Efficacy of Enforced Parsimony on a Machine Learning Model of Gravity

Joseph Ehlert

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Mark K. Transtrum, Advisor

Department of Physics and Astronomy

Brigham Young University

April 2022

ABSTRACT

Efficacy of Enforced Parsimony on a Machine Learning Model of Gravity

Joseph Ehlert
Department of Physics and Astronomy, BYU
Bachelor of Science

Many machine learning models are often overly complicated and require simplification, making it difficult to use them to discover fundamental physical laws. We examine the role of parsimony in the scientific process using a 14-parameter, model of gravity created by the *SirIsaac* algorithm, an S-Systems model. S-Systems, a universal function approximator for dynamical systems, are an interesting case study because they include true gravity, i.e., the inverse square law, as a special case. We explore the question whether model reduction methods can find true gravity as an optimal approximation to the machine-learned *SirIsaac* model. We use the Manifold Boundary Approximation Method (MBAM) as a parameter reduction algorithm. MBAM is a computational approach based on the information geometry of the model. We found that MBAM produces a reduced model of *SirIsaac* that accurately describes the four orbits of Newtonian gravity (circular, elliptical, parabolic, and hyperbolic). The final reduced model is different than Newtonian gravity, although the two reduction paths share four limits. By using two subsets (bound and unbound orbits, respectively) of the data, we identified, via MBAM, a model that accurately fit each subset. We find that all the limits necessary for Newtonian gravity appear in at least one of the reduction paths of the bound and unbound orbits.

Keywords: machine learning, information geometry, model manifold, sloppy model, *SirIsaac*

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Mark Transtrum, for his expertise and time. His questions have inspired me to be a sharper thinker and better researcher. I would also like to thank Dr. Chesnel, Dr. Bergeson, and Prof. Galley for their help and comments on writing this thesis. I would also like to acknowledge the help of Yonatan Kurniawan, whose ideas and thoughts have frequently sparked my research. Finally, I would like to thank my family members and friends who have supported me throughout my time at BYU.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Science, physics in particular, has long adhered to the reductionist hypothesis, meaning that the physical world is controlled by the same set of fundamental laws and that understanding those laws will help physicists create a Theory of Everything [1], [2]. This hypothesis is one reason that physicists break matter into smaller and smaller parts - to understand how the smallest constituents of matter behave to build a coherent Theory of Everything. Earlier scientists, especially those in the 20^{th} century, used reductionist thinking to produce many scientific laws, such as quantum mechanics [2]. In contrast, Nobel Laureate Phil Anderson [1] argues that although the sciences can be described using a hierarchy of complexity, each level of complexity is more than just an applied version of the less complex and more fundamental previous level. Each stage of the hierarchy needs new laws and concepts that require creative scientific input. Laughlin and Pines [2] similarly argue that the reductionist hypothesis has peaked as a primary guiding principle of science. Rather, physics will be guided by understanding the connections and different properties between different levels of complexity, dubbed "the study of complex adaptive matter" [2].

The study of complex adaptive matter requires the marriage of theoretical physics with computational methods. Theory is important in physics because it allows extrapolation and physical interpretation. In recent years, new computational modeling algorithms have enabled discoveries in many fields of physics; however, these new modeling algorithms have created a disconnect between the physical theory and the computational model [3]. This disconnect necessitates different algorithms for physics and other scientific modeling that can assist in creating and even form theories. Such a computer-based scientist could help drive the study of complex adaptive matter by engaging in the scientific method, including experimenting, collecting data, modeling, theorizing, and advising potential future experiments.

We focus on the modeling portion of a computer-based scientist using machine learning. Machine learning constitutes algorithms that learn relationships between input data and output data. Recent increases in computing speed and power have enabled the widespread use of machine learning in science. Scientists have been successful in using machine learning to model many topics, including speech recognition [4] and cancer prediction [5]. Using symbolic regression, a type of machine learning algorithm, physicists found some success in reproducing Hamiltonian mechanics [6]. Machine learning provides a powerful modeling tool for a future computer-based scientist, because it does not use extensive *a priori* knowledge to find a relationship.

Despite the successes of machine learning, it still has several issues, especially in the context of a computer-based theorist. I will highlight two of these issues. First, models created by machine learning algorithms tend to be sloppy, i.e., a model whose behavior is determined by relatively few parameter combinations [7]. I will call this problem the sloppiness problem. Model sloppiness implies overparameterization, or using too many parameters. Although overparameterization is closely related to sloppiness, they are separate concepts; whereas overparameterization is more parameters than needed, sloppy models have only a few parameters that determine the behavior of the model and the other parameters have little effect. Second, machine learning models are often

difficult to physically interpret, meaning that the parameters fail to have physical meaning. I will call this problem the interpretability problem. Interpretability, or the ability to interpret the meaning of a parameter and the model as a whole, is not only a problem in physics machine learning, but also in machine learning in general [8]. This problem is magnified in physics because physicists often want model parameters to have physical meaning that relate to the real world.

To solve these two problems, we turn to a guiding principle in the scientific method: parsimony. Scientists follow Occam's Razor, i.e., given two theories, all other things being equal, the simpler of the two is more likely to be true [9]. One prominent example of Occam's Razor at work in science is the transition from the Ptolemaic system to the Newtonian system. The Ptolemaic system supposed that the earth was the stationary center of the universe and that all celestial bodies moved around the earth in epicyclic orbits. When a new inconsistency was discovered with planetary motion, astronomers would add another epicycle to the orbit of a planet to match the data. In the Newtonian system, the sun is the center of the solar system and orbits of all the planets are ellipses, rather than epicycles. The Newtonian system explained planetary motion just as well as the Ptolemaic system, but because it was simpler, it was gradually accepted by the scientific community. It later turned out that the Newtonian system could also predict the parabolic and hyperbolic trajectories of celestial bodies. Because simplicity has played a role in the human scientific method, parsimony and model reduction may be able to solve the sloppiness and interpretability problems.

We examine the impact of parsimony by systematically reducing a machine learned, gravity-based model into its simplest form in an attempt to reestablish Newton's law of gravity. Reestablishing a physical law could help us begin to answer several questions: how does simplicity relate to the scientific process and what are the limits of simplicity as a guiding principle in a computer-based scientist? With this motivation in mind, I will explain Newtonian Gravity in Section 1.2. I will then explain the machine-learning algorithm, named *SirIsaac*, in Section 1.3. Finally, I will describe the specific set of research questions we investigated in Section 1.4.

1.2 Newtonian Gravity

Newtonian gravity refers to the two-body problem and provides a simple case for machine learning to reproduce for several reasons. First, Newtonian gravity is a well understood system - it has been studied for several hundred years. Second, Newtonian gravity is a relatively simple dynamic system - it can be described by a single second order differential equation. Third, we already know the answer, making it easy to check if we are correct. If machine learning and model reduction techniques can identify Newtonian gravity, then other, more complex physical laws may also be discovered using similar methods.

Since computers usually solve only first-order differential equations, we need to split the one second-order differential equation into two first-order differential equations. The following derivation of Newtonian gravity is taken from [10]. For a mass, m, in orbit around another mass, M, such that $M \gg m$, distance r is given by

$$\frac{d^2r}{dt^2} = \frac{h^2}{r^3} - \frac{GM}{r^2}, \tag{1.1}$$

where $h = (\mathbf{v}_0 \cdot \hat{\theta})r_0$ is the specific angular momentum, $\mathbf{v}_0$ is the initial velocity, r_0 is the initial distance, $\hat{\theta}$ is the unit vector perpendicular to the line between the two masses and G is the gravitational constant. By measuring the distance in units of $\frac{GM}{v_0^2}$, measuring the time in units of $\frac{GM}{v_0^3}$, and setting the initial velocity parallel to $\hat{\theta}$, we obtain

$$\frac{d^2r}{dt^2} = \frac{1}{r^2}\left(\frac{r_0^2}{r} - 1\right). \tag{1.2}$$

Rewriting this as two first-order differential equations, the dynamics become

$$\begin{aligned} \frac{dr}{dt} &= \chi - 1 \\ \frac{d\chi}{dt} &= r_0^2 r^{-3} - r^{-2}, \end{aligned} \tag{1.3}$$

where $\chi = \frac{dr}{dt} + 1$. These unit conventions will be used throughout this thesis. This version of Newtonian gravity I will refer to as true gravity for the remainder of my thesis. In the next section, I will explain *SirIsaac* and point out its important features.

Parameter	*SirIsaac* value	True gravity value
α_1	0.0332082	1
β_1	0.0507828	1
g_{11}	0.0490472	0
h_{11}	0.0463394	0
g_{10}	3.42771	0
h_{10}	2.93633	0
g_{12}	7.3715	1
h_{12}	4.92514	0
β_2	0.994249	1
g_{22}	0.0144952	0
g_{20}	0.651365	-2
h_{20}	4.28848	0
g_{21}	3.43506	3
h_{21}	1.59462	2

Table 1.1 Original parameter values as fit by *SirIsaac*. The *SirIsaac* equations are found in Eq. (1.4).

1.3 *SirIsaac*

Daniels and Nemenman [10] created *SirIsaac*, an S-Systems class algorithm. S-Systems are a class of models that act as universal function generator for dynamical systems, meaning that it can approximate a model for any system described by differential equations. The *SirIsaac* algorithm walks through a hierarchy of differential models, starting with the simplest (least parameters) and working its way to the most complex (most parameters). For further specifics on the *SirIsaac* algorithm, see [10].

Daniels and Nemenman created gravitational data for *SirIsaac* using Eq. (1.3) and adding random noise to the data to imitate real world data [10]. The dynamic model created is given by,

$$\begin{aligned} \frac{dr}{dt} &= \alpha_1 r_{init}^{g_{10}} r^{g_{11}} X_2^{g_{12}} - \beta_1 r_{init}^{h_{10}} r^{h_{11}} X_2^{-h_{12}} \\ \frac{dX_2}{dt} &= r_{init}^{-g_{20}} r^{-g_{21}} X_2^{-g_{22}} - \beta_2 r_{init}^{-h_{20}} r^{-h_{21}}, \end{aligned} \tag{1.4}$$

where r_{init} is the initial condition of the radius, r is the radius, X_2 is an unobserved state variable for the second order equation and the remaining fourteen symbols are parameters, whose values are found in Table 1.1. From here on, Eq. (1.4) will be referred to as *SirIsaac*. It is important to note that true gravity, Eq. (1.3) is a special case of *SirIsaac*, Eq. (1.4). This fact means that with parameters set to certain values, found in the right column of Table 1.1, *SirIsaac* will have the same equation of true gravity.

1.4 Research Question

We explore the role of simplicity in the scientific process using the Manifold Boundary Approximation Method (MBAM) applied to a model of gravity learned by the *SirIsaac* algorithm [11]. Since parsimony plays an important role in the scientific reasoning of humans, we conjecture that simple models may be important for a computer-based scientist. We begin from an overparameterized, machine-learned *SirIsaac*. This model captures the behavior of the two-body gravity problem and contains, as a special, simplified case, true gravity. Here, true gravity refers to the inverse square law derived by the human Sir Isaac Newton. True gravity can derived by setting to zero a handful of parameters in the computer-generated *SirIsaac* model. We then ask: Can MBAM systematically reduce *SirIsaac* into true gravity by setting these parameters to zero? If we do not recover true gravity, what does an MBAM-reduced *SirIsaac* look like and how similar is it to true gravity? Is the reduced *SirIsaac* simpler (fewer parameters) or more complex (more parameters) than true gravity?

Additionally, we explore the role that different types of data play in the modeling processes.

Scientists never have all possible data at their disposal. Indeed, the human Newton derived the inverse square law from data for near-earth trajectories and for the orbit of Mars. However, the inverse square law accurately models other qualitatively distinct types of behaviors, including the parabolic and hyperbolic orbits of comets. Similarly, we generate data from four qualitatively distinct types of orbits: circular, elliptical, parabolic, and hyperbolic. We then consider the data from bound orbits (circular and elliptical) and unbound orbits (parabolic and hyperbolic), each in isolation. We then use MBAM to reduce *SirIsaac* into models constrained by data from only the bound and unbound orbits, respectively. How do these two models relate to true gravity and to the MBAM-reduced *SirIsaac*? How are the details of the reduction process of all four models different? How are they the same?

Chapter 2

Methods

In this section, I will introduce the general methods used to approach the research question. Section 2.1 will discuss the concepts necessary to perform model reduction using MBAM. Then, I will show in Section 2.2 what the data looks like and how we split the data into subsets.

2.1 Manifold Boundary Approximation Method

We use MBAM as the model reduction technique to answer our research question. The discussion in Section 2.1 comes from the discussion in [12]. MBAM is a model reduction technique that relies on the geometry of the model to reduce parameters one at a time. Three things are necessary for MBAM [12]: (1) a collection of data, denoted by $\{y_m\}_{m=1}^{M}$, where M is the number of data points, and m gives a specific data point; (2) a family of models to make predictions, denoted by $\{f_m(\theta)\}_{m=1}^{M}$; (3) a metric for comparing the model predictions to the data, or cost. To illustrate key ideas about MBAM throughout this section, we have devised a 2-parameter toy model given by

$$f(t;\theta) = \frac{1}{t^2 + \theta_1 t + \theta_2}, \tag{2.1}$$

where f is the model, t is the independent variable, θ represents a vector of parameters, $[\theta_1, \theta_2]^T$, and T represents transpose. Although many models, especially machine learning models, have many more parameters and predictions, a simple model with fewer parameters and predictions is ideal for visualizing each step in the MBAM process. We will predict using this model at three time points, $t = [1.0, 2.0, 3.0]$, while the data is given by $y_m = [1/3, 1/7, 1/13]^T$.

2.1.1 Cost and Cost Surfaces

The metric for comparing the model predictions and data is referred to as the cost. The cost can take on a variety of options, but it is most often computed as the weighted least squares:

$$C(\theta) = \frac{1}{2} \sum_{m=1}^{M} r_m(\theta)^2 \tag{2.2}$$

where r_m are the residuals, given by

$$r_m(\theta) = \frac{y_m - f_m(\theta)}{\sigma_m}, \tag{2.3}$$

where the residuals depend on the inverse weight σ_m. The best fit of the model is a vector, denoted θ^*, and defined as the point with the lowest cost, i.e., the best fit is found by minimizing the cost with respect to θ. The concept of best fit is illustrated in Fig. 2.1. The lines that lie closer to the data points have a lower cost and are therefore considered better fit.

As an important side note, we have restricted $\theta_i \geq 0$ in Eq. (2.1). Enforcing a positive parameter is a common physical constraint on parameter values. Enforcing positive parameter values suggests log transforming parameters, since logarithms have a strictly positive domain. Throughout the remainder of this section, all values of θ are assumed to be log-transformed.

The cost surface is the generalization of the cost for a single set of parameters of a model. Every θ vector has a cost associated with it. The cost surface is the result of plotting the cost at many points as we systematically vary θ across its domain. Since a cost surface is plotted in parameter

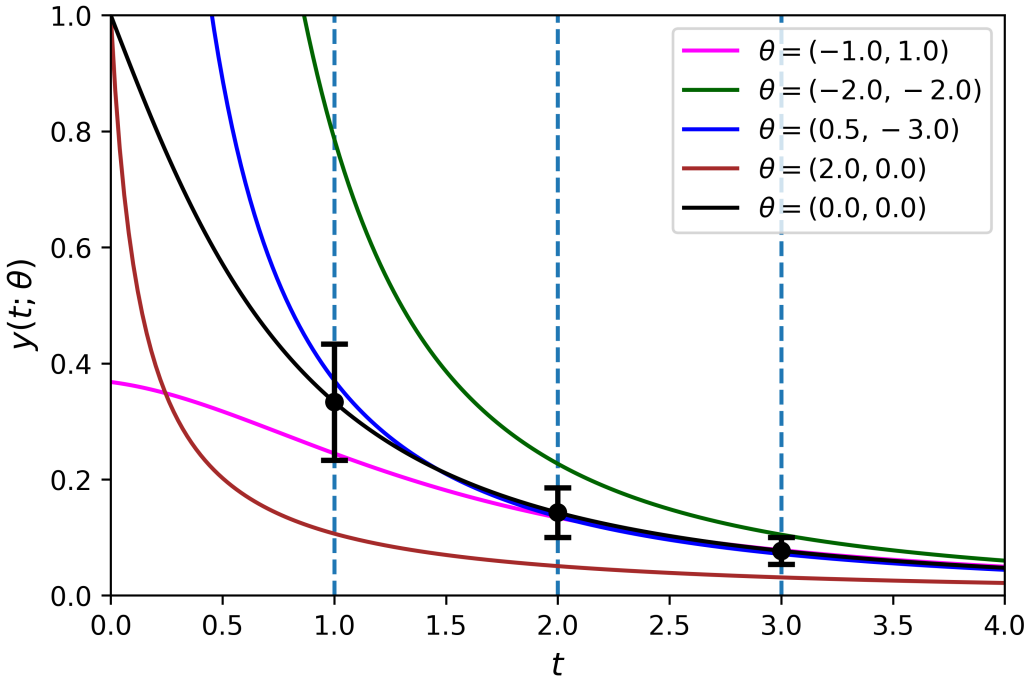

Figure 2.1 Various fits for the time series data, y_m, for the toy model in Eq. (2.1). The black dots are the data points with error bars. Each line represents a different value of the parameters, θ. By varying θ, we observe a variety of model predictions with different costs. The cost is lower the closer the line is to the data, therefore, the black line is the best fit. Note that the parameter values in the legend are log-transformed, meaning that negative numbers are associated with approaching 0 and positive numbers are associated with approaching infinity. Figure extracted with permission from [12].

space, creating a cost surface requires the same number of dimensions as there are parameters in the model. The cost surface for the toy model is shown in Fig. 2.2. We can visually associate the best fit prediction with the place of lowest cost on the cost surface. However, the model in Eq. (2.1) is sloppy, meaning that it is insensitive to coordinated variations in some parameter directions. The key way to identify sloppy models from the cost surface is by the aspect ratio of the cost canyons; when the cost surface forms long, thin canyons, the model is sloppy, and when the cost surface forms a bowl-like structure, the model is not sloppy. The canyons of low cost indicate that the predictions of the model change minimally as we vary parameters in a parallel direction to the canyon. Conversely, a small change in the data in the prediction space could inadvertently lead to a large change in best fit parameter values. For example, reviewing Fig. 2.1, if y at $t = 1.0$ increased slightly, the blue line may fit much better than the black line, yet the parameters of the blue line ($\theta = [0.5, -3.0]^T$) are not close to the parameters of the black line (($\theta = [0.0, 0.0]^T$). Therefore, a small amount of noisiness or uncertainty in the data can lead to a large amount of uncertainty in the best fit parameters. Finding parameter combinations with large uncertainty is central to model reduction with MBAM. Parameter combinations with large uncertainty are called unidentifiable parameter combinations because the predictions will look the same even as we adjust parameters parallel to the cost canyons.

2.1.2 Fisher Information Matrix

To quantify the uncertainty of θ^*, we use a linear approximation of the cost surface in the vicinity of the best fit. First, we linearize the residuals around θ^*:

$$r_m(\theta) \approx r_m(\theta^*) + \frac{\partial r_m}{\partial \theta}(\theta - \theta^*). \tag{2.4}$$

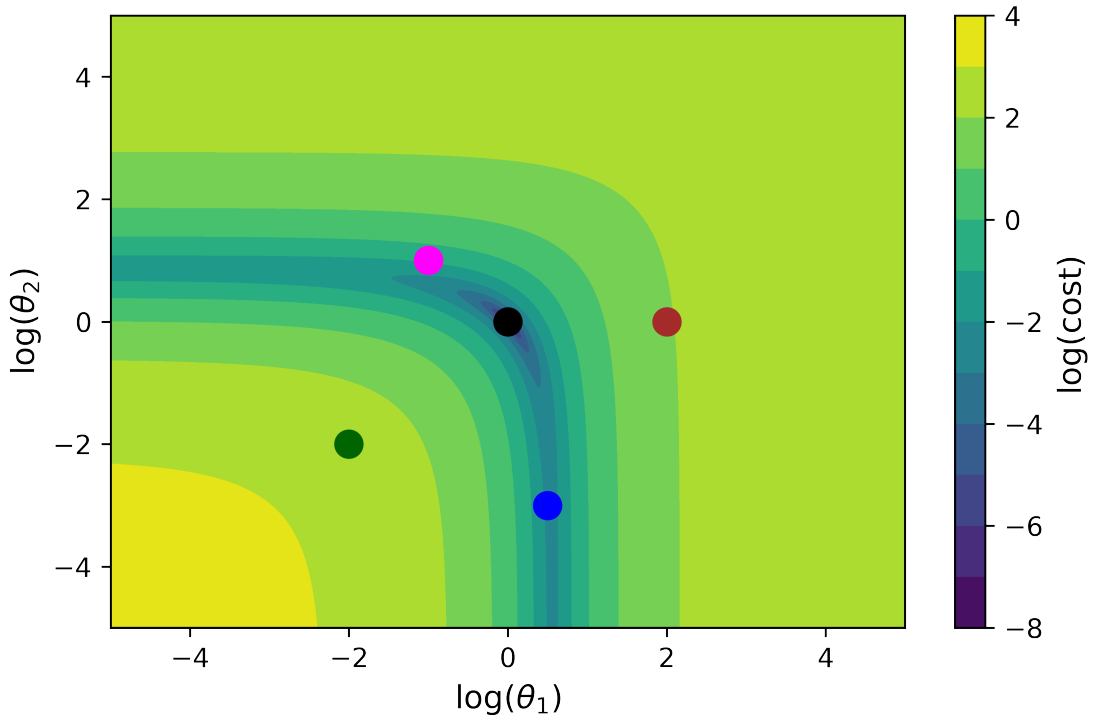

Figure 2.2 The cost surface of the toy model, Eq. (2.1). We created the plot by varying both $\log(\theta_1)$ (x-axis) and $\log(\theta_2)$ (y-axis) between -5 and 5. Darker colors represent lower cost, meaning that the darkest point is the point of best fit. Each colored dot is associated with the prediction line of the same color in Fig. 2.1. This shows again that the black dot (represented by the black line in Fig. 2.1) is the best fit at $(0,0)$. It is also significant to note there exist canyons of low cost that extend from the point of best fit. Figure extracted with permission from [12].

Since the cost is at a minimum at the best fit, we know that $\nabla C = 0$ at θ^*, where ∇ is the gradient. By substituting Eq. (2.4) into the cost function Eq. (2.2) and using the fact that $\nabla C = 0$, we obtain

$$C(\theta) \approx C(\theta^*) + \frac{1}{2}(\theta - \theta^*)^T (J^T J)(\theta - \theta^*), \tag{2.5}$$

where we have introduced the Jacobian of the residual function, J, and T is again the transpose. The Jacobian is defined in Eq. (2.6) and is evaluated at θ^*,

$$J_{m,n} = \frac{\partial r_m}{\partial \theta_n} = -\frac{1}{\sigma_m}\frac{\partial f_m}{\partial \theta_n}. \tag{2.6}$$

The quantity $J^T J$ found in Eq. (2.5) is an important statistical quantity known as the Fisher Information Matrix (FIM):

$$\mathcal{I} = J^T J. \tag{2.7}$$

The inverse of the FIM is the covariance matrix of parameter uncertainty, giving a quantification of the parameter uncertainty.

The FIM also describes the local geometry of the cost surface, as shown in Fig. 2.3. Local cost contours form ellipses that can be described by the FIM. The diagonals describe the change in cost to each parameter individually. The cost contour ellipses are aligned with the eigenvectors of the FIM, with an aspect ratio given by the square root of the ratio of eigenvalues (λ). Long directions in the ellipse are parallel to eigenvectors with small eigenvalues. The parameter combinations with small eigenvalues will have greater uncertainty, and therefore carry less information about the data. In addition, the projection of the ellipses onto the axes estimates the uncertainty of a parameter, given by the diagonals of the inverse FIM. The directions with small eigenvalues of the FIM are also useful for isolating the unidentifiable parameter combinations.

2.1.3 Model Manifolds

The cost surface and FIM help us determine the best fit and quantify the parameter uncertainty, but we can also understand the model using a high-dimensional model manifold. The model manifold is

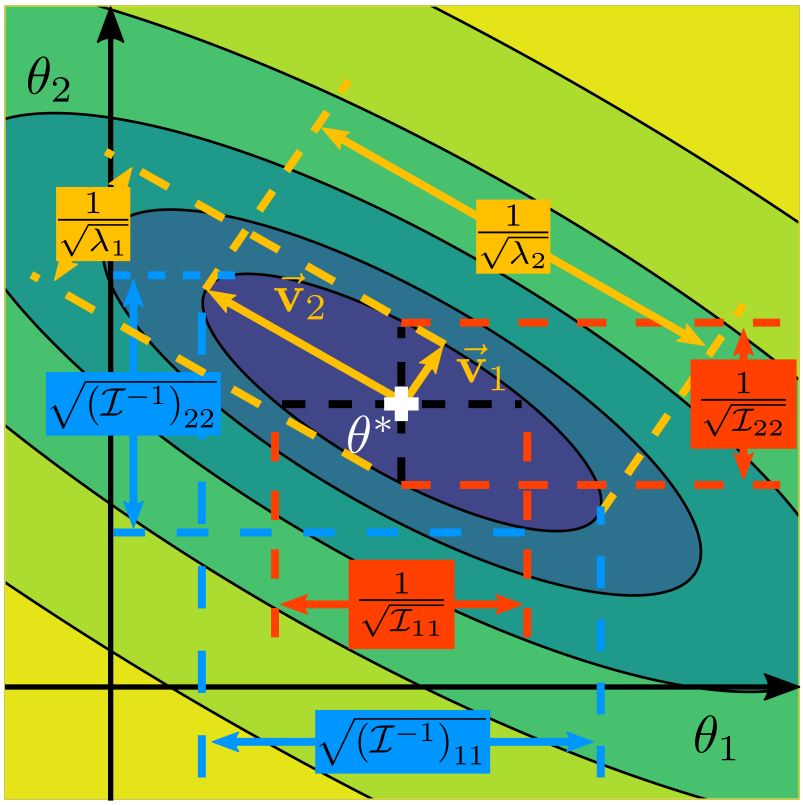

Figure 2.3 The local geometry of the cost surface of the best fit described by the Fisher Information Matrix for Eq. (2.1). Local cost contours are described by ellipses. $\vec{v}_i$ and λ_i represent the eigenvector and eigenvalue, respectively, of the FIM. They describe the direction of the major axes and aspect ratio of the cost contour ellipses. The diagonals of the FIM represent the change in cost of each parameter individually, and the diagonals of the inverse FIM are used to calculate the uncertainty for each parameter. Figure extracted with permission from [12].

constructed by mapping all possible parameter values to their predictions in data space. This process creates a surface in data space that is the set of all possible predictions called a model manifold. The axes for a model manifold are the value of the model at every time point in the data - for the toy model, with three time points, there would be three dimensions, each with associated with the value of the model at the respective time. Since distance in data space corresponds to cost, the best fit is the point on the model manifold closest to the data and, conversely, the farther from the data, the higher the cost. The model manifold of the toy model, Eq. (2.1), is found in Fig. 2.4. The model manifold of the toy model is a two dimensional surface in three dimensional space because there are two parameters and we predict at three time points.

Notice in Fig. 2.4 that the manifold is bounded by two one-dimensional segments. These boundaries suggest that even though parameter space extends to infinity, there is a finite limit to possible predictions, i.e., infinite distances in parameter space are mapped to a finite distance on the model manifold. This phenomena explains why canyons appear in the cost surface: as parameter values change an infinite amount, the distance from the best fit remains unchanged on the model manifold. Therefore, the boundaries have a corresponding unidentifiable parameter combinations. The bounded segments are a manifold of co-dimension one, meaning that a 1-D line can intersect the boundary; in a higher-dimensional model, the same rule of co-dimensions applies - a boundary of co-dimension one is a surface with one dimension less than the manifold itself. The distance between the best fit and each boundary informs the cost associated with taking the limit associated with that boundary. The distance between the best fit and each boundary creates a hierarchy of costs associated with each limit, with the closest boundaries corresponding to the least change in cost, and the farthest boundary corresponding to the greatest change in cost. In order to reduce the model using MBAM, we take the limit associated with the closest boundary.

We use geodesics to find the unidentifiable parameter combinations associated with each boundary. A geodesic is the generalization of a straight line on a curved surface, i.e., a distance

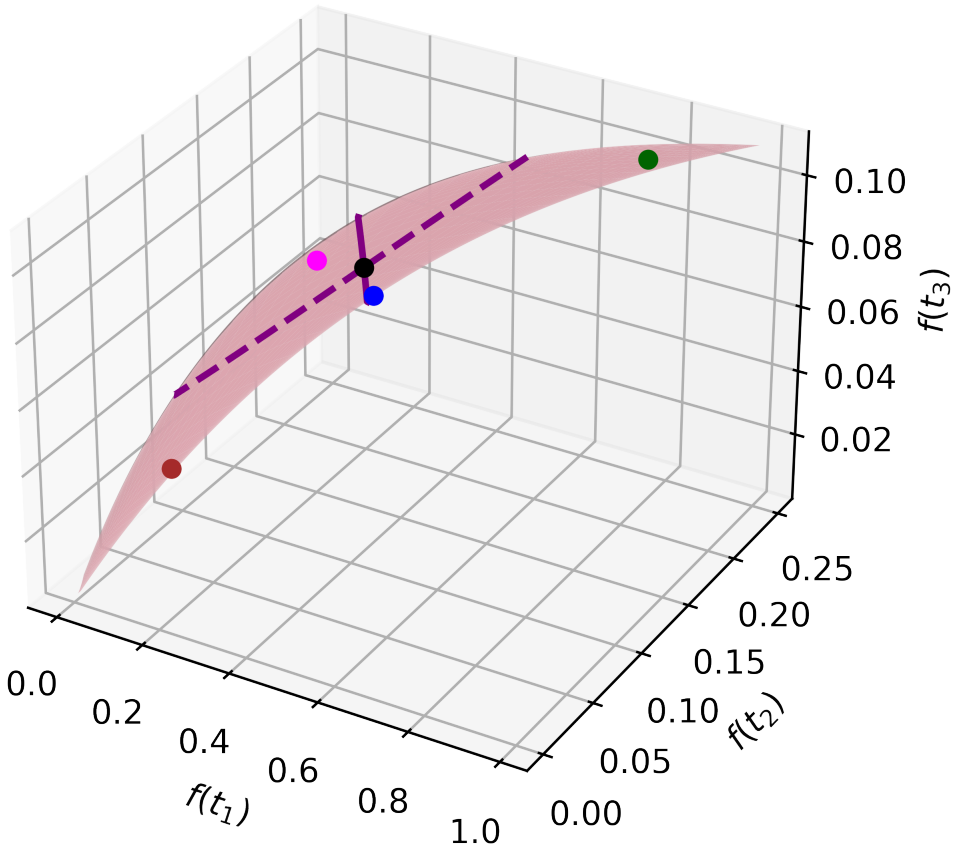

Figure 2.4 Model manifold of the toy model. Every point in parameter space is associated with a set of predictions in data space. The axes are the values of the model at three time points, where f is given by Eq. (**??**). The colored dots correspond to the models found in Fig. 2.1, Fig. 2.2 and Fig. 2.5, and correspond to various parameter values. There are two geodesics (dashed and solid lines) calculated that radiate outward from the best fit. Figure extracted with permission from [12].

minimizing curve on a specific surface. [13] A geodesic is calculated by solving the geodesic equation,

$$\frac{\partial^2 \theta^i}{\partial \tau^2} = -\sum_{j,k} \Gamma^i_{jk} \frac{\partial \theta^j}{\partial \tau} \frac{\partial \theta^k}{\partial \tau}, \tag{2.8}$$

where

$$\Gamma^i_{jk} = \sum_{l,m} (\mathcal{I}^{-1})^{il} \frac{\partial y_m}{\partial \theta^l} \frac{\partial^2 y_m}{\partial \theta^j \partial \theta^k} \tag{2.9}$$

are the Christoffel symbols, $\mathcal{I}$ is the FIM, θ are the parameters, and τ is the arclength of the geodesic along the model manifold. We solve the geodesic equation numerically as an initial value problem, where the initial conditions are given by the best fit and the initial direction is given by an eigenvector, starting with the sloppiest one. We use the toy model to illustrate geodesics: Fig. 2.4 shows the model manifold with two geodesics through the best fit and Fig. 2.5 shows the cost surface with the same two geodesics through the best fit, in addition to the two eigenvectors used as the initial direction. Note that while the geodesics remain mostly straight on the model manifold in data space, they change direction in parameter space to align with the cost canyons or unidentifiable parameter combinations. From the cost surface, we can see that either $\theta_1 \to 0$ or $\theta_2 \to 0$ at the end of the geodesic. By comparing the cost surface with the model manifold picture, we can deduce that $\theta_1 \to 0$ corresponds with the upper boundary of the model manifold because three geodesics end at that boundary in both the cost surface and the model manifold. The comparison also implies that $\theta_2 \to 0$ corresponds with the lower boundary. These limits indicate the unidentifiable parameter combinations, and, by comparing the arclength of the geodesic on the model manifold, called τ in Eq. (2.8), we can isolate the best approximation of the model.

With a more complex model, like the 14-parameter *SirIsaac*, isolating the unidentifiable parameter combinations visually becomes impossible. Complex models, therefore, necessitate the use of the geodesic for selecting the unidentifiable parameter combinations. In addition, for the simple model with two parameters, the unidentifiable parameters consisted of only one parameter going to 0; in more complex examples, unidentifiable parameter combinations are often coordinated

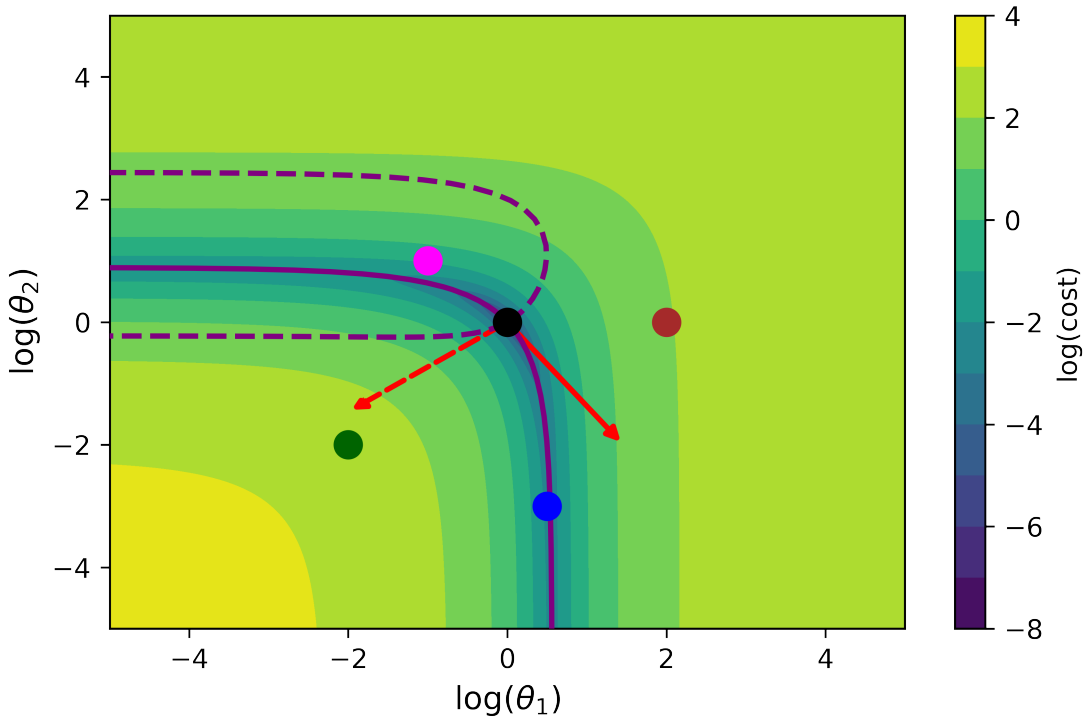

Figure 2.5 Cost surface of the toy model with geodesics. The red arrows indicate the directions of the eigenvectors of the FIM. Two geodesics (dashed and solid purple lines) radiating outward from the best fit, corresponding to those in Fig. 2.4, are also shown. Figure extracted with permission from [12].

combinations of parameters, i.e., multiple parameters are limited on one boundary of the model manifold. A geodesic can identify all parameters associated with a manifold boundary, and find an identifiable reparameterization of the model.

2.1.4 Hasse Diagrams

The Hasse diagram is a diagram that shows the limits associated with unidentifiable parameter combinations, mapping the reduction path of MBAM. The reduction path refers to the set of limits found by MBAM to reduce a model. A Hasse diagram for the toy model is found in Fig. 2.6. Every node of the Hasse diagram represents a different model with the number of parameters given on the left. The first node, or the top node, represents the original model, found in Eq. (2.1). Since two limits were identified with geodesics ($\theta_1 \to 0$, $\theta_2 \to 0$), there are two possible reductions paths shown by arrows labeled with the accompanying unidentifiable parameter combination. The next level of limits would reduce the original model to zero parameters and is also shown in the Hasse diagram. A reduction path is constructed by following the limits found by MBAM and mapping those limits onto the Hasse diagram. Although I have shown all possible reduction paths in the toy model Hasse diagram, to simplify the diagrams, the Hasse diagram will show only the reduction path taken by MBAM, rather than all possible paths.

2.1.5 MBAM as an Iterative Process

Using the computational methods described above, I will now describe how MBAM works as a reduction method. MBAM is an iterative process that systematically approximates a model one unidentifiable parameter combination at a time. First, we find the best fit of the model using a Levenberg-Marquardt algorithm and check that the model approximates the data well. [14] The exact workings of the Levenberg-Marquardt are beyond the scope of this thesis, but it is an optimized algorithm for fitting data with complex cost surfaces. Once we have the best fit, we solve several

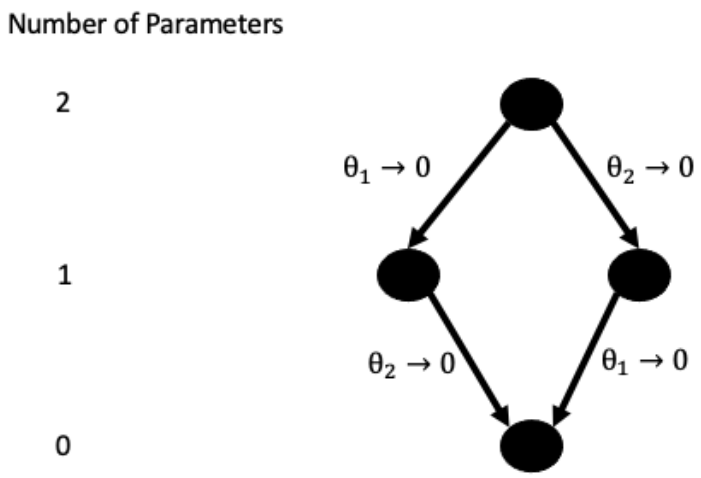

Figure 2.6 Hasse Diagram for the toy model. A Hasse diagram demonstrates the model reductions found using MBAM. Each node represents a model with the number of parameters given on the left. The top node represents the full, original model and each arrow represents a reduction, labelled with the corresponding limit, therefore the farther down you go in a Hasse diagram, the simpler the model is.

geodesics to find several unidentifiable parameter combinations. We select the unidentifiable parameter combination with the smallest τ, the distance from the best fit on the model manifold, and take the associated limit to reduce the model. By taking this limit, our new, reduced model manifold becomes the co-dimension one manifold associated the bounded segment of the original model manifold. Using the reduced model and its manifold, we repeat the process of finding the best fit of the reduced model, calculating several geodesics, isolating and taking the limit of the closest unidentifiable parameter combination until the geodesic no longer finds an unidentifiable parameter combination or until the best fit no longer approximates the data well. We used MBAM to find and document a reduction path of *SirIsaac*.

2.2 Understanding the Data

There are four regimes of orbits in the two-body problem: circular, elliptical, parabolic, and hyperbolic. Each orbit results from different initial conditions. In the setup of *SirIsaac*, we fit the data to the radius, i.e., the radius is the quantity of interest. In Fig 2.7, we see the data split into the four orbits: the blue line is the original *SirIsaac* model (Eq. (1.4)) and the orange line is true gravity, Eq. (1.3). The two models are near perfect replicas in the elliptical, parabolic and hyperbolic regimes, but appear to be quite different in the circular regime; note that the scaling changes on each y-axis, and that the *SirIsaac* data are within the error bars of true Gravity. Notice that there are two general types of orbits: bound orbits, with a definitive minimum and maximum radius, and unbound orbits, without a maximum. The bound type includes circular and elliptical orbits and the unbound includes parabolic and hyperbolic. We performed MBAM on three sets of data: all, bound orbits only, and unbound orbits only.

We used the original *SirIsaac* data (the blue line in Fig. 2.7) when fitting the reduced models. *SirIsaac* was originally fit using true gravity with noise added in to replicate the noisiness of data in real data measurements. Since using true gravity implies *a priori* knowledge about Newtonian gravity, the original *SirIsaac* data provided the most natural way to fit the data during MBAM.

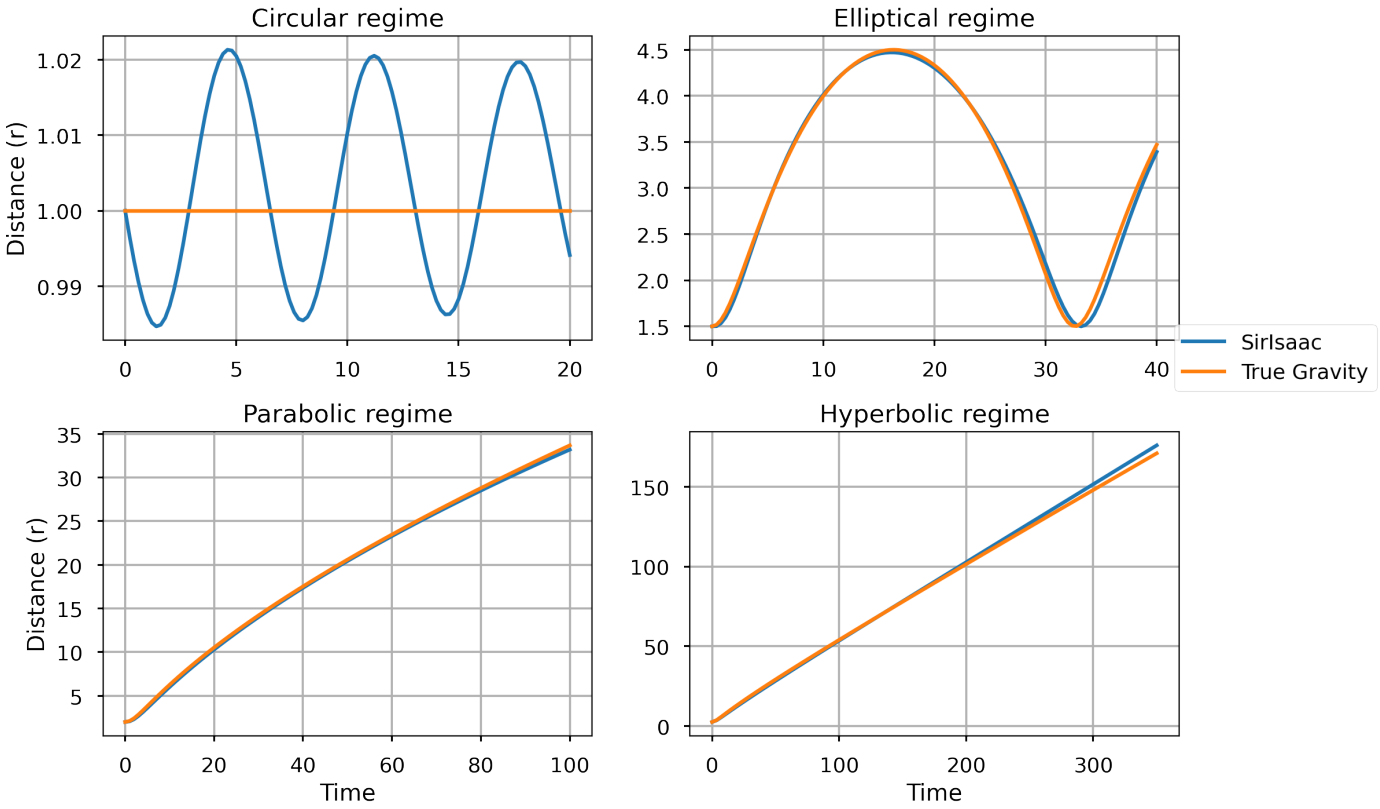

Figure 2.7 Data of original *SirIsaac* and true gravity for all four orbits. The blue line is *SirIsaac* and the orange line is true gravity. Bound orbits (circular and elliptical) are found on top, and unbound orbits (parabolic and hyperbolic) are found on the bottom. The y-axis is the distance, r, and the x-axis is time, both in arbitrary units. Notice that each regime has a different scale in both the horizontal and vertical axes. Although the circular orbit of *SirIsaac* and true gravity appear to be quite different, the scale in the y-axis is so narrow that the circular orbits fall well within the error bars.

Chapter 3

Results and Discussion

Using MBAM, we obtained several different reduced models of *SirIsaac*, which are presented in this section. First, I will show how to reduce from *SirIsaac* to true gravity. I will demonstrate how the reduced models succeed and fail to predict the radius of the orbit. Then, I will compare the true gravity reduction with reduced models obtained using all all orbits, bound orbits, and then unbound orbits. Then I will draw several conclusions and pose further research questions that could be considered.

3.1 True Gravity Reduction

The reduction from *SirIsaac* to true gravity is shown in Fig. 3.1. There are seven reductions required to get to true gravity from *SirIsaac*. The Hasse diagram shows these seven steps, from *SirIsaac* at the top node, to the reduced model, i.e., true gravity, as the last node. The true gravity model contains seven, nonzero parameters. Since the reduction path was done analytically, the order of reductions is not significant. We will compare the reduction paths found by MBAM for all, bound, and unbound orbits to the reduction path of true gravity.

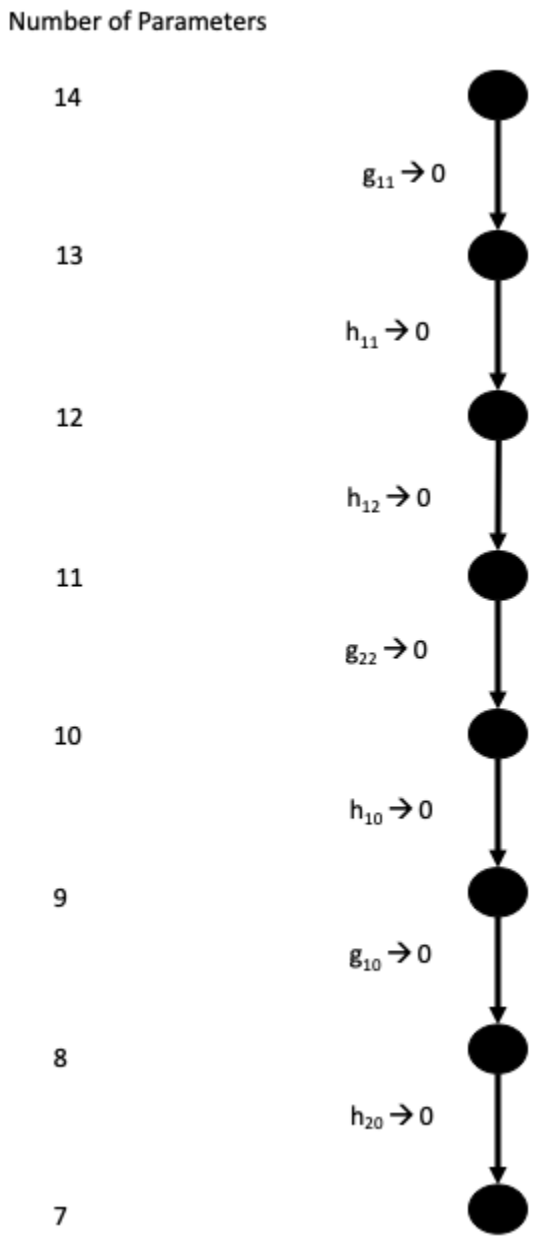

Figure 3.1 Hasse Diagram of true gravity. The full *SirIsaac* model is the top node with fourteen parameters, and the final node is the model for true gravity with seven parameters.

3.2 Fitting Reduced Models

Figure 3.2 shows the fit of the reduced all data model in comparison to *SirIsaac* and true gravity. In general, *SirIsaac* tracks closer to true gravity than the reduced model, but, in the circular regime, the reduced model performs significantly better. One reason for this is that models with more parameters tend to over fit the data used to create the model. *SirIsaac* was created using messy data with twice as many parameters as necessary, therefore causing an over fit problem that is especially obvious in the circular regime. With fewer parameters, the reduced model catches only the most important behavior in the circular regime, i.e., keeping the orbit as close to constant as possible.

The reduced model for both unbound and bound is two 4-parameter models. Fit for the reduced, 4-parameter model for the bound orbits is found in Fig. 3.3. In the elliptical orbit, much the same as the all data reduced model, the reduced bound model follows both the *SirIsaac* and true gravity data closely. In the circular regime, the radius continues growing throughout the range of time and looks like it will continue growing forever. Within the range of the data though, the predictions from the bound model are still within or close to the error bars of the *SirIsaac* data. Because of the uncertainty in the data over this range, the fit is still optimized for this specific set of data. The unbound model is also four parameters and traces out the data well, as shown in Fig. 3.4. Both unbound orbits demonstrate no significant deviations from either *SirIsaac* or true gravity. The one feature that the unbound model fails to reproduce is the bump in radius when r is less than 10, but even after a long time, the reduced model still closely approximates true gravity.

A phase space plot provides another way to compare the behavior of the reduced models to true gravity. The phase space plots for true gravity and the three reduced models (all, bound, and unbound orbits) are found in Fig. 3.5. Since the models contain a factor of r_0 that determines the shape of the orbit, it is necessary to create four phase space plots for each model to compare behaviors. This phase space plot uses $\ddot{r}$, the acceleration, as the vertical component of a vector and $\dot{(r)}$, the velocity, as the horizontal component of a vector. The x-axis corresponds to the position

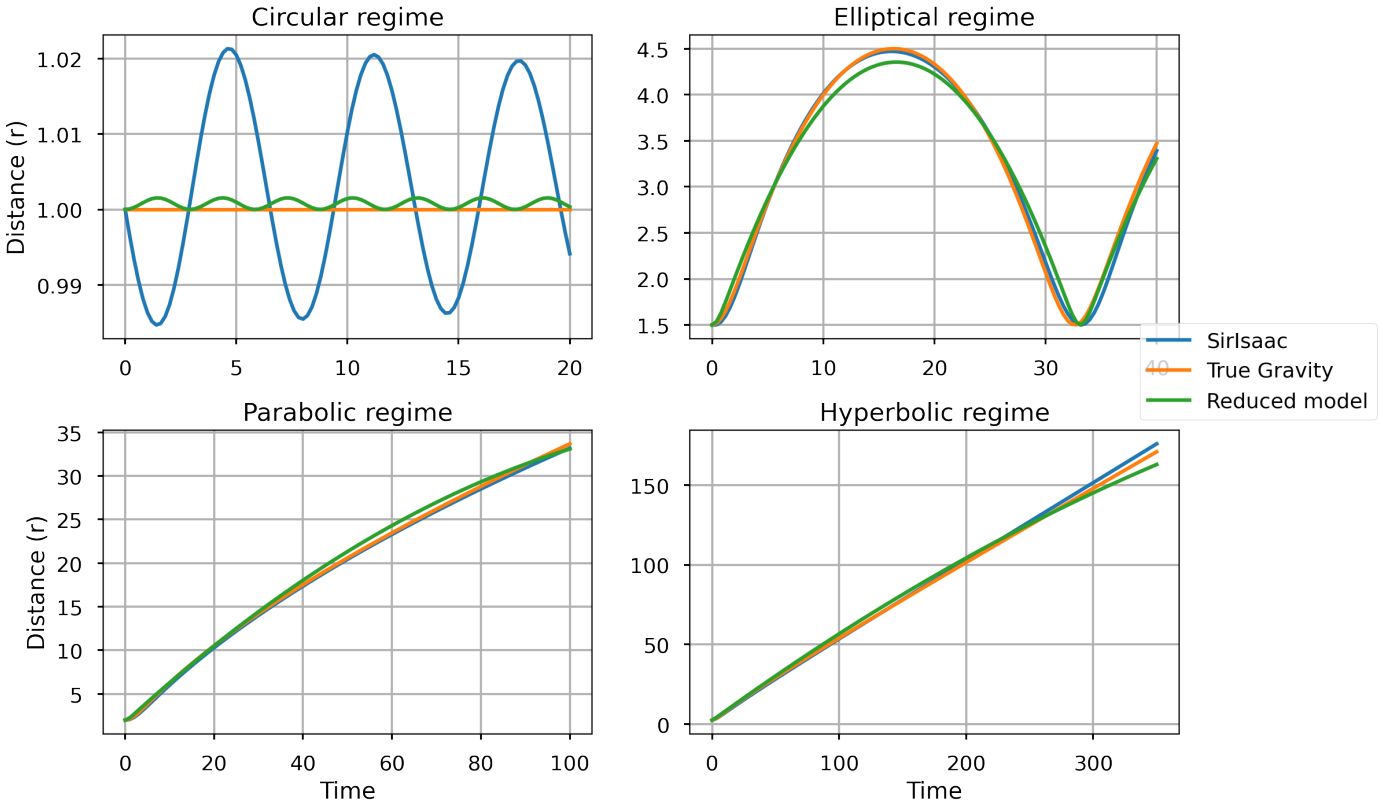

Figure 3.2 Fit of *SirIsaac* (blue), true gravity (orange), and the reduced all data model (green). *SirIsaac* does not match the true gravity perfectly, especially in the circular regime. While *SirIsaac* performs better than the reduced model at replicating the behavior of gravity in the elliptical, parabolic and hyperbolic regimes, the reduced model has significant improvement in the circular regime.

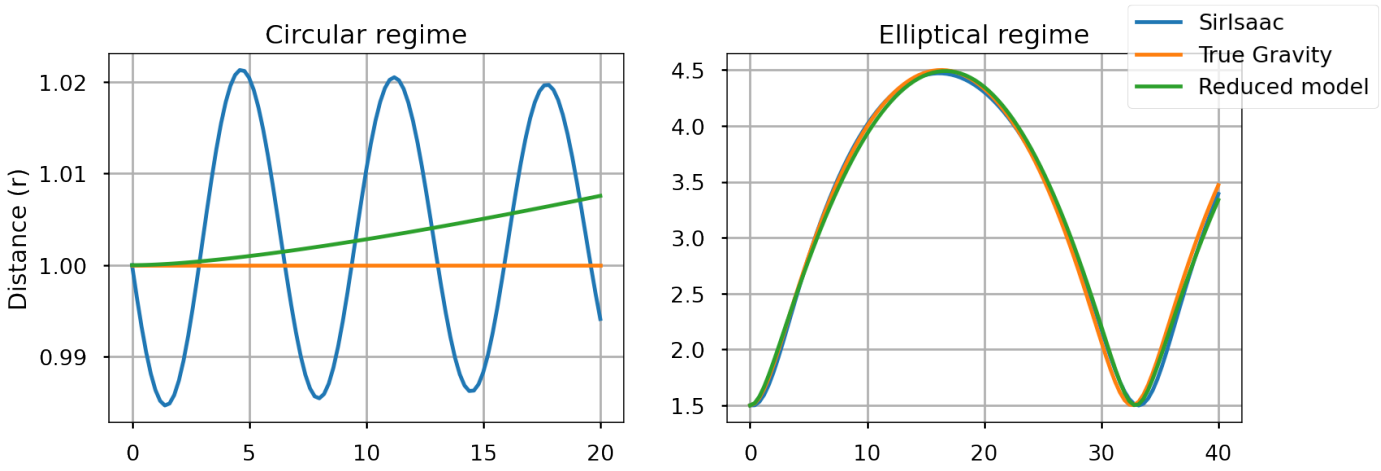

Figure 3.3 Fit of *SirIsaac* (blue), true gravity (orange), and the reduced bound model (green). Similar to Fig. 3.2, the predictions in the elliptical regime follow the same shape and amplitude of both the *SirIsaac* data and true gravity. However, in the circular regime, the reduced model takes on another behavior - it gradually increases throughout the data range without reaching a clear inflection point.

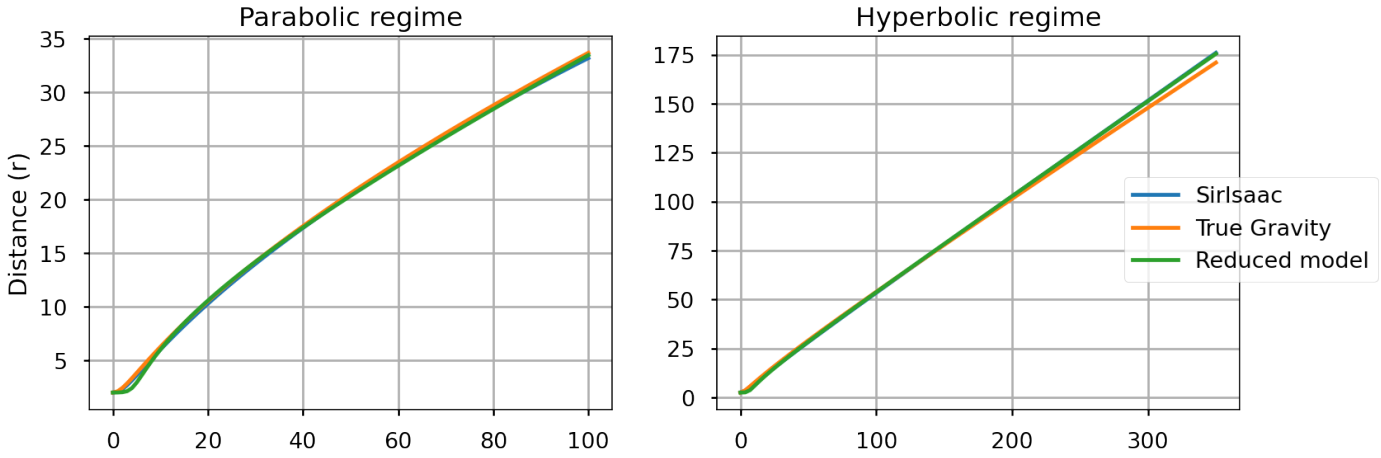

Figure 3.4 Fit of *SirIsaac* (blue), true gravity (orange), and the reduced unbound model (green). The reduced model does an excellent job reproducing both the parabolic and hyperbolic data. The only feature the model fails to reproduce is the slight bump in the data when r is small, i.e., less than 10.

and the y-axis to the velocity.

Comparing the phase space plots of the same orbital regime informs how each model succeeds in reproducing the gravitational data and how the model's behavior differs from other models. Naturally, for the bound model, the unbound phase space plots, since the obits weren't fit on, do not match true gravity. The same goes for the bound orbits in the unbound model. In the circular regime, the all model forms a different shape of orbit than true gravity, yet at there is a point that has a circular orbit, near (1,0). The bound model, instead of producing an orbit, makes the velocity go to 0 when in the circular regime. However, with slight perturbation of the velocity, the radius continues to grow without slowing down. This plot explains the instability seen in 3.3 - the radius will steady and slowly increase forever, because of the instability in the velocity. The elliptical phase space plots, barring the unbound model, all appear to work by a similar mechanism to true gravity. The parabolic orbit is well-reproduced by the all model, yet the unbound model looks entirely different than expected from true gravity. The same is also true of the hyperbolic plots. The unbound model's phase plots shows totally different behavior than true gravity or the all data model, which both have similar phase space plots. Rather, the phase space plots of the unbound model indicate that for positive perturbations of the velocity, the radius will grow quickly. If there is a negative perturbation, the radius will flow toward zero.

3.3 Reduction Path Comparison

Fig. 3.6 presents a comparison of true gravity and MBAM reduction paths. It also introduces the concept of the supremum model, which is the simplest model in common between two or more reduction paths. [15] There are four common reductions between the true gravity and MBAM, meaning that the 10-parameter model is the supremum model for these two reduction paths. The reduction paths branch and true gravity has three more reductions and MBAM identifies five more

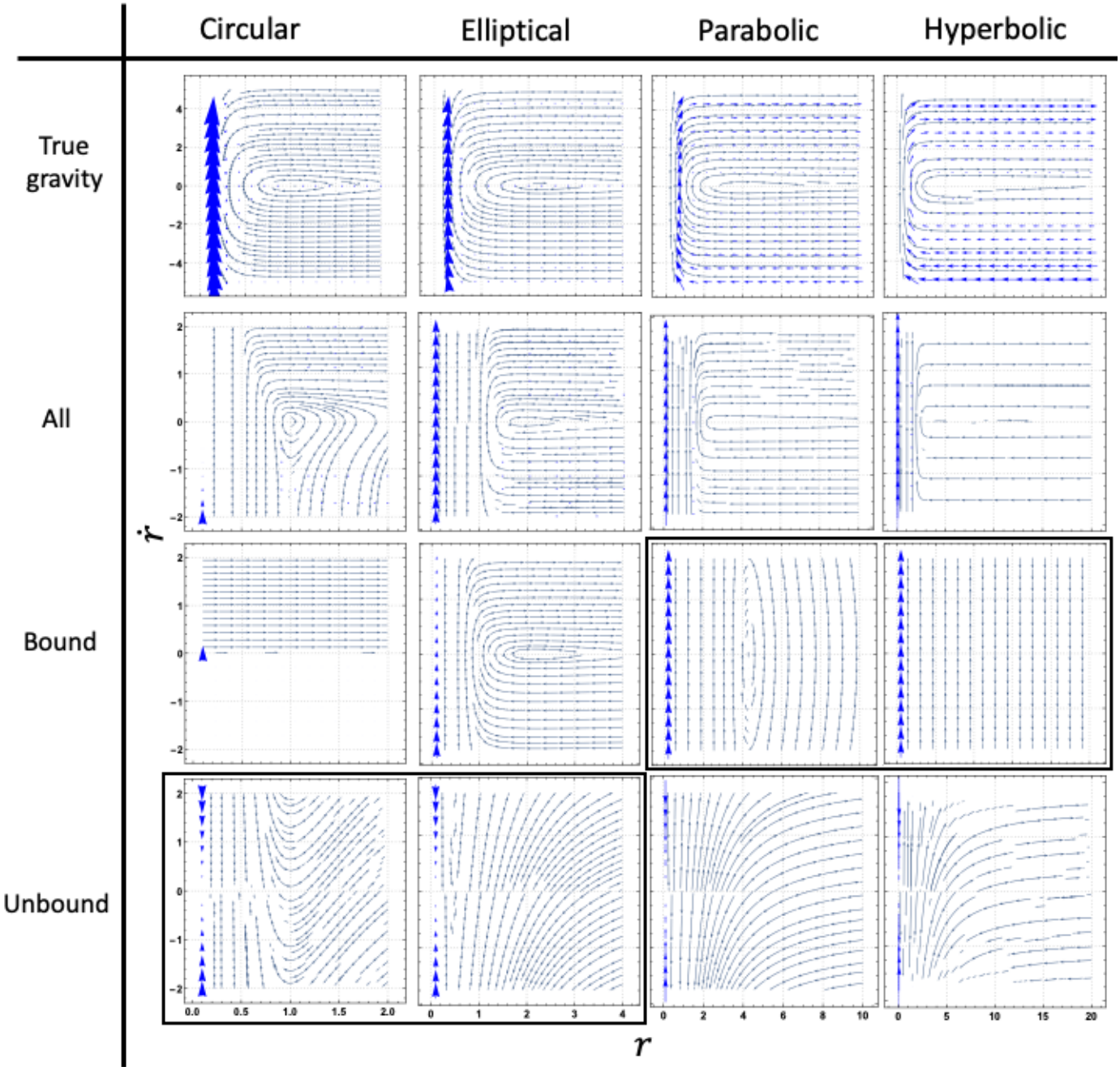

Figure 3.5 Phase space plots of all models for all regimes. Since the model depends on the r_0 or the initial condition of r, each orbit needs its own phase space plots. These phase space plots have r (position) on the x-axis and $\dot{r}$ (velocity) on the y-axis. The horizontal component of a vector is $\dot{r}$ and the vertical component is $\ddot{r}$ (acceleration). The all data reduced model accurately reproduces the behavior in all four regimes, although the circular regime's orbit is a different shape than true gravity. The bound model shows instability in the circular regime, but in the elliptical regime, looks almost identical to the true gravity. The black boxed areas indicate nonsensical areas because the data from those orbits was not used in generating the model. The parabolic and hyperbolic phase space plots for the unbound model also do not match true gravity.

reductions following the split. Although the order is important for MBAM-reduced models, we focus on the path taken rather than each individual step when computing the distance between the two models. For the specific order of limits, see Appendix A. In the seventh reduction, the limit forced a reparameterization so that new parameters appear as combinations of other parameters. The MBAM-reduced model maintains accuracy even with fewer parameters than true gravity. The distance between the all data model and true gravity is eight steps. The final reduced model of MBAM has two fewer parameters than the true gravity reduction and the reduction path of MBAM finds several unique limits that true gravity did not have.

We combine all four reduction paths into one Hasse diagram in Fig 3.7. All four reduction paths have the same four limits that led to the supremum model of true gravity and the all data model. From there, there are three limits required to reach true gravity, shown in the Hasse diagram as a green arrow ($h_{20} \to 0$) and a blue arrow ($h_{10} \to 0$ and $g_{10} \to 0$). Neither of these arrows are found in the reduction path of the all data model, yet the green arrow limit is found in the bound reduction path and the blue arrow is found in the unbound reduction path. This fact indicates that all the necessary limits for true gravity are found in the reduction paths of the unbound and bound models. By clever combination of the two reduction paths, we could potentially recreate true gravity, although we would have to help deliberately assist MBAM in its process. The red arrow in the Hasse diagram ($g_{20} \to 0$) appears in each of the three reduction paths for the MBAM-reduced models, yet it does not appear in true gravity. Lastly, it is fascinating that there is a simpler model than true gravity that accurately reproduces the phenomena of gravity - the all data model only has five parameters versus the seven in true gravity. Despite this further simplicity, aspects of gravitational interaction are lost and the beauty of true gravity encourages it's use.

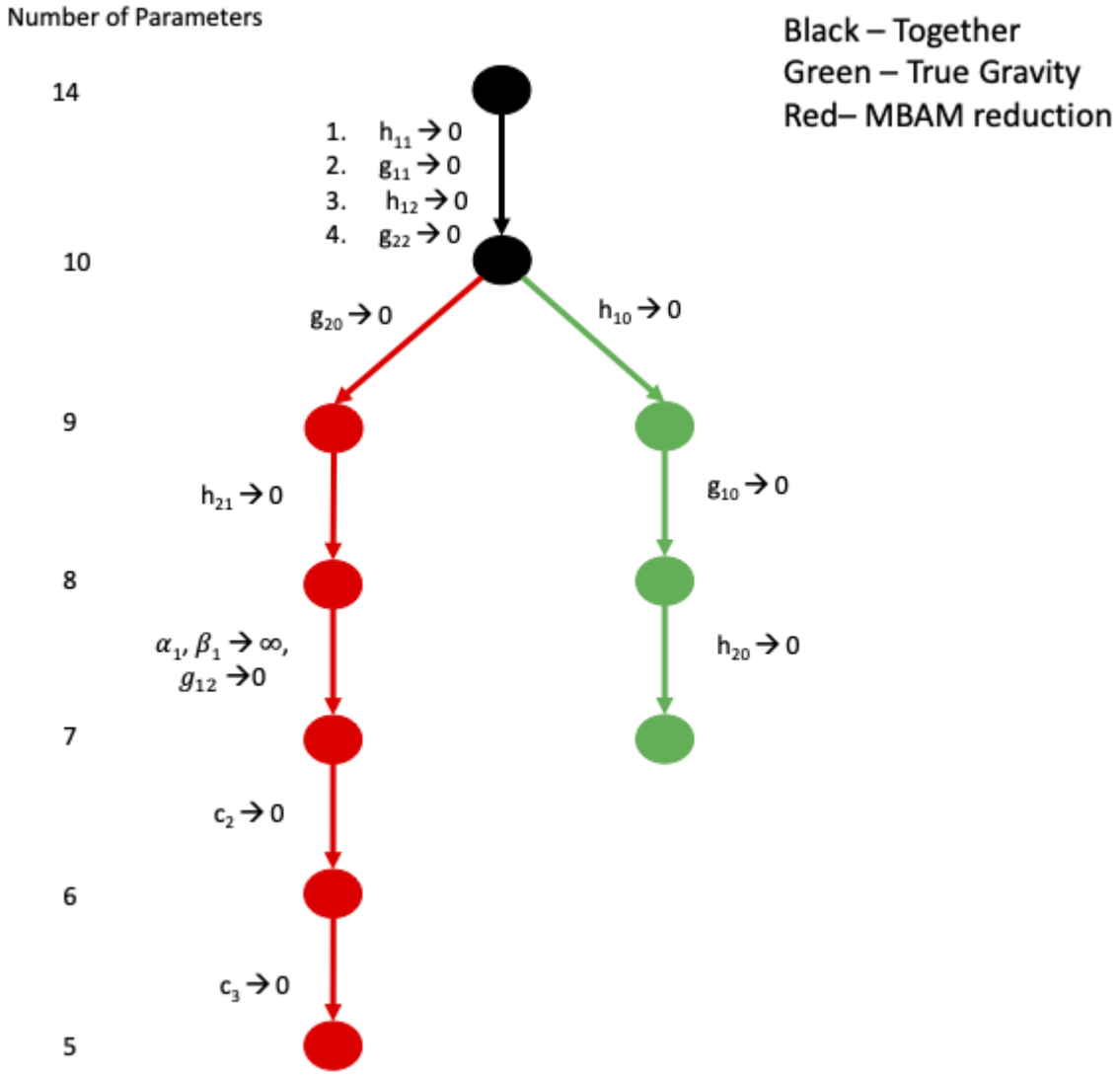

Figure 3.6 Hasse diagram of true gravity and MBAM reduction of *SirIsaac*. The black nodes represent models that are common to both the true gravity and MBAM reduction paths. There are four common reductions through the ten-parameter model. The red represents models that are only found in the MBAM reduction path and green represents models only found in the true gravity reduction path. The 10-parameter model is the supremum model for the all data and true gravity reduction paths. The fully reduced models are eight steps apart.

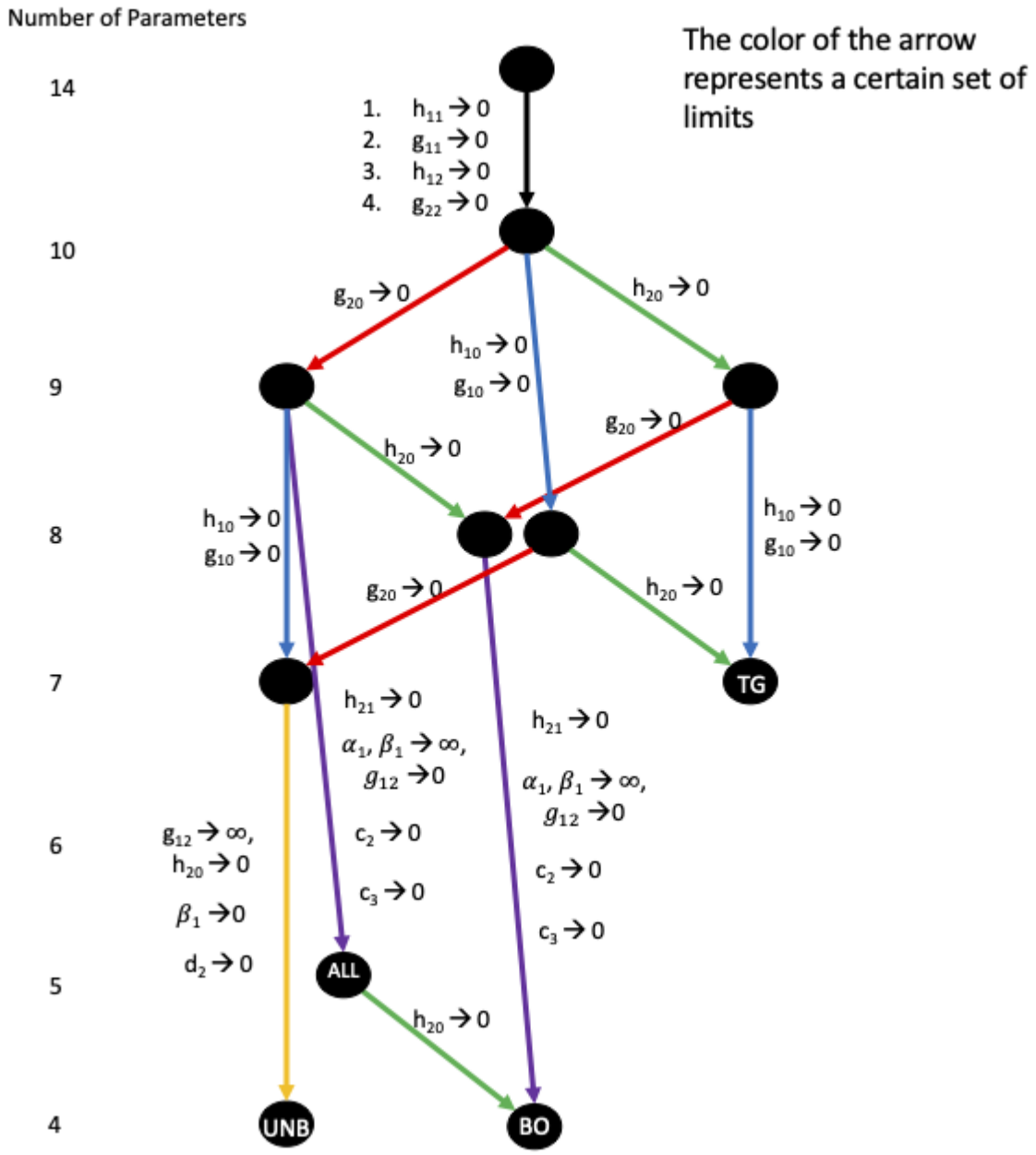

Figure 3.7 Hasse diagram of true gravity, all data model, bound model, and unbound model. Arrows that are the same color and same direction represent the same limit or series of limits. For example, red arrows, generally pointed at 45 degrees down to the left represent the limit $g_{20} \to 0$. The nodes nearest the bottom of the diagram are the fully reduced models. Notice that all the limits necessary to get to true gravity are present in the bound reduction path, the unbound reduction path or both.

3.4 Conclusion

While MBAM was able to bring *SirIsaac* closer to true gravity, MBAM still failed to reproduce Newton's law of gravity. However, there is still methods of splitting data and reparametrizing the model that could be experimented with to try to help machine learning reproduce fundamental laws. For example, the coefficients of *SirIsaac* (α_1,β_1, and β_2) are currently enforced to fall between 0 and infinity, as per the problem formulation. These could be reparametrized to enforce them to fall between 1 and infinity, which means we are adding 1 as a limit to these specific parameters. By doing so, we would slightly change the number of parameters in true gravity, but 1 seems to be a much more natural parameter for a coefficient than 0. Although this adjustment requires some *a priori* knowledge, it could provide additional insight into answering how machine learning can help discover fundamental laws.

SirIsaac also revealed an important fact about the topology of models created with S-systems: their domain is naturally positive, from 0 to infinity. When the parameters were allowed to explore the full parameter space, only two reductions were found using MBAM before they all disappeared. By enforcing 0 as a limit in *SirIsaac*, MBAM identified many reductions that barely affected the accuracy of the predictions. In the future, it is important to remember that S-systems models, and maybe many other machine learning models, should enforce positive domains.

By comparing of the fit plots, phase space plots, and the Hasse diagram, we see that parsimony is not everything. Despite the simplicity of the bound and unbound models and their accuracy in fit, the phase space plots show that the models are not stable. The five parameter all model though manages to be stable while also reproducing the behavior throughout all four orbits. Although simplicity has driven physics discoveries in the past, it is important to understand that simplicity alone is not enough. Data drives machine learning and all of the data is often necessary to make good predictions using machine learning, which would explain why the all model fits the behavior of gravity better than the bound and unbound when considering the phase space portraits. Including

more data adds inherent cost to reducing the number of parameters.

Since this project was motivated in part by the future creation of a computer-based scientist, one natural next step would be to pursue the automation of this system, from using the *SirIsaac* algorithm on a dynamical system, to reducing the model with MBAM. Even though the final result didn't match up perfectly with true gravity, the MBAM-reduced *SirIsaac* did a remarkable job fitting data with a simple model. Future work could also include fitting with more data, specifically including more orbits to fit on. We only used one example from each orbit type - what would happen if we doubled or tripled the number of orbits we fit on? The increased number of orbits, and therefore data points, may increase the number of parameters, but also draw the fully simplified model closer to true gravity. In addition, because all of the limits for true gravity were found between the bound and unbound models, there may be a clever way to split the data and then recombine the models produced that results in the rediscovery of Newtonian gravity. If we iterate over this model several times, can we gradually draw closer to Newtonian gravity [9]? Finally, parsimony, or at least this brand of it, did not provide enough to rediscover Newtonian gravity - we, as humans, have followed Occam's Razor in science, but it appears that machine learning needs more. What other principles have we used to conduct the scientific method that could be applied to machine learning? Dirac [16] has suggested that a principle of beauty is another guiding principle in physics. Can we find a way to quantify mathematical beauty in machine learning to take advantage of this principle?

Appendix A

Reduction Order

The order of reductions found by MBAM for each of the models is found in Table A.1. The order is significant because the first reductions likely affect the predictions the least, and the later predictions affect the predictions the most. The order of true gravity reduction is not significant because that is an analytical, rather than an MBAM, reduction path.

The equations for each of the fully simplified models can be found below. True gravity is the same as Eq. (1.3), with 7 parameters:

$$\begin{aligned} \frac{dr}{dt} &= \chi - 1 \\ \frac{d\chi}{dt} &= r_0^2 r^{-3} - r^{-2}. \end{aligned} \tag{A.1}$$

The 5 parameter all data model:

$$\begin{aligned} \frac{dr}{dt} &= r_{init}^{h_{10}} c_1 \log(X_2) \\ \frac{dX_2}{dt} &= r^{-g_{21}} - \beta_2 r_{init}^{-h_{20}}. \end{aligned} \tag{A.2}$$

The 4-parameter bound model:

$$\begin{aligned} \frac{dr}{dt} &= r_{init}^{h_{10}} c_1 \log(X_2) \\ \frac{dX_2}{dt} &= r^{-g_{21}} - \beta_2. \end{aligned} \tag{A.3}$$

Num. Parameters	True gravity	All data	Bound	Unbound
13	$g_{10} \to 0$	$h_{11} \to 0$	$g_{11} \to 0$	$g_{22} \to 0$
12	$g_{11} \to 0$	$g_{22} \to 0$	$h_{11} \to 0$	$g_{20} \to 0$
11	$h_{10} \to 0$	$g_{20} \to 0$	$g_{20} \to 0$	$h_{11} \to 0$
10	$h_{11} \to 0$	$h_{12} \to 0$	$h_{21} \to 0$	$h_{12} \to 0$
9	$h_{12} \to 0$	$g_{11} \to 0$	$h_{20} \to 0$	$h_{10} \to 0$
8	$g_{22} \to 0$	$h_{21} \to 0$	$g_{22} \to 0$	$\beta_1 \to 0$
7	$h_{20} \to 0$	$\alpha_1, \beta_1 \to \infty, g_{12} \to 0$	$h_{12} \to 0$	$g_{10} \to 0$
6		$c_2 \to 0$	$\alpha_1, \beta_1 \to \infty, g_{12} \to 0$	$g_{11} \to 0$
5		$c_3 \to 0$	$c_2 \to 0$	$g_{12} \to \infty, h_{20} \to 0$
4			$c_3 \to 0$	$c_2 \to 0$

Table A.1 Parameter reduction order. Unlike the Hasse diagram, this reduction order is given in the order that it occurred using MBAM. Only the order of true gravity is not significant, because it was taken analytically. The number of parameters column is the number of parameters after the respective limit has been taken

The 4-parameter unbound model:

$$\begin{aligned} \frac{dr}{dt} &= \alpha_1 \exp(c_1 z) \\ \frac{dz}{dt} &= -r^{-g_{21}} c_3 \exp(r) + \log(r_{init}). \end{aligned} \tag{A.4}$$

Bibliography

[1] P. W. Anderson, "More Is Different," Science **177,** 393–396 (1972).

[2] R. B. Laughlin and D. Pines, "The Theory of Everything," Proceedings of the National Academy of Sciences **97,** 28–31 (2000).

[3] J. P. Crutchfield, "The dreams of theory," WIREs Computational Statistics **6,** 75–79 (2014).

[4] L. Deng and X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," IEEE Transactions on Audio, Speech, and Language Processing **21,** 1060–1089 (2013).

[5] J. A. Cruz and D. S. Wishart, "Applications of Machine Learning in Cancer Prediction and Prognosis," Cancer Informatics 2 (2006).

[6] S. Sun, R. Ouyang, B. Zhang, and T.-Y. Zhang, "Data-driven discovery of formulas by symbolic regression," MRS Bulletin **44,** 559–564 (2019).

[7] M. K. Transtrum, B. B. Machta, K. S. Brown, B. C. Daniels, C. R. Myers, and J. P. Sethna, "Perspective: Sloppiness and emergent theories in physics, biology, and beyond," The Journal of Chemical Physics **143,** 010901 (2015).

[8] W. Pan and C. Zhang, "The Definitions of Interpretability and Learning of Interpretable Models,", 2021.

[9] G. E. P. Box, "Science and Statistics," Journal of the American Statistical Association **71,** 791–799 (1976).

[10] B. C. Daniels and I. Nemenman, "Automated adaptive inference of phenomenological dynamical models," Nature Communications 6 (2015).

[11] M. K. Transtrum and P. Qiu, "Model Reduction by Manifold Boundaries," Phys. Rev. Lett. 113 (2014).

[12] Y. Kurniawan, C. L. Petrie, K. J. Williams, M. K. Transtrum, E. B. Tadmor, R. S. Elliott, D. S. Karls, and M. Wen, "Bayesian, Frequentist, and Information Geometry approaches to parametric uncertainty quantification of classical empirical interatomic potentials,", 2021.

[13] M. K. Transtrum and P. Qiu, "Model Reduction by Manifold Boundaries," Phys. Rev. Lett. **113,** 098701 (2014).

[14] M. K. Transtrum and J. P. Sethna, "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization,", 2012.

[15] C. Petrie, C. Anderson, C. Maekawa, T. Maekawa, and M. K. Transtrum, "The supremum principle selects simple, transferable models,", 2021.

[16] P. A. M. Dirac, "XI.—The Relation between Mathematics and Physics," Proceedings of the Royal Society of Edinburgh **59,** 122–129 (1940).

Index