A Computational Method for Investigating Bifurcations in Oscillatory Biochemical Reaction Networks

Noah Trebesch[1] and Jorge Viñals[2]

[1]*School of Physics and Astronomy and Department of Computer Science and Engineering*
[2]*School of Physics and Astronomy and Minnesota Supercomputing Institute*
University of Minnesota - Twin Cities, Minneapolis, MN 55455

May 13, 2014

A computational method for numerically investigating the design principles behind robust synthetic gene oscillators is presented in this thesis. This method was build specifically to allow for rigorous examination of the bifurcation points and transience that govern the behavior of many gene oscillators. As a proof of concept, this method was tested on the two Schlögl models, whose deterministic and stochastic numerical behavior is already well known. With this method, new insight into the physical principles that govern the behavior of gene oscillators can be found. It represents the first step in developing a method which may be used to create robust synthetic gene oscillators which output ideal, tunable signals. The source code for this method along with instructions on how to compile and use it are publicly available at <https://code.google.com/p/rxn-sys-sim/>.

1 Introduction

Oscillatory electrical signals are often used to drive complicated behavior in electrical circuits. For example, all computers rely on an internal clock signal to perform their functions [1]. Bioengineers would like to create analogous circuitry out of biological components to perform similarly complicated tasks in biological systems [2]. To do this, they need to have a robust system which can produce an ideal oscillatory signal with a tunable period and amplitude [2]. In nature, there are many examples of gene regulatory networks in which the concentration of one or more proteins that participate in the network fluctuates periodically throughout time [3]. (A gene regulatory network is a network of genes whose expression regulates the expression of other genes within the network [4]. These networks will be discussed in more detail later.)

The problem with these natural systems is that they are very complicated, which makes them difficult to study and tune. Their output is also not necessarily ideal, so bioengineers have turned to developing their own synthetic oscillatory gene networks in the hopes of developing gene oscillators with the properties they seek [2]. Many attempts to create robust synthetic gene oscillators with tunable output signals have been made with varying degrees of success [5]. The authors believe that such attempts would be more successful if the physical properties that govern the behavior of these networks were better understood. One way to study these physical properties is to investigate the numerical properties of the oscillators. This has been the focus of this thesis project.

2 Background and Theory

2.1 Gene Regulatory Networks

In molecular biology, proteins are the primary functional entities in cells, and genes are the units of DNA that code for different proteins [4]. The central dogma of molecular biology gives the pathway through which genes are transformed into functional proteins [4]. Basically, DNA stored in the nucleus is first transcribed into RNA. After some processing, the RNA can be transferred from the cell nucleus to a ribosome, where it can be translated into a polypeptide chain. Finally, once this polypeptide chain has been subjected to some post-translational modifications and has been folded, it becomes a functional protein [4]. The process of transforming a gene into a functional protein is called gene expression [4].

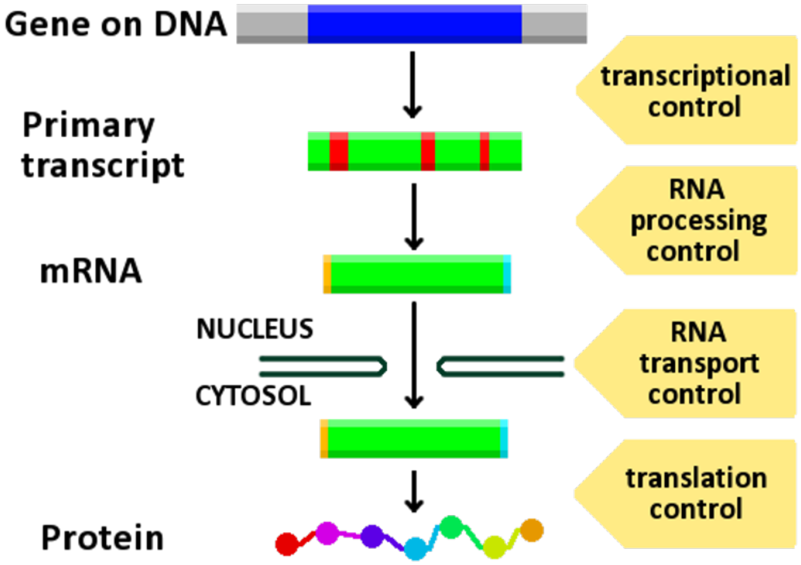

Figure 1: A diagram highlighting the steps in gene expression and the different points at which regulation can occur. Source: [7].

At each point in the process of expressing a gene, there is an opportunity to enhance or inhibit the expression of the gene [4]. Enhancing gene expression is called up-regulation, while inhibiting gene expression is called down-regulation [4]. The different points in gene expression at which regulation can occur are highlighted in figure 1. In gene regulation, a protein interacts with one of these steps to up- or down-regulate the expression of a gene [4]. The protein interacting with the expression process is itself the result the expression of a different gene that is subject to its own regulation. Because of this, the expression of the gene that codes for the protein is said to regulate the expression of the original gene [4]. A set of genes whose expression regulate each other's expression is called a gene regulatory network [6]. Gene regulatory networks can give rise to all kinds of complex molecular, cellular, and organismic behavior [6]. The primary behavior of interest for this thesis is oscillation.

2.2 Biochemical Reaction Networks

To study the numerical properties of gene regulatory networks, they must first be transformed into biochemical reaction networks. A biochemical reaction network is simply a set of chemical equations that describe the way the species in the system interact with one another. A discussion on how gene regulatory networks are transformed into biochemical reaction networks is beyond the scope of this thesis, but it suffices to say that it can be very challenging to come up with a biochemical reaction network that accurately captures the behavior of a gene regulatory network. To highlight some vocabulary, the chemical equation that describes the reaction of molecular hydrogen with molecular oxygen to form water is shown below [8].

$$2\,\mathrm{H_2} + \mathrm{O_2} \xrightarrow{k} 2\,\mathrm{H_2O} \tag{1}$$

In this equation, H_2 and O_2 are called the reactants, and H_2O is called the product [8, 9]. Generally, the products and reactants are called the species of the reaction [9, 10]. The 2 in front of H_2 and H_2O and the implied 1 in front of the O_2 are called the stoichiometric coefficients, and the k represents the reaction rate constant of the reaction [9, 10]. A chemical equation can be used to represent complicated chemistry in which different reactants form several intermediates that react with other reactants or each other in multiple steps before ultimately becoming products [8, 9]. In the rest of this thesis, only elementary reactions, or reactions with only one step, will be considered [8, 9].

The rate of a reaction quantifies the speed at which a reaction occurs [9]. It can be used to relate the time derivative of each species concentration to the concentrations of the reactants in a reaction [9]. This relationship is an ordinary differential equation which can be solved to give the concentration of each species as a function of time given an initial state. In these reaction networks, a state is the combination of the number of each species in the system [10]. Below, the reaction rate r is given for a general chemical equation representing an elementary reaction (which will be referred to as an elementary chemical equation) [9].

$$m\mathrm{A} + n\mathrm{B} \xrightarrow{k} o\mathrm{B} + p\mathrm{C} \tag{2}$$

$$r = \frac{-[\dot{A}]}{m} = \frac{[\dot{B}]}{o-n} = \frac{[\dot{C}]}{p} = k[A]^m[B]^n \tag{3}$$

In the rate equation, the square bracket notation is used to denote the species number concentration, and the overdot notation is used to represent time derivatives. That is, $[A] = \frac{N_A}{\Omega}$, where N_A represents the number of species A in the system, and Ω represents the system volume, and $[\dot{A}] = \frac{d[A]}{dt}$, where t represents time. In any chemical equation, a species may be a reactant, a product, or both. Note that all three possibilities are represented in this example.

In a biochemical reaction network, there are multiple chemical equations. The overall time derivative of each species concentration in the system is given by the sum of the time derivatives of the species concentration derived from each chemical equation in the system [11]. For simple reaction networks, this system of differential equations can be solved analytically to determine the time evolution of the concentration of each species in the system. Even when the system of differential equations is too complicated to be solved analytically, it can still be studied numerically to provide insight into the time evolution of the species concentrations the biochemical reaction network.

These systems of differential equations can also be used to define a thermodynamic potential surface using the following equation: [11, 12, 13]

$$\frac{-\partial V}{\partial [S]} = [\dot{S}] \tag{4}$$

Here, V represents the potential, and S is a general species in the reaction network. Treating a biochemical reaction network as a set of static chemical equations necessarily makes such a network thermodynamically isolated. In open or closed systems, the potential V would be equivalent or related to such thermodynamic potentials as the internal energy, Helmholtz or Gibbs free energy, or enthalpy of the system, but there is no specific name for the potential in isolated systems. System states which minimize the potential are favored over system states that correspond to higher potentials, so looking at a potential surface gives information about toward which direction a system will proceed from any initial state [12, 13].

In a biochemical reaction network, a stationary state is a state in which the concentration of the species in the system do not change over time [9]. Setting the time derivatives of the species concentrations derived from a biochemical reaction network to zero allows the stationary states of a system to be found and analyzed. From equation 4, it can be seen that stationary states correspond to critical points in the potential surface. A stationary state can be classified as either stable or unstable. Stable stationary states are stationary states that the system will return to when the concentration of a species is perturbed [13]. They correspond to local minima in the potential surface [12, 13]. When the species concentration is perturbed from an unstable stationary state, the system will end up at a stable stationary state rather than the original unstable stationary state [13]. Unstable stationary states correspond to local maxima or saddle points in the potential surface [12, 13].

A bifurcation is defined as a qualitative change in the stationary states of a system that occur as one of the parameters in the system (called the bifurcation parameter) is changed [2]. A bifurcation point is the point at which the qualitative change emerges, and a bifurcation diagram plots the stationary states of the system as a function of the bifurcation parameter [2]. In this thesis, continuous and discontinuous bifurcations will be considered. In a discontinuous bifurcation, two or more stable stationary states coexist given the same set of parameters, and there is a discontinuous jump between them [14]. With a continuous bifurcation, no such jumps exits [14].

2.3 Stochasticity

In the previous sections, the biochemical reaction networks have been treated deterministically. That is to say that they have been treated as if the reactions in the system execute continuously at exactly defined rates. At the level of the individual reaction, this treatment is not accurate because chemical and especially biochemical reactions are inherently probabilistic [8]. Given a system in a certain state, it is only ever possible to estimate the probability that the reactants in a reaction will collide with the proper orientation and energy in a certain amount of time to form products [8]. The rate of a reaction represents the average number of reactions that execute in a certain time period [8].

When only the behavior of an ensemble of reactions is of interest, a deterministic treatment of the chemical reaction networks is sufficient to provide chemical insight. However, when the behavior of individual reactions is of interest, the reaction networks must be treated stochastically. Deterministic treatment of biochemical reaction networks can sometimes fail to capture the qualitative experimentally observed behavior of the networks. As an extreme example, there are some systems which have been experimentally observed to oscillate, but a deterministic treatment of the chemical equations that describe these networks completely fail to capture this oscillatory behavior [2]. A stochastic treatment of the chemical equations does not have this problem [2].

A stochastic system is a system where non-deterministic, "random" noise is present [13]. In treating biochemical reaction networks stochastically, a stochastic noise term is introduced into the reaction rates [13]. In equilibrium, stochastic chemical systems do not remain in a constant stationary state. Rather, species concentrations fluctuate around their stationary states [13]. The thermodynamic potential defined in equation 4 is still valid, and the shape of this potential and the amount of noise in the system govern how much the species concentrations fluctuate. When these fluctuations are present, it is possible, though improbable, for fluctuations to build up and direct the system away from equilibrium [13]. This means that every state in the system is reachable from any other starting state in the system [13]. It may take an extraordinarily long time for the system to transition from one state to the other, but, because the probability of the transition is nonzero, the transition is guaranteed to happen given enough time [13].

The presence of stochastic fluctuations in a chemical system can cause a stable stationary state to become transient. In certain reaction networks, it is possible to exhaust the supply of a species in the system. If all the reactions require one or more of the exhausted species to be present for the reaction to execute, then there are no reactions that can execute, and the system becomes stuck in a single state. Because it is not possible to leave this state once it has been reached, the state is called an absorbing state [11]. If such absorbing states exist in a reaction network, the presence of stochastic fluctuations guarantee that the system will become stuck in the absorbing state if left fluctuating for enough time. The absorbing state may not be one of the stable stationary states of the system. In such instances, the stable stationary state is said to be transient.

2.4 Chemical Master Equation

When a stochastic chemical system is in equilibrium, there is a certain probability of finding the system in each possible state. This probability distribution does not change with time, so it is called stationary. When a stochastic chemical system is not in equilibrium, there is a different, non-stationary probability distribution that describes how likely it is to find the system in each possible state. The chemical master equation, derived from statistical mechanics, gives the stationary and non-stationary probability distributions at each point in time given an initial set of conditions [13]. The form of the chemical master equation is given below: [13]

$$\partial_t P(\vec{x}, t|\vec{x}_o, t_o) = \sum_{j=1}^{M} (a_j(\vec{x} - \vec{\nu}_j) P(\vec{x} - \vec{\nu}_j, t|\vec{x}_o, t_o) - a_j(\vec{x}) P(\vec{x}, t|\vec{x}_o, t_o)) \qquad (5)$$

In this equation, P represents the probability of finding the system in a particular state at a particular point in time t. The state vector $\vec{x}$ represents the number of each species in the system, and t represents time. The $\vec{x}_o$ and t_o variables represent the initial species numbers and initial time. The scalar M represents the number of reactions in the system, and a_j is a quantity called the reaction propensity, which is related to the rate of a reaction, defined by equation 3. Finally, the vector $\vec{\nu}_j$ represents the stoichiometric coefficients for reaction j for each each species in the system. (If a species does not participate in the reaction, the stoichiometric coefficient is zero.)

For simple reaction networks, the chemical master equation can be solved analytically. However, it quickly becomes intractable when there are many reactions present in the network [13]. In such situations, the most common way to approximate the probability distribution given by the chemical master equation is to use a computational method called the Stochastic Simulation Algorithm (SSA) [10]. The SSA works by starting the system in an initial state and continuously advancing the system through time by executing the next most likely reaction in the network, as predicted using the rate from equation 3 with an added stochastic noise term. This process stops when a predefined end time is hit [10].

After each reaction is executed, the state of the system is recorded. The recording of the state of the system as a function of time is called a trajectory [10]. At each point in time, the probability of the trajectory being in a particular state is given by the probability distribution of the chemical master equation [10]. Each trajectory produced using the SSA is thus equivalent to a stochastic sample of chemical master equation probability distribution [10]. Creating a time evolving histogram from an ensemble of trajectories and normalizing it will approximate the probability distribution given by the chemical master equation [10].

3 Computational Method

Oscillatory gene networks operate in a limit where stochastic fluctuations in the system play a huge role in the behavior of the oscillator [2]. Many oscillatory gene networks contain a discontinuous bifurcation, and, when they oscillate, they are actually jumping from one of their stable stationary states to another [2]. Such gene networks also often operate in a limit where one of their stable stationary states is transient, so it is believed that investigating the behavior of oscillatory gene networks at their bifurcation points and investigating the transience of stable stationary states will provide new insight which may be used to create more robust synthetic gene oscillators which output ideal signals with tunable amplitudes and periods. The ability to rigorously and efficiently study these features in gene networks provides the motivation for the computational method that was developed in this thesis project.

Ultimately, this method is based on the Stochastic Simulation Algorithm. However, use of the SSA alone is not sufficient to study gene network features, so the developed method extended it in several ways. The most important addition to the SSA was the ability to perform simulations in reverse time. In forward time, the SSA continuously executes the most probable reaction in the network based on the reaction rates [10]. In reverse time, what were the least probable reactions in forward time become the most probable reactions, and they are executed continuously instead [15]. In reverse time, probability distributions are given by the reverse chemical master equation, shown below [15].

$$\partial_t P(\vec{x}, t|\vec{x}_o, t_o) = \sum_{j=1}^{M} (a_j(\vec{x} + \vec{\nu}_j) P(\vec{x} + \vec{\nu}_j, t|\vec{x}_o, t_o) - a_j(\vec{x}) P(\vec{x}, t|\vec{x}_o, t_o)) \tag{6}$$

Effectively, switching to reverse time makes the stable stationary states act as unstable stationary states and vice versa, which means the system behaves as though its potential curve were flipped upside down [15]. The ability to perform simulations in reverse time is necessary because it allows unstable stationary states to be sampled easily. More generally, areas of the potential curve that are hard to sample in forward time due to how improbable it is to find the system in that area can be sampled efficiently by switching to reverse time. The importance of being able to study such regions will be discussed later in the context of an example.

A second modification to the SSA is that the number of any species in the system can be held constant. This is accomplished by not updating the number of the species held constant whenever a reaction is executed. By including this feature, the dependence on just one or two species of the probability distribution given by the chemical master equation can be examined at one time.

A third modification to the SSA is that the state space can be bounded. Upper and lower bounds can be placed on any and all species in the system to restrict the region of the probability distribution that is sampled. When the system hits a bound, there are two ways the situation can be handled. First, the program can simply delete the trajectory and start it over. This is done when the bounds represent uninteresting or rare events that can safely be ignored. Second, the program can record the time in which the trajectory reached the bound and stop simulating that trajectory. Once all trajectories in a simulation reach a specified time, the trajectory that hit a boundary can be given a new state based on the probability distribution formed by the other trajectories at the time the boundary was hit. (If N_l and N_u represent the lower and upper boundaries and t_b represents the time a trajectory hit one of the boundaries, the trajectory can be given a new state at t_b that is sampled from the probability distribution $P(s, t_b | s_o, t_o, N_l < s < N_u)$.) Once the trajectory has been assigned a new state, simulation of that trajectory can be continued. The importance of this feature will be discussed later in the context of an example.

Beyond the extensions made to the SSA, it is important to highlight some of the features of the implementation of the program that make it particularly well suited to efficiently achieving the goals that motivate the development of this program. First, the program is written in C++. Using this language allows for an object-oriented implementation that is easy to scale and debug as the scope of the program evolves and grows. Second, the program is parallelized using OpenMP [16]. The behavior of each trajectory is necessarily independent of the behavior of the others, so each trajectory can easily be run in parallel using separate threads of execution. This greatly reduces the runtime needed to simulate multiple trajectories, and it allows the program to take advantage of supercomputing resources.

Third, the program uses a specially formatted NetCDF [17] file to store all of the details of the reaction network, like which species participate in which reactions, the kinetic constants associated with the reactions, the initial state of the system, how long the system should be simulated, etc. This allows the program to easily be used on any general reaction network. The simulation data is also stored in this same file to facilitate analysis using other programs, like MATLAB. The NetCDF file type is designed to hold a lot of data efficiently and to allow that data to be transferred into or out of the file quickly and easily [17]. The special formatting of the NetCDF file used by the program is based on the NetCDF file associated with a project called the Synthetic Biology Software Suite (SynBioSS) [18]. SynBioSS also runs modified SSA simulations, and there was a graphical user interface (GUI) created for this project which allows users to easily generate these specially formatted files for simple or complex reaction networks [18]. This GUI was developed because creating these files by hand can be tedious and is prone to error, especially for complex networks. To make use of this existing resource, a utility was created to allow the file generated by the SynBioSS GUI to be converted into files usable by the program developed in this thesis project.

Finally, it should be noted that the implementation of the program is very efficient. In twelve hours and using eight CPUs, the program ran the equivalent of sixty traditional SSA simulations each with ten thousand trajectories over thirty time units. These simulations produced over six hundred million data points, or about ten gigabytes of data. This is a very large volume of data produced using a very reasonable amount of computation.

4 Computational Study

To provide evidence that the program created in this thesis project can accomplish the goals which motivated its development, it was tested on two simple reaction networks called the Schlögl models. The Schlögl models do not represent any naturally occurring chemical system [14]. Rather, they were designed specifically to exhibit discontinuous and continuous bifurcations [14]. The second Schlögl model also has a stable stationary state that is transient [11]. The numerical properties of these models are well known, so the results of simulations using the models are very predictable [14]. The fact that these models contain the features of gene oscillators under investigation in this thesis and that they are simple and predictable made them ideal for showcasing the abilities of the computational method developed in this thesis project.

4.1 Schlögl Model 1

The first Schlögl model contains a discontinuous bifurcation [14]. It is defined by the following chemical equations: [14]

$$\mathrm{B} + 2\,\mathrm{X} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} 2\,\mathrm{X} \tag{7}$$

$$\mathrm{B} \underset{k_4}{\overset{k_3}{\rightleftharpoons}} \mathrm{X} \tag{8}$$

Using a deterministic analysis, these equations show that:

$$[\dot{X}] = -[\dot{B}] = k_1[B][X]^2 - k_2[X]^3 + k_3[B] - k_4[X] \tag{9}$$

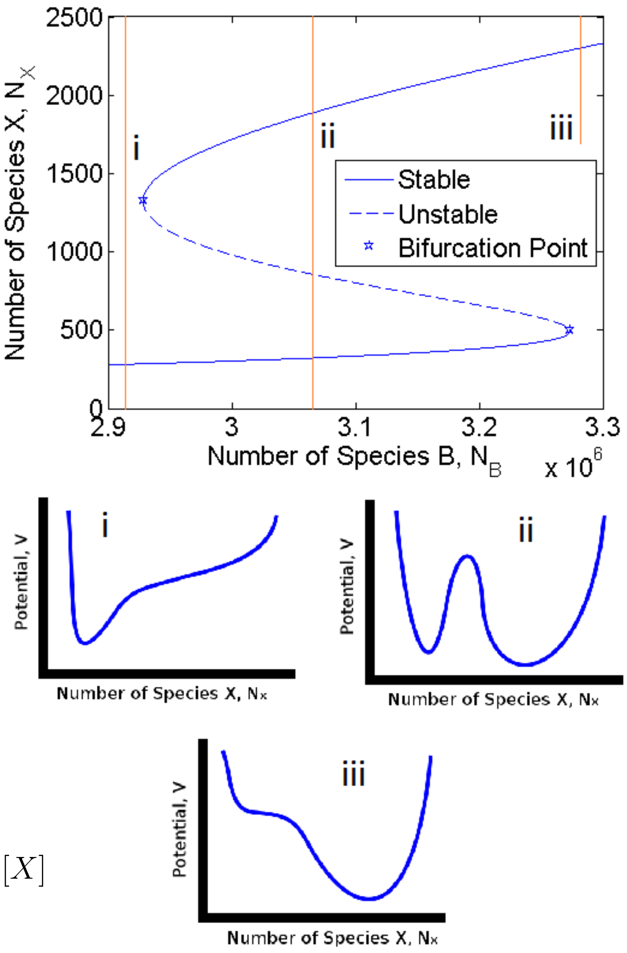

Figure 2: *Top:* The bifurcation diagram for the first Schlögl model. *i:* A cartoon of the potential for low values of N_B. *ii:* A cartoon of the potential for mid-range values of N_B. *iii:* A cartoon of the potential for high values of N_B. All plots are original.

The bifurcation diagram using N_B as the bifurcation parameter is shown in figure 2. Cartoons of the potential as a function of N_X for three different values of N_B can be seen below the bifurcation diagram. These plots show the discontinuous bifurcation of the first Schlögl model. At low values of N_B, there is only one stable stationary value of N_X, and there is only one well in the potential curve. As N_B is increased, a second stable stationary state emerges along with an unstable stationary state. In the potential curve, a second well and a local maximum between the wells emerge. Where there are two stable stationary states, the system is said to be bistable, and the discontinuous jump between the two stable stationary states is the defining characteristic of the discontinuous bifurcation [14, 13]. Finally, as N_B increases further, the original stable stationary state and the unstable stationary state disappear, and the system becomes monostable once again.

This model was used to showcase the program's ability to investigate bifurcation points. Ten different simulations were run. In each simulation, $k_1 = 10^{-9}$, $k_2 = 10^{-6}$, $k_3 = 1.7*10^{-4}$, and $k_4 = 2.5$. The initial number of species X in the system was set at 500, and the volume was set to 1. In each simulation, the number of species B in the system was held at different constant values. The value of N_B was swept from $2.92 * 10^6$ to $3.28 * 10^6$ in increments of $0.04 * 10^5$. These values were chosen to allow all features seen in figure 2 to be examined through simulation. For each simulation, three iterations of simulating in forward time followed by reverse time were run. Each simulation in forward time was run from 0 to 30 time units, and each simulation in reverse time was run from 30 to 0 time units. Finally, the state space of N_X was bounded, with the lower bound at 0 and the upper bound at 9000, and the situation of a trajectory hitting a boundary was handled by deleting the trajectory and restarting it from the beginning.

Using this procedure, the data needed to compute the approximation to the probability distribution given by the chemical master equation for each set of initial conditions can be produced. From this information, the stable and unstable stationary probability distributions can be found, and the stochastic bifurcation diagram can be created. A comparison to the deterministic bifurcation diagram shown in figure 2 can then be made.

4.2 Schlögl Model 2

The second Schlögl model contains a continuous bifurcation [14]. It is defined by the following chemical equations: [14]

.

$$\mathrm{X} + \mathrm{A} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} 2\,\mathrm{X} \tag{10}$$

$$\mathrm{X} + \mathrm{B} \xrightarrow{k_3} \mathrm{C} \tag{11}$$

Using a deterministic analysis, these equations show that:

$$[\dot{X}] = k_1[X][A] - k_2[X]^2 - k_3[X][B] \tag{12}$$

$$[\dot{A}] = k_2[X]^2 - k_1[X][A] \tag{13}$$

$$[\dot{B}] = -[\dot{C}] = -k_3[X][B] \tag{14}$$

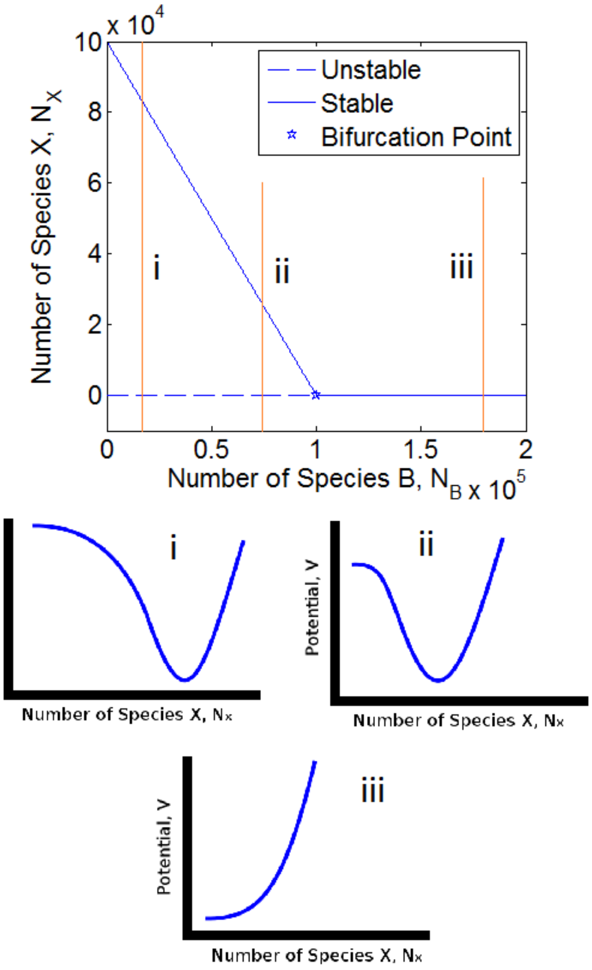

Figure 3: *Top:* The bifurcation diagram for the second Schlögl model with N_A and N_C held constant. *i:* A cartoon of the potential for low values of N_B. *ii:* A cartoon of the potential for mid-range values of N_B. *iii:* A cartoon of the potential for high values of N_B. All plots are original.

The bifurcation diagram using N_B as the bifurcation parameter with N_A and N_C held constant is shown in figure 3. Cartoons of the potential as a function of N_X for three different values of N_B can be found in figure 3. These plots show the continuous bifurcation of the second Schlögl model. At low values of N_B, there is a single stable stationary state of N_X and a single well in the potential curve which linearly approach zero as N_B increases. When the stable stationary state hits zero, zero becomes a stable stationary state. Before the bifurcation point, zero is an unstable stationary state, and there is a crest in the potential curve at zero. Notice that every reaction in equations 10 and 11 requires at least one species of X to proceed. This means that when $N_X = 0$, no reactions can proceed. This makes $N_X = 0$ an absorbing state, and the non-zero stable stationary state is transient.

This model will be used to showcase the program's ability to investigate transience. At the time of writing, the simulations involving this model have not yet been run, so the proposed simulation will only be outlined here. The average amount of time it takes a system to go from a stable stationary state to an absorbing state is called the escape time [11]. Given $k_1[A] = k_2 = 1$, a deterministic analysis gives the escape time for the second Schlögl model as: [11]

$$t_e = \exp(\Omega(\eta \ln \eta + 1 - \eta)) \tag{15}$$

$$\eta = k_3[B] \tag{16}$$

Recall that Ω represents the volume of the system. Ten simulations will be run in forward time with different Ω values to show the program's ability to capture this relationship. To calculate the escape time, a lower bound will be placed at zero. Whenever a trajectory hits this bound, the second method of handling the situation will be employed. That is to say that the trajectory will be given a new, non-zero state sampled from the probability distribution given by the chemical master equation at that point in time, and the simulation will be continued.

Whenever the bound is hit, the time the bound was hit will be recorded. The simulation will proceed until the average number of trajectories that hit the boundary in a given unit of time is nearly constant. From this constant value, the escape time can be calculated, and running the simulation with different Ω values can verify relationship described by equation 15. Once these simulations have been run and analyzed, the results will be able to be found in an updated version of this thesis at <https://code.google.com/p/rxn-sys-sim/>.

5 Simulation Results and Analysis

5.1 Raw Data

In figure 4, the raw simulation data obtained from the simulation of the first Schlögl model when $N_B = 3.12 * 10^6$ is shown. Ten thousand individual trajectories are shown in each plot, though most of them cannot be seen because they cover each other up. In plot A, all trajectories start at $t = 0$ and $N_X = 500$, and the simulation is run forward in time. It can be seen that the trajectories quickly spread out and settle down to a straight segment

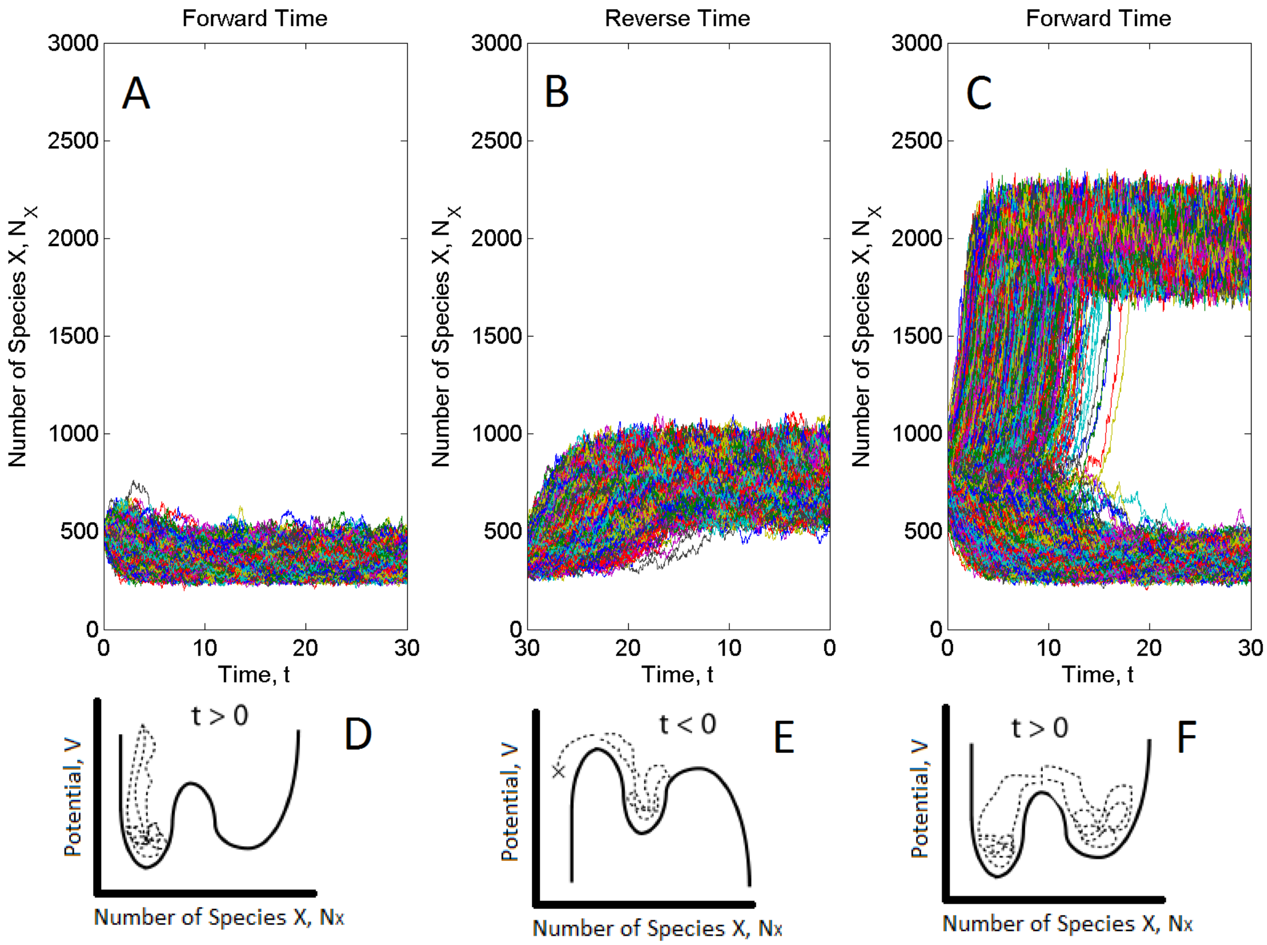

Figure 4: *A-C:* Original plots showing the raw simulation data for the first Schlögl model in forward and reverse time. *D-F:* Cartoons adapted from [15] showing the qualitative behavior of the trajectories (in dashed lines) on the potential curve (solid line).

with a fairly constant width. This is one of the stable stationary states of this system. Below the plot, a cartoon of the potential is also shown with dashed lines representing the behavior of individual trajectories. Initially, the trajectories start off in the same state with a high potential value. This is not favorable, so, as time passes, the trajectories fall down into the potential wells and start fluctuating about the potential minimum.

After $t = 30$ has been reached, the simulation is stopped and switched to reverse time, which is shown in plot B in figure 4. Recall that switching to reverse time is like flipping the potential curve upside down in forward time. This is pictured in the potential cartoon below the second plot. In the plot, it can be seen that the system transitions from the original stable stationary state to a new stationary state. This state corresponds to the unstable stationary state. Again, the qualitative behavior of the trajectories is shown on the cartoon potential. When the potential is flipped upside down, some trajectories fall into the new well, but others fall towards $N_X = 0$. As previously stated, a boundary was erected at zero to handle these trajectories. Eventually, all trajectories end up fluctuating around the unstable stationary state.

Once all trajectories reach $t = 0$, the simulation is stopped again and switched back to forward time, which is shown in plot C in figure 4. The cartoon of the qualitative behavior of the trajectories on the potential curve is shown below the plot. This time, since trajectories start around the unstable stationary state, a portion of the trajectories fall back into the original well, but another portion fall into the second potential well. On the plot, all trajectories can be seen to start around a single stationary state, but they quickly diverge into fluctuating around two different stationary states. With these three simulations, all three stationary states have been sampled by the trajectories. For more complicated potential surfaces, this procedure of iteratively switching from forward to reverse time and back again can be repeated as many times as necessary to ensure all stationary states of interest are sampled. In this study, another simulation in forward time and two additional simulations in reverse time were run, just for good measure. The bound at $N_X = 9000$ became important

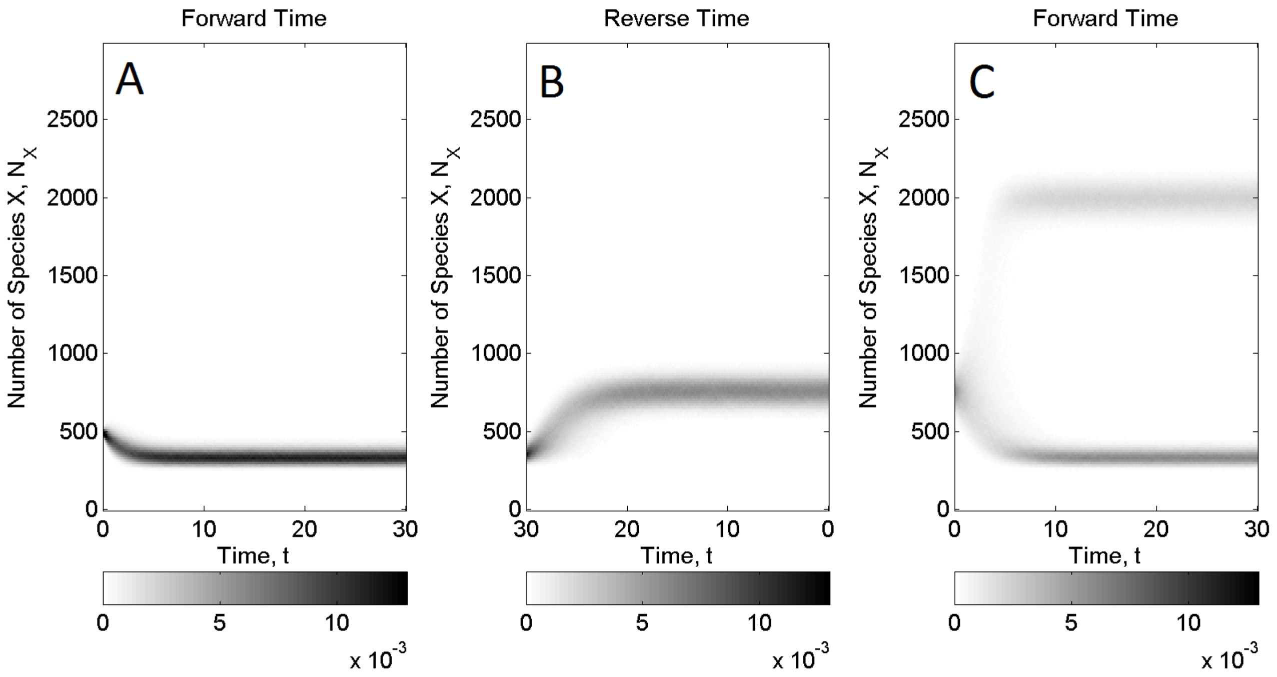

Figure 5: Original plots of the probability distributions derived from the raw data produced by the simulations of the first Schlögl model. The color bars map the darkness of the plots to the probability values.

when switching from forward time back to reverse time in the second iteration onward.

5.2 Analysis

Histograms of the value of N_X at each point in time for each simulation can be made. Normalizing these histograms produce probability distributions that are approximations to the probability distributions given by the chemical master equation. The probability distributions produced from the data in figure 4 are plotted in figure 5. In this figure, the darkness of the plot corresponds to the probability of finding the system in that state at that point in time.

From these probability distributions, approximations to the stationary probability distributions can be made. This can be done by averaging the probability distributions from figure 5 after they have stopped changing throughout time. In forward time, all probability distributions after $t = 15$ were averaged, and, in reverse time, all probability distributions before $t = 15$ were averaged. The resulting stable and unstable stationary probability distributions are plotted for three values of N_B in figure 6. These figures show the bifurcation in the first Schlögl model. As N_B increases, the location of the unstable distribution changes from overlapping with the stable state on the right, to being between the two, to overlapping with the stable state on the left. The locations and shapes of the stable distributions also change as the bifurcation parameter changes.

The stationary stable and unstable probability distributions represent vertical slices of the stochastic bifurcation diagram. The stochastic bifurcation diagram created by interpolating between the slices can be seen in figure 7 with the deterministic solution overlayed. Again, the darkness of the plot represents a greater probability of finding the system with a given combination of N_X and N_B. The stable stationary states can be seen on the top and bottom of this plot, and the

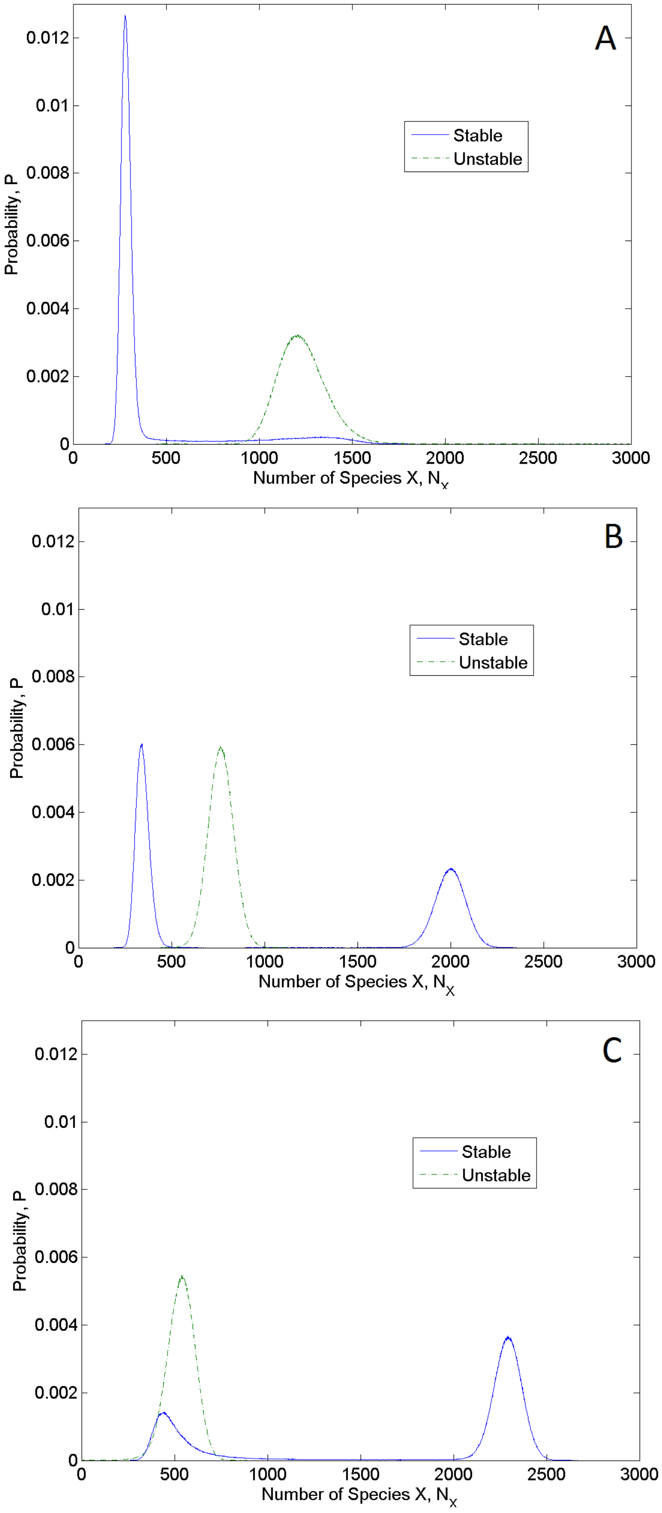

Figure 6: Original plots showing the stable and unstable stationary probability distributions for the first Schlögl model.
A: $N_B = 2.92 * 10^6$. *B:* $N_B = 3.12 * 10^6$. *C:* $N_B = 3.28 * 10^6$.

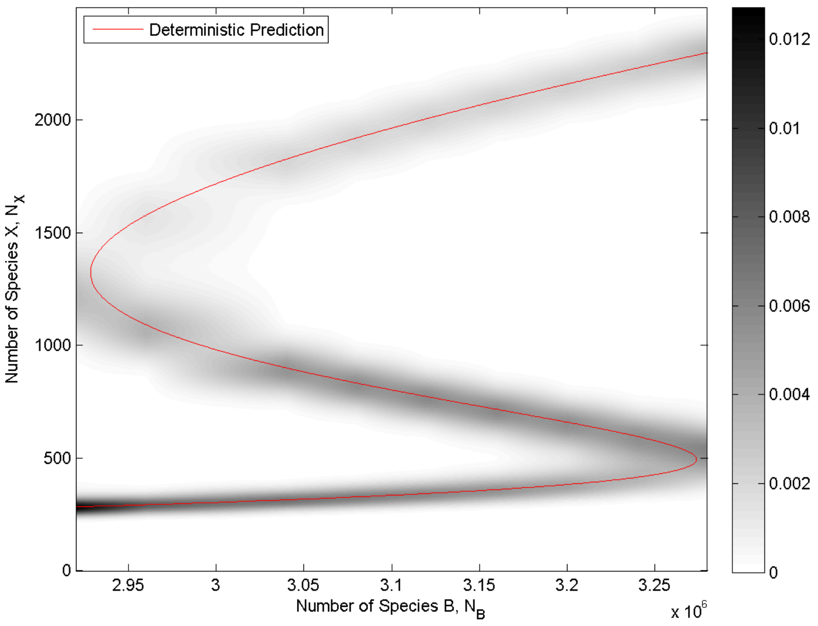

Figure 7: An original plot of the stochastic bifurcation diagram for the fist Schlögl model. The color bar maps the darkness of the plot to the probability, and the deterministic prediction is shown as a red line.

unstable stationary states can be seen in the middle. It can be seen that the stochastic bifurcation diagram match the deterministic predictions quite well.

6 Discussion

The fundamental goal of this thesis project has been to develop a piece of software to perform simulations that can be used to better understand the design principles behind robust biological oscillators. Specifically, the computational method was designed to investigate the role stochastic noise plays in the bifurcation points in biochemical reaction networks of interest and to investigate the transience of stable stationary states in these networks. The source code for the program along with instructions on how to compile and use it can be found at <https://code.google.com/p/rxn-sys-sim/>.

The results presented for the first Schlögl model show the potential for the computational method to be used for investigating the role stochastic noise plays in bifurcation points. To investigate the behavior of the system at the bifurcation points, one can imagine designing a computational study where N_B is more tightly ranged over a single bifurcation point, and the state space of N_X is more tightly bounded. Such a study would be easy to perform using this program. Once the simulations using the second Schlögl model have been performed, the results will show the potential for the computational method to be used for investigating the transience of stable stationary states. (These results can be found in an updated version of this thesis at the aforementioned website.) These results combined have and will affirm the ability of the method to achieve the fundamental goal of this thesis project.

There is much future work that can be done using this method. Once the simulations for the second Schlögl model have been run, a rigorous study of the bifurcation points and transience in the Schlögl models can be performed. Models with more complexity that represent actual synthetic gene oscillators can also be studied. As such systems are investigated, more features and improvements enabling the investigation of more numerical features of the biochemical reaction networks can be added to the program as desired. The ability to rigorously investigate the numerical properties of gene oscillators will undoubtedly provide new understanding of the physical principles that govern the behavior of such oscillators. In the future, bioengineers will be able to use this information to more successfully design robust, tunable, synthetic oscillators for use in complex biological circuitry. The program created in this thesis project has the potential to directly contribute to this goal.

7 Authors' Contributions

Noah Trebesch wrote the thesis and performed the research presented herein. Professor Jorge Viñals thought up the thesis project and acted as an advisor during its execution. All authors and readers read and approved the final thesis.

8 Acknowledgments

The authors would like to thank Professors Yiannis Kaznessis and Vincent Noireaux for their roles as readers in this thesis project. The authors would also like to thank Professor

E. Dan Dahlberg for his role as a thesis committee member during the defense of this thesis. Finally, the authors acknowledge the Minnesota Supercomputing Institute for providing computational resources used to complete this project.

9 References

[1] R. E. Bryant and D. R. O'Hallaron, *Computer Systems: A Programmer's Perspective.* Addison-Wesley, 2nd ed., 2010.

[2] Y. N. Kaznessis and J. Viñals, "Multistage Modeling of Gene Regulatory Modules," *Unpublished National Science Foundation Grant Proposal*, 2011.

[3] B. Novák and J. J. Tyson, "Design principles of biochemical oscillators," *Nature Reviews: Molecular Cell Biology*, vol. 9, no. 12, pp. 981–991, 2008.

[4] N. A. Campbell, J. B. Reece, L. A. Urry, M. L. Cain, S. A. Wasserman, P. V. Minorsky, and R. B. Jackson, *Biology.* Pearson: Benjamin Cummings, University of Minnesota Custom ed., 2008.

[5] O. Purcell, N. J. Savery, C. S. Grierson, and M. di Bernando, "A comparative analysis of synthetic genetic oscillators," *Journal of the Royal Society Interface*, vol. 7, pp. 1503–1524, 2010.

[6] B. H. Junker and F. Schreiber, *Analysis of Biological Networks.* Wiley-Interscience, 2008.

[7] "Regulation of gene expression." http://en.wikipedia.org/wiki/Gene_regulation, 2014.

[8] M. Silberberg, *Chemistry: The Molecular Nature of Matter and Change.* McGraw-Hill, 5th ed., 2008.

[9] IUPAC, *Compendium of Chemical Terminology.* Blackwell Scientific Publications, 2nd ed.

[10] D. T. Gillespie, "A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions," *Journal of Computational Physics*, vol. 22, pp. 403–434, 1976.

[11] P. Hanggi, H. Grabert, P. Talkner, and H. Thomas, "Bistable systems: Master equation versus Fokker-Planck modeling," *Physical Review A*, vol. 29, no. 1, 1984.

[12] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry.* North Holland Publishing Company, 1981.

[13] C. W. Gardiner, *Handbook of Stochastic Methods.* Springer-Verlag, 2nd ed., 1985.

[14] F. Schlögl, "Chemical Reaction Models for Non-Equilibrium Phase Transitions," *Zeitschrift für Physik*, vol. 253, no. 2, pp. 147–161, 1972.

[15] H. Salis and Y. N. Kaznessis, "Numerical Bifurcation Analysis of Master Equations: A Bistable Example," *Unpublished American Physical Society Journal Manuscript*, 2006.

[16] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *Computational Science & Engineering, IEEE*, vol. 5, no. 1, pp. 46–55, 1998.

[17] R. Rew and G. Davis, "NetCDF: an interface for scientific data access," *Computer Graphics and Applications, IEEE*, vol. 10, no. 4, pp. 76–82, 1990.

[18] A. D. Hill, J. R. Tomshine, E. M. B. Weeding, V. Sotiropoulos, and Y. N. Kaznessis, "Synbioss: the synthetic biology modelling suite," *Bioinformatics*, vol. 24, no. 21, pp. 2551–2553, 2008.