned but welcome accomplishments. The first of these was a better understanding of the mor-

phological degeneracy inherent to interacting systems (in particular, restricted three-body systems). This work was cast in terms of *symmetries*, i.e., transformations of the SPAM parameters which preserve morphology. Regarding geometric symmetries, we identified that for any given set of SPAM parameters, there are seven additional distinct parameter sets which reproduce the morphology identically (though some are only up to projection), for a total 8-fold degeneracy. Moreover, there are numerous parameter transformations which only affect one galaxy in the system, further compounding degeneracy in more unpredictable ways. To combat the performance-suppressing effects of these symmetries on optimization, we simply restrict the parameters so that only one of the eight identical models is in the search space. Regarding dynamical symmetries, we showed that in cases where there is no additional v_z data to disambiguate whether the galaxy is approaching or receding, one can hold the total mass fixed and simply fit the morphology. This is due to the fact that velocities, masses, and interaction time can be scaled in a way that preserves morphology. The second of these unintended accomplishments was the development of our kernel mixing strategy for application in stochastic optimization and MCMC contexts. Originally developed during an earlier phase of our research when we were applying MCMC methods to the interacting galaxy problem, we realized that the method had potential to improve performance in a variety of contexts. We demonstrated the method’s quality by applying to two common benchmark problems.

Despite these achievements, science never sleeps. As such, we have several avenues in mind for continuing this research:

- 1. Fitting to real, observational targets.** Perhaps the most obvious next step to take would be to apply our methods to observational images of real target systems. Thus far, we have only tested our methods on various synthetic targets. Doing so allowed us to directly assess optimization performance by comparing parameter values

between the target and best-fit model. One item which will likely be required is the more realistic imaging technique being developed by another member of our team.

2. Further development of fitness functions. Our current fitness function performs well; however, there are many different techniques for constructing these functions. One which we have looked into and seems to show promise is training machine learning and statistical techniques on the GZM models. In particular, we have done preliminary testing in which we trained statistical models to predict the fitness of SPAM models. The training and testing data was obtain by using the image analysis software WNDCHRM [32] to generate various image features and coefficients. Early results show that these statistical models can achieve a high predictive accuracy with a small feature count. Surprisingly, even poorly predictive features (when combined into a single model) can achieve surprisingly high accuracy if there are a sufficient number of them. The possible downside of this technique is the need for a large amount of data in the form of pre-scored models. Thanks to GZM we have this data for 62 different targets, but that leaves the countless other targets not in the GZM dataset. Our proposed solution to this problem is to train a single model on all 62 targets simultaneously, in hopes of finding a universal statistical model for predicting fitness. If this is possible, then the problem is solved. If it is not possible, it is unclear whether the problem can be solved. Only further testing will reveal the answer.

3. Parameter estimation and uncertainty quantification of galaxy stuff. Another clear direction would be to apply our adaptive kernel mixing MCMC method to quantify the uncertainty in the SPAM parameters for a given target system. Through some other testing performed which was omitted from this dissertation (due to time constraints and the fact that it was not done in as systematic and comprehensive a way to claim the results with certainty), it seems that fitness is more sensitive to

changes in some parameters than others. For example, the masses and radii do not affect fitness as strongly as the velocities and altitude angles do. A careful analysis of the sensitivity and uncertainty of the various parameters would be very beneficial.

4. Further analysis of half-symmetries and approximate symmetries.

Though we have largely addressed the problem of degeneracy, there is still a great deal of it in the system, the majority of which stems from half-symmetries and approximate symmetries. Since the relative similarity of these with the control model are not a fixed value (the true symmetries are always a value of ≈ 1), it is difficult to predict when they pose a threat to optimization performance. A further analysis on their nature would be beneficial for further understanding the dynamics of the restricted three-body system and improving optimization performance.

REFERENCES

1. Alfonso, J. & Augerri, L. Properties of the Photometric Components of Lenticular Galaxies. *Advances in Astronomy* **321**, 1–137 (1999).
2. Appleton, P. N. & Struck-Marcell, C. Models of Ring Galaxies. I. The growth and distribution of clouds in the expanding density wave. *The Astrophysical Journal* **318**, 103–123 (1987).
3. Baker, J. E. *Adaptive Selection Methods for Genetic Algorithms* in *ICGA* (1985).
4. Barnes, J. & Hut, P. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* **324**, 446–449 (1986).
5. Chandrasekhar, S. & von Neumann, J. The Statistics of the Gravitational Field Arising from a Random Distribution of Stars. I. The Speed of Fluctuations. *The Astrophysical Journal* **95**, 489 (May 1942).
6. Curtis, H. D. Novae in Spiral Nebulae and the Island Universe Theory. *Publications of the Astronomical Society of the Pacific* **100**, 6 (Jan. 1988).
7. Davis, M., Efstathiou, G., Frenk, C. S. & White, S. D. M. The evolution of large-scale structure in a universe dominated by cold dark matter. *The Astrophysical Journal* **292**, 371–394 (May 1985).
8. Dreyer, J. L. E. A New General Catalogue of Nebulae and Clusters of Stars, being the Catalogue of the late Sir John F.W. Herschel, Bart., revised, corrected, and enlarged. *Memoirs of the Royal Astronomical Society* **49**, 1–237 (1888).
9. Gomez, J. C., Fuentes, O. & Puerari, I. *Determination of Orbital Parameters of Interacting Galaxies Using Evolution Strategies in Astronomical Data Analysis Software and Systems XI* (eds Bohlender, D. A., Durand, D. & Handley, T. H.) **281** (Jan. 2002), 409.

10. Guglielmo, M., Lewis, G. F. & Bland-Hawthorn, J. A genetic approach to the history of the Magellanic Clouds. *MNRAS* **444**, 1759–1774. arXiv: 1407.8298 [astro-ph.GA] (Oct. 2014).
11. Haario, H., Saksman, E. & Tamminen, J. An Adaptive Metropolis Algorithm. *Bernoulli* **7**, 223–242 (2001).
12. Hastings, W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
13. Hernquist, L. & Quinn, P. J. Formation of Shell Galaxies. I. Spherical Potentials. *The Astrophysical Journal* **331**, 682 (Aug. 1988).
14. Holincheck, A. J. *et al.* Galaxy Zoo: Mergers - Dynamical models of interacting galaxies. *MNRAS* **459**, 720–745. arXiv: 1604.00435 [astro-ph.GA] (June 2016).
15. Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (The MIT Press, Apr. 1992).
16. Holmberg, E. On the Clustering Tendencies Among the Nebulae. *Astrophysical Journal* **92**, 200–234 (1940).
17. Holmberg, E. On the Clustering Tendencies Among the Nebulae II: A Study of Encounters Between Laboratory Models of Stellar Systems by a New Integration Procedure. *Astrophysical Journal* **94**, 385–396 (1941).
18. Hubble, E. P. A spiral nebula as a stellar system, Messier 31. *The Astrophysical Journal* **69**, 103–158 (Mar. 1929).
19. I., O. & S., K. *A real-coded genetic algorithm for functional optimization using unimodal normal distribution crossover* in *Proceedings of the 7th International*

- Conference on Genetic Algorithms* (Morgan Kaufmann Publishers Inc., East Lansing, MI, USA, 1997), 246–253.
20. Larson, R. B. & Tinsley, B. M. Star formation rates in normal and peculiar galaxies. *Astrophysical Journal* **219**, 46–59 (1978).
 21. L.J., E. & J.D., S. *Real-coded genetic algorithms and interval-schemata* in *Second Workshop on Foundations of Genetic Algorithms* (Morgan Kaufmann Publishers Inc., San Mateo, CA, USA, 1993), 187–202.
 22. Lynds, R. & Toomre, A. On the interpretation of ring galaxies: the binary ring system II Hz 4. *The Astrophysical Journal* **209**, 382–388 (Oct. 1976).
 23. Messier, C. *Catalogue des Nébuleuses et des Amas d'Étoiles (Catalog of Nebulae and Star Clusters)* Connaissance des Temps ou des Mouvements Célestes. Jan. 1781.
 24. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21**, 1087–1092 (1953).
 25. Meyer, T. P. & Packard, N. H. Local forecasting of high-dimensional chaotic dynamics. *M. Casdagli and S. Eubank, eds., Nonlinear Modeling and Forecasting. Addison-Wesley.* (1992).
 26. Mo, H., van den Bosch, F. C. & White, S. *Galaxy Formation and Evolution* (2010).
 27. Montana, D. J. & Davis, L. *Training Feedforward Neural Networks Using Genetic Algorithms* in *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1* (Morgan Kaufmann Publishers Inc., Detroit, Michigan, 1989), 762–767.

28. Oehm, W., Thies, I. & Kroupa, P. Constraints on the dynamical evolution of the galaxy group M81. *MNRAS* **467**, 273–289. arXiv: 1701.01441 [astro-ph.GA] (May 2017).
29. Pedersen, J. & J., M. Genetic algorithms for protein structure prediction. *Current opinion in structural biology* **6**, 227–231 (1996).
30. Petsch, H. P. & Theis, C. *Determining properties of the Antennae system - Merging ability for restricted N-body* in *EAS Publications Series* **44** (Jan. 2010), 33–36.
31. Roukema, B. F., Quinn, P. J. & Peterson, B. A. *Spectral Evolution of Merging/Accreting Galaxies* in *Observational Cosmology* (eds Chincarini, G. L., Iovino, A., Maccacaro, T. & Maccagni, D.) **51** (Jan. 1993), 298.
32. Shamir, L. *et al.* Wndchrm - an open source utility for biological image analysis. *Source Code for Biology and Medicine* **3** (2008).
33. Shapley, H. Studies based on the colors and magnitudes in stellar clusters. XII. Remarks on the arrangement of the sidereal universe. *The Astrophysical Journal* **49**, 311–336 (June 1919).
34. Struck, C. Galaxy collisions. *Physics Reports* **321**, 1–137 (1999).
35. Theis, C. & Kohle, S. Multi-method-modeling of interacting galaxies. I. A unique scenario for NGC 4449? *Astrophysics and Space Science* **370**, 365–383. arXiv: astro-ph/0104304 [astro-ph] (May 2001).
36. Theis, C. Modeling Encounters of Galaxies: The Case of NGC 4449. *Reviews in Modern Astronomy* **12**, 309. arXiv: astro-ph/9907237 [astro-ph] (Jan. 1999).
37. Theis, C. & Spinneker, C. M51 Revisited: A Genetic Algorithm Approach of Its Interaction History. *Astrophysics and Space Science* **284**, 495–498 (Apr. 2003).

38. Toomre, A. *Mergers and Some Consequences in Evolution of Galaxies and Stellar Populations* (eds Tinsley, B. M. & Larson Richard B. Gehret, D. C.) (Jan. 1977), 401.
39. Toomre, A. & Toomre, J. Galactic Bridges and Tails. *The Astrophysical Journal* **178**, 623–666 (Dec. 1972).
40. Wahde, M. Determination of orbital parameters of interacting galaxies using a genetic algorithm. Description of the method and application to artificial data. *Astronomy & Astrophysics Suppl. Ser.* **132**, 417–429 (1998).
41. Wallin, J. F., Holincheck, A. J. & Harvey, A. JSPAM: A restricted three-body code for simulating interacting galaxies. *Astronomy and Computing* **16**, 26–33. arXiv: 1511.05041 [astro-ph.GA] (July 2016).