Simplifying and optimizing the stochastic simulation of rare biochemical events

by

Max C Klein

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

January, 2019

Abstract

foo

Acknowledgments

My deepest thanks to my advisor, my committee members, my parents, and all of the others who bore with me during a difficult time. Your support has meant the world to me.

Table of Contents

Chapter 1

Introduction

> *One of the great arguments of the day was vitalism versus mechanism, a disguised form of the old and continuing debate between those, including the religious, who believe the world has purpose and those who believe it operates automatically and by chance... The German chemist who scoffed in 1895 at the "purely mechanical world" of "scientific materialism" that would allow a butterfly to turn back into a caterpillar was disputing the same issue, an issue as old as Aristotle.*
>
> – Richard Rhodes, *Making of the Atomic Bomb*[1]

Are living organisms special? Are biological molecules inherently unique, or do they obey the same laws of chemistry and physics as every other molecule? Since the time of Mendel[2] and Cajal[3] the scientific consensus has been converging towards the latter view: that life is made of the same material as the rest of the world. In other words, that the information encoded in an organism's genotype and (more broadly) biochemical state defines the organism's phenotype (*i.e.* it's appearance and behavior). Over the course

of the 20th century this materialistic viewpoint has become an implicit assumption that underlies all current work done in the life sciences. However, this view is largely taken on faith, as a complete mechanistic link between biochemistry and behavior has been established for only a small subset of cellular systems. Though there is now an extensive (and ever growing) catalog of the biochemical elements from which living systems are formed, the mechanisms by which they interact remain in general unexplained. The aim of this thesis is to present work towards a general method of mechanistically linking biochemistry to phenotype.

1.1 Models of biochemistry

The question at hand is one of dynamics. How does the biochemistry of a cell "unfold" over time to produce a phenotype*? Models, and modeling, form the cornerstone of a workable approach for explaining large-scale biological phenomena in terms of what goes on at the smallest scale. First, we need to establish a theoretical framework in which to reason about biochemistry. Any chemical reaction can be written in the form:

$$\alpha_1 a_1 + \alpha_2 a_2 + \ldots \longrightarrow \beta_1 b_1 + \beta_2 b_2 + \ldots, \tag{1.1}$$

where $\{a_1, a_2, \ldots\}$ and $\{b_1, b_2, \ldots\}$ are the reactants and products, and $\{\alpha_1, \alpha_2, \ldots\}$ and $\{\beta_1, \beta_2, \ldots\}$ are their respective stoichiometries. In some cases, a chemical species may appear on both sides of the reaction with the same stoichiometry,

*Equivalently, imagine that you are given a perfect description of the contents of a cell at one moment in time. What can you then say about the contents of that cell at some moment in the future? What can you say about the cell's behavior during all of the moments in between?

as so:

$$X \longrightarrow X + Y.$$

This implies that X is required for the reaction, but is not consumed by it.

To give a concrete example, consider protein expression from a gene. A simple model of this process can be derived from the central dogma[4,5]: say there is a segment of DNA that comprises the gene g_p. g_p is transcribed to produce an RNA rna_p. In turn, rna_p is then translated to a protein p. Both rna_p and p also degrade. The reactions that describe this system can be written in terms of Eq 1.1 as so:

$$\begin{aligned} g_p &\longrightarrow g_p + rna_p \\ rna_p &\longrightarrow rna_p + p \\ rna_p &\longrightarrow \varnothing \\ p &\longrightarrow \varnothing \end{aligned} \quad . \tag{1.2}$$

In a sense, a list of reactions like Eq 1.2 give a complete description of a system. This level of description can be thought of as forming a qualitative model. Qualitative models can be used to answer questions about what is possible in a system, or how a particular event occurs.

The same approach used to come up with Eq 1.2 can also be used to build more complex, regulated models of expression. For example, say that p is able to dimerize, and that p_2 dimers are able to bind back to g_p and repress further

transcription. The reactions that describe the system would then be:

$$\begin{aligned} g_p &\longrightarrow g_p + rna_p \\ rna_p &\longrightarrow rna_p + p \\ p + p &\longleftrightarrow p_2 \\ p_2 + g_p &\longleftrightarrow p_2 \cdot g_p \\ rna_p &\longrightarrow \varnothing \\ p &\longrightarrow \varnothing \end{aligned} \quad . \tag{1.3}$$

1.2 Epigenetic landscapes and phenotypes

Low-level models of biochemistry such as Eq 1.2 can be linked to higher-level phenomena via the concept of the epigenetic landscape. Waddington introduced his idea of the epigenetic landscape[6,7] by first asking the reader to imagine a cell as represented by a high-dimensional space. We call this space the cell's state space. Each dimension of the state space has a one-to-one correspondence with one of the species of metabolites or biomolecules present in the cell. The (non-negative, integer valued) coordinate along each dimension is the count of the corresponding molecule. Thus, each point in the complete state space uniquely defines a possible state of the cell. Just as there can be thousands, or tens of thousands, of distinct chemical species present in a cell, so too can a state space have thousands of dimensions.

The possible states of a cell are combinatorially vast. Simply enumerating

them all would be a herculean computational task. Thus, it is legitimate to ask: what insight about a cell can actually be gained by thinking about it in terms of its state space? An analogy to the study of protein folding is useful here. Levinthal's paradox[8] says that there are more possible states for a protein to be in than atoms in the universe. Yet somehow proteins still fold. Similarly, cells tend to remain in homeostasis with respect to a particular phenotype (or sometimes to transition from one phenotype to another in an orderly fashion). In both cases, physical forces conspire to limit the occupied states to restricted regions. The epigenetic landscape is a description of the states that tend to be occupied, and of the forces that drive a cell to those states. A single "neighborhood" of states on the landscape constitutes a phenotype.

Waddington originally developed the idea of epigenetic landscapes as a way to explain the constrained diversity he observed in developing organisms. "Diversity" in the sense that embryonic cells have many possible end states (in terms of the cell type of their lineage in the mature organism), and "constrained" in the sense that even harsh chemical perturbations often shift those end states only slightly. As he described it, the landscape of a developing cell is like rough, hilly terrain that is covered in divots connected by valleys. The divots represent the various possible phenotypes, and the valleys the transition pathways in between them. The developing cell, then, is like a ball that starts at a a high point on this terrain (see Fig 1.1). The robustness of the development process at any given moment can be thought of as analogous to the steepness of the landscape in the cell's immediate vicinity. As the cell (or its lineage) travels downhill, it proceeds through various phenotypes and

the branching paths in-between. Eventually it reaches the bottom of the hill, along with its terminal cell type.

In the decades since Waddington first proposed it, much work has gone into filling in the details of the theory behind epigenetic landscapes. In the modern view[9,10,11], the epigenetic landscape is a representation of the physics of a non-equilibrium system. It can be (approximately) split into two parts: a potential surface (analogous to the potential energy surface of an equilibrium system) and a probability flux. If the potential surface is equivalent to Waddington's rough terrain, then the probability flux is like a strong wind blowing across it. Because of this "wind", a cell's fate is not determined by its potential surface alone. In recent years epigenetic landscapes of various cellular systems, such as the cell cycle[12,13] and oncogenesis[14,15], have been constructed. These high-dimensional landscapes can be plotted (via projection) in terms of one or more species of interest. This gives a quantitative picture of how transitions in the population count of any given species drives transitions between phenotypes (see figure Fig 1.2).

1.3 Quantitative models and networks

Given that an epigenetic landscape can be used to link biochemistry to phenotype, the question then is how to calculate one. A qualitative model is not enough. What is needed is a quantitative model, one that can reproduce the actual species counts. In order to build a quantitative model of biochemical state and behavior, 4 pieces of information are required:

1. The identity of the distinct interacting biochemical elements. These are

often referred to as chemical species, or just species.

2. The list of reactions in which the chemical species participate.

3. The rate laws that describe how quickly each reaction occurs.

4. The quantity of each species present in the system. These are dynamical systems under consideration, so precise quantities can only be spoken of with respect to a particular moment in time (such as an initial time $t = 0$).

Once these 4 pieces have been determined, the next step is to combine them into a cohesive, mathematically tractable model. The standard way to do so is to build a network.

There are many different representations of biochemical networks, each with their own strengths and weaknesses. A simple network representation of our protein expression model can be seen in Fig 1.3. Here, the protein expression model is shown as a digraph (directed graph). Like any network, a digraph consists of a set of nodes and the set of edges which connect them. Additionally, each edge in a digraph encodes a direction, such that they start at a reactant node and end at a product node.

Simple digraph representations are useful since they offer an intuitable picture of the lists of species and reactions that make up a model. Another useful network representation is the Petri net. Originally developed by Petri in the 1960's[16] to model linked chemical reactions, Petri nets offer a natural way to model agent-based processes in general[†]. Petri nets were first applied to

[†]When applying Petri nets to biochemical systems, the chemical species are the agents. This is in the sense that each molecule acts independently.

problems in systems biology around the turn of the century[17], and have since been extensively developed for biological applications[18,19,20]. Though less easy to understand at a glance than simple digraphs, Petri nets have the virtue of being able to encode unambiguous descriptions of biochemical systems. Due to this formal rigor, once set up they can be mechanically transcribed into various mathematical forms that can then be used as input for many different simulation techniques.

Petri nets are bipartite digraphs.“Bipartite" refers to the fact that every Petri net contains two distinct sets of nodes: one set, called the places, represent chemical species, and the other, called the transitions, represent chemical reactions. For every reaction that a species participates in as a reactant there is an edge that starts at the corresponding place and ends at the corresponding reaction. Similarly, for every reaction product there is an edge that starts at the corresponding transition and ends at the corresponding place. The weight of each edge is the stoichiometry of the attached species (whether as product or reactant) with respect to the attached reaction. Additionally, a Petri net has a marking, a list that contains the initial quantities of each species.

Figure 1.4 shows a Petri net representation of the simple expression model of Eq 1.2. In addition to the graphical representation, a Petri net can be uniquely specified as a set of matrices $\{P,\ T,\ M,\ Pre,\ Post\}$[21]. For the simple

expression model these matrices would be:

$$P = \begin{pmatrix} g_p \\ rna_p \\ p \end{pmatrix},\ T = \begin{pmatrix} \text{transcription} \\ \text{translation} \\ rna_p \text{ decay} \\ p \text{ decay} \end{pmatrix},\ M = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$
$$Pre = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ Post = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{1.4}$$

where P is the list of places/species, T is the list of transitions/reactions, M is the list of markings corresponding to P, Pre is a $M \times N$ matrix (where M is the count of reactions, and N is the count of unique species) in which Pre_{ij} gives the weight of the edge connecting the ith species to the jth reaction (equivalently, the reactant stoichiometry), and $Post$ is the same thing as Pre except that $Post_{ij}$ gives the weight of the edge connecting the jth reaction to the ith species.

For the more complex regulated expression system described in Eq 1.3, the

equivalent $\{P,\ T,\ M,\ Pre,\ Post\}$ matrices would be:

$$P = \begin{pmatrix} g_p \\ rna_p \\ p \\ p_2 \\ p_2 g_p \end{pmatrix},\ T = \begin{pmatrix} \text{transcription} \\ \text{translation} \\ rna_p \text{ decay} \\ p \text{ decay} \\ p \text{ dimerization} \\ p \text{ dedimerization} \\ p_2 + g_p \text{ binding} \\ p_2 g_p \text{ unbinding} \end{pmatrix},\ M = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$Pre = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},\ Post = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.5}$$

Fig 1.5 shows a plot of the actual Petri net that the above matrices describe.

1.4 Deterministic simulation

1.4.1 ODE models of biochemistry

The traditional way to simulate biochemical systems begins with constructing a set of ordinary differential equations (ODEs). This type of simulation is commonly referred to as deterministic simulation (as opposed to stochastic simulation, which is discussed later in Sec 1.5). If there are N chemical species in a system, for each distinct species X_j there will be an equation of the form:

$$\frac{dX_j}{dt} = f_j\left(X_1, X_2, ..., X_N\right).$$

On the right hand side of the equation is a derivative that represents the rate in change over time in the quantity of X_j. On the left hand side is a function of the quantity of every species (including X_j). The exact form of f_j will depend on the type (and precise mathematical formulation) of the chemical kinetics being modeled.

Given the Petri net representation of a system, it is straightforward to construct the set of ODEs. Say that the system involves N species participating in M reactions. First one constructs the M rate laws $r_i(X)$. The rate law $r_i(X)$ tells you how quickly each reaction i is occurring. Traditionally, the $r_i(X)$ are given a form based on mass action kinetics (which can be traced back to a set of papers published in the 1860s[22]). In the mass action view, the rate at which any reaction occurs is directly proportional to the quantity of each reactant. This implies that every reaction has an associated rate law $r_i(X)$ of the form:

$$r_i(X) = k_i X_1^{p_{i1}} X_2^{p_{i2}} ... X_N^{p_{iN}} = \prod_{j=1}^{N} X_N^{p_{ij}}, \tag{1.6}$$

where k_i is a rate constant, and p_{ij} is a shorthand for the ijth entry of the Pre matrix.

Next, one determines the stoichiometry matrix S. Essentially, the stoichiometry matrix is the difference of the $Post$ and Pre matrices. The exact definition of S varies in the literature, but for our purposes it will be convenient to define S as the transpose of the difference:

$$S = (Post - Pre)^{\mathsf{T}}.$$

Thus, S will be a $N \times M$ matrix. With the rate laws $r_i(X)$ and the stoichiometry

matrix S in hand, the entire set of N ODEs can be written as a single matrix equation:

$$\frac{dP}{dt} = Sr, \tag{1.7}$$

where P is the places matrix, $\frac{dP}{dt}$ is a column vector of the ODEs, and r is a column vector in which the ith entry is the rate law $r_i(X)$.

The nature of the compact form given in Eq 1.7 is most easily explained with an example. For our simple expression system (described in Eq 1.2 and given in Petri net matrix form in Eq 1.4), the column vector of rate laws r can be found from Eq 1.6:

$$r = \begin{pmatrix} k_1 \cdot g_p^1 \cdot rna_p^0 \cdot p^0 \\ k_2 \cdot g_p^0 \cdot rna_p^1 \cdot p^0 \\ k_3 \cdot g_p^0 \cdot rna_p^1 \cdot p^0 \\ k_4 \cdot g_p^0 \cdot rna_p^0 \cdot p^1 \end{pmatrix} = \begin{pmatrix} k_1 \cdot g_p \\ k_2 \cdot rna_p \\ k_3 \cdot rna_p \\ k_4 \cdot p \end{pmatrix}$$

and the stoichiometry matrix S is:

$$S = \left[\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right]^\intercal = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}^\intercal = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}.$$

The set of ODEs that describe the system is then:

$$\begin{pmatrix} \frac{d}{dt} g_p \\ \frac{d}{dt} rna_p \\ \frac{d}{dt} p \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} k_1 g_p \\ k_2 rna_p \\ k_3 rna_p \\ k_4 p \end{pmatrix} = \begin{pmatrix} 0 \\ k_1 g_p - k_3 rna_p \\ k_2 rna_p - k_4 p \end{pmatrix}.$$

Solving[‡] the above system of ODEs yields:

$$\begin{pmatrix} g_p \\ rna_p \\ p \end{pmatrix} = \begin{pmatrix} M_1 \\ \frac{k_1 M_1}{k_3} + e^{-tk_3}\left(M_2 - \frac{k_1 M_1}{k_3}\right) \\ \frac{k_1 k_2 M_1}{k_3 k_4} + e^{-tk_3}\frac{(k_2(k_1 M_1 - k_3 M_2))}{k_3(k_3 - k_4)} + e^{-tk_4}\left(M_3 + \frac{k_2(k_4 M_2 - k_1 M_1)}{k_4(k_3 - k_4)}\right) \end{pmatrix}, \tag{1.8}$$

where M_j is the initial marking (*i.e.* quantity) of species j. For any given marking M and set of rate constants k_i, the expressions of Eq 1.8 can be used to find the quantity of each species at any given time t.

Often, a modeler is not concerned with the initial behavior of a system[§], but only with the behavior of the system over long periods of time. In these cases, solutions such as those found in Eq 1.8 can be simplified by considering their value in the limit $t \to \infty$, also called the steady state limit. The only terms in Eq 1.8 that explicitly depend on time are exponential decay terms of the form e^{-tx}. These exponential decay terms quickly approach 0 as t increases, yielding:

$$\begin{pmatrix} g_p \\ rna_p \\ p \end{pmatrix} \approx \begin{pmatrix} M_1 \\ \frac{k_1 M_1}{k_3} \\ \frac{k_1 k_2 M_1}{k_3 k_4} \end{pmatrix}.$$

[‡]The complete details of solving a system of ODEs is outside of the scope of this thesis. In general, it suffices that a solution can easily be obtained using a computational algebra system such as Mathematica[23].

[§]As far as reproducing experimental results go, the initial behavior of a system is usually irrelevant. When studying biological systems, it is typical for the system to preexist any experimental observation (though an obvious exception would be any stop-flow experiment).

1.4.2 The benefits and shortcomings of deterministic simulations

Deterministic, ODE based approaches such as the one given in the previous section were first used to model biological systems during the early 1900's[24,25,26,27]. Since then, ODEs have seen wide use in the simulation of biological systems. Most such systems of ODEs do not have a closed-form solution as in Eq 1.8. Instead, numerical integration[28] can be used to determine the quantity of each species up to a finite time. Even when numerical integration is required, ODE based methods tend to be some of the most computationally efficient and easy to implement techniques, contributing to their popularity.

Fig 1.6 shows a realization of an ODE simulation of our simple expression system. The expression in Eq 1.8 was used to find the concentration of protein at every time point up to ten hours. On the right hand side of the figure is a histogram of the count of protein along the time series. This can be thought of as an empirical estimation of the epigenetic landscape of the simple expression system. Thus, the ODE simulation predicts that the landcape of this system effectively consists of a single state at protein count $= 50$.

If we were to repeat this simulation, the exact same result would be produced. This highlights a key feature of ODE simulations: they're deterministic, in the sense that, given a particular system and set of initial conditions, the method will always produce exactly the same result (at least up to numerical accuracy, in the cases where numerical integration is required). It is because

of this reproducibility of outcome that ODE simulations are also called deterministic simulations.

The theoretical formulation of the ODE approach relies on a number of assumptions[29]. In particular, every chemical species involved in a reaction is assumed to be distributed across the reaction volume in a completely homogenous fashion. The concentration of any given species at any point in space is equal to the concentration of that species at any other point. This is in effect a continuum view, in which chemical species are not made of discrete particles but instead can be treated as infinitely divisible interacting fluids.

In the words of van Kampen, at the scale of test tubes the continuum assumptions are "actually better obeyed than might be expected"[30]. At significantly smaller scales, such as those of a single cell, these assumptions break down. At some point it becomes no longer reasonable to assume that a chemical substance is an infinitely divisible fluid. Instead, one begins to have to account for effects[31] arising from the influence of the single molecules of which chemical substances are truly made. In addition to small scales, these single molecule effects also become prominent when a substance is present in a very low concentration. In this type of situation it becomes more appropriate (and more accurate) to measure quantities in terms of particle count, also sometimes called copy number.

The assumptions of ODE modeling break down to some extent when applied to any gene expression system. This is due to the fact that the DNA of each gene is typically present at very low copy numbers (just 2-4 for most

genes in a diploid organism) in each cell. Moreover, many of the other components of expression systems, such as specific RNAs and proteins, also have low copy number. Thus ODE models cannot recapitulate the full behavior of gene expression. Even for our simple expression model, deterministic simulation fails to reproduce many of its details (as shown in Fig 1.7 and discussed in Sec 1.6). Alternative methods have to be used in order to capture the full dynamics through which the information encoded in genes becomes proteins.

1.5 Stochastic simulation

> *The justification for using the stochastic approach, as opposed to the mathematically simpler deterministic approach, was that the former presumably took account of fluctuations and correlations, whereas the latter did not... in the deterministic formulation no distinction is made between the average of a product and the product of the averages; i.e., it is automatically assumed that* $\langle x_i x_j \rangle = \langle x_i \rangle \langle x_j \rangle$. *For* $i \neq j$ *this assumption nullifies the effects of correlations, and for* $i = j$ *it nullifies the effects of fluctuations.*
>
> – Daniel T Gillespie, *A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions*[32]

Stochastic simulation is a particularly robust method for recreating the behavior of interacting biomolecules. The stochastic simulation method stems from work done by Gillespie. In his seminal 1976 paper "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions"[32], Gillespie laid out the theoretical framework of stochastic chemical

kinetics (which was still in its infancy[33,34,35] at that point) and proposed the first practical algorithm for simulating stochastic chemical systems.

1.5.1 The theory of stochastic simulation

The essential assumption of stochastic chemical kinetics is that for every reaction R_i there exists a constant c_i such that:

$$c_i dt \equiv \text{average probability that any set of competent}^{\P} \text{ molecules react per } R_i \text{ during the time interval } (t, t+dt]. \tag{1.9}$$

It can be shown[36] that for any well stirred system that is also at thermal equilibrium, the condition in Eq 1.9 will be true for any reaction that follows mass-action kinetics.

The continuum chemical kinetics used in deterministic simulations assumes that each chemical species is homogeneously distributed across the reaction volume under study. In this view, each substance is in some sense spread infinitely finely over the volume. In stochastic kinetics, this homogeneity assumption is recast as the well-stirred assumption, in order to account for the existence of discrete molecules. Now it is assumed that only the probability distribution of each species is uniformly distributed over the volume. This means that in the stochastic view, at any given moment each particle of each species in a volume is equally likely to be anywhere within that volume. It is reasonable to assume that a system is well-stirred if it is small enough and if non-reactive collisions greatly outnumber collisions that result in a reaction. This description happens to describe most cellular compartments reasonably

well. The fact that water molecules tend to vastly outnumber all other kinds within a cell is enough to satisfy the "non-reactive collisions" assumption[32].

Building on Eq 1.9, a propensity function (referred to by some sources as a hazard equation[21]) can be defined for each reaction R_i:

$$a_i(X)\,dt \equiv \text{probability that during the time interval } (t, t+dt] \text{ an } R_i \text{ reaction will take place,} \tag{1.10}$$

where X is a vector of the species counts $(X_1, X_2, \ldots, X_N)$. The propensity function of any reaction following mass action kinetics can be defined as:

$$a_i(X) = c_i h_i(X), \tag{1.11}$$

where $h_i(X)$ is the count of sets of molecules that can participate in reaction R_i. Given the relevant Petri net matrices, a propensity function from the family defined by Eq 1.11 can be constructed for each of the M reactions of a system with N species by means of a general formula[21]:

$$a_i(X) = c_i \prod_{j=1}^{N} \binom{X_j}{p_{ij}}, \tag{1.12}$$

where p_{ij} is the ijth element of the Petri net matrix Pre.

In theory, all systems can be modeled using the propensity functions for

these four elementary reaction types:

$$\begin{aligned}
a_i(X) &= c_i, & \varnothing &\xrightarrow{c_i} \text{product} && \text{(zeroth order)},\\
a_i(X) &= c_i X_j, & X_j &\xrightarrow{c_i} \text{product} && \text{(first order)},\\
a_i(X) &= c_i X_j X_k, & X_j + X_k &\xrightarrow{c_i} \text{product} && \text{(second order)},\\
a_i(X) &= c_i \frac{X_j(X_j-1)}{2}, & 2X_j &\xrightarrow{c_i} \text{product} && \text{(second order self)},
\end{aligned} \tag{1.13}$$

where the actual formulae in the left column were derived from Eq 1.12. Other reactions, such as those of order three or above, are considered nonphysical, since an instantaneous collision of any three molecules is extremely rare. Given a unit volume[||], the first three propensity functions above are identical to the corresponding deterministic rate laws, *i.e.* $a_i(X) = r_i(X)$ and $c_i = k_i$. There is no deterministic rate law equivalent to the second order self propensity function. For large enough X_j there is a reasonable approximation:

$$r_i(X) \approx \frac{c_i}{2} X_j^2 = k_i X_j^2, \qquad \text{(deterministic second order self)}, \tag{1.14}$$

where we've set $k_i = \frac{c_i}{2}$ above.

Technically, any reaction not in Eq 1.13 should be considered a compound reaction, and modeled by decomposition into 2 or more elementary reactions. In practice, however, it is not uncommon for modelers to use propensity

[||] see Wilkinson[21], chapter 6, for full details on converting between deterministic rate laws and stochastic propensities when dealing with real volumes.

functions with a wide variety of forms, such as Hill equations:

$$a_i(X) = \frac{X_j^h}{\kappa^h + X_j^h}, \qquad X_j \longrightarrow \text{product} \quad \text{(Hill equation)},$$

where h and κ are arbitrary constants. The use of such alternative propensity functions are then justified on an empirical basis (*e.g.* they reproduce a particular feature of the modeled system), rather than on the basis of stochastic kinetics per se[37].

1.5.2 The stochastic simulation algorithm

The Gillespie algorithm, or (as Gillespie himself refers to it[38]) the stochastic simulation algorithm (SSA), was first proposed by Gillespie in 1976[32]. Though there have been many improvements developed for the implementation[39,40,41,42] of the algorithm, the basic algorithm (specifically the direct method (DM) variant) itself remains 40 plus years later the gold standard of stochastic simulation.

SSA treats the underlying chemical system as a Markov process. This means that the next state that a system occupies is determined (in a probabilistic sense) solely by its present state[**]. The probability of each possible next state can be calculated by means of probability expressions of the form:

$$p\left(t', i|X\right) dt \equiv \text{probability that the next reaction in the system takes place in the instantaneous interval } (t+t', t+t'+dt] \text{ and is of the type } R_i. \tag{1.15}$$

[**]Although the framework is also flexible enough to allow factors that are explicitly time-dependent, such as externally applied chemical driving forces

The goal of SSA is to generate trajectories (*i.e.* time series) of the underlying stochastic chemical system by means of successively calculating and then drawing a random sample from the probability distribution $p\left(t', i|X\right)$ described in Eq 1.15.

The DM variant of SSA is based on the fact that $p\left(t', i|X\right)$ can be split into two independent distributions:

$$p\left(t', i|X\right) = p_1\left(t'|X\right) \cdot p_2\left(i|t', X\right)$$

Stochastic kinetic theory and basic probability can be used to derive expressions[32] for p_1 and p_2:

$$\begin{aligned} p_1\left(t'|X\right) &= Ae^{-At} \\ p_2\left(i|t', X\right) &= \frac{a_i}{A}, \end{aligned} \tag{1.16}$$

where

$$\begin{aligned} a_i &= a_i\left(X\right) \\ A &= \sum_{i=1}^{M} a_i\left(X\right), \end{aligned}$$

The actual DM algorithm can be though of as a "kinetic" Monte Carlo method[29] that works based on the generation of two uniform random numbers in each loop:

1. Calculate the current value of the propensity functions $a_i\left(X\right)$ and their sum $A\left(X\right)$, given the current state $\left(t, X\right)$ where t is the current time and X is a vector of the current species counts.

2. Sample p_1 and p_2 by drawing two uniform random numbers u_1 and u_2 from the unit interval:

   (a) Calculate t' as:

   $$t' = \frac{1}{A} \ln\left(\frac{1}{u_1}\right)$$

   (b) Calculate i as:

   $$i = \text{the first value of } i \text{ such that } \sum_{j=1}^{i} a_j > u_2 A$$

3. Update the current time as $t + t'$, update the current species counts as $X + S_{i*}$, where S_{i*} is the ith column of the stoichiometry matrix S (see Secs 1.3 and 1.4.1).

4. Write out any desired information about the state, then either terminate the simulation (given an appropriate condition has been met), or continue by returning to step 1.

After running (and recording) many such iterations, the data collected can be thought of as an exact realization of one replicate from the ensemble of the modeled system. In real computational experiments, typically many such replicates are run and the data from them are combined to produce a final analysis. For example, the epigenetic landscape of a system can be calculated by simply binning (*i.e.* into a histogram) the species count data of one or more replicates.

1.6 Deterministic vs stochastic

In general, deterministic simulation requires less computational effort than stochastic. So why should a computational scientist bother using stochastic simulation? The theoretical justifications discussed in the previous few sections can be put into concrete terms by comparing deterministic and stochastic simulations performed on the same model. Even for very simple gene regulatory networks (GRNs) the difference in the level of detail captured by each simulation type is clear.

A comparison of deterministic and stochastic simulations of our simple expression system (see Eq 1.2) can be seen in Fig 1.7. The top panel shows results from a version of the simple expression system with a relatively high average protein expression level (around $\sim 5 \cdot 10^3$). Under these conditions, the time series produced by the deterministic and stochastic simulations (shown in the top left panel) are in reasonably good agreement. As well, the epigenetic landscape predicted by stochastic simulation (shown in the top right panel) is in good agreement with the landscape predicted by deterministic simulation (which will just be a single peak around $\sim 5 \cdot 10^3$).

On the other hand, the deterministic and stochastic simulations diverge in the limit of low protein expression. The bottom two panels of Fig 1.7 show simulations of low expression variants of the simple expression system. Though the mean expression levels of the different simulation methods match (all are around $\sim$50 counts), the moment-to-moment behavior of the simulated systems do not. The deterministic simulations predict that the protein count will be constant. On the other hand, the stochastic simulations

predict (correctly) that the protein count will fluctuate significantly about the mean. Further, the deterministic simulations predict that there is no difference between the systems depicted in the middle and bottom panels, while the stochastic simulations again correctly show that the fluctuations in the bottom variant will be significantly larger. This agrees with the analysis of Ozbudak and coworkers[43]. They defined the fluctuation strength in terms of the Fano factor $\frac{\sigma_p^2}{\langle p \rangle}$, then showed that it will be equal to:

$$\frac{\sigma_p^2}{\langle p \rangle} = 1 + \frac{k_2}{k_3 + k_4} \tag{1.17}$$

The simulations in Fig 1.7 are clear examples of one of the key weaknesses of deterministic simulation: it does not capture fluctuations correctly. These kinds of species count fluctuations are often referred to in the literature as "noise"[44]. This is something of a misnomer, as cellular noise has been found to play an important role in (and sometimes be a primary driver of[45,46]) a wide variety of decision making[47,48] and developmental processes[49]. On a more immediately visible note, the bottom two panels of Fig 1.7 show the dominant role that noise can play in moulding the epigenetic landscape. Most of the landscapes of the simple expression system variants are roughly symmetrical. Alternatively, the large fluctuations present in the noisiest variant (at the bottom of the figure) skew its landscape towards larger counts, giving it a long tail. This long tail is a nice illustration of the complex, non-symmetrical behaviors that can emerge in the limit of low copy number, demonstrating the need for a stochastic approach to the simulation of even the simplest GRNs.

If a system includes at least one nonlinear reaction (*i.e.* a reaction for which

the corresponding rate law/propensity is nonlinear), it becomes possible for the results of deterministic and stochastic simulation to diverge completely. When working with nonlinear systems, deterministic simulation is in general unable to reproduce even the correct mean behavior[50]. Both the p dimerization and the $p_2 + g_p$ binding reactions of our self regulating expression system (see Eq 1.3) are second order, and thus nonlinear. Fig 1.8 shows the results from a deterministic and a stochastic simulation of this system. The deterministic and stochastic means do indeed differ signficantly, by $\sim$15%.

Regardless of the ability of deterministic simulation to reproduce the correct mean behavior, a larger issue remains. The self regulating expression system depicted in Fig 1.8 has more than one metastable (*i.e.* long-lived) state. Evidence for this can be seen in both the time series (*e.g.* the long stretch around hour 6 during which the protein count stays fixed near 0) and via the fact that its epigenetic landscape has more than one mode/peak. Deterministic simulation cannot be used effectively to map the various states of this kind of multi-stable system, nor to calculate the dynamics of the transitions in between them. Thus, in order to construct the epigenetic landscape of a complex, nonlinear system with multiple possible states, stochastic simulation is required.

1.7 Rare events, statistics, and the limits of stochastic simulation

With respect to a particular system, an event is rare[51] if it occurs at a much slower rate than the fastest event[††]. In a multi-stable system, switching between states tends to be a rare event since it often requires the co-occurrence of specific fluctuations in two or more noisy reaction channels. For example, in order for the self regulating expression system shown in Fig 1.8 to switch from active to repressed, first a p dimerization reaction (itself a rare event) must occur. Then the $p_2 + g_p$ binding reaction must happen before the p_2 dedimerization reaction has a chance to fire.

The existence of a rare event implies a large separation between the signficant timescales of a system. This timescale separation can prove challenging[52,53] for many different simulation techniques. Theoretically, SSA simulation can exactly reproduce the behavior of any system, regardless of the presence of rare events. In practice however, it can be difficult, or even impossible, to produce an accurate simulation of a rare event systems using SSA. This apparent paradox can be understood by considering the error statistics of SSA.

When studying a rare event, it is standard practice to first determine its mean first passage time ($MFPT$), the mean waiting time before the event occurs. Given a rare event, stochastic simulation can be used to determine its $MFPT$. The procedure is equivalent to estimating the mean of a random

[††]How much slower? There is no formal definition of a rare event, but as a rule of thumb assume that in order to qualify as rare, an event must occur at least 3 orders of magnitude less often than the system's most frequent event

variable $\mathfrak{w}$ (in this case, $\mathfrak{w}$ is the waiting time in between occurrences of the event). A single sample $\mathfrak{w}_i$ is drawn from $\mathfrak{w}$ by running a trajectory until the first occurrence of the rare event, then recording the final simulation time. n samples are drawn by running n such replicates. The mean of these samples:

$$\frac{1}{n}\sum_{i=1}^{n}\mathfrak{w}_i,$$

is then an estimator of $MFPT$.

The accuracy of the $MFPT$ calculated by SSA depends on the size of the sample count n. The exact form of the dependence depends in turn on the exact form of $\mathfrak{w}$ (though it can in general be said that the accuracy increases along with the n). For a rare switching event (*e.g.* a transition that takes a system between two well separated states $\mathcal{A}$ and $\mathcal{B}$), $\mathfrak{w}$ will tend towards an exponential distribution[54], assuming that there are no intervening states. In the limit of large sample sizes, the margin of error of the stochastic $MFPT$ calculation is then given by a simple formula (see Sec **??** for derivations and a complete discussion):

$$\frac{z_\alpha}{\sqrt{n}},$$

where z_α is the z score for confidence level α (*i.e.* $z_{.95} \approx 1.96$). Fig 1.10 shows the margins of error for a wide range of sample count values. In particular, it shows that a very large (38415) sample count is required in order to ensure that a simulation produces an accurate (*i.e.* no more than 1% error) $MFPT$ value.

For simple enough systems with few enough components, the computational cost (in terms of CPU time) required for accurate stochastic simulation

is merely an inconvenience. For complex systems involving rare events, the computational cost can be prohibitive. Very roughly, the cost of sampling a rare event using standard SSA will scale as:

$$\frac{\Phi_{\text{fastest}}}{\Phi_{\text{rare}}}, \tag{1.18}$$

the quotient of the fluxes of the fastest event and the rare event. This means that for a rare enough event, the computational requirements of accurate simulation will easily eclipse the resources provided by any single computer. When a single computer is insufficient, one approach is to scale up your simulations to make use of HPC/cluster resources (an example of this is shown in Fig 1.9). A better approach is to make use of enhanced sampling. Enhanced sampling is a family of simulation methods that scale more efficiently than Eq 1.18 with respect to rare events.

1.8 Enhanced sampling

> *We are here concerned with essentially the same problem that some of the other speakers have spoken about. We wish to estimate the probability that a particle is transmitted through a shield, when this probability is of the order of* $10 - 6$ *to* 10^8*, and we wish to do this by sampling about a thousand life histories. It's clear that a straightforward approach will not give the results desired.*
>
> – Kahn and Harris, *Estimation of particle transmission by random sampling*[55]

So begins the first paper ever written on the topic of enhanced sampling[‡‡]. The family of enhanced sampling methods can be traced back to work done by von Neumann[56] and a few others[57,58] in the 1950s, during the early days of Monte Carlo nuclear physics simulations.

Given that a researcher is interested in a rare event, stochastic simulation, though potentially more rigorous and accurate than the alternatives, is grossly inefficient. The trajectories of stochastic simulations are bound to follow the probability distributions of their underlying systems, and so by definition waste all but a slim minority of their time fluctuating around the high-likelihood regions of state space. Many researchers (both in systems biology and in many related fields[**vanErp:2005jua**, 59,60,61]) over the years have asked themselves, "why can't I just confine my simulations to the part of state space I actually want to see?"

The typical goal of running a stochastic simulation is to collect samples, and then use those samples to estimate the value of some system property that cannot be calculated directly. Certainly it is technically feasible to simply bias or confine a simulation so that it remains within a region of interest, producing more relevant samples more quickly. However, any artificial bias added to the simulation will also bias the samples, and thereby distort the statistics of the estimation process. It is possible to both have your cake (confine your simulation to a region) and eat it too (produce an unbiased sample) by applying bias in a controlled fashion, while at the same time keeping track

[‡‡]Technically, Kahn et al., 1951[55] is one of two papers about enhanced sampling that were published in the same conference proceedings. However, Kahn starts on an earlier page than von Neumann, 1951[56], so I'll call Kahn the first.

of any distortion you cause via bookkeeping. With the right information the collected samples can then be unbiased, producing the desired result. This is the essence of the enhanced sampling methods.

In the broadest terms, all enhanced sampling methods follow the same basic script:

1. Generate a biased version of the original system.
2. Draw samples from the biased system.
3. Based on bookkeeping information recorded during the preceding steps, recover an unbiased sample by applying statistical methods to the biased sample.

There are now dozens (if not hundreds) of different enhanced sampling methods, and while some of them may not strictly follow these steps, each and every one of them performs the 3 actions listed above in some fashion.

1.9 Enhanced sampling methods for stochastic simulation of biochemistry

Beginning with the proposal of forward flux sampling (FFS) in 2005[62], a number of different enhanced sampling methods have been developed for use in stochastic simulations of biochemical systems. Along with FFS, nonequilibrium umbrella sampling (NEUS) and weighted ensemble (WE) are also highly cited methods.

FFS, NEUS and WE have a number of features in common. Each method requires the user to specify a function of their system's complete state to use

as an order parameter:

$$\mathcal{O}(t, X) = (o_0, o_1, \cdots)$$

Additionally, each requires the user to specify a region of interest in terms of a set of bins that span an interval along the order parameter. Taken together, the order parameter and the tiling define a constraint that will be used to bias the user's system. The order parameter defines the degrees of freedom of the constraint, while the tiling defines one or more bounded regions along those degrees of freedom. Ultimately this constraint is used to guide the system into and along the region of interest, though the actual implementation of the constraint varies from method to method.

1.9.1 Forward flux sampling

The first step of FFS is to define two points of interest in your system's state space (we'll call these $\mathcal{A}$ and $\mathcal{B}$), a 1D order parameter that can unambiguously differentiate those two points, and a 1D sequence $(\lambda_0, \lambda_1, \ldots, \lambda_N)$ of "interfaces" (*i.e.* bins) that lie along the order parameter in between $\mathcal{A}$ and $\mathcal{B}$. If a trajectory passes below $\lambda_\mathcal{A} = \lambda_0$ it is said to be in the $\mathcal{A}$ state, and if it passes above $\lambda_\mathcal{B} = \lambda_N$ it is said to be in the $\mathcal{B}$ state. FFS also requires the user to manually specify a maximum simulation time T and a set of trajectory counts $(n_1, n_2, \ldots, n_N)$, the significance of which are discussed in the following paragraphs.

Given the above setup, the goal of an FFS simulation is to calculate $\Phi_{\mathcal{A},\mathcal{B}}$, the flux from state $\mathcal{A}$ into $\mathcal{B}$. $\Phi_{\mathcal{A},\mathcal{B}}$ can be decomposed into a smaller flux and

a probability term:

$$\Phi_{\mathcal{A},\mathcal{B}} = \Phi_{\mathcal{A},0} P\left(\lambda_{\mathcal{B}} \mid \lambda_0\right),$$

where $\Phi_{\mathcal{A},0}$ is the flux from the vicinity of $\mathcal{A}$ to the initial interface λ_0, and $P\left(\lambda_{\mathcal{B}} \mid \lambda_0\right)$ is the probability that a trajectory will reach $\mathcal{B}$ before it falls below λ_0, given that the trajectory is currently at λ_0. The probability can be further decomposed into a sequence of probabilities along the interfaces:

$$\Phi_{\mathcal{A},\mathcal{B}} = \Phi_{\mathcal{A},0} \prod_{i=1}^{N} P\left(\lambda_i \mid \lambda_{i-1}\right). \tag{1.19}$$

The actual FFS simulation consists of an ordered sequence of $N+1$ distinct phases. A schematic illustration of these phases is shown in Fig 1.11. Each phase produces an estimate of one of the terms on the left hand side of Eq 1.19.

The simulation begins with phase 0, at the start of which a single unbiased trajectory is initialized at $\mathcal{A}$ and allowed to run until the user-specified simulation time T. Whenever this trajectory crosses λ_0 while traveling forward (*i.e.* towards $\mathcal{B}$) the species counts at the point of crossing are recorded in a list C_0. If the trajectory ever leaves state $\mathcal{A}$ (*e.g.* by crossing into $\mathcal{B}$), the simulation time is paused, and doesn't resume incrementing until the trajectory reenters $\mathcal{A}$. The phase 0 trajectory is terminated once it reaches time T, and the flux term in Eq 1.19 is then estimated as:

$$\Phi_{\mathcal{A},0} = \frac{n_0^s}{T},$$

where n_0^s is the count of entries in C_0. Phase 0 then ends, and phase 1 begins.

During each phase $i > 0$, a user-defined number n_i of trajectories are run.

Each trajectory is initialized at a randomly chosen point from C_{i-1}, the list of crossing points from the previous phase. The trajectory is run until it either falls below λ_0 or passes above λ_i. If the trajectory passed above λ_i its species count at the point of crossing is recorded in C_i, and in either case the trajectory is terminated. Once n_i trajectories have run, the corresponding probability term in Eq 1.19 is then estimated as:

$$P\left(\lambda_i \mid \lambda_{i-1}\right) = \frac{n_i^s}{n_i}.$$

Phase i then ends, and $i+1$ begins. Once phase N finishes, the FFS simulation is over. The flux $\Phi_{\mathcal{A},\mathcal{B}}$ can be calculated from Eq 1.19, and the $MFPT$ can be calculated as the inverse flux, ${\Phi_{\mathcal{A},\mathcal{B}}}^{-1} = MFPT$.

Various extensions for the FFS method have been developed. One of the more useful ones is from a follow-up paper[63] that showed how FFS can be used to calculate a wide variety of values beyond flux and $MFPT$. They define a set of weights that can be used to combine biased data collected during the individual phases into a single set of unbiased results, equivalent to those that could be gained from an unconstrained SSA simulation. Among other things, this enables the use of FFS to rapidly construct the epigenetic landscape of a system.

A more complete description of FFS (including a including a step-by-step listing of the algorithm) is given later in this thesis in Sec **??**. The version given in Sec **??** differs somewhat from the original implementation of FFS, with each change made either for the sake of parallelizing the algorithm, or to make one of the method's outputs more amenable to statistical analysis. On the

other hand, the description given above is faithful to the version from the originating papers[62,64,65].

1.9.2 Nonequilibrium umbrella sampling

NEUS was first proposed[66] in 2007, a couple of years after FFS was introduced, and bears it a number of similarities.[67,68]

Fig 1.12

1.9.3 Weighted ensemble

WE is a method that originated[59] in the molecular dynamics community. More than a decade after it was first proposed, a paper[69] was published that showed how WE could be adapted for use with stochastic biochemical simulations.[70,71,72]

Fig 1.13

Interestingly, WE turns out to be nearly identical to the “splitting technique" attributed to von Neumann in the first enhanced sampling paper[55]. Thus, WE can be thought of as an accidental rediscovery of the splitting method[73].

1.10 Simplifying and optimizing forward flux

The original formulation of FFS[62] left a great of room for improvement. Specifically, FFS adds a large number of unspecified degrees of freedom to a system,

in the form of a large set of extra simulation parameters. When a computational scientist sets up a FFS simulation, she must specify two points of interest, a 1D order parameter, a set of interfaces λ_i, the phase 0 maximum simulation time T, and the set of phase $i > 0$ trajectory counts n_i. These simulation parameters are required in addition to all of the model data required for a standard SSA simulation. In theory, the choice of FFS parameters will not have an effect on simulation outcome[65], only on simulation efficiency.

The additional simulation parameters present several challenges to users. For new users, the FFS parameters can be an imposing hurdle. They are likely one of the stumbling blocks preventing wider adoption of FFS methods. In particular, the order parameter and the interfaces are conceptually complicated. It takes time for a beginner to learn how to find reasonable values for these parameters. Even for an experienced user it can take a significant amount of trial and error[74] to find good values when working with a novel system. Further, for most systems there exists an optimal[75] set of FFS parameters that will maximize their efficiency and minimize their runtime.

It is possible to devise automated methods that can find and set the optimal FFS parameters values without user input. Several examples are discussed in the paragraphs below. These optimization routines kill two birds with one stone: they simplify the setup of an FFS simulation and make the method easier to use, while also speeding the simulation up. Between the published optimization methods and the work presented in this thesis (see Sec 1.11), it is possible to find optimized values of every FFS parameter (even the order parameter[76]). Thus, it should be possible to devise a method that can

automatically set an optimal method of every parameter all at once, though none has yet been published.

Along this vein, Borrero and coworkers devised a number of different optimization routines[77] for FFS. These routines are designed around the concept that the user's computational resources are limited. Thus, they require the user to input a quantity of computational effort, given in terms of the total number of trajectories that will be run during a single FFS simulation. The routines then find an optimal set of parameters that minimizes overall simulation error while holding the computational effort fixed at the user-specified level. For standard FFS, they devised one routine that can be used to find the optimal choice of n_i (the number of trajectories to run in each phase $i > 0$), and another that can be used to find the optimal choice of λ_i (the placement of the interfaces along the 1D order parameter). Since computational effort is held fixed, Borrero's optimization methods will not speed a simulation up, but will instead decrease the level of error in the simulation's results. Indeed, they found that their optimization methods could reduce simulation error by as much as 40%.

Borrero's interface optimization method generates an initial rough guess of λ_i and then using iterative rounds of FFS simulation in order to refine it. Kratzer and coworkers developed a more elegant alternative approach[74] to optimizing λ_i that avoids the need to perform entire extra simulations. Instead, they figured out a way to dynamically generate an optimal placement of interfaces, one at a time, during an otherwise standard FFS simulation. Kratzer describes two different variants of his method, but both follow roughly

the same script: at the start of a simulation only the first and last interfaces, λ_0 and λ_N, are defined. At the start of each phase $i > 0$, a fixed number of trial trajectories are launched from λ_{i-1} under controlled conditions. Based on the outcomes of these trial trajectories, a location for λ_i is chosen, and the FFS simulation proceeds as normal (until the start of the next phase). Kratzer found that his dynamic interface optimization methods were able to speed simulations up by as much as 2X, relative to a simulation run with manually placed interfaces.

The optimization schemes of both Borrero and Kratzer ignore the contribution of phase 0 to the simulation error. Technically, they both treat the outcome of phase 0 as deterministic, causing it to "fall out" of their analyses of the simulation variances and errors. This is standard practice[78] in analyses of the error statistics of FFS. We devised a novel approach to the analysis of FFS error that allows the contribution of phase 0 to be calculated directly (see Sec **??**). Contrary to previous claims[65] in the literature, it can be shown that phase 0 can be a large contributor to overall simulation error (see Secs **??** and **??**).

1.11 Forward flux pilot sampling

In many ways enhanced sampling seems too good to be true. It seems kind of like something for nothing, or a free lunch. Though the developers of various enhanced sampling methods have claimed that they are faster than direct sampling while introducing no extra approximations or errors[79,78], it remains to be definitely proven one way or the other. Thus it is reasonable to ask "does enhanced sampling actually work? Will it actually produce an answer

of equivalent accuracy in less time than the established methods?" More prosaically, when performing FFS simulations, a researcher might wonder "how many trajectories should I run in each phase in order to get a result without too much error?" All of these are questions that the work presented in the subsequent chapters of this thesis attempts to answer.

In the next chapter we present a new analysis of the error in the output of FFS simulations. Using a novel derivation, we find a more general form of the FFS error relation (*i.e.* simulation error as a function of the simulation parameters) than has been presented in the literature[78,77,65]. In particular, this allows us to calculate the previously unconsidered error arising from phase 0. We show that this phase 0 error can indeed be a significant contribution to the overall simulation error.

Next, we derive an equation that gives the optimal (in terms of computational cost) number of trajectories to run in each phase, given a user-defined desired maximum level of error in the simulation results, which we call an error goal. Based on this optimizing equation, we develop a novel variant of the FFS enhanced sampling method which we call forward flux pilot sampling (FFPilot). FFPilot replaces the T and the entire set of n_i parameters of standard FFS (*i.e.* the parameters that determine the computational effort expended during each phase) with a single error goal parameter. Given that error goal, FFPilot will plan out and run the fastest possible FFS simulation. Thus, FFPilot is both a simplification and an optimization of the FFS approach. We wrote an optimized, fully parallelized implementation of the FFPilot alogorithm in C++. This implementation of FFPilot was added to the Lattice Microbes[80], a

stochastic simulation software package published by the Roberts Lab.

The remainder of the chapter is dedicated to a thorough validation and exploration of FFPilot. In simulations of 1D biochemical systems (*i.e.* systems with only a single chemical species), we show that FFPilot is indeed able to control error in the final simulation results as expected. In fact, FFPilot controls error so well that it is able to uncover a previously invisible problem in all existing analyses of FFS error, including my own. The published error analyses[78,77,65] all assume that sampling error (*i.e.* error due to running too few trajectories) is the sole source of simulation error. However, when using FFS to simulate a complex system with a rough, multidimensional epigenetic landscape, it turns out that other sources of error can become significant. Results from FFPilot simulations of multidimensional systems show that while sampling error is still the dominant source of error, there is an anomalous extra error that FFPilot is unable to control. We demonstrate that the anomalous error is due to correlations between the trajectory starting point distributions along different interfaces.

The final chapter of the thesis is a tutorial that explains in detail how to use the FFPilot implementation in Lattice Microbes. Complete examples are given that show how to use FFPilot to calculate both the $MFPT$ and the epigenetic landscape of a system.

Figures

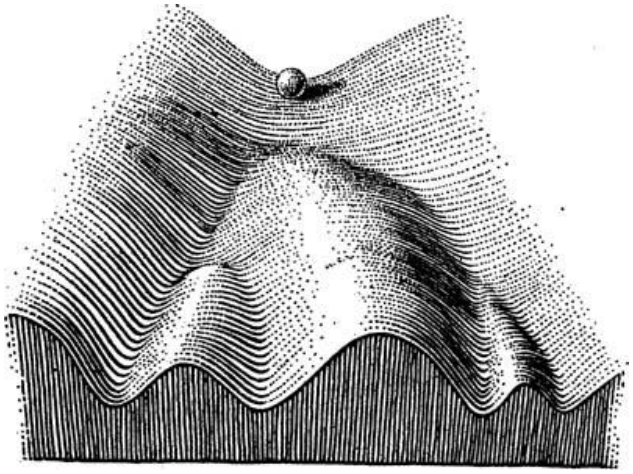

Figure 1.1: The epigenetic landscape of a developing cell, as envisioned by Waddington. The ball represents a cell at the beginning of development. Eventually the ball/cell will reach the bottom of the hill, ending up in one of several possible terminal cell types. The steep terrain counteracts most perturbations by forcing the ball back onto the path, making the whole process robust. Reprinted from Waddington, 1957[7].

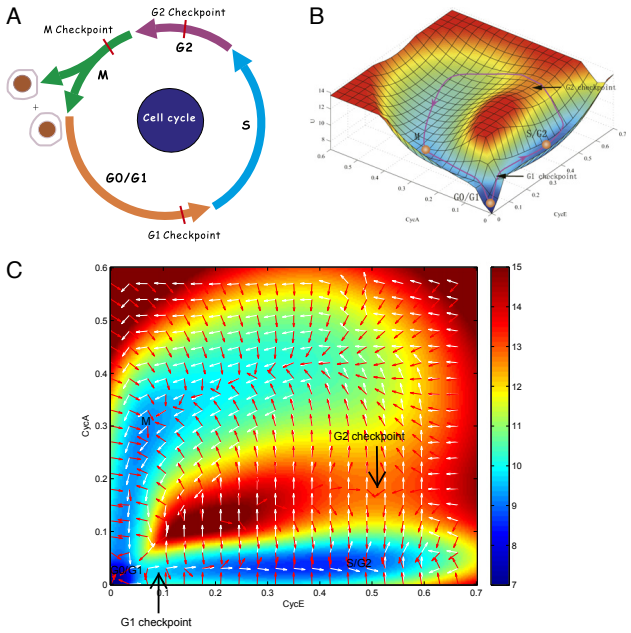

Figure 1.2: Three views of the mammalian cell cycle. A) The classical schematic view. B) The potential surface of the cell cycle. Data was taken from a 44 dimensional model and projected onto the concentrations of two cyclins, *CycA* and *CycE* (arbitrary units). The low points on the surface correspond to the various phenotypes along the cell cycle, and the transition path between them is shown. C) A 2D rendition of the complete epigenetic landscape of the model shown in B. The red arrows show the gradient of the potential, whereas the white arrows show the probability flux. Note that they never exactly coincide. Reprinted from Li et al., 2014[12].

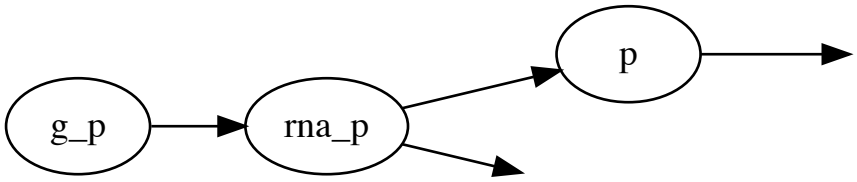

Figure 1.3: A simple digraph representation of the simple gene expression model from Eq 1.2. An arrow from one species to another implies that the first species is involved in the production of the second. Arrows to empty space imply a decay reaction.

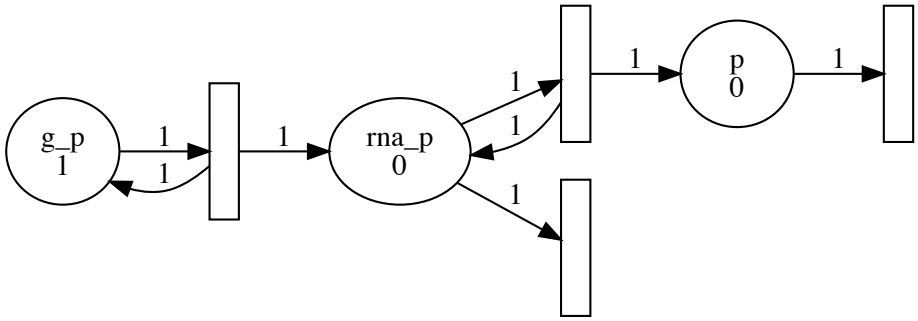

Figure 1.4: A petri net representation of the simple genetic expression model from Eq 1.2. Arrows starting at places (round nodes) and ending at transitions (rectangular nodes) imply that the connected species is a reactant in the connected reaction. Arrows starting at transitions and ending at places imply that the connected reaction produces the connected species. Above each arrow is a weight that gives its stoichiometry. The marking (the count of each species) is shown underneath the name of each species. Transitions that have no arrows leaving them imply a decay reaction.

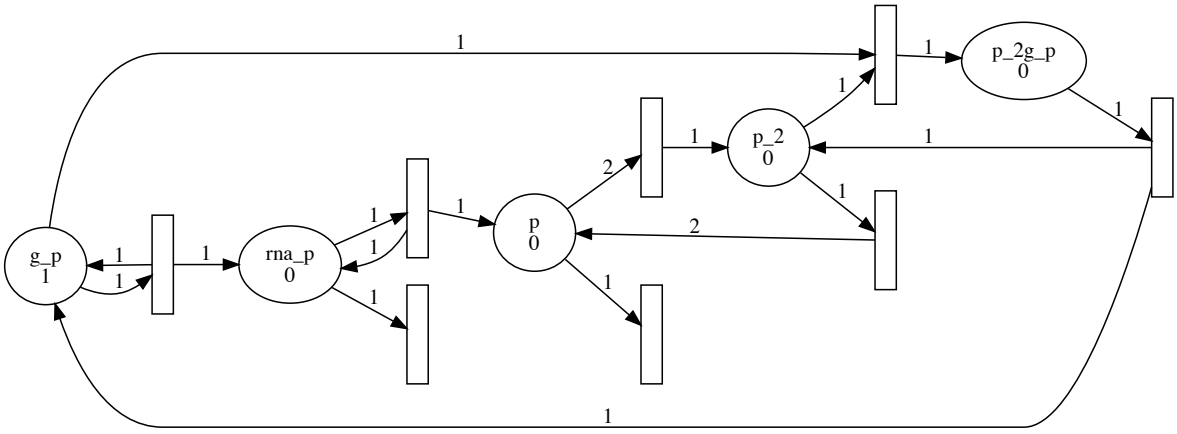

Figure 1.5: A petri net representation of the self regulating genetic expression model from Eq 1.3. The network is represented as described in Fig 1.4.

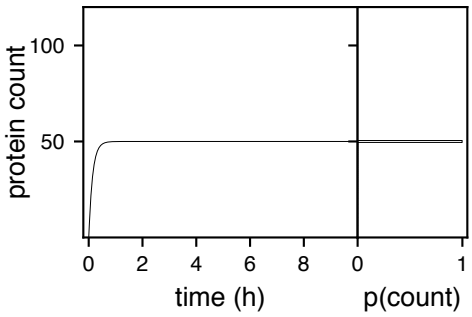

Figure 1.6: Time series from a deterministic simulation of the simple expression system (see Eq 1.2) with $k_1 = .1, k_2 = .1, k_3 = .1, k_4 = 2 \cdot 10^{-3}$. The subplot along the left side shows the epigenetic landscape (as calculated via a histogram of the adjacent time series).

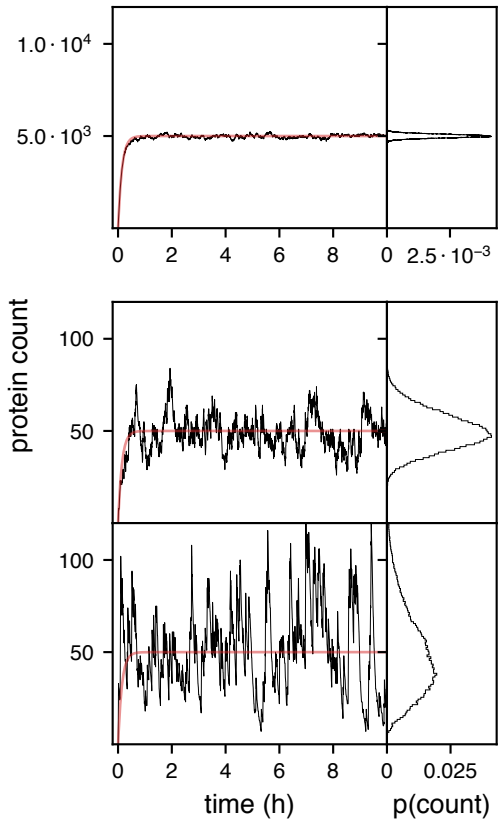

Figure 1.7: (Black lines) time series from stochastic simulations of the simple expression system (see Eq 1.2). The subplots along the left side show the epigenetic landscape of each variant (as calculated via a histogram of data from 10 simulation days, including the adjacent stochastic time series). (Red lines) the deterministic simulation of the same system, for comparison purposes. (Top panel) simple expression system with $k_1 = 10, k_2 = .1, k_3 = .1, k_4 = 2 \cdot 10^{-3}$ (Middle panel) same, with $k_1 = .1, k_2 = .1, k_3 = .1, k_4 = 2 \cdot 10^{-3}$. (Bottom panel) same, with $k_1 = .01, k_2 = 1, k_3 = .1, k_4 = 2 \cdot 10^{-3}$. The same constants were used in the deterministic and stochastic simulations (*i.e.* $c_i = k_i$). The mean expression level of the system simulated in the top panel is 100x that of the systems in the bottom two panels. From top to bottom the noise strengths are 1.98, 1.98, and 10.8039 (see Eq 1.17).

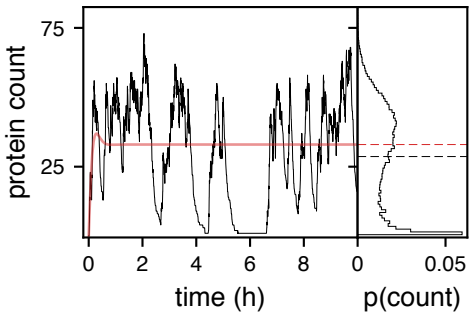

Figure 1.8: (Black line) time series from a stochastic simulation of the self regulating expression system (see Eq 1.3). (Red line) the deterministic simulation of the same system. The subplot along the left side shows the epigenetic landscape. (Black and red dashed lines) the mean protein expression value as calculated by stochastic and deterministic simulation, at 28.7 and 33.0 respectively (15% divergence). The deterministic rate constants used were $k_1 = .1, k_2 = .1, k_3 = .1, k_4 = 2 \cdot 10^{-3}, k_5 = 10^{-6}, k_6 = 1, k_7 = 1, k_8 = 2 \cdot 10^{-3}$. The stochastic rate constants were the same (*i.e.* $c_i = k_i$), except for $c_5 = 2 \cdot k_5 = 2 \cdot 10^{-6}$ (as per Eq 1.14).

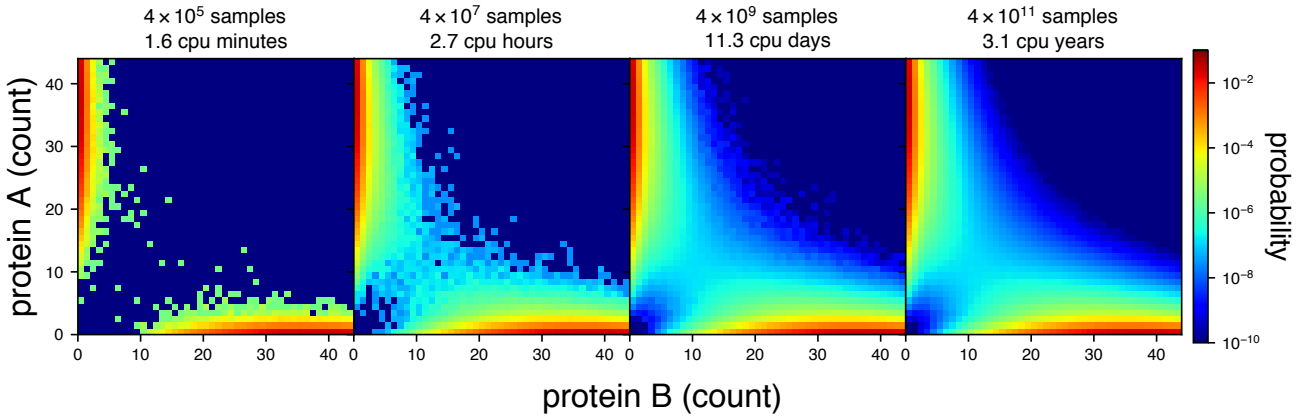

Figure 1.9: Two dimensional epigenetic landscapes generated from the $\text{GTS}_{\theta=.1}$ system (described in Sec **??**). From left to right, the subplots show versions of the same landscape calculated using an increasing number of species count samples. The total sample count used in each landscape, and the CPU time used to collect those samples, is shown above each subplot. The samples were taken from 20 trajectories, each of which was generated using several thousand CPU cores running a parallelized implementation of SSA[80].

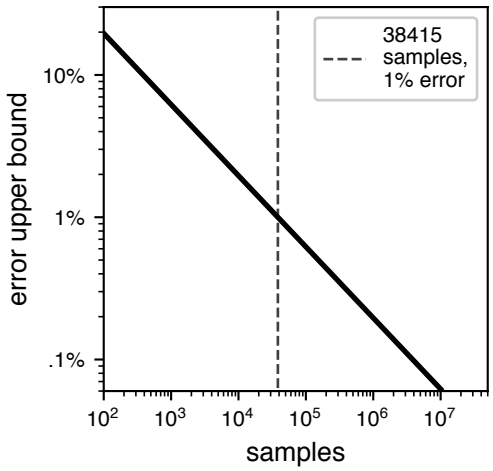

Figure 1.10: The margin of error (95% confidence) vs sample count when using stochastic simulation to calculate the $MFPT$ of a rare state switching event. The x-axis begins at 100 in order to emphasize that the relationship is only valid in the limit of large sample size. See Sec **??**for derivation and details.

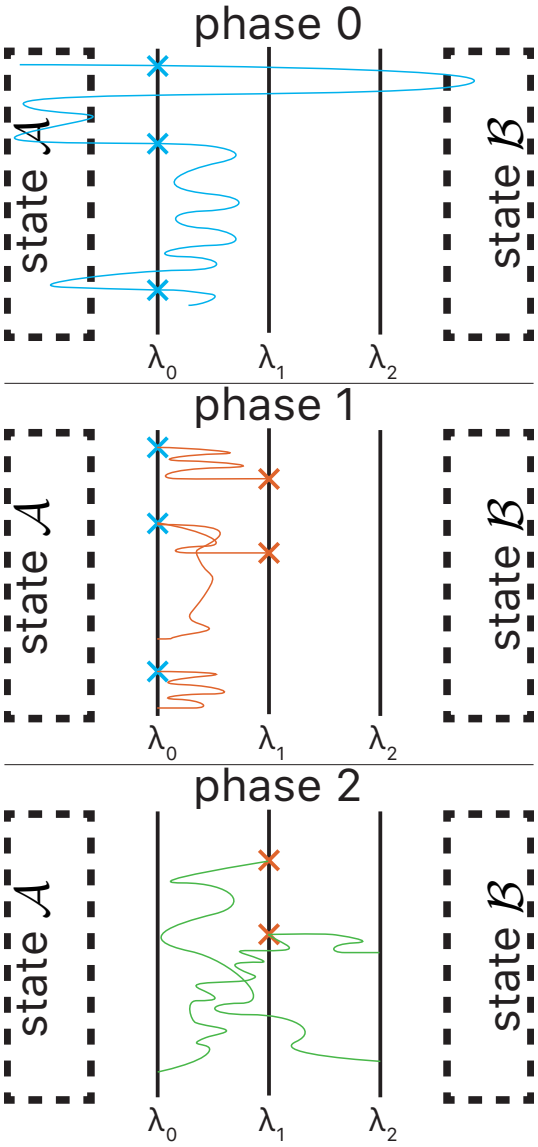

Figure 1.11: A schematic representation of an FFS simulation run from $\mathcal{A} \rightarrow \mathcal{B}$. The simulation shown has 3 phases, one for each interface λ_i that is defined.During each phase $i > 0$, trajectories are initialized from one of the points (chosen at random) where a trajectory in the previous phase crossed λ_{i-1} while traveling in the forward direction (*i.e.* towards $\mathcal{B}$). During phase 2 (the final phase), trajectories that cross λ_2 are considered to have completed the transition into $\mathcal{B}$.

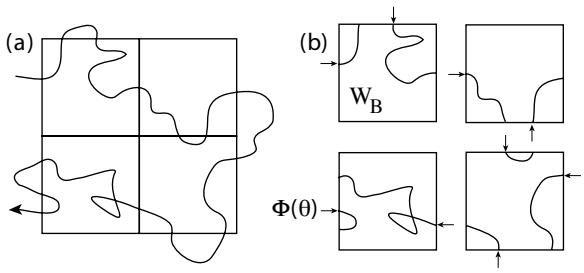

Figure 1.12: Reprinted from Dickson et al., 2004[67].

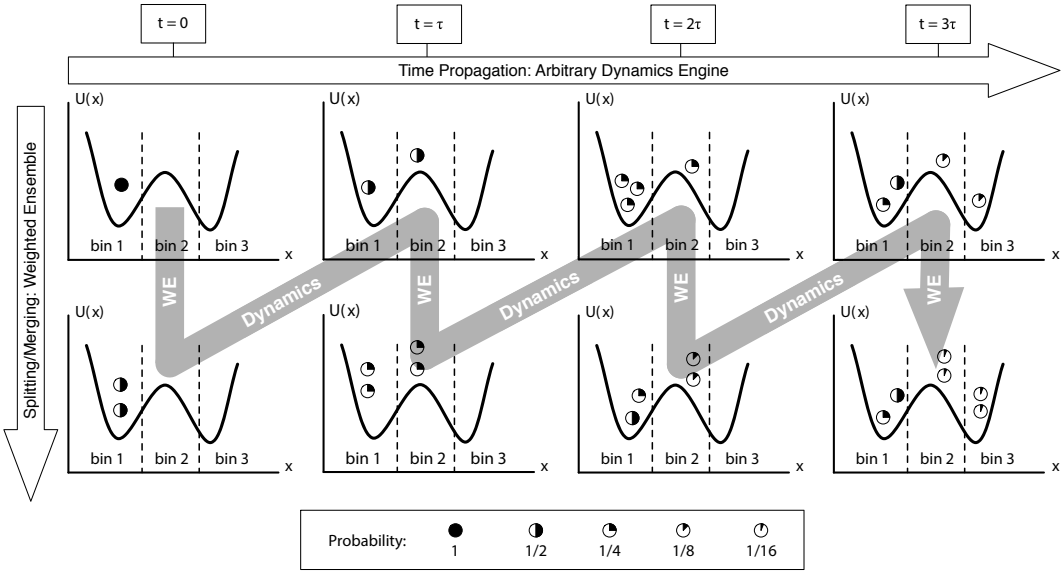

Figure 1.13: Reprinted from Donovan et al., 2013[70].

References

[1] Richard Rhodes. *The making of the atomic bomb*. New York: Simon & Schuster, 1986.

[2] W Johannsen. "The Genotype Conception of Heredity". In: *The American Naturalist* 45.531 (1911), pp. 129–159.

[3] H Hellman. *Great Feuds in Medicine: Ten of the Liveliest Disputes Ever*. Medical Humanities collection PAH. Wiley, 2001.

[4] Francis Crick. "On protein synthesis." In: *Symp. Soc. Exp. Biol.* 12 (1958), pp. 138–163.

[5] Francis Crick. "Central dogma of molecular biology". In: *Nature* 227.5258 (1970), pp. 561–563.

[6] Conrad Hal Waddington. *Principles of Embryology*. 1965.

[7] Conrad Hall Waddington. *The Strategy of the Genes : a Discussion of Some Aspects of Theoretical Biology*. London: Allen and Unwin, 1957.

[8] Cyrus Levinthal. "How to Fold Graciously". In: *Mossbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois*. Ed. by JTP Debrunnder and E Munck. University of Illinois Press, 1969, pp. 22–24.

[9] Li Xu, Xiakun Chu, Zhiqiang Yan, Xiliang Zheng, Kun Zhang, Feng Zhang, Han Yan, Wei Wu, and Jin Wang. "Uncovering the underlying physical mechanisms of biological systems via quantification of landscape and flux". In: *Chinese Phys. B* 25.1 (2016), pp. 016401–27.

[10] Wei Wu and Jin Wang. "Potential and flux field landscape theory. II. Non-equilibrium thermodynamics of spatially inhomogeneous stochastic dynamical systems". In: *J Chem Phys* 141.10 (2014), pp. 105104–48.

[11] Wei Wu and Jin Wang. "Potential and flux field landscape theory. I. Global stability and dynamics of spatially dependent non-equilibrium systems". In: *J Chem Phys* 139.12 (2013), p. 121920.

[12] Chunhe Li and Jin Wang. "Landscape and flux reveal a new global view and physical quantification of mammalian cell cycle." In: *Proc Natl Acad Sci USA* 111.39 (2014), pp. 14130–14135.

[13] Xiaosheng Luo, Liufang Xu, Bo Han, and Jin Wang. "Funneled potential and flux landscapes dictate the stabilities of both the states and the flow: Fission yeast cell cycle". In: *PLoS Comput Biol* 13.9 (2017), e1005710–31.

[14] Chunhe Li and Jin Wang. "Quantifying the underlying landscape and paths of cancer." In: *J R Soc Interface* 11.100 (2014), pp. 20140774–20140774.

[15] Armando Aranda-Anzaldo and Myrna A R Dent. "Landscaping the epigenetic landscape of cancer". In: *Progress in Biophysics and Molecular Biology* (2018), pp. 1–20.

[16] Carl Adam Petri. "Communication with automata". PhD thesis. University of Hamburg, 1966.

[17] P J Goss and J Peccoud. "Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets." In: *Proc Natl Acad Sci USA* 95.12 (1998), pp. 6750–6755.

[18] J W Pinney, D R Westhead, and G A McConkey. "Petri Net representations in systems biology." In: *Biochem. Soc. Trans.* 31.Pt 6 (2003), pp. 1513–1515.

[19] Simon Hardy and Pierre N Robillard. "Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches." In: *J Bioinform Comput Biol* 2.4 (2004), pp. 595–613.

[20] Peter J Haas. *Stochastic Petri Nets*. Modelling, Stability, Simulation. Springer Science & Business Media, 2006.

[21] Darren J Wilkinson. *Stochastic Modelling for Systems Biology, Second Edition*. CRC Press, 2012.

[22] P Waage and C M Gulberg. "Studies concerning affinity". In: *J. Chem. Educ.* 63.12 (1986), p. 1044.

[23] Wolfram Research, Inc. "Mathematica 11.0". In: ().

[24] Alfred J Lotka. "Contribution to the Theory of Periodic Reactions". In: *J. Phys. Chem. ...* 14.3 (1909), pp. 271–274.

[25] Alfred J Lotka. *Elements of Physical Biology*. Williams and Wilkins, 1925.

[26] Loots I Dublin and Alfred J Lotka. "On the True Rate of Natural Increase: As Exemplified by the Population of the United States, 1920". In: *Journal of the American Statistical Association* 20.151 (1925), pp. 305–339.

[27] Vito Volterra. “Fluctuations in the Abundance of a Species considered Mathematically”. In: *Nature* 118.2972 (1926), pp. 558–560.

[28] A M Turing. “The Chemical Basis of Morphogenesis”. In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641 (1952), pp. 37–72.

[29] D T Gillespie. “Stochastic simulation of chemical kinetics”. In: *Annu Rev Phys Chem* (2007).

[30] N G Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 2011.

[31] J Goutsias and G Jenkinson. “Physics Reports”. In: *Physics Reports* 529.2 (2013), pp. 199–264.

[32] D T Gillespie. “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions”. In: *J Comput Phys* 22.4 (1976), pp. 403–434.

[33] Donald A McQuarrie. “Stochastic approach to chemical kinetics”. In: *Journal of Applied Probability* 4.3 (1967), pp. 413–478.

[34] I Oppenheim, K E Shuler, and G H Weiss. “Stochastic and Deterministic Formulation of Chemical Rate Equations”. In: *J Chem Phys* 50.1 (1969), pp. 460–466.

[35] Thomas G Kurtz. “The Relationship between Stochastic and Deterministic Models for Chemical Reactions”. In: *J Chem Phys* 57.7 (1972), pp. 2976–2978.

[36] D T Gillespie. “A rigorous derivation of the chemical master equation”. In: *Physica a-Statistical Mechanics and Its Applications* 188.1-3 (1992), pp. 404–425.

[37] Daniel T Gillespie. *Markov processes : an introduction for physical scientists*. Boston: Academic Press, 1992.

[38] Daniel T Gillespie, Andreas Hellander, and Linda R Petzold. “Perspective: Stochastic algorithms for chemical kinetics”. In: *J Chem Phys* 138.17 (2013), pp. 170901–15.

[39] Michael A Gibson and Jehoshua Bruck. “Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels”. In: *J Phys Chem A* 104.9 (2000), pp. 1876–1889.

[40] Yang Cao, Hong Li, and Linda Petzold. "Efficient formulation of the stochastic simulation algorithm for chemically reacting systems". In: *J Chem Phys* 121.9 (2004), pp. 4059–4067.

[41] Larry Lok and Roger Brent. "Automatic generation of cellular reaction networks with Moleculizer 1.0". In: *Nat Biotechnol* 23.1 (2005), pp. 131–136.

[42] James M McCollum, Gregory D Peterson, Chris D Cox, Michael L Simpson, and Nagiza F Samatova. "The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior". In: *Computational Biology and Chemistry* 30.1 (2006), pp. 39–49.

[43] Ertugrul M Ozbudak, Mukund Thattai, Iren Kurtser, Alan D Grossman, and Alexander van Oudenaarden. "Regulation of noise in the expression of a single gene." In: *Nat Genet* 31.1 (2002), pp. 69–73.

[44] M B Elowitz. "Stochastic Gene Expression in a Single Cell". In: *Science* 297.5584 (2002), pp. 1183–1186.

[45] Joanna Jaruszewicz and Tomasz Lipniacki. "Toggle switch: noise determines the winning gene". In: *Phys Biol* 10.3 (2013), p. 035007.

[46] R Ahrends, A Ota, K M Kovary, T Kudo, B O Park, and M N Teruel. "Controlling low rates of cell differentiation through noise and ultrahigh feedback". In: *Science* 344.6190 (2014), pp. 1384–1389.

[47] Burton W Andrews and Pablo A Iglesias. "An information-theoretic characterization of the optimal gradient sensing response of cells." In: *PLoS Comput Biol* 3.8 (2007), e153.

[48] Gábor G Balázsi, Alexander A van Oudenaarden, and James J JJ Collins. "Cellular Decision Making and Biological Noise: From Microbes to Mammals". In: *Cell* 144.6 (2011), pp. 910–925.

[49] Elisabet Pujadas and Andrew P Feinberg. "Regulated Noise in the Epigenetic Landscape of Development and Disease". In: *Cell* 148.6 (2012), pp. 1123–1131.

[50] Sayuri K Hahl and Andreas Kremling. "A Comparison of Deterministic and Stochastic Modeling Approaches for Biochemical Reaction Systems: On Fixed Points, Means, and Modes". In: *Front. Genet.* 7.46 (2016), p. 054103.

[51] Peters Baron. *Reaction Rate Theory and Rare Events*. First edition. Elsevier B.V., 2017.

[52] L F Shampine and C W Gear. "A User's View of Solving Stiff Ordinary Differential Equations". In: *SIAM Rev.* 21.1 (1979), pp. 1–17.

[53] Linda Petzold. "Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations". In: *SIAM J. Sci. and Stat. Comput.* 4.1 (1983), pp. 136–148.

[54] David J Aldous. "Markov chains with almost exponential hitting times". In: *Stochastic Processes and their Applications* 13.3 (1982), pp. 305–310.

[55] Herman Kahn and Theodore E Harris. "Estimation of particle transmission by random sampling". In: *Nat Bur Stand Math* 12 (1951), pp. 27–30.

[56] John von Neumann. "Various Techniques Used in Connection with Random Digits". In: *Nat Bur Stand Math* 12 (1951), pp. 36–38.

[57] H Kahn and A W Marshall. "Methods of Reducing Sample Size in Monte Carlo Computations". In: *OR* 1.5 (1953), pp. 263–278.

[58] George E Forsythe. "Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions". In: *Mathematics of Computation* 26.120 (1972), pp. 817–11.

[59] G A Huber and S Kim. "Weighted-ensemble Brownian dynamics simulations for protein association reactions." In: *Biophys J* 70.1 (1996), pp. 97–110.

[60] Christoph Dellago, Peter G Bolhuis, Félix S Csajka, and David Chandler. "Transition path sampling and the calculation of rate constants". In: *J Chem Phys* 108.5 (1998), pp. 1964–15.

[61] Rafael C Bernardi, Marcelo C R Melo, and Klaus Schulten. "Enhanced sampling techniques in molecular dynamics simulations of biological systems". In: *BBA - General Subjects* 1850.5 (2015), pp. 872–877.

[62] Rosalind J Allen, Patrick B Warren, and Pieter Rein ten Wolde. "Sampling rare switching events in biochemical networks." In: *Phys Rev Lett* 94.1 (2005), p. 018104.

[63] Chantal Valeriani, Rosalind J Allen, Marco J Morelli, Daan Frenkel, and Pieter Rein ten Wolde. "Computing stationary distributions in equilibrium and nonequilibrium systems with forward flux sampling." In: *J Chem Phys* 127.11 (2007), p. 114109.

[64] Rosalind J Allen, Daan Frenkel, and Pieter Rein ten Wolde. "Simulating rare events in equilibrium or nonequilibrium stochastic systems." In: *J Chem Phys* 124.2 (2006), p. 024102.

[65] Rosalind J Allen, Chantal Valeriani, and Pieter Rein ten Wolde. "Forward flux sampling for rare event simulations." In: *J Phys Condens Matter* 21.46 (2009), p. 463102.

[66] Aryeh Warmflash, Prabhakar Bhimalapuram, and Aaron R Dinner. "Umbrella sampling for nonequilibrium processes." In: *J Chem Phys* 127.15 (2007), p. 154112.

[67] Alex Dickson, Aryeh Warmflash, and Aaron R Dinner. "Nonequilibrium umbrella sampling in spaces of many order parameters." In: *J Chem Phys* 130.7 (2009), p. 074104.

[68] Alex Dickson, Aryeh Warmflash, and Aaron R Dinner. "Separating forward and backward pathways in nonequilibrium umbrella sampling". In: *J Chem Phys* 131.15 (2009), pp. 154104–11.

[69] Divesh Bhatt, Bin W Zhang, and Daniel M Zuckerman. "Steady-state simulations using weighted ensemble path sampling." In: *J Chem Phys* 133.1 (2010), p. 014110.

[70] Rory M Donovan, Andrew J Sedgewick, James R Faeder, and Daniel M Zuckerman. "Efficient stochastic simulation of chemical kinetics networks using a weighted ensemble of trajectories." In: *J Chem Phys* 139.11 (2013), p. 115105.

[71] Rory M Donovan, Jose-Juan Tapia, Devin P Sullivan, James R Faeder, Robert F Murphy, Markus Dittrich, and Daniel M Zuckerman. "Unbiased Rare Event Sampling in Spatial Stochastic Systems Biology Models Using a Weighted Ensemble of Trajectories". In: *PLoS Comput Biol* 12.2 (2016), e1004611–25.

[72] Daniel M Zuckerman and Lillian T Chong. "Weighted Ensemble Simulation: Review of Methodology, Applications, and Software". In: *Annu Rev Biophys* 46.1 (2017), pp. 43–57.

[73] Lillian T Chong, Ali S Saglam, and Daniel M Zuckerman. "Path-sampling strategies for simulating rare events in biomolecular systems". In: *Curr Opin Struct Biol* 43 (2017), pp. 88–94.

[74] Kai Kratzer, Axel Arnold, and Rosalind J Allen. "Automatic, optimized interface placement in forward flux sampling simulations". In: *J Chem Phys* 138.16 (2013), pp. 164112–164112.

[75] Ao Ma and Aaron R Dinner. "Automatic Method for Identifying Reaction Coordinates in Complex Systems †". In: *J Phys Chem B* 109.14 (2005), pp. 6769–6779.

[76] Ernesto E Borrero and Fernando A Escobedo. "Reaction coordinates and transition pathways of rare events via forward flux sampling". In: *J Chem Phys* 127.16 (2007), p. 164101.

[77] Ernesto E Borrero and Fernando A Escobedo. "Optimizing the sampling and staging for simulations of rare events via forward flux sampling schemes". In: *J Chem Phys* 129.2 (2008), pp. 024115–024117.

[78] Rosalind J Allen, Daan Frenkel, and Pieter Rein ten Wolde. "Forward flux sampling-type schemes for simulating rare events: efficiency analysis." In: *J Chem Phys* 124.19 (2006), p. 194111.

[79] Bin W Zhang, David Jasnow, and Daniel M Zuckerman. "The "weighted ensemble" path sampling method is statistically exact for a broad class of stochastic processes and binning procedures". In: *J Chem Phys* 132.5 (2010), pp. 054107–8.

[80] Elijah Roberts, John E Stone, and Zaida Luthey-Schulten. "Lattice Microbes: high-performance stochastic simulation method for the reaction-diffusion master equation." In: *J Comput Chem* 34.3 (2013), pp. 245–255.