Stochastic Simulation of Genetic Regulatory Networks with Delayed Reactions

Max Veit

Submitted under the supervision of Jorge Viñals to the University Honors Program at the University of Minnesota–Twin Cities in partial fulfillment of the requirements for the degree of Bachelor of Science, *summa cum laude* in Physics.

9 May 2014

Acknowledgments

I would like to thank my advisor, Professor Jorge Viñals, for his helpful guidance throughout this project and for his suggestions for improving my computational approaches and algorithms. I additionally thank Professor Vincent Noireaux and Professor Yiannis Kaznessis for reviewing this thesis and offering useful comments.

I am also grateful to my academic advisers in the University Honors Program and the School of Physics and Astronomy. Their advice and support over the past four years was instrumental in helping me explore my academic interests and find a project that appealed to those interests.

Finally, I would like to thank the authors and contributors of the open software projects Python, IPython, SciPy, Matplotlib, and LaTeX, whose software allowed me to quickly and easily implement algorithms, test ideas, explore and visualize results, and professionally communicate them in this thesis.

Abstract

Many important processes in cells are controlled by genetic regulatory networks. To accurately model such networks, it is often necessary to include reactions with delays. In this work I apply the weighted ensemble (WE) method to simulate models of genetic regulatory networks that incorporate delays. In order to accurately capture the discreteness and stochasticity present in small systems, the Gillespie stochastic simulation algorithm (SSA), extended to include delayed reactions, is used to evolve trajectories in time. Tests of this method on two simple model systems show that the WE method yields an unbiased estimate of the system's probability distribution in the presence of delays, despite the SSA's non-uniform timesteps. I additionally use the extended SSA to investigate the assumptions used in analytical models of the simple delayed-degradation system. The numerical results indicate that a mean-field approximation is not justified near the system's bifurcation point, but is conditionally justified both in the limit of small and (surprisingly) of large propensity of the delayed reaction.

Non-technical Summary

Biologists and biophysicists have observed that living cells can behave in ways that cannot be explained only by the genes encoded in their DNA. Put another way, two separate cells with exactly the same genetic code can behave in different ways. Cells can effectively make decisions, turning genes on and off in response to their surroundings. In this work I present a computational algorithm that can be used to simulate systems of interacting genetic switches to better understand how they function. I test the algorithm on two simplified model systems.

Corrections

This document was updated on 6 June 2014 to correct two errors, one in the labeling of a figure and another in the error bars of some of the plots.

Contents

Chapter 1

Introduction

The behavior and internal workings of a living cell are much more rich and complex than the bare instructions coded into its DNA would suggest. Multicellular organisms like plants and animals are the most recognizable examples: They consist of immense numbers of cells with essentially identical sets of genes, yet each cell is specialized to perform one of a huge variety of tasks [1]. Even simple organisms like bacteria can decide which genes to express, or turn into proteins, based on environmental factors [2]. The common theme is that a cell can control the expression of its genes by binding specific proteins to matching sites on its DNA, either blocking or enabling that gene's expression.

The proteins that participate in genetic regulation are produced from different genes which themselves could be regulated, so genetic switches are in general coupled to one another. The interaction of multiple genetic switches, which can number in the hundreds even in simple cells (like *E. Coli*, a commonly-studied model organism in biophysics) [3], form what is known as a **genetic regulatory network**. Such networks regulate functions ranging from cellular differentiation [1] to daily circadian rhythms [4] in complex multicelluar organisms. It is important to develop accurate and general mathematical models of genetic networks [5]. Not only is a better fundamental understanding of cellular processes worthwhile in its own right, but it would help diagnose and treat the disorders that could arise if genetic networks function improperly.

Another relatively new application of modeling genetic networks is in synthetic biology, the development of artificial cells for a variety of tasks such as producing useful chemicals like drugs or fuels, as well as delivering drugs in a targeted manner [6]. In order to perform these functions, artificial cells will require artificial genetic networks. Basic components of these networks, such as synthetic clocks, have already been synthesized in laboratories [7]. Further advancement of this field, however, also depends on a better understanding of genetic networks and their behavior.

Many diverse strategies have been employed to model genetic regulatory networks [5][8][9][10]. Often genetic regulation is approximated as a deterministic process. However, in very small systems (such as cells), stochastic thermal fluctuations in the concentrations of molecules can be expected to be large relative to their averages. It is becoming clear, with evidence from both theory and experiment, that stochasticity is a fundamental feature and often a driving force in the function of genetic regulatory networks [2][11][12][13]. In particular, there is a large amount of variation in genetic expression between individual cells in genetically identical colonies.

One way of accounting for this stochasticity is a technique known as system size expansion, which treats the concentrations of chemical species as continuous quantities without any restriction that they be nonnegative [14][15]. However, these models break down in a fundamental way when considering very small numbers of molecules, where the inherent discreteness of molecular concentrations plays an important role.

The method explored in this work is to directly simulate the sequence of chemical reactions occurring in the cell using a Monte Carlo algorithm that naturally accounts for the stochasticity in discreteness present in small systems. The algorithm is known as the Gillespie stochastic simulation algorithm (SSA) and is widely used to model genetic networks [16][17]. The aim of this work is to extend the SSA to make it more practical for simulating real-world (natural or synthetic) genetic regulatory networks in order to analyze their behavior.

1.1 Stochastic Chemical Kinetics

In order to study the behavior of genetic networks on a molecular level they are usually modeled as **chemical systems**, bounded volumes containing molecules evolving under sets of coupled chemical reactions. The reactions represent actions such as the binding of repressors to DNA sites, protein production, and protein degradation. This representation allows genetic networks to be studied from the perspective of chemical kinetics, which seeks to understand the time evolution of the concentrations of the reactants in a system. In the case of genetic networks, the reactants are the proteins that characterize the dynamics of the genetic network. The **state** of a chemical system refers to the set of concentrations (or equivalently, for a fixed system volume, populations) of all the reactants – in this case, all the proteins, mRNA, and other biological molecules one is interested in – involved in the system. As a chemical system evolves in time, it moves through the **state space** whose coordinates correspond to each of the individual concentrations.

One approach to modeling chemical systems is to go to the continuum limit, i.e. a very large (macroscopic) system where the effects of discreteness and thermal fluctuations are negligible. Specifically, this limit assumes that the smallest change possible in the concentration of any reactant X (the difference caused by adding or removing one molecule of X) is negligibly small relative to its average concentration $\bar{x}$. One may also assume in this limit that the thermal fluctuations in the concentration are negligible compared to the average, so each concentration can be treated as a *deterministic* quantity. This second assumption is the basis of the formalism of reaction-rate equations (RRE), which specifies how to solve for the concentrations using a set of deterministic differential equations.

Another technique, known as system-size expansion, makes the assumption of continuous concentrations but not of negligible fluctuations. The zeroth order of this expansion is equivalent to RRE. The first order is known as the Langevin limit; in this limit, it is possible to describe the extent of the thermal fluctuations from the RRE concentrations [14]. However, neither theory is an exact description of very small chemical systems where discreteness and the constraint of non-negative molecular populations play an important role.

The theory of stochastic chemical kinetics explicitly treats the discreteness and stochasticity present in these systems. It avoids solving for a deterministic trajectory describing how the reactants evolve in time, as the RRE method does. Instead, it attempts to find the *probability* that the chemical system will be be in a given state at a given time.

It does this by making several simplifying assumptions about the system. Namely, it assumes that the chemical system is a homogeneous (well-stirred) ideal gas or dilute solution in thermal equilibrium [14].

While this assumption begins to break down in living cells (see Section 4.1), the utility and ease of implementation of algorithms to simulate or solve the chemical master equation have led to wide use of this theory in modeling the biochemistry of cells [16][17].

Under the above condition, we can assume that the probability that a given reaction R_j will occur within the system volume in the next infinitesimal time interval of length $\mathrm{d}t$ depends only on the current state $\mathbf{x}$. This probability is written as $a_j(\mathbf{x})\,\mathrm{d}t$, where the function $a_j(\mathbf{x})$ is known as the **propensity function** [9]. In other words, the fundamental assumption of chemical kinetics is that the chemical system can be represented as a (continuous-time) Markov chain.

Using this definition of propensity, it is possible to derive a differential equation that describes the time evolution of the probability distribution of the system. This probability distribution is written $\mathrm{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)$, which means "the probability, given that the system started in the state $\mathbf{x}_0$ at time t_0, that the system will be in the state $\mathbf{x}$ at some later time t." The equation is called the chemical master equation, and in the form given in [9], it reads:

$$\frac{\partial}{\partial t}\mathrm{P}(\mathbf{x}, t|\mathbf{x}_0, t_0) = \sum_{j=1}^{N}\left(a_j(\mathbf{x}-\mathbf{s}_j)\mathrm{P}(\mathbf{x}-\mathbf{s}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x})\mathrm{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)\right) \quad (1.1)$$

The sum runs over all N reaction pathways in the chemical system. The vector $\mathbf{s}_j$, known as the state-change vector of reaction j, indicates the effect of the reaction j on the state of the chemical system: reaction j takes the state instantaneously from $\mathbf{x}$ to $\mathbf{x}+\mathbf{s}_j$.

The first term under the sum can be seen as a source term; it represents the influx of probability into state $\mathbf{x}$ caused by the reaction R_j taking the state $\mathbf{x}-\mathbf{s}_j$ to the state $\mathbf{x}$. Similarly, the second term can be seen as a drain term, representing the state change from $\mathbf{x}$ to $\mathbf{x}+\mathbf{s}_j$.

In order to simplify the above expression, we can average over all possible initial states $\mathbf{x}_0$ at time t_0 and write instead $P(\mathbf{x},t)$. Additionally, we can write the equation more compactly by introducing the shift operator $\hat{T}^{\mathbf{s}_j}$, where $\hat{T}^{\mathbf{s}_j}P(\mathbf{x},t)=P(\mathbf{x}+\mathbf{s}_j,t)$. The result is the more convenient form:

$$\frac{\partial}{\partial t}\mathrm{P}(\mathbf{x},t)=\sum_{j=1}^{N}a_j(\mathbf{x})(\hat{T}^{-\mathbf{s}_j}-1)\mathrm{P}(\mathbf{x},t) \tag{1.2}$$

1.1.1 Propensity Functions

The forms of the propensity functions defined above can be derived under the homogeneous ideal-gas assumption. It is from these derivations that we get the fundamental assumption of chemical kinetics, that the propensities depend only on the current state.

The derivations are different depending on the order of the associated reactions, i.e. whether they are production (zero-order), unimolecular, bimolecular, or higher-order reactions [9]. The order of the reaction refers to the number of separate molecules that constitute the inputs, or reactants. A production reaction, i.e. a reaction with no input reactants, can simply be modeled by using a constant propensity $a_j^{(0)} = c_j^{(0)}$.

For a unimolecular reaction, e.g. a reaction R_j taking one molecule of X_1 to some products, the probability that an isolated molecule of X_1 will undergo the reaction in the next infinitesimal time interval $\mathrm{d}t$ can be assumed to be a constant $c_j^{(1)}\,\mathrm{d}t$. The probability of *any* reaction R_j occurring within the system volume in the time interval $\mathrm{d}t$ is thus proportional to the number of molecules of X_1 in the system, so $a_j^{(1)}(\mathbf{x}) = c_j^{(1)} x_1$.

For a bimolecular reaction, the propensity is the probability (per unit time) that any pair of reactant molecules will collide within the system volume, multiplied by the probability that such a collision will actually result in a reaction. One could therefore assume the probability per unit time of any pair of molecules reacting is a constant $c_j^{(2)}$, implying the propensity function is proportional to the number of pairs of reactants in the system volume. If the two reactant molecules are of different species X_1 and X_2, the propensity $a_j^{(2)}(\mathbf{x}) = c_j^{(2)} \cdot x_1 x_2$. If they are of the same species X_1, $a_j^{(2)}(\mathbf{x}) = c_j^{(2)} \cdot \frac{1}{2} x_1(x_1 - 1)$.

1.1.2 Delayed Reactions

In order to reduce the computational cost and model complexity of applying the SSA to biological systems, an additional abstraction is adopted. Processes common in cell biology, such as DNA replication, protein production, and protein digestion, actually consist of sequences of thousands of individual chemical reactions like the binding of individual nucleotides to a developing RNA strand. It would be tedious and expensive to simulate each individual step of the process. More importantly, it would be extremely wasteful to do so if one is only interested in the high-level dynamics of proteins and genes, not the details of the constituent process.

We can avoid simulating each step of a complex biological process by modeling the entire process as a single reaction. For example, a protein-production reaction (which itself consists of many complex multi-step processes such as RNA transcription and protein folding) can be abstracted as a single reaction that produces a protein from nothing. However, this reaction cannot be said to occur instantaneously (as with simple chemical reactions), as the entire process it represents requires a nonzero duration to complete. Experimental evidence indicates that the duration of biologically important reactions such as protein production can be on the order of minutes to hours. Since this duration is long compared to the timescales of other cellular processes, it is important to account for it in the modeling of genetic networks [13].

To account for this duration, we can associate a **delay** with the reaction to represent the time the underlying process needs to complete. In effect, the propensity function for the delayed reaction depends not on the current state, but on the history of the chemical system. More precisely, if reaction j is delayed with a time τ_j, its propensity function is written $a_j\big(\mathbf{x}(t-\tau_j)\big)$. For simplicity, τ_j is assumed to be a number. In general, however, the delay may be better characterized by a probability distribution.

In the case where some of the reactions in a system are delayed, we can modify Equation (1.2) to:

$$\begin{aligned}\frac{\partial}{\partial t}\mathrm{P}(\mathbf{x},t) &= \sum_{j=P+1}^{N} a_j(\mathbf{x})(\hat{T}^{-\mathbf{s}_j}-1)\mathrm{P}(\mathbf{x},t)\\ &\quad+\sum_{j=1}^{P}\sum_{\mathbf{x}'} H(\mathbf{x})a_j(\mathbf{x}')(\hat{T}^{-\mathbf{s}_j}-1)\mathrm{P}_2(\mathbf{x},t;\mathbf{x}',t-\tau_j) \qquad (1.3)\end{aligned}$$

where P of the N total reactions are delayed and each delay τ_j is assumed to be a single number. The second sum term involves an inner sum over all states $\mathbf{x}'$ in the state space; the joint probability distribution $\mathrm{P}_2(\mathbf{x},t;\mathbf{x}',t-\tau_j)$ gives the probability of the system visiting state $\mathbf{x}'$ at time $t-\tau_j$ *and* reaching state $\mathbf{x}$ at time t. The inner sum thus covers all possible paths the system could take to reach the current state $\mathbf{x}$ over the duration of the delay. The multidimensional step function $H(\mathbf{x})$ is included to block unphysical reactions, i.e. those that take any single concentration below zero (the value of $H(\mathbf{x})$ is one if all the x_i are positive, zero otherwise).

Finally, introducing delayed reactions has an important consequence: Since delayed reactions depend on the history of the chemical system, models incorporating them are non-Markovian. This fact severely limits the analytical tools one can apply to analyze the stochastic behavior of systems with delayed reactions; Section 3.2.2 discusses the implications of non-Markovian dynamics for analytical models. It also has important consequences in the numerical simulation of delayed systems. These consequences are explored in Section 2.5.

Chapter 2

Methodology

2.1 Gillespie Stochastic Simulation Algorithm

One of the most well-known and widely used [9][16][17] algorithms for simulating small chemical systems is the stochastic simulation algorithm (SSA), which was introduced in 1976 by Daniel Gillespie [18]. As a Monte Carlo technique, it does not attempt to solve the chemical master equation (1.2) explicitly. Rather, the approach taken by the SSA is to generate a **trajectory**, which is the path $\mathbf{x}(t)$ that one possible *instance* of the chemical system might take through the state space given some initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$. Unlike the deterministic trajectories found using RRE, these trajectories are generated probabilistically. In effect, the SSA simulates the time evolution of a single instance of the chemical system with some arbitrary initial conditions.

The trajectories generated by the SSA can be seen as samples of the underlying probability distribution $\mathrm{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)$ that describes the chemical system. In principle, one can generate a very large number of samples (trajectories) in order to estimate a distribution that converges to the true one.

The SSA generates trajectories (with the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$) by repeating the following steps for each iteration n [9]:

1. Compute the propensities $a_j(\mathbf{x}_n)$ for all reactions j and their sum $a_{\text{tot}}(\mathbf{x}_n)$ as described in Section 1.1.1.

2. Choose the next reaction type and the waiting time until that reaction from the following probability distributions:

   - Waiting time: $\mathrm{P}(t_w)\,\mathrm{d}t_w = a_{\text{tot}}(\mathbf{x}_n)\exp(-a_{\text{tot}}(\mathbf{x}_n)t_w)\,\mathrm{d}t_w$
   - Reaction type: $\mathrm{P}(j) = a_j(\mathbf{x}_n)/a_{\text{tot}}(\mathbf{x}_n)$

3. Execute the reaction, i.e. update the current state and time:

   - $t_{n+1} = t_n + t_w$
   - $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_a$, where $\mathbf{s}_a$ is the state-change vector for the reaction chosen above

The iteration is continued typically until the time t reaches (exceeds) a predefined stop time. Since a chemical system remains in its current state until another reaction occurs, we can interpret the sequence $(\mathbf{x_n}, t_n)_{n=1}^{N}$ as the continuous-time trajectory $\mathbf{x}(t)$ of the instance.

The forms of the probability distributions in Step 2 derive in a straightforward way from the same assumption used to derive the chemical master equation. In a sense, the master equation and the SSA are two equivalent formulations of the same underlying theory.

2.2 Extension to Non-Markovian Dynamics

The classic SSA was designed only for explicitly Markovian chemical systems. However, one could imagine extending the algorithm to include delayed reactions. One such extension is proposed by [13]. The modified algorithm, upon selecting a delayed reaction in Step 2 (see Section 2.1), schedules the reaction to fire at a later time (that is, Step 3 is postponed by a time τ_a).

In this work, a method that more closely follows the analytical modeling of delayed reactions is chosen. The SSA is modified in the propensity calculation of Step 1: To calculate the propensity of a reaction R_j with delay τ_j, we simply use the state from τ_j time units earlier[1]. Thus, with a trajectory at time t, the propensity would be computed as $a_j = a_j\left(\mathbf{x}(t-\tau_j)\right)$. This method is consistent with the delayed-reaction formulation of the master equation, Equation (1.3).

One undesirable feature is introduced by including delayed reactions

[1] If $t-\tau_j$ is less than the trajectory's starting time t_0, we just use the initial state $\mathbf{x}_0$ to compute the propensity. This special case is only relevant for the first $(\max_j \tau_j)$ time units of the simulation, after which we sample the history normally.

using either method: Since the delayed reactions lag behind the current state, it is possible for individual concentrations to go below zero, an obviously unphysical result. In the non-delayed SSA, any reaction that could decrease the concentration x_i has a propensity proportional to that concentration (to some power). Therefore, when x_i goes to zero, any reaction that could decrease x_i has a zero propensity and is blocked from running. A delayed reaction, however, does not "notice" zero concentrations in real time so it is not blocked in time to avoid decreasing concentrations from zero[2].

Luckily, it is much easier to impose the constraint $x_i \geq 0 \; \forall i$ within the SSA than in analytical models. The method used in this work is to manually block the offending reactions: In Step 2 of the SSA, if a reaction is selected that would make any individual concentration go negative, the reaction is discarded and another is selected. This procedure amounts to manually setting that reaction's propensity to zero.

2.3 Weighted-Ensemble Resampling

The most straightforward way to obtain a probability distribution using the SSA is to run an ensemble of trajectories in parallel, then estimate

[2] Some analytical models of delayed systems have this problem owing to the difficulty of analytically constraining concentrations to be nonnegative. The result is often nonsensical behavior, such as oscillations with exponentially increasing amplitude.

the probability density $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ from those samples[3].

The main issue with this method is that it generally samples state space unevenly. Trajectories in the ensemble tend to congregate, by construction, in the most probable regions of state space while less probable regions (the "tails" of the distribution) are more rarely visited. The result is that the accuracy of the sample of the probability distribution obtained using this simple ensemble increases with the probability density at the same location. This effect makes this method extremely inefficient if one is most interested in the least probable regions, as is the case in many problems in chemistry and biology (e.g. computing transition rates). In many cases, one would need to use a prohibitively large number of trajectories to adequately sample the improbable transition regions.

To overcome this problem, we can apply a method known (in the context of molecular dynamics and stochastic simulation) as the **weighted ensemble** (WE) method [19], a type of importance sampling. Its straightforward implementation, statistically exact properties, and generality and applicability to high-dimensional concentration spaces [20][16] make it ideal for this work.

The basic strategy of the WE method is to periodically redistribute

[3] If the probability distribution is known to be steady-state, i.e. independent of time, then one can simply average a single trajectory over a long period of time to obtain $P(\mathbf{x})$. However, in systems where time independence is not known *a priori*, the ensemble method must be used.

the samples in a way that does not change our estimate of the probability distribution $P(\mathbf{x},t|\mathbf{x}_0,t_0)$. This is achieved by assigning each trajectory a **weight** w_k such that the sum of all weights is always 1. In effect, the method biases the underlying distribution so that rare events are sampled as frequently as the common ones, while keeping track of weights in order to obtain a sample of the original, *unbiased* probability distribution.

The WE method has been applied to stochastic chemical kinetics before. The implementation used in this work generally follows the version presented in [16]. The overall procedure is as follows: First, choose an initial ensemble of P trajectories. Then, repeat the following steps as many times as desired:

1. Perform a resampling step to equalize the distribution of trajectories over state space.
2. Run all the trajectories independently for a time τ_p.

Step 1 is usually referred to as the **resampling step** and Step 2 as the **dynamics step**. In this case, running a trajectory means applying the SSA to evolve the trajectory forward in time. However, the generality of the WE method permits the use of other time evolution algorithms for the dynamics step. In the more commonly encountered applications of the WE method, a molecular dynamics algorithm is used to evolve independent trajectories through phase space.

The resampling step is implemented by partitioning the state space into discrete **bins**. To equalize the distribution of trajectories over state space, we can use two basic operations: To increase the number of trajectories in a bin, we can replicate a trajectory T_a to obtain M copies. Each of the copies receives a weight w_a/M; in this way, the total weight within a bin is conserved. To decrease the number of trajectories, we can a delete a trajectory T_b (typically the one with the lowest weight) and assign its weight to another trajectory T_c in the same bin, i.e. T_c receives a new weight $w_b + w_c$. Again, the operation conserves the total weight within each bin. This conservation of weight is the key feature that ensures the statistical correctness of the resampling method.

In this work, a uniform binning strategy is adopted, i.e. the bin boundaries are spaced uniformly over some range $[x_{i,\text{min}}, x_{i,\text{max}}]$ for each coordinate x_i. Coordinates outside this range are grouped into the closest in-bounds bin. These bins are statically defined, so the estimate of the probability distribution is taken to be a histogram over the bins. This is by no means the only strategy possible; [20] gives more examples of binning strategies to show how general the weighted-ensemble method is.

2.4 Weighted-Ensemble SSA

Some important conceptual issues arise when applying the weighted-ensemble method to chemical-kinetics trajectories evolved using the SSA. The WE method is only guaranteed to be statistically exact if the resampling step occurs at the same time for all the trajectories in the ensemble [20]. However, the SSA chooses non-uniform timesteps, so it would seem impossible to "pause" all the trajectories at the same time t_{sync}.

Figure 2.1 illustrates the problem. The SSA only generates a discrete set of steps $(\mathbf{x}_n, t_n)$. Thus, the closest one can to come to pausing a trajectory at t_{sync} is to stop each trajectory once it chooses a reaction time that is after t_{sync}, but before the trajectory actually updates its state. However, this procedure results in an ensemble of trajectories each effectively at different times, which means that it may not be correct to apply resampling to this ensemble.

In the case of purely Markovian (non-delayed) dynamics, it is in fact perfectly valid to apply WE resampling to the set of paused trajectories described above. The reason is that the future probability distribution for a Markovian trajectory depends, by definition, only on its current state. This means that the distribution of waiting times until the next reaction on a trajectory is the memoryless exponential distribution. If a trajectory determines its next reaction to occur after the time t_{sync}, it is thus possible to update the trajectory's time to t_{sync} and sample the next

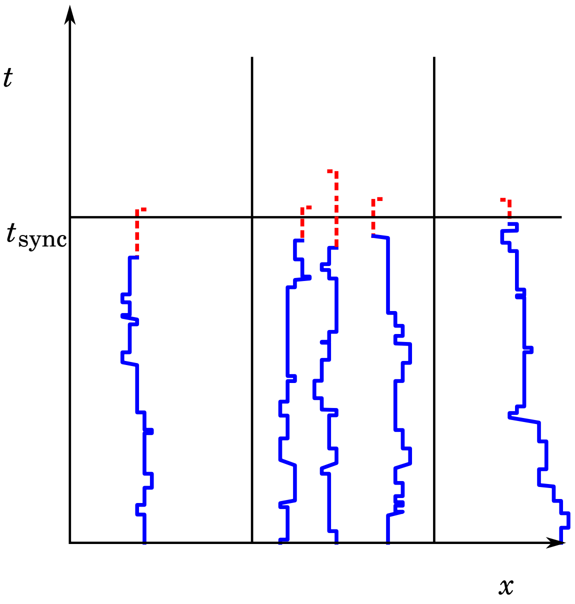


Figure 2.1: The SSA takes non-uniform timesteps, so it is not possible to pause each trajectory exactly at time t_{sync}. Instead, one must pause it at the last reaction before t_{sync}, resulting in an ensemble of trajectories each at different times.

reaction from there without biasing the overall measured probability distribution.

Therefore, it is valid to pause a Markovian trajectory at an arbitrary time t_{sync}. This is statistically equivalent to leaving the trajectory at the state of its last reaction before t_{sync}. One can thus run WE resampling on trajectories paused this way without biasing the estimate of the underlying probability distribution.

2.5 Weighted Ensemble With Delays

In systems with non-Markovian dynamics, i.e. delayed reactions, a different resolution is necessary in order to be able to apply the weighted-ensemble method. Since the propensities for delayed reactions depend on a trajectory's history, they can change in between reactions. The distribution of waiting times for a given reaction is thus no longer memoryless.

To circumvent this problem, the following implementation of trajectory pausing is used in this work: If a trajectory is to be paused at a time t, it executes the SSA until a reaction is generated that would fire later than t. Instead of executing that reaction, the trajectory saves it and becomes paused. Upon resuming, the trajectory recalls and executes the saved reaction and continues the SSA from the time of that reaction.

However, it is still an open question whether it is correct to pause an SSA trajectory in between reactions. The more fundamental question is whether it is valid to interpret the discrete sequence $(\mathbf{x}_n, t_n)$ generated by the SSA as a continuous-time trajectory. This question is not easy to address on a mathematically rigorous level. However, numerical evidence from systems with analytical descriptions indicates that WE resampling does not bias the numerical estimate of the probability distribution. This conclusion holds for systems that include delayed reactions.

Chapter 3

Results

3.1 Simple Production-Degradation System

Accurate analytical distributions exist only for the simplest model systems. One such system is the production-degradation system, consisting of two reactions with the equations:

$$\begin{aligned} \varnothing &\xrightarrow{A} X \\ X &\xrightarrow{B} \varnothing \end{aligned} \tag{3.1}$$

The parameters A and B are the propensity constants for each reaction. The parameter A sets the scale of the system; it is often written as the product ΩA of the system's volume and a parameter of order one. Here, for simplicity, the system volume is absorbed into the constant A. Using this notation, the parameter A plays a similar role to the volume Ω.

The deterministic RRE method predicts an equilibrium population of $n^\star = \frac{A}{B}$. In the Langevin limit (large Ω, continuous concentration x), the steady-state probability distribution is predicted to be a Gaussian with standard deviation also equal to $\frac{A}{B}$. Explicitly,

$$\mathrm{P}(n) = \sqrt{\frac{B}{2\pi A}} \exp\left(-\frac{B}{2A} \left(n - \frac{A}{B} \right)^2 \right) \tag{3.2}$$

Alternatively, since the fluctuations in the population are expected to scale as $\sqrt{A}$, it makes sense to define the normalized fluctuation $\xi = A^{-1/2}(n - n^\star)$. In terms of ξ, the steady-state probability distribution is then

$$\mathrm{P}(\xi) = \sqrt{\frac{B}{2\pi}} \exp\left(-\frac{B\xi^2}{2} \right) \tag{3.3}$$

3.1.1 Weighted-Ensemble Verification

The system was simulated using the weighted-ensemble method with individual trajectories evolved using the unmodified SSA. The ensemble was evolved in a sequence of dynamics steps of duration τ_p up to a total time T. At the end of the run, the probability distribution was estimated as the sum of weights within each bin. As described in Section 2.3, the division of the state space into a number N_b of bins was done uniformly over the coordinate n. The boundaries of the bins were linearly spaced between values $n_{\min}$ and $n_{\max}$. Any trajectories with states outside this range were placed in the nearest in-bounds bin. The initial trajectories

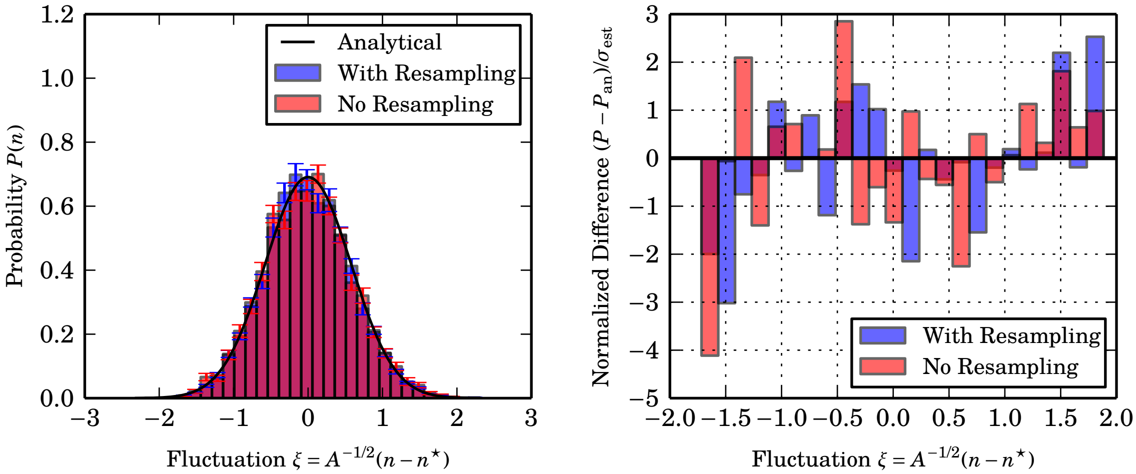
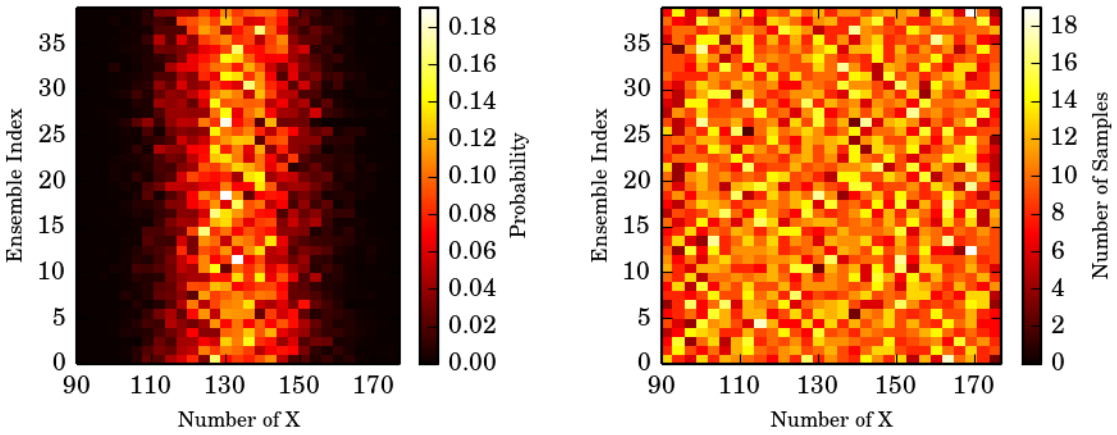


Figure 3.1: Left: comparison of the weighted-ensemble method with a non-resampled ensemble on the simple production-degradation system with $A = 400$ and $B = 3$. The probability distributions were taken after $T = 60$ (arbitrary) time units. Right: differences from the analytical distribution, normalized by each estimate's standard deviation.

used to seed the WE method were assigned random initial states chosen from the uniform distribution on $[n_{\min}, n_{\max}]$.

For all weighted-ensemble runs shown in this work, $N_b = 30$ bins were used and the range $[n_{\min}, n_{\max}]$ was chosen to encompass about four standard deviations of the distribution in either direction.

Resampling was done at the beginning of each step to ensure each bin had the target number of trajectories N_T. Note that it is not possible to increase the number of trajectories in empty bins. By resampling often enough (choosing τ_p sufficiently small) and using a large enough target number N_T, we can be reasonably certain that no bins will become empty within the concentration range in which we are in-



(a)

(b)

Figure 3.2: Data from the weighted-ensemble simulation of the simple production-degradation system in Figure 3.1. Left: Probability distributions obtained from the 40 independent ensembles that were run to estimate variance; each row is an independent distribution. Right: Number of trajectories in each bin at the end of the total run time, showing no empty bins; note that no resampling step was done after the final dynamics step.

terested. Unless otherwise noted, the parameters used were $\tau_p = 0.01$ and $N_T = 10$; each ensemble was inspected at the end of the run to make sure all relevant bins were populated (see Figure 3.2b for an example). The total time T was chosen so that the samples would be equilibriated to the steady-state distribution by then. This equilibriation time was determined by inspecting the sequence of distributions generated by a sample run of the WE simulation.

For comparison, a similar simulation was done with the resampling turned off at each step; this procedure is equivalent to running all the

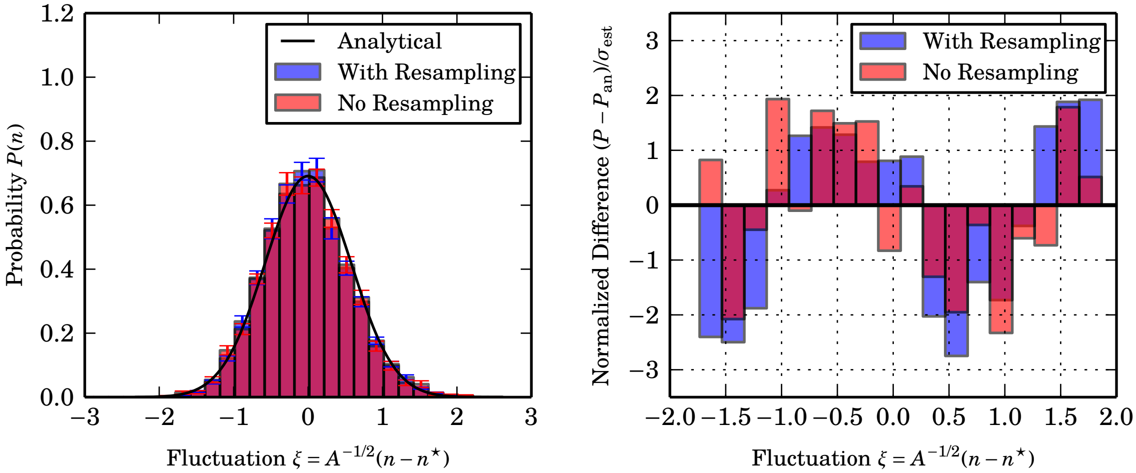


Figure 3.3: Left: comparison of weighted and non-resampled ensembles against the analytical distribution for the simple production-degradation system with $A = 100$ *and* $B = 3$ *(*$T = 60$, $\tau_p = 0.02$*), showing discreteness effects. The discrete minimum* $n = 0$ *corresponds to* $\xi \approx -3.3$*. Right: differences from the analytical distribution, normalized by each estimate's standard deviation.*

initial trajectories in parallel up until the time T. The number of trajectories used was equal to the number of bins multiplied by the target number N_T. The probability distribution was estimated simply as the number of trajectories within each bin (normalized by the total number of trajectories).

A comparison between the resampled and non-resampled distributions with a reasonably large system volume is shown in Figure 3.1.

In order to compare the two distributions in a statistically meaningful way, it is useful to estimate the uncertainty of the value of any given bin. A naïve estimate can be obtained by running many separate ensembles with the same parameters as illustrated in Figure 3.2a, then taking the standard error s_i of the values for a given bin i across all the resulting distributions (this corresponds to taking the standard deviation of each column in Figure 3.2a). Reported here is the mean value of each bin across N_{ens} independent ensembles, with the standard error on that mean estimated by $\frac{s_i}{\sqrt{N_{\text{ens}}}}$. The number of ensembles used was $N_{\text{ens}} = 40$ for all the results presented here.

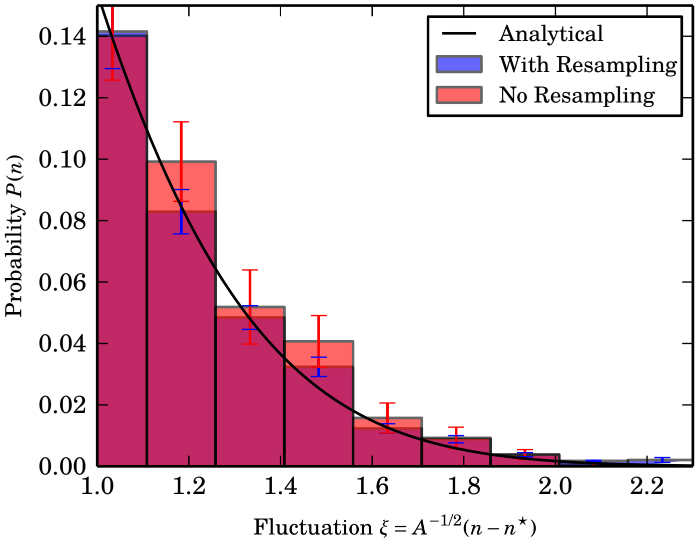


Figure 3.4: Close-up of the probability distribution estimates of Figure 3.1, showing the variances in a low-probability area of the concentration space.

The benefit of the WE method is that it improves our sampling over low-probability regions of the concentration space. Intuitively, if a bin has been visited by ten trajectories, we can be much more certain about the reported value for that bin than if we only found one trajectory there. This improved sampling manifests itself in a reduced variance in the values of the bins in the tails of the distribution, as Figure 3.4 exemplifies. For this particular model system, the variance reduction is small. The variance reduction becomes significant in more complicated models such as those used for biological systems.

Furthermore, neither of the comparisons here shows a *systematic* difference between the resampled and non-resampled distributions. Therefore, we can conclude that WE resampling does not bias our estimate of the probability distribution for the simple production-degradation system.

This conclusion is exactly what one would expect, as this system has no delayed reactions and is therefore Markovian. As discussed in Section 2.4, WE resampling is guaranteed to be correct for Markovian systems. For the purposes of verifying the WE method, it is more interesting to study systems that incorporate delayed reactions.

3.2 Delayed-Degradation System

One of the simplest possible systems that includes delayed reactions is the delayed-degradation model, defined by the reactions:

$$\begin{aligned} \varnothing &\xrightarrow{A} X \\ X &\xrightarrow{B} \varnothing \\ X &\overset{C}{\underset{(\tau)}{\Longrightarrow}} \varnothing \end{aligned} \tag{3.4}$$

This is simply the production-degradation system with an additional reaction added; the notation for the third reaction means it is a delayed reaction with propensity constant C (analogous to B) and delay τ. As in the simple production-degradation model the parameter A sets the scale of the system, absorbing the system volume Ω.

The additional delayed reaction has a profound effect on the overall qualitative behavior of this system. As the value of C is varied, the system undergoes a **bifurcation**, i.e. the system goes into a different qualitative mode of behavior. For large enough values of C, the concentration oscillates periodically between two values (the "oscillatory mode"). However, for smaller values of C the system behaves much like the simple production-degradation system, fluctuating about a stable mean value (the "stable mode"). An analytical form for the probability distribution is only available for the stable mode. In the limit of short delay τ, the distribution is [15]

$$P(\xi) = \sqrt{\frac{B+C}{2\pi(1+C\tau)}} \exp\left(-\frac{B+C}{2(1+C\tau)}\xi^2\right) \tag{3.5}$$

where ξ is defined as in Section 3.1, only here the steady-state mean population is $n^\star = \frac{A}{B+C}$. The effect of the delay is to broaden the distribution compared to the simple production-degradation system.

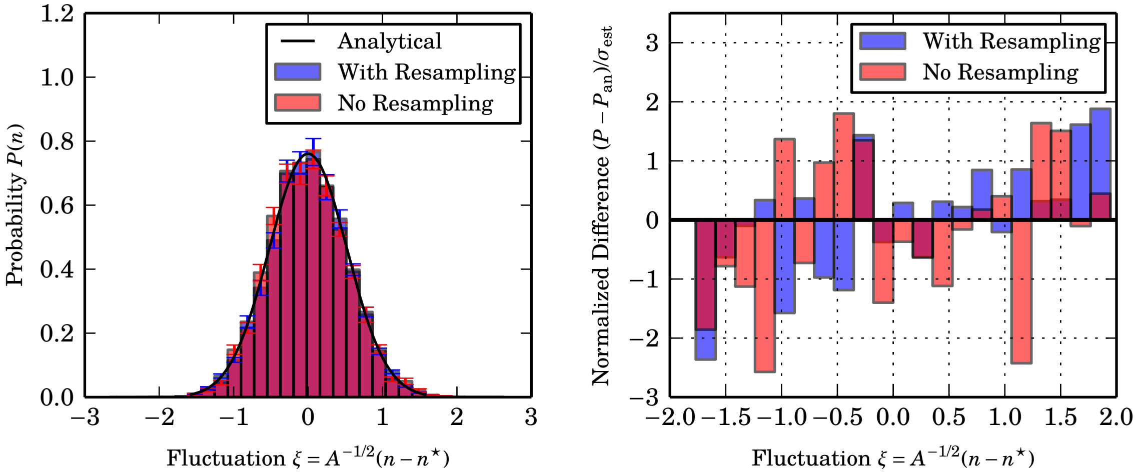


Figure 3.5: Left: comparison of weighted and non-resampled ensembles against the analytical distribution for the delayed-degradation system with $A = 800$, $B = 3$, $C = 1$, and $\tau = 0.1$ ($T = 3$). Right: differences from the analytical distribution, normalized by each estimate's standard deviation.

3.2.1 Weighted-Ensemble Verification

The procedure used to assess the effect of resampling on the numerical probability distribution for this system is the same as the one of Section 3.1.1, with a few modifications. The first modification accounts for the fact that the delay introduces an effective phase to each trajectory (the effect is more obvious in the oscillatory mode). To ensure that the probability distributions were sampled uniformly over this phase, the starting times of the seed trajectories were randomized relative to each other. Specifically, the initial time of each trajectory, relative to the ensemble time, was drawn from a uniform distribution on $[0, \tau]$.

Second, the total time T that each ensemble was run had to be chosen

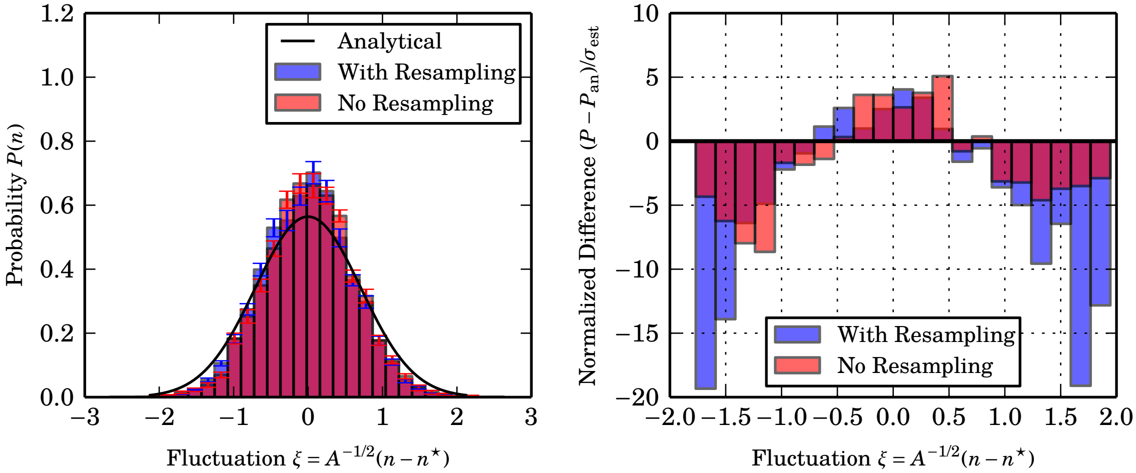


Figure 3.6: Left: comparison of weighted and non-resampled ensembles against the analytical distribution for the delayed-degradation system with $A = 800$, $B = 3$, $C = 1$, and $\tau = 1$ ($T = 3$), showing deviation from the analytical variance derived for small τ. Right: differences from the analytical distribution, normalized by each estimate's standard deviation.

to avoid any effects due to the initial phase randomization. The value $T = 3\tau$ was deemed a safe choice for this purpose, as trajectories sample their histories only as far as τ time units earlier. By time 3τ, all trajectories would have been sampling active histories (that is, not sampling earlier than their starting times) for at least τ time units.

Figure 3.5 shows a comparison between the simulation results, both with and without resampling, and the analytical distribution (3.5). As expected, the simulations agree well with the analytical distribution for a relatively short delay of $\tau = 0.1$. The agreement with the analytical distribution begins to break down for larger values of τ; see Figure 3.6. However, as Figure 3.7 shows, the distributions obtained with resam-

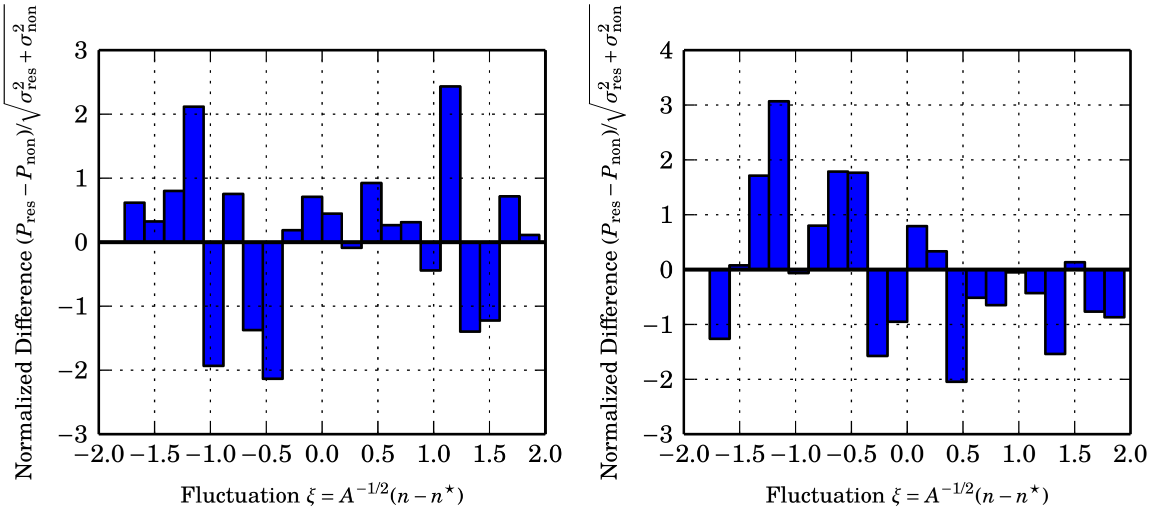


Figure 3.7: Differences between the distributions obtained with and without resampling, normalized by the two-parameter standard deviations $\sqrt{\sigma^2_{res} + \sigma^2_{nonres}}$ *for each bin. Parameters:* $A = 800$, $B = 3$, $C = 1$. *Left:* $\tau = 0.1$. *Right:* $\tau = 1$.

pling and without using resampling still agree well with each other, with no obvious systematic differences.

Based on this data, we can tentatively conclude that weighted-ensemble resampling does not bias our estimate of the steady-state probability distribution of the delayed-degradation system. This is a stronger conclusion than the one for the simple production-degradation system. This data provides strong evidence that, even in the presence of delayed reactions, the WE resampling method does not have a biasing effect. It remains to be seen whether this conclusion holds in general for more complex delayed systems, although this data does provide good reason to be optimistic.

3.2.2 Delayed Joint Probability Distribution

The chemical master equation (Equation (1.3)) written for the delayed-degradation system takes the form:

$$\frac{\partial}{\partial t}\mathrm{P}(n,t) = A(\hat{E}^{-1}-1)\mathrm{P}(n,t) + Bn(\hat{E}-1)\mathrm{P}(n,t) + \sum_{m=0}^{\infty} H(n)Cm(\hat{E}-1)\mathrm{P}_2(n,t;m,t-\tau) \quad (3.6)$$

with $\hat{E}$ representing the unit shift operator acting only on n, i.e.

$$\hat{E}f(n,t;m,t-\tau) = f(n+1,t;m,t-\tau).$$

The equation suffers from a closure problem due to the non-Markovian properties of this system. The question is how to model the joint probability distribution $P_2(m,t;n,t-\tau)$. Writing an equation for the time evolution of this distribution only results in a dependence on higher-order distributions. Analytical models of the delayed-degradation system therefore need to model this joint distribution in some other way.

To cast Equation (3.6) in a slightly more convenient form, we can eliminate the sum over m by replacing the joint probability distribution with the **conditional average**:

$$\langle m,t-\tau|n,t\rangle = \frac{\sum_m mP_2(n,t;m,t-\tau)}{\sum_m P_2(n,t;m,t-\tau)} \quad (3.7)$$

The master equation then takes the form [15]:

$$\frac{\partial}{\partial t}\mathrm{P}(n,t) = A(\hat{E}^{-1}-1)\mathrm{P}(n,t) + Bn(\hat{E}-1)\mathrm{P}(n,t) + \\ C(\hat{E}-1)(\langle m, t-\tau | n, t\rangle \mathrm{P}(n,t)) \quad (3.8)$$

One method for modeling the delayed joint distribution is used in [13], a type of mean-field approximation where events at time t are assumed to be decorrelated from events at time $t-\tau$. This assumption implies $\langle m, t-\tau | n, t\rangle = \langle m\rangle$ – in other words, that the conditional average of $m(t-\tau)$ is a constant that does not depend on $n(t)$. The authors of [13] note that in order for this assumption to hold, the coupling between events at time t and $t-\tau$, which is controlled by the value of C, should be relatively small. The approximation could very well break down close to the bifurcation.

One way to evaluate the conditions under which this assumption holds is to use stochastic simulation, which does not suffer from the closure problem encountered in analytical models. It is possible to compute the delayed joint distribution based on a single trajectory generated by the SSA. Interpreting the sequence (n_i, t_i) as a continuous-time trajectory $n(t)$, the probability $P_2(p,t;q,t-\tau)$ is proportional to the amount of time the system spends in the state $n(t) = p; n(t-\tau) = q$. More formally,

$$P_2(p,t;q,t-\tau) = \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} \delta_{p,n(t)}\delta_{q,n(t-\tau)}\,\mathrm{d}t \quad (3.9)$$

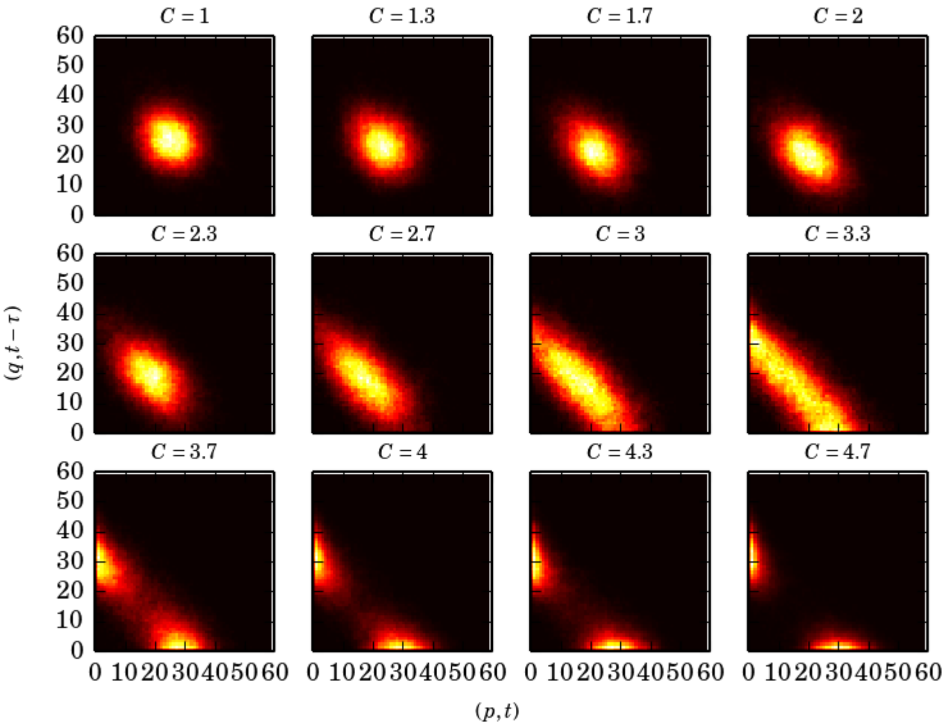


Figure 3.8: Behavior of the delayed joint distribution as the parameter C is varied. The parameters $A = 100$, $B = 3$, and $\tau = 20$ were kept fixed; each distribution was computed from $T = t_1 - t_0 = 800$ time units of a single SSA trajectory with initial state $n_0 = 0$.

where t_0 is at least τ time units later than the trajectory's starting time; $\delta_{i,j}$ is the Kronecker delta symbol. The conditional average is then straightforward to calculate from Equation (3.7).

Figure 3.8 shows the qualitative behavior of the delayed joint distribution as the parameter C (the coupling between events at time t and time $t - \tau$) is varied. For very weak coupling, the distribution is essentially an uncorrelated two-dimensional Gaussian. As the coupling

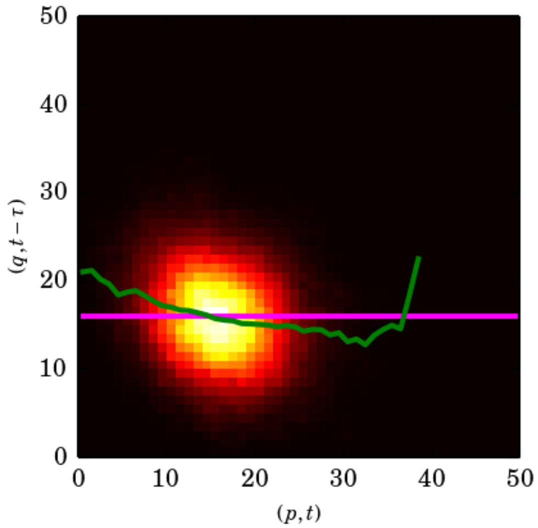


Figure 3.9: Conditional average of the delayed joint distribution with $A = 100$*,* $B = 4.5$*,* $C = 2$*, and* $\tau = 1$ *computed from* $T = 800$ *time units of an SSA trajectory, showing approximate independence of the conditional average of* q *from* p*. The flat magenta line indicates the average value of* q *over the entire trajectory. The delayed-joint distribution is provided in the background for context; the conditional average is simply a moment of that distribution.*

is increased, however, the central peak stretches and eventually splits into two. For very strong coupling, the distribution is characterized by two completely separated peaks.

We can evaluate the extent to which the mean-field approximation applies to the system by computing the conditional average of the SSA-derived distribution. The conditional average $\langle q, t-\tau | p, t \rangle$ for a reasonably small value of C is shown in Figure 3.9. The mean-field approximation implies a conditional average independent of p, or a horizontal line on the plot in Figure 3.9. The approximation appears reasonably

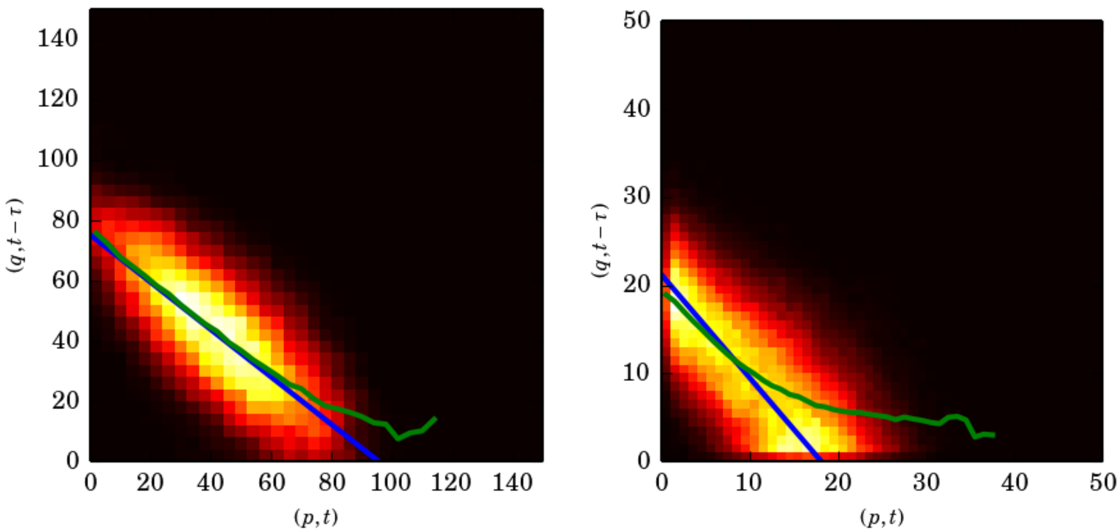


Figure 3.10: Conditional average of the delayed joint distribution near the bifurcation, showing a breakdown of the mean-field approximation and saturation for large values of p. *Left:* $A = 400$, $B = 4.5$, $C = 5$, *and* $\tau = 1$. *Right:* $A = 100$, $B = 4.5$, $C = 5.75$, *and* $\tau = 1$. *The straight blue line indicates the prediction of Equation* (3.10). *Each distribution was computed from* $T = 800$ *time units of an SSA trajectory.*

good for the value of C shown there; we could expect better agreement with a smaller value of C.

Closer to the bifurcation, however, this approximation completely breaks down. Figure 3.10 shows an example for a value of C just below the stochastic bifurcation point. The highly correlated nature of the delayed joint distribution in this regime evidently requires a different description from the mean-field approximation.

Another model for the conditional average can be made in the Langevin limit and near the stochastic bifurcation point. In terms of the normalized fluctuation $\xi = \frac{n-n^\star}{\sqrt{A}}$ defined in Section 3.2.1, the conditional average in this limit is the linear function [15]

$$\langle \xi_\tau, t-\tau | \xi, t \rangle = \left(\frac{\omega \cos(\omega\tau) - B\sin(\omega\tau)}{\omega + C\sin(\omega\tau)} \right) \xi \tag{3.10}$$

with the Hopf frequency $\omega = \sqrt{C^2 - B^2}$, so this equation is only valid for $C > B$. The derivation of this formula also assumes the system is below the stochastic bifurcation point[1], i.e. that the delayed-joint distribution is unimodal.

This formula agrees well with the computed conditional average within a certain range of p, as Figure 3.10 shows. However, an interesting effect occurs for large values of p, where the computed conditional average deviates from Equation 3.10. In the complete stochastic description modeled by the SSA the number n cannot go below zero. This means that the conditional average must eventually flatten out, or saturate, for large values of p. The Langevin limit does not generally impose this constraint, so the conditional average as predicted by Equation (3.10) does go below zero above some value of p.

The saturation effect is most important in small systems because they have lower average concentrations and spend more of their time close to $n = 0$. This observation is consistent with the fact that the Langevin

[1] The stochastic bifurcation point is in general different from the bifurcation point that can be derived from a deterministic description of this system. Stochastic effects shift the bifurcation, opening up a “window” in which Equation (3.10) can be applied.

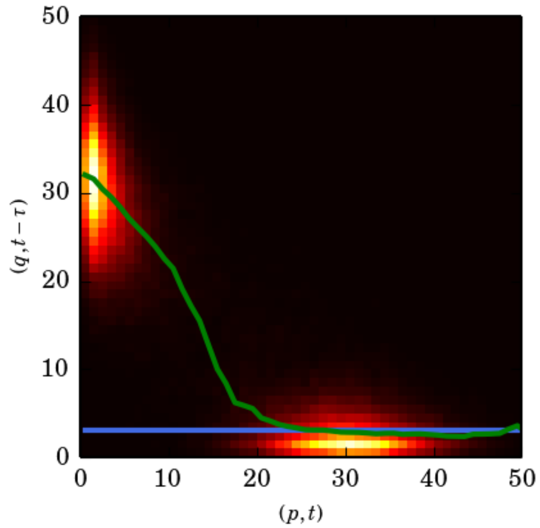


Figure 3.11: Conditional average of the delayed-joint distribution with $A = 100$, $B = 3$, $C = 5$, and $\tau = 20$ computed from $T = 800$ time units of an SSA trajectory, showing saturation to a constant value. The flat blue line indicates the average value of q only over the lower peak.

limit, which in this case fails to account for saturation, becomes more accurate as the system volume is increased.

Interestingly enough, the saturation effect appears to partially restore the mean-field approximation in the oscillatory mode. As Figure 3.11 shows, the conditional average for an oscillating system becomes essentially constant for large enough values of p. Thus, we can cautiously assume (for large enough values of p) that the conditional average $\langle q, t - \tau | p, t \rangle$ is independent of p far enough from the bifurcation in *both* the weak and strong coupling limits.

Chapter 4

Conclusions and Future Work

The weighted-ensemble stochastic simulation algorithm is a useful tool for studying genetic networks with delays. The SSA explicitly treats the discreteness and stochasticity that become significant in small systems, allowing us to describe them in a statistically exact manner. In addition, it is straightforward to extend the SSA to include delayed reactions. The WE resampling method ensures that we sample the system's state space evenly, enabling us to simulate realistic, complex genetic networks efficiently. Results from this work indicate that the resampling maintains statistical accuracy, i.e. that we can obtain an unbiased estimate of the measured distribution, even in systems containing delayed reactions.

The results of Section 3.2.2 illustrate another important application of the SSA as a statistically exact numerical simulation. We can use it to assess approximations or assumptions that analytical models make to study a system's behavior. We can also use it as a stand-alone tool to explore how a system behaves. The example discussed in Section 3.2.2 was the delayed-degradation system, where we saw that the behavior of the system is decorrelated on the timescale τ as long as the delayed reaction fires relatively rarely. This approximation breaks down near the bifurcation point. However, we saw that the approximation is restored in a sense on the other side of the bifurcation. This is a conclusion that is not at all obvious from an analytical or intuitive description of the system.

The next step in this project is to model systems that are also realizable experimentally. A comparison between numerical predictions and experimental results would serve as an excellent check on the validity of the assumptions underlying the application of the SSA to genetic networks.

4.1 Modeling Crowded Environments

The most important assumption underlying the SSA and the chemical master equation is that the chemical system is well-stirred (homogeneous) and behaves like an ideal gas [14]. Both of these assumptions often break down, to some extent, in living cells. First, cells are highly inhomogeneous environments, with various internal membranes (especially, in eukaryotic cells, the nuclear membrane) hindering the free movement of molecules throughout the cell. Second, the high concentrations of bulky molecules such as proteins would also be expected to slow down molecular diffusion [21].

There are several possible ways to model these deviations from the ideal-gas assumption. One way would be to add a phenomenological retardation factor that decreases each reaction's propensity in the presence of high concentrations of other molecules. Another way would be to run the SSA on a spatial grid of cells coupled by a diffusion law.

Although it may intuitively seem improper to model a complex cellular environment as a simple ideal gas, this method still seems to be the dominant one for modeling genetic regulatory networks. Comparison with experimental data should be the most useful guide in determining whether the ideal-gas assumption is in fact justified or whether one of the above strategies needs to be applied to realistically model genetic networks in cells.

Appendix A

Technical Notes

I implemented all computational algorithms used in this work in Python version 3.4.0 (`http://python.org`) using the packages SciPy 0.13.3 and NumPy 1.8.1 (`http://scipy.org`). I used the IPython interactive scientific computing system [22] to help develop and test my code, along with the package Matplotlib (`http://matplotlib.org`) to explore and visualize my results. Pseudorandom numbers for sampling from probability distributions in my implementation of the SSA (Section 2.1) came from NumPy, which uses the Mersenne twister algorithm.

All the code written for this project, including Python modules and IPython notebooks, is publicly available at `https://github.com/max-veit/wessa-delay`. The code is versioned in a Git repository. The version of the code used to obtain the results in this thesis is recorded under the commit with the SHA1 hash `7b259428f475041697e7ed8ddca7719518d83dbd` and tagged with the name `thesis-final`. The corrections made to the code after this thesis was first submitted are recorded under the tag `thesis-code-corr`.

Bibliography

[1] E. H. Davidson, Nature **468**, 911 (2010), URL http://dx.doi.org/10.1038/nature09645.

[2] J. Kondev, Phys. Today **67**, 31 (2014), URL http://dx.doi.org/10.1063/PT.3.2276.

[3] H. Salgado, G. Moreno-Hagelsieb, T. F. Smith, and J. Collado-Vides, Proc. Nat. Acad. Sci. USA **97**, 6652 (2000), URL http://www.pnas.org/content/97/12/6652.abstract.

[4] S. M. Reppert and D. R. Weaver, Annu. Rev. Physiol. **63**, 647 (2001), pMID: 11181971, URL http://www.annualreviews.org/doi/abs/10.1146/annurev.physiol.63.1.647.

[5] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, Nat. Rev. Genet. **2**, 268 (2001), URL http://dx.doi.org/10.1038/35066056.

[6] A. S. Khalil and J. J. Collins, Nat. Rev. Genet. **11**, 367 (2010), URL http://dx.doi.org/10.1038/nrg2775.

[7] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, Nature **456**, 516 (2008), URL http://dx.doi.org/10.1038/nature07389.

[8] P. Hanggi, H. Grabert, P. Talkner, and H. Thomas, Phys. Rev. A **29**, 371 (1984), URL http://link.aps.org/doi/10.1103/PhysRevA.29.371.

[9] D. T. Gillespie, Annu. Rev. Phys. Chem. **58**, 35 (2007), URL http://www.annualreviews.org/doi/abs/10.1146/annurev.physchem.58.032806.104637.

[10] P. Smadbeck and Y. N. Kaznessis, Proc. Nat. Acad. Sci. USA **110**, 14261 (2013), URL http://www.pnas.org/content/110/35/14261.abstract.

[11] M. Kaern, T. C. Elston, W. J. Blake, and J. J. Collins, Nat. Rev. Genet. **6**, 451 (2005), URL http://dx.doi.org/10.1038/nrg1615.

[12] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, Science **297**, 1183 (2002), URL http://www.sciencemag.org/content/297/5584/1183.abstract.

[13] D. Bratsun, D. Volfson, L. S. Tsimring, and J. Hasty, Proc. Nat. Acad. Sci. USA **102**, 14593 (2005), URL http://www.pnas.org/content/102/41/14593.abstract.

[14] D. T. Gillespie, J. Chem. Phys. **113**, 297 (2000), URL http://scitation.aip.org/content/aip/journal/jcp/113/1/10.1063/1.481811.

[15] J. Wentworth, M. Gaudreault, and J. Viñals (2012), unpublished research notes regarding the delayed-degradation system.

[16] R. M. Donovan, A. J. Sedgewick, J. R. Faeder, and D. M. Zuckerman, J. Chem. Phys. **139**, 115105 (2013), URL http://scitation.aip.org/content/aip/journal/jcp/139/11/10.1063/1.4821167.

[17] M. Ullah and O. Wolkenhauer, Wiley Interdiscip. Rev. Sys. Biol. Med. **2**, 385 (2010), ISSN 1939-005X, URL http://dx.doi.org/10.1002/wsbm.78.

[18] D. T. Gillespie, J. Comp. Phys. **22**, 403 (1976), ISSN 0021-9991, URL http://www.sciencedirect.com/science/article/pii/0021999176900413.

[19] G. Huber and S. Kim, Biophys. J. **70**, 97 (1996), ISSN 0006-3495, URL http://www.sciencedirect.com/science/article/pii/S0006349596795528.

[20] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, J. Chem. Phys. **132**, 054107 (2010), URL http://scitation.aip.org/content/aip/journal/jcp/132/5/10.1063/1.3306345.

[21] F. C. Simmel, Nat. Nano. **8**, 545 (2013), m3: News and Views, URL `http://dx.doi.org/10.1038/nnano.2013.156`.

[22] F. Pérez and B. E. Granger, Comput. Sci. Eng. **9**, 21 (2007), ISSN 1521-9615, URL `http://ipython.org`.