EFFICIENT APPROXIMATION OF THE STATIONARY SOLUTION TO THE CHEMICAL MASTER EQUATION

by

BRANDON REID

ROGER B. SIDJE, COMMITTEE CHAIR
BRENDAN AMES
MIN SUN
DAVID HALPERN
IAN W. KNOWLES

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2019

ABSTRACT

When studying chemical reactions on the cellular level, it is often helpful to model the system using the continuous-time Markov chain (CTMC) that results from the chemical master equation (CME). It is frequently instructive to compute the probability distribution of this CTMC at statistical equilibrium, thereby gaining insight into the stationary, or long-term, behavior of the system. Computing such a distribution directly is problematic when the state space of the system is large. To alleviate this difficulty, it has become popular to constrain the computational burden by using a finite state projection (FSP), which aims only to capture the most likely states of the system, rather than every possible state. We propose efficient methods to further narrow these states to those that remain highly probable in the long run, after the transient behavior of the system has dissipated.

Our strategy is to quickly estimate the local maxima of the stationary distribution using the reaction rate formulation, which is of considerably smaller size than the full-blown chemical master equation, and from there develop adaptive schemes to profile the distribution around the maxima. The primary focus is on constructing an efficient FSP; however, we also examine how some of our initial estimates perform on their own and discuss how they might be applied to tensor-based methods. We include numerical tests that show the efficiency of our approaches.

DEDICATION

In honor of my grandpa, Frank Taft McCoy, Jr.

ACKNOWLEDGMENTS

A big thank you to my advisor, Dr. Roger B. Sidje for all the hard work, encouragement, coaching, advice, and late nights editing manuscripts. You have taught me a lot over the past five years - not only how to research well, but also how to navigate the academic world. I have greatly enjoyed working with you and with the other members of our research team. Thank you to Dr. Vo Liem, who taught four of my undergraduate courses and played a pivotal role in encouraging me to return to The University of Alabama for graduate school. Thank you to my parents, Steve and Lisa Reid, and the other members of my family, for their support, love, and prayers over the years. Mom and Dad, I hope my life is characterized by the same faith, diligence, and compassion that you have exemplified. Thank you to all the friends who have been with me and tried to keep me sane throughout my time in grad school, especially the members of the Bible study group that met at the Hendersons' and Luetzes' houses. You have helped me in my walk with Jesus and were always there for me. Our gatherings each week, the retreats, anime nights, road trips, working together at JH Outback, and countless other adventures have brought me a lot of joy.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1

INTRODUCTION

1.1 General Background

Many ongoing research efforts have aimed at solving the chemical master equation (CME) in the context of modeling the dynamics of cellular processes through biochemical reactions. It can prove to be a challenging problem due to the high number of possible chemical configurations present in many systems. A variety of methods have been proposed to aid in this question.

For instance, the number of states under consideration can be reduced using a discretized Fokker-Planck equation [24]. Also, high-dimensional systems can be broken down into multiple lower-dimensional systems and analyzed separately. This is sometimes referred to as stochastic partitioning [22], an example of which is to divide the system into fast and slow time-scales [17]. Another type of partitioning is to break a chemical system down into pieces that can be modeled sufficiently well with continuous approaches like the chemical Langevin equation, or even deterministic ODEs, and pieces that must be modeled with discrete distributions. For species that are present in a large number of molecules, the chemical Langevin equation or even deterministic modeling will often suffice, whereas the need for a discrete probability distribution is more prominent for pieces with species present in few copies [22, p. 1360] [1]. Monte Carlo simulation is a widely-used approach to address species that must be modeled stochastically. The Stochastic Simulation Algorithm (SSA) described by Gillespie [4], [5] is the most widely known Monte Carlo technique, and numerous efforts have been made to improve on its efficiency. In particular, tau-leaping has been employed to reduce the

amount of time spent calculating fast reactions and the model can be simplified even further if fast reactions are modeled deterministically [1].

Beyond being large, the chemical master equation may actually have a countably infinite state space. The finite state projection (FSP) of Munsky and Khammash [19] addresses both cases by truncating the CME to a more tractable size that captures the most important states and discards the rest. The majority of this dissertation focuses on developing techniques to make the FSP more efficient in the context of finding the stationary solution. As will be discussed in greater depth, after states are selected for an FSP, it is possible to set up a corresponding transition rate matrix, $\boldsymbol{A}^{FSP}$, and thereby solve the linear system

$$\boldsymbol{A}^{FSP}\hat{\boldsymbol{p}} = \frac{d}{dt}\hat{\boldsymbol{p}} = \boldsymbol{0}$$

to estimate the stationary probability distribution for the states in the FSP. By selecting the states that are most relevant in the long run and eliminating less probable states, we can reduce the size of the state space and thereby increase computational efficiency of solving the linear system.

However, even after developing an efficient FSP, sometimes the linear system is still too large to tackle efficiently with traditional methods. This is frequently the case in high-dimensional models, because the computational burden can increase exponentially with the number of chemical species modeled in the system [12, 24]. Representing such high-dimensional systems in low-rank tensor form, by leveraging techniques such as the quantized tensor train, has proven to be an effective approach to combat the computational strain of large systems. Numerous tensor-based techniques have been developed [2, 7, 9, 12, 14, 15, 20, 21, 26], and tensor-based methods for approximating the stationary solution have had recent success when applied to massive systems [8, 13]. While not the primary focus of this dissertation, we will also discuss how a couple of the methods we present to compute an FSP more efficiently might be applied to tensor-based algorithms.

We note that while many of the methods developed to study the time-dependent, or transient, solution, the focus of this dissertation is studying the stationary solution. The following are two primary ways to formulate the stationary solution $\boldsymbol{p}$ for the CME:

$$\dot{\boldsymbol{p}} = \boldsymbol{A}\boldsymbol{p} = \boldsymbol{0} \tag{1.1}$$

and

$$p(\boldsymbol{x}) = \lim_{t \to \infty} P(\boldsymbol{x}, t) \tag{1.2}$$

Both equations represent the stationary solution but do so through two different perspectives. Equation 1.1 makes use of the transition rate matrix (which will be described more fully in the next chapter) to find the derivative of the probability vector with respect to time ($\dot{\boldsymbol{p}}$). If that derivative is $\boldsymbol{0}$ then the system is in stochastic equilibrium at $\boldsymbol{p}$, so $\boldsymbol{p}$ is a stationary probability vector. Equation 1.2 describes the stationary probability distribution (which is a function of $\boldsymbol{x}$, the state vector representing the chemical composition) as the limiting case of the transient solution as t becomes large. Studying the stationary solution, therefore, gives insight into the long-term behavior of the system.

1.2 Reaction Rate Equations (RREs)

We start by describing how RREs are central to our approach, by allowing us to first quickly estimate critical points of the probability distribution function, through a much smaller problem than the full-blown chemical master equation. It is from there that we will develop adaptive schemes to profile the distribution around the estimated maxima in subsequent sections.

In large chemical systems, deterministic models of chemical reactions are typically used. Using a similar notation to Gillespie [6, Eq. (14)], this approach has the classical reaction rate equation:

$$\frac{d\boldsymbol{z}}{dt} = \sum_{k=1}^{M} \boldsymbol{\nu}_k \tilde{\alpha}_k(\boldsymbol{z}(t)) \tag{1.3}$$

where $\boldsymbol{z}$ is a vector with each component $z_i, 1 \leq i \leq N$ representing the concentration of the ith chemical species out of N total species, $\tilde{\alpha}_k$ is the rate at which the kth reaction is occurring, and $\boldsymbol{\nu}_k$ is the stoichiometric change vector for the kth reaction, with its ith component ν_{ik} representing the change to the copy number of the ith species. Moreover, $\boldsymbol{z}$ is directly proportional to the vector $\boldsymbol{x}$, which represents the copy number of each species present, through a relation of the form $\boldsymbol{z} = \frac{\boldsymbol{x}}{\Omega}$, where Ω is a constant. As described by Gillespie [6, Eq. (11)], we take

$$\frac{d\boldsymbol{x}}{dt} = \sum_{k=1}^{M} \boldsymbol{\nu}_k \alpha_k(\boldsymbol{x}) \tag{1.4}$$

which is similar to Equation 1.3, except that it is expressed in terms of species count rather than concentrations. We can break Equation 1.4 into a system of linear ODEs as

$$\frac{dx_i}{dt} = \sum_{k} \alpha_k(x_1, x_2, ..., x_N)\nu_{ik} \quad \forall\, 1 \leq i \leq N \tag{1.5}$$

To find an equilibrium point of this system we simply find a point that satisfies

$$\sum_{k} \alpha_k(x_1, x_2, ..., x_N)\nu_{ik} = 0 \quad \forall\, 1 \leq i \leq N \tag{1.6}$$

When working with small systems, we must recognize that the interpretation of the propensity functions α_k differs from that of their counterparts $\tilde{\alpha}_k$ in systems large enough to be modeled as continuous. In large systems $\tilde{\alpha}_k$ simply describes the deterministic rate at which reaction k is occurring; in systems small enough that we are concerned with tracking individual species counts rather than concentrations, chemical reactions are generally viewed as stochastic events and we therefore do not know exactly how many reactions will occur in a given time interval. Therefore, $\alpha_k(\boldsymbol{x})$ is not a deterministic rate

Proposed Estimation (System of 2 nonlinear equations)	Linear System $\boldsymbol{Ap} = \mathbf{0}$ (System of 360,000 linear equations)
(27.2886, 377.6974)	(27, 377)
(414.9431, 23.3947)	(414, 23)
(126.0291, 140.4839)	N/A (Saddle Point)

Table 1.1: Comparison of actual and estimated Modes for Gene Toggle example.

at which the reaction is occurring; rather it is the average rate at which the reaction occurs. More specifically, the waiting time for reaction k is modeled as an exponential random variable with mean $\frac{1}{\alpha_k(\boldsymbol{x})}$. Although for small systems we can no longer model $\frac{d\boldsymbol{x}}{dt}$ deterministically, Equation 1.4 allows us to estimate equilibrium points. Indeed, we can solve for

$$\sum_k \alpha_k(x_1, x_2, ..., x_N)\nu_{ik} = 0 \quad \forall\, 1 \leq i \leq N \tag{1.7}$$

as a starting point when searching for states where this probability distribution is at or near a relative maximum. We also bring attention to the fact that in reaction systems where conservation-of-matter relationships exist, we must include these alongside Equation 1.7 when solving for possible equilibrium points. Using the system of equations derived from the reaction rate and possible conservation-of-matter equations proves to be effective and more computationally efficient in some experimental examples than using traditional approaches, such as Equation 1.1. To give a preview of one of the successes of this strategy that will be covered in greater depth in the following chapter, we consider the stationary solution in a Gene Toggle system approximated by Vo using Monte Carlo simulation [23]. Figure 1.1 includes data from Vo's original simulation overlaid with the equilibrium points estimated using the reaction rate equations.

As Table 1.1 shows, our proposed equilibrium point approximation using Equation 1.7 yields a nearly identical set of relative maxima as solving the linear system (in this example a system of $360,000$ equations) directly. Note that our equilibrium point approximation can yield critical points other than relative maxima; here it yields not

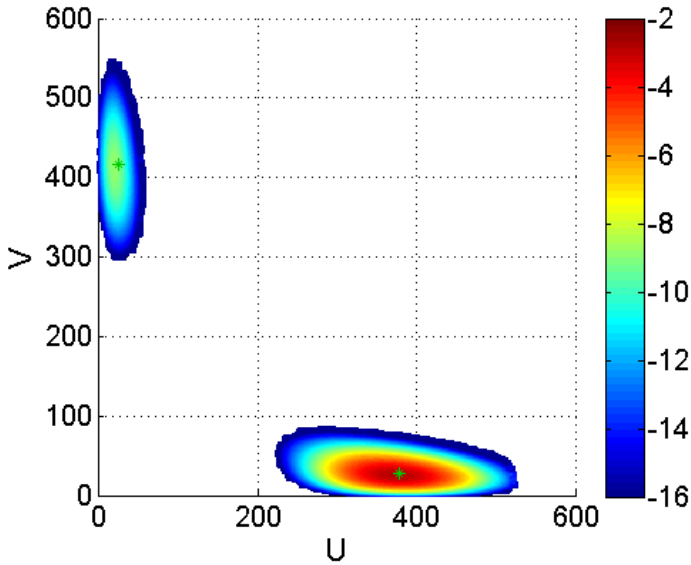

Figure 1.1: SSA approximation of bimodal stationary probability distribution for the Gene Toggle from Vo's experiment with estimated modes superimposed (represented as asterisks). Color represents the log of probability, that is $\log_{10} p(\boldsymbol{x})$, according to the values on the colorbar, with $\log_{10} p(\boldsymbol{x}) \geq -2$ dark red and $\log_{10} p(\boldsymbol{x}) < -16$ left blank.

only each mode of the system but additionally a saddle point.

Approaching the problem from a different angle, we also observe that the system

$$\frac{dx_i}{dt} = \sum_k \alpha_k(x_1, x_2, ..., x_N)\nu_{ik} \quad \forall\, 1 \leq i \leq N \tag{1.8}$$

approaches the same equilibrium points in the long run, as demonstrated in Figure 1.2.

Figure 1.2 illustrates an ODE solver in Matlab, tracing the paths of a system, one with an initial state of $([U],[V]) = (30, 0)$ and another with an initial state of $([U],[V]) = (0, 30)$. The calculated modes of $(414.9431, 23.3947)$ and $(27.2886, 377.6974)$ are superimposed on the graph, lining up with the endpoints of the paths traced by the ODE solver. It can also be seen that at the saddle point, there are vectors pointing in opposite directions. (Note that in this visual, instead of displaying a true vector field for the ODE, magnitude is dropped and only direction is shown; this was done because the magnitude of the vectors at points far from equilibrium is so much larger than at points

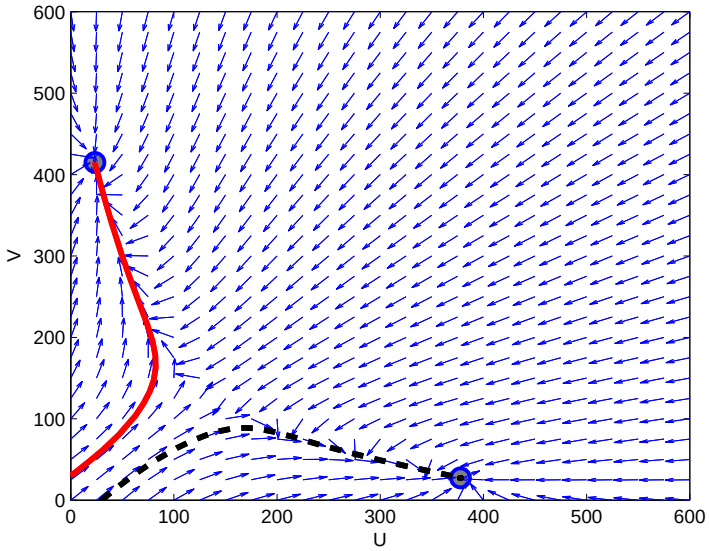

Figure 1.2: Estimated Gene Toggle relative maxima match Matlab ODE solver

closer to equilibrium that it is difficult to represent the magnitude of each to scale on the same graph.)

Specific details regarding this Gene Toggle system and setting up the reaction rate equations to study equilibrium points will be given in the next chapter. Furthermore, because there can be positive stationary probability in other states as well, we will address methods of finding the states surrounding the equilibrium points that are most likely to persist in the long run. We especially focus on Monte Carlo simulation and detailed balance-based techniques. We note that the propensity functions give us another way to formulate the stationary solution, using notation similar to [26]:

$$\frac{dP(\boldsymbol{x},t)}{dt} = \sum_{k=1}^{M} \alpha_k(\boldsymbol{x}-\boldsymbol{\nu}_k)P(\boldsymbol{x}-\boldsymbol{\nu}_k,t) - \alpha_k(\boldsymbol{x})P(\boldsymbol{x},t) = \mathbf{0} \tag{1.9}$$

and that detailed balance differs from this formulation in that, while Equation 1.9 requires that the total probability flowing into a given state must be equal to the probability flowing out of that state, detailed balance requires the flow between two states to be equal in both directions. If detailed balance were required to hold for all states in the system,

it would therefore be a stronger condition than Equation 1.9. However, detailed balance does not always hold for all states in the system. Nevertheless, as we will see, it can be a powerful approximation tool for determining the relevant state space for an FSP.

2

DEVELOPING EFFICIENT FSP STATE SELECTION TECHNIQUES FOR THE STATIONARY SOLUTION

2.1 Application Example: Michaelis-Menten

While more common in large-scale continuous models, it is possible for a CME to end up in a single equilbrium state with probability one in the equilibrium case. We consider the Michaelis-Menten system [11, 18] described by the reactions in Figure 2.1.

	Reaction	Propensity
1	$S + E \rightarrow ES$	$k_1 \times [S] \times [E]$
2	$S + E \leftarrow ES$	$k_2 \times [ES]$
3	$ES \rightarrow P + E$	$k_3 \times [ES]$

Figure 2.1: Michaelis-Menten reactions and their propensity functions with $k_1 = 0.1, k_2 = 1, k_3 = 1$

with initial state vector $([S], [E], [ES], [P]) = (10, 4, 0, 0)$.

Now Equation 1.7 along with two conservation-of-matter equations inherent to the set of reactions and initial state yield the following system of equations:

$$
\begin{cases}
-k_1 \times [S] \times [E] + k_2 \times [ES] = 0 \\
-k_1 \times [S] \times [E] + k_2 \times [ES] + k_3 \times [ES] = 0 \\
k_1 \times [S] \times [E] - k_2 \times [ES] - k_3 \times [ES] = 0 \\
k_3 \times [ES] = 0 \\
([E] + [ES]) = 4 \\
([S] + [ES] + [P]) = 10
\end{cases}
$$

The final two conservation-of-matter equations come from the fact that, based on the possible reactions, any increase of one unit of E will be met by a decrease of one unit of ES and vice versa; likewise, any one unit increase of S, ES, or P will require a one unit decrease of S, ES, or P.

Note that while the first two reactions are reversible, the third reaction is irreversible. The system as a whole is irreversible, because P is formed by the third reaction, but there is no mechanism present through which P is destroyed. In terms of the underlying chemical reactions, the substrate S is binding to the enzyme E. After the substrate and enzyme become bound in the ES state, the enzyme can either release the substrate, and they go back into the unbound E and S states, or the substrate can be converted into the product P and subsequently be released by the enzyme, which returns to its unbound state. Eventually, the entire quantity of substrate will be converted into product. Therefore, only one state remains in the stationary solution, namely $([S], [E], [ES], [P]) = (0, 4, 0, 10)$. Due to the irreversibility of the system, in the stationary case this state has probability 1; all other states are transient and have 0 stationary probability. A more general observation can be made that applies to other systems: only reversible reactions will remain active when a closed system is at statistical equilibrium.

We provide the Jacobian matrix associated with the reaction rate-based equations (the first four equations in this system) and its eigenvalues. The two 0 eigenvalues arise from the two layers of redundancy coming from the two conservation-of-matter equations

and the negative eigenvalues indicate stability for the equilibrium point $(0, 4, 0, 10)$.

$$J = \begin{bmatrix} -k_1[E] & -k_1[S] & k_2 & 0 \\ -k_1[E] & -k_1[S] & k_2 + k_3 & 0 \\ k_1[E] & k_1[S] & -(k_2 + k_3) & 0 \\ 0 & 0 & k_3 & 0 \end{bmatrix}$$

Equilibrium Point	Jacobian Eigenvalues	Identification
(0,4,0,10)	-2.22 -0.180 0 0	Maximum (p=1)

Table 2.1: Equilibrium point analysis for Michaelis-Menten

2.2 Application Example: Dimerization

We seek to examine more interesting stationary probability distributions that have more than a single state with positive stationary probability. One such example is the Dimerization system given below:

	Reaction	Propensity
1	$M + M \to D$	$k_1[M]([M] - 1)$
2	$D \to M + M$	$k_2[D]$

Figure 2.2: Dimerization reactions and their propensity functions with $k_1 = 0.1, k_2 = 2$

Here, two monomers can be fused into a single dimer. The reaction is reversible;

the dimer can likewise be split apart into two monomers. Note that for this system, there is a conservation-of-matter relationship present:

$$M + 2D = \textbf{constant}$$

If we take the initial condition to be $([M], [D]) = (150, 0)$, an equilibrium point in the stationary will satisfy the following system of equations:

$$\begin{cases} -2k_1[M]([M]-1) + 2k_2[D] = 0 \\ k_1[M]([M]-1) - k_2[D] = 0 \\ ([M] + 2[D]) = 150 \end{cases}$$

where the first two equations come from the reaction rate equations and the third is the conservation-of-matter equation. The first two equations yield the following Jacobian:

$$J = \begin{bmatrix} -2k_1(2[M]-1) & 2k_2 \\ k_1(2[M]-1) & -k_2 \end{bmatrix}$$

The point satisfying the system is $([M], [D]) = (34.4904, 57.7548)$. However, this value is clearly not a valid state, given that D must be a whole number and M must be an even whole number. Therefore, we round to the nearest allowable state, $([M], [D]) = (34, 58)$. This is indeed the equilibrium point, as can be observed both by computing the eigenvalues of the Jacobian at that point and by directly solving for the stationary probability distribution, as shown in Table 2.2,

Estimated Equilibrium Point	Jacobian Eigenvalues	Identification	Calculated Maximum (Linear Solve)
(34,58)	-15.4 0	Maximum	(34,58)

Table 2.2: Equilibrium point estimation for Dimerization

where the second eigenvalue of 0 comes from the linear dependence of the system, which results from the conservation-of-matter equation. However, this time, unlike the Michaelis-Menten example, other states surrounding the equilibrium point have positive probability in the stationary case. The probability of each state in this system can be explicitly determined using detailed balance, as shown below. We recall the rate at which the probability distribution is changing is

$$\frac{dP(\boldsymbol{x},t)}{dt} = \sum_{k=1}^{M} \alpha_k(\boldsymbol{x}-\boldsymbol{\nu}_k)P(\boldsymbol{x}-\boldsymbol{\nu}_k,t) - \alpha_k(\boldsymbol{x})P(\boldsymbol{x},t)$$

as given in the introduction 1.9. The stationary solution, by definition, must satisfy $\frac{dP(\boldsymbol{x},t)}{dt} = 0$. Replacing the time dependent probability distribution $P(\boldsymbol{x},t)$ with our stationary distribution $p(\boldsymbol{x})$ and rearranging,

$$\sum_{k=1}^{M} \alpha_k(\boldsymbol{x}-\boldsymbol{\nu}_k)p(\boldsymbol{x}-\boldsymbol{\nu}) = \sum_{k=1}^{M} \alpha_k(\boldsymbol{x})p(\boldsymbol{x}) \tag{2.1}$$

Moreover, we know in this system that $M = 2$ (M is the number or possible reactions, not to be confused with $[M]$, the monomer count in Dimerization) and that

$$\alpha_1(\boldsymbol{x}) = \alpha_1([M],[D]) = k_1[M]([M]-1)$$

$$\alpha_2(\boldsymbol{x}) = \alpha_2([M],[D]) = k_2[D]$$

Using these equations and the fact that the only two states that could be reachable from any specific $([M]), [D])$ are $([M-2], [D+1])$ and $([M+2], [D-1])$, using 1.9 we can specify

$$k_1([M+2])([M+1])p([M+2], [D-1]) + k_2[D+1]p([M-2], [D+1])$$

$$= k_1([M])([M-1])p([M], [D]) + k_2[D]p([M], [D])$$

for any particular $([M]), [D])$. For ease of notation, we define

$$p_D \equiv p([M], [D]) = p(150 - 2[D], [D])$$

Furthermore, we can even reduce the system to a single variable using the conservation relationship $[M] + 2[D] = 150$:

$$k_1(152-2[D])(151-2[D])\boldsymbol{p}_{D-1}+k_2[D+1]\boldsymbol{p}_{D+1} = k_1(150-2[D])([149-2[D])\boldsymbol{p}_D+k_2[D]\boldsymbol{p}_D$$

for any specific state $([M], [D])$. Corresponding to our definition for p_D, we let

$$p_D \equiv p(150 - 2[D], [D])$$

Then, using these relationships, it is possible to express any $p_D, [D] > 0$ in terms of p_{D-1}; the two are directly proportional. Starting with $(150, 0)$ and $(148, 1)$ (following the notation of p_D, we refer to these as $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$), letting $([M], [D]) = (150, 0)$ we find

$$k_1(152)(151)p_{-1} + k_2(1)p_1 = k_1(150)(149)p_0 + k_2(0)p_0$$

There is, of course, however, no reaction allowing a state with a negative quantity of

dimers to occur, so $p_{-1} = 0$ and we can reduce the equation to

$$k_2(1)p_1 = k_1(150)(149)p_0$$

This could also be expressed as

$$p_1\alpha_2(\boldsymbol{x}_1) = p_0\alpha_1(\boldsymbol{x}_0)$$

Because $\alpha_1(\boldsymbol{x}_0)$ is the rate at which the system flows from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$, given that the system is currently in state $\boldsymbol{x}_0$, and $\alpha_2(\boldsymbol{x}_1)$ is the rate at which the system flows from $\boldsymbol{x}_1$ to $\boldsymbol{x}_0$, given that the system is currently in state $\boldsymbol{x}_1$, the system exhibits detailed balance between these two states, meaning that the rate at which probability mass is flowing from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$ is the same rate at which probability mass is flowing from $\boldsymbol{x}_1$ to $\boldsymbol{x}_0$, which is equivalent to stating that the net probability flux between the two states is zero [25]. Moreover, we find that

$$p_1 = \frac{k_1(150)(149)}{k_2(1)}p_0$$

In other words,

$$p_1 = c_1 p_0$$

for $c_1 = \frac{k_1(150)(149)}{k_2(1)}$ We have shown, therefore, that p_1 is directly proportional to p_0 and that detailed balance is exhibited between states $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$. By induction, we will show that all probabilities in the Dimerization example are proportional to p_0 and can be solved for explicitly and that detailed balance holds between all adjacent states. Suppose that the premises hold for all states up to state $\boldsymbol{x}_i$; that is, that $p_j = c_j p_0$, $\forall j \leq i$ and some $c_j \in \mathbb{R}$ and that $p_{j-1}\alpha_1(\boldsymbol{x}_{j-1}) = p_j\alpha_2(\boldsymbol{x}_j)\ \forall j \leq i$.

Then, for $\boldsymbol{x}_{i-1}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_{i+1}$,

$$k_1(152-2i)(151-2i)p_{i-1} + k_2(i+1)p_{i+1} = k_1(150-2i)(149-2i)p_i + k_2(i)p_i$$

However, since we know

$$p_{i-1}\alpha_1(\boldsymbol{x}_{i-1}) = p_i\alpha_2(\boldsymbol{x}_i)$$

then

$$k_1(152-2i)(151-2i)p_{i-1} = k_2(i)p_i$$

and, removing those terms, we are left with

$$k_2(i+1)p_{i+1} = k_1(150-2i)(149-2i)p_i$$

In other words,

$$p_i\alpha_1(\boldsymbol{x}_i) = p_{i+1}\alpha_2(\boldsymbol{x}_{i+1})$$

Thus, detailed balance holds between the two states. Furthermore,

$$p_{i+1} = \frac{k_1(150-2i)(149-2i)}{k_2(i+1)}p_i = \frac{k_1(150-2i)(149-2i)}{k_2(i+1)}c_ip_0$$

Letting $c_{i+1} = \frac{k_1(150-2i)(149-2i)}{k_2(i+1)}c_i$, we find that

$$p_{i+1} = c_{i+1}p_0$$

Therefore, by induction, detailed balance must hold between each set of neighboring states and it is possible to calculate the probabilities of each state as a proportion of p_0. Furthermore, since we know the stationary probabilities must add up to one:

$$\sum_{i=0}^{75} p_i = \sum_{i=0}^{75} c_ip_0 = 1$$

with $c_0 = 1$, and it is possible to calculate all each c_i, p_0 and thereby all other probabilities can be solved for explicitly.

It is not necessary, however, to solve the system using detailed balance. Recalling Equation 1.1, the more traditional approach would be to turn $\frac{dP(\boldsymbol{x},t)}{dt} = \mathbf{0}$ into a system of 76 equations represented by

$$\dot{\boldsymbol{p}} = \boldsymbol{A}\boldsymbol{p} = \mathbf{0}$$

where $\boldsymbol{p}$ is a vector with components p_i representing the stationary probability of finding the system in state $\boldsymbol{x}_i$ (where i is the number assigned to a state after all possible states in the system have been organized into an enumerated list). Note that here, like in the derivation of the Dimerization stationary probability distribution using detailed balance, the subscript i is used not to denote one component (i.e., species) of a vector $\boldsymbol{x}$, but rather to denote the ith possible state in which the system can be found out of our list of all possible enumerated states. (We use the boldface $\boldsymbol{x}_i$ to denote a state in an enumerated list of states and x_i to denote the copy number of a specific species.) A, on the other hand, is the transition rate matrix, which measures the rate at which the system transitions from state $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$, given that the system is currently in $\boldsymbol{x}_j$, with entries a_{ij} that are populated according to the following formulas:

$$a_{ij} = \sum_{k|\boldsymbol{x}_j+\boldsymbol{\nu}_k=\boldsymbol{x}_i} \alpha_k(\boldsymbol{x}_j), \quad \forall i \neq j \tag{2.2}$$

$$a_{jj} = -\sum_{i\neq j} a_{ij} \tag{2.3}$$

where each column of $\boldsymbol{A}$ must necessarily sum to 0.

In the Dimerization example, there are 76 possible states in which the system can be found, ranging from $(150, 0)$ to $(0, 75)$. Therefore, A is a 76×76 matrix and $\boldsymbol{p}$ is a 76×1 vector.

Method	Total States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{4}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
Full	7.60e+01	1.08e-14	1.73e-16	1.08e-14	0.00e+00	0.00e+00
DB Est.	7.60e+01	2.88e-14	4.60e-16	2.88e-14	1.45e-16	1.45e-16

Table 2.3: Results for Dimerization. Column descriptions given in Section 2.5.

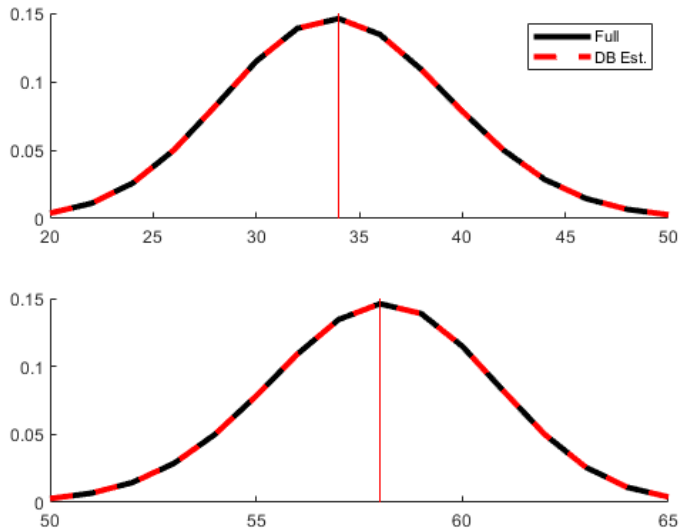

Figure 2.3: Dimerizaton marginal probability functions for species M (upper graph) and D (lower graph).

As can be seen from the Matlab output, as well as the marginal distribution graphs for M and D, the traditional methodology of solving a system of linear equations and the detailed balance approach yield identical results. Note that the traditional linear solve method is listed as the "Full" solve and the detailed balance approach, while it yields an exact solution for this problem, is labeled "Detailed Balance Est." They are labeled as such because they are equivalent to the Full solve and the Detailed Balance Initial Estimate used in the remainder of this dissertation. (For this particular system, the Detailed Balance method is mathematically equivalent to the Detailed Balance Axes Initial Estimate discussed later as well.)

2.3 Classifying Equilibrium Points

Once the critical points or zeros of Equation 1.4 have been identified, it remains to determine whether or not they are stable equilibrium points. The behavior of the system can be analyzed by using the Jacobian of Equation 1.4.

If the Jacobian has only negative eigenvalues, the equilibrium point is stable and the estimated probability there is an estimate of a relative maximum. If all eigenvalues are positive, it refers instead to a relative minimum. A mixture of positive and negative eigenvalues indicates a saddle point. Complex eigenvalues indicate a circulation of probability about the equilibrium point, but if all the real components are negative, the equilibrium is still stable.

2.4 Constructing an FSP Around the Stable Equilibrium Points

If approximating the equilibrium points was our only objective, we could stop here. However, we wish to discover more information about the stationary probability distribution by examining the regions surrounding each of the stable equilibrium points. The probability that the system will be found at one of the estimated points is not necessarily 1, even when the system is at stochastic equilibrium. Rather, other states in the system usually also have positive probability, particularly the states nearby the relative maxima. At the same time, there are often also many states with negligible probability. It would greatly increase computational efficiency to construct a narrower FSP with these low-probability states removed. We will examine several strategies of using the estimated equilibrium points to identify nearby relevant states that can be included in an FSP.

Then, using the collection of states that we find for an FSP, we obtain a system of equations

$$\hat{\boldsymbol{A}}^{FSP}\hat{\boldsymbol{p}} = \boldsymbol{0} \tag{2.4}$$

of smaller size compared to the original system (and therefore, having reduced runtime to solve). To build $\hat{\boldsymbol{A}}^{FSP}$, we first define $\hat{\boldsymbol{A}}$ as a truncation of the matrix $\boldsymbol{A}$, which keeps only the entries that are in both the columns and rows of $\boldsymbol{A}$ that correspond to states in the FSP. Let σ map the column number in $\hat{\boldsymbol{A}}$ of a state in our truncated FSP to the column number of the same state in the original $\boldsymbol{A}$ matrix. We define the entries of $\hat{\boldsymbol{A}}$ as follows:

$$a_{ij} = a_{\sigma(i),\sigma(j)} \quad \forall i, j \tag{2.5}$$

However, by truncating most of the entries in each column, we will now likely have some columns that no longer sum to zero. To create a truncated FSP that preserves each column summing to zero, we modify $\hat{\boldsymbol{A}}$ into $\hat{\boldsymbol{A}}^{FSP}$ so that it satisfies

$$\hat{a}_{jj}^{FSP} = -\sum_{i \neq j} \hat{a}_{ij}^{FSP} \tag{2.6}$$

in order for no probability mass to leak out from our FSP (a necessary condition for our equilibrium probabilities to add up to 1). As mentioned, however, for some columns, we may not have $\hat{a}_{jj} = -\sum_{i \neq j} \hat{a}_{ij}$. This is especially often the case for columns representing states along the outer boundary of the FSP. To remedy this issue, let $\alpha_j^{excess} = -\sum_i \hat{a}_{ij}$, which is always non-negative (because it represents the sum of the propensities of all reactions flowing out of the state j, captured in full by $-a_{jj}$, less the sum of the propensities to flow from state j to other states in the FSP, which leaves the propensity to flow from state j to truncated states outside the FSP), represent the residual propensity to flow out of a given state j. Then, in accordance with the stationary FSP procedure suggested by Gupta et. al. [8] we map this excess flow that would otherwise be leaving the finite space projection and entering the truncated region back to a designated

point within the FSP. According to this procedure, we may designate any point in our FSP to receive this excess flow. In a unimodal system, we choose this point to be the stable equilibrium point. In the case of a system with w stable equilibria, we modify Gupta's procedure slightly and model a flow of propensity $\frac{1}{w}\alpha_j^{excess}$ to each equilibrium point. Letting $\gamma(k)$ represent the index of the kth mode in $\hat{\boldsymbol{A}}$, we then add $\frac{1}{w}\alpha_j^{excess}$ to $\hat{a}^{FSP}_{\gamma(k),j}$ $\forall 1 \leq k \leq w$, which allows the $\hat{a}^{FSP}_{jj} = -\sum_{i \neq j} \hat{a}^{FSP}_{ij}$ relation we require.

After completing this procedure, we can solve $\hat{\boldsymbol{A}}^{FSP}\hat{\boldsymbol{p}} = \boldsymbol{0}$ and use $\hat{\boldsymbol{p}}$ to construct the approximation $\sigma(\hat{\boldsymbol{p}})$ of the original vector $\boldsymbol{p}$ to estimate a solution to the original equation $\boldsymbol{Ap} = \boldsymbol{0}$ for the whole state space as follows:

$$(\sigma(\hat{p}))_i = \begin{cases} \hat{p}_j, & i = \sigma(j) \text{ for some state j in the FSP} \\ 0, & \text{otherwise} \end{cases}$$

2.4.1 R-Step Reachability

One method of identifying which states to keep for an FSP is the r-step reachability method [19], [23]. Starting with a set R_0 consisting only of the stable estimated equilibrium points identified by the procedure discussed in Section 2.3, we keep these points and add all states that can be reached from the estimated equilibria in r or fewer reactions, where r is simply a whole number that can be adjusted to yield results with an error below a specified tolerance. Thus, this is an *adaptive* FSP method, because r can be adjusted to fit the user's needs.

$$R_1 = R_0 \cup \{\boldsymbol{x} | \boldsymbol{x} \text{ can be reached from some } \boldsymbol{x}' \in R_0 \text{ in one reaction}\}$$

$$R_2 = R_1 \cup \{\boldsymbol{x} | \boldsymbol{x} \text{ can be reached from some } \boldsymbol{x}' \in R_1 \text{ in one reaction}\}$$

$$\vdots$$

$$R_r = R_{r-1} \cup \{\boldsymbol{x} | \boldsymbol{x} \text{ can be reached from some } \boldsymbol{x}' \in R_{r-1} \text{ in one reaction}\}$$

2.4.2 Traditional SSA

r-step reachability is a useful first step, but it does nothing to determine which states within a given distance (that is, steps of reachability) are more probable (and therefore better-suited for inclusion in the FSP) and which states are less probable. To address this concern and improve our method for constructing FSPs, one strategy is to implement Monte Carlo simulation. One of the most basic Monte Carlo approaches is to use the Stochastic Simulation Algorithm (SSA) developed and refined by Gillespie [4, 5]. We implement the following version of the SSA to form our FSP.

Algorithm 1 Traditional SSA FSP (abbreviated algorithm)

for counter = 1 to NumberOfTrials

: Start with a stable equilibrium point $\boldsymbol{x}_{eq}$

: $t = 0$

while $t < t_f$

: Choose next reaction: $p(j) = \frac{\alpha_j(\boldsymbol{x})}{\sum_{k=1}^{M} \alpha_k(\boldsymbol{x})}$

: $\tau = \frac{1}{\sum_{k=1}^{M} \alpha_k(\boldsymbol{x})}$

: $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{\nu}_k$

: $t = t + \tau$

end while

: Add final $\boldsymbol{x}$ to the FSP if it isn't yet included.

end for

Our approach here also takes advantage of the equilibrium points found by the procedure outlined in 2.3. Here, we run a high number of simulations, where the starting points for each of these of simulations run is divided equally among the estimated equi-

librium points. Each run is allowed a large amount of simulated time, and the ending point of the simulation is selected for inclusion in the FSP.

2.4.3 R-Step SSA

A modification of r-step reachability based on the SSA can be used to make an educated selection of states to be included in the FSP. Similar to r-step reachability, we start with a set T_0 consisting only of the estimated equilibrium points identified from the analysis of the reaction rate equations. This time, however, instead of taking T_1 to be the union of T_0 and all states a single reaction away from T_0, we consider a set Γ_0, where for each element of T_0, we stochastically sample a single reaction and collect the resulting states into Γ_0. The sampling is done using the Next Reaction Index scheme $\frac{\alpha_j(\boldsymbol{x})}{\sum_i \alpha_i(\boldsymbol{x})}$ [10] for the probability of choosing reaction j as the next reaction given that the system is currently at state $\boldsymbol{x}$.

Here, note that any element of Γ_0 is at most one reaction away from T_0. Now taking $T_1 = T_0 \cup \Gamma_0$, we can iteratively construct sets Γ_k and $T_{k+1} = T_k \cup \Gamma_k$ until we obtain an T_r that meets our error tolerance requirements, thus forming another adaptive procedure that we refer to as r-step SSA throughout the rest of this dissertation.

The algorithm can be summarized as follows:

Algorithm 2 Traditional SSA FSP (abbreviated algorithm)

for r = 0 to NumberOfSteps-1

: $T_{r+1} = T_r$

for i = 1 to size(T_r)

: $\boldsymbol{x} = (T_r)_i$

: Choose next reaction: $p(j) = \frac{\alpha_j(\boldsymbol{x})}{\sum_{k=1}^{M} \alpha_k(\boldsymbol{x})}$

: $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{\nu}_k$

: Add $\boldsymbol{x}$ to the T_{r+1} if it isn't yet included.

end for

end for

This method is very similar to the SSA projection method described by Vo and

Sidje [23]. There will necessarily be some redundancy in some of the states discovered by this procedure, but that is by design. High-probability states near the stable equilibria are virtually guaranteed to be selected for inclusion in the FSP, while less probable states are more likely to be left out. Note that $T_k \subset R_k \ \forall k$; thus, this method helps us to keep the more states from r-step reachability that are more likely to be reached and throw away the ones less likely to be reached. In our implementation here, we let r be the smallest value that allows us to reach a specified minimum number of states to be included in our FSP. Therefore, it is the specified number of states, rather than r, that is our controlling parameter in the experiments presented here.

2.4.4 Detailed Balance

The observations and experimental evidence presented in the Introduction are already strong reasons for seeking to identify possible equilibrium points using the reaction rate equations to estimate equilibrium points for chemical reaction networks. We present another argument here using probability flux.

First, we recall that in reality the components of $\boldsymbol{x}$ can take on only whole number values since they represent species counts. As such, we can define a stationary probability mass function $p(\boldsymbol{x})$ to describe the equilibrium probability of each possible state in our state space. However, for our analysis, it is often useful for our equilibrium point approximations to let $\boldsymbol{x}$ be a continuous in each of its N dimensions. Before we restrict $\boldsymbol{x}$ to only take on whole number values, we define a continuous extension $\rho(\boldsymbol{x})$ of our original stationary probability density function $p(\boldsymbol{x})$ to analyze the behavior of the system.

We seek to identify possible equilibrium points of ρ using the net probability flux, which we denote as Φ. This net probability flux can be expressed as the sum of each of its components:

$$\Phi(\boldsymbol{x}) = \sum_{i=1}^{N} \mathcal{F}_i(\boldsymbol{x})\hat{\boldsymbol{e}}_i \tag{2.7}$$

where each $\mathcal{F}_i(\boldsymbol{x})$ is the amount of the probability flux in the direction of the ith species

at a given state $\boldsymbol{x}$ and $\hat{\boldsymbol{e}}_i$ is the unit vector in the direction of species i, indicating an increase of one unit of that species. Furthermore, we institute a change of coordinates for each of the M reaction channels to enable us to calculate the flux resulting from the reaction at any given $\boldsymbol{x}$. We let

$$\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k} = \frac{\boldsymbol{\nu}_k}{||\boldsymbol{\nu}_k||_2} \tag{2.8}$$

denote the unit vector in the direction of the kth reaction. Furthermore, we define

$$g_{\boldsymbol{x},k}(s) = \alpha_k(\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k})\rho(\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}) \tag{2.9}$$

to denote the probability density flow from state $\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}$ to state $\boldsymbol{x}$ due to reaction k, provided that the distance between the two states is such that state $\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}$ can reach or pass through state $\boldsymbol{x}$ by a single firing of reaction k - i.e., that $s \leq ||\boldsymbol{\nu}_k||_2$ and that $\boldsymbol{x}$ falls on the line $\boldsymbol{x} + s^*\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}$ for some $s^* \in \mathbb{R}$. Thus, the total probability flux at a point $\boldsymbol{x}$ due to reaction k is:

$$\int_0^{||\boldsymbol{\nu}_k||_2} g_{\boldsymbol{x},k}(s)ds \;\; \hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}$$

with a component in the direction of the ith species:

$$\frac{\nu_{ik}}{||\boldsymbol{\nu}_k||_2} \int_0^{||\boldsymbol{\nu}_k||_2} g_{\boldsymbol{x},k}(s)ds \;\; \hat{\boldsymbol{e}}_i$$

Using this, we can calculate the flux in the ith direction:

$$\mathcal{F}_i(\boldsymbol{x}) = \sum_{k=1}^{M} \frac{\nu_{ik}}{||\boldsymbol{\nu}_k||_2} \int_0^{||\boldsymbol{\nu}_k||_2} g_{\boldsymbol{x},k}(s)ds \tag{2.10}$$

However, we can make the following two approximations, which are equivalent to two of the assumptions involved in the Fokker-Planck approximation of master equations, since α_k and ρ both vary slowly with changes to $\boldsymbol{x}$ [3]:

$$\alpha_k(\boldsymbol{x}) \approx \alpha_k(\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}), |s| < ||\boldsymbol{\nu}_k||_2 \tag{2.11}$$

$$\rho(\boldsymbol{x}) \approx \rho(\boldsymbol{x} - s\hat{\boldsymbol{e}}_{\boldsymbol{\nu}_k}), |s| < ||\boldsymbol{\nu}_k||_2 \tag{2.12}$$

The latter assumption is especially appropriate when $\rho(\boldsymbol{x})$ is near an equilibrium point, since its partial derivatives are 0 in all directions. Therefore,

$$g_{\boldsymbol{x},k}(s) \approx g_{\boldsymbol{x},k}(0) = \alpha_k(\boldsymbol{x})\rho(\boldsymbol{x})$$

when $\rho(\boldsymbol{x})$ is a relative maximum or other critical point of the density function. Therefore, near such critical points we can approximate $\mathcal{F}_i$ with F_i as follows:

$$F_i = \sum_{k=1}^{M} \frac{\nu_{ik}}{\|\boldsymbol{\nu}_k\|_2} \int_0^{||\boldsymbol{\nu}_k||_2} g_{\boldsymbol{x},k}(0)ds$$

Thus,

$$F_i = \sum_{k=1}^{M} \frac{\nu_{ik}}{\|\boldsymbol{\nu}_k\|_2} \|\boldsymbol{\nu}_k\|_2 g_{\boldsymbol{x},k}(0)$$

or more simply,

$$F_i = \rho(\boldsymbol{x}) \sum_{k=1}^{M} \nu_{ik}\alpha_k(\boldsymbol{x}) \tag{2.13}$$

Additionally, we can use each of the F_i's to approximate the probability flux near critical points:

$$\Phi(\boldsymbol{x}) \approx \sum_{i=1}^{N} F_i(\boldsymbol{x})\hat{\boldsymbol{e}}_i$$

Furthermore, probability flux will be $\mathbf{0}$ [25] at equilibrium points, so we can use a linear system

$$F_i = \rho(\boldsymbol{x}) \sum_{k=1}^{M} \nu_{ik}\alpha_k(\boldsymbol{x}) = 0 \quad \forall i \tag{2.14}$$

to find possible relative maxima for the system.

We know that probability flux is $\mathbf{0}$ at equilibrium and we have assumed that the probability function and that the propensity functions are changing slowly near a relative maximum. Because of this, $\Phi(\boldsymbol{x}) \approx \mathbf{0}$ near an relative maximum. In terms of the equations from this section, Assumption (2.11) means the α_k's change slowly and Assumption (2.12) means that ρ's changes slowly, so since each $F_i = 0$ at equilibrium by Equation 2.14, we also have

$$F_i = \rho(\boldsymbol{x}) \sum_{k=1}^{M} \nu_{ik} \alpha_k(\boldsymbol{x}) \approx 0 \tag{2.15}$$

near a relative maximum. Since flux is nearly $\mathbf{0}$, detailed balance approximately holds in all directions at points near the relative maximum.

This insight yields another way to construct an FSP around a stable equilibrium point. Once again, we start with only the stable equilibrium point(s) included in the FSP and expand outward from them. We refer to our initial collection of equilibrium states as S_0. This time, we use a detailed balance-based approximation of the probability of the states to determine how to expand our state space. Any states that are reachable in a single step from S_0 are candidate states. A state's probability is estimated relative to that of the equilibrium point it was reached from using detailed balance and if this probability estimate meets a threshold proportion β of the equilibrium point's probability - that is, for a candidate state $\boldsymbol{x}_i$ if

$$p(\boldsymbol{x}_i) \geq \beta p(\boldsymbol{x}_{eq}) \tag{2.16}$$

- then it is included in S_1 as part of the FSP. For the next step, states that are reachable from the states in the most recent shell computed (that is, S_1) become the candidate states. States in S_1 are used as the baseline from which to compute detailed balance estimations. The new candidate states are compared back to the original equilibrium point according to Equation 2.16 and states that meet the threshold are included in S_2.

The process continues until either no new states are added to the FSP in a given iteration or a maximum number of states is reached.

To give greater detail on how the detailed balance probability estimations are performed, we recall Section 2.2, where detailed balance relationships were computed explicitly in the inductive proof that showed that detailed balance must hold completely for the entire Dimerization state space in the stationary solution. To place these calculations in more general terms, suppose we have two states $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, Reaction 1 with propensity $\alpha_1(\boldsymbol{x})$ and stoichiometric change vector $\boldsymbol{\nu}_1 = (\boldsymbol{x}_j - \boldsymbol{x}_i)$, Reaction 2 with propensity $\alpha_2(\boldsymbol{x})$ and stoichiometric change vector $\boldsymbol{\nu}_2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)$, and either no other reactions in the system or else only reactions that are perpendicular to $(\boldsymbol{x}_i - \boldsymbol{x}_j)$. Then a flux of $\mathbf{0}$ means that there must be no net flux in any direction, including the direction $(\boldsymbol{x}_i - \boldsymbol{x}_j)$. Then since Reactions 1 and 2 are the only reactions that have a bearing on the flux in this direction, by Equation 2.14 we must have

$$p(\boldsymbol{x}_i)\alpha_1(\boldsymbol{x}_i)\boldsymbol{\nu}_1 + p(\boldsymbol{x}_j)\alpha_2(\boldsymbol{x}_j)\boldsymbol{\nu}_2 = 0 \tag{2.17}$$

Since $\boldsymbol{\nu}_1 = -\boldsymbol{\nu}_2$, we can simplify this to

$$p(\boldsymbol{x}_i)\alpha_1(\boldsymbol{x}_i) = p(\boldsymbol{x}_j)\alpha_2(\boldsymbol{x}_j) \tag{2.18}$$

Now it is possible to solve for $p(\boldsymbol{x}_j)$ relative to $p(\boldsymbol{x}_i)$:

$$p(\boldsymbol{x}_j) = p(\boldsymbol{x}_i)\frac{\alpha_1(\boldsymbol{x}_i)}{\alpha_2(\boldsymbol{x}_j)} \tag{2.19}$$

In situations where detailed balance is only approximate, we can still use it to estimate the probability of a new state when evaluating possible inclusion in an FSP:

$$p_{\mathsf{est}}(\boldsymbol{x}_j) = p_{\mathsf{est}}(\boldsymbol{x}_i)\frac{\alpha_1(\boldsymbol{x}_i)}{\alpha_2(\boldsymbol{x}_j)} \tag{2.20}$$

For the remainder of this section, we will drop the notation of p_{est} in favor of simply p, but it is understood that the p's being discussed are the values estimated from detailed balance.

Sometimes, two states from the previous shell of an FSP (call them $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$) may reach the same candidate state $\boldsymbol{x}_k$ from perpendicular directions. For instance, in the Gene Toggle example that will be discussed in the next section, $([U],[V]) = (378, 27)$ is an estimated equilibrium point, $(379, 27)$ and $(378, 28)$ are both members of S_1, and $(379, 28)$ is a candidate for S_2 that is reachable from both $(379, 27)$ and $(378, 28)$. In a case like this, we let $p(\boldsymbol{x}_k)$ be estimated from a weighted average of the estimates that would come from $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ come from the previous shell of the FSP and the weighting is assigned by the probability estimates for $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively:

$$p(\boldsymbol{x}_k) = \frac{p(\boldsymbol{x}_i)^2 \frac{\alpha_1(\boldsymbol{x}_i)}{\alpha_2(\boldsymbol{x}_k)} + p(\boldsymbol{x}_j)^2 \frac{\alpha_3(\boldsymbol{x}_j)}{\alpha_4(\boldsymbol{x}_k)}}{p(\boldsymbol{x}_i) + p(\boldsymbol{x}_j)} \tag{2.21}$$

where $\boldsymbol{\nu}_1 = (\boldsymbol{x}_k - \boldsymbol{x}_i)$, $\boldsymbol{\nu}_2 = (\boldsymbol{x}_i - \boldsymbol{x}_k)$, $\boldsymbol{\nu}_3 = (\boldsymbol{x}_k - \boldsymbol{x}_j)$, and $\boldsymbol{\nu}_4 = (\boldsymbol{x}_j - \boldsymbol{x}_k)$ and both α_1 and α_2, along with α_3 and α_4, are the only reactions not perpendicular in their own respective directions as in Equation 2.20.

For a generalized case where a candidate state probability estimation is computed from multiple states from the most recently-produced shell, each state meeting the conditions involved for Equations 2.20 and 2.21, we use

$$p(\boldsymbol{x}_j) = \frac{\sum_{i,k,\ell | \boldsymbol{x}_j - \boldsymbol{x}_i = \boldsymbol{\nu}_k, \boldsymbol{x}_i - \boldsymbol{x}_j = \boldsymbol{\nu}_\ell} p(\boldsymbol{x}_i)^2 \frac{\alpha_k(\boldsymbol{x}_i)}{\alpha_\ell(\boldsymbol{x}_j)}}{\sum_{i,k,\ell | \boldsymbol{x}_j - \boldsymbol{x}_i = \boldsymbol{\nu}_k, \boldsymbol{x}_i - \boldsymbol{x}_j = \boldsymbol{\nu}_\ell} p(\boldsymbol{x}_i)} \tag{2.22}$$

To provide a way to estimate a new candidate state $p(\boldsymbol{x}_i)$ for an even more general situation where multiple states in the previously-calculated shell of the FSP may reach $\boldsymbol{x}_i$ via a reaction channel whose stoichiometric vector is not perpendicular to the unit vector of a given chemical species or a single state has multiple reactions channels that fire in a direction that is not perpendicular to the unit vector of a given chemical species,

we use the following:

$$p(\boldsymbol{x}_\ell) = \frac{\sum_{i=1}^{N} \sum_{j|(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i > 0} [p(\boldsymbol{x}_j)]^2 \frac{\sum_{k|\nu_{ik}<0} \alpha_k(\boldsymbol{x}_j)|\nu_{ik}|}{\sum_{k|\nu_{ik}>0} \alpha_k(\boldsymbol{x}_\ell)\nu_{ik}} + \sum_{i=1}^{N} \sum_{j|(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i < 0} [p(\boldsymbol{x}_j)]^2 \frac{\sum_{k|\nu_{ik}>0} \alpha_k(\boldsymbol{x}_j)\nu_{ik}}{\sum_{k|\nu_{ik}<0} \alpha_k(\boldsymbol{x}_\ell)|\nu_{ik}|}}{\sum_{i=1}^{N} \sum_{j|(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i > 0} p(\boldsymbol{x}_j) + \sum_{i=1}^{N} \sum_{j|(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i < 0} p(\boldsymbol{x}_j)} \tag{2.23}$$

where $\hat{\boldsymbol{e}}_i$ in the above equation represents the unit vector in the direction of the ith species and each $\boldsymbol{x}_j$ meets two conditions in addition to the ones explicitly included under each summation: first, that if $\boldsymbol{x}_\ell$ is a candidate state for a shell S_r in the FSP's state space, $\boldsymbol{x}_j$ is in the previous shell S_{r-1} and second, that $\boldsymbol{x}_\ell$ is reachable from $\boldsymbol{x}_j$ in a single reaction.

In the computational experiments we refer to separate detailed balance trials based on their β parameters. For instance, we will refer to a detailed balance FSP with $\beta = 10^{-2}$ as 10^{-2} DB and its initial estimate as 10^{-2} DB Est.

2.4.5 Detailed Balance Axes

It is possible to make a quick estimate of the marginal distributions using detailed balance without going through the whole algorithm. To do so, instead of performing the detailed balance procedure over the direction of every species (corresponding to the sums from 1 to N in Equation 2.23) for each new state being added to the FSP, we implement a change-of-coordinates for each stable equilibrium point, making the equilibrium point the origin of our new coordinate system and estimating the probability of the points along each new axis relative to the equilibrium point using detailed balance. The calculations along each axis are used to estimate the marginal distributions for each species. For each axis, we hold fixed the species count of the other species, and only allow the count of the ith species to vary for the ith axis. (The exception to this is in cases where conservation relationships force more than one species to change at a time; in these cases, if there are m conservation relationships that must be satisfied, we let there be m "dependent" variables, whose marginal distributions will be calculated separately. For instance, for the Schlogl example to follow in Section 2.7, the conservation relationship $[A]+[X]+[B] = 499$ must

be satisfied, and we choose to assign $[X]$ to be the dependent variable. Therefore, with an equilibrium point of $([A],[X],[B]) = (308, 17, 174)$, calculations for the A-axis are carried out along the line $[B] = 174$, allowing $[A]$ and $[X]$ to change in tandem; likewise, calculations for the B-axis are carried out along the line $[A] = 308$, allowing $[B]$ and $[X]$ to change in tandem.)

To explain the procedure, let us assume there is only one stable equilibrium point in the system. Afterward, we will discuss how to handle cases where there are multiple equilibrium points. If only only reaction is available to increase the count of the ith species and only one reaction is available to decrease the count of the ith species, we can employ Equation 2.20, where $\boldsymbol{x}_i$ is the state that is closer to the equilibrium point (or the equilibrium point itself in the case of the initial step) and $\boldsymbol{x}_j$ is the state that is farther away. In the case where multiple reactions are available, we take a variation of Equation 2.23 to use the previous point along the axis and all its associated outward (in the sense of moving away from the equilibrium point according to the copy number for the species under consideration) reactions to estimate the outward flow and the new point along the axis and all its associated inward (in the sense of moving closer to the equilibrium point in the copy number of the species under consideration) reactions:

$$p(\boldsymbol{x}_\ell) = p(\boldsymbol{x}_j)\frac{\sum_{k|\nu_{ik}<0}\alpha_k(\boldsymbol{x}_j)|\nu_{ik}|}{\sum_{k|\nu_{ik}>0}\alpha_k(\boldsymbol{x}_\ell)\nu_{ik}} \tag{2.24}$$

where $\boldsymbol{x}_\ell$ is the outer state and $\boldsymbol{x}_j$ is the inner state in the direction of the ith species compared to the equilibrium point in the case where $(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i > 0$. In the case where $(\boldsymbol{x}_j - \boldsymbol{x}_\ell)\cdot\hat{\boldsymbol{e}}_i < 0$, we have

$$p(\boldsymbol{x}_\ell) = [p(\boldsymbol{x}_j)]\frac{\sum_{k|\nu_{ik}>0}\alpha_k(\boldsymbol{x}_j)\nu_{ik}}{\sum_{k|\nu_{ik}<0}\alpha_k(\boldsymbol{x}_\ell)|\nu_{ik}|} \tag{2.25}$$

We expand from the equilibrium point in both directions along each axis, continuing in any direction as long as the estimated probability for each $\boldsymbol{x}_\ell$ satisfies $p(\boldsymbol{x}_\ell) \geq$

$\beta p(\boldsymbol{x}_{eq})$ for our given threshold β.

After the procedure has been carried out along each axis, we normalize to estimate the marginal probabilities, with $p_i(x_i)$ denoting the marginal distribution for the ith species and x_i the specific copy number of that species. We have

$$p_i(x_i^*) = \frac{p(\boldsymbol{x}^*)}{\sum_{\boldsymbol{x} \in C} p(\boldsymbol{x})} \tag{2.26}$$

where C is the set containing all points on the axis identified by the method that meet the estimated probability threshold.

If there are any dependent species, we modify Equation 2.26 by first computing the average over all axes' states with a species count x_i for the ith species; for instance, in the Schlogl example mentioned above, the marginal probability for $[X] = 18$ would be computed using an average of the point $(307, 18, 174)$ from the A-axis and $(308, 18, 173)$ from the B-axis. Our equation becomes

$$p_i(x_i) = \frac{p_i'(x_i)}{\sum_y p_i'(y)} \tag{2.27}$$

with

$$p_i'(y) = \frac{\sum_{x_i = y} p(\boldsymbol{x})}{\sum_{x_i = y} 1} \tag{2.28}$$

Once the marginal distributions have been estimated, we build an FSP shell by shell in a process similar to that of Section 2.4.4, accepting a state for the new shell if it is reachable in one reaction from a state in the previous shell and if the probability estimate for the new state is at least $\beta p(\boldsymbol{x}_{eq})$. The only difference is in how the probability estimate is made. The probabilities have already been estimated for the states lying along each axis. We use these as our initial probabilities (before we normalize) of each state $\boldsymbol{x}$ that lies along the x_i-axis:

$$p^{init}(\boldsymbol{x}) = p_i(x_i) \tag{2.29}$$

where, if any dependent variables are present, the i's are selected to be only associated with axes of variables that are *not* dependent. To compute probabilities for the states lying off the axes, we simply take

$$p^{init}(\boldsymbol{x}) = \prod_i p_i(x_i) \tag{2.30}$$

where the product does *not* include any dependent variables (such as $[X]$ in the Schlogl example). Once all probabilities have been computed, we normalize over the FSP's state space:

$$p(\boldsymbol{x}_\ell) = \frac{p^{init}(\boldsymbol{x}_\ell)}{\sum_j p^{init}(\boldsymbol{x}_j)} \tag{2.31}$$

where the sum is over every $\boldsymbol{x}_j$ in the FSP.

This discussion has assumed there is only one stable equilibrium point. In the case of w stable equilibria, we can compute Equations 2.26, 2.28, 2.27, 2.29, 2.30, and 2.31 for each equilibrium point separately. We can then calculate the estimates for each of these items for the entire system as an average of the computations from each of the equilibria by summing up corresponding terms from each equilibrium point and dividing by w (or by performing a different weighted average appropriate to the particular problem).

2.4.6 Reduced Full

The Reduced Full Solve is another FSP construction method we introduce, not because it is practical to construct, but because it can serve as a good reference point by which to evaluate other methods. First, we perform a Full Solve of the system for the entire state space under consideration (rendering this an impractical method under normal circumstances, since the point of the FSP is to avoid having to solve the full linear system). After the Full Solve has been completed, we select the equilibrium points along with any point that has at least some proportion β of the probability of the nearest

equilibrium point. That is, if

$$p(\boldsymbol{x}_i) \geq \beta p(\boldsymbol{x}_{eq})$$

then $\boldsymbol{x}_i$ will be included in the FSP. The point of this method is to show what an FSP would look like if we knew in advance exactly which states are the most likely and therefore exactly which states we would want to select in an FSP. Of course, we normally do not have this information, but having the Reduced Full FSP gives a useful point of comparison. It relays an ideal FSP and its results can be compared to that of other FSP methods to compare their performance.

2.5 Methods of Error Measurement

There are five ways given in Table 2.3 to measure error. It can be seen that the detailed balance and traditional methods matched each other exactly, except for minuscule rounding errors in Matlab. The "true" measure of error is given in the last column as $\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$, where $\boldsymbol{p}$ is the probability calculated from $\boldsymbol{Ap} = \boldsymbol{0}$ in the traditional (full) solve. $\hat{\boldsymbol{p}}$ is the estimated probability vector from any given method and σ is a function that rearranges this estimated probability vector into the same enumerated state order as the full solve; it assigns a probability of 0 to any state that is not represented in the estimate. The $\frac{1}{2}$ term is present for scaling; it ensures that the error is kept between 0 and 1.

The next-to-last column measures error in the marginal distribution estimates $\hat{\boldsymbol{p}}_i$. (Note that when in boldface, we use $\hat{\boldsymbol{p}}_i$ to denote the marginal probability function for the ith species, whereas p_i denotes the joint probability for the ith state in an enumerated list.) The leading coefficient $\frac{1}{2N}$ is also present for scaling; error is kept between 0 and 1. Note that all the methods described in the paper, with the sole exception of the Detailed Balance Axes Initial Estimate described later, are focused on providing a good estimate to the joint probability functions rather than the marginals. Nevertheless, the marginal

distribution error is included as a matter of interest and to highlight the performance and give a point of comparison for the Detailed Balance Axes Initial Estimate, which is tailored to quickly approximate marginal distributions.

Both of the above measures of error depend on knowing the full joint probability function and full marginal probability functions, respectively. However, this information is often unavailable. An FSP can be implemented in situations where the state space of a chemical reaction network is infinite or otherwise large enough that setting up the entire system is computationally prohibitive. The majority of the text that follows in this dissertation focuses on setting up FSPs for estimating stationary solutions to reaction networks efficiently by developing methods to select the most probable states for the FSP and discarding other states. Where feasible, the first two error methods will be included to study the performance of the methods under discussion. However, we include a couple measures of error that do not require solving the entire system, since the point of the FSP is to have methods that can be used on systems that are large enough that setting up the entire system is undesirable or infeasible. The first such error measurement is $||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1$, where $\hat{\boldsymbol{A}}$ once again is the portion of the original transition rate matrix $\boldsymbol{A}$ that remains after the rows and columns representing states not associated with states that remain in the FSP have been removed. In cases where $\hat{\boldsymbol{p}}$ is chosen to satisfy the equation $\boldsymbol{A}^{FSP}\hat{\boldsymbol{p}} = \hat{\boldsymbol{0}}$, since $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}^{FSP}$ differ only in that $\boldsymbol{A}^{FSP}$ maps any probability leaking out of the FSP back to the estimated equilibrium points whereas $\hat{\boldsymbol{A}}$ allows it to escape the FSP, $||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1$ becomes a measure of how quickly probability is leaking out of the FSP. $||\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})||_1$ is a similar measure included for comparison with $||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1$, but it once again requires knowing $\boldsymbol{A}$ in its entirety. $\frac{||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1}{||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}||_1}$ is likewise used as a variant of $||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1$, dividing it by the corresponding error from a probability distribution consisting of the same state space, but giving equal probability to every state in the state space. It is included because the magnitude of $||\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}||_1$ varies greatly based on the magnitude of the values in $\boldsymbol{A}$, making it difficult to compare across different systems, which may have

very different magnitudes in their reaction rates. In many cases, these three methods that are not dependent on knowing the actual solution may give a good indication of how well the FSP is doing. However, we will note again that when $\hat{\boldsymbol{p}}$ is chosen to solve the linear system $\boldsymbol{A}^{FSP}\hat{\boldsymbol{p}} = \hat{\mathbf{0}}$ that their interpretation basically boils down to how fast probability mass is leaving the FSP. On one hand, this can be very helpful; in many cases probability mass will be leaking out of the FSP very slowly if the FSP has been selected in a way that retains the most probable states. However, as we will see in the Schlogl example, when one reaction is much faster than another, these measures can be misleading. Nonetheless, we can use these measures as a proxy for error when the true error measure is not easily obtainable.

We also compute runtime for many of our experiments as a measure of computational efficiency. For consistency, all experiments in this chapter that report runtime were performed using Matlab on the same desktop computer. (The experiments from Chapter 3 were completed in Matlab using a single laptop.)

2.6 Application Example: Gene Toggle

The Michaelis-Menten system led to a single stationary state and the Dimerization system yielded numerous states with positive stationary probability but a single maximum. We now consider another system that yields a bimodal stationary solution: the reversible chemical system involving a Gene Toggle Reaction described by Sidje and Vo [23]:

	Reaction	Propensity
1	$u \to \emptyset$	$(d_1 + \frac{\gamma s}{1+s})u$
2	$\emptyset \to u$	$\epsilon(\alpha_1 + \frac{\beta_1 K_1^3}{K_1^3+v^3})$
3	$v \to \emptyset$	$d_2 v$
4	$\emptyset \to v$	$\epsilon(\alpha_2 + \frac{\beta_2 K_2^3}{K_2^3+u^3})$

Figure 2.4: Gene Toggle reactions and their propensity functions with $d_1 = d_2 = 1, s = 0.1, K_1 = K_2 = 100, \beta_1 = \beta_2 = 400, \alpha_1 = \alpha_2 = 20, \gamma = 1, \epsilon = 1$

There are no conservation equations present, so we have only to solve a system based on reaction rates

$$\begin{cases} \epsilon(\alpha_1 + \frac{\beta_1 K_1^3}{K_1^3+[V]^3}) - [U](d_1 + \frac{\gamma s}{1+s}) = 0 \\ \epsilon(\alpha_2 + \frac{\beta_2 K_2^3}{K_2^3+[U]^3}) - d_2[V] = 0 \end{cases}$$

The system has Jacobian

$$J = \begin{bmatrix} -(d_1 + \frac{\gamma s}{1+s}) & \frac{\epsilon\beta_1 K_1^3}{(K_1^3+V^3)^2} 3[V]^2 \\ \frac{\epsilon\beta_2 K_2^3}{(K_2^3+U^3)^2} 3[U]^2 & -d_2 \end{bmatrix}$$

Solving the system reveals two equilibrium points: a saddle point and two relative maxima.

Equilibrium Point	Jacobian Eigenvalues	Identification
(378,27)	-1.2685 -0.8224	Relative Maximum
(140,126)	-2.9304 0.8395	Saddle Point
(23,415)	-1.2081 -0.8828	Relative Maximum

Table 2.4: Equilibrium point analysis for Gene Toggle

We have presented both the original results given by Matlab and "reweighted" results. This is due to the fact that we have two relative maxima and the probabilities of the states between those maxima are so small that the two clusters of probability essentially do not communicate. As a result, there are infinitely many solutions that allow an $\mathcal{O}(10^{-13})$-level accuracy (the level of precision Matlab yields for this problem) for the equation $\boldsymbol{Ap} = \mathbf{0}$. We can weight each cluster however we want as long as both weights are nonnegative and sum to one. We present the original results given by Matlab, but also reweight each method so that each cluster has 50% of the probability mass in each method so that the methods can be better-compared to one another.

After this reweighting scheme, we find that all the methods do well and match the general shape of the "Full" solution. (We note that the "Full" solution is an FSP itself; based on the propensity functions in Table 2.4, the state space is actually infinite. However, the propensity functions also show that for large values of $[U]$, Reaction 1

Method	States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$
Full	3.60e+05	1.19e-13	2.65e-14	1.19e-13
Reduced Full $10^{-1}p_{eq}$	2.95e+03	4.08e-01	2.25e-02	8.16e-01
Reduced Full $10^{-2}p_{eq}$	5.90e+03	6.72e-02	4.91e-03	1.34e-01
Reduced Full $10^{-3}p_{eq}$	8.83e+03	1.04e-02	8.86e-04	2.09e-02
Reduced Full $10^{-4}p_{eq}$	1.18e+04	1.42e-03	1.34e-04	2.85e-03
Reduced Full $10^{-5}p_{eq}$	1.47e+04	1.81e-04	1.83e-05	3.61e-04
Reduced Full $10^{-6}p_{eq}$	1.76e+04	2.14e-05	2.31e-06	4.27e-05
Reduced Full $10^{-15}p_{eq}$	4.33e+04	1.19e-13	1.69e-14	1.66e-13
Trad. SSA Trial 1:	6.03e+03	1.96e-01	1.47e-03	3.91e-01
Trad. SSA Trial 2:	6.06e+03	1.79e-01	1.35e-03	3.58e-01
Trad. SSA Trial 3:	5.98e+03	2.01e-01	1.49e-03	4.02e-01
Trad. SSA Trials 1 and 2:	6.91e+03	1.16e-01	9.92e-04	9.15e+02
Trad. SSA Trials 1 and 3:	6.92e+03	1.09e-01	9.35e-04	9.09e+02
Trad. SSA Trials 2 and 3:	6.89e+03	1.31e-01	1.11e-03	9.21e+02
Trad. SSA Trials 1-3:	7.44e+03	8.27e-02	7.67e-04	8.88e+02
R-Step SSA (3000 **min.**)	3.06e+03	4.55e-01	1.74e-02	9.10e-01
R-Step SSA (6000 **min.**)	6.07e+03	7.43e-02	4.01e-03	1.49e-01
R-Step SSA (9000 **min.**)	9.06e+03	1.63e-02	9.61e-04	3.27e-02
R-Step SSA (12000 **min.**)	1.20e+04	2.79e-03	2.01e-04	5.58e-03
R-Step SSA (15000 **min.**)	1.50e+04	4.49e-04	3.31e-05	8.99e-04
R-Step SSA (18000 **min.**)	1.82e+04	7.34e-05	6.00e-06	1.47e-04
R-step Reach. (40 steps)	6.10e+03	3.26e-01	1.49e-02	6.52e-01
R-step Reach. (50 steps)	8.94e+03	1.27e-01	7.13e-03	2.54e-01
R-step Reach. (60 steps)	1.22e+04	4.41e-02	2.89e-03	8.82e-02
R-step Reach. (70 steps)	1.58e+04	1.29e-02	9.54e-04	2.57e-02
R-step Reach. (80 steps)	1.99e+04	3.06e-03	2.51e-04	6.11e-03
R-step Reach. (90 steps)	2.43e+04	5.91e-04	5.31e-05	1.18e-03
DB $10^{-1}p_{eq}$	2.95e+03	4.07e-01	2.25e-02	8.14e-01
DB Est. $10^{-1}p_{eq}$	2.95e+03	3.97e+00	2.19e-01	7.90e+00
DB $10^{-2}p_{eq}$	5.89e+03	6.71e-02	4.90e-03	1.34e-01
DB Est. $10^{-2}p_{eq}$	5.89e+03	5.50e-01	4.02e-02	1.06e+00
DB $10^{-3}p_{eq}$	8.86e+03	1.03e-02	8.72e-04	2.05e-02
DB Est. $10^{-3}p_{eq}$	8.86e+03	1.08e-01	9.20e-03	1.69e-01
DB $10^{-4}p_{eq}$	1.18e+04	1.39e-03	1.31e-04	2.79e-03
DB Est. $10^{-4}p_{eq}$	1.18e+04	5.51e-02	5.19e-03	6.22e-02
DB $10^{-5}p_{eq}$	1.48e+04	1.77e-04	1.79e-05	3.53e-04
DB Est. $10^{-5}p_{eq}$	1.48e+04	4.90e-02	4.96e-03	4.98e-02
DB $10^{-6}p_{eq}$	1.77e+04	2.11e-05	2.28e-06	4.22e-05
DB Est. $10^{-6}p_{eq}$	1.77e+04	4.83e-02	5.23e-03	4.84e-02
DB Axes $10^{-1}p_{eq}$	2.89e+03	4.14e-01	2.29e-02	8.27e-01
DB Axes Est. $10^{-1}p_{eq}$	2.89e+03	4.09e+00	2.26e-01	8.02e+00
DB Axes $10^{-2}p_{eq}$	5.77e+03	7.41e-02	5.33e-03	1.48e-01
DB Axes Est. $10^{-2}p_{eq}$	5.77e+03	7.41e-01	5.33e-02	1.25e+00
DB Axes $10^{-3}p_{eq}$	8.63e+03	1.41e-02	1.19e-03	2.81e-02
DB Axes Est. $10^{-3}p_{eq}$	8.63e+03	3.05e-01	2.57e-02	3.67e-01
DB Axes $10^{-4}p_{eq}$	1.15e+04	2.63e-03	2.47e-04	5.27e-03
DB Axes Est. $10^{-4}p_{eq}$	1.15e+04	2.53e-01	2.37e-02	2.60e-01
DB Axes $10^{-5}p_{eq}$	1.43e+04	5.11e-04	5.16e-05	1.02e-03
DB Axes Est. $10^{-5}p_{eq}$	1.43e+04	2.47e-01	2.50e-02	2.48e-01
DB Axes $10^{-6}p_{eq}$	1.72e+04	1.02e-04	1.10e-05	2.03e-04
DB Axes Est. $10^{-6}p_{eq}$	1.72e+04	2.47e-01	2.66e-02	2.47e-01

Table 2.5: Results for Gene Toggle (not reweighted)

Method	Total States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{4}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
Full	3.60e+05	1.47e-13	3.28e-14	1.47e-13	0.00e+00	0.00e+00
Reduced Full $10^{-1}p_{eq}$	2.95e+03	4.08e-01	2.25e-02	8.16e-01	1.42e-01	2.09e-01
Reduced Full $10^{-2}p_{eq}$	5.90e+03	6.72e-02	4.91e-03	1.34e-01	3.41e-02	4.89e-02
Reduced Full $10^{-3}p_{eq}$	8.83e+03	1.04e-02	8.86e-04	2.09e-02	6.49e-03	9.14e-03
Reduced Full $10^{-4}p_{eq}$	1.18e+04	1.42e-03	1.34e-04	2.85e-03	1.02e-03	1.41e-03
Reduced Full $10^{-5}p_{eq}$	1.47e+04	1.81e-04	1.83e-05	3.61e-04	1.43e-04	1.96e-04
Reduced Full $10^{-6}p_{eq}$	1.76e+04	2.14e-05	2.31e-06	4.27e-05	1.84e-05	2.49e-05
Reduced Full $10^{-15}p_{eq}$	4.33e+04	1.31e-13	1.86e-14	1.78e-13	6.13e-14	7.89e-14
Trad. SSA Trial 1:	6.03e+03	1.96e-01	1.47e-03	3.92e-01	8.11e-02	1.17e-01
Trad. SSA Trial 2:	6.06e+03	1.79e-01	1.35e-03	3.58e-01	7.57e-02	1.10e-01
Trad. SSA Trial 3:	5.98e+03	2.01e-01	1.49e-03	4.03e-01	8.32e-02	1.22e-01
Trad. SSA Trials 1 and 2:	6.91e+03	1.17e-01	9.97e-04	2.33e-01	6.09e-02	7.78e-02
Trad. SSA Trials 1 and 3:	6.92e+03	1.10e-01	9.38e-04	2.19e-01	5.55e-02	7.39e-02
Trad. SSA Trials 2 and 3:	6.89e+03	1.31e-01	1.11e-03	2.63e-01	6.11e-02	8.69e-02
Trad. SSA Trials 1-3:	7.44e+03	8.27e-02	7.67e-04	1.65e-01	4.20e-02	5.82e-02
R-Step SSA (3000 **min.**)	3.06e+03	4.56e-01	1.74e-02	9.13e-01	1.56e-01	2.28e-01
R-Step SSA (6000 **min.**)	6.07e+03	8.50e-02	4.58e-03	1.70e-01	4.20e-02	6.02e-02
R-Step SSA (9000 **min.**)	9.06e+03	1.85e-02	1.09e-03	3.69e-02	1.21e-02	1.67e-02
R-Step SSA (12000 **min.**)	1.20e+04	3.09e-03	2.22e-04	6.19e-03	2.35e-03	3.19e-03
R-Step SSA (15000 **min.**)	1.50e+04	4.79e-04	3.53e-05	9.58e-04	4.48e-04	5.87e-04
R-Step SSA (18000 **min.**)	1.82e+04	7.63e-05	6.25e-06	1.53e-04	8.52e-05	1.06e-04
R-step Reach. (40 steps)	6.10e+03	3.26e-01	1.49e-02	6.52e-01	1.20e-01	1.74e-01
R-step Reach. (50 steps)	8.94e+03	1.27e-01	7.14e-03	2.54e-01	5.89e-02	8.58e-02
R-step Reach. (60 steps)	1.22e+04	4.42e-02	2.90e-03	8.84e-02	2.47e-02	3.58e-02
R-step Reach. (70 steps)	1.58e+04	1.29e-02	9.57e-04	2.58e-02	8.45e-03	1.21e-02
R-step Reach. (80 steps)	1.99e+04	3.07e-03	2.53e-04	6.14e-03	2.31e-03	3.27e-03
R-step Reach. (90 steps)	2.43e+04	5.94e-04	5.33e-05	1.19e-03	5.07e-04	7.04e-04
DB $10^{-1}p_{eq}$	2.95e+03	4.07e-01	2.25e-02	8.14e-01	1.42e-01	2.09e-01
DB Est. $10^{-1}p_{eq}$	2.95e+03	3.97e+00	2.19e-01	7.90e+00	5.24e-02	9.98e-02
DB $10^{-2}p_{eq}$	5.89e+03	6.71e-02	4.90e-03	1.34e-01	3.43e-02	4.92e-02
DB Est. $10^{-2}p_{eq}$	5.89e+03	5.50e-01	4.01e-02	1.06e+00	1.05e-02	1.73e-02
DB $10^{-3}p_{eq}$	8.86e+03	1.03e-02	8.72e-04	2.05e-02	6.48e-03	9.06e-03
DB Est. $10^{-3}p_{eq}$	8.86e+03	1.08e-01	9.19e-03	1.69e-01	7.38e-03	1.23e-02
DB $10^{-4}p_{eq}$	1.18e+04	1.39e-03	1.31e-04	2.79e-03	1.02e-03	1.40e-03
DB Est. $10^{-4}p_{eq}$	1.18e+04	5.50e-02	5.17e-03	6.20e-02	7.16e-03	1.19e-02
DB $10^{-5}p_{eq}$	1.48e+04	1.77e-04	1.79e-05	3.54e-04	1.44e-04	1.94e-04
DB Est. $10^{-5}p_{eq}$	1.48e+04	4.88e-02	4.94e-03	4.96e-02	7.14e-03	1.19e-02
DB $10^{-6}p_{eq}$	1.77e+04	2.11e-05	2.28e-06	4.22e-05	1.88e-05	2.50e-05
DB Est. $10^{-6}p_{eq}$	1.77e+04	4.81e-02	5.21e-03	4.82e-02	7.14e-03	1.19e-02
DB Axes $10^{-1}p_{eq}$	2.89e+03	4.14e-01	2.29e-02	8.27e-01	1.48e-01	2.17e-01
DB Axes Est. $10^{-1}p_{eq}$	2.89e+03	4.09e+00	2.26e-01	8.02e+00	4.20e-02	1.24e-01
DB Axes $10^{-2}p_{eq}$	5.77e+03	7.42e-02	5.34e-03	1.48e-01	4.10e-02	5.75e-02
DB Axes Est. $10^{-2}p_{eq}$	5.77e+03	7.41e-01	5.33e-02	1.25e+00	1.87e-02	6.65e-02
DB Axes $10^{-3}p_{eq}$	8.63e+03	1.42e-02	1.19e-03	2.84e-02	1.05e-02	1.41e-02
DB Axes Est. $10^{-3}p_{eq}$	8.63e+03	3.05e-01	2.57e-02	3.67e-01	1.74e-02	6.29e-02
DB Axes $10^{-4}p_{eq}$	1.15e+04	2.72e-03	2.54e-04	5.43e-03	2.59e-03	3.31e-03
DB Axes Est. $10^{-4}p_{eq}$	1.15e+04	2.53e-01	2.37e-02	2.60e-01	1.73e-02	6.26e-02
DB Axes $10^{-5}p_{eq}$	1.43e+04	5.49e-04	5.53e-05	1.10e-03	6.42e-04	7.89e-04
DB Axes Est. $10^{-5}p_{eq}$	1.43e+04	2.47e-01	2.50e-02	2.48e-01	1.73e-02	6.26e-02
DB Axes $10^{-6}p_{eq}$	1.72e+04	1.17e-04	1.26e-05	2.34e-04	1.62e-04	1.93e-04
DB Axes Est. $10^{-6}p_{eq}$	1.72e+04	2.47e-01	2.66e-02	2.47e-01	1.73e-02	6.26e-02

Table 2.6: Results for Gene Toggle (reweighted)

Method	Total States	State Selection Runtime (s)	Linear Solve Runtime (s)	Marginals Runtime (s)
Full	3.60e+05	1.01e+05	3.68e+01	1.01e+05
Reduced Full $10^{-1}p_{eq}$	2.95e+03	1.01e+05	3.68e+01	1.01e+05
Reduced Full $10^{-2}p_{eq}$	5.90e+03	1.01e+05	3.68e+01	1.01e+05
Reduced Full $10^{-3}p_{eq}$	8.83e+03	1.01e+05	3.68e+01	1.01e+05
Reduced Full $10^{-4}p_{eq}$	1.18e+04	1.01e+05	3.68e+01	1.01e+05
Reduced Full $10^{-5}p_{eq}$	1.47e+04	1.01e+05	3.69e+01	1.01e+05
Reduced Full $10^{-6}p_{eq}$	1.76e+04	1.01e+05	3.69e+01	1.01e+05
Reduced Full $10^{-15}p_{eq}$	4.33e+04	1.02e+05	3.74e+01	1.02e+05
Trad. SSA Trial 1:	6.03e+03	8.79e+04	2.83e-02	8.79e+04
Trad. SSA Trial 2:	6.06e+03	8.91e+04	2.80e-02	8.91e+04
Trad. SSA Trial 3:	5.98e+03	8.80e+04	4.09e-02	8.80e+04
Trad. SSA Trials 1 and 2:	6.91e+03	1.77e+05	3.52e-02	1.77e+05
Trad. SSA Trials 1 and 3:	6.92e+03	1.76e+05	3.68e-02	1.76e+05
Trad. SSA Trials 2 and 3:	6.89e+03	1.77e+05	3.52e-02	1.77e+05
Trad. SSA Trials 1-3:	7.44e+03	2.65e+05	4.07e-02	2.65e+05
R-Step SSA (3000 **min.**)	3.06e+03	4.31e+01	1.32e-02	4.31e+01
R-Step SSA (6000 **min.**)	6.07e+03	2.29e+02	3.32e-02	2.29e+02
R-Step SSA (9000 **min.**)	9.06e+03	5.51e+02	6.08e-02	5.51e+02
R-Step SSA (12000 **min.**)	1.20e+04	1.15e+03	1.02e-01	1.15e+03
R-Step SSA (15000 **min.**)	1.50e+04	1.96e+03	1.14e-01	1.96e+03
R-Step SSA (18000 **min.**)	1.82e+04	3.19e+03	1.83e-01	3.19e+03
R-step Reach. (40 steps)	6.10e+03	2.98e+01	3.23e-02	2.98e+01
R-step Reach. (50 steps)	8.94e+03	6.38e+01	5.79e-02	6.38e+01
R-step Reach. (60 steps)	1.22e+04	1.18e+02	9.61e-02	1.18e+02
R-step Reach. (70 steps)	1.58e+04	1.97e+02	1.23e-01	1.97e+02
R-step Reach. (80 steps)	1.99e+04	3.11e+02	1.86e-01	3.12e+02
R-step Reach. (90 steps)	2.43e+04	4.64e+02	2.58e-01	4.64e+02
DB $10^{-1}p_{eq}$	2.95e+03	1.37e+01	1.26e-02	1.37e+01
DB Est. $10^{-1}p_{eq}$	2.95e+03	1.37e+01	0.00e+00	1.37e+01
DB $10^{-2}p_{eq}$	5.89e+03	5.13e+01	2.97e-02	5.13e+01
DB Est. $10^{-2}p_{eq}$	5.89e+03	5.13e+01	0.00e+00	5.13e+01
DB $10^{-3}p_{eq}$	8.86e+03	1.13e+02	5.28e-02	1.13e+02
DB Est. $10^{-3}p_{eq}$	8.86e+03	1.13e+02	0.00e+00	1.13e+02
DB $10^{-4}p_{eq}$	1.18e+04	1.99e+02	9.82e-02	1.99e+02
DB Est. $10^{-4}p_{eq}$	1.18e+04	1.99e+02	0.00e+00	1.99e+02
DB $10^{-5}p_{eq}$	1.48e+04	3.09e+02	1.09e-01	3.09e+02
DB Est. $10^{-5}p_{eq}$	1.48e+04	3.09e+02	0.00e+00	3.09e+02
DB $10^{-6}p_{eq}$	1.77e+04	4.42e+02	1.50e-01	4.42e+02
DB Est. $10^{-6}p_{eq}$	1.77e+04	4.42e+02	0.00e+00	4.42e+02
DB Axes $10^{-1}p_{eq}$	2.89e+03	1.20e+01	1.15e-02	1.21e+01
DB Axes Est. $10^{-1}p_{eq}$	2.89e+03	1.20e+01	0.00e+00	6.36e-02
DB Axes $10^{-2}p_{eq}$	5.77e+03	4.59e+01	2.85e-02	4.59e+01
DB Axes Est. $10^{-2}p_{eq}$	5.77e+03	4.59e+01	0.00e+00	6.19e-02
DB Axes $10^{-3}p_{eq}$	8.63e+03	1.01e+02	4.84e-02	1.01e+02
DB Axes Est. $10^{-3}p_{eq}$	8.63e+03	1.01e+02	0.00e+00	5.31e-02
DB Axes $10^{-4}p_{eq}$	1.15e+04	1.77e+02	7.45e-02	1.77e+02
DB Axes Est. $10^{-4}p_{eq}$	1.15e+04	1.77e+02	0.00e+00	5.20e-02
DB Axes $10^{-5}p_{eq}$	1.43e+04	2.68e+02	1.03e-01	2.69e+02
DB Axes Est. $10^{-5}p_{eq}$	1.43e+04	2.68e+02	0.00e+00	6.08e-02
DB Axes $10^{-6}p_{eq}$	1.72e+04	3.85e+02	1.33e-01	3.85e+02
DB Axes Est. $10^{-6}p_{eq}$	1.72e+04	3.85e+02	0.00e+00	6.17e-02

Table 2.7: Results for Gene Toggle

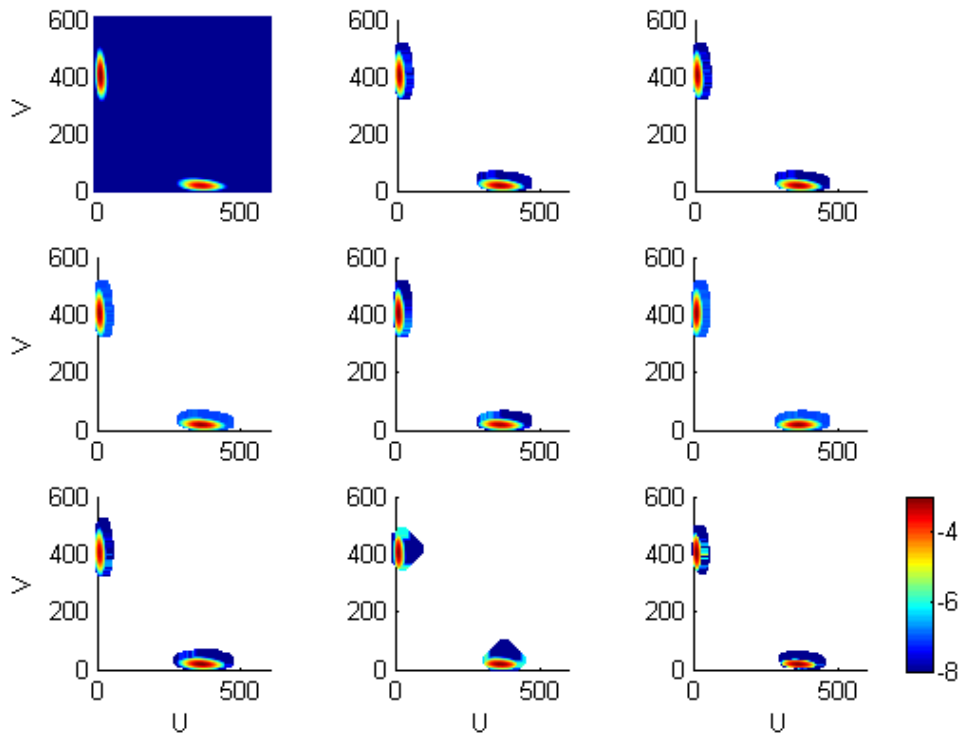

Figure 2.5: Probability distribution from Gene Toggle reaction system. Top row (left to right): Full Solve, 10^{-4} Reduced, 10^{-4} DB. Second row: 10^{-4} DB Est., 10^{-4} DB Axes, 10^{-4} DB Axes Est. Third row: $12,000$-state R-Step SSA, 60-step R-step reachability, Traditional SSA Trials 1-3. Color represents the log of probability - that is, $\log_{10} p(\boldsymbol{x})$ - according to the values on the colorbar, with any values of $\log_{10} p(\boldsymbol{x})$ in the state space exceeding the maximum value shown on the colorbar made to be dark red, any values of $\log_{10} p(\boldsymbol{x})$ below the minimum value shown on the colorbar made to be dark blue, and values with 0 probabiliy or outside the state space left blank. This is the color scheme used on two-variable probability plots throughout the rest of the dissertation.

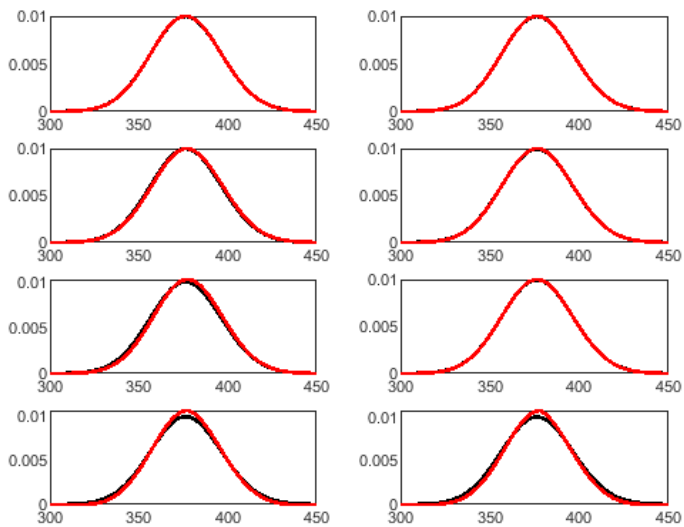

Figure 2.6: Marginal distribution for $[U]$ around the $(378, 27)$ equilibrium point of the Gene Toggle reaction system. Top Row: 10^{-4} Reduced (left), 10^{-4} DB (right). Second Row: 10^{-4} DB Est., 10^{-4} DB Axes. Third Row: 10^{-4} DB Axes Est, $12,000$-state R-Step SSA. Fourth Row: 60-step R-step reachability, Traditional SSA Trials 1-3. Full Solve shown in black in each graph; other method shown in red for each graph. If the Full Solve is not visible, it is overlaid by the other method.

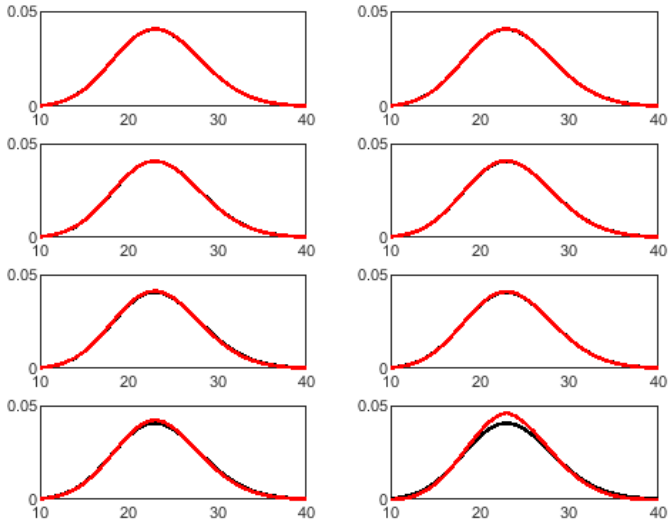

Figure 2.7: Marginal distribution for $[U]$ around the $(23, 415)$ equilibrium point of the Gene Toggle reaction system. Top Row: 10^{-4} Reduced (left), 10^{-4} DB (right). Second Row: 10^{-4} DB Est., 10^{-4} DB Axes. Third Row: 10^{-4} DB Axes Est, $12,000$-state R-Step SSA. Fourth Row: 60-step R-step reachability, Traditional SSA Trials 1-3. Full Solve shown in black in each graph; other method shown in red for each graph. If the Full Solve is not visible, it is overlaid by the other method.

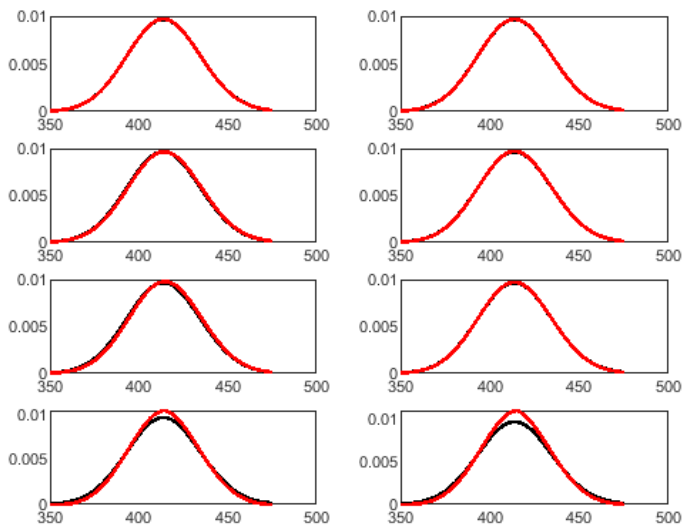

Figure 2.8: Marginal distribution for $[V]$ around the $(23, 415)$ equilibrium point of the Gene Toggle reaction system. Top Row: 10^{-4} Reduced (left), 10^{-4} DB (right). Second Row: 10^{-4} DB Est., 10^{-4} DB Axes. Third Row: 10^{-4} DB Axes Est, $12,000$-state R-Step SSA. Fourth Row: 60-step R-step reachability, Traditional SSA Trials 1-3. Full Solve shown in black in each graph; other method shown in red for each graph. If the Full Solve is not visible, it is overlaid by the other method.

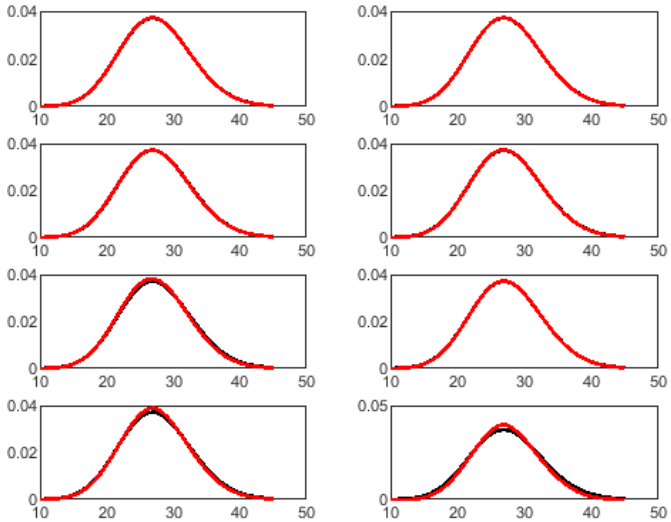

Figure 2.9: Marginal distribution for $[V]$ around the $(378, 27)$ equilibrium point of the Gene Toggle reaction system. Top Row: 10^{-4} Reduced (left), 10^{-4} DB (right). Second Row: 10^{-4} DB Est., 10^{-4} DB Axes. Third Row: 10^{-4} DB Axes Est, $12,000$-state R-Step SSA. Fourth Row: 60-step R-step reachability, Traditional SSA Trials 1-3. Full Solve shown in black in each graph; other method shown in red for each graph. If the Full Solve is not visible, it is overlaid by the other method.

dominates Reaction 2; likewise, for large values of $[V]$, Reaction 3 dominates Reaction 4. The probability that either species has a count of 600 or greater in the stationary case is extremely small, so we construct an FSP allowing both variables to take on any value from 0 to 599 with states outside these bounds omitted, leading to a state space of $360,000$ states.)

We observe that each of our Traditional FSP trials (which were $80,000$ SSA runs apiece) took an extremely long time to run. This was because many reactions were allowed to give the system sufficient time to diffuse, and only the final state was selected for inclusion in the state space. Moreover, after a while, they also had a very hard time finding new states, as evidenced by the fact that combining all three trials led to a combined state space only about $1,400$ states larger than the state space from any individual trial, because the states near the relative maxima are much more likely than states far away. For these reasons, we omit the Traditional FSP in the rest of our experiments.

We note that the DB Axes Initial Estimate produces very accurate results despite the fact that its marginal distributions are produced almost instantly. It is able to produce these marginal estimations so quickly because, unlike the other techniques, the DB Axes marginal estimations do not rely on knowledge of a probability estimate for all of the states included in the FSP.

All the other methods performed well, though it is worth noting that because it does nothing to discriminate between likely and unlikely states, the r-step reachability method included many unnecessary states.

2.7 Application Example: Schlogl

Consider a Schlögl reaction system given in Figure 2.10 where we are concerned with tracking the copy number of three species: A, X, and Y.

Though the system involves tracking three chemical species, one layer of redun-

	Reaction	Propensity
1	$A + 2X \rightarrow 3X$	$k_1[A][X]([X]-1)$
2	$3X \rightarrow A + 2X$	$k_2[X]([X]-1)([X]-2)$
3	$B \rightarrow X$	$k_3[B]$
4	$X \rightarrow B$	$k_4[X]$

Figure 2.10: Schlögl reactions and their propensity functions with $k_1 = 0.1, k_2 = 2, k_3 = 0.1, k_4 = 1$

dancy exists, because the system obeys the conservation equation

$$[A] + [X] + [B] = \textbf{constant}$$

.

For this study, we will take the constant to be 499, yielding the following system of equations:

$$\begin{cases} -k_1[A][X]([X]-1) + k_2[X]([X]-1)([X]-2) = 0 \\ k_1[A][X]([X]-1) - k_2[X]([X]-1)([X]-2) + k_3[B] - k_4[X] = 0 \\ -k_3[B] + k_4[X] = 0 \\ ([A] + [X] + [B]) = 499 \end{cases}$$

with Jacobian

$$J = \begin{bmatrix} -k_1([X])([X]-1) & 3k_2[B][X]^2 - 6k_2[X] + 2k_2 & 0 \\ k_1([X])([X]-1) & -(3k_2[B][X]^2 - 6k_2[X] + 2k_2) - k_4 & k_3 \\ 0 & k_4 & -k_3 \end{bmatrix}$$

The system yields $(308, 17, 174)$ as the approximate equilibrium state, which is taken to be a maximum based on the negative eigenvalues, as seen in Table 2.8. (One zero eigenvalue is present since the system is linearly dependent, given the conservation

Estimated Equilibrium Point	Jacobian Eigenvalues	Identification	Calculated Maximum (Linear Solve)
$(308, 17, 174)$	-1.56e+03 -1.17e-01 1.19e-14	Relative Maximum	(322,16,161)

Table 2.8: Equilibrium point estimation for Schlögl

equation.)

We observe that while many of the methods struggle to approximate the true probability distribution, the Detailed Balance Initial Estimate thrives. We take a look at challenges facing each method.

The DB Axes Initial Estimate fails to produce the negative correlation between A and B observed in the results. This is because embedded in the method is the assumption that A and B are independent random variables. The DB Axes Initial Estimate only approximates the marginal distributions $\boldsymbol{p}_A$ and $\boldsymbol{p}_B$ directly. To estimate the joint probability, a product of the marginals is taken:

$$\boldsymbol{p}([A],[B]) = \boldsymbol{p}_A([A])\boldsymbol{p}_B([B]) \tag{2.32}$$

(Due to the conservation equation, X was taken to be a dependent random variable that could be perfectly calculated according to the equation $[X] = 499 - [A] - [B]$. If the conservation equation were not present, the joint probability estimation would have been $\boldsymbol{p}([A],[X],[B]) = \boldsymbol{p}_A([A])\boldsymbol{p}_X([X])\boldsymbol{p}_B([B]))$. Equation 2.32 is precisely a statement that A and B are independent random variables. However, as shown by the strong negative correlation between the variables in the joint probability distribution, A and B are clearly not independent. (This negative correlation might have been anticipated from the conservation equation $[B] = 499 - [A] - [X]$). Therefore, the Schlögl system is not an ideal reaction network to employ the DB Axes Initial Estimate.

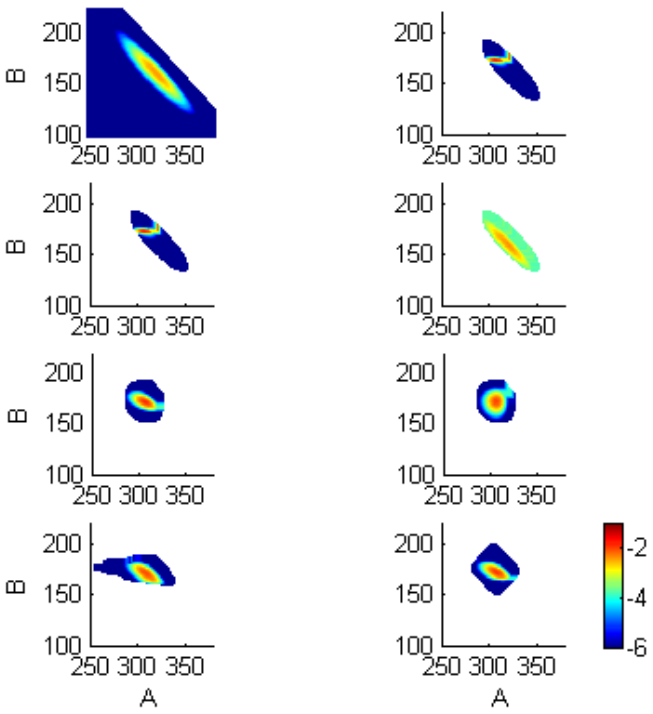

Figure 2.11: Probability distribution from Schlögl reaction system. Top row: Full Solve (left), 10^{-1} Reduced (right). Second row: 10^{-1} DB, 10^{-1} DB Est. Third row: 10^{-5} DB Axes, 10^{-5} DB Axes Est. Fourth row: 1000-state R-Step SSA, 1000-state R-step reachability.

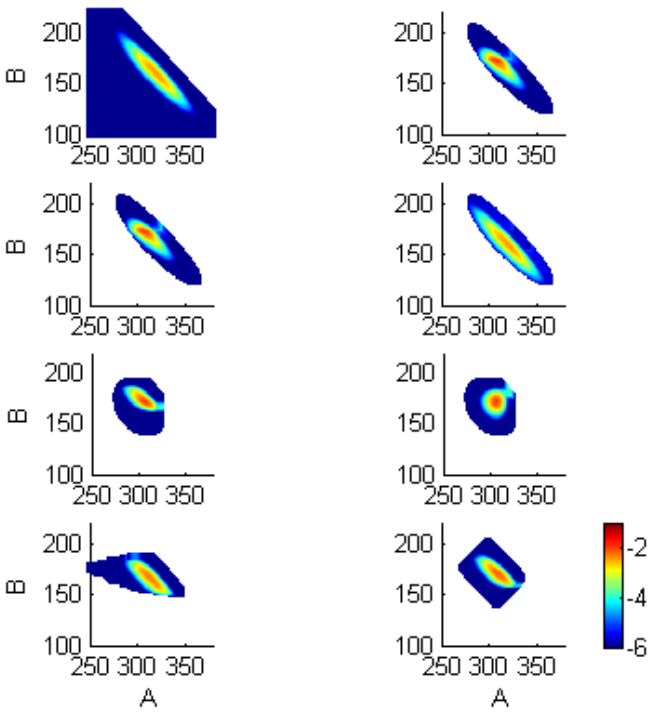

Figure 2.12: Probability distribution from Schlögl reaction system. Top row: Full Solve (left), 10^{-3} Reduced (right). Second row: 10^{-3} DB, 10^{-3} DB Est. Third row: 10^{-11} DB Axes, 10^{-11} DB Axes Est. Fourth row: 2000-state R-Step SSA, 2000-state R-step reachability.

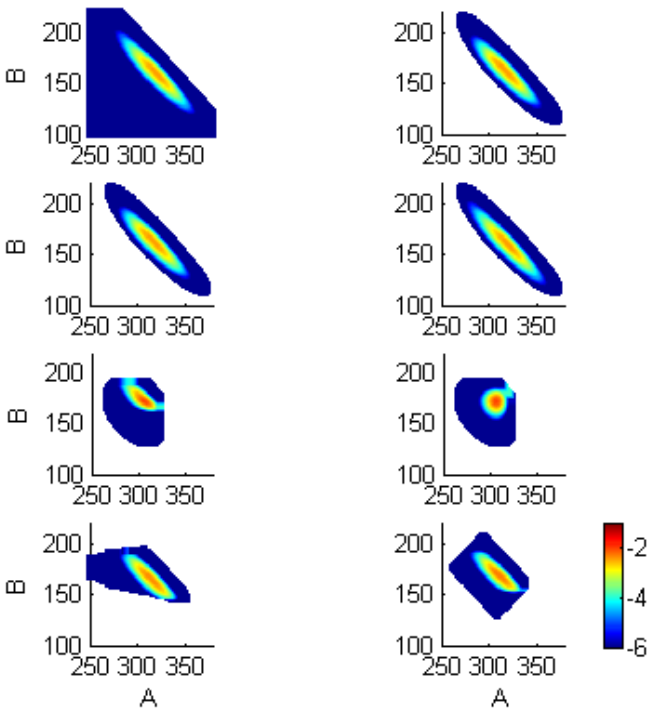

Figure 2.13: Probability distribution from Schlögl reaction system. Top row: Full Solve (left), 10^{-5} Reduced (right). Second row: 10^{-5} DB, 10^{-5} DB Est. Third row: 10^{-18} DB Axes, 10^{-18} DB Axes Est. Fourth row: 3000-state R-Step SSA, 3000-state R-step reachability.

Method	Total States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{6}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
Full	1.25e+05	1.53e-12	3.10e-18	1.53e-12	0.00e+00	0.00e+00
Reduced Full $10^{-1}p_{eq}$	8.18e+02	6.24e+01	2.65e-02	1.25e+02	5.78e-01	9.08e-01
Reduced Full $10^{-2}p_{eq}$	1.41e+03	1.04e+01	3.92e-03	2.07e+01	5.11e-01	8.08e-01
Reduced Full $10^{-3}p_{eq}$	1.98e+03	1.56e+00	5.25e-04	3.13e+00	3.89e-01	5.96e-01
Reduced Full $10^{-4}p_{eq}$	2.55e+03	2.14e-01	6.36e-05	4.27e-01	1.92e-01	2.88e-01
Reduced Full $10^{-5}p_{eq}$	3.11e+03	3.24e-02	8.71e-06	6.49e-02	6.46e-02	9.57e-02
Reduced Full $10^{-6}p_{eq}$	3.67e+03	4.46e-03	1.08e-06	8.92e-03	1.13e-02	1.66e-02
Reduced Full $10^{-9}p_{eq}$	5.24e+03	9.87e-06	1.82e-09	1.97e-05	3.20e-05	4.71e-05
R-Step SSA (1000 **min.**)	1.00e+03	3.12e-01	4.03e-05	6.23e-01	3.98e-01	6.20e-01
R-Step SSA (2000 **min.**)	2.00e+03	7.82e-02	6.06e-06	1.56e-01	2.08e-01	3.13e-01
R-Step SSA (3000 **min.**)	3.00e+03	4.01e-02	2.24e-06	8.02e-02	1.60e-01	2.37e-01
R-Step SSA (4000 **min.**)	4.00e+03	1.24e-02	5.65e-07	2.47e-02	6.03e-02	8.81e-02
R-Step SSA (5000 **min.**)	5.01e+03	4.46e-03	1.75e-07	8.92e-03	2.56e-02	3.71e-02
R-Step SSA (6000 **min.**)	6.01e+03	1.11e-03	3.93e-08	2.22e-03	6.73e-03	9.72e-03
R-step Reach. (1000 min.)	1.00e+03	5.81e-01	9.87e-05	1.16e+00	4.64e-01	7.32e-01
R-step Reach. (2000 min.)	2.00e+03	1.53e-01	1.46e-05	3.07e-01	3.56e-01	5.40e-01
R-step Reach. (3000 min.)	3.00e+03	7.82e-02	5.20e-06	1.56e-01	2.73e-01	4.06e-01
R-step Reach. (4000 min.)	4.00e+03	4.81e-02	2.46e-06	9.62e-02	2.09e-01	3.08e-01
R-step Reach. (5000 min.)	5.00e+03	3.00e-02	1.25e-06	5.99e-02	1.50e-01	2.19e-01
R-step Reach. (6000 min.)	6.00e+03	2.03e-02	7.10e-07	4.06e-02	1.10e-01	1.60e-01

Table 2.9: Results for Schlögl: Full, Reduced Full, r-Step SSA, and r-Step Reachability

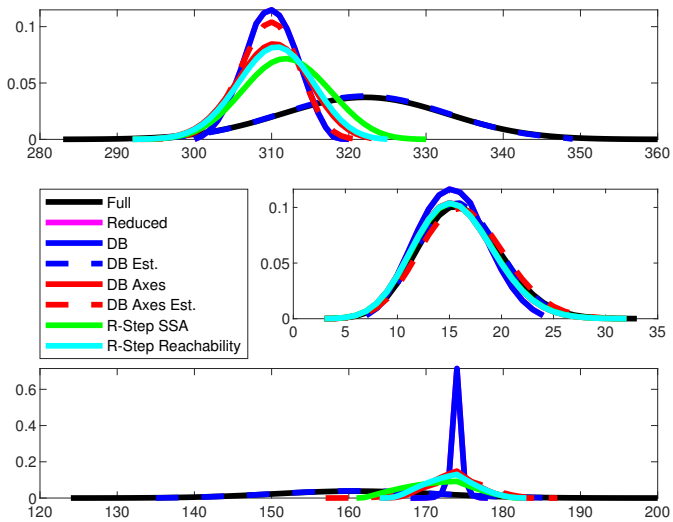

Figure 2.14: Marginals from Schlögl reaction system (top: A; middle: X; bottom: B), using the Full Solve, 10^{-1} Reduced, 10^{-1} DB, 10^{-1} DB Est., 10^{-5} DB Axes, 10^{-5} DB Axes Est., 1000-state R-Step SSA, and 1000-state R-step reachability

Method	Total States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{6}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
DB $10^{-1}p_{eq}$	8.20e+02	6.24e+01	2.66e-02	1.25e+02	5.78e-01	9.08e-01
DB Est. $10^{-1}p_{eq}$	8.20e+02	2.00e+02	8.54e-02	4.00e+02	2.16e-02	4.01e-02
DB $10^{-2}p_{eq}$	1.41e+03	1.04e+01	3.92e-03	2.07e+01	5.11e-01	8.08e-01
DB Est. $10^{-2}p_{eq}$	1.41e+03	3.22e+01	1.22e-02	6.42e+01	2.24e-03	3.88e-03
DB $10^{-3}p_{eq}$	1.98e+03	1.57e+00	5.25e-04	3.13e+00	3.89e-01	5.96e-01
DB Est. $10^{-3}p_{eq}$	1.98e+03	4.91e+00	1.65e-03	9.65e+00	3.03e-04	5.42e-04
DB $10^{-4}p_{eq}$	2.56e+03	2.14e-01	6.36e-05	4.27e-01	1.92e-01	2.88e-01
DB Est. $10^{-4}p_{eq}$	2.56e+03	8.06e-01	2.40e-04	1.44e+00	2.38e-04	3.64e-04
DB $10^{-5}p_{eq}$	3.12e+03	3.24e-02	8.72e-06	6.49e-02	6.46e-02	9.57e-02
DB Est. $10^{-5}p_{eq}$	3.12e+03	2.61e-01	7.02e-05	3.47e-01	2.37e-04	3.52e-04
DB $10^{-6}p_{eq}$	3.68e+03	4.46e-03	1.08e-06	8.92e-03	1.13e-02	1.66e-02
DB Est. $10^{-6}p_{eq}$	3.68e+03	1.85e-01	4.47e-05	1.95e-01	2.37e-04	3.51e-04
DB Axes $10^{-1}p_{eq}$	2.31e+02	6.15e+01	2.27e-02	1.23e+02	5.80e-01	9.15e-01
DB Axes Est. $10^{-1}p_{eq}$	2.31e+02	6.27e+02	2.32e-01	1.03e+03	4.87e-01	7.94e-01
DB Axes $10^{-2}p_{eq}$	4.49e+02	1.09e+01	3.30e-03	2.19e+01	5.35e-01	8.66e-01
DB Axes Est. $10^{-2}p_{eq}$	4.49e+02	3.99e+02	1.20e-01	4.72e+02	4.68e-01	7.64e-01
DB Axes $10^{-3}p_{eq}$	6.39e+02	2.96e+00	7.14e-04	5.91e+00	5.06e-01	8.11e-01
DB Axes Est. $10^{-3}p_{eq}$	6.39e+02	3.61e+02	8.72e-02	3.73e+02	4.66e-01	7.61e-01
DB Axes $10^{-4}p_{eq}$	8.34e+02	1.23e+00	2.44e-04	2.45e+00	4.89e-01	7.78e-01
DB Axes Est. $10^{-4}p_{eq}$	8.34e+02	3.56e+02	7.08e-02	3.58e+02	4.66e-01	7.61e-01
DB Axes $10^{-5}p_{eq}$	9.99e+02	7.87e-01	1.34e-04	1.57e+00	4.77e-01	7.50e-01
DB Axes Est. $10^{-5}p_{eq}$	9.99e+02	3.55e+02	6.03e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-6}p_{eq}$	1.17e+03	6.34e-01	9.07e-05	1.27e+00	4.78e-01	7.46e-01
DB Axes Est. $10^{-6}p_{eq}$	1.17e+03	3.55e+02	5.08e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-7}p_{eq}$	1.33e+03	5.43e-01	6.77e-05	1.09e+00	4.78e-01	7.44e-01
DB Axes Est. $10^{-7}p_{eq}$	1.33e+03	3.55e+02	4.43e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-8}p_{eq}$	1.49e+03	4.98e-01	5.49e-05	9.95e-01	4.79e-01	7.42e-01
DB Axes Est. $10^{-8}p_{eq}$	1.49e+03	3.55e+02	3.92e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-9}p_{eq}$	1.64e+03	4.63e-01	4.55e-05	9.25e-01	4.79e-01	7.41e-01
DB Axes Est. $10^{-9}p_{eq}$	1.64e+03	3.55e+02	3.49e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-10}p_{eq}$	1.80e+03	4.36e-01	3.85e-05	8.73e-01	4.80e-01	7.40e-01
DB Axes Est. $10^{-10}p_{eq}$	1.80e+03	3.55e+02	3.13e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-11}p_{eq}$	1.94e+03	4.20e-01	3.37e-05	8.39e-01	4.81e-01	7.40e-01
DB Axes Est. $10^{-11}p_{eq}$	1.94e+03	3.55e+02	2.85e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-12}p_{eq}$	2.09e+03	4.06e-01	2.99e-05	8.11e-01	4.81e-01	7.39e-01
DB Axes Est. $10^{-12}p_{eq}$	2.09e+03	3.55e+02	2.62e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-13}p_{eq}$	2.24e+03	3.95e-01	2.66e-05	7.89e-01	4.82e-01	7.39e-01
DB Axes Est. $10^{-13}p_{eq}$	2.24e+03	3.55e+02	2.39e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-14}p_{eq}$	2.39e+03	3.86e-01	2.40e-05	7.73e-01	4.82e-01	7.39e-01
DB Axes Est. $10^{-14}p_{eq}$	2.39e+03	3.55e+02	2.21e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-15}p_{eq}$	2.53e+03	3.81e-01	2.21e-05	7.63e-01	4.83e-01	7.39e-01
DB Axes Est. $10^{-15}p_{eq}$	2.53e+03	3.55e+02	2.06e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-16}p_{eq}$	2.68e+03	3.78e-01	2.04e-05	7.56e-01	4.83e-01	7.39e-01
DB Axes Est. $10^{-16}p_{eq}$	2.68e+03	3.55e+02	1.91e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-17}p_{eq}$	2.82e+03	3.76e-01	1.91e-05	7.52e-01	4.83e-01	7.39e-01
DB Axes Est. $10^{-17}p_{eq}$	2.82e+03	3.55e+02	1.80e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-18}p_{eq}$	2.97e+03	3.76e-01	1.78e-05	7.51e-01	4.83e-01	7.39e-01
DB Axes Est. $10^{-18}p_{eq}$	2.97e+03	3.55e+02	1.68e-02	3.55e+02	4.66e-01	7.61e-01
DB Axes $10^{-19}p_{eq}$	3.12e+03	3.75e-01	1.68e-05	7.51e-01	4.83e-01	7.39e-01
DB Axes Est. $10^{-19}p_{eq}$	3.12e+03	3.55e+02	1.59e-02	3.55e+02	4.66e-01	7.61e-01

Table 2.10: Results for Schlögl: Detailed Balance and Detailed Balance Axes

Method	Total States	State Selection Runtime (s)	Linear Solve Runtime (s)	Marginals Runtime (s)
Full	1.25e+05	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-1}p_{eq}$	8.18e+02	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-2}p_{eq}$	1.41e+03	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-3}p_{eq}$	1.98e+03	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-4}p_{eq}$	2.55e+03	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-5}p_{eq}$	3.11e+03	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-6}p_{eq}$	3.67e+03	1.23e+04	1.88e+01	1.23e+04
Reduced Full $10^{-9}p_{eq}$	5.24e+03	1.23e+04	1.88e+01	1.23e+04
R-Step SSA (1000 **min.**)	1.00e+03	1.33e+01	3.20e-03	1.33e+01
R-Step SSA (2000 **min.**)	2.00e+03	6.99e+01	7.32e-03	6.99e+01
R-Step SSA (3000 **min.**)	3.00e+03	2.02e+02	1.21e-02	2.02e+02
R-Step SSA (4000 **min.**)	4.00e+03	4.23e+02	4.11e-02	4.23e+02
R-Step SSA (5000 **min.**)	5.01e+03	7.28e+02	2.64e-02	7.28e+02
R-Step SSA (6000 **min.**)	6.01e+03	1.18e+03	3.48e-02	1.18e+03
R-step Reach. (1000 min.)	1.00e+03	8.60e-01	3.30e-03	8.65e-01
R-step Reach. (2000 min.)	2.00e+03	3.30e+00	7.86e-03	3.31e+00
R-step Reach. (3000 min.)	3.00e+03	7.29e+00	2.50e-02	7.32e+00
R-step Reach. (4000 min.)	4.00e+03	1.28e+01	1.74e-02	1.28e+01
R-step Reach. (5000 min.)	5.00e+03	1.99e+01	2.48e-02	2.00e+01
R-step Reach. (6000 min.)	6.00e+03	2.86e+01	3.25e-02	2.86e+01

Table 2.11: Results for Schlögl: Full, Reduced Full, r-Step SSA, and r-Step Reachability

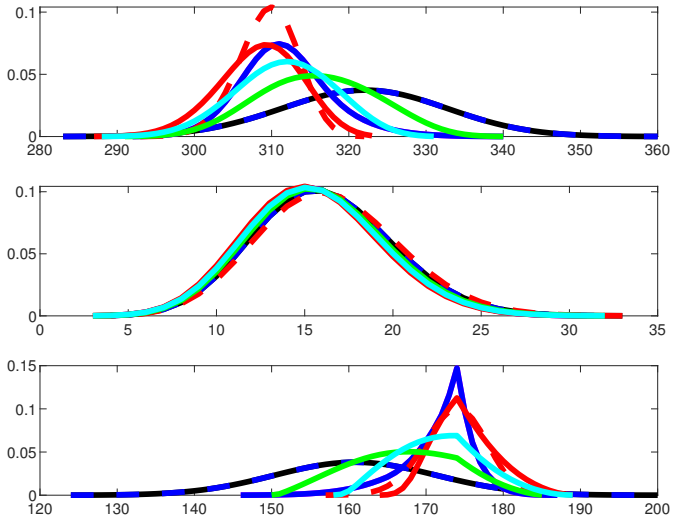

Figure 2.15: Marginals from Schlögl reaction system (top: A; middle: X; bottom: B), using the Full Solve, 10^{-3} Reduced, 10^{-3} DB, 10^{-3} DB Est., 10^{-11} DB Axes, 10^{-11} DB Axes Est., 2000-state R-Step SSA, and 2000-state R-step reachability

Method	Total States	State Selection Runtime (s)	Linear Solve Runtime (s)	Marginals Runtime (s)
DB $10^{-1}p_{eq}$	8.20e+02	1.43e+00	2.49e-03	1.43e+00
DB Est. $10^{-1}p_{eq}$	8.20e+02	1.43e+00	0.00e+00	1.43e+00
DB $10^{-2}p_{eq}$	1.41e+03	3.41e+00	4.43e-03	3.41e+00
DB Est. $10^{-2}p_{eq}$	1.41e+03	3.41e+00	0.00e+00	3.41e+00
DB $10^{-3}p_{eq}$	1.98e+03	6.45e+00	6.98e-03	6.46e+00
DB Est. $10^{-3}p_{eq}$	1.98e+03	6.45e+00	0.00e+00	6.45e+00
DB $10^{-4}p_{eq}$	2.56e+03	1.04e+01	9.98e-03	1.04e+01
DB Est. $10^{-4}p_{eq}$	2.56e+03	1.04e+01	0.00e+00	1.04e+01
DB $10^{-5}p_{eq}$	3.12e+03	1.50e+01	1.24e-02	1.50e+01
DB Est. $10^{-5}p_{eq}$	3.12e+03	1.50e+01	0.00e+00	1.50e+01
DB $10^{-6}p_{eq}$	3.68e+03	2.04e+01	3.69e-02	2.05e+01
DB Est. $10^{-6}p_{eq}$	3.68e+03	2.04e+01	0.00e+00	2.04e+01
DB Axes $10^{-1}p_{eq}$	2.31e+02	1.15e-01	7.94e-04	1.16e-01
DB Axes Est. $10^{-1}p_{eq}$	2.31e+02	1.15e-01	0.00e+00	3.37e-02
DB Axes $10^{-2}p_{eq}$	4.49e+02	2.17e-01	1.74e-03	2.19e-01
DB Axes Est. $10^{-2}p_{eq}$	4.49e+02	2.17e-01	0.00e+00	1.09e-02
DB Axes $10^{-3}p_{eq}$	6.39e+02	3.92e-01	1.99e-03	3.94e-01
DB Axes Est. $10^{-3}p_{eq}$	6.39e+02	3.92e-01	0.00e+00	8.70e-03
DB Axes $10^{-4}p_{eq}$	8.34e+02	6.52e-01	2.55e-03	6.54e-01
DB Axes Est. $10^{-4}p_{eq}$	8.34e+02	6.52e-01	0.00e+00	7.69e-03
DB Axes $10^{-5}p_{eq}$	9.99e+02	9.31e-01	3.40e-03	9.35e-01
DB Axes Est. $10^{-5}p_{eq}$	9.99e+02	9.31e-01	0.00e+00	1.08e-02
DB Axes $10^{-6}p_{eq}$	1.17e+03	1.22e+00	2.16e-02	1.25e+00
DB Axes Est. $10^{-6}p_{eq}$	1.17e+03	1.22e+00	0.00e+00	8.51e-03
DB Axes $10^{-7}p_{eq}$	1.33e+03	1.61e+00	4.79e-03	1.61e+00
DB Axes Est. $10^{-7}p_{eq}$	1.33e+03	1.61e+00	0.00e+00	3.85e-02
DB Axes $10^{-8}p_{eq}$	1.49e+03	1.95e+00	5.66e-03	1.96e+00
DB Axes Est. $10^{-8}p_{eq}$	1.49e+03	1.95e+00	0.00e+00	1.51e-02
DB Axes $10^{-9}p_{eq}$	1.64e+03	2.36e+00	5.81e-03	2.37e+00
DB Axes Est. $10^{-9}p_{eq}$	1.64e+03	2.36e+00	0.00e+00	1.25e-02
DB Axes $10^{-10}p_{eq}$	1.80e+03	2.80e+00	7.23e-03	2.81e+00
DB Axes Est. $10^{-10}p_{eq}$	1.80e+03	2.80e+00	0.00e+00	9.73e-03
DB Axes $10^{-11}p_{eq}$	1.94e+03	3.27e+00	7.72e-03	3.28e+00
DB Axes Est. $10^{-11}p_{eq}$	1.94e+03	3.27e+00	0.00e+00	1.37e-02
DB Axes $10^{-12}p_{eq}$	2.09e+03	3.73e+00	9.37e-03	3.74e+00
DB Axes Est. $10^{-12}p_{eq}$	2.09e+03	3.73e+00	0.00e+00	1.11e-02
DB Axes $10^{-13}p_{eq}$	2.24e+03	4.29e+00	8.54e-03	4.30e+00
DB Axes Est. $10^{-13}p_{eq}$	2.24e+03	4.29e+00	0.00e+00	1.02e-02
DB Axes $10^{-14}p_{eq}$	2.39e+03	4.85e+00	9.32e-03	4.86e+00
DB Axes Est. $10^{-14}p_{eq}$	2.39e+03	4.85e+00	0.00e+00	1.04e-02
DB Axes $10^{-15}p_{eq}$	2.53e+03	5.42e+00	1.12e-02	5.43e+00
DB Axes Est. $10^{-15}p_{eq}$	2.53e+03	5.42e+00	0.00e+00	1.07e-02
DB Axes $10^{-16}p_{eq}$	2.68e+03	6.05e+00	1.08e-02	6.06e+00
DB Axes Est. $10^{-16}p_{eq}$	2.68e+03	6.05e+00	0.00e+00	1.08e-02
DB Axes $10^{-17}p_{eq}$	2.82e+03	6.68e+00	1.15e-02	6.69e+00
DB Axes Est. $10^{-17}p_{eq}$	2.82e+03	6.68e+00	0.00e+00	1.15e-02
DB Axes $10^{-18}p_{eq}$	2.97e+03	7.40e+00	1.42e-02	7.42e+00
DB Axes Est. $10^{-18}p_{eq}$	2.97e+03	7.40e+00	0.00e+00	1.18e-02
DB Axes $10^{-19}p_{eq}$	3.12e+03	7.58e+00	1.30e-02	7.60e+00
DB Axes Est. $10^{-19}p_{eq}$	3.12e+03	7.58e+00	0.00e+00	1.88e-02

Table 2.12: Results for Schlögl: Detailed Balance and Detailed Balance Axes

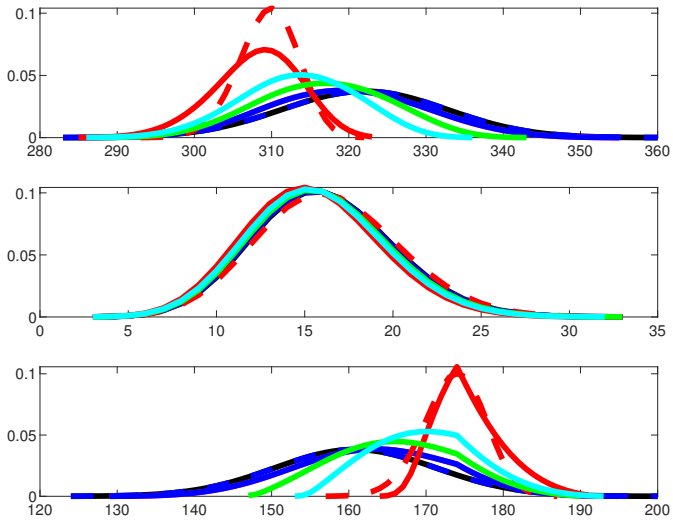

Figure 2.16: Marginals from Schlögl reaction system (top: A; middle: X; bottom: B), using the Full Solve, 10^{-5} Reduced, 10^{-5} DB, 10^{-5} DB Est., 10^{-18} DB Axes, 10^{-18} DB Axes Est., 3000-state R-Step SSA, and 3000-state R-step reachability

The r-Step SSA also struggled to handle the Schlögl system, but for a different reason. At the point $(308, 17, 174)$, reactions 1 and 2, which cause a decrease or increase to species A, respectively, both occur at rates exceeding 8000. On the other hand, reactions 3 and 4, which cause a decrease or increase to species B, respectively, both occur at a rate of about 17. Thus, the reactions causing a change to A are much faster - nearly 500 times faster - than the reactions causing a change to B. The r-Step SSA finds new states to add to the FSP by recording the first reaction that occurs in a stochastic simulation. Since reactions that cause a change to the amount of A occur so much more frequently than reactions causing a change in the amount of B, the method overwhelmingly selects states that are reached from an occurrence of one of the first two reactions. In doing so, it selects numerous unnecessary states due to large swings in the value of A (such states may be reached quickly, but if the system is prone to leave those states even more quickly, there may still be low probability that the system is found in those states in the stationary case). Likewise, the method fails to select some of the

states with larger changes to B that remain highly probable states in the stationary case. It may take the system longer to reach some of those states from the point $(308, 17, 174)$, but if the copy number of B is changing slowly, the system may linger in those states for longer, keeping those states relevant in the stationary case.

The failure of the FSPs, especially for the Reduced and Detailed Balance methods, proves especially interesting. Though the Reduced method and Detailed Balance method both chose the state space for the FSP almost perfectly, their initial performance in predicting the probability distribution is especially poor. The explanation is once again related to the speed at which reactions 1 and 2 occur. Recall that to ensure that Equation 2.3 is satisfied, we map any probability that is leaking out of the state space back onto an interior point, which we select to be the estimated equilibrium point, in this case $(308, 17, 174)$. Because the reactions causing a change in A occur so rapidly, and because the state space is sufficiently narrow, probability is constantly leaking out of the state space and being mapped back to $(308, 17, 174)$. The probability does not have enough time to migrate to some of the outer states with more distant values of B, because it leaks so quickly due to the fast A reactions and is reset to the estimated equilibrium point. (It would be interesting to attempt other strategies, such as leaving any probability leaking out of the FSP at the boundary rather than mapping it to the equilibrium point to see if such an approach might improve the probability estimation.) Again, note that it is not a contradiction that there can be states just outside the FSP that have low stationary probability but nonetheless tend to be visited frequently, because the stationary distribution describes the probability of finding a system in a certain state in the long run. Even if a state is visited often, if the system stays in that state for a very short amount of time each time it is visited, it can still have a low stationary probability.

Despite the struggles of many methods, it should be noted that the Detailed Balance Initial Estimate performed extremely well. The estimation did not suffer from

the problems that plagued other methods. It produced state spaces that were nearly identical to that of the Reduced method. (The 10^{-1} DB method produced an 820-state FSP while the 10^{-1} Reduced method produced an FSP consisting of 818 states, all of which were included in the DB FSP. The only two states included in the DB version and not the Reduced version were $(347, 10, 142)$ and $(348, 16, 135)$. The 10^{-3} DB FSP and the 10^{-3} Reduced FSP each consisted of 1983 states, with 1982 of those states being identical.) Moreover, the DB initial estimate produced probability that was also very close to that of the Full Solve.

Note that all methods improved once more states were added to the state space. As the size of the FSPs grew, all of the FSPs began to make far more accurate predictions for the probability distributions.

2.8 Application Example: p53

It is interesting to observe how the methodology we have developed performs in higher-dimensional systems. We consider the p53 reaction system detailed by [16] and implemented in [26], but with a different parameterization in Table 2.17. (The parameterization from Vo [**?**, **?**] which is detailed in Table 3.1 leads to a very large state space and will be considered in the next chapter.) This reaction network will track six species. We let $x_1 = [\mathsf{RNA}_{nuc}], x_2 = [\mathsf{RNA}_{cyt}], x_3 = [\mathsf{MDM2}_{cyt}], x_4 = [\mathsf{MDM2}_{nuc}], x_5 = [\mathsf{ARF}], x_6 = [\mathsf{p53}]$. (Note that we are not tracking the species count of $\mathsf{MDM2}_{nuc}\mathsf{ARF}$ in

	Reaction	Propensity
1	$\emptyset \rightarrow \mathsf{p53}$	k_p
2	$\mathsf{p53} \rightarrow \emptyset$	$d_p[\mathsf{p}_{53}] + k_1[\mathsf{p53}][\mathsf{MDM2}_{nuc}]$
3	$\emptyset \rightarrow \mathsf{RNA}_{nuc}$	$k_m + \frac{k_2[\mathsf{p53}]^{1.8}}{k_D^{1.8}+[\mathsf{p53}]^{1.8}}$
4	$\mathsf{RNA}_{nuc} \rightarrow \mathsf{RNA}_{cyt}$	$k_0[\mathsf{RNA}_{nuc}]$
5	$\mathsf{RNA}_{cyt} \rightarrow \emptyset$	$d_{rc}[\mathsf{RNA}_{cyt}]$
6	$\mathsf{RNA}_{cyt} \rightarrow \mathsf{RNA}_{cyt} + \mathsf{MDM2}_{cyt}$	$k_T[\mathsf{RNA}_{cyt}]$
7	$\mathsf{MDM2}_{cyt} \rightarrow \mathsf{MDM2}_{nuc}$	$k_i[\mathsf{MDM2}_{cyt}]$
8	$\mathsf{MDM2}_{nuc} + \mathsf{MDM2}_{nuc} \rightarrow \mathsf{MDM2}_{nuc}$	$d_{mn}[\mathsf{MDM2}_{nuc}][\mathsf{MDM2}_{nuc} - 1]$
9	$\mathsf{MDM2}_{nuc} + \mathsf{ARF} \rightarrow \mathsf{MDM2}_{nuc}\mathsf{ARF}$	$k_3[\mathsf{MDM2}_{nuc}][\mathsf{ARF}]$
10	$\emptyset \rightarrow \mathsf{ARF}$	k_a
11	$\mathsf{ARF} \rightarrow \emptyset$	$d_a[\mathsf{ARF}]$

Figure 2.17: p53 reactions and their propensity functions with $k_p = 0.05, k_1 = 5 \times 10^{-4}, d_p = 4.925 \times 10^{-3}, k_m = 1.5 \times 10^{-3}, k_2 = 1.5 \times 10^{-2}, k_D = 740, k_0 = 1 \times 10^{-3}, d_{rc} = 5.444 \times 10^{-4}, k_T = 3.66 \times 10^{-4}, k_i = 2.0 \times 10^{-3}, d_{mn} = 3 \times 10^{-5}, k_3 = 9.963 \times 10^{-5}, k_a = 0.05, d_a = 3.209 \times 10^{-2}$

this experiment.) Setting up Equation 1.7 yields the system of equations 3.1

$$\begin{cases} k_m + \frac{k_2 x_6^{1.8}}{k_D^{1.8}+x_6^{1.8}} - k_0 x_1 = 0 \\ k_0 x_1 - d_{rc} x_2 = 0 \\ k_T x_2 - k_i x_3 = 0 \\ k_i x_3 - d_{mn} x_4 (x_4 - 1) - k_3 x_4 x_5 = 0 \\ k_a - k_3 x_4 x_5 - d_a x_5 = 0 \\ k_p - d_p x_6 - k_1 x_4 x_6 = 0 \end{cases} \tag{2.33}$$

with corresponding Jacobian

$$J = \begin{bmatrix} -k0 & 0 & 0 & 0 & 0 & \frac{1.8 k_2 p53^{0.8}}{k_D^{1.8}+p53^{1.8}} - \frac{k2[\mathbf{p53}]^{1.8}}{(k_D^{1.8}+p53^{1.8})^2}(1.8[\mathbf{p53}]^{0.8}) \\ k_0 & -d_{rc} & 0 & 0 & 0 & 0 \\ 0 & k_T & -k_i & 0 & 0 & 0 \\ 0 & 0 & k_i & -d_{mn}(2[\mathbf{MDM2nuc}] - 1) - k_3[\mathbf{ARF}] & -k_3[\mathbf{MDM2nuc}] & 0 \\ 0 & 0 & 0 & -k_3[\mathbf{ARF}] & -k_3[\mathbf{MDM2nuc}] - d_a & 0 \\ 0 & 0 & 0 & -k_1[\mathbf{p53}] & 0 & -d_p - k_1[\mathbf{MDM2nuc}] \end{bmatrix}$$

Solving, we obtain the estimated equilibrium point $\boldsymbol{x}^T = (2, 3, 1, 4, 2, 7)$. Further-

	Total States	**State Selection Runtime (s)**	**Linear Solve Runtime (s)**	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$
DB $10^{-1}p_{eq}$	5.00e+04	7.85e+03	1.19e+03	1.38e-03	2.04e-02
DB Est. $10^{-1}p_{eq}$	5.00e+04	7.85e+03	0.00e+00	6.30e-02	9.33e-01
DB Axes Only $10^{-1}p_{eq}$	1.95e+04	6.43e+02	2.94e+01	3.64e-03	5.73e-02
DB Axes Only Est. $10^{-1}p_{eq}$	1.95e+04	6.43e+02	0.00e+00	1.23e-02	1.95e-01
5-step Reachability	2.39e+03	1.39e+01	5.17e-02	6.53e-03	6.72e-02
6-step Reachability	5.25e+03	6.17e+01	4.96e-01	4.36e-03	4.85e-02
7-step Reachability	1.03e+04	2.25e+02	3.46e+00	3.03e-03	3.56e-02
8-step Reachability	1.85e+04	6.91e+02	1.40e+01	2.14e-03	2.63e-02
R-Step SSA (19000 **min.**)	2.09e+04	1.26e+03	2.27e+01	1.18e-03	1.62e-02

Table 2.14: Results for p53

more, this equilibrium point shows evidence of stability, as shown in in Table 2.13.

Equilibrium Point	Jacobian Eigenvalues	Identification
(2,3,1,4,2,7)	-0.0325	Relative
	-0.0069	Maximum
	-0.0020	
	-0.0010	
	-0.0005	
	-0.0004	

Table 2.13: Equilibrium point analysis for p53 (smaller version)

We do not set up the Full Solve here used in other examples, partly because a six-dimensional system can require a very large state space and partly because the p53 system has very high linear solve runtimes compared with other systems we have studied with a similar number of states in the FSP. This makes some sense given that this system has 11 reactions; as a result, the $\hat{A}^{FSP}$ matrices will be less sparse than in other examples, making the linear systems more computationally expensive to solve.

However, despite not having a full solution, it is encouraging that all of the FSP

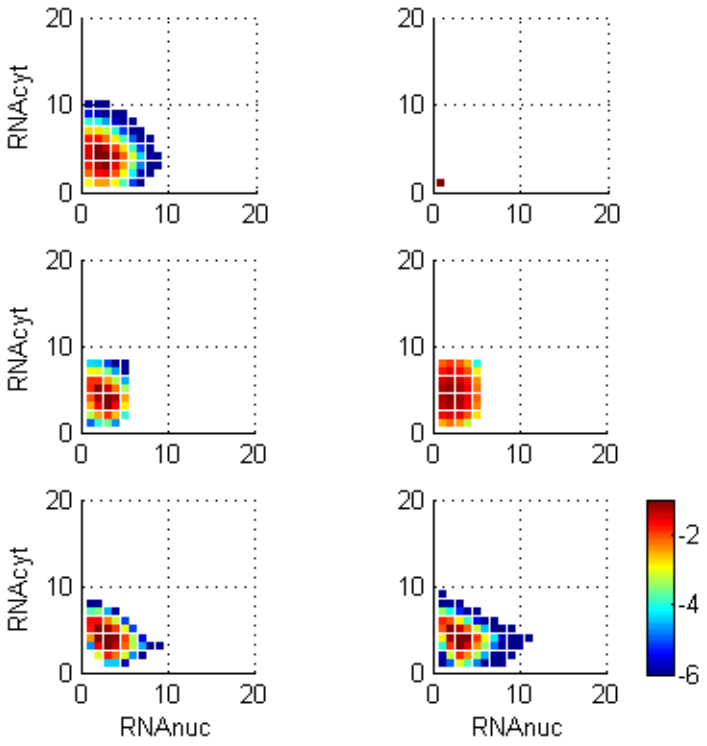

Figure 2.18: Probability distribution for $\mathbf{RNA_{nuc}}$ and $\mathbf{RNA_{cyt}}$ for the smaller p53 example, summed over all possible values of the other species. Top row: 10^{-1} DB (left), 10^{-1} DB Est (right). Second row: 10^{-1} DB Axes, 10^{-1} DB Axes Est. Third row: R-Step SSA ($19,000$ min.), R-Step Reachability (8 steps).

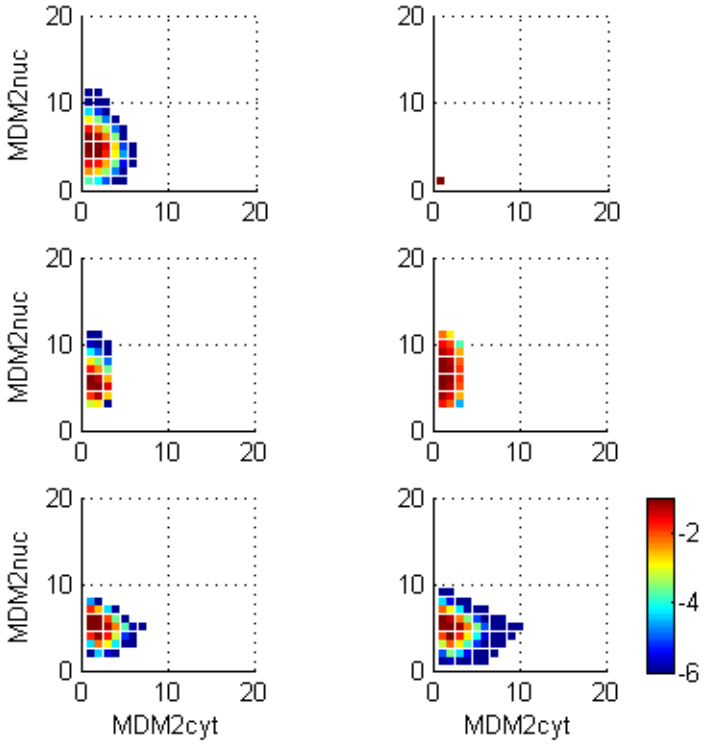

Figure 2.19: Probability distribution for $\mathbf{MDM2_{cyt}}$ and $\mathbf{MDM2_{nuc}}$ for the smaller p53 example, summed over all possible values of the other species. Top row: 10^{-1} DB (left), 10^{-1} DB Est (right). Second row: 10^{-1} DB Axes, 10^{-1} DB Axes Est. Third row: R-Step SSA ($19,000$ min.), R-step reachability (8 steps).

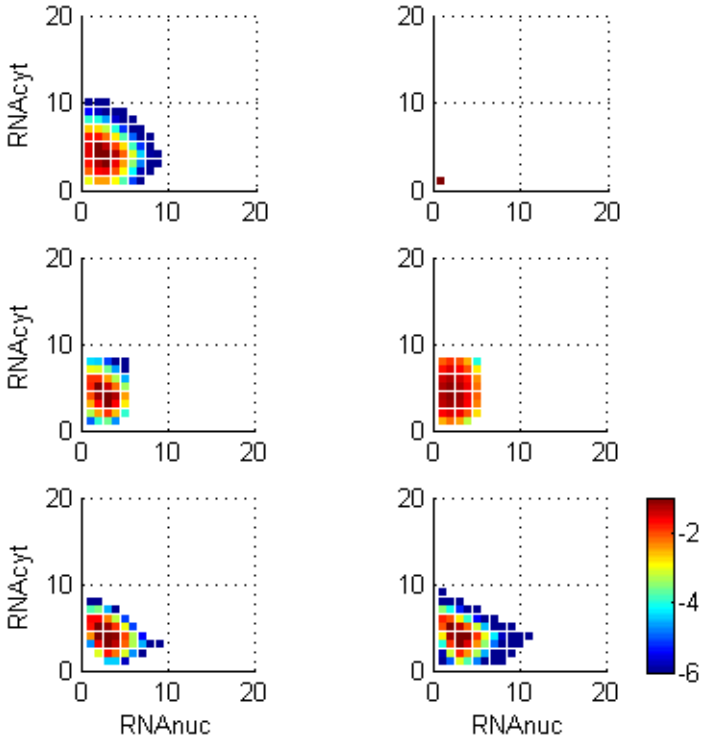

Figure 2.20: Probability distribution for **ARF** and **p53** for the smaller p53 example, summed over all possible values of the other species. Top row: 10^{-1} DB (left), 10^{-1} DB Est (right). Second row: 10^{-1} DB Axes, 10^{-1} DB Axes Est. Third row: R-Step SSA ($19,000$ min.), R-step reachability (8 steps).

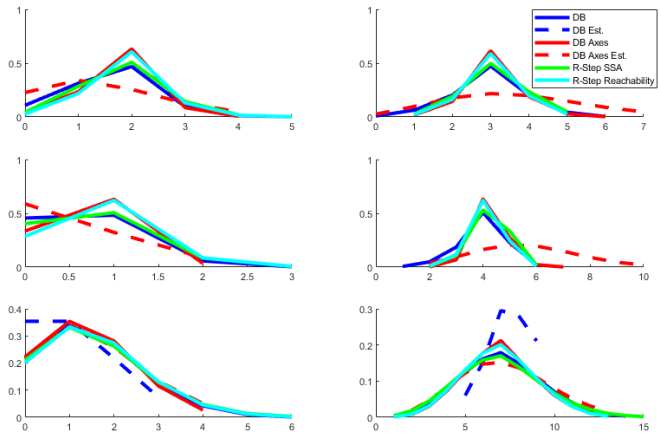

Figure 2.21: Marginals from p53 reaction system (top row: $\mathbf{RNA_{nuc}}$ (left), $\mathbf{RNA_{cyt}}(right)$; middle row: $\mathbf{MDM2_{cyt}}$, $\mathbf{MDM2_{nuc}}$; bottom row: **ARF**), **p53**)using the 10^{-1} DB, 10^{-1} DB Est., 10^{-1} DB Axes, 10^{-1} DB Axes Est., R-Step SSA (19,000 min.), and R-step reachability (8 steps).

methods seem to have similar marginal distributions. The distributions of both initial estimates (DB and DB Axes) differ from the others in several of the variables. The Detailed Balance Initial Estimate fails to perform well in general for this system (we will soon elaborate upon the reasons for this failure). One reason the DB Axes Initial Estimate may differ materially from the FSP methods have been discussed already in Section 2.7. Recalling that discussion, the DB Axes Initial Estimate may be suspect, because it fails to capture correlation between different variables. (There is evidence of correlation in the reaction rate equations; for instance, the presence of $\mathbf{RNA_{cyt}}$ is required to produce $\mathbf{MDM2_{nuc}}$ and no $\mathbf{RNA_{cyt}}$ is consumed in this reaction, so it is reasonable to assume there would be a positive correlation between these two variables.) However, it is also possible that it performs better than the FSP methods in a couple respects. The true state space is so large that it had to be truncated relatively quickly, leading the FSP methods to be biased towards states within a narrow range. Likewise, probability leaking out of this relatively small state space is mapped back to the equilibrium point. Also, because states near the equilibrium point are more likely to be included

in an FSP, there naturally tends to be a higher count of states with values at or near that of the equilibrium point included in the sum used to calculate the value of a marginal distributions at any given value, which also causes bias. Usually, the FSP methods tend to consistently overweight the amount of probability assigned to values near that of the equilibrium point. The DB Axes Initial Estimate does not suffer from the same level of susceptibility to these errors. Given all these possible sources of differences, the DB Initial Estimate actually provides some degree of confidence that the FSP methods are giving reasonable answers. It approaches the problem with a very different methodology, yet gives results that are at least in a similar range to the other methods, even if its distributions are not quite the same shape.

Of interest in this particular problem is how poorly the Detailed Balance Initial Estimate performed. (The actual Detailed Balance FSP seemed to perform reasonably well, other than taking on a very large state space, which damaged its runtime. The code capped the number of states it was allowed to select at $50,000$; otherwise, it could have been considerably larger. When allowed to take on more states, Matlab was unable to solve the linear system in on the desktop in the time it was allotted.) The poor performance of the DB Initial Estimate is due to a violation of Assumption 2.11; of particular interest is when α_4 drops from k_0 to 0 when $[\mathbf{RNA_{nuc}}]$ falls from 1 to 0. (Regardless of the size of k_0, this represents a 100% decrease in propensity for a 1-unit change in $[\mathbf{RNA_{nuc}}]$. The state $(0,3,1,4,2,7)$ is both adjacent to the state $(0,2,1,4,2,7)$ and closer to the equilibrium point than $(0,2,1,4,2,7)$. Therefore, when employing the detailed balance algorithm, when calculating the probability estimate for $(0,2,1,4,2,7)$, $\frac{\alpha_5(0,3,1,4,2,7)}{\alpha_4(0,2,1,4,2,7)} p_{est}(0,3,1,4,2,7)$ is used as an input. The algorithm detects that $\alpha_4(0,2,1,4,2,7) = 0$ and treats $(0,2,1,4,2,7)$ as the discovery of an absorbing state (or absorbing cluster of states). However, in reality, the state is not absorbing. More $\mathbf{RNA_{cyt}}$ can be produced; it is only that $\mathbf{RNA_{nuc}}$ must be produced first. However, at $(0,2,1,4,2,7)$, the state receives probability flowing from states with higher values

	Reaction	Propensity
1	$A \to 2A$	$k_1[A]$
2	$A + B \to 2B$	$k_2[A][B]$
3	$B \to \emptyset$	$k_3[B]$

Figure 2.22: Lotka-Volterra (Original) reactions and their propensity functions with $k_1 = 4, k_2 = 0.1, k_3 = 20$

of [$\mathbf{RNA_{cyt}}$], but no $\mathbf{RNA_{cyt}}$ is produced from which this probability flow can return directly. Therefore, the instantaneous drop in α_4 causes the detailed balance estimate to break down. This system also experiences a similar difficulty for the production of $\mathbf{MDM2_{nuc}}$ when [$\mathbf{MDM2_{nuc}}$] falls to 0.

2.9 Application Example: Lotka-Volterra (Original)

The reaction network framework can be applied outside the realm of biochemistry. Here, we consider the Lotka-Volterra model 2.22, which is frequently used to model predator-prey interactions. A denotes the prey (rabbits) and B denotes the predators (wolves). The system is famous for oscillatory behavior. One typical scenario that is illustrative of the system may occur as follows: a high rabbit population and low wolf population means the wolves have a large food source. As such the wolf population increases. Eventually the wolf population increases so much that too many rabbits are consumed, and the rabbit population starts to dwindle. As the rabbit population fades, the wolves lose their food source and the wolves start to die off faster than they can reproduce. As the wolf population dwindles, the rabbits begin to make a recovery and the system returns to the starting situation where the rabbit population is high and the wolf population is low. When we plot the rabbbit population on the x-axis and the wolf population on the y-axis, such a scenario would be represented by a counterclockwise rotation of the system over time around a central point.

Such a system can be modeled using the same mathematical constructions as a

Equilibrium Point	Jacobian Eigenvalues	Identification
$(200, 40)$	$0 + 8.9443i$ $0 - 8.9443i$	Unstable Spiral Point
$(0, 0)$	-20 4	Saddle Point

Table 2.15: Equilibrium point estimation for Lotka-Volterra (Original)

chemical reaction network. However, this particular model produces several difficulties that make it less than ideal for the methodology presented in this dissertation thus far.

To begin with, it can be seen that from solving the system of equations based on the reaction rates and its corresponding Jacobian

$$\begin{cases} k_1[A] - k_2[A][B] = 0 \\ k_2[A][B] - k_3[B] = 0 \end{cases}$$

$$J = \begin{bmatrix} k_1 - k_2[B] & -k_2[A] \\ k_2[B] & k_2[A] - k_3 \end{bmatrix}$$

that there are no states that register as stable equilibrium points, as seen in Table 2.15.

This presents a problem from the very beginning, since stable equilibria are meant to be the starting point for crafting an FSP under our methods. The strong circular flow of probability, as evidenced by the complex components of the eigenvalues, means that $\Phi(\boldsymbol{x})$ will get far from $\mathbf{0}$ as we get farther away from the spiral point. As such, detailed balance can be expected to perform poorly in its probability estimation. Furthermore, from observing the reaction rate equations, it can be seen that if the wolf population completely dies out, there will be unbounded growth in the rabbit population. In such a case there is no stationary solution. Finally, on the boundaries where $A = 0$ or $B = 0$, we run into a related issue that the p53 model caused when we tried to use detailed balance to estimate the relative probabilities; namely, there are situations where probability can flow into a point from one direction and there is no escape for it to return back to the

Method	Total States	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{4}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
Full	1.00e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
Reduced Full $10^{-1}p_{eq}$	1.00e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
DB $10^{-1}p_{eq}$	2.52e+04	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
DB Est. $10^{-1}p_{eq}$	2.52e+04	6.28e+02	1.34e+01	1.26e+03	2.50e-01	5.00e-01
DB Axes $10^{-1}p_{eq}$	2.40e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
DB Axes Est. $10^{-1}p_{eq}$	2.40e+03	4.00e+02	8.34e-02	8.00e+02	0.00e+00	5.00e-01
R-Step SSA (6000 **min.**)	6.06e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
R-step Reach. (6000 min.)	6.00e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00

Table 2.16: Results for Lotka-Volterra (Original)

Method	Total States	State Selection Runtime (s)	Linear Solve Runtime (s)	Marginals Runtime (s)
Full	1.00e+05	5.78e+03	8.54e-01	5.78e+03
Reduced Full $10^{-1}p_{eq}$	1.00e+05	1.16e+04	1.74e+00	1.74e+04
DB $10^{-1}p_{eq}$	2.52e+04	6.32e+02	2.91e-01	6.32e+02
DB Est. $10^{-1}p_{eq}$	2.52e+04	6.32e+02	0.00e+00	6.32e+02
DB Axes $10^{-1}p_{eq}$	2.40e+03	1.82e+01	3.89e-01	1.86e+01
DB Axes Est. $10^{-1}p_{eq}$	2.40e+03	1.82e+01	0.00e+00	4.27e-01
R-Step SSA (6000 **min.**)	6.06e+03	1.83e+02	5.39e-02	1.83e+02
R-step Reach. (6000 min.)	6.00e+03	2.17e+01	3.66e-02	2.17e+01

Table 2.17: Results for Lotka-Volterra (Original)

original state in the opposite direction. The difference here is that once the system hits a boundary (either $A = 0$ or $B = 0$), it is in actuality trapped, so modeling it as incapable of escape is correct.

Despite the considerable problems this method poses, it is still possible to glean useful information using the FSP methods. First, in the absence of stable equilibria, we must choose the points at which to begin the state selection process. Selecting the equilibrium points $(0, 0)$ and $(200, 40)$, we can implement each of our FSP methods.

Each of the FSP methods (including the "Full" Solve, which in this case, like the Gene Toggle, is technically an FSP with a very large state space, since the actual state space is countably infinite) maps all the probability to $(0, 0)$. Our Jacobian analysis had marked this point as a saddle point, because a slight perturbation to increase A would lead the system to producing more and more A without bound. However, there

is actually no reaction available at $(0,0)$ to cause this slight perturbation as some A is required to be present to produce more A. Thus, the FSP models correctly treat $(0,0)$ as an absorbing state.

We also wish to discuss the performance of the initial estimates. Note that the "error" coming from these methods mean they disagree with the "Full" Solve on some level, but in this case, there is still insight to be gleaned.

If B reaches 0 before A does, the rabbit population will increase without bound; in this case there is no stationary solution. The Detailed Balance initial estimate evidences this possibility with probability mass at its outer value of $([A], 0)$.

The Detailed Balance Initial Estimate agrees perfectly in its marginal distributions with the FSPs. However, the actual values of its $\hat{\boldsymbol{p}}$ differ. This is because, unlike the other methods, the Detailed Balance Axes method is intentionally designed to produce a marginal distribution without prior knowledge of what the method produces for $\hat{\boldsymbol{p}}$. The portion of the distribution calculated from $(0,0)$, of course, remains trapped at $(0,0)$. The portion of the marginal distribution for $[A]$ coming from $(200,40)$ is calculated outward from $(200,40)$ along the $[B] = 40$ axis. It recognizes $(0,40)$ as absorbing all the probability along this axis and registers all the marginal probability for the $[A]$ direction occurring at $[A] = 0$. Likewise, the portion of the marginal distribution for $[B]$ coming from $(200,40)$ is calculated outward from $(200,40)$ along the $[A] = 200$ axis. It recognizes $(200,0)$ as absorbing all the probability along this axis and registers all the marginal probability for the $[B]$ direction occurring at $[B] = 0$. Thus, the marginal distribution agrees with those of the FSP. However, when calculating $\hat{\boldsymbol{p}}$, the algorithm takes the probability from the axes points as input and uses them to estimate all the other probabilities in the state space. Therefore, since they were all inputs, the three points $(0,0)$, $(200,0)$, and $(0,40)$ are all given positive probabilities in $\hat{\boldsymbol{p}}$. (They end up being the only states with positive probability in the estimation.) This is the reason, therefore, for the probability discrepancy between the estimate and the FSP methods.

	Reaction	Propensity
1	$A \to 2A$	$k_1[A]$
2	$A + B \to 2B$	$k_2[A][B]$
3	$B \to \emptyset$	$k_3[B]$
4	$\emptyset \to A$	k_4
5	$A \to \emptyset$	$k_5[A]$
6	$\emptyset \to B$	k_6

Figure 2.23: Lotka-Volterra with Migration reactions and their propensity functions with $k_1 = 4, k_2 = 0.1, k_3 = 20, k_4 = 160, k_5 = 2, k_6 = 80$

2.10 Application Example: Lotka-Volterra with Migration

We examine a similar system in Figure 2.23 that avoids the population number becoming trapped at 0 for either species and the rabbit population growing unchecked by introducing three additional migration reactions: migration of rabbits into the area being monitored (Reaction 4), migration of rabbits out of the area being monitored (Reaction 5), and migration of wolves into the area being monitored (Reaction 6).

This reaction networks leads to the following system of equations to approximate an equilibrium point:

$$\begin{cases} k_1[A] - k_2[A][B] + k_4 - k_5[A] = 0 \\ k_2[A][B] - k_3[B] + k_6 = 0 \end{cases}$$

with corresponding Jacobian:

$$J = \begin{bmatrix} k_1 - k_2[B] - k_5 & -k_2[A] \\ k_2[B] & k_2[A] - k_3 \end{bmatrix}$$

This time, the equilibrium at $(0, 0)$ is removed and only one equilibrium point is found. We detect an equilibrium point that is barely stable, with strong circulation around it. Due to this circulation, the system is a poor candidate for the regular detailed balance methodology. However, we will see that the DB Axes method performs extremely

Equilibrium Point	Jacobian Eigenvalues	Identification
$(173, 29)$	$-1.8000 + 7.0257i$ $-1.8000 - 7.0257i$	Stable Spiral Point

Table 2.18: Equilibrium point estimation for Lotka-Volterra with Migration

Method	**Total States**	$\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1$	$\frac{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}\|\|_1}{\|\|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}_{unif}\|\|_1}$	$\|\|\boldsymbol{A}\sigma(\hat{\boldsymbol{p}})\|\|_1$	$\frac{1}{4}\sum_{i=1}^{2}\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1$	$\frac{1}{2}\|\boldsymbol{p} - \sigma(\hat{\boldsymbol{p}})\|_1$
Full	1.00e+05	8.03e-06	1.76e-07	8.03e-06	0.00e+00	0.00e+00
Reduced Full $10^{-1}p_{eq}$	6.93e+03	8.01e-01	1.90e-02	1.60e+00	1.50e-01	2.30e-01
DB $10^{-1}p_{eq}$	2.50e+04	1.13e-01	2.84e-03	2.26e-01	3.22e-02	4.90e-02
DB Est. $10^{-1}p_{eq}$	2.50e+04	5.35e+02	1.34e+01	9.75e+02	9.73e-01	9.98e-01
DB Axes $10^{-1}p_{eq}$	6.24e+03	9.27e-01	2.15e-02	1.85e+00	1.74e-01	2.70e-01
DB Axes Est. $10^{-1}p_{eq}$	6.24e+03	1.07e+01	2.50e-01	1.99e+01	8.29e-02	1.77e-01
R-Step SSA (6000 **min.**)	6.02e+03	1.36e+00	2.18e-02	2.71e+00	2.09e-01	3.43e-01
R-step Reach. (6000 min.)	6.00e+03	1.51e+00	3.26e-02	3.02e+00	2.39e-01	3.76e-01

Table 2.19: Results for Lotka-Volterra with Migration

well. The other FSP methods, R-Step SSA and R-Step Reachability, also yield good results.

From the results in Table 2.19 and the graphs, we see that the Detailed Balance Axes Initial Estimate provided the best estimator of both the overall probability distribution and the marginal distributions (excepting the Detailed Balance FSP, which

Method	**Total States**	**State Selection Runtime (s)**	**Linear Solve Runtime (s)**	**Marginals Runtime (s)**
Full	1.00e+05	1.16e+04	4.14e+01	1.17e+04
Reduced Full $10^{-1}p_{eq}$	6.93e+03	1.17e+04	4.16e+01	2.34e+04
DB $10^{-1}p_{eq}$	2.50e+04	1.34e+03	1.58e+00	1.34e+03
DB Est. $10^{-1}p_{eq}$	2.50e+04	1.34e+03	0.00e+00	1.34e+03
DB Axes $10^{-1}p_{eq}$	6.24e+03	4.82e+01	1.27e-01	4.83e+01
DB Axes Est. $10^{-1}p_{eq}$	6.24e+03	4.82e+01	0.00e+00	2.92e-02
R-Step SSA (6000 **min.**)	6.02e+03	2.23e+02	1.06e-01	2.23e+02
R-step Reach. (6000 min.)	6.00e+03	4.26e+01	1.16e-01	4.27e+01

Table 2.20: Results for Lotka-Volterra with Migration

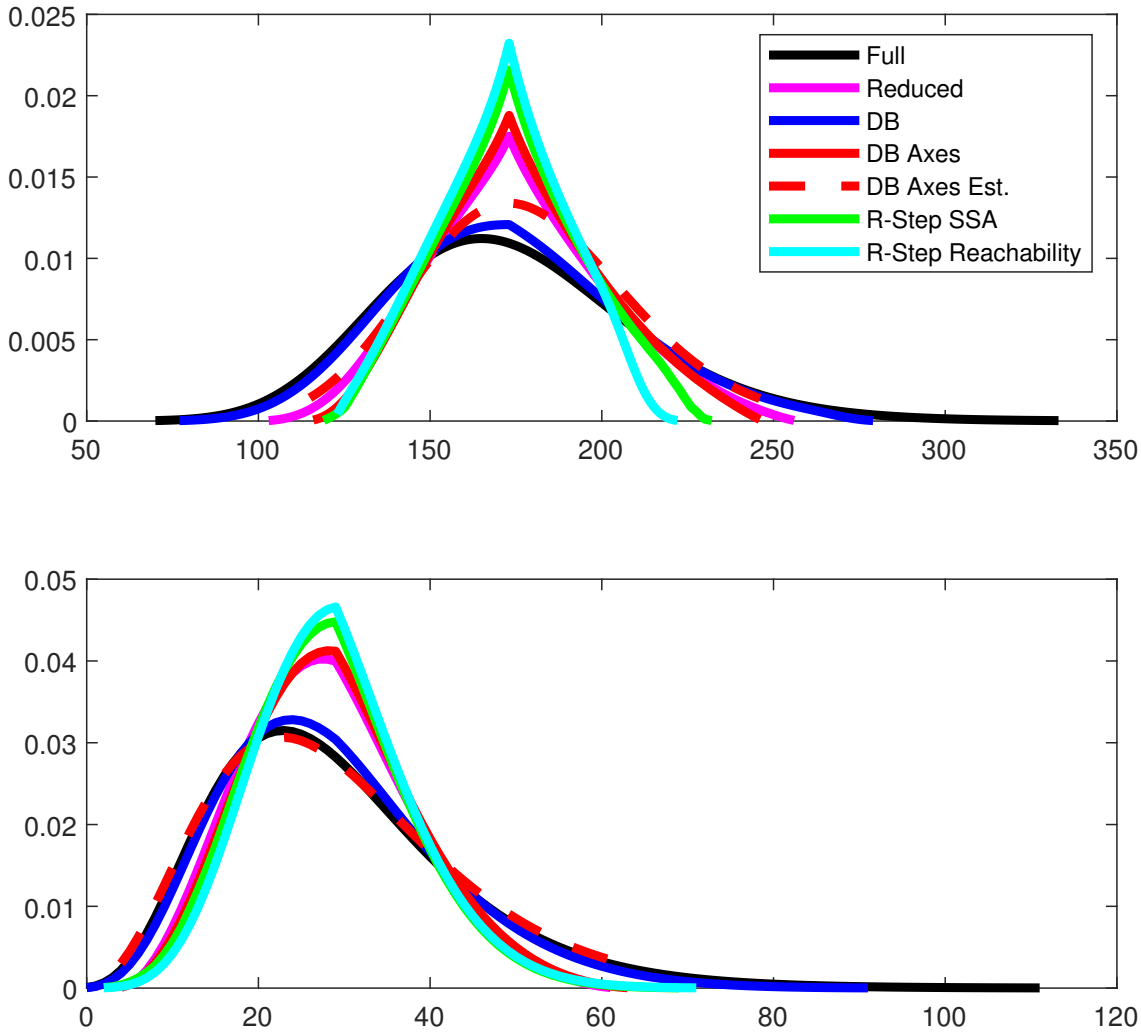

Figure 2.24: Marginal Distributions for species A (upper graph) and species B (lower graph) for the Lotka-Volterra with Migration system. DB Est. not included.

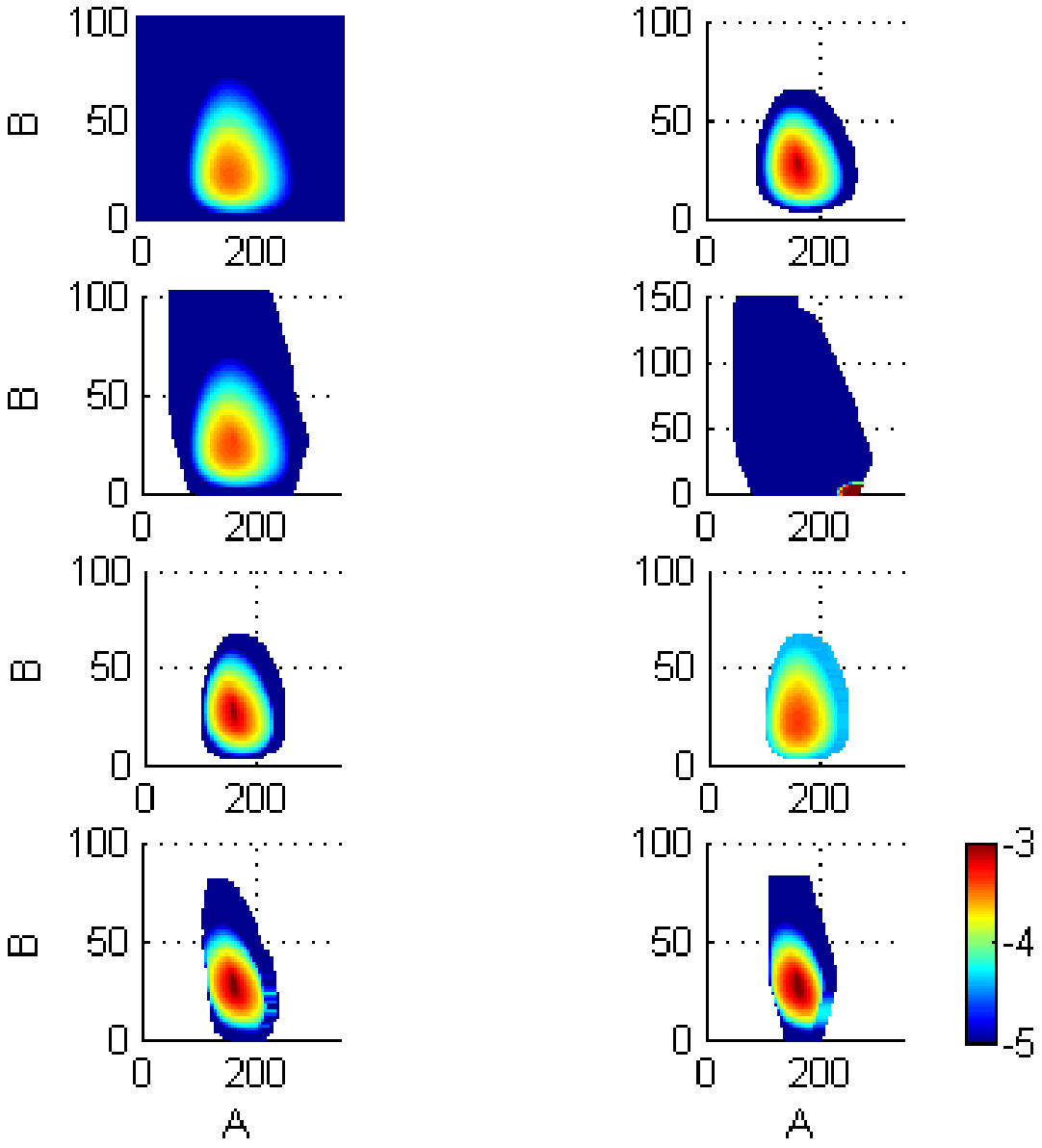

Figure 2.25: Probability distributions for Lotka-Volterra with Migration. Top Row (left to right): Full Solve, Reduced Solve. Second Row: DB, DB Est. Third Row: DB Axes, DB Axes Est. Fourth Row: R-Step SSA, R-Step Reachability.

performed well due to including a very high number of states). There are a number of factors that led to the very strong performance of the DB Axes Initial Estimate.

The FSP methods tended to have cusps in the marginal distributions at or near the equilibrium values, $A = 173, B = 29$, because of the way in which states were selected for inclusion in the FSP. By design, states closer to the equilibrium point are more often selected for the FSP; as a result, a disproportionately high number of states with values near $A = 173$ and/or $B = 29$ are included and a disproportionately high number of states further away from these values are left out, leading to overstated marginal probabilities near equilibrium (and a cusp at or near the values for which the highest number of states are included) and understated marginal probabilities at the tails. Furthermore, the strong counterclockwise (from the point of view of Figure 2.25) flux further away from the equilibrium point means that much of the probability density that flows into states far away from equilibrium comes from states far away from equilibrium; FSP models that cut out some of the more distant states remove a major source of probability for the more distant states that remain in the FSP. Furthermore, any probability that leaks out of the state space is mapped back to the equilibrium point.

Note that the Detailed Balance initial estimate performs very poorly in this experiment.

By analyzing the F_i's from Equation 2.13 we can better understand why this system is so difficult for the first Detailed Balance method to model. Let F_1 represent the approximate flux in the direction of A and F_2 represent the approximate flux in the direction of B. We further break down each F_i into $F_i = F_{i+} - F_{i-}$ with

$$F_{i+} = \rho(\boldsymbol{x}) \sum_{\nu_{ik}>0} \alpha_k(\boldsymbol{x})\nu_{ik} \tag{2.34}$$

$$F_{i-} = \rho(\boldsymbol{x}) \sum_{\nu_{ik}<0} \alpha_k(\boldsymbol{x})\|\nu_{ik}\| \tag{2.35}$$

Here, we substitute the values of $p(\boldsymbol{x})$ calculated in the Full Solve for $\rho(\boldsymbol{x})$.

	$\boldsymbol{x}$	$\sum_{\nu_{1k}>0} \alpha_k(\boldsymbol{x})\nu_{1k}$	$\sum_{\nu_{1k}<0} \alpha_k(\boldsymbol{x})\lVert\nu_{1k}\rVert$	$\sum_{\nu_{2k}>0} \alpha_k(\boldsymbol{x})\nu_{2k}$	$\sum_{\nu_{2k}<0} \alpha_k(\boldsymbol{x})\lVert\nu_{2k}\rVert$
(173,29)	3.11e-04	8.52e+02	8.48e+02	5.82e+02	5.80e+02
(168,29)	3.19e-04	8.32e+02	8.23e+02	5.67e+02	5.80e+02
(163,29)	3.21e-04	8.12e+02	7.99e+02	5.53e+02	5.80e+02
(158,29)	3.17e-04	7.92e+02	7.74e+02	5.38e+02	5.80e+02
(153,29)	3.05e-04	7.72e+02	7.50e+02	5.24e+02	5.80e+02
(148,29)	2.87e-04	7.52e+02	7.25e+02	5.09e+02	5.80e+02
(173,24)	3.45e-04	8.52e+02	7.61e+02	4.95e+02	4.80e+02
(173,19)	3.28e-04	8.52e+02	6.75e+02	4.09e+02	3.80e+02
(173,14)	2.43e-04	8.52e+02	5.88e+02	3.22e+02	2.80e+02
(173,9)	1.17e-04	8.52e+02	5.02e+02	2.36e+02	1.80e+02
(173,4)	2.21e-05	8.52e+02	4.15e+02	1.49e+02	8.00e+01
(178,29)	2.97e-04	8.72e+02	8.72e+02	5.96e+02	5.80e+02
(183,29)	2.79e-04	8.92e+02	8.97e+02	6.11e+02	5.80e+02
(188,29)	2.58e-04	9.12e+02	9.21e+02	6.25e+02	5.80e+02
(193,29)	2.35e-04	9.32e+02	9.46e+02	6.40e+02	5.80e+02
(198,29)	2.10e-04	9.52e+02	9.70e+02	6.54e+02	5.80e+02
(173,34)	2.51e-04	8.52e+02	9.34e+02	6.68e+02	6.80e+02
(173,39)	1.88e-04	8.52e+02	1.02e+03	7.55e+02	7.80e+02
(173,44)	1.33e-04	8.52e+02	1.11e+03	8.41e+02	8.80e+02
(173,49)	9.00e-05	8.52e+02	1.19e+03	9.28e+02	9.80e+02
(173,54)	5.91e-05	8.52e+02	1.28e+03	1.01e+03	1.08e+03

Table 2.21: Data for calculating estimates for the F_i's for Lotka-Volterra with Migration

$\frac{F_{ccw}}{F_{cw}}$	$\boldsymbol{x}$	F_{1+}	F_{1-}	F_{2+}	F_{2-}	F_1	F_2	$\frac{F_{out}}{F_{in}}$
(173,29)	2.65e-01	2.63e-01	1.81e-01	1.80e-01	1.34e-03	5.28e-04	-	-
(168,29)	2.66e-01	2.63e-01	1.81e-01	1.85e-01	2.81e-03	-4.08e-03	9.89e-01	1.02e+00
(163,29)	2.61e-01	2.57e-01	1.78e-01	1.86e-01	4.27e-03	-8.77e-03	9.84e-01	1.05e+00
(158,29)	2.51e-01	2.45e-01	1.70e-01	1.84e-01	5.63e-03	-1.32e-02	9.78e-01	1.08e+00
(153,29)	2.35e-01	2.29e-01	1.60e-01	1.77e-01	6.80e-03	-1.72e-02	9.71e-01	1.11e+00
(148,29)	2.15e-01	2.08e-01	1.46e-01	1.66e-01	7.68e-03	-2.03e-02	9.64e-01	1.14e+00
(173,24)	2.94e-01	2.63e-01	1.71e-01	1.66e-01	3.14e-02	5.25e-03	9.69e-01	1.12e+00
(173,19)	2.79e-01	2.21e-01	1.34e-01	1.25e-01	5.81e-02	9.41e-03	9.30e-01	1.26e+00
(173,14)	2.07e-01	1.43e-01	7.83e-02	6.81e-02	6.41e-02	1.03e-02	8.69e-01	1.45e+00
(173,9)	9.97e-02	5.87e-02	2.76e-02	2.11e-02	4.10e-02	6.52e-03	7.64e-01	1.70e+00
(173,4)	1.88e-02	9.16e-03	3.29e-03	1.76e-03	9.63e-03	1.53e-03	5.36e-01	2.05e+00
(178,29)	2.59e-01	2.59e-01	1.77e-01	1.72e-01	-5.94e-05	4.81e-03	1.00e+00	1.03e+00
(183,29)	2.49e-01	2.50e-01	1.70e-01	1.62e-01	-1.31e-03	8.56e-03	9.95e-01	1.05e+00
(188,29)	2.35e-01	2.37e-01	1.61e-01	1.49e-01	-2.37e-03	1.16e-02	9.90e-01	1.08e+00
(193,29)	2.19e-01	2.22e-01	1.50e-01	1.36e-01	-3.21e-03	1.40e-02	9.86e-01	1.10e+00
(198,29)	2.00e-01	2.04e-01	1.38e-01	1.22e-01	-3.83e-03	1.56e-02	9.81e-01	1.13e+00
(173,34)	2.14e-01	2.35e-01	1.68e-01	1.71e-01	-2.06e-02	-2.96e-03	9.83e-01	1.10e+00
(173,39)	1.60e-01	1.92e-01	1.42e-01	1.46e-01	-3.17e-02	-4.75e-03	9.68e-01	1.20e+00
(173,44)	1.13e-01	1.47e-01	1.12e-01	1.17e-01	-3.39e-02	-5.15e-03	9.56e-01	1.30e+00
(173,49)	7.67e-02	1.07e-01	8.35e-02	8.82e-02	-3.08e-02	-4.71e-03	9.47e-01	1.40e+00
(173,54)	5.04e-02	7.57e-02	6.00e-02	6.39e-02	-2.53e-02	-3.89e-03	9.39e-01	1.50e+00

Table 2.22: Estimating radial flux and rotational flux for Lotka-Volterra with Migration

Based on Equation 2.14, we know that each $F_i = 0$ and therefore $F_{i+} = F_{i-}$ and $\frac{F_{i+}}{F_{i-}} = 1$ at equilibrium. The actual values for $(173, 29)$ differ slightly, because $(173, 29)$ is actually the nearest allowable state (since each value is a whole number), rather than the actual point that satisfies Equation 2.14 perfectly.

Furthermore, we have from Equation 2.11 that since the α's change slowly, $\frac{F_{i+}}{F_{i-}} \approx 1$ near the equilibrium point. However, as shown by the Jacobian, the negative real component has a relatively small magnitude, leading to a relatively large state space for the FSP, since the probabilities of the states surrounding $(173, 29)$ are falling relatively slowly. However, the magnitude of the complex component is comparatively large, meaning that the rotational force acting on the probability mass surrounding the equilibrium point tends to be strong relative to the force pulling that probability mass towards the equilibrium point.

This is manifest in values of $\frac{F_{i+}}{F_{i-}}$ that are far from 1 despite the fact that $p(\boldsymbol{x})$ is not changing quickly between adjacent states. This means that $\Phi(\boldsymbol{x}) \approx \mathbf{0}$ no longer holds and that detailed balance is no longer a valid approximation as we get farther away from the equilibrium point. Since $p(\boldsymbol{x})$ is not changing quickly, many states must be included in the FSP; even though the α's change slowly, they are given enough space for the rotational force to alter them enough that detailed balance becomes a poor assumption for the outer states.

However, it is the detailed balance calculations that are in a rotational direction that cause the problems. Table 2.22 demonstrates how $\frac{F_{ccw}}{F_{cw}}$ (where *ccw* signifies counterclockwise and *cw* means clockwise) increases as the points get farther from the equilibrium point. However this doesn't mean that $\Phi(\boldsymbol{x}) \approx \mathbf{0}$ is necessarily violated in the calculations going in an inward or outward path from the equilibrium point.

In fact, another reason that the DB Axes initial estimate performs so well is that this $\Phi(\boldsymbol{x}) \approx \mathbf{0}$ assumption is violated in a primarily counterclockwise direction around the equilibrium point. Because the DB Axes Initial Estimate is computed in straight

line segments emanating from the equilibrium point, the detailed balance estimations are evaluated in a direction that is roughly perpendicular to the flux at any given point, so the estimations are not subject to the same impairment as the regular Detailed Balance Initial Estimate. In the case of the marginal distribution for species B, the DB Axes Initial Estimate is nearly identical to the Full solve.

Therefore, overall, all the methods except the regular Detailed Balance Initial Estimate performed reasonably well. Despite the poor performance of its initial estimate, the Detailed Balance FSP also attained high accuracy by including a large number of states. (It would have had more states had the code not capped the number of allowable states at about $25,000$.)

3

APPLICATION TO TENSOR-BASED APPROACHES

Even after choosing an efficient FSP, we are sometimes left with a linear system that is too large to solve using traditional means.

To observe one such system, we return to the p53 reaction system detailed by [16] and use a parameterization implemented in [26], as displayed in Table 3.1. Once again letting $x_1 = [\mathsf{RNA}_{nuc}], x_2 = [\mathsf{RNA}_{cyt}], x_3 = [\mathsf{MDM2}_{cyt}], x_4 = [\mathsf{MDM2}_{nuc}], x_5 = [\mathsf{ARF}], x_6 = [\mathsf{p53}]$. (Once more, we are not tracking the species count of $\mathsf{MDM2}_{nuc}\mathsf{ARF}$ in this experiment.) Setting up the reaction rate-based system of equations yields the following system of equations:

$$\begin{cases} k_m + \frac{k_2 x_6^{1.8}}{k_D^{1.8} + x_6^{1.8}} - k_0 x_1 = 0 \\ k_0 x_1 - d_{rc} x_2 = 0 \\ k_T x_2 - k_i x_3 = 0 \\ k_i x_3 - d_{mn} x_4 (x_4 - 1) - k_3 x_4 x_5 = 0 \\ k_a - k_3 x_4 x_5 - d_a x_5 = 0 \\ k_p - d_p x_6 - k_1 x_4 x_6 = 0 \end{cases} \tag{3.1}$$

Under this parameterization, solving the nonlinear system (3.1) and rounding each component to the nearest whole number (since each represents a species copy number), we now obtain $x^T = (5, 30, 548, 152, 323, 326)$.

	Reaction	Propensity
1	$\emptyset \rightarrow \mathsf{p53}$	k_p
2	$\mathsf{p53} \rightarrow \emptyset$	$d_p[\mathsf{p}_{53}] + k_1[\mathsf{p53}][\mathsf{MDM2}_{nuc}]$
3	$\emptyset \rightarrow \mathsf{RNA}_{nuc}$	$k_m + \frac{k_2[\mathsf{p53}]^{1.8}}{k_D^{1.8}+[\mathsf{p53}]^{1.8}}$
4	$\mathsf{RNA}_{nuc} \rightarrow \mathsf{RNA}_{cyt}$	$k_0[\mathsf{RNA}_{nuc}]$
5	$\mathsf{RNA}_{cyt} \rightarrow \emptyset$	$d_{rc}[\mathsf{RNA}_{cyt}]$
6	$\mathsf{RNA}_{cyt} \rightarrow \mathsf{RNA}_{cyt} + \mathsf{MDM2}_{cyt}$	$k_T[\mathsf{RNA}_{cyt}]$
7	$\mathsf{MDM2}_{cyt} \rightarrow \mathsf{MDM2}_{nuc}$	$k_i[\mathsf{MDM2}_{cyt}]$
8	$\mathsf{MDM2}_{nuc} + \mathsf{MDM2}_{nuc} \rightarrow \mathsf{MDM2}_{nuc}$	$d_{mn}[\mathsf{MDM2}_{nuc}][\mathsf{MDM2}_{nuc} - 1]$
9	$\mathsf{MDM2}_{nuc} + \mathsf{ARF} \rightarrow \mathsf{MDM2}_{nuc}\mathsf{ARF}$	$k_3[\mathsf{MDM2}_{nuc}][\mathsf{ARF}]$
10	$\emptyset \rightarrow \mathsf{ARF}$	k_a
11	$\mathsf{ARF} \rightarrow \emptyset$	$d_a[\mathsf{ARF}]$

Figure 3.1: p53 reactions and their propensity functions with $k_p = 0.5, k_1 = 9.963 \times 10^{-6}, d_p = 1.925 \times 10^{-5}, k_m = 1.5 \times 10^{-3}, k_2 = 1.5 \times 10^{-2}, k_D = 740, k_0 = 8.0 \times 10^{-4}, d_{rc} = 1.444 \times 10^{-4}, k_T = 1.66 \times 10^{-2}, k_i = 9.0 \times 10^{-4}, d_{mn} = 1.66 \times 10^{-7}, k_3 = 9.963 \times 10^{-6}, k_a = 0.5, d_a = 3.209 \times 10^{-5}$

Equilibrium Point	Jacobian Eigenvalues	Identification
(5,30,548,152,323,326)	-0.0048	Spiral
	-0.0015	Point
	-9.7434e-04 + 3.5490e-04i	
	-9.7434e-04 - 3.5490e-04i	
	3.4762e-05 + 3.1291e-04i	
	3.4762e-05 - 3.1291e-04i	

Table 3.1: Equilibrium point analysis for p53 (larger version)

We note that while the largest-magnitude eigenvalues are negative, there are smaller-magnitude eigenvalues with complex components and/or positive (albeit small) real components. Therefore, there is evidence of some instability around this equilibrium point, which makes analyzing the probability distribution surrounding it more challenging.

Nevertheless, we use the Detailed Balance Axes Initial Estimate method to estimate the marginal distributions, which we can in turn use to approximate the size of a relevant FSP by implementing the following algorithm, which is closely related to the Detailed Balance Axes algorithm described in detail in the Appendix.

Algorithm 3 Detailed Balance Axes - FSP Size Approximation

Input:

pMarg: the estimated marginal distributions from the Detailed Balance Axes method (each marginal distribution, pMarg_i, has a domain consisting of a collection $\boldsymbol{x}_i$ of potential copy numbers for the ith species and a range consisting of the associated marginal probabilities.

thresholdFactor: minimum estimated probability for a state to be allowed in the FSP, relative to the maximum estimated probability

Procedure:

: listSize = 0;

: list = [];

for i=1:6

: $\text{pmax}_i = \max(\text{pMarg}_i)$;

end for

: threshold = thresholdFactor $\times \prod_{i=1}^{6} pmax_i$;

for i1 = $\min(\boldsymbol{x}_1)$:$\max(\boldsymbol{x}_1)$

for i2 = $\min(\boldsymbol{x}_2)$:$\max(\boldsymbol{x}_2)$

for i3 = $\min(\boldsymbol{x}_3)$:$\max(\boldsymbol{x}_3)$

for i4 = $\min(\boldsymbol{x}_4)$:$\max(\boldsymbol{x}_4)$

for i5 = $\min(\boldsymbol{x}_5)$:$\max(\boldsymbol{x}_5)$

for i6 = $\min(\boldsymbol{x}_6)$:$\max(\boldsymbol{x}_6)$

if $\text{pMarg}_1(\text{i1}) \times \text{pMarg}_2(\text{i2}) \times \text{pMarg}_3(\text{i3}) \times \text{pMarg}_4(\text{i4}) \times$

$\text{pMarg}_5(\text{i5}) \times \text{pMarg}_6(\text{i6}) \geq \text{threshold}$

: listSize=listSize+1;

: list(listSize) = [i1,i2,i3,i4,i5,i6];

end if

end for

end for

end for

end for

end for

end for

Output:

list: list of states in the FSP

listSize: number of states in the FSP

Taking minp=10^{-3} for the main Detailed Balance Axes algorithm to compute the marginals, then using thresholdFactor = 10^{-2} to approximate the size for an FSP and omitting the step that actually records the list of states, implementing this algorithm in Matlab on a laptop took about 77 minutes of runtime. It estimates an FSP containing about 4.57 billion states. In actuality, an appropriate FSP would likely be much larger, due to the fact that the Detailed Balance FSP often handles strongly correlated species poorly and can thereby leave out numerous important states. In fact, there is evidence of this happening in the results that follow; the transient solver results show marginal distributions that are much broader than that of the DB Axes initial approximation for $\mathbf{MDM2_{cyt}}$, $\mathbf{MDM2_{cyt}}$, **ARF**, and **p53**.

Therefore, even using an FSP that chooses the state space as efficiently as possible would be extremely computationally expensive using the linear system methods we have explored so far. Such a problem is a good candidate to model using tensor-based

approaches, since such methods are well-suited to handle high-dimensional systems.

To observe the behavior of this p53 system, We implement the transient solver from [26], with initial conditions stating that all of the probability mass begins in the $(0, 0, 0, 0, 0, 0)$ state ("Original Transient Solver"). Given sufficient time, the transient solution will converge to the stationary solution. In fact, recalling Equation 1.2 one way the transient solution can be defined is

$$p(\boldsymbol{x}) = \lim_{t\to\infty} P(\boldsymbol{x}, t) = \lim_{t\to\infty} e^{tA}\boldsymbol{p}_0$$

However, methods that compute transient solutions may require a long time to approach equilibrium if they start off far away from it. Therefore, we implement a couple techniques based on the some of the ideas discussed earlier in this dissertation in the hopes of finding more useful initial conditions (i.e. initial conditions that are closer to the true stationary solution). To this end, we first run the transient solver with initial conditions stating that all of the probability mass begins at the estimated equilibrium point (referred to as the "Equilibrium Point Transient Solver"). Finally, we approximate the marginal distributions using the Detailed Balance Axes Initial Estimate in an attempt to set $\boldsymbol{p}_0$ in a way that is close enough to the stationary solution. Since $\boldsymbol{p}_0$ would require a full state space, we run the transient solver with initial conditions stating that the initial probability is spread throughout numerous states according to these marginal distributions ("Detailed Balance Axes Transient Solver"). We also provide this initial approximation to the marginal distributions from the Detailed Balance Axes calculations for comparison ("Detailed Balance Axes Initial Estimate"). We mark the changes in each distribution after 1 hour of simulated time to mark how well each method seems to be converging. As a metric of how readily each version of the transient solver appears to be converging, we use the following formula to track the change in the marginal distributions

	Original Transient Solver	Equilibrium Point Transient Solver	Detailed Balance Axes Transient Solver	Detailed Balance Axes Initial Estimate
Setup Runtime	N/A (no additional setup required)	< 1 second	< 1 second	< 1 second
$err_j < 10^{-2}$	N/A (never reaches $err_j < 10^{-2}$)	9 iterations 661 seconds	1 iteration 47 seconds	N/A
$err_j < 10^{-3}$	N/A (never reaches $err_j < 10^{-3}$)	45 iterations 1579 seconds	93 iterations 5148 seconds	N/A
err_j (final)	1.45×10^{-2}	4.59×10^{-4}	2.23×10^{-4}	N/A

Figure 3.2: p53 transient solver convergence results. Setup runtime shows that estimation of the relative maximum and the initial marginal distributions can be calculated almost instantly. $err_j < 10^{-2}$ and $err_j < 10^{-3}$ detail the iteration and amount of runtime required to fall beneath the given error threshold. err_j (final) indicates the error at the end of the simulation.

at the jth iteration:

$$err_j = \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{p}_i(t_{j+1}) - \boldsymbol{p}_i(t_j)||_1 \tag{3.2}$$

where t_j represents the time modeled by the jth iteration of the transient solver. Our numerical experiments (which were performed in Matlab on a laptop) show that estimating a relative maximum using the reaction rate equations and further approximating the relative probability of the points surrounding that maximum through the DB Initial Estimate provides more effective initial conditions for the simulation, and allows low-error solutions to be reached. As seen in Figure 3.3 for the p53 model, the Detailed Balance Axes Transient Solver yielded fatter tails in its marginal distributions than the other methods and its peaks were sometimes shifted compared to the Detailed Balance Axes Initial Estimation, especially in the case of $\mathbf{MDM2_{nuc}}$. These findings are consistent with the simplifying assumptions made by the other methods: the other transient solvers have all the initial probability mass being concentrated in a single state, and the Detailed Balance Axes Initial Estimate only updates the reaction rate equations for one species at a time while leaving the others fixed to achieve its marginal distribution

approximation almost instantly. However, it does not take into account correlation between variables. The peaks for the Detailed Balance Axes Initial Estimate are too sharp, but the approximation is useful given that, in a runtime of less than 1 second, they give a reasonable first approximation to a massive 6-dimensional system where even a finite state projection would be massive.

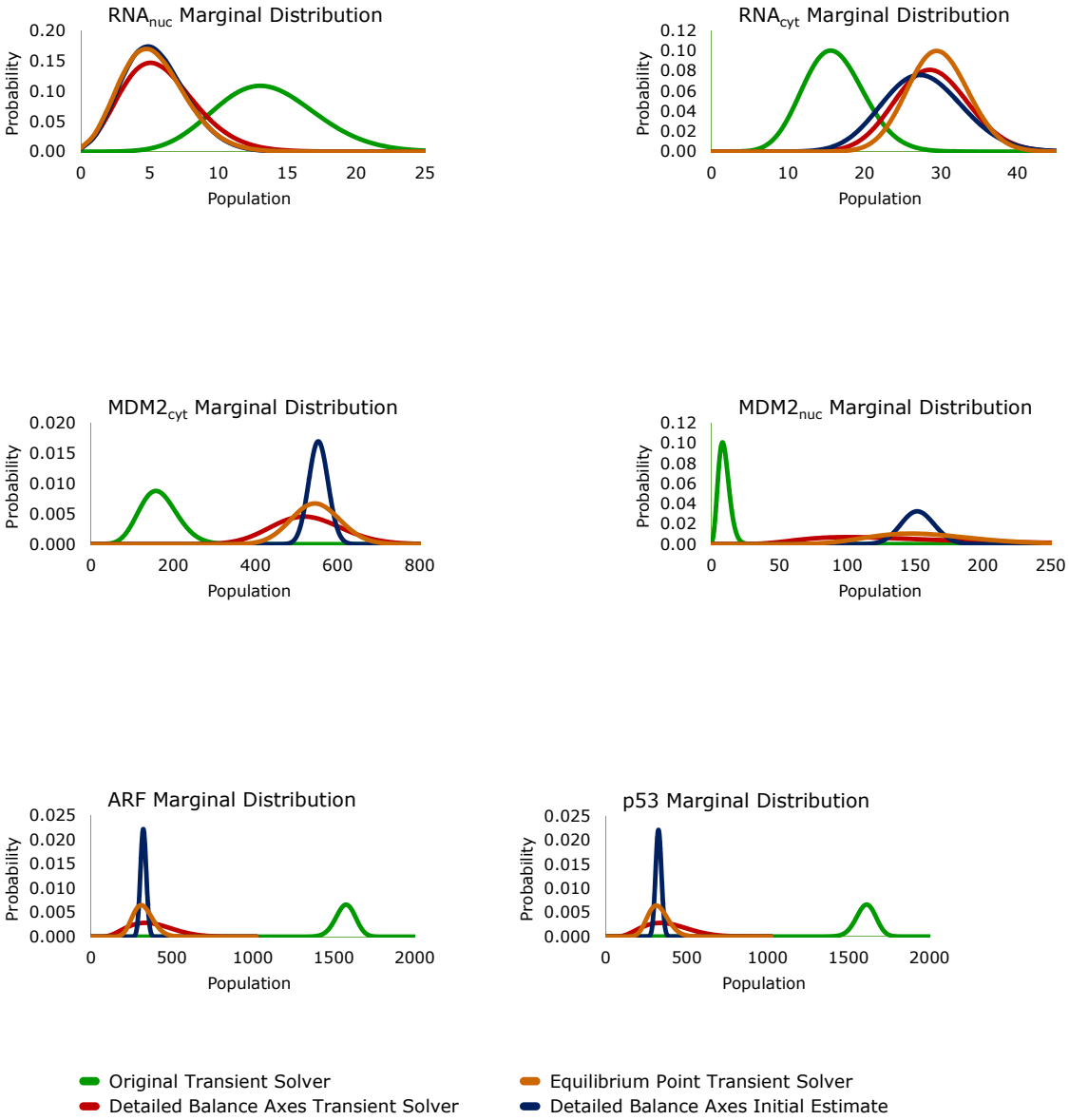

Figure 3.3: Results for the p53 model

4

CONCLUSIONS AND FUTURE DIRECTIONS

Using reaction rate equations, we can identify possible stable equilibria very quickly. After obtaining them, there are numerous techniques to truncate the sample space to a more manageable size while retaining the most probable states. Detailed balance approximation can be a very efficient way to identify states to include in the FSP and, in some cases, even to directly estimate the relative probability of the states surrounding each mode. Monte-Carlo simulation can also be gainfully applied to help identify the most relevant states. By solving the reduced linear system, we can find results that have meaningful precision and accuracy, while greatly improving computational efficiency.

When some of these techniques were applied to a tensor-based algorithm, both the Equilibrium Point Transient Solver and the Detailed Balance Axes Transient Solver considerably outperformed the original transient solver. The DB Axes method seemed very promising on a couple levels, especially because it makes a marginal distribution approximation far faster than any of the other methods. Its primary weakness is that it does not capture correlation between different variables. If techniques are developed to overcome this weakness, the method could become even more powerful. Likewise, the R-Step SSA seemed to perform very well, except when confronted with systems that contained a mixture of fast and slow reactions. If this weakness could be mitigated, it also could become more useful. Furthermore, it would be interesting to apply this research to more tensor-based problems. This research seems to indicate the methods discussed here (specifically the Equilibrium Point and DB Axes analyses), when properly applied, may be able to provide even more efficiency to tensor-based methods by supplying helpful

initial conditions. It would be interesting to see more experiments with other tensor-based algorithms to put these methods to the test.

REFERENCES

[1] Y. Cao, D. Gillespie, and L. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395–411, 2005.

[2] S. Dolgov and B. Khoromskij. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numerical Linear Algebra with Applications*, 22(2):197–219, 2015.

[3] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*, volume 1, page 214. 1981.

[4] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403 – 434, 1976.

[5] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[6] D. T. Gillespie. The chemical langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[7] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[8] A. Gupta, J. Mikelson, and M. Khammash. A finite state projection algorithm for the stationary solution of the chemical master equation. *The Journal of Chemical Physics*, 147(15):154101, 2017.

[9] M. Hegland and J. Garcke. On the numerical solution of the chemical master equation with sums of rank one tensors. *ANZIAM Journal*, 52:628–643, 2011.

[10] D. J. Higham. Modeling and simulating chemical reactions. *SIAM Review*, 50(2):347–368, 2008.

[11] K. A. Johnson and R. S. Goody. The original michaelis constant: Translation of the 1913 michaelis–menten paper. *Biochemistry*, 50(39):8264–8269, 2011. PMID: 21888353.

[12] V. Kazeev, M. Khammash, M. Nip, and C. Schwab. Direct solution of the chemical master equation using quantized tensor trains. *PLoS Computational Biology*, 10(3):e1003359, 2014.

[13] V. Kazeev and C. Schwab. Tensor approximation of stationary distributions of chemical reaction networks. *SIAM Journal on Matrix Analysis and Applications*, 36(3):1221–1247, 2015.

[14] B. N. Khoromskij. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 110(1):1–19, 2012.

[15] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[16] G. Leenders and J. Tuszynski. Stochastic and deterministic models of cellular p53 regulation. *Frontiers in Oncology*, 3:64, 2013.

[17] B. Melykuti, J. P. Hespanha, and M. Khammash. Equilibrium distributions of simple biochemical reaction systems for time-scale separation in stochastic reaction networks. *Journal of The Royal Society Interface*, 11(97):20140054–20140054, 2014.

[18] L. Michaelis and M. Menten. Die kinetik der invertinwirkung biochem z 49: 333–369. *Find this article online*, 1913.

[19] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124(4):044104, 2006.

[20] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[21] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use svd in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, 2009.

[22] J. Puchałka and A. M. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86(3):1357–1372, 2004.

[23] R. Sidje and H. Vo. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. *Mathematical Biosciences*, 269:10–16, 2015.

[24] P. Sjöberg, P. Lötstedt, and J. Elf. Fokker–planck approximation of the master equation in molecular biology. *Computing and Visualization in Science*, 12(1):37–50, 2007.

[25] K. Tomita and H. Tomita. Irreversible circulation of fluctuation. *Progress of Theoretical Physics*, 51(6):1731–1749, 1974.

[26] H. D. Vo and R. B. Sidje. An adaptive solution to the chemical master equation using tensors. *The Journal of Chemical Physics*, 147(4):044102, 2017.

APPENDIX: FSP SELECTION ALGORITHMS

Included here are the detailed versions of the algorithms used to generate the state space under the R-Step Reachability, Traditional SSA, R-Step SSA, Reduced Full, Detailed Balance, and Detailed Balance Axes methods. In the case of Detailed Balance and Detailed Balance Axes, the initial estimate is included in the same pseudocode. In the case of Detailed Balance Axes, the initial marginal distribution approximation is included.

There are additional functions whose code is not included explicitly. However, several such functions are mentioned below.

The ReachableStates function finds all states that can be reached from the input state given the stoichiometry of the system (and, when necessary, the values of the propensity functions leading to each of those states from the current state) and the IsInList function evaluates whether or not the state has already been recorded in the FSP list.

The functions **UpFlow** and **DownFlow** give vector output as follows:

$$\mathbf{UpFlow}_i(\boldsymbol{x}) = \sum_{\nu_{ik}>0} \alpha(\boldsymbol{x})\nu_{ik}$$

$$\mathbf{DownFlow}_i(\boldsymbol{x}) = \sum_{\nu_{ik}<0} \alpha(\boldsymbol{x})|\nu_{ik}|$$

The function GetIndex finds the location of an item in a list, The function MaxAbsorbingState finds the maximum value of the variable absorbingState over all items in a list with the same modeNumber.

Algorithm 4 R-step Reachability

Input:

v0: the stable equilibria

R: the maximum allowed step number

cutoffNumber: the cutoff number of states

Procedure:

```
: listSize = length(v0);
: oldListSize = 0;
: step=1;
: list=v0;
while oldListSize < listSize AND step ≤ R
    : oldListSizeCopy = oldListSize;
    : oldListSize = listSize;
    : listSizeCopy = listSize;
    for k=oldListSizeCopy+1:listSizeCopy
        : rs = ReachableStates(list(k));
        : len = length(rs);
        for i=1:len
            if IsInList(list,rs(i))==False AND listSize < cutoffNumber
                : listSize = listSize+1;
                : list(listSize) = rs(i);
            end if
        end for
    end for
    : step = step+1;
end while
```

Output:

list: list of states in the FSP

Algorithm 5 Traditional SSA

Input:

v0: the stable equilibria

tf: amount of simulated time in each SSA run

numRuns: the number of SSA runs

Procedure:

```
: listSize = length(v0);
: modeCount = listSize;
: list=v0;
for j=1:(numRuns/modeCount)
    for k=1:modeCount
      : initialState = v0(k);
      : finalState = SSA(initialState, tf);
       if IsInList(list,finalState)==False
         : listSize = listSize+1;
         : list(listSize) = finalState;
       end if
    end for
end for
```

Output:

list: list of states in the FSP

Algorithm 6 R-step SSA

Input:

v0: the stable equilibria

R: the maximum allowed step number

cutoffNumber: the cutoff number of states

Procedure:

```
: listSize = length(v0);
: step=1;
: list=v0;
while listSize < cutoffNumber AND step ≤ R
    : listSizeCopy = listSize;
    for k=1:listSizeCopy
        : [rs, props] = ReachableStates(list(k));
        : propSum = sum(props);
        : randSelect = rand(0,1);
        : accumulatedProps = 0;
        : len = length(rs);
        for i=1:len
            : newState = rs(i);
            : accumulatedProps = accumulatedProps + props(i)/propSum;
            if randSelect ≤ accumulatedProps
                break
            end if
        end for
        if IsInList(list,rs(i))==False
            : listSize = listSize+1;
```

: list(listSize) = rs(i);

end if

end for

: step = step+1;

end while

Output:

list: list of states in the FSP

Algorithm 7 Reduced Full Solve

Input:

v0: the stable equilibria

R: the maximum allowed step number

cutoffNumber: the cutoff number of states

fullp: the probability vector calculated for the entire state space; i.e., the $\boldsymbol{p}$ from $A\boldsymbol{p} = \mathbf{0}$, where A is the transition rate matrix for the whole state space

fullList: full list of states for the entire state space, where the stationary probability of the ith state is equal to the ith entry of fullp

pointOfOrigin: for each state in fullList, this variable lists the state in v0 that could reach the state in fullList the most quickly, in the sense of fewest steps of reachability

upperBounds: a vector consisting of the maximum allowable values of each species for the entire state space

minp: minimum probability, as a ratio of the probability of the nearest equilibrium point, for the FSP

Procedure:

: listSize = length(v0);

: oldListSize = 0;

```
: step=1;
: list=v0;
for i=1:length(fullList)
    for j=1:listSize
        if pointOfOrigin(i)==v0(j)
            : modeNumber(i) = j;
        end if
    end for
end for
for i=1:length(fullList)
    if fullp(i) ≥ minp × fullp(modeNumber(i))
      : listSize = listSize + 1;
      : list(listSize) = fullList(i);
    end if
end for
```

Output:

list: list of states in the FSP

Algorithm 8 Detailed Balance

Input:

v0: the stable equilibriar

cutoffNumber: the cutoff number of states

N: number of chemical species in the system

minp: minimum probability, as a ratio of the probability of the nearest equilibrium point, for the FSP

Procedure:

```
: listSize = length(v0);
: oldListSize = 0;
: list=v0;
for i=1:listSize
    : modeNumber(i) = i;
    : absorbingState(i) = 0;
    : pEst(i) = 1;
end for
while oldListSize < listSize
    : oldListSizeCopy = oldListSize;
    : oldListSize = listSize;
    : listSizeCopy = listSize;
    : candidateList = [];
    : candidateListSize = 0;
    : candidateAbsorbingState = [];
    : candidateModeNumber = [];
     for k=oldListSizeCopy+1:listSizeCopy
        : rs = ReachableStates(list(k));
        : len = length(rs);
        : upFlowOld = UpFlow(list(k));
        : downFlowOld = downFlow(list(k));
         for i=1:len
            : upFlowNew = UpFlow(rs(i));
            : downFlowNew = DownFlow(rs(i));
             if IsInList(candidateList,rs(i))==False AND IsInList(list,rs(i))==False AND
              listSize < cutoffNumber
               : candidateListSize = candidateListSize+1;
```

```
: candidateList(candidateListSize) = rs(i);
: candidateAbsorbingState(candidateListSize) = absorbingState(k);
: candidateModeNumber(candidateListSize) = modeNumber(k);
 for j=1:N
    if (rs(i,j) > list(k,j))
       if downNew(x)==0
          : candidateAbsorbingState(candidateListSize) = absorbingState(k)+1;
       else
          : candidateAbsorbingState(candidateListSize) = absorbingState(k);
          : candidateNumerator(candidateListSize) = pEst(k)^2× upOld(x)/downNew(x);
          : candidateDenominator(candidateListSize) = pEst(k);
       end if
    else if (rs(i,j) < list(k,j))
       if upNew(x)==0
          : candidateAbsorbingState(candidateListSize) = absorbingState(k)+1;
       else
          : candidateAbsorbingState(candidateListSize) = absorbingState(k);
          : candidateNumerator(candidateListSize) = pEst(k)^2× downOld(x)/upNew(x);
          : candidateDenominator(candidateListSize) = pEst(k);
       end if
    end if
 end for
else if IsInList(candidateList,rs(i))==True
 : index=GetIndex(candidateList,rs(i));
 for j=1:N
    if (rs(i,j) > list(k,j))
       if downNew(x)==0
```

```
        : candidateAbsorbingState(index) =
          max(candidateAbsorbingState(index), absorbingState(k)+1);
      else
        : candidateAbsorbingState(index) =
          max(candidateAbsorbingState(index), absorbingState(k));
        : candidateNumerator(index) =
          candidateNumerator(index) + pEst(k)^2× upOld(x)/downNew(x);
        : candidateDenominator(index) =
          candidateDenominator(index) + pEst(k);
      end if
    else if (rs(i,j) < list(k,j))
      if upNew(x)==0
        : candidateAbsorbingState(index) =
          max(candidateAbsorbingState(index), absorbingState(k)+1);
      else
        : candidateAbsorbingState(index) =
          max(candidateAbsorbingState(index), absorbingState(k));
        : candidateNumerator(index) =
          candidateNumerator(index) + pEst(k)^2× downOld(x)/upNew(x);
        : candidateDenominator(index) =
          candidateDenominator(index) + pEst(k);
      end if
    end if
  end for
end if
end for
end for
```

```
    for k=1:candidateListSize
        if candidateDenominator(k) > 0
            : probEst = candidateNumerator(k)/candidateDenominator(k);
             if probEst > minp
                : listSize = listSize+1;
                : list(listSize) = candidateList(k);
                : pEst(listSize) = probEst;
                : absorbingState(listSize) = candidateAbsorbingState(k);
                : modeNumber(listSize) = candidateModeNumber(k);
        else
            : listSize = listSize+1;
            : list(listSize) = candidateList(k);
            : pEst(listSize) = 1;
            : absorbingState(listSize) = candidateAbsorbingState(k);
            : modeNumber(listSize) = candidateModeNumber(k);
        end if
    end for
end while
for k=1:listSize
    if absorbingState(k) < MaxAbsorbingState(modeNumber(k))
        : pEst(k)=0;
    end if
end for
: pEst = pEst/Sum(pEst);
```

Output:

list: list of states in the FSP

pEst: initial estimate of the probability for all states in the FSP

Algorithm 9 Detailed Balance Axes

Input:

v0: the stable equilibria

cutoffNumber: the cutoff number of states

N: number of chemical species in the system

minp: minimum probability, as a ratio of the probability of the nearest equilibrium point, for the FSP

directions: N-dimensional vector indicating "False" for every species listed as a dependent variable

eqns: matrix to represent equations for how much the dependent variables must move in response to changes in the independent variables to account for any conservation relationships

skip: vector representing minimum step size over all possible reactions for each species: $\text{skip}_i = \min_{k=1}^{M} |\nu_{ik}|$

Procedure:

: modeCount=length(v0);

for i=1:modeCount

: pointOfOrigin(i,1) = v0(i);

: modeNumber(i,1) = i;

: absorbingState(i,1) = 0;

: pEst(i,1) = 1;

: listSize(i) = 1;

: oldListSize(i) = 0;

: list(i,1)=v0(i);

```
    for x=1:N
      : pMarg_x(i,v0(i,x)) = 1;
    end for
end for
for h=1:modeCount
   : pointOfOrigin(h,1) = v0(h);
    for x=1:N
      : pLast = pEst(h,1);
      : absorbingStateTracker = 0;
       if directions(x)==True
         : newState = v0(h);
         : probEst = pEst(h,1);
          while pLast ≥ minp
            : oldState = newState;
             for y=1:N
                if x==y
                   newState(x) = oldState(x) + skip(x);
                else if directions(y)==True
                   newState(y) = oldState(y);
                else
                   newState(y) = oldState(y) + eqns(y,x) × skip(x);
                end if
               : upFlowOld = UpFlow(oldState);
               : downFlowNew = DownFlow(newState);
                if downFlowNew(x)==0 AND upFlowOld(x) > 0 AND
                  IsAllowableState(newState)==True
                    : listSize(h) = listSize(h)+1;
```

```
        : list(h,listSize(h)) = newState;
        : probEst = 1;
        : pEst(h,listSize(h)) = probEst;
        : absorbingStateTracker = absorbingStateTracker+1;
        : absorbingState(h,listSize(h)) = absorbingStateTracker;
        : pointOfOrigin(h,listSize(h)) = v0(h);
        : modeNumber(h,listSize(h)) = h;
        : pMarg_x(h,newState(x)) = probEst;
      else if upFlowOld(x)>0
        : probEst = probEst × upFlowOld(x)/downFlowNew(x);
         if probEst ≥ minp
           : listSize(h) = listSize(h)+1;
           : list(h,listSize(h)) = newState;
           : pEst(h,listSize(h)) = probEst;
           : absorbingState(h,listSize(h)) = absorbingStateTracker;
           : pointOfOrigin(h,listSize(h)) = v0(h);
           : modeNumber(h,listSize(h)) = h;
           : pMarg_x(h,newState(x)) = probEst;
         end if
      else
        : probEst=0;
      end if
     : pLast = probEst;
    end for
  while
 end if
end for
```

```
for x=1:N
  : pLast = pEst(h,1);
  : absorbingStateTracker = 0;
   if directions(x)==True
     : newState = v0(h);
     : probEst = pEst(h,1);
      while pLast ≥ minp
        : oldState = newState;
         for y=1:N
            if x==y
               newState(x) = oldState(x) - skip(x);
            else if directions(y)==True
               newState(y) = oldState(y);
            else
               newState(y) = oldState(y) - eqns(y,x)*skip(x);
            end if
           : downFlowOld = DownFlow(oldState);
           : upFlowNew = UpFlow(newState);
            if upFlowNew(x)==0 AND IsAllowableState(newState)==True
              : listSize(h) = listSize(h)+1;
              : list(h,listSize(h)) = newState;
              : probEst = 1;
              : pEst(h,listSize(h)) = probEst;
              : absorbingStateTracker = absorbingStateTracker+1;
              : absorbingState(h,listSize(h)) = absorbingStateTracker;
              : pointOfOrigin(h,listSize(h)) = v0(h);
              : modeNumber(h,listSize(h)) = h;
```

```
        : pMarg_x(h,newState(x)) = probEst;
      else
        : probEst = probEst × downFlowOld(x)/upFlowNew(x);
         if probEst ≥ minp
           : listSize(h) = listSize(h)+1;
           : list(h,listSize(h)) = newState;
           : pEst(h,listSize(h)) = probEst;
           : absorbingState(h,listSize(h)) = absorbingStateTracker;
           : pointOfOrigin(h,listSize(h)) = v0(h);
           : modeNumber(h,listSize(h)) = h;
           : pMarg_x(h,newState(x)) = probEst;
         end if
      end if
     : pLast = probEst;
    end for
  while
 end if
end for
for k=1:listSize(h)
 if absorbingState(h,k) < MaxAbsorbingState(modeNumber(k))
   : pEst(h,k)=0;
   for x=1:N
     : pMarg_x(h,list(h,k,x)) = 0;
   end for
 end if
end for
for x=1:N
```

```
        if directions(x)==False
            for j=0:max_{k=1}^{listSize(h)}(list(h,k,x))
                : pMarg_x(h,j) = 0;
                : margNumerator = 0;
                : margDenominator = 0;
                for k=1:listSize(h)
                    if list(h,k,x)==j
                        : margNumerator = margNumerator + pEst(k);
                        : margDenominator = margDenominator + 1;
                        : pMarg_x(h,j) = margNumerator/margDenominator;
                    end if
                end for
        end if
        : pMarg_x(h) = pMarg_x(h)/Sum(pMarg_x(h));
    end for
    : axesList(h) = list(h);
end for
: pMarg_x = (�sum_{h=1}^{modeCount} pMarg_x(h)) / modeCount;
for h=1:modeCount
    while oldListSize(h) < listSize(h)
        : oldListSizeCopy(h) = oldListSize(h);
        : oldListSize(h) = listSize(h);
        : listSizeCopy(h) = listSize(h);
        for k=oldListSizeCopy(h)+1:listSizeCopy(h)
            : rs = ReachableStates(list(h,k));
            : len = length(rs);
            for i=1:len
```

```
if IsInList(list(h),rs(i))==False AND listSize(h) < cutoffNumber
  : probEst=1;
   for x=1:N
      if directions(x)==True
         for y=1:N
            if x==y
               : axisPoint(y) = rs(i,x);
            else if directions(y)==True
               : axisPoint(y) = pointOfOrigin(h,k,y);
            else
               : axisPoint(y) = pointOfOrigin(h,k,y)
                 + eqns(y,x) × (rs(i,x) - pointOfOrigin(h,k,x));
            end if
         end for
         if IsInList(axesList(h), axisPoint)==True
            : index = GetIndex(axesList(h), axisPoint);
            : probEst = probEst*pEst(h,index);
         end if
      end if
   end for
   if probEst ≥ minp
      : listSize(h) = listSize(h)+1;
      : list(h,listSize(h)) = rs(i);
      : pointOfOrigin(h,listSize(h)) = pointOfOrigin(h,k);
      : pEst(h,listSize(h)) = probEst;
   end if
if
```

```
            end for
        for
    end while
   : pEst(h) = pEst(h)/Sum(pEst(h));
end for
: wholeList = [];
: wholeListSize = 0;
for h=1:modeCount
    for i=1:listSize(h)
      : state = list(h,i);
        if IsInList(wholeList, state)==False
          : wholeListSize = wholeListSize + 1;
          : wholeList(wholeListSize) = state;
          : wholePEst(wholeListSize) = pEst(h,i)/modeCount;
        else
          : index=GetIndex(wholeList,state);
          : wholePEst(index) = wholePEst(index) + pEst(h,i)/modeCount;
        end if
    end for
end for
: list = wholeList;
: pEst = wholePEst;
```

Output:

list: list of states in the FSP

pEst: initial estimate of the probability for all states in the FSP

pMarg: marginal probability distribution estimates for each species