MULTISCALE SIMULATION OF REACTION DYNAMICS IN CHEMICAL, BIOLOGICAL AND MATERIALS SYSTEMS

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Leonard Alfredo Harris

February 2010

MULTISCALE SIMULATION OF REACTION DYNAMICS IN CHEMICAL, BIOLOGICAL AND MATERIALS SYSTEMS

Leonard Alfredo Harris, Ph.D.

Cornell University 2010

In this dissertation, we introduce a novel accelerated-stochastic simulation method, known as the 'partitioned-leaping algorithm' (PLA), for efficiently simulating chemical reaction networks. The technique is *multiscale* in that it considers dynamics at scales ranging from the discrete-stochastic to the continuous-deterministic. It is particularly useful when considering nanoscale-sized systems that exhibit fluctuating dynamics and contain species with large disparities in populations. We present the theoretical foundations of the PLA, discuss various extensions and variants of the method and provide illustrative examples demonstrating its practical utility in chemistry, biology and materials science.

In Chapter 1, we provide a general overview of the origins and consequences of stochastic "noise" in nanoscale-sized systems. We elucidate the implications of this phenomenon, which arises because of the discrete and probabilistic nature of molecular interactions, in both biological and materials settings and discuss mathematical approaches that have been applied previously to model such behaviors. The shortcomings of these methods provide the primary motivation for the work presented in this dissertation.

In Chapter 2, we present the theoretical foundations of so-called "exact" stochastic simulation approaches. This material lays the foundation for all that is to follow. It can be seen as a review/tutorial of the subject at the level of advanced undergraduate and beginning graduate students. Our presentation closely follows

the work of Gillespie ca. 1976. Though many equivalent formalisms have been presented in the literature, Gillespie's has the advantage of being developed within the language of chemistry and, thus, being more accessible to chemical engineers than other approaches that are often cited, e.g., within the physics literature.

In Chapter 3, we present the main contribution of this dissertation, the partitioned-leaping algorithm. Building upon the work of Gillespie ca. 2000 and concepts presented in Chapter 2, we develop an accelerated-stochastic simulation approach that efficiently describes stochastic effects in chemical reaction networks with very little loss in accuracy relative to exact methods. The method is simple, relatively easy to implement and is based on firm theoretical grounds. We also consider numerous variants of the method and discuss areas of possible future extension.

In Chapter 4, we proceed to select applications of the PLA. We consider example systems inspired by chemistry, biology and materials science. We begin with various toy problems and then advance to simple, yet relevant, biochemical networks. In all cases, we compare the performance characteristics of the PLA, in terms of accuracy and efficiency, to exact approaches. We also identify conditions where the method does *not* perform particularly well, investigate the underlying reasons for this and discuss possible strategies for overcoming them.

Finally, we conclude in Chapter 5 by summarizing the main results of this dissertation and laying out a vision for the future.

BIOGRAPHICAL SKETCH

Leonard Alfredo Harris was born on the 29th of February 1976 in Greenwich, CT to Leonard and Rosa. He spent the first 13 years of his life in Connecticut before moving to Highlands Ranch, CO in 1989. He graduated from Highlands Ranch High School in 1994 and received a B.S. in Chemical Engineering from the University of Colorado-Boulder in 1998. He came to Cornell in Fall 1999 in the M.Eng. program in Chemical Engineering and transferred to the Ph.D. program in Fall 2000. In 2002, he spent a productive year at Lawrence Livermore National Laboratory working in the Computational Systems Biology group. He is currently working as a pending postdoctoral associate in the Department of Computational Biology at the University of Pittsburgh School of Medicine.

To my beautiful wife Hauna, without whom this would have never been possible.

ACKNOWLEDGEMENTS

First and foremost, I must thank my family, whose support, both emotional and otherwise, has been crucial in making this possible. Thanks to my beautiful wife Hauna, who came to me in the most difficult time of my life and helped me emerge stronger than ever. I never would have succeeded without her support. Thanks to my mom and dad for their commitment to myself and our family. Knowing that they were there helped me get through many a restless night. Thanks to my sister Mary Ann, and my nephews Raul and Sebastian, for the joy that they bring me. And thanks to Howard and Francelle for accepting me into their family and always being so welcoming and warm.

I would also like to thank my thesis advisor, Paulette Clancy, for giving me the freedom to explore the countless ideas that I have had over the years. Thanks go out to all of the members of the Clancy group as well with whom I have had the pleasure to work with: Chin-Lung Kuo, Dev Choudhary, Erik Albenze, Ben Nicholson, Mohit Haran, Jay Catherwood, Joe Goose, Michelle Giron, Becky Cantrell and Krishna Iyengar. I must also thank the three excellent undergraduates that I had the pleasure to mentor: Hoeun Lee, Aaron Piccirrili and Emily Majusiak. And thanks to my two good friends Andy Harrison and Ivan Gospodinov. Those first-year all-nighters did eventually pay off.

Finally, I thank Andrew Quong, Babak Sadigh and Chris Mundy for their mentorship and friendship during 2002, my year spent at Lawrence Livermore National Laboratory, Profs. Fernando Escobedo and Jim Sethna, my committee members, for various stimulating discussions over the years, Cornell's Learning Initiatives for Future Engineers (LIFE) program for financial support of my undergraduate mentees, and the Semiconductor Research Corporation for funding through the SRC Graduate Fellowship Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Origins and Importance of Stochasticity in Nanoscale-Sized Reactive Systems

A field of study that has widespread use in a variety of scientific disciplines is *population dynamics*. Population-dynamical models consider how the populations of certain entities evolve in time by following the natural logic of a set of prespecified interaction rules. The entities in question can be almost anything, from interacting galaxies (cosmology) to financial institutions (economics), animals (ecology), atoms and molecules (chemistry) or sub-atomic particles (particle physics). Similarly, the interaction channels can be of almost any form. The simplest cases are "mass-action" rules, where the interaction rates are directly proportional to the population levels of the interacting species (see Appendix B). Other examples include Michaelis-Menten or Hill rules in biochemical kinetics or the Langmuir-Hinshelwood mechanism in surface physics. In this dissertation, we will focus our attention on atoms, molecules and chemical reactions as this is the natural language of chemical engineering. It should be remembered throughout, however, that the methods discussed within are applicable to a much wider range of problems in the general area of population dynamics.

Historically, the mathematical methods used to analyze population-dynamical models assume that (i) the populations are continuous quantities and (ii) the time-evolution behavior is deterministic in nature. These assumptions are the basis for analysis methods based on ordinary and partial differential equations (ODEs and PDEs) that consider spatially homogenous and heterogenous systems, respectively. In reality, however, both assumptions are false. By definition, populations are

collections of discrete entities and the physics that govern their interactions are fundamentally probabilistic, or *stochastic*. Recognition of this fact goes as far back as the advent of the atomic theory and the seminal work by Einstein on Brownian motion. Nevertheless, the continuous and deterministic assumptions of classical population dynamics tend to work exceedingly well in practice. This is because the "noise" characteristics of a collection of interacting entities scales as $1/\sqrt{\Omega}$, where Ω is a measure of the system size. Thus, in the common chemical scenario where interactions take place between molecules numbering on the order of Avogadro's number, e.g., in a laboratory beaker or a chemical reactor, stochastic effects are effectively absent.

The current state of technology is beginning to change this situation. Advances in experimental techniques are increasingly giving scientific investigators access to the dynamical behaviors of exceedingly small systems. Examples include miniaturized silicon technologies, nanofluidic devices and individual biological cells. It is now possible to probe the behaviors of just a handful, or even an individual, atom or molecule in the laboratory. A profound, albeit not unexpected, result of such experimentation has been the observation that the dynamics of nanoscale-sized systems are highly stochastic and often do not follow the predictions of classical continuum theories.

As a result, a renewed interest in stochastic theory and simulation methods has arisen. A well-developed approach within population dynamics that has at its roots the works on Brownian motion is that of stochastic differential equations (SDEs). SDEs are essentially modified ODEs that have an added noise term included. An immense amount of literature exists on the theory of SDEs and many advanced computational methods have been developed to analyze them. A his-

torical shortcoming of SDEs, however, within the context of population dynamics, concerns the origin of the noise term and the functional form that it should take. There has been much debate on this issue with confusing conclusions. Moreover, while SDEs lift the assumption of determinism in the system dynamics they retain the continuum description of populations.

An alternative stochastic analysis method that has become particularly popular recently is kinetic Monte Carlo (kMC). kMC is a generic term that refers to any numerical simulation method that acts to evolve a system of interacting entities forward in time by randomly firing events via the generation of random numbers. Within this framework, both the deterministic and continuum assumptions that underly ODE/PDE-based approaches are lifted. There are countless flavors of kMC, some heuristic and some with a firm footing in theory. Perhaps the most well-known and commonly-used kMC method in the chemical literature is Gillespie's *stochastic simulation algorithm*, or SSA. The SSA is an "exact" stochastic method, based firmly in Markov process theory, that operates by generating random samples of event sequences and firing times. The primary shortcoming of the SSA, and all kMC methods in general, is the fact that each and every event firing in a system is explicitly simulated. If one or more species exist in large numbers or if the rate parameters vary widely between event channels the efficiency of the approach is significantly hindered. Basically, kMC methods tend to spend most of their effort simulating frequently occurring events and only infrequently sample the rare events which are often of the most interest. As such, kMC approaches are primarily restricted to only the smallest of systems.

The time is thus ripe for the development of stochastic simulation methods that can bridge the divide between kMC, SDE and ODE/PDE-type approaches. Such

multiscale methods should be capable of accurately and efficiently simulating the behaviors of systems characterized by wide disparities in both species populations and rate parameters. In this dissertation, we propose just such a method, termed the "partitioned-leaping algorithm" (PLA). Built upon recent work by Gillespie on accelerated-stochastic simulation approaches and drawing on prior theoretical developments within the kMC realm, the PLA provides a theoretically-sound and easy-to-implement platform for performing multiscale simulations of reaction dynamics in chemical, biological and materials settings.

In Chapter 2, we lay the theoretical groundwork for the PLA by reviewing in detail the foundations of exact-stochastic simulation. We go well beyond the usual treatment, considering the general case of both intrinsic sources of stochasticity, i.e., those associated with the probabilistic nature of molecular interactions, and extrinsic sources associated with fluctuations in the reaction environment. We also consider different variations of the approach that will be important in later stages of our development, various optimization strategies that are commonly employed and extensions to the method that allow for, e.g., the treatment of spatially-inhomogeneous systems.

In Chapter 3, we introduce the PLA. We begin by reviewing the multiscale theoretical framework recently developed by Gillespie and discussing the associated "τ-leaping" approach. We then merge these ideas with ones presented in Chapter 2 to produce the PLA. Various implementation details are discussed and an in-depth presentation of the critical time step-selection procedure is provided. We then compare and contrast the PLA to various related approaches that have been proposed in literature, explaining why we believe the PLA to be superior and discussing which aspects of those methods can be incorporated into the PLA,

before concluding with a presentation of possible future extensions.

In Chapter 4, we apply the PLA to various example systems inspired by biology and materials science to demonstrate the applicability of the approach. We show that the method can achieve significant computational savings relative to kMC methods under the right circumstances and expound upon those scenarios in which the method does not perform particularly well. This provides the motivation for the future directions discussed in Chapter 3.

Finally, we conclude in Chapter 5 by summarizing the contributions advanced in this dissertation and offering an eye towards the future.

Chapter 2

Significant Previous Work: Theoretical Foundations of Exact-Stochastic Simulation Methods

2.1 Introduction, terminology and syntax

The scenario that we are interested in in this dissertation involves N interacting species $\mathbf{S} = \{S_1, \ldots, S_N\}$ participating in M interaction channels $\mathbf{R} = \{R_1, \ldots, R_M\}$. In general, the system in which the species interact can be of fixed or time-varying volume, temperature or any other such "environmental" quantity. Furthermore, the reactive entities can be homogenously or heterogeneously distributed throughout the domain. However, if they are heterogeneously distributed, then the system must be divided into multiple homogeneously-distributed subdomains in order to apply the simulation methods discussed in this dissertation (see Secs. 2.5 and 3.6.2 for further discussion).

Within the chemistry literature, stochastic simulation is most often discussed within the framework developed by Gillespie [41, 44, 49]. The starting point for discussing the theoretical foundations of stochastic simulation methods is thus Gillespie's "fundamental postulate" of stochastic chemical kinetics [41]:

$$a_\mu(t)dt = a_\mu\left(\mathbf{X}(t), \mathbf{\Theta}(t)\right) dt \tag{2.1}$$
$$\equiv \text{probability, at time } t, \text{ that one instance of reaction } R_\mu \text{ will fire within the next infinitesimal time interval } dt.$$

Here, $a_\mu(t)$ is known as the reaction "propensity" and is analogous to the reaction "rate" of deterministic chemical kinetics.[1] $\mathbf{X}(t)$ is the vector of species populations at time t and $\mathbf{\Theta}(t)$ is the vector of environmental quantities that can, in general, vary in time. Written this way, postulate (2.1) expresses the time dependence of $a_\mu(t)$ in terms of *intrinsic* quantities (species populations) and *extrinsic* quantities (environmental variables). Importantly, intrinsic quantities are constant between reaction firings while extrinsic quantities need not be.

Throughout this dissertation, when referring to quantities associated with reactions we will use Greek characters such as μ, ν and τ. This will distinguish them from those associated with species, which will be labeled with Latin characters such as i, j and T. Sets of quantities will be placed within curly braces ($\{\cdot\}$) and the quantity inside will generally be labeled with the subscript ν for reactions and j for species. These labels will imply $\nu = 1, \ldots, M$ and $j = 1, \ldots, N$ unless otherwise specified. The labels μ and i will be reserved for *specific* reactions and species, respectively. At times, we will also denote collections of values in vector and matrix notation. Vectors will be denoted as bold characters in Times New Roman font (e.g., $\mathbf{A}$—as above) while matrices will be denoted as bold Sans Serif characters (e.g., $\mathsf{\mathbf{A}}$).

In what follows, we will discuss the implications of postulate (2.1) in terms of approaches for simulating stochastic population dynamics. In the remainder of Chapter 2 we will discuss "exact" methods that are useful in the case of small numbers of interacting entities. In Chapter 3, we will present accelerated methods, including a novel approach developed by the author, that build upon the foundations laid in this chapter. Along the way, we will also discuss numerous variants

[1] Since our system is stochastic, this terminology more accurately reflects the inherent randomness of the interactions.

and consider an assortment of possible extensions.

2.2 Foundational theory

The most general approach to the problem of stochastic population dynamics is that of the "master equation" [45, 76]. The master equation describes the time evolution of the probabilistic state-space of a system. In other words, we think of the state vector $\mathbf{X}(t)$ and ask: "What is the probability that the system will be in the state $\mathbf{X}(t)=\mathbf{x_t}$ at time t given that it began in state $\mathbf{x_0}$ at time t_0?" Using the definition of the propensity in postulate (2.1), the master equation can be written as

$$\begin{aligned}\frac{\partial}{\partial t}P(\mathbf{x_t},t|\mathbf{x_0},t_0) &= \sum_{\nu=1}^{M}\left[a_\nu(\mathbf{x_t}-\mathbf{z}_\nu,\theta_\mathbf{t})P(\mathbf{x_t}-\mathbf{z}_\nu,t|\mathbf{x_0},t_0)\right. \\ &\quad \left. - a_\nu(\mathbf{x_t},\theta_\mathbf{t})P(\mathbf{x_t},t|\mathbf{x_0},t_0)\right]. \end{aligned} \tag{2.2}$$

$P(\mathbf{x_t},t|\mathbf{x_0},t_0)$ is known as the "grand probability function" while the vector $\mathbf{z}_\mu = \{z_{\mu 1},\ldots,z_{\mu N}\}$ holds the stoichiometric coefficients for each species in reaction R_μ (note that most of the entries in $\mathbf{z}_\mu$ are zero since, generally speaking, only a few species participate in any given reaction). In words, Eq. (2.2) states that the rate of change of the probability of being in state $\mathbf{x_t}$ at time t, given that the system was in state $\mathbf{x_0}$ at time t_0, is equal to the probability that the system is a single reaction firing away from state $\mathbf{x_t}$ at time t multiplied by the propensity of moving into $\mathbf{x_t}$ [with $\mathbf{\Theta}(t)=\theta_\mathbf{t}$], less the probability that it is in state $\mathbf{x_t}$ at time t multiplied by the propensity of moving away from $\mathbf{x_t}$.

In general, if one can solve Eq. (2.2) then one has all of the information that one could ever hope for regarding the time evolution of a system. With an expression for $P(\mathbf{x_t},t|\mathbf{x_0},t_0)$ in hand, mean values, standard deviations and all higher-order

moments can, in principle, be calculated for every species in the system at all times $> t_0$. Unfortunately, this is only possible for the simplest of systems, meaning that the master equation approach is generally infeasible for problems of practical interest [41].

As a result, Gillespie took an alternative approach, deriving a simulation scheme, known as the "stochastic simulation algorithm" (SSA) [41], that produces time-evolution trajectories consistent with the grand probability function of the master equation. The SSA is derived from the same fundamental hypothesis (2.1) that led us to Eq. (2.2), meaning that in the limit of an infinite number of sample trajectories, the SSA "exactly" reproduces $P(\mathbf{x_t}, t|\mathbf{x_0}, t_0)$ [45]. The SSA thus provides a means by which a solution to the master equation (2.2) can be obtained *without actually solving* Eq. (2.2).

The theoretical construct that underlies the SSA is known as the "next-reaction probability density function," $p(\tau, \mu|\mathbf{x_t}, t)$, defined as [41]

$$p(\tau, \mu|\mathbf{x_t}, t)d\tau \equiv \text{probability, at time } t, \text{ that the } \textit{next} \text{ reaction to fire in the system will occur within the infinitesimal time interval } [t+\tau, t+\tau+d\tau) \textit{ and} \text{ will be of type } R_\mu. \tag{2.3}$$

$p(\tau, \mu|\mathbf{x_t}, t)$ is evidently a joint-probability density function that governs the firing times, τ, and reaction types, μ, of the next reaction to fire in the system given the current state $\mathbf{x_t}$ at time t. Thus, if we can derive an expression for $p(\tau, \mu|\mathbf{x_t}, t)$ we could *simulate* the temporal evolution of a system by repeatedly sampling values of τ and μ from the density function (using one of various sampling techniques—see Secs. 2.3–2.5) and updating the system accordingly. The resulting trajectory would represent a random sample of the grand probability function $P(\mathbf{x_t}, t|\mathbf{x_0}, t_0)$.

This, very simply, is the essence of the SSA, a beautifully simple, and yet powerful, simulation algorithm that almost anyone can implement.

Algorithm 1 (Gillespie, 1976 [41]):

Stochastic simulation algorithm.

1. *Initialize: Define reaction network, rate parameters, initial species populations, set* $t = t_{\text{start}}$.
2. *Calculate propensities* $\{a_\nu\}$ *for all reactions.*
3. *Generate sample values* $\{\tau, \mu\}$ *from* $p(\tau, \mu|\mathbf{x_t}, t)$ *(see Secs. 2.3–2.5).*
4. *Advance the clock to* $t + \tau$ *and enact reaction* R_μ *by updating the populations of the species involved in the reaction.*
5. *If a stopping criterion has been reached then terminate, else go to step 2.*

In Fig. 2.1, we provide a simple illustration of the SSA in action. Though developed in 1976, the SSA has become immensely popular only recently, particularly in the field of computational systems biology where stochastic interactions between small numbers of biological molecules within cells are of particular interest.

Our first goal in this chapter is to derive an expression for $p(\tau, \mu|\mathbf{x_t}, t)$. To do so, we follow the approach of Gillespie [44] (note that the following is more general than the original derivation given in [41]). We begin by "conditioning" the density function $p(\tau, \mu|\mathbf{x_t}, t)$ as follows,

$$p(\tau, \mu|\mathbf{x_t}, t)d\tau = \Pr\{0, \tau|\mathbf{x_t}, t\} \times P_1(\mu|t + \tau)d\tau. \tag{2.4}$$

Here, $\Pr\{0, \tau|\mathbf{x_t}, t\}$ is the probability that no reactions of any kind fire within $[t, t + \tau)$ while $P_1(\mu|t + \tau)d\tau$ is the probability, at time $t + \tau$, that reaction R_μ

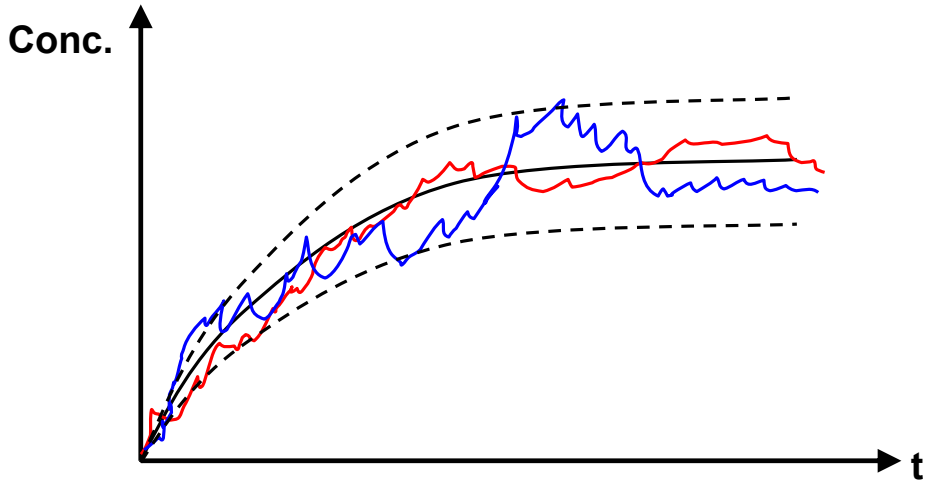

Figure 2.1: Simple illustration of the SSA in action. The blue and red traces signify individual realizations of the temporal evolution of a particular species population (concentration). The solid black curve represents the mean obtained from multiple SSA simulations while the dashed curves represent the spread (e.g., standard deviation). In the limit of an infinite number of simulations, the SSA provides an exact solution to Eq. (2.2).

fires once within the next $d\tau$. Note that we are implicitly assuming here that it is impossible for more than one reaction firing to occur within $d\tau$, which is valid in the limit $d\tau \to 0$.

Given the definition of the propensity in postulate (2.1), we immediately know that

$$P_1(\mu|t+\tau)d\tau = a_\mu(t+\tau)d\tau = a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})d\tau. \tag{2.5}$$

Note that in the second equality, $\mathbf{x_{t+\tau}}$ has been replaced with $\mathbf{x_t}$ because $P_1(\mu|t+\tau)$ has been defined in Eq. (2.4) under the assumption that no reactions fire within $[t, t+\tau)$. Thus, the species populations cannot change during this period. In general, however, the environmental quantities $\mathbf{\Theta}(t)$ can change, hence the dependency upon $\theta_{\mathbf{t}+\tau}$.

It follows from Eq. (2.5) that the probability that reaction R_μ will *not* fire

within $[t+\tau, t+\tau+d\tau)$ is $1-a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})d\tau$. Therefore, by the multiplication law of probability theory, the probability that no reactions of *any* kind fire during $[t+\tau, t+\tau+d\tau)$ is

$$\Pr\{0, d\tau|\mathbf{x}_{\mathbf{t}+\tau}=\mathbf{x_t}, t+\tau\} = \prod_{\nu=1}^{M}[1-a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})d\tau] = 1-\sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})d\tau, \quad (2.6)$$

where the second equality holds for $d\tau \to 0$ (i.e., upon expansion of $\prod_{\nu=1}^{M}[1-a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})d\tau]$, all terms of $\mathcal{O}(d\tau^2)$ fall out).

Using Eq. (2.6) and the multiplication law of probability theory, we can then obtain an expression for the probability that zero reactions of any kind fire within the interval $[t, t+\tau+d\tau)$,

$$\begin{aligned}\Pr\{0, \tau+d\tau|\mathbf{x_t}, t\} &= \Pr\{0, \tau|\mathbf{x_t}, t\} \times \Pr\{0, d\tau|\mathbf{x}_{\mathbf{t}+\tau}=\mathbf{x_t}, t+\tau\} \\ &= \Pr\{0, \tau|\mathbf{x_t}, t\} \times \left(1-\sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})\mathrm{d}\tau\right). \quad (2.7)\end{aligned}$$

Subtracting $\Pr\{0, \tau|\mathbf{x_t}, t\}$ from both sides and dividing through by $d\tau$ gives the simple first-order differential equation

$$\begin{aligned}\frac{\Pr\{0, \tau+\mathrm{d}\tau|\mathbf{x_t}, t\} - \Pr\{0, \tau|\mathbf{x_t}, t\}}{d\tau} &\equiv \frac{d\Pr\{0, \tau|\mathbf{x_t}, t\}}{d\tau} \\ &= -\left(\sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})\right)\Pr\{0, \tau|\mathbf{x_t}, t\}. \quad (2.8)\end{aligned}$$

Upon separation of variables and integration from 0 to τ, we get

$$\Pr\{0, \tau|\mathbf{x_t}, t\} = \exp\left[-\int_0^\tau \sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})\mathrm{d}\tau'\right]. \quad (2.9)$$

Finally, substituting Eqs. (2.5) and (2.9) into (2.4) gives our desired expression,

$$p(\tau, \mu|\mathbf{x_t}, t)d\tau = a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \exp\left[-\int_0^\tau \sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right] d\tau. \quad (2.10)$$

Equation (2.10) represents the major result of this section and is the expression that lies at the heart of the SSA. With this expression in hand, the problem now

becomes one of generating random sample pairs $\{\tau, \mu\}$ that are consistent with Eq. (2.10). In Secs. 2.3 and 2.4, we present two alternative approaches for doing so. Both are due to Gillespie [41] and provide simple analytical expressions and procedures for generating random samples of τ and μ.

2.3 Direct method

The most commonly-used approach for obtaining random samples $\{\tau, \mu\}$ from Eq. (2.10) is known as the direct method (DM) [41]. The DM is "direct" in the sense that the next-reaction probability function in Eq. (2.10) is conditioned into two one-variable probability functions, one for τ and one for μ, that are then sampled independently. We condition Eq. (2.10) as follows,

$$p(\tau, \mu|\mathbf{x_t}, t)d\tau = P_1(\tau|\mathbf{x_t}, t)d\tau \times P_2(\mu|t+\tau). \tag{2.11}$$

$P_1(\tau|\mathbf{x_t}, t)d\tau$ is the probability that the next reaction firing in the system will occur within $[t+\tau, t+\tau+d\tau)$, *regardless of type*, while $P_2(\mu|t+\tau)$ is the probability that the next reaction to fire will be of type R_μ, *given that* a firing will next occur within $[t+\tau, t+\tau+d\tau)$.

Using the addition property of probability theory, it is easy to derive an expression for $P_1(\tau|\mathbf{x_t}, t)d\tau$ by simply summing Eq. (2.10) over all reactions,

$$\begin{aligned} P_1(\tau|\mathbf{x_t}, t)d\tau &= \sum_{\nu=1}^{M} p(\tau, \nu|\mathbf{x_t}, t)d\tau \\ &= a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \exp\left[-\int_0^\tau a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right] d\tau, \end{aligned} \tag{2.12}$$

where $a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \equiv \sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})$. An expression for $P_2(\mu|t+\tau)$ can then obtained by substituting Eq. (2.12) into (2.11), rearranging, and substituting from

Eq. (2.10),

$$\begin{aligned} P_2(\mu|t+\tau) &= p(\tau,\mu|\mathbf{x_t},t) \Big/ \sum_{\nu=1}^{M} p(\tau,\nu|\mathbf{x_t},t) \\ &= a_\mu(\mathbf{x_t},\theta_{\mathbf{t}+\tau}) \,/ a_0(\mathbf{x_t},\theta_{\mathbf{t}+\tau}) \,. \end{aligned} \tag{2.13}$$

Random samples τ can now be obtained from Eq. (2.12), and samples μ from Eq. (2.13), by using the continuous and discrete versions of the "inversion generating technique" [41, 44] (see Appendix C), respectively. In the case of Eq. (2.13), we show in Appendix C that application of the technique is straightforward, yielding μ as the integer that satisfies the relationship

$$\sum_{\nu=1}^{\mu-1} a_\nu(\mathbf{x_t},\theta_{\mathbf{t}+\tau}) \le r_2 \times a_0(\mathbf{x_t},\theta_{\mathbf{t}+\tau}) < \sum_{\nu=1}^{\mu} a_\nu(\mathbf{x_t},\theta_{\mathbf{t}+\tau}), \tag{2.14}$$

where r_2 is a unit-uniform random number on $[0,1)$ [from this point forward, we will denote the unit-uniform distribution as $\mathcal{U}(0,1)$].

In the case of Eq. (2.12), the approach can, in general, be applied with an arbitrary dependence of $a_0(\mathbf{x_t},\theta_{\mathbf{t}+\tau})$ on τ. If the dependence is not simple, however, numerical integration is necessary, which may or may not be practical. The most common implementation of the SSA, however, is in the case of τ-*independent* propensity functions. τ independence implies constancy during $[t,t+\tau)$, which is different from *time* independence. Thus, the species populations $\mathbf{X}(t)$, which remain constant between successive reaction firings (as discussed above), are τ independent. The environmental quantities $\mathbf{\Theta}(t)$ need not be, however. τ independence thus implies that environmental quantities, such as temperature and volume, do not vary in time, i.e., $a_0(\mathbf{x_t},\theta_{\mathbf{t}+\tau}) = a_0(\mathbf{x_t},\theta_{\mathbf{t}}) = a_0(t)$. As shown in Appendix C, random samples τ can be obtained from Eq. (2.12) in this special case as

$$\tau = -\frac{\ln(r_1)}{a_0(t)}, \tag{2.15}$$

where r_1 is another random sample from $\mathcal{U}(0,1)$. Equation (2.15) is used extensively throughout the literature and is the expression most commonly associated with the DM. It is important to recognize, however, that it is formally valid *only* in the case of τ-independent propensities. Note, however, that Eq. (2.14) is valid in either case.

2.3.1 Optimized versions

Optimizations of the DM generally focus on reducing the number of operations required in selecting the reaction type μ via the inequality Eq. (2.14) [49]. The standard approach, known as a "linear search," is to progressively sum the values of the propensities until the partial sum $\sum_{\nu=1}^{\mu} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})$ exceeds $r_2 \times a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau})$. The number of iterations required in this procedure has been termed the "search depth" $\mathbb{S}$ [22]. For (hypothetical) reaction systems in which all of the propensities are *equal*, the average search depth $\langle \mathbb{S} \rangle = M/2$. In practical situations, however, some reactions will invariably have larger propensities than others, meaning that they will be selected more often. Thus, $\langle \mathbb{S} \rangle$ will be skewed towards the index of the reaction with the largest propensity.

Recognizing this, Cao et al. [22] proposed ordering the reactions from largest propensity to smallest in order to minimize the number of iterations required to select μ. However, because propensities change as the system evolves, one cannot simply base the ordering on the initial state of the system. Their strategy, therefore, is to run a *pre-simulation* and collect statistics on the frequencies with which each reaction fires and then order the reactions accordingly for subsequent simulation runs. For the example systems that they considered, Cao et al. [22] showed that this simple procedure can result in significant computational savings over a

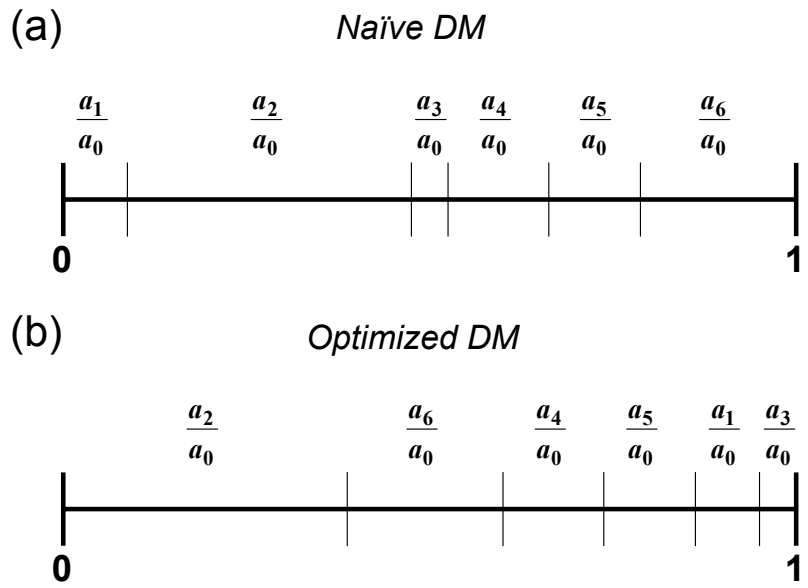

Figure 2.2: Illustration of the basic idea underlying optimized versions of the DM. By ordering the reactions from largest propensity to smallest, the average number of iterations required in selecting the reaction type μ from Eq. (2.14) is minimized. The optimized DM of Cao et al. [22] calls for a pre-simulation in order to determine the appropriate ordering. The methods of McCollum et al. [75] and Fricke and Wendt [39] employ dynamic sorting in order to account for the fact that the species populations, and hence propensities, can change significantly during the course of a simulation.

naive implementation of the DM. In Fig. 2.2, we illustrate the difference between a naive ordering of reactions and an optimized ordering based on the values of the propensities.

The main shortcoming of the approach of Cao et al. [22] lies in its reliance on the pre-simulation. As pointed out by McCollum et al. [75], there is no guarantee that the pre-simulation will explore all possible behaviors of a network. Thus, situations can arise where the pre-simulation-based ordering is significantly sub-optimal, particularly in biochemical networks where gene expression dynamics can result in dramatic changes in population levels. It makes sense, therefore, to sort

the reactions *dynamically* as the simulation proceeds. McCollum et al. [75] proposed a simple and efficient method for doing this: every time a reaction fires move it up one spot in the reaction list, if possible. Repeating this procedure at each step of a simulation will tend to congregate reactions with large propensities at the top of the list and those with small propensities at the bottom, thus minimizing $\langle \mathbb{S} \rangle$. Moreover, the ordering will adjust as the simulation proceeds to reflect any changes in the propensities. McCollum et al. [75] showed, via various example systems, that this "bubble-up" procedure imposes very little overhead on the algorithm, making it always comparable to, and in some cases significantly more efficient than, the method of Cao et al. [22]. The methods of Cao et al. [22] and McCollum et al. [75], among others [68], have been reviewed recently by Gillespie [49].

It is important to note that the idea of sorting reactions from fastest to slowest in order to minimize the expense of event selection in kinetic Monte Carlo algorithms is not a new one. Fricke and Wendt [39], for example, proposed over a decade ago a bubble-up procedure very similar to that of McCollum et al. [75]. The difference is that during the linear search they compare the propensity of the current reaction to that of the former (if there is one) and exchange the order of the reactions if the current is larger than the former. Recent articles by Schulze [101] and Slepoy et al. [107] also make clear that these types of optimizations are not new. Thus, the values of the works of Cao et al. [22] and McCollum et al. [75] lie in introducing these approaches within the context of Gillespie's SSA.

2.4 First-reaction method

In Gillespie's seminal article on the SSA [41], an alternative to the DM was proposed, dubbed the first-reaction method (FRM). The FRM operates on a very different premise than the DM but, as we shall see, is formally equivalent in that it also produces sample values $\{\tau, \mu\}$ that are consistent with Eq. (2.10).

The basic idea behind the FRM is to ask the question: "If reaction R_μ were the *only* reaction present in the system, when would it next fire?" The probability function that governs the answer comes directly from Eq. (2.10) with $M=1$,

$$p_\mu(\tau|\mathbf{x_t}, t)d\tau = a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \exp\left[-\int_0^\tau a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right] d\tau, \tag{2.16}$$

where we have used the subscript μ to signify that this is specific to reaction R_μ. As in the DM, sample values τ_μ, termed "tentative next-reaction times," can be obtained from Eq. (2.16) using the inversion generating technique (Appendix C). In the τ-independent case, the sampling formula is

$$\tau_\mu = -\frac{\ln(r_\mu)}{a_\mu(t)}, \tag{2.17}$$

which follows directly from Eq. (2.15) with $M=1$. r_μ is again a random sample from $\mathcal{U}(0,1)$.

The FRM operates by generating one value of τ_μ for each reaction in the system, choosing as τ the smallest of these and μ as the corresponding reaction. A simple schematic illustrating the procedure is shown in Fig. 2.3. Intuitively, we can explain why this procedure works by noting that, as far as the reaction with the smallest τ_μ is concerned, it *is* the only reaction in the system. In other words, Eq. (2.16) holds for R_μ as long as no other reactions fire first. The reaction with the smallest τ_μ is thus the only one for which this assumption actually holds.

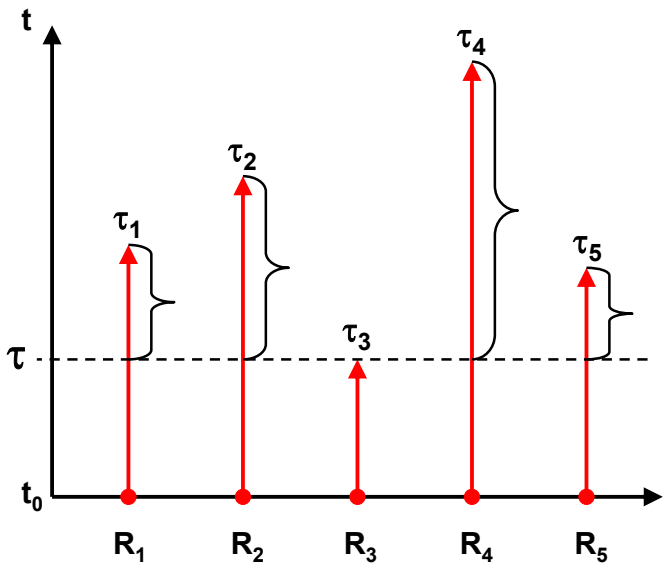

Figure 2.3: Schematic representation of the FRM. A "tentative next-reaction time," τ_μ, is calculated for each reaction in the system and τ set to the smallest of these. μ is set to the corresponding reaction. Here, $\tau = \tau_3$ and $\mu = 3$. Also shown are the "residuals" $\tau_\mu - \tau$.

Mathematically speaking, we can prove that the FRM correctly samples Eq. (2.10) by first defining

$$\widetilde{p}(\tau, \mu|\mathbf{x_t}, t)d\tau \equiv \text{probability, in the FRM, that the next reaction to fire in the system will do so within } [t+\tau, t+\tau+d\tau) \textit{ and} \text{ will be of type } R_\mu. \tag{2.18}$$

Our goal is then to derive an expression for $\widetilde{p}(\tau, \mu|\mathbf{x_t}, t)$ and show that it is equivalent to $p(\tau, \mu|\mathbf{x_t}, t)$ from Eq. (2.10). (Note that what follows is a more general proof than that given in [41]; it is based on the work in Ref. [43].)

We begin by conditioning $\widetilde{p}(\tau, \mu|\mathbf{x_t}, t)$ as follows,

$$\widetilde{p}(\tau, \mu|\mathbf{x_t}, t)d\tau = \Pr\{\tau < \tau_\mu < \tau + d\tau\} \times \Pr\{\tau_\nu > \tau, \textit{ for all } \nu \neq \mu\}. \tag{2.19}$$

The first term, $\Pr\{\tau < \tau_\mu < \tau + d\tau\}$, is simply $p_\mu(\tau|\mathbf{x_t}, t)d\tau$, i.e., Eq. (2.16). The

second term can be written as

$$\begin{aligned}
\Pr\{\tau_\nu > \tau,\ \textit{for all}\ \nu \neq \mu\} &= \prod_{\substack{\nu=1 \\ \nu\neq\mu}}^{M} \Pr\{\tau_\nu > \tau\} \\
&= \prod_{\substack{\nu=1 \\ \nu\neq\mu}}^{M} \Pr\{F_\nu^{-1}(r_\nu|\mathbf{x_t}, t) > \tau\} \\
&= \prod_{\substack{\nu=1 \\ \nu\neq\mu}}^{M} \Pr\{r_\nu > F_\nu(\tau|\mathbf{x_t}, t)\} \\
&= \prod_{\substack{\nu=1 \\ \nu\neq\mu}}^{M} [1 - F_\nu(\tau|\mathbf{x_t}, t)].
\end{aligned} \tag{2.20}$$

The first equality is due to the fact that tentative next-reaction times are statistically independent, the second utilizes the inversion formula Eq. (C.2) in Appendix C, the third uses the identity $F(F^{-1}(r)) = r$, and the fourth comes from the fact that the probability that a unit-uniform random number is greater than $x \in [0, 1)$ is simply $1 - x$.

Using Eqs. (2.16) and (C.1), the probability distribution function $F_\nu(\tau|\mathbf{x_t}, t)$ in Eq. (2.20) can be written as

$$F_\nu(\tau|\mathbf{x_t}, t) = \int_0^\tau a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'}) \exp\left(-\int_0^{\tau'} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau''}) d\tau''\right) d\tau'. \tag{2.21}$$

We now make the following variable substitution [43],

$$b_\nu(\tau') \equiv \int_0^{\tau'} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau''}) d\tau'', \tag{2.22}$$

which gives

$$db_\nu(\tau') = a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'}) d\tau'. \tag{2.23}$$

This allows us to evaluate Eq. (2.21) as

$$\begin{aligned}
F_\nu(\tau|\mathbf{x_t}, t) &= \int_0^{b_\nu(\tau)} e^{-b_\nu(\tau')} db_\nu(\tau') \\
&= 1 - \exp\left[-\int_0^\tau a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'}) d\tau'\right].
\end{aligned} \tag{2.24}$$

Equation (2.20) can then be written as

$$\Pr\{\tau_\nu > \tau,\ \textit{for all } \nu \neq \mu\} = \prod_{\substack{\nu=1 \\ \nu\neq\mu}}^{M} \exp\left[-\int_0^\tau a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right]. \tag{2.25}$$

Substituting Eqs. (2.16) and (2.25) into Eq. (2.19) gives

$$\begin{aligned}
\widetilde{p}(\tau, \mu|\mathbf{x_t}, t)d\tau &= a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \prod_{\nu=1}^{M} \exp\left[-\int_0^\tau a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right] d\tau \qquad (2.26)\\
&= a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \exp\left[-\int_0^\tau \sum_{\nu=1}^{M} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau'})d\tau'\right] d\tau \\
&= p(\tau, \mu|\mathbf{x_t}, t)d\tau.
\end{aligned}$$

Thus, we see that the FRM does indeed correctly sample Eq. (2.10). □

The obvious downfall of the FRM, as originally formulated [41], is that it requires M random number generations at each simulation step, one for each reaction in the system. Since the DM requires only two random numbers at each step, regardless of network size, the FRM is clearly inferior. As such, the FRM was initially relegated to nothing more than an academic curiosity, useful for illuminating the concepts underlying the SSA [41] but largely ignored for practical purposes. By implementing some simple modifications, however, we will see that the FRM can be transformed into a much more efficient version, comparable to, and sometimes superior to, the DM. We will discuss these modifications in the subsequent subsections. Moreover, the idea of considering reactions individually (though not independently), which underlies the FRM, is the key ingredient necessary for seamlessly incorporating the SSA into an accelerated-stochastic simulation framework known as the "partitioned-leaping algorithm" (PLA), which is the main contribution of this dissertation. The PLA will be presented in Chapter 3.

2.4.1 Simple optimization

Towards the end of the presentation of the FRM in Ref. [41], Gillespie discusses the temptation to extend the idea to allow the "second next" reaction to fire in the system to be that with the second smallest τ_μ, a "second-reaction method" as it were. He rightly points out, however, that this is an illegitimate approach because it (i) precludes the possibility that the reaction with the smallest τ_μ could itself fire again before any other reaction, and (ii) ignores the fact that molecules were created or destroyed in the first reaction (the propensities would not be updated to reflect this).

Obviously, a naive implementation of this idea is illegitimate, but it turns out that it is not as bad an idea as one might think. To see this, let us consider Fig. 2.3, a hypothetical step of the FRM. Here, we see that τ_3 is the minimum of the set $\{\tau_\nu\}$, $\nu = 1$–5. Thus, following the procedure proposed in Ref. [41], we would (i) change the system time from t to $t+\tau$ with $\tau = \tau_3$, (ii) update the species populations to reflect the firing of R_3, (iii) calculate new values of $\{a_\nu(t+\tau)\}$ for all reactions, and (iv) generate five samples from $\mathcal{U}(0,1)$ and use them to calculate new values of $\{\tau_\nu\}$.

Consider for a moment, however, that invariably one or more of the reactions in Fig. 2.3 will be completely independent of R_3, i.e., none of the species either created or destroyed in R_3 act as reactants in those reactions. For these reactions we clearly need not generate new random numbers and calculate new values of τ_μ, as called for in step (iv) above; the times at which these reactions are next scheduled to fire are unaffected by the firing of R_3. Put another way, Eq. (2.16) still holds for those reactions because no reactions have yet fired that alter their propensities. It would be completely legitimate, therefore, to set the new tentative next-reaction

times for all *unaffected* reactions equal to their "residuals" (see Fig. 2.3), i.e.,

$$\tau_\mu = \tau_\mu^\circ - \widetilde{\tau}, \quad \mu \neq \mu^\circ \in \textit{unaffected reactions}, \tag{2.27}$$

where τ_μ° is the tentative next-reaction time from the previous simulation step (the "old" value, hence the superscript '$\circ$') and $\widetilde{\tau}=\min\{\tau_\nu^\circ\}$. However, for the affected reactions, including that which just fired (i.e., $\mu=\mu^\circ$), we would still generate new τ_μ values by sampling from Eq. (2.16) [i.e., using Eq. (2.17) in the τ-independent case].

Let us take this idea a step further. Imagine that instead of keeping track of the *relative* times between reaction firings, we instead thought in terms of *absolute* time, i.e., from the start of the simulation. Our tentative next-reaction times would then be defined on an absolute basis as

$$\widehat{\tau}_\mu \equiv \tau_\mu + t. \tag{2.28}$$

If we were to do this, then there would be no need to update the values of $\hat{\tau}_\mu$ for any of the unaffected reactions. The scheduled firing times would remain exactly as before, i.e., $\widehat{\tau}_\mu=\widehat{\tau}_\mu^\circ$.

What we have done in this thought experiment is basically demonstrate how something akin to a "second-reaction method" can be implemented. Clearly, upon the firing of a reaction, the propensities of all reactions affected by that firing must be updated to reflect the changes in the species populations. If we are clever, however, we have shown that we can greatly reduce the computational expense of our approach by (i) only generating new random numbers for those reactions affected by the last reaction firing, and (ii) changing our frame of reference from relative to absolute time. By doing this, not only do we reduce the number of random number generations required at each simulation step, we also eliminate the need to evaluate Eq. (2.27) for all unaffected reactions.

Implementing these modifications is quite easy. The change to absolute time is straightforward and simple, requiring nothing more than modifying the appropriate equation for τ_μ [e.g., Eq. (2.17) in the τ-independent case] by adding to it the current time t, as in Eq. (2.28). Selective updating of only the affected reactions can be carried out by generating, at the outset of a simulation, a data structure which stores which reactions affect which. This object, termed a "dependency graph" in [40], would be accessed at every simulation step.

By implementation these changes, the main improvement that we make to the FRM is in reducing the number of random number generations required at each simulation step. Because most reaction networks are sparse, i.e., most reactions affect only a small subset of all reactions, the number of random numbers required per step can generally be reduced to a number $\ll M$. In the subsequent subsection, we show how making one additional modification can reduce this number to exactly one (subsequent to the first step). The resultant method, due to Gibson and Bruck [40], is known as the next-reaction method (NRM).

2.4.2 Next-reaction method

In an attempt to improve the efficiency of the SSA, Gibson and Bruck [40] did what few investigators had thought to do in the preceding 25 years, revisit the FRM. The method that they developed, the next-reaction method, can be thought of as an enhanced version of the FRM with three primary modifications: the two constituting the simple optimization described in the previous subsection and an additional variable transformation, which we will discuss here, that allows random samples to be reused at each simulation step.

In this dissertation, we have separated the first two modifications from the latter in order to more clearly convey what exactly constitutes the NRM and what, we believe, is the main contribution of Ref. [40], i.e., the variable transformation formula. It is not that we believe the modifications of the previous subsection to be unimportant. Quite to the contrary, the FRM reformulated in this simple way is much more amenable to efficient methods of computer science, which is extremely significant in and of itself. However, they are not novel innovations, *per se*. The variable transformation formula on the other hand, which reduces the number of random number generations required at each simulation step to a single one, *is* novel and is what makes the NRM a non-trivial extension of the FRM.

The general approach of all Monte Carlo sampling techniques is to take a random sample from a simple distribution for which standard sampling approaches exist, e.g., the unit-uniform distribution $\mathcal{U}(0,1)$, and transform it, in some way, into a sample from a different distribution that is not so easily, or feasibly, sampled. Equation (2.17) is an example of this, where a sample r_μ from $\mathcal{U}(0,1)$ is transformed into a sample τ_μ from Eq. (2.16). The great insight that Gibson and Bruck had in [40] was recognizing that the same approach can be used for generating random samples of τ_μ from Eq. (2.16) at time t by transforming the "leftover" samples τ_μ° obtained from Eq. (2.16) at time $t-\widetilde{\tau}$. The formula for doing so is known as the *next-reaction transformation theorem*.

***Next-reaction transformation theorem* [40].** *Let τ_μ° be a sample from Eq. (2.16) at time $t-\widetilde{\tau}$ and let $\tau_\mu^\circ \neq \widetilde{\tau}$, where $\widetilde{\tau}=\min\{\tau_\nu^\circ\}$, $\nu=1,\ldots,M$. New tentative next-reaction times $\{\tau_\nu\}$ at time t can be obtained for all reactions, other than the one that just fired, by applying the transformation formula*

$$\tau_\mu = F_\mu^{-1}\left[\left(\frac{F_\mu(\tau_\mu^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}},t-\widetilde{\tau})-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}},t-\widetilde{\tau})}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}},t-\widetilde{\tau})}\right)\middle|\,\mathbf{x}_{\mathbf{t}},t\right]. \tag{2.29}$$

To see how this works, consider the common case of τ-independent propensities where Eq. (2.16) can be written as

$$p_\mu(\tau|\mathbf{x_t},t)d\tau = a_\mu(t)\exp\left[-a_\mu(t)\tau\right]d\tau. \tag{2.30}$$

Using Eq. (C.1) in Appendix C, we can get the distribution function

$$F_\mu(\tau|\mathbf{x_t},t) = \int_0^\tau p_\mu(\tau'|\mathbf{x_t},t)d\tau' = 1-\exp\left[-a_\mu(t)\tau\right], \tag{2.31}$$

and the associated inverse function

$$F_\mu^{-1}(r|\mathbf{x_t},t) = \frac{-\ln(1-r)}{a_\mu(t)}. \tag{2.32}$$

Now, using Eqs. (2.31) and (2.32) in concert with Eq. (2.29), we get

$$\begin{aligned}
\tau_\mu &= F_\mu^{-1}\left[\left.\frac{\left(1-e^{-a_\mu(t-\widetilde{\tau})\tau_\mu^\circ}\right)-\left(1-e^{-a_\mu(t-\widetilde{\tau})\widetilde{\tau}}\right)}{1-\left(1-e^{-a_\mu(t-\widetilde{\tau})\widetilde{\tau}}\right)}\right|\mathbf{x_t},t\right] \\
&= F_\mu^{-1}\left[1-e^{-a_\mu(t-\widetilde{\tau})(\tau_\mu^\circ-\widetilde{\tau})}\middle|\,\mathbf{x_t},t\right] \\
&= \frac{a_\mu(t-\widetilde{\tau})}{a_\mu(t)}(\tau_\mu^\circ-\widetilde{\tau}).
\end{aligned} \tag{2.33}$$

Thus, we see that the variable transformation amounts, in this case, to a simple rescaling of the residuals $(\tau_\mu^\circ-\widetilde{\tau})$ (see Fig. 2.3) that accounts for the changes in the reactant species populations due to the last reaction firing.[2] If the reaction is unaffected by the last firing, we see that Eq. (2.33) reduces to Eq. (2.27), just as we expected based on intuitive arguments. The variable transformation theorem Eq. (2.29) puts this intuition on firm theoretical ground.

A typical NRM simulation proceeds by first generating M random samples $\{r_\nu\}$ from $\mathcal{U}(0,1)$ and using them to generate M values of $\{\tau_\nu\}$, just as in the original FRM. The clock is then advanced by $\tau=\min\{\tau_\nu\}$ and the corresponding

[2]Note that Eq. (2.33) is slightly different from the expression given in Ref. [40] in that it is a *relative* time version of the transformation formula. The relation in Ref. [40] is recovered by simply adding to Eq. (2.33) the current time t.

reaction enacted by changing the species populations associated with that reaction and updating the propensities of all affected reactions. The latter operation can be performed by accessing the dependency graph [40]. For the reaction that just fired, a new value of τ_μ is then generated, as before, using a new sample of $\mathcal{U}(0,1)$. For all other reactions, however, Eq. (2.29) is used [Eq. (2.33) in the τ-independent case]. If we operate in absolute time, then Eq. (2.29) need only be evaluated for the affected reactions (again, the dependency graph can be used for this). If we operate in relative time, then it must be evaluated for all reactions, although, as discussed above, it reduces to Eq. (2.27) for all unaffected reactions.

An important implementation detail that deserves discussion here is what to do if a reaction becomes inactive. In this situation, $a_\mu = 0$ and $\tau_\mu = \infty$. Clearly, when the reaction becomes active again we cannot use these values in Eq. (2.29) to generate a new value of τ_μ. What we *can* do, however, is use the values from the last simulation step at which the reaction was active. Thus, at each step of a NRM simulation, values of a_μ, τ_μ and τ are stored for each reaction and used in Eq. (2.29) at the next step at which the reaction is active. Usually this is the subsequent step, but sometimes it is not.[3]

With the transformation formula (2.29) in hand and the implementation of the NRM outlined, our final task in this subsection is to prove that Eq. (2.29) does, in fact, produce valid sample values of Eq. (2.16). To do this, we must first recognize that once the time step $\widetilde{\tau}$ (from the previous simulation step) has been chosen as $\min\{\tau_\nu^\circ\}$, the leftover values of $\{\tau_\nu^\circ\}$ can be viewed as random samples from a density function that governs the probability, at time $t - \widetilde{\tau}$, that τ_μ° lies within $[\tau^\circ, \tau^\circ + d\tau^\circ)$ *given that* $\tau_\mu^\circ > \widetilde{\tau}$. Basically, this is different from Eq. (2.16) because we now have additional information regarding the values of $\{\tau_\nu^\circ\}$, i.e., that they

[3]This issue was briefly addressed in notes 11 and 14 of Ref. [40].

are all larger than $\widetilde{\tau}$.

This probability can be expressed using the multiplication law of probability theory as

$$\Pr\{\tau^\circ < \tau_\mu^\circ < \tau^\circ + d\tau^\circ | \tau_\mu^\circ > \widetilde{\tau}\} = \frac{\Pr\{\tau^\circ < \tau_\mu^\circ < \tau^\circ + d\tau^\circ \cap \tau_\mu^\circ > \widetilde{\tau}\}}{\Pr\{\tau_\mu^\circ > \widetilde{\tau}\}} \qquad (2.34)$$

(note that $\cap$ denotes '*and*'). The numerator can then be written as

$$\begin{aligned} \Pr\{\tau^\circ < \tau_\mu^\circ < \tau^\circ + d\tau^\circ \cap \tau_\mu^\circ > \widetilde{\tau}\} &= \begin{cases} 0 & for \quad \tau^\circ \leq \widetilde{\tau} \\ p_\mu(\tau^\circ | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}) d\tau^\circ & for \quad \tau^\circ > \widetilde{\tau} \end{cases} \\ &= H(\tau^\circ - \widetilde{\tau}) p_\mu(\tau^\circ | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}) d\tau^\circ, \qquad (2.35) \end{aligned}$$

where $H(\cdot)$ is the Heaviside step function [Eq. (A.3)]. The denominator can be written as

$$\begin{aligned} \Pr\{\tau_\mu^\circ > \widetilde{\tau}\} &= \Pr\{F_\mu^{-1}(r_\mu^\circ | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}) > \widetilde{\tau}\} \\ &= \Pr\{r_\mu^\circ > F_\mu(\widetilde{\tau} | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})\} \\ &= 1 - F_\mu(\widetilde{\tau} | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}), \qquad (2.36) \end{aligned}$$

where again we have used Eqs. (C.1) and (C.2) from Appendix C. Substituting Eqs. (2.35) and (2.36) into Eq. (2.34) then gives

$$\Pr\{\tau^\circ < \tau_\mu^\circ < \tau^\circ + d\tau^\circ | \tau_\mu^\circ > \widetilde{\tau}\} = H(\tau^\circ - \widetilde{\tau}) \frac{p_\mu(\tau^\circ | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}{1 - F_\mu(\widetilde{\tau} | \mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})} d\tau^\circ. \qquad (2.37)$$

In Eq. (2.37) we now have the probability density function that governs the random samples $\{\tau_\nu^\circ\}$, which are leftover tentative next-reaction times from time $t - \widetilde{\tau}$. We also have the expression for calculating new random samples $\{\tau_\nu\}$ at time t from these, i.e., Eq. (2.29). Thus, we have all the information that we need to determine the (unknown) probability density function $Q_\mu(\tau)$ that governs the $\{\tau_\nu\}$. Let us see if it, in fact, corresponds to $p_\mu(\tau | \mathbf{x}_\mathbf{t}, t)$ from Eq. (2.16).

We apply the *random variable transformation (RVT) theorem* [44] (Appendix A) to Eq. (2.37) using Eq. (2.29),

$$\begin{aligned} Q_\mu(\tau) &= \int_{\widetilde{\tau}}^{\infty} d\tau^\circ \frac{p_\mu(\tau^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})} \\ &\times \delta\left\{\tau - F_\mu^{-1}\left[\left(\frac{F_\mu(\tau^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}) - F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}\right)\middle| \mathbf{x}_\mathbf{t}, t\right]\right\}. \end{aligned} \tag{2.38}$$

Here, $\delta(\cdot)$ is the Dirac delta function, defined in Eq. (A.1) of Appendix A. It is important to recognize that the quantities τ and τ° that appear in the delta function are random *variables*, of which τ_μ and τ_μ° are random *samples*. The former are abstract quantities while the latter, which appear in Eq. (2.29), are literal numbers. Also note that the effect of $H(\tau^\circ - \widetilde{\tau})$ in Eq. (2.37) is to change the lower limit of integration in Eq. (2.38).

At first glance, Eq. (2.38) looks imposing. However, we can simplify things greatly by making the following variable substitution,

$$u \equiv \frac{F_\mu(\tau^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau}) - F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}, \tag{2.39}$$

which leads to

$$du = \left(\frac{dF_\mu(\tau^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})/d\tau^\circ}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}\right) d\tau^\circ = \frac{p_\mu(\tau^\circ|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})}{1-F_\mu(\widetilde{\tau}|\mathbf{x}_{\mathbf{t}-\widetilde{\tau}}, t-\widetilde{\tau})} d\tau^\circ, \tag{2.40}$$

and allows us to rewrite Eq. (2.38) as

$$Q_\mu(\tau) = \int_0^1 du \times \delta\left\{\tau - F_\mu^{-1}(u|\mathbf{x}_\mathbf{t}, t)\right\}. \tag{2.41}$$

We can then simplify things even more by defining

$$w \equiv F_\mu^{-1}(u|\mathbf{x}_\mathbf{t}, t), \tag{2.42}$$

which implies that

$$\begin{aligned} u &= F_\mu(w|\mathbf{x}_\mathbf{t}, t), \\ du &= \left(\frac{dF_\mu(w|\mathbf{x}_\mathbf{t}, t)}{dw}\right) dw = p_\mu(w|\mathbf{x}_\mathbf{t}, t) dw. \end{aligned} \tag{2.43}$$

Finally, substituting Eqs. (2.42) and (2.43) into Eq. (2.41) gives

$$\begin{aligned} Q_\mu(\tau) &= \int_{-\infty}^{\infty} dw \times p_\mu(w|\mathbf{x_t}, t)\delta(\tau - w) \\ &= p_\mu(\tau|\mathbf{x_t}, t), \end{aligned} \tag{2.44}$$

where the last equality is due to the "sifting property" of the Dirac delta function [Eq. (A.2)]. Thus, we see that the values of τ_μ given by Eq. (2.29) do, indeed, correspond to random samples of Eq. (2.16) at time t. □

2.5 Spatial extensions and diffusion—Next-subvolume method

The DM and FRM of Secs. 2.3 and 2.4 are mathematically equivalent methods for sampling the next-reaction probability density function of Eq. (2.10). Conceptually, however, the two approaches are obviously quite different. On the one hand, the DM can be seen as considering all of the reactions in the system *as a group*. The time to the next firing within the group is determined via, e.g., Eq. (2.15), and the identity of the firing reaction within the group is determined via Eq. (2.14). The FRM, on the other hand, considers each reaction in the system *individually*, i.e., as a group of one. A tentative next-reaction time is calculated for each reaction using, e.g., Eq. (2.17), which is simply Eq. (2.15) with $M = 1$. The reaction to fire is then chosen as that corresponding to the smallest of these values.

If we think of these two approaches as representing two ends of a spectrum with regard to the grouping of reactions, the fact that they are both mathematically-sound procedures for sampling Eq. (2.10) implies that any method intermediate between them is also a mathematically-sound procedure. In other words, we can

group reactions into subgroups if we like and then apply successive iterations of the DM, the FRM, or a combination of the two in order to determine τ and μ. The choice of approach will depend on the system being investigated and the preference of the investigator.

An example of such a hybrid SSA implementation is the next-subvolume method (NSM) of Elf and Ehrenberg [31, 60]. These authors were specifically concerned with stochastic simulations of coupled reaction and diffusion processes in inhomogeneous systems. In order to account for the inhomogeneity, they discretize the system into multiple homogeneous subvolumes. Identical copies of each reaction can then fire in each subvolume and species can diffuse between neighboring subvolumes. Diffusion is modeled in this regard as a type of chemical reaction, often as a first-order process though it can take on other forms as well (e.g., assisted diffusion through a membrane via a Michaelis-Menten mechanism). In Fig. 2.4, we show an example of such a spatial discretization.

Discretizing space in this way leads to a natural grouping of reaction and diffusion events based on their physical location within the system. The approach of Elf and Ehrenberg [31] is to sum the propensities of each subvolume, which we will denote as a_0^l, $l = 1, \ldots, L$ (L being the number of subvolumes), and then choose the subvolume within which the next reaction will fire as in the FRM, i.e., using Eq. (2.17) with a_μ replaced with a_0^l. Once the subvolume has been chosen, the identity of the firing reaction is determined as in the DM, i.e., using Eq. (2.14) with a_ν replaced with a_ν^l (the propensity of the local copy of R_ν) and a_0 replaced with a_0^l. This procedure parses out the computational load into two stages and can result in significant savings relative to straightforward implementations of the SSA to inhomogeneous systems (e.g., [55]), particularly if optimized variants of

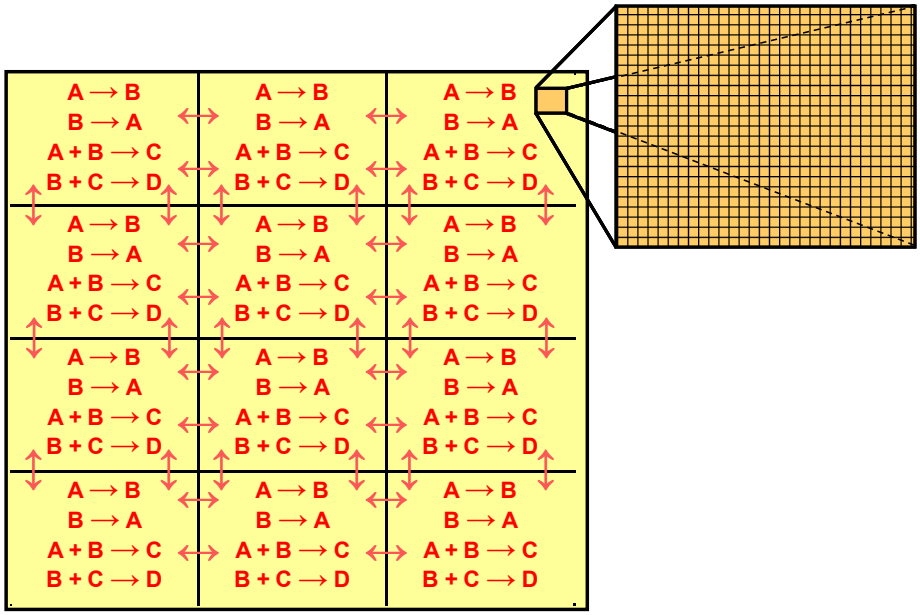

Figure 2.4: Example of a discretization of an inhomogeneous reaction volume into numerous homogeneous subvolumes. Within each subvolume local versions of each reaction can fire. Species can also diffuse between neighboring subvolumes. In principle, each subvolume has an underlying microscopic structure which is coarse-grained out at the mesoscopic level.

the DM [22, 39, 75] and FRM [40] are utilized.

It is important to recognize that strategies for improving the performance of kMC algorithms by grouping reactions appear throughout the literature, though often in different guises. For example, Gillespie describes in a recent review [49] an approach that he terms the first-family method, which is exactly equivalent to the NSM except that the groupings are arbitrary, i.e., they are done for convenience and need not be based on physical location, though they certainly can be. Other examples include Blue et al. [9], who proposed grouping pairs of reactions, and then pairs of pairs of reactions, and so on, in a binary tree-like structure, and Fricke and Wendt [39], who employed a three-tiered approach that involves grouping by process (e.g., reaction and diffusion) and then by subvolume. Blue et al. [9]

basically apply the DM at each level of their tree but since each level consists of only two choices the procedure is simple and fast. The technique has been used by Bernstein [6] within a spatial context. Fricke and Wendt [39] use the DM at the level of processes and for choosing reactions within subvolumes but employ an acceptance-rejection method, which is an alternative sampling procedure not discussed in this dissertation, at the level of subvolumes. Interestingly, similar approaches have been proposed recently by Schulze [101] and Slepoy et al. [107], the latter of which is specifically concerned with applications to the SSA.

Chapter 3

Development of a Novel Accelerated-Stochastic Simulation Approach: The Partitioned-Leaping Algorithm

By construction, the exact-stochastic simulation approach of Chapter 2, along with all of its variants, is fundamentally limited in applicability because of its "one reaction at a time" nature. In the SSA, one reaction firing is simulated at every simulation step, with the time step τ being inversely proportional to the propensity of the fastest reaction in the system. For example, in the DM version of the SSA (Sec. 2.3), this propensity contributes the most to the sum $a_0 \equiv \sum_\nu a_\nu$ in Eq. (2.15). In the FRM variant (and the NRM by extension—Sec. 2.4), this reaction will most often correspond to $\min\{\tau_\nu\}$, since $\langle \tau_\mu \rangle = 1/a_\mu$. As such, if one reaction is much faster than all the rest, either because it has larger reactant populations or rate parameters, then most of the computational effort will be spent repeatedly simulating firings of that reaction. This severely limits the utility of the method, especially in situations where rare events are of interest.

This limitation of the SSA has been well known since its inception [41]. Interest in developing methods for overcoming it, however, has only become prevalent recently, driven primarily by the ever-increasing interest in the role and consequences of stochastic noise in biological systems, where large disparities in dynamical timescales are common [34, 35, 61, 69, 74, 89, 90, 99, 100]. Indeed, a multitude of approximate approaches have been proposed in recent years aimed at accurately capturing stochastic effects in (bio)chemical reaction networks at sig-

nificantly reduced computational cost relative to the SSA. These approaches have been wide-ranging in scope, from methods for solving simplified versions of the master equation [Eq. (2.2)] [79, 80], to techniques for systematically reducing the complexity of reaction networks [27, 38, 64], to strategies that ignore the exact times at which reactions fire and group multiple firings into a single simulation step [3, 14, 20, 21, 25, 47, 50, 56, 83, 84, 91, 92, 109, 114]. This latter class of technique, due originally to Gillespie [46, 47], is known as "leaping," and is the focus of this chapter.

In what follows, we will present the underlying theory laid down by Gillespie [46, 47] on which all leaping methods are based. We will then discuss the original and modified versions of the "τ-leaping" method [20, 47, 50] and present a novel leaping variant known as "partitioned leaping" [56], developed by the author of this dissertation. We will conclude by discussing a number of alternative leaping strategies that have been proposed in the literature and consider various extensions that can expand the utility of the approach.

3.1 Building a bridge to the continuum[1]

The fundamental idea underlying the leaping approach is if a species has a large population then a single firing of a reaction involving that species will only change the reaction rate, or propensity, minimally [46, 47]. As such, if we choose to ignore the *exact* moments at which reactions fire, which is usually not of great interest, we can achieve significant computational accelerations by assuming that the reaction dynamics obey *Poissonian statistics*. Simply put, the Poisson distribution

[1]The material in this subsection has been adapted from Ref. [56].

[see Appendix A, Eqs. (A.8)–(A.10)] governs the probability of discrete events occurring over continuous intervals, where the probability of an event occurring is constant throughout the interval. In our case, the number of reaction firings is the discrete quantity, τ is the continuous interval of time, and we are positing that a_μ, the reaction probability, or propensity, remains constant throughout τ. Defining $K_\mu(\tau|\mathbf{x_t}, t)$ as the number of times reaction R_μ fires within τ given the state $\mathbf{x_t}$ at time t, we can express the Poissonian assumption as

$$K_\mu(\tau|\mathbf{x_t}, t) \approx \mathcal{P}_\mu(a_\mu(t)\tau), \tag{3.1}$$

where the Poisson random variable $\mathcal{P}_\mu(a_\mu(t)\tau)$ has mean and variance $a_\mu(t)\tau$ [Eq. (A.10)].

The advantages of implementing a simulation method based on Eq. (3.1) are clear: assuming that a "suitable" time interval τ can be obtained over which all reaction propensities remain essentially constant, we can simulate multiple firings of *every* reaction in a system at every simulation step by generating Poisson random deviates for each reaction based on the current values of the propensities.[2] Contrast this with the single reaction firing simulated at each step of a SSA simulation. Furthermore, if we are careful in our selection of τ, then the error introduced by this procedure will be minimal. This, in its simplest form, is the leaping approach. Of course, obtaining a suitable time step is a non-trivial task, and one that has received significant attention in the literature [20, 47, 50]. We will discuss various such "τ-selection" procedures in Sec. 3.4.

In Ref. [47], Gillespie went a step further, noting that it is a well-known property of the Poisson distribution that it can be approximated by a *normal*, or *Gaussian*, distribution [see Appendix A, Eqs. (A.11)–(A.14)], with the same mean

[2]Standard methods exist for doing this, e.g., Ref. [86].

and variance, when the mean value ($a_\mu\tau$ in this case) is "large" (i.e., $\gg 1$). Thus, in this case, we can make the additional assumption

$$\begin{aligned} K_\mu(\tau|\mathbf{x_t},t) &\approx \mathcal{N}_\mu(a_\mu(t)\tau, a_\mu(t)\tau) \\ &= a_\mu(t)\tau + \sqrt{a_\mu(t)\tau}\times\mathcal{N}(0,1), \end{aligned} \tag{3.2}$$

where the second line is due to the "linear combination theorem" for normal random variables [46, 47], i.e., $\mathcal{N}(m,\sigma^2) = m + \sigma\mathcal{N}(0,1)$, where m is the mean and σ^2 the variance. Written this way, Eq. (3.2) has the form of a *Langevin* equation, a stochastic differential equation (SDE) comprised of a "deterministic" term ($a_\mu\tau$) and a fluctuating "noise" term ($\sqrt{a_\mu\tau}\times\mathcal{N}(0,1)$). This is significant as much confusion exists regarding the source of, and proper functional form for, the noise term in SDEs [46]. Equation (3.2) resolves much of this confusion, at least with regards to *intrinsic* noise.

Finally, Gillespie went one step further and noted that as $a_\mu\tau \to \infty$ the noise term in Eq. (3.2) can be neglected relative to the deterministic term (i.e., when their ratio $a_\mu\tau/\sqrt{a_\mu\tau} = \sqrt{a_\mu\tau} \gg 1$). This gives the "deterministic" approximation

$$K_\mu(\tau|\mathbf{x_t},t) \approx a_\mu(t)\tau. \tag{3.3}$$

Equations (3.1)–(3.3) represent a significant contribution to the field of chemical kinetics, amounting to a *derivation* of the deterministic reaction-rate equation approach from first principles. Deterministic kinetics is commonly presented to undergraduate chemistry and chemical engineering students from a phenomenological point of view within an ODE context. However, Eqs. (3.1)–(3.3), and the conditions identified by Gillespie for transitioning between them, provide a deeper understanding, illustrating that chemical kinetics is neither phenomenological nor continuous and deterministic by nature. Rather, it has a firm theoretical foundation based on probabilistic interactions between discrete molecular entities, with

the common approach of treating the dynamics continuously and deterministically being an *approximation* in the large-number limit. This physically-based view of chemical kinetics has significant educational value. We will see in what follows that it has great practical utility as well.

3.2 τ-leaping

Strictly speaking, the method known as τ-leaping [18, 20, 47, 50] utilizes only the discrete-stochastic Poisson approximation of Eq. (3.1) (as alluded to above) along with a *proviso* for switching to the SSA when the expected number of reaction firings within the interval τ is "small" (typically $\lesssim 10$). Moreover, numerous refinements and modifications have been made to the approach since its inception in Ref. [47]. In particular, attempts to apply the original τ-leaping algorithm to prototypical biochemical reaction networks encountered problems with species populations becoming negative [25, 109]. This is due to the fact that the Poisson distribution is positive unbounded and can give rise to unphysical numbers of reaction firings.[3] As a result, numerous strategies have been proposed for overcoming this problem. These include replacing the Poisson description with one based on a bounded binomial distribution [25, 109] and a somewhat *ad hoc* procedure for distinguishing "critical" reactions in danger of exhausting their reactant populations

[3] In fact, given a long enough simulation, this will invariably occur in any reaction system. However, it is important to note that the difficulties experienced in Refs. [25] and [109] appear to have been due to a specific type of biochemical reaction, gene-protein binding and unbinding. Since there is only a single gene, these reactions can only occur once in a simulation step. It appears, however, that Poisson random deviates were generated in Refs. [25] and [109] that were larger than unity, causing the gene populations to go negative. Note that the method presented in Sec. 3.3 naturally overcomes this problem.

from non-critical ones [18].

Another important point is that a distinction has been made between "explicit" leaping methods that utilize the value of the propensity at the beginning of the time step, i.e., $a_\mu(t)$ (the original τ-leaping algorithm is explicit), and "implicit" methods that incorporate the value at the end of the time step, i.e., $a_\mu(t+\tau)$. Indeed, in analogy with methods used in the numerical solution of ODEs, Petzold and co-workers have introduced various implicit τ-leaping methods [24, 92, 93]. These methods have been shown to maintain numerical stability in situations where explicit methods cannot. The cost, however, is that fluctuations are dampened. In the case of ODEs, this is a desirable property of a method since any fluctuations that are seen are purely numerical in nature. This is not true in the case of stochastic simulations, however, because the fluctuations have physical significance, being intrinsic to the probabilistic nature of the interactions. To address this problem, a strategy for intermixing explicit and implicit τ-leaping has been proposed [21]. Nevertheless, the legitimacy of implicit leaping remains somewhat questionable at the current time [49].

In this dissertation, we will not delve into these modifications in any great detail. An overview of these methods and other leaping approaches is provided in Sec. 3.5. For a deeper discussion, the interested reader can consult Ref. [20] for the latest incarnation of explicit τ-leaping and Refs. [24, 92, 93] for a discussion of implicit τ-leaping. However, for the sake of subsequent comparison with the partitioned-leaping algorithm (PLA) [56] (Sec. 3.3), we present here a simplified algorithm for *explicit*, *Poisson-based* τ-leaping that implements a simple "try again" procedure [18] for avoiding negative species populations.

Algorithm 2 (Gillespie and co-workers, 2001–2006 [18, 20, 47, 50]): ***Explicit Poisson τ-leaping with simple negative population check.***

1. *Initialize: Define reaction network, rate parameters, initial species populations.*
2. *Determine an "appropriate" time step τ (see Sec. 3.4).*
3. *If $a_0\tau$, the* total *number of expected reaction firings within τ, is >10, determine how many times each reaction fires within τ by generating M Poisson random deviates $\{k_\nu\}$ using, e.g., the method in Ref. [86].*
4. *If $a_0\tau \leq 10$, use your favorite SSA approach to determine the time and type of the next reaction firing in the system (this results in a new value of τ).*
5. *Fire all reactions, update the species populations and propensities, and advance the clock to time $t + \tau$.*
6. *If any $X_i(t + \tau) < 0$, reverse all updates, set $\tau = \tau/2$ and return to step 3 (i.e., try again).*
7. *If the stopping criterion has been met then terminate, otherwise go to step 2.*

3.3 Considering reactions individually—The 'partitioned-leaping algorithm'

3.3.1 The idea

In the view of the author of this dissertation, there are three main shortcomings of Algorithm 2, the simple τ-leaping algorithm: (i) the incorporation of the SSA is somewhat forced and unnatural, (ii) it does not utilize the entire theoretical

framework developed in Ref. [47], and reviewed in Sec. 3.1, for bridging from the discrete-stochastic regime to the continuous-deterministic, and (iii) the simple procedure for avoiding negative populations, while successful, can be inefficient [18], particularly when considering reactions involving species with very small populations, such as genes (see footnote 3).

It turns out, however, that by making one simple adjustment to our thinking all three of these shortcomings can be resolved. Specifically, we follow the example of the FRM variant of the SSA (Sec. 2.4) and consider each reaction in the system on an *individual* basis. This is not to say that we consider the reactions independently, we do not. The interconnectivity of the network is accounted for in the process of selecting τ (see Sec. 3.4). However, by definition, τ is the time interval over which each reaction in the system can be considered to be a statistically independent Poisson process. Thus, the primary change that we make to τ-leaping is to *classify* each reaction in the system into one of four categories once the time step τ has been determined. The categories correspond to the three levels of description discussed in Sec. 3.1, i.e., Eqs. (3.1)–(3.3), as well as a *proviso*, analogous to Gillespie's [47], for treating a reaction at the exact-stochastic level if it is expected to fire on the order of once or less within τ.

This procedure amounts to a theoretically justifiable procedure for partitioning reactions into 'very slow,' 'slow,' 'medium,' and 'fast' subsets based on the propensity values, the calculated time step τ, and the criteria identified by Gillespie [47] for transitioning between the descriptions Eqs. (3.1)–(3.3). The classifications are made as follows.

- If $a_\mu \tau \lesssim 1 \rightarrow$ *exact stochastic* (very slow).
- If $a_\mu \tau > 1$ but $\not\gg 1 \rightarrow$ *Poisson* (slow).

- If $a_\mu \tau \gg 1$ but $\sqrt{a_\mu \tau} \not\gg 1 \rightarrow$ *Langevin* (medium).
- If $\sqrt{a_\mu \tau} \gg 1 \rightarrow$ *deterministic* (fast).

These classifications constitute the foundation of the partitioned-leaping approach. At each simulation step, a time step τ is calculated (see Sec. 3.4) and each reaction classified in the manner outlined above. The numbers of firings of each reaction are then determined based on these classifications and the system is evolved accordingly. For reactions classified at the exact-stochastic (ES) level, a tentative-next reaction time, which we will subsequently denote as τ_μ^{ES}, can be calculated in the manner discussed in Sec. 2.4 [e.g., using Eqs. (2.17) and (2.33)] and the reaction deemed to fire if $\tau_\mu^{\mathrm{ES}} \leq \tau$. For reactions classified at coarser levels, Eqs. (3.1), (3.2) and (3.3) are used.

This procedure resolves the three shortcomings of τ-leaping discussed above because: (i) being analogous to the FRM, the SSA is naturally and seamlessly incorporated into the algorithmic framework, (ii) classifying reactions individually (rather than classifying the entire system, as in the original τ-leaping algorithm [47]) allows the entire theoretical framework of Sec. 3.1 to be utilized, and (iii) the ES classification provides a natural mechanism by which reactions with very small reactant populations are prevented from firing multiple times within a simulation step.

Before the approach can be implemented in a practical sense, however, numerous technical issues must be addressed, some of which are quite subtle. In the following subsection we discuss these issues and strategies for overcoming them.

3.3.2 Technical issues

The first technical issue that we must consider involves the inclusion of the ES classification into the algorithm and the random nature of tentative next-reaction times. As explained above, the procedure that we propose is to classify each reaction in the system into one of four categories based on the values $\{a_\nu\tau\}$. If $a_\mu\tau \lesssim 1$, reaction R_μ is to be classified as ES and a tentative next-reaction time $\tau_\mu^{\rm ES}$ calculated. If $\tau_\mu^{\rm ES} \leq \tau$, then we can say that R_μ fires *at least* once within the interval. However, we must recognize that if we simply fire the reaction once and advance the clock to $t+\tau$, then we are precluding the possibility that R_μ fires again within the interval $(\tau - \tau_\mu^{\rm ES})$. The error introduced by doing so might be small. However, in the opinion of this author, the mere fact that the ES classification is being included at all indicates that maintaining detailed accuracy is desired. Thus, this shortcoming should be resolved in a rigorous manner. If one is not interested in maintaining such detailed accuracy then the ES classification can be excluded from the algorithm altogether, leaving "Poisson" as the finest level of description.

To overcome this complication we employ an iterative procedure that involves (i) calculating τ, (ii) classifying reactions, (iii) calculating $\{\tau_\nu^{\rm ES}\}$ values for any ES reactions, (iv) identifying the smallest of these, (v) changing τ to $\min\{\tau_\nu^{\rm ES}\}$ if $\min\{\tau_\nu^{\rm ES}\} < \tau$, (vi) reclassifying reactions based on the new value of τ, and (vii) repeating as necessary.[4] Step (vi) is necessary because with a smaller τ some reactions may become classified as ES that previously were not. Step (vii) is necessary because the smallest of these new tentative next-reaction times may, again, be smaller than the new τ.

[4]The number of iterations required in this procedure is definitively finite. In extreme situations, if τ is continually reduced, at some point *all* reactions will become classified as ES. Reclassifications will then no longer be necessary and the iterative loop will terminate.

A couple of important points deserve mention here. First, it is always valid to decrease τ, as in step (v) above, because if the Poisson approximation Eq. (3.1) is valid for a given τ then it is assured to be valid for any time interval smaller than τ. Second, if all reactions are classified as ES, then the algorithm will revert to the FRM or NRM variant (depending on how one chooses to implement the method) of the SSA. In this situation, it is legitimate to jump to the time at which the next reaction fires in the system even if this happens to be larger than the initial calculated value of τ. Finally, although there are cases where it is valid to increase τ even if all reactions are *not* classified as ES, it is difficult to distinguish these from the cases in which it is not valid to do so, especially when considering large, complex reaction networks. Thus, in general, one should never increase τ unless all reactions are classified as ES.

Another technical issue that we must consider concerns the proper use of Eq. (2.29), the next-reaction transformation formula, in our algorithm. Recall that Eq. (2.29) provides a recipe for calculating new tentative next-reaction times from the old values of $\tau_\mu^{\rm ES}$, a_μ and τ. In the NRM, the "old" values are usually those from the previous simulation step. However, in the case of partitioned leaping, it will often happen that a reaction that was classified as ES at the previous simulation step will not be so at the current step. Thus, there is no need to use Eq. (2.29) at the current step. Nevertheless, we can still use Eq. (2.29) at the next step that the reaction is classified as ES provided that we store the values of $\tau_\mu^{\rm ES}$, a_μ and τ from the current step. In fact, we discussed this issue in the 'Next-reaction method' subsection of Sec. 2.4 in a different context, when reactions become inactive (see footnote 3). Thus, we simply extend the approach of "carrying over" the values of $\tau_\mu^{\rm ES}$, a_μ and τ used in the NRM to the leaping algorithm.

We must also consider the problem of negative populations, which has received extensive mention in the literature [18, 25, 109]. One of the great strengths of the method proposed here, in fact, is that this problem is largely overcome simply via the structure of the algorithm. If any reaction has a reactant population that is very small, say on the order of unity, then that reaction will automatically be flagged as ES because even one firing of the reaction will change the propensity "appreciably." Thus, there will be no chance of the reaction firing multiple times in a step and resulting in negative populations. However, it is important to recognize (as discussed in footnote 3 above) that because the Poisson distribution (and the Gaussian for that matter) is positively unbounded, it is still possible that an unphysical number of reaction firings can be generated in a simulation step. In our case, however, this should be exceedingly rare. Thus, we can protect against this rare occurrence by employing the simple "try again" procedure that was proposed in Ref. [18] and included in the simple τ-leaping algorithm above (Algorithm 2).

Finally, the last issue that we must consider is the fact that using Eqs. (3.2) and (3.3) for Langevin and deterministic reactions, respectively, will result in values that are real numbers rather than integers. Since it is difficult to determine at what point a continuous population description is acceptable in lieu of an integer description, we choose to round these values before updating the species populations. In Ref. [47], Gillespie argued that the use of Eq. (3.2), when appropriate, as opposed to Eq. (3.1), is an improvement computationally because generating normal random deviates is faster than generating Poisson random deviates. Clearly, therefore, some of this improvement is negated by including the rounding operation, although we have yet to quantify to what extent this matters (intuitively speaking, it probably matters little). The same argument could be made with regards to deterministic reactions. However, the elimination of the random num-

ber generation operation for these reactions should more than compensate for the added cost of rounding.

3.3.3 The algorithm

With the technical issues of the previous subsection accounted for, we now present the partitioned-leaping algorithm (PLA), which represents the major contribution of this dissertation.

Algorithm 3 (Harris and Clancy, 2006 [56]):
Partitioned-leaping algorithm (PLA).

1. *Initialize: Define reaction network, rate parameters, initial species populations, define* $\ll 1$, ≈ 1 *and* $\gg 1$.[5]
2. *Determine the* ***initial*** *time step* τ *(see Sec. 3.4).*
3. *Classify all reactions (not already classified as ES) using the criteria presented above.*
4. *For all (newly classified) ES reactions, calculate tentative next-reaction times,* $\tau_\mu^{\rm ES}$*, using, e.g., Eqs. (2.17) and (2.33).*[6]
5. *If* $\min\{\tau_\nu^{\rm ES}\} \neq \tau$ ***and*** *all reactions are classified as ES, set* $\tau = \min\{\tau_\nu^{\rm ES}\}$.
6. *If* $\min\{\tau_\nu^{\rm ES}\} < \tau$*, set* $\tau = \min\{\tau_\nu^{\rm ES}\}$ *and return to step 3.*

[5] The parameter '$\ll 1$' quantifies the concept of "essentially constant" and is used in τ-selection (Sec. 3.4), '≈ 1' is used for classifying reactions at the ES level and '$\gg 1$' is used for classifying reactions at the Langevin and deterministic levels (see Sec. 3.1). Typical values that we use are 0.01–0.05, 3 and 100, respectively.

[6] In the τ-dependent case, the more general approach outlined in Sec. 2.4 must be used.

7. *Determine the set of reaction firings* $\{k_\nu(\tau)\}$ *using the appropriate formulas and update the species populations.*[7]

8. *If any* $X_i(t+\tau) < 0$*, reverse all population updates, set* $\tau = \tau/2$ *and return to step 3 (i.e., try again).*

9. *Advance the clock to* $t+\tau$ *and return to step 2 if stopping criterion not met.*

An important aspect of this approach is the minimal amount of user intervention required for implementation. Indeed, once the reaction network is defined and the associated rate parameters set, one need only define three model-*independent* parameters quantifying the concepts ≈ 1, $\gg 1$ and $\ll 1$ (see footnote 5) before instantiating a simulation. The algorithm will then automatically and dynamically partition the reactions into various subsets, correctly accounting for stochastic noise and "leaping" over unimportant reaction events. This ease of use is a particular strength of the method and differentiates it from other leaping approaches that have been proposed (see Sec. 3.5 for a more detailed discussion).

One final element is missing, however, before the algorithm above can be implemented in full: calculation of the time step τ. We address this issue in the following subsection, presenting three alternative approaches that have been proposed in the literature. Each have their strengths and weaknesses and, in the opinion of this author, none of them are ideal. τ-selection turns out to be the most time-consuming aspect of the leaping approach and, hence, represents an area where significant improvements in efficiency can be obtained from novel innovations. In Sec. 3.6, we will briefly address this issue and discuss some possible strategies.

[7]Standard techniques exist for generating Poisson and normal random deviates (e.g., Ref. [86]). For ES reactions, if $\tau_\mu^{\mathrm{ES}} = \tau$ then $k_\mu(\tau) = 1$, otherwise zero.

3.4 Time step calculation—'τ-selection' procedures

Though largely glossed over up until this point, calculation of the time step τ is probably the single most important component of the leaping algorithm. Indeed, the entire approach is reliant upon the validity of the Poisson approximation Eq. (3.1). In Ref. [47], Gillespie provided the basis for all τ-selection procedures. The "leap condition," as he termed it, quantifies the concept of "essentially constant" on which the Poisson approximation depends. It is a constraint imposed on the magnitude of the change of each reaction propensity,

$$\frac{\left|\, \Delta a_\mu(\tau_\mu^{\text{leap}}) \,\right|}{\xi} \leq \epsilon, \;\; (0 < \epsilon \ll 1), \tag{3.4}$$

where $\Delta a_\mu(\tau_\mu^{\text{leap}}) \equiv a_\mu(t + \tau_\mu^{\text{leap}}) - a_\mu(t)$ and ξ is an appropriate scaling factor (see below for further discussion). In applying this constraint, one seeks to identify the time interval τ_μ^{leap} over which the propensity a_μ for reaction R_μ is expected to remain essentially constant (i.e., within a factor of ϵ) *assuming* that the propensities for all other reactions also remain essentially constant (we will illustrate this assumption explicitly below). One can then set τ equal to the smallest of $\{\tau_\nu^{\text{leap}}\}$ as this is the only one for which the Poisson approximation actually holds for *all* reactions. This is analogous to the manner in which the time step is generated in the FRM and NRM variants of the SSA (Sec. 2.4) and thus provides a simple means by which the SSA can be seamlessly integrated into the leaping algorithm.

In what follows, we will discuss three alternative strategies for calculating time steps τ that satisfy the leap condition (3.4). The first uses Eq. (3.4) directly, and is what we term a "reaction-based" (RB) approach. RB τ-selection was initially introduced by Gillespie in Ref. [47] and later modified by Gillespie and Petzold [50], Cao et al. [20] and Harris and Clancy [56]. The second approach is a "species-based" (SB) τ-selection procedure that places constraints, analogous to that in

Eq. (3.4), on the relative changes of each *species population* such that Eq. (3.4) is satisfied for each reaction. The technique, valid in the case of τ-independent propensities (recall the definition of τ independence in Chapter 2), was introduced in Ref. [20] and further modified in Ref. [56]. Finally, we briefly discuss a more recent innovation [1] that involves choosing a time step, generating reaction firings and then *checking* to see whether or not Eq. (3.4) has been violated. If it has, the simulation is "backed up" in a way that avoids introducing bias into the sampling procedure. This "postleap checking" method is reminiscent of a predictor-corrector approach used in the solution of ODEs and is considered by the author of this dissertation to be an important innovation for the future development of more efficient τ-selection strategies.

3.4.1 Reaction-based τ-selection[8]

Underlying theory. The RB τ-selection procedure has been briefly outlined above. The idea is to calculate a "tentative leap time" τ_μ^{leap} (in analogy with the tentative next-reaction times τ_μ^{ES} of the FRM and NRM variants of the SSA—Sec. 2.4) for each reaction directly from Eq. (3.4) and then set $\tau = \min\{\tau_\nu^{\text{leap}}\}$. The calculation is accomplished by Taylor expanding the numerator in Eq. (3.4) out to first order in order to obtain an expression involving τ_μ^{leap}, which is then solved for via algebraic manipulation.

To do this, we first write

$$a_\mu(t+\tau_\mu^{\text{leap}}) = a_\mu\left(\mathbf{X}(t) + \mathbf{\Delta X}(\tau_\mu^{\text{leap}}), \mathbf{\Theta}(t+\tau_\mu^{\text{leap}})\right), \tag{3.5}$$

[8]This subsection contains material that has been presented, in highly condensed form, in Ref. [56].

where $\mathbf{\Delta X}(\tau_\mu^{\text{leap}})$ represents the *changes* in the species populations over the time interval τ_μ^{leap}. Note that we have taken into account here both the time dependence of the propensity with respect to the species populations $\mathbf{X}(t)$ *and* the environmental quantities $\mathbf{\Theta}(t)$.[9] We then perform a Taylor expansion around $a_\mu(t)$ as follows,

$$a_\mu(t+\tau_\mu^{\text{leap}}) \approx a_\mu(t) + \sum_{j=1}^{N} \frac{\partial a_\mu(t)}{\partial X_j} \Delta X_j(\tau_\mu^{\text{leap}}) + \tau_\mu^{\text{leap}} \sum_{q=1}^{Q} \frac{\partial a_\mu(t)}{\partial \theta_q} \dot{\theta}_q(t). \tag{3.6}$$

Here, $\dot{\theta}_q(t) \equiv d\theta_q(t)/dt$ is the time derivative of the qth environmental variable (e.g., temperature, volume) and Q is the total number of environmental variables being considered. Equation (3.6) is written assuming that the environmental quantities are continuous and differentiable in time whereas the species populations are not, i.e., they change discontinuously at the moment of each reaction firing.

Next, we write the state-change element $\Delta X_i(\tau_\mu^{\text{leap}})$ (recall that the subscript i is used for specific species while j is used for summations—Sec. 2.1) as

$$\Delta X_i(\tau_\mu^{\text{leap}}) = \sum_{\nu=1}^{M} z_{\nu i} K_\nu(\tau_\mu^{\text{leap}} | \mathbf{x_t}, t). \tag{3.7}$$

Note that this is an exact expression, not an approximation.[10] Substituting Eq. (3.7) into (3.6) then gives us an approximate expression for the propensity change $\Delta a_\mu(\tau_\mu^{\text{leap}})$,

$$\Delta a_\mu(\tau_\mu^{\text{leap}}) \approx \sum_{\nu=1}^{M} f_{\mu\nu}(t) K_\nu(\tau_\mu^{\text{leap}} | \mathbf{x_t}, t) + \tau_\mu^{\text{leap}} \sum_{q=1}^{Q} \frac{\partial a_\mu(t)}{\partial \theta_q} \dot{\theta}_q(t), \tag{3.8}$$

where

$$f_{\mu\nu}(t) \equiv \sum_{j=1}^{N} z_{\nu j} \frac{\partial a_\mu(t)}{\partial X_j}. \tag{3.9}$$

[9] As such, the derivation given here is quite different from those in Refs. [47] and [50] where only the dependence of the propensity on the species populations was considered.

[10] Also note the distinction between the subscripts μ and ν in Eq. (3.7). μ is used in τ_μ^{leap} because we are considering the leap condition applied specifically to reaction R_μ, whereas ν is used when summing over all the reactions in the system (see Sec. 2.1).

Equation (3.8) evidently expresses the *random variable* $\Delta a_\mu(\tau_\mu^{\rm leap})$ as a *linear combination* of random variables $\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\}$ and $\{\dot\theta_q(t)\}$. The $\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\}$ are random variables because of the random nature of the species interactions while the $\{\dot\theta_q(t)\}$ are random variables due to fluctuations in the environmental quantities. Furthermore, these random variables can be correlated. The number of times reaction R_μ fires can depend on the number of times $R_{\mu'}$ fires, and vice versa. It can also depend on any fluctuations in the environmental quantities and the environmental variables can depend upon each other (e.g., fluctuations in temperature can be related to fluctuations in volume, and vice versa).

As such, we can obtain expressions for both the mean and variance of $\Delta a_\mu(\tau_\mu^{\rm leap})$ by applying to Eq. (3.8) the "linear combination theorem" for random variables [44], i.e., Eqs. (A.15) and (A.16) of Appendix A, giving

$$\left\langle \Delta a_\mu(\tau_\mu^{\rm leap})\right\rangle \approx \sum_{\nu=1}^{M} f_{\mu\nu}(t)\left\langle K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\right\rangle + \tau_\mu^{\rm leap}\sum_{q=1}^{Q}\frac{\partial a_\mu(t)}{\partial\theta_q}\left\langle\dot\theta_q(t)\right\rangle, \tag{3.10}$$

$$\begin{aligned}
{\rm var}\{\Delta a_\mu(\tau_\mu^{\rm leap})\} \approx & \sum_{\nu=1}^{M} f_{\mu\nu}^2(t){\rm var}\left\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\right\} \\
& + \left(\tau_\mu^{\rm leap}\right)^2\sum_{q=1}^{Q}\left(\frac{\partial a_\mu(t)}{\partial\theta_q}\right)^2{\rm var}\{\dot\theta_q(t)\} \\
& + 2\sum_{\nu=1}^{M-1}\sum_{\nu'=\nu+1}^{M} f_{\mu\nu}(t)f_{\mu\nu'}(t){\rm cov}\left\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t),K_{\nu'}(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\right\} \\
& + 2\tau_\mu^{\rm leap}\sum_{\nu=1}^{M}\sum_{q=1}^{Q} f_{\mu\nu}(t)\frac{\partial a_\mu(t)}{\partial\theta_q}{\rm cov}\left\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t),\dot\theta_q(t)\right\} \\
& + 2\left(\tau_\mu^{\rm leap}\right)^2\sum_{q=1}^{Q-1}\sum_{q'=q+1}^{Q}\frac{\partial a_\mu(t)}{\partial\theta_q}\frac{\partial a_\mu(t)}{\partial\theta_{q'}}{\rm cov}\left\{\dot\theta_q(t),\dot\theta_{q'}(t)\right\}.
\end{aligned} \tag{3.11}$$

Here, the first two terms on the right-hand side of Eq. (3.11) account for the variances of the two sets of random variables $\{K_\nu(\tau_\mu^{\rm leap}|\mathbf{x_t},t)\}$ and $\{\dot\theta_q(t)\}$, the third term accounts for the covariances among the reaction firings, the fourth for the covariances between the reaction firings and the environmental quantities, and

the last for the covariances among the environmental quantities.

With these expressions in hand, the approach to τ selection proposed by Gillespie and Petzold [50] is to approximate

$$\Delta a_\mu(\tau_\mu^{\text{leap}}) \approx \left\langle \Delta a_\mu(\tau_\mu^{\text{leap}}) \right\rangle \pm \text{sdev}\left\{ \Delta a_\mu(\tau_\mu^{\text{leap}}) \right\} \tag{3.12}$$

and then obtain an expression for τ_μ^{leap} by substituting Eqs. (3.10) and (3.11) into Eq. (3.12) and then Eq. (3.12) into (3.4). Of course, in general, this can be difficult, if not impossible, to accomplish analytically. By making certain assumptions, however, we can reduce the expressions in Eqs. (3.10) and (3.11) down to more manageable levels that are amenable to analytical treatment.

The simplest and most useful of these assumptions is the Poisson approximation of Eq. (3.1). It is critical to recognize that implicit in this assumption is the stipulation that the reaction firings are *statistically independent* Poisson processes, i.e., $\text{cov}\left\{ \mathcal{P}_\nu\left(a_\nu(t)\tau_\mu^{\text{leap}}\right), \mathcal{P}_{\nu'}\left(a_{\nu'}(t)\tau_\mu^{\text{leap}}\right) \right\} = 0$.[11] Thus, by making this assumption, and using the fact that $\left\langle \mathcal{P}_\nu\left(a_\nu(t)\tau_\mu^{\text{leap}}\right) \right\rangle = \text{var}\left\{ \mathcal{P}_\nu\left(a_\nu(t)\tau_\mu^{\text{leap}}\right) \right\} = a_\nu(t)\tau_\mu^{\text{leap}}$

[11] Note that the goal of implicit leaping methods [24, 92] is to take time steps over which the Poisson approximation Eq. (3.1) does *not* hold. Clearly, this complicates matters as in this case the covariances among the reaction firings do not fall out. It should be of no surprise, therefore, that τ selection procedures for implicit τ-leaping have been conspicuously absent from the literature.

[Eq. (A.10)], we can simplify Eqs. (3.10) and (3.11) to

$$\left\langle \Delta a_\mu(\tau_\mu^{\rm leap}) \right\rangle \approx \tau_\mu^{\rm leap} \left(\sum_{\nu=1}^{M} f_{\mu\nu}(t) a_\nu(t) + \sum_{q=1}^{Q} \frac{\partial a_\mu(t)}{\partial \theta_q} \left\langle \dot{\theta}_q(t) \right\rangle \right), \tag{3.13}$$

$$\begin{aligned} \mathrm{var}\{\Delta a_\mu(\tau_\mu^{\rm leap})\} \approx\ & \tau_\mu^{\rm leap} \sum_{\nu=1}^{M} f_{\mu\nu}^2(t) a_\nu(t) \\ & + \left(\tau_\mu^{\rm leap}\right)^2 \sum_{q=1}^{Q} \left(\frac{\partial a_\mu(t)}{\partial \theta_q} \right)^2 \mathrm{var}\left\{ \dot{\theta}_q(t) \right\} \\ & + 2\tau_\mu^{\rm leap} \sum_{\nu=1}^{M} \sum_{q=1}^{Q} f_{\mu\nu}(t) \frac{\partial a_\mu(t)}{\partial \theta_q} \mathrm{cov}\left\{ \mathcal{P}_\nu\left(a_\nu(t)\tau_\mu^{\rm leap}\right), \dot{\theta}_q(t) \right\} \\ & + 2\left(\tau_\mu^{\rm leap}\right)^2 \sum_{q=1}^{Q-1} \sum_{q'=q+1}^{Q} \frac{\partial a_\mu(t)}{\partial \theta_q} \frac{\partial a_\mu(t)}{\partial \theta_{q'}} \mathrm{cov}\left\{ \dot{\theta}_q(t), \dot{\theta}_{q'}(t) \right\} \end{aligned} \tag{3.14}$$

It is here where we are explicitly making the assumption that *all* of the reaction propensities remain essentially constant within the interval $\tau_\mu^{\rm leap}$ (i.e., because of the sums over $\nu = 1 \ldots M$), which is why τ must be chosen as the smallest of $\{\tau_\nu^{\rm leap}\}$. Also note that the covariances involving the environmental variables [the fourth and fifth terms in Eq. (3.11)] do not fall out of Eq. (3.14) since the Poisson approximation has no bearing on these quantities.

This leads us to our next possible assumption: statistical independence among all random variables. This causes the rest of the covariance terms to drop out of Eq. (3.14), leaving

$$\mathrm{var}\{\Delta a_\mu(\tau_\mu^{\rm leap})\} \approx \tau_\mu^{\rm leap} \sum_{\nu=1}^{M} f_{\mu\nu}^2(t) a_\nu(t) + \left(\tau_\mu^{\rm leap}\right)^2 \sum_{q=1}^{Q} \left(\frac{\partial a_\mu(t)}{\partial \theta_q} \right)^2 \mathrm{var}\left\{ \dot{\theta}_q(t) \right\}. \tag{3.15}$$

If we then choose to ignore any fluctuations in the environmental quantities, we get

$$\left\langle \Delta a_\mu(\tau_\mu^{\rm leap}) \right\rangle \approx \tau_\mu^{\rm leap} \left(\sum_{\nu=1}^{M} f_{\mu\nu}(t) a_\nu(t) + \sum_{q=1}^{Q} \frac{\partial a_\mu(t)}{\partial \theta_q} \overline{\dot{\theta}_q(t)} \right), \tag{3.16}$$

$$\mathrm{var}\{\Delta a_\mu(\tau_\mu^{\rm leap})\} \approx \tau_\mu^{\rm leap} \sum_{\nu=1}^{M} f_{\mu\nu}^2(t) a_\nu(t), \tag{3.17}$$

where the 'overline' in $\overline{\dot{\theta}_q(t)}$ indicates that $\dot{\theta}_q(t)$ is no longer a random variable but a sure value, the evaluation of $d\theta_q(t)/dt$ at time t. Finally, if we assume that the propensities do not depend on the environmental quantities $\boldsymbol{\Theta}(t)$, i.e., the τ-independent case, then we can reduce Eq. (3.16) even further, giving

$$\left\langle \Delta a_\mu(\tau_\mu^{\text{leap}}) \right\rangle \approx \tau_\mu^{\text{leap}} \sum_{\nu=1}^{M} f_{\mu\nu}(t) a_\nu(t). \tag{3.18}$$

Equation (3.18) alone [i.e., with sdev $\left\{\Delta a_\mu(\tau_\mu^{\text{leap}})\right\} = 0$ in Eq. (3.12)] constitutes the original τ-selection procedure proposed by Gillespie in Ref. [47]. Inclusion of Eq. (3.17) along with (3.18) into Eq. (3.12) constitutes the improved τ-selection procedure of Gillespie and Petzold [50] (as well as the modified versions presented in Refs. [20] and [56]—see below). As of this writing, τ-selection procedures accounting for the effects of the environmental quantities have yet to be proposed.

Before writing out in detail the original and improved τ-selection formulas of Refs. [47] and [20, 50, 56], however, we deviate momentarily to address an issue that we have forestalled up until this point, the scaling factor ξ in Eq. (3.4).

The scaling factor ξ**.** Equation (3.4) has been written in terms of the abstract scaling factor ξ because the proper scaling for the leap condition was a matter of debate for some time. In the original τ-leaping work [47], Gillespie proposed that $\xi \equiv a_0(t)$, the *sum* of all of the reaction propensities at time t. Though not stated explicitly, it appears that the reason for this choice was two-fold: (i) it draws an analogy with the DM variant of the SSA [cf., Eq. (2.15)], and (ii) there is no chance that $a_0(t) = 0$ (for the simulation would terminate in this case) and, hence, no chance that Eq. (3.4) diverges to infinity. The issue was briefly revisited in subsequent work [50] but largely brushed aside as it was considered of secondary importance to the improved τ-selection procedure that was proposed

in that article. In Ref. [20], however, the authors revisited the issue once again and modified the approach by making the more intuitive choice of $\xi \equiv a_\mu(t)$ and devising a strategy for overcoming the problem that arises when $a_\mu(t) \to 0$ (more on this below). This approach was subsequently improved upon by Harris and Clancy [56].[12] As such, we rewrite here the leap condition with $\xi \equiv a_\mu(t)$, which is now the consensus choice for the scaling factor,

$$\frac{\left| \Delta a_\mu(\tau_\mu^{\text{leap}}) \right|}{a_\mu(t)} = \epsilon, \;\; (0 < \epsilon \ll 1). \tag{3.19}$$

Note that this choice has no bearing on the theoretical foundation for RB τ selection that has been presented above [Eqs. (3.6)–(3.18)].

Now, the source of the problem that arises when $a_\mu(t) \to 0$ is, very simply, that as the species populations become small even a single reaction firing can lead to a violation of Eq. (3.19). For example, if we consider the first-order decay reaction $S_i \to \emptyset$ with $X_i = 10$, then a single firing will change the propensity by 10%. Since we generally set $\epsilon = 0.01$–0.05, we see that this would violate the leap condition. However, there is nothing in what we have presented so far ([Eqs. (3.6)–(3.18)]) that would account for this fact. Thus, a naive RB τ-selection procedure utilizing the theoretical framework above will produce a value of τ_μ^{leap} that corresponds to an unphysical change in the reaction propensity. Considering our simple example again, if we set $\epsilon = 0.01$, then the τ_μ^{leap} generated will correspond to the time necessary for the reaction to fire 1/10th of a time, i.e., $\tau_\mu^{\text{leap}} = \left\langle \tau_\mu^{\text{ES}} \right\rangle / 10$. Clearly,

[12] Actually, our approach was developed independently from Cao et al. [20] and published in an early version of Ref. [56] on the arXiv preprint server (http://arxiv.org/abs/physics/0601217v1). Reference [20] appeared while our article was under review, however, forcing us to modify our article and acknowledge Cao et al. as the first to propose the idea of using $\xi \equiv a_\mu(t)$. Nevertheless, as will be shown subsequently, our approach is somewhat different from that in Ref. [20] and, in certain situations, can significantly improve the efficiency of the algorithm.

this will adversely affect the efficiency of the algorithm because the time steps will be much smaller than they need be.[13]

Thus, the modifications proposed in Refs. [20] and [56] for overcoming this problem involve including machinery that assures that the *minimum possible* propensity change is respected. This is done by again recasting the leap condition as

$$\left| \Delta a_\mu(\tau_\mu^{\text{leap}}) \right| = \max\left\{\epsilon a_\mu(t), \beta_\mu(t)\right\} \equiv \epsilon_\mu(t), \quad (0 < \epsilon \ll 1), \tag{3.20}$$

where $\beta_\mu(t)$ is the minimum possible change in a_μ at time t and is calculated for each reaction in the system at each simulation step [56]. The idea here is that if the minimum change possible is larger than what we have defined as "essentially constant" (basically ϵ) then we need to change our definition of essentially constant *for reaction* R_μ.

But what is the proper expression for β_μ? In determining this, it is important to recognize that because of the interconnectivity of reaction networks, it is possible that a reaction's propensity can be affected more so by the firing of *other* reactions in the system than by its own firings. For simple first-order reactions following mass-action kinetics (see Appendix B) this is of little consequence since incrementing and decrementing the reactant population has an equal effect on the propensity change (in an absolute sense, of course). It becomes important, however, when considering higher-order and non-mass-action type reactions, where incrementing and decrementing the reactant population(s) can have unequal effects, or reactions can fire that change one of the reactant populations but not the

[13] Of course, in this simple example this will be of no consequence since the reaction will be classified as ES and a value of τ_μ^{ES} generated. However, for interconnected reaction networks this *can* be a significant problem, as other reactions in the network can be limited by the small time step. Remember that in the PLA (Algorithm 3) the time step is not allowed to increase unless all reactions are classified as ES.

other(s). In principle, we must take into account all of these different possibilities when calculating $\beta_\mu(t)$.

The approach taken in Ref. [56] is to simplify the situation by ignoring all correlations and only considering how the propensity is affected by changes in each of the individual reactant populations, i.e., we set

$$\beta_\mu(t) = \begin{cases} \min\limits_{j=1\ldots N} \left\{ \left\| \dfrac{\partial a_\mu(t)}{\partial X_j} \right\| \right\} & \textit{if} \quad \min\limits_{j=1\ldots N} \left\{ \left\| \dfrac{\partial a_\mu(t)}{\partial X_j} \right\| \right\} < \infty \\ a_\mu^{\min} & \textit{otherwise} \end{cases} , \tag{3.21}$$

where

$$\left\| \frac{\partial a_\mu(t)}{\partial X_i} \right\| \equiv \begin{cases} \partial a_\mu(t)/\partial X_i & \textit{if} \quad \partial a_\mu(t)/\partial X_i > 0 \\ \infty & \textit{otherwise} \end{cases} . \tag{3.22}$$

In words, all that we are saying here is that $\beta_\mu(t)$ is the smallest *non-zero* element of $\{\partial a_\mu(t)/\partial X_j\}$. If all of the elements of $\{\partial a_\mu(t)/\partial X_j\}$ equal zero, however, then $\beta_\mu(t)$ should be set to the smallest possible value of a_μ, i.e., that with all of the reactant populations set to unity.

Two issues deserve discussion here. First, the related approach proposed by Cao et al. [20] basically amounts to defining $\beta_\mu(t) = \beta_\mu \equiv a_\mu^{\min}$, our lower limit on $\beta_\mu(t)$ in Eq. (3.21).[14] Although this choice eliminates the time dependence of β_μ, and hence means that it need not be calculated at every simulation step (as ours does), a major drawback is that $a_\mu^{\min}$ will often be an unnecessarily restrictive choice that will lead to small times steps and significantly diminished computational performance. In many cases, we believe that this cost will outweigh the benefit. Nevertheless, continued research into this issue is warranted. Second, the reason for choosing $\beta_\mu(t)$ as the *minimum* (non-zero) element of $\{\partial a_\mu(t)/\partial X_j\}$, as opposed

[14] Since Cao et al. [20] only consider elementary reaction types, they actually define $\beta_\mu(t) \equiv c_\mu$, the rate, or propensity, constant. $a_\mu^{\min}$ is a generalization of this, which is valid for non-elementary reaction types.

to, say, the maximum, is that this conservative choice assures us that we do not violate the leap condition. Consider a simple second-order reaction $S_i + S_j \to \emptyset$ with $X_i = 10$ and $X_j = 1000$. With $a_\mu = c_\mu X_i X_j$, we see that changing $X_i \pm 1$ changes a_μ by 10% while changing $X_j \pm 1$ results in a 0.1% change. We must choose $\beta_\mu(t) = \partial a_\mu(t)/\partial X_j$, therefore, because to not do so would allow for the possibility that a_μ changes by 10% due to changes in X_j. In other words, we would allow X_j to change by more than we should (and, hence, reactions involving X_j to fire more than they should) because there is no simple way to distinguish between changes in a_μ arising from changes in X_i *vs.* changes in X_j.

Two simple τ-selection procedures. With the addition of Eqs. (3.20)–(3.22), we are now in position to write out the two simple RB τ-selection procedures that have been proposed in the literature [20, 47, 50, 56]. The original τ-selection procedure of Gillespie [47] assumes $\Delta a_\mu(\tau_\mu^{\rm leap}) = \left\langle \Delta a_\mu(\tau_\mu^{\rm leap}) \right\rangle$ and uses Eq. (3.18) in conjunction with Eq. (3.20) to give

$$\tau_\mu^{\rm leap} = \frac{\epsilon_\mu(t)}{|\, m_\mu(t)\,|}, \tag{3.23}$$

where

$$m_\mu(t) \equiv \sum_{\nu=1}^{M} f_{\mu\nu}(t) a_\nu(t). \tag{3.24}$$

The second approach uses Eq. (3.17) in addition to (3.18) to generate an expression for $\tau_\mu^{\rm leap}$ via Eq. (3.12). There are a number of complications that we must overcome to do this, however. The first is that, in principle, we should account for both signs of the '$\pm$' in Eq. (3.12). This is difficult to do. We can simplify things, however, by making the conservative choice [50]

$$\left|\Delta a_\mu(\tau_\mu^{\rm leap})\right| \lesssim \left|\left\langle \Delta a_\mu(\tau_\mu^{\rm leap})\right\rangle\right| + {\rm sdev}\left\{\Delta a_\mu(\tau_\mu^{\rm leap})\right\}. \tag{3.25}$$

We could then try substituting this into Eq. (3.20) and deriving an expression for τ_μ^{leap} using Eqs. (3.17) and (3.18). We can see from Eqs. (3.17) and (3.18), however, that the result would be a quadratic equation; a factor of $\left(\tau_\mu^{\text{leap}}\right)^{1/2}$ would come from Eq. (3.17) while Eq. (3.18) would contribute a factor of τ_μ^{leap}. To circumvent this complication, Gillespie and Petzold [50] proposed an additional simplification, placing constraints on each term in Eq. (3.25) separately. Basically, if we constrain

$$\left|\left\langle \Delta a_\mu(\tau_\mu^{\text{leap}})\right\rangle\right| = \text{sdev}\left\{\Delta a_\mu(\tau_\mu^{\text{leap}})\right\} = \epsilon_\mu(t)/2 \tag{3.26}$$

then we will assure that $\left|\Delta a_\mu(\tau_\mu^{\text{leap}})\right| \leq \epsilon_\mu(t)$.[15] Doing this gives us two expressions for τ_μ^{leap}, from which we can choose the smaller. Thus, the improved τ-selection procedure of Gillespie and Petzold [50], as modified by Cao et al. [20] [$\xi \equiv a_\mu(t)$] and later by Harris and Clancy [56] [introduction of $\beta_\mu(t)$], can be written as

$$\tau_\mu^{\text{leap}} = \min\left\{\frac{\epsilon_\mu(t)/2}{|\,m_\mu(t)\,|}, \frac{\epsilon_\mu^2(t)/4}{\sigma_\mu^2(t)}\right\}, \tag{3.27}$$

where

$$\sigma_\mu^2(t) \equiv \sum_{\nu=1}^{M} f_{\mu\nu}^2(t) a_\nu(t). \tag{3.28}$$

As mentioned above, no τ-selection procedures have yet been proposed that account for the effects of the environmental variables $\boldsymbol{\Theta}(t)$, either as random variables

[15] In Ref. [50], Gillespie and Petzold actually propose constraining each term in Eq. (3.25) by ϵ, not $\epsilon/2$ [since they were still using $\xi \equiv a_0(t)$, there was no concept of $\epsilon_\mu(t)$ yet]. Though not explained explicitly, the reason for this seems to be because in that work the parameter ϵ lacked a physical meaning, i.e., because $\xi \equiv a_0(t)$. Without a true physical meaning, factors like $1/2$ are basically irrelevant since the parameter is tunable anyway. In fact, we believe that this is why Gillespie refers to ϵ as the "error control parameter," a purposely vague and abstract terminology, in his early works on τ-leaping [47, 50]. With $\xi \equiv a_\mu(t)$, however, ϵ takes on the physical meaning as the relative change in the propensity a_μ. This means that we should be careful when considering ϵ *vs.* fractional counterparts like $\epsilon/2$, since not doing so can introduce unintended additional sources of error.

or as simple time-varying functions. In principle, this should be straightforward to do. In practice, however, it is likely to be more difficult, perhaps requiring numerical techniques in many cases. In this dissertation, we will not proceed in this direction any further. Suffice it to say, however, that this is an area of continued interest and that the framework that we have laid out in this subsection should prove invaluable in our future endeavors towards this end.

3.4.2 Species-based τ-selection[16]

In Ref. [20], in addition to proposing the modified RB τ-selection procedure discussed in the preceding subsection, Cao et al. also proposed a different approach, a τ-selection scheme that is based on constraining the relative changes in each *species population*. The underlying idea of SB τ selection is very simple: If the propensities depend on time *only* via their dependence on the species populations (this is an important distinction from RB τ selection, which can, in principle, be applied in all cases), then it should be possible to impose a constraint on the changes in each species population such that the leap condition (3.20) will be satisfied for all reactions. The trick is identifying, for each species, what constraint will simultaneously satisfy the leap condition for all of the reactions in which the species participates in. In general, these can be of different orders and types.

In developing the approach, we begin by imposing the following constraint [20],

$$\left| \Delta X_i(T_i^{\text{leap}}) \right| = \max\left\{\epsilon X_i(t)/g_i(t), 1\right\} \equiv e_i(t), \quad (0 < g_i(t) < \infty), \tag{3.29}$$

which is written analogously to the leap condition Eq. (3.20): T_i^{leap} is the quantity

[16]This subsection contains material that has been presented, in highly condensed form, in Ref. [56].

that we are looking for, the time interval over which X_i is expected to remain essentially constant; $g_i(t)$ is a scaling factor that, in general, is a function of the species population $X_i(t)$ (see below) and will assure that Eq. (3.20) is satisfied for each reaction in which S_i appears as a reactant; and the minimum possible change in X_i is unity. The idea then is to calculate a value of $T_i^{\rm leap}$ for each species in the system and set $\tau = \min\{T_j^{\rm leap}\}$.

To do this, we follow a very similar procedure to that of RB τ selection: We (i) obtain a general expression for $\left| \Delta X_i(T_i^{\rm leap}) \right|$ [actually, we already have an exact expression in Eq. (3.7)], (ii) assume that all reactions in the system obey Poissonian statistics [Eq. (3.1)], and (iii) obtain expressions for the mean and variance of $\left| \Delta X_i(T_i^{\rm leap}) \right|$ using the linear combination theorem for random variables [Eqs. (A.15) and (A.16)]. These expressions can then be substituted into an equation analogous to Eq. (3.12) to derive an expression for $T_i^{\rm leap}$. The procedure is greatly simplified as compared to RB τ selection, however, by the fact that SB τ selection assumes, *by definition*, that the propensities do not depend on time through the environmental quantities $\mathbf{\Theta}(t)$. Thus, in the general case, we have

$$T_i^{\rm leap} = \min\left\{ \frac{e_i(t)/2}{|\,\widetilde{m}_i(t)\,|}, \frac{e_i^2(t)/4}{\widetilde{\sigma}_i^2(t)} \right\}, \tag{3.30}$$

where

$$\widetilde{m}_i(t) \equiv \sum_{\nu=1}^{M} z_{\nu i} a_\nu(t), \tag{3.31}$$

$$\widetilde{\sigma}_i^2(t) \equiv \sum_{\nu=1}^{M} z_{\nu i}^2 a_\nu(t). \tag{3.32}$$

These are clearly analogous to the improved RB τ-selection equations of the preceding subsection [Eqs. (3.27), (3.24) and (3.28), respectively]. Furthermore, if one wishes to ignore the fluctuations in $\Delta X_i(T_i^{\rm leap})$, a simplified τ-selection procedure is to, as in Eq. (3.23), use just the first term in the brackets on the right-hand side

of Eq. (3.30), less the factor of 1/2, i.e.,

$$T_i^{\text{leap}} = \frac{e_i(t)}{|\,\widetilde{m}_i(t)\,|}. \tag{3.33}$$

The main advantage to using Eqs. (3.30) or (3.33) in lieu of their RB counterparts Eqs. (3.27) and (3.23), is that, in general, the SB expressions require fewer computational operations. In particular, we need not calculate any derivatives in SB τ selection [compare Eqs. (3.24) and (3.28) with Eqs. (3.31) and (3.32)]. Thus, each T_i^{leap} calculation is generally faster than each τ_μ^{leap}. Moreover, in many cases there are far fewer species than reactions, meaning that fewer total calculations must be carried out to determine τ. The one drawback, however, is that the constraint that we impose in Eq. (3.29) often turns out to be more restrictive than that in Eq. (3.20) (see below). This means that SB τ-selection time steps are often smaller than those calculated via the RB approach. However, the advantages of SB τ-selection discussed above often outweigh these costs, though not always (see examples in Chapter 4). In any event, SB τ-selection represents an important milestone in the continued quest for improving the efficiency, and hence the practicality, of the leaping approach.

The only question that remains, therefore, is how to determine the scaling factor $g_i(t)$ in Eq. (3.29). In general, it is useful to think of each reaction in the system as having associated with it a different value of g_i for each species in the system. Basically, changes in X_i might affect the propensity a_μ of reaction R_μ differently than $a_{\mu'}$ of $R_{\mu'}$. Thus, it may be necessary, depending on the details of the reactions, to more or less strongly constrain $\Delta X_i(T_i^{\text{leap}})$ in R_μ relative to $R_{\mu'}$. As such, we can think in terms of a matrix $\mathbf{G}(t)$ with elements $\{g_{\nu j}(t)\}_{\nu=1\ldots M}^{j=1\ldots N}$, where, for a given species i, each $g_{\nu i}(t)$ can be different. For species that are not present in the rate expression for $a_\mu(t)$ (for elementary reactions, this just means

that they are not reactant species), we have $g_{\mu i}(t)=0$, i.e., we place no constraint on their changes [$e_i(t)=\infty$ in Eq. (3.29)]. For species that are present in the rate expression, $g_{\mu i}(t)$ depends strongly on the details of the reaction (see below). In Eq. (3.29), we see that $\Delta X_i(T_i^{\text{leap}})$ is inversely proportional to $g_i(t)$, meaning that the larger the value of $g_i(t)$ the tighter the constraint on $\Delta X_i(T_i^{\text{leap}})$. In order to assure that all of the reactions in which S_i participates in obey the leap condition (3.20), it is necessary, therefore, to choose $g_i(t)$ as

$$g_i(t) = \max_{\nu=1\ldots M}\{g_{\nu i}(t)\}. \tag{3.34}$$

This requirement is why the SB τ selection approach often produces smaller time steps than the RB approach. Equation (3.34) implies that we will often over-constrain $|\Delta a_\mu|/a_\mu$ for certain reactions in order to assure that the leap condition is not violated for other reactions. Nevertheless, as discussed above, the advantages, in terms of computational speed, of the SB approach often outweigh this cost.

So, how do we calculate the individual elements of $\mathbf{G}(t)$? The basic strategy is to derive an expression of the following form for $\Delta a_\mu/a_\mu$ [20]:

$$\frac{\Delta a_\mu}{a_\mu} \approx \sum_{j=1}^{N} \alpha_{\mu j}(X_j)\frac{\Delta X_j}{X_j}, \quad [\alpha_j(X_j) \geq 0], \tag{3.35}$$

where the $\alpha_{\mu j}(X_j)$ are non-negative coefficients that can, in general, be functions of X_j.[17] It is important to recognize that in writing this expression, we are neglecting any correlations between the ΔX_j's [20] [just as we did in Eq. (3.21)]. Then, in an approach reminiscent of what we did in going from Eq. (3.12) to (3.25) to (3.26), we say that

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \sum_{j=1}^{N} \alpha_{\mu j}(X_j)\frac{|\Delta X_j|}{X_j} = \epsilon_\mu, \tag{3.36}$$

[17] Since only a few species are involved in any given reaction, most of the values of $\alpha_{\mu j}(X_j)$ are zero.

and place constraints on each individual term such that the relationship is satisfied, i.e.,

$$\alpha_{\mu i}(X_i)\frac{|\Delta X_i|}{X_i} = \epsilon_\mu/\zeta_{\mu i}, \;\; (\zeta_{\mu i} \geq 1), \tag{3.37}$$

where the values of the scaling factors are constrained by the condition

$$\sum_{j=1}^{N} \frac{1}{\zeta_{\mu j}} = 1. \tag{3.38}$$

For species that do not act as reactants in R_μ (i.e., are not present in the rate expression), we simply choose $\zeta_{\mu i} = \infty$. This causes the right-hand side of Eq. (3.37) to equal zero, which is consistent with the left-hand side since $\alpha_{\mu i}(X_i) = 0$ for these species (see footnote 17). For reactant species, we often choose $\zeta_{\mu i}$ equal to the number of non-zero terms in Eq. (3.35). For example, this is what we did in going from Eq. (3.25) to Eq. (3.26) above. Since there are two terms in Eq. (3.25), we chose to constrain each term in Eq. (3.26) by $\epsilon_\mu(t)/2$, i.e., we chose $\zeta_{\mu i} = 2$. In some cases, however, it might be preferable to weight each term differently.[18]

In any case, we see from Eq. (3.37) that the elements of $\mathbf{G}(t)$ are given by

$$g_{\mu i}(t) = \zeta_{\mu i}\alpha_{\mu i}\left(X_i(t)\right). \tag{3.39}$$

In Appendix D, we derive expressions for $g_{\mu i}(t)$ for various elementary and non-elementary reaction types. We also demonstrate how, in many cases, it is advantageous, in terms of computational efficiency, to eliminate the dependence on $X_i(t)$

[18]For example, for the third-order "elementary" reaction $S_i + 2S_j \rightarrow$ *products*, Cao et al. [20] doubled the weight on the ΔX_j term relative to the ΔX_i term. Basically, they constrained the ΔX_i term by $\epsilon/3$ and the ΔX_j term by $2\epsilon/3$ (see Appendix D). In our notation, they used $\zeta_{\mu i} = 3$ and $\zeta_{\mu j} = 3/2$. Intuitively, this makes sense given the stoichiometry of the reaction. If we want to automate this procedure, however, it is not clear how, in the general case, we can determine such differential weights, especially when considering non-elementary reaction types. This remains an open area of interest.

in Eq. (3.39) by considering the values of $g_{\mu i}$ in the limits $X_i \to 0$ and $X_i \to \infty$ and then choosing $g_{\mu i}$ as the larger of the two. Doing this means that the scaling factor g_i in Eq. (3.29) loses its time dependence and, hence, need only be calculated once during the initialization stage of a simulation.

3.4.3 Post-leap checking

In Ref. [47], while pondering possible strategies for selecting values of τ in τ-leaping, Gillespie alluded to the possibility of performing *post-leap checks* on the values of $\{|\Delta a_\nu(\tau)|\}$. The basic idea would be to pick a τ, update the system and then check to see whether any of the $\{|\Delta a_\nu(\tau)|\}$ violated the leap condition Eq. (3.20). If so, then a smaller value of τ would be chosen and the leap attempted again. Conversely, if all the values satisfied the leap condition by a wide margin then a larger value of τ could be selected. However, Gillespie concluded, rightly so, that this procedure would introduce bias into the system as rare, yet legitimate, large fluctuations would be rejected without cause. This led Gillespie on the path to developing the pre-leap approaches of Secs. 3.4.1 and 3.4.2.

Recently, however, Anderson [1] has revisited the idea of performing post-leap checks in τ-leaping and has provided a theoretically-justified procedure for doing so. The idea underlying his method is surprisingly simple: τ can be reduced to τ^* upon a post-leap check so long as the information gathered in τ is used in determining the numbers of firings in τ^*. In more technical language, the random deviates $\{k_\nu^*\}$ must be determined by conditioning upon the already-generated values of $\{k_\nu\}$. By doing this, no information is discarded and, hence, no bias is introduced.

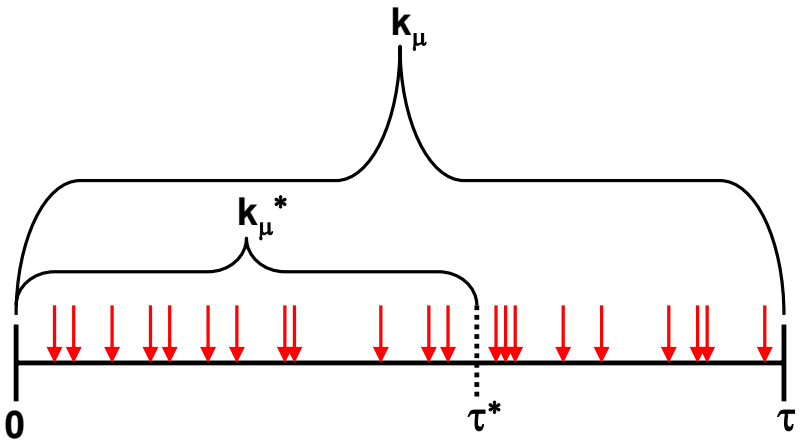

Figure 3.1: Illustration of the basic idea underlying the post-leap checking procedure of Anderson [1]. Red arrows indicate individual firing times of reaction R_μ. If the leap condition Eq. (3.20) is deemed to have been violated due to the k_μ firings of R_μ over the time interval τ, then a smaller interval $\tau^* < \tau$ is chosen and the number of firings k_μ^* over this interval is extracted with the correct statistics. k_μ^* happens to be a binomial random variable with success probability $p = \tau^*/\tau$ for the $n = k_\mu$ trials [see text and Appendix A, Eqs. (A.4)–(A.7)].

To make the idea clearer, a simple illustration is presented in Fig. 3.1. The red arrows represent individual firings of a given reaction R_μ over the time interval $[t, t+\tau)$ *under the assumption* that a_μ remains constant during the interval. Hence, these firings are not physical but, rather, are random samples that faithfully describe reality *only* if the assumption that a_μ remains essentially constant during the interval is upheld. We assume here that this assumption is *not* upheld over the interval $[t, t+\tau)$, i.e., the leap condition is violated. However, it is upheld over the shorter interval $[t, t+\tau^*)$. Therefore, we can legitimately reduce τ to τ^* so long as we extract, with the correct statistics, the numbers of firings over the shorter interval given those over the longer interval.

This is actually quite easy to do. Each firing in Fig. 3.1 either falls to the left or to the right of τ^*. Moreover, they are equally likely to land anywhere in the

interval $[0, \tau)$. Thus, the numbers falling to either side of τ^* are *binomial random variables* [see Appendix A, Eqs. (A.4)–(A.7)]. The probability of falling to the left of τ^*, which is what we are interested in, is simply τ^*/τ. Thus, given the total number of firings k_μ over $[t, t+\tau)$, the number of firings k_μ^* over $[t, t+\tau^*)$ is the binomial random variable $\mathcal{B}_\mu(k_\mu, \tau^*/\tau)$. This is easily proved mathematically [1]:

$$
\begin{aligned}
&\Pr\{k_\mu^*, \tau^* | k_\mu, \tau\} \qquad (3.40)\\
&\quad = \Pr\{k_\mu^*, \tau^* \cap k_\mu, \tau\} / \Pr\{k_\mu, \tau\} \\
&\quad = \Pr\{k_\mu, \tau | k_\mu^*, \tau^*\} \Pr\{k_\mu^*, \tau^*\} / \Pr\{k_\mu, \tau\} \\
&\quad = \Pr\{k_\mu - k_\mu^*, \tau - \tau^*\} \Pr\{k_\mu^*, \tau^*\} / \Pr\{k_\mu, \tau\} \\
&\quad = \left[\frac{(a_\mu(\tau-\tau^*))^{k_\mu - k_\mu^*} e^{-a_\mu(\tau-\tau^*)}}{(k_\mu - k_\mu^*)!}\right] \left[\frac{(a_\mu \tau^*)^{k_\mu^*} e^{-a_\mu \tau^*}}{k_\mu^*!}\right] \left[\frac{k_\mu!}{(a_\mu\tau)^{k_\mu} e^{-a_\mu \tau}}\right] \\
&\quad = \left(\frac{k_\mu!}{k_\mu^*!(k_\mu - k_\mu^*)!}\right) \frac{(\tau-\tau^*)^{k_\mu - k_\mu^*} \tau^{*k_\mu^*}}{\tau^{k_\mu}} \times \left(\frac{\tau^{k_\mu^*}}{\tau^{k_\mu^*}}\right) \\
&\quad = \binom{k_\mu}{k_\mu^*} \left(\frac{\tau^*}{\tau}\right)^{k_\mu^*} \left(1 - \frac{\tau^*}{\tau}\right)^{k_\mu - k_\mu^*} \\
&\quad = \mathcal{B}_\mu(k_\mu, \tau^*/\tau),
\end{aligned}
$$

where, for simplicity, we have used the shorthand $\Pr\{k_\mu^*, \tau^* | k_\mu, \tau\} \equiv \Pr\{\mathcal{P}_\mu(a_\mu\tau^*) = k_\mu^* | \mathcal{P}_\mu(a_\mu\tau) = k_\mu\}$. □

The post-leap checking procedure of Anderson [1] thus amounts to choosing a value of τ (either using the pre-leap formulas of Secs. 3.4.1 and 3.4.2 or otherwise), checking to see whether the leap condition has been violated, and if so choosing a smaller value of τ and determining the new numbers of firings by generating M binomial random deviates conditioned on the old random deviates. This procedure is repeated until a value of τ is identified for which the leap condition is satisfied for all reactions. This greatly improves the accuracy of the approach since the leap condition is assured to be satisfied at all times. The process of "backing up"

can be quite expensive, however, due to the high expense of generating binomial random numbers. This can be attenuated to some extent by clever choices of τ. Anderson [1] proposes a τ-selection strategy whereby if the leap condition is: (i) *violated* then the step is rejected and τ is reduced by some pre-specified amount, (ii) *barely satisfied* then the step is accepted but τ is reduced for the subsequent step, albeit by an amount less than for case (i), and (iii) *strongly satisfied* then the step is accepted and τ increased by some pre-specified amount. Step (ii) increases the probability that the subsequent leap will be accepted, thus reducing the computational burden associated with backing up. Of course, the trick lies in determining the parameters that optimize this procedure.

As of this writing, no articles have yet been published that use the post-leap checking approach of Anderson [1] and we have yet to implement it in the PLA. However, we have included it here because it is the opinion of the author of this dissertation that the approach is a particularly important innovation that will lead to much more efficient τ-selection procedures in the near future. τ selection is by far the most computationally intensive aspect of the leaping approach and advancements in this area are sure to have a tremendous impact on its practical utility. In Secs. 3.6.1 and 3.6.2, we discuss some preliminary ideas regarding τ-selection procedures that incorporate post-leap checking.

3.5 Relation to other leaping approaches

Obviously, the PLA, the central contribution of this dissertation, extends and builds upon the ideas laid out in Gillespie's seminal works on τ leaping [46, 47]. Since the publication of those works, numerous other exten-

sions have been published as well, both from within Gillespie's direct collaboration [18, 20, 21, 24, 50, 92, 93] and from outside contributors [1, 3, 12–14, 25, 67, 70, 83, 84, 91, 94, 109, 114, 116], such as ourselves [56].

It is important, therefore, to understand the motivations behind these extensions in order to ascertain where the PLA fits in and what its contribution is to the field. Therefore, we provide below a mini-review of various leaping methods that have been proposed in the literature to date. We divide these methods into five categories: (i) extensions per Gillespie, Petzold and co-workers, (ii) binomial τ-leaping methods, (iii) K-leaping methods, (iv) higher-order leaping methods, and (v) miscellaneous leaping approaches. We conclude with a discussion comparing and contrasting these methods with the PLA and elaborating on what we see as being the primary contribution of the PLA. Note that some of these methods [18, 24, 25, 92, 93, 109] have been discussed briefly in Sec. 3.2 above. Furthermore, reviews covering some, but not all, of the methods discussed below can be found in Refs. [49] and [81].

Extensions by Gillespie, Petzold and co-workers [18, 20, 21, 24, 50, 92, 93]. Two years after publication of the original τ-leaping article [47], two articles from Gillespie, Petzold and co-workers appeared proposing extensions to the approach [50, 92]. Gillespie and Petzold [50] presented an improved RB τ-selection formula [on which Eq. (3.27) is based] while Rathinam et al. [92] proposed an "implicit" formulation of τ leaping. These articles can be seen as the progenitors of two related yet separate series of works by this group, one focused on enhancements to explicit τ-leaping [18, 20] and the other on the theoretical foundations and alternative formulations of implicit τ leaping [24, 93]. Culmination of both series of works was realized in Ref. [21], where a hybrid explicit/implicit τ-leaping

approach was proposed.

In Ref. [18], Cao et al. proposed a strategy for avoiding the occurrence of negative populations during the course of a τ-leaping simulation. This article was in direct response to a shortcoming of the original τ-leaping method [47] identified, simultaneously and independently, by Tian and Burrage [109] and Chatterjee et al. [25] (see below). As discussed briefly in Sec. 3.2 above, the approach proposed in [18] involves partitioning reactions into "critical" and "non-critical" subsets. Reactions in the critical subset have small populations and, hence, are treated using the DM variant of the SSA. As such, they cannot fire more than once in a simulation step and cannot give rise to negative populations. The partitioning process, however, is somewhat *ad hoc*, although the authors do attempt to provide some justification for it by utilizing ideas proposed in Ref. [109].

Subsequently, in Ref. [20] Cao et al. proposed a new variation of RB τ selection [identical to Eq. (3.27) except with $\beta_\mu(t)$ replaced by c_μ—see Sec. 3.4.1] and the SB τ-selection procedure of Sec. 3.4.2 (which we subsequently improved upon [56]). Integrating the method proposed in Ref. [18] with these two new τ-selection approaches, Cao et al. [20] enumerated what can be considered to be the current formulation of explicit τ leaping.

The implicit τ-leaping approach proposed by Rathinam et al. [92] was motivated by the realization in [47] that the simplest formulation of τ leaping is analogous to the explicit forward Euler method for solving deterministic ODEs. Indeed, Gillespie recognized that using the values of the propensities at the initial time t, i.e., $\{a_\nu(t)\}$, to generate Poisson random deviates for the time interval $[t, t+\tau)$ will invariably introduce error into the method. Thus, in addition to the simple approach, Gillespie also proposed in [47] a higher-order explicit method

based on estimating the state of the system at the time $t+\tau/2$ and then using the estimated propensities at that time to generate the required Poisson random deviates. This approach, termed the "estimated midpoint" τ-leap method (see below for further discussion), is analogous to a second-order Runge-Kutta method used in the solution of ODEs [47].

The implicit τ-leaping methods proposed in Refs. [24, 92, 93] can be seen as attempts to extend and generalize this idea. In Ref. [92], Rathinam et al. proposed a τ-leaping approach based on the following assumption,

$$K_\mu(\tau|\mathbf{x_t},t) \approx \mathcal{P}_\mu(a_\mu(t)\tau) + \text{round}\left\{\left[a_\mu(t+\tau) - a_\mu(t)\right]\tau\right\}. \tag{3.41}$$

This equation can be seen as a modification to Eq. (3.1) that includes a correction to account for the change in the propensity over the time interval $[t, t+\tau)$. If the propensity increases over the interval then the number of firings is increased to reflect this, and *vice versa*. The rounding operation is needed because, in general, $[a_\mu(t+\tau) - a_\mu(t)]\,\tau$ is not integer-valued. Equation (3.41) is implicit in the sense that $K_\mu(\tau|\mathbf{x_t},t)$ is dependent on the *unknown* propensity $a_\mu(t+\tau)$ at the end of the time step. However, $a_\mu(t+\tau)$ can be written in terms of the set of quantities $\{K_\nu(\tau|\mathbf{x_t},t)\}$ [Eq. (3.8)]. Thus, there are M coupled algebraic equations that can be solved numerically using, e.g., Newton's method. This is the same approach used in implicit ODE solvers [92].

The motivation behind Eq. (3.41) is the fact that for $a_\mu(t)\tau \gg 1$, it reduces to

$$K_\mu(\tau|\mathbf{x_t},t) \approx a_\mu(t+\tau)\tau + \sqrt{a_\mu(t)}\mathcal{N}_\mu(0,1)\tau^{1/2}. \tag{3.42}$$

This is equivalent to the semi-implicit Euler method used in the solution of stochastic differential equations (SDEs) [92]. Furthermore, for $a_\mu(t)\tau \to \infty$, Eq. (3.42) reduces to

$$K_\mu(\tau|\mathbf{x_t},t) \approx a_\mu(t+\tau)\tau, \tag{3.43}$$

which is equivalent to the implicit, or backwards, Euler method used in the solution of ODEs [92]. Thus, just as in the case of Eqs. (3.1)–(3.3), we have in Eqs. (3.41)–(3.43) a hierarchy of descriptions that converges to an approach well known at the deterministic level.

The utility of the implicit τ-leaping method of Eq. (3.41) was demonstrated in Ref. [92] on two simple example systems. Both systems contained reactions with very large rate constants so as to render them "stiff." Rathinam et al. [92] demonstrated, using fixed time steps, that the implicit method can retain stability in situations where the explicit approach [Eq. (3.1)] cannot. The primary shortcoming of the method, however, is that it tends to dampen fluctuations. This is particularly concerning because fluctuations are of critical importance in stochastic simulations (i.e., they are not numerical artifacts). To overcome this problem, the authors proposed a rather *ad hoc* strategy, termed "downshifting," that involves interlacing intermittent bursts of short time steps in with the long time steps of the implicit simulation. The short time steps are simulated using the SSA or the explicit τ-leaping procedure. By doing this, one hopes to obtain sufficient statistics on the fluctuations while retaining the efficiency afforded by the implicit approach.

In Ref. [24], Cao et al. investigated the numerical stability properties of the explicit τ-leaping approach of Eq. (3.1), the implicit approach of Eq. (3.41), and an additional method that they termed "trapezoidal" τ-leaping. The trapezoidal method is also implicit and is based on the following assumption,

$$K_\mu(\tau|\mathbf{x_t},t) \approx \mathcal{P}_\mu(a_\mu(t)\tau) + \text{round}\left\{\frac{1}{2}\left[a_\mu(t+\tau) - a_\mu(t)\right]\tau\right\}. \tag{3.44}$$

For $a_\mu(t)\tau \gg 1$, this reduces to

$$K_\mu(\tau|\mathbf{x_t},t) \approx \frac{1}{2}\left[a_\mu(t+\tau) + a_\mu(t)\right]\tau + \sqrt{a_\mu(t)}\mathcal{N}_\mu(0,1)\tau^{1/2}, \tag{3.45}$$

and for $a_\mu(t)\tau \to \infty$, we have

$$K_\mu(\tau|\mathbf{x_t}, t) \approx \frac{1}{2}\left[a_\mu(t+\tau) + a_\mu(t)\right]\tau, \tag{3.46}$$

which is equivalent to the well-known trapezoidal method for solving ODEs. Cao et al. [24] considered how each of the three methods behave when applied to the simple test system $S_1 \rightleftharpoons S_2$ as $t \to \infty$ for a fixed time step τ, a property known as "absolute stability." In particular, they were interested in assessing (i) whether the methods converge to a stable solution and (ii) whether that solution corresponds to the theoretical solution [48]. Because the methods are stochastic they considered not only the mean but the variance and all higher-order moments as well.

In all three cases, the authors derived the conditions required for the leaping methods to retain stability. These amounted to constraints on the allowable sizes of τ for each method. They showed that for the implicit methods (backwards and trapezoidal) stability is retained for arbitrarily large values of τ, as is the case for implicit ODE methods. Conversely, there is a well-defined upper limit on τ in the explicit method. The authors also showed that, for values of τ within the region of stability, the mean values obtained from all three methods converge to the theoretical mean. In the case of the variance, however, they showed that the explicit method provides an over approximation while the implicit (backwards) method under approximates it. The latter result explains the damping effect seen in Ref. [92]. Interestingly, Cao et al. [24] also showed that the variance obtained from the trapezoidal method *does* converge to the theoretical value. This begged the question whether all higher-order moments of the trapezoidal method converge to the theoretical values. However, this was shown not to be the case.

In a complementary work, Rathinam et al. [93] performed a more extensive

stability analysis of the explicit and backwards τ-leaping methods [Eqs. (3.1) and (3.41)]. They confirmed the absolutely stability results of Cao et al. [24] and showed that both methods also retain stability as $\tau \to 0$, a property known as "0-stability." Furthermore, they proved that both methods are accurate in their means and variances up to terms of $\mathcal{O}(\tau)$ (first-order consistency) and provided formulas for the $\mathcal{O}(\tau^2)$ components of the errors (the dominant contributions). These results were subsequently verified via application to three simple example systems. Finally, in the special case of purely linear propensities (systems of first-order elementary reactions) the authors showed that *all* moments of both methods are first-order consistent.

In Ref. [21], Cao et al. proposed an approach for integrating the explicit and implicit τ-leaping methods in an adaptive way. It can be seen as the culmination of the two series of works comprised of Refs. [18, 20, 50] and Refs. [24, 92, 93]. The method relies on the presence of small reaction subnetworks (e.g., reversible reaction pairs) that can be deemed to be in rapid equilibrium. If this is the case, the authors argue that the equilibrated reactions can be ignored in the τ-selection process because the slow reactions see only the mean effects of the fast. They propose a simple criterion for determining whether specified sets (pairs) of reactions are in rapid equilibrium. If so, they calculate two time steps, one including all of the reactions in the system, τ^{explicit}, and one excluding the equilibrated reactions, τ^{implicit}. If $\tau^{\text{implicit}} \gg \tau^{\text{explicit}}$ (e.g., 100-fold) then they set $\tau = \tau^{\text{implicit}}$ and determine the set of reaction firings $\{k_\nu(\tau)\}$ using either the backwards or trapezoidal implicit method. Importantly, this includes the equilibrated reactions as well, i.e., they are not reduced out of the system they are only ignored in the τ-selection procedure.

It is for this reason that Cao et al. [21] argue that an implicit method is re-

quired. Since the equilibrated reactions do not satisfy the leap condition Eq. (3.20) in $[t, t+\tau^{\text{implicit}})$ the implicit method is necessary to maintain stability. Consequently, however, the fluctuations in the fast reactions are damped and they again propose using downshifting to rectify the problem. Finally, if $\tau^{\text{implicit}} \not\gg \tau^{\text{explicit}}$, then $\tau = \tau^{\text{explicit}}$ and the explicit method is used because it is less computationally expensive than the implicit method. In this way the algorithm automatically switches between explicit and implicit τ leaping based on the state of the system. When applied to two simple example systems containing fast-reversible reaction pairs the approach was shown to significantly reduce computational time [21]. Note that one shortcoming of the method is that it is unable to handle fast reactions involving small-population species (e.g., genes) because these are always classified as critical and, hence, are not excluded from the τ-selection process [21].

Binomial τ leaping [25, 67, 83, 84, 109]. As mentioned above, a shortcoming of Gillespie's original τ-leaping algorithm [47], which was recognized relatively early on, is its potential to generate negative population levels. The reason for this is that the Poisson random variable is positively unbounded. As such, there is a 100% chance, given a long enough simulation run, that an outlier random deviate will be generated that will push a reactant population negative [25]. In practice, the problem seems to arise most commonly in cases where a species with a small population (e.g., a gene) interacts with a large-population species (e.g., transcription factors) [25, 57, 109]. The small population limits the number of times that the reaction can fire but the large population results in a large propensity that can lead to an unphysical Poisson random deviate.

This flaw was first pointed out, simultaneously and independently, by Tian and Burrage [109] and Chatterjee et al. [25] who both advocated using binomial

random variables rather than Poisson for approximating $K_\mu(\tau|\mathbf{x_t}, t)$. Binomial random variables are characterized by two quantities: the number of trials, n, and the success probability, p (see Appendix A). The former places an upper limit on the number of times a reaction can fire during τ and an appropriate choice should, therefore, eliminate the possibility of producing negative populations.

The basic idea behind the methods proposed by both Tian and Burrage [109] and Chatterjee et al. [25] is the same: for each reaction R_μ choose a value of $n_\mu(t)$ (the maximum number of allowed firings at time t) and approximate

$$K_\mu(\tau|\mathbf{x_t}, t) \approx \mathcal{B}_\mu\Big(n_\mu(t), p_\mu(t) \equiv a_\mu(t)\tau/n_\mu(t)\Big)\,, \tag{3.47}$$

a binomial random variable with mean $a_\mu(t)\tau$ [Eq. (A.6)] and variance $a_\mu(t)\tau\,[1-a_\mu(t)\tau/n_\mu(t)]$ [Eq. (A.7)]. We see that the mean is equivalent to that of the Poisson random variable $\mathcal{P}_\mu(a_\mu(t)\tau)$ while the variance is also equivalent if $a_\mu(t)\tau/n_\mu(t) \to 0$. The primary challenge, therefore, to implementing a successful binomial τ-leaping algorithm is in choosing appropriate values of $n_\mu(t)$ that both eliminate the possibility of producing negative populations and retain the correct statistics of the dynamical process.

In the case of a single first-order elementary reaction $S_i \to \textit{products}$, the choice is simple: $n_\mu(t) = X_i(t)$. Equally simple relations exist for other elementary reaction types [25, 109]. Complications arise, however, when considering non-elementary reaction types and networks of reactions where species appear as reactants and/or products in multiple reactions. Tian and Burrage [109] and Chatterjee et al. [25] proposed different strategies for handling the latter situation. Both are somewhat limited in their applicability and neither is based on strong theoretical grounds. We refer the reader to the original articles for the specifics.

Subsequent work by Peng et al. [83] and Leier et al. [67] aimed at refining and

generalizing the procedure for choosing $n_\mu(t)$. A good overview of both methods and their relation to the method of Tian and Burrage [109] is given in Ref. [67]. Briefly, Peng et al. [83] recognized that the approach proposed in Ref. [109] often gives overly conservative values of $n_\mu(t)$, reducing the efficiency of the method. Moreover, it is unable to handle complex couplings between reactions. They proposed an alternative approach that is significantly more computationally expensive but can handle generic reaction networks and allows for the use of larger time steps. Leier et al. [67] proposed a somewhat simpler approach for handling generic reaction couplings. They also discussed the problem of non-elementary reaction types [Hill reactions in particular; see Appendix D, Eq. (D.24)] and extended the method to handle time-delayed reactions (as are often used for modeling, e.g., transcription and translation).

In all cases [25, 67, 83, 109], the authors demonstrated through simple examples that the binomial τ-leaping methods are comparable in accuracy to the Poisson-based methods. However, there is some variability in the results (i.e., sometimes binomial τ leaping is more accurate and sometimes it is less) that is likely due to the choices of example systems and parameters. Less attention was given to efficiency although the general theme was that larger time steps could be employed which obviously reduced computational effort.

Finally, Pettigrew and Resat [84] proposed an extension to the binomial τ-leaping approach that they termed multinomial τ leaping. The primary modification that they made is to group reactions and sample the *total* number of firings of the group from a binomial distribution, rather than those for individual reactions. The firings of individual reactions are then allocated from the total by sampling from a multinomial distribution, which is simply a generalization of the binomial

distribution with more than two possible outcomes. The approach can be seen as a generalization of that proposed by Tian and Burrage [109] for handling the special case where a species appears as a reactant in exactly two reactions.

Pettigrew and Resat [84] assemble groups by identifying sets of reactions where no reactant species appear as reactants in any other group, a condition known as "reactant closure." In the simplest case, one can consider the entire network of reactions as a single group. An upper bound, $n_G(t)$, on the number of firings within each group G is then chosen and the actual number of firings is sampled from the binomial distribution $\mathcal{B}_G(n_G(t), a_G(t)\tau/n_G(t))$ where $a_G(t) \equiv \sum_{\nu \in G} a_\nu(t)$. The authors provide two procedures for selecting $n_G(t)$: a conservative approach that ensures that no populations become negative and a more aggressive trial-and-error strategy that often gives larger values of $n_G(t)$ in practice and converges to the conservative approach in the worst-case scenario. The authors also offer an alternative τ-selection strategy that involves choosing a value of τ (some multiple of the previous τ), estimating the state of the system at $t+\tau$ by assuming that each reaction fires $a_\mu(t)\tau$ times (the mean of the associated Poisson random variable), and checking to see whether the leap condition has been violated for any reaction. If so, τ is reduced and the procedure repeated until an acceptable value is found. This approach is similar in spirit to, but lacks the rigorous theoretical foundation of, Anderson's post-leap checking procedure (Sec. 3.4.3).

K leaping [3, 14]. In Gillespie's original τ-leaping article [47], yet another simulation approach was offered dubbed the k_α-leap method. The basic idea of this approach is that instead of calculating τ and generating random samples for the firings of all reactions, one calculates the number of times, k_α, that a given reaction R_α can fire before violating the leap condition Eq. (3.20). The time step τ is then

drawn from a Gamma distribution (essentially the inverse of the Poisson distribution; see Appendix in Ref. [47]) with parameters $a_\alpha(t)$ and k_α and the numbers of firings of all other $M-1$ reactions during τ are drawn from Poisson distributions as in standard τ leaping. The procedure proposed for selecting k_α is very similar to that proposed for selecting τ [47].

Although the practical utility of k_α leaping is minimal (it offers little advantage over τ leaping) the basic idea behind the approach motivated two subsequent works by Auger et al. [3] and Cai and Xu [14]. These authors, simultaneously and independently, proposed leaping methods based on calculating the *total* number of reaction firings within the next simulation step (in a manner analogous to the τ-selection procedures of Sec. 3.4), drawing τ from a Gamma distribution and then allocating the firings amongst the individual reactions according to a multinomial distribution. The latter procedure is essentially the same approach taken by Pettigrew and Resat [84]. The focus on the total number of firings, rather than that for an individual reaction, is what differentiates these methods from the k_α-leap method of Ref. [47]. For the sake of simplicity and to retain consistency with the terminology used in this dissertation, we refer to these methods as K-leaping approaches, with $K \equiv \sum_{\nu=1}^{M} k_\nu(\tau)$.

In essence, the K-leaping approach is the inverse to the τ-leaping approach. It is based on the same fundamental concepts as τ leaping (i.e., the near invariance of all $\{a_\nu(t)\}$ during τ and the Poisson approximation for reaction firings) but it treats τ as the random variable and calculates K as the deterministic quantity. The question then is: What is the advantage, if any, of K leaping over τ leaping? Auger et al. [3] and Cai and Xu [14] argue that the advantages are two-fold: (i) focusing on K rather than τ makes it easier to ensure satisfaction of the leap con-

dition Eq. (3.20) (including avoiding negative populations), and (ii) computational effort can be reduced by ordering the reactions from largest propensity to smallest during the allocation phase.

Point (i) is valid to an extent in that K represents an upper bound on the possible number of firings of an individual reaction R_μ in the same vein as $n_\mu(t)$ in the binomial τ-leaping approach (see above). In principle, therefore, it is possible to choose a value of K that ensures that no reaction can fire enough times so as to cause a violation of the leap condition. However, this is far too restrictive a criterion for any practical simulation method. Alternatively, one could loosen the criterion to simply ensure that negative populations do not arise. Doing so erodes the ability to ensure satisfaction of the leap condition for all reactions but this may be acceptable in some situations. Still, even this approach will be overly restrictive in many cases [14]. Thus, it is preferable in practice to choose K in a manner analogous to the τ-selection approaches of Sec. 3.4 and provide a mechanism for avoiding negative populations. Of course, just as in the binomial τ-leaping case, this can be a challenging task in general. We refer the reader to Refs. [3] and [14] for a more detailed discussion of this issue and the specific K-selection procedures proposed by the authors. The important point here, however, is that the contention that K leaping is better at ensuring satisfaction of the leap condition than τ leaping is largely untrue. The one caveat is that K leaping does avoid the extreme outlier deviates that can arise in τ leaping.

The ability to reduce computational effort by ordering reactions is a more intriguing characteristic of K leaping and is reminiscent of the approach used to optimize the linear search phase of the DM variant of the SSA [22, 39, 75] (see Sec. 2.3.1). In principle, both τ leaping and K leaping require a total of M

random number generations at each step: M Poisson random deviates in the case of τ leaping, M–1 binomial random deviates [because $k_M(\tau) = K - \sum_{\nu=1}^{M-1} k_\nu(\tau)$] and one Gamma random deviate (for τ) in the case of K leaping. However, by ordering the reactions in K leaping from largest propensity to smallest one increases the chances that the K total firings will be completely allocated in $M' < M-1$ trials. This is analogous to reducing the search depth $\mathbb{S}$ in the linear search phase of the DM and the ordering methods discussed in Sec. 2.3.1 can thus be employed in this context. Through various example systems, both Auger et al. [3] and Cai and Xu [14] demonstrated modest, yet not insignificant, improvements in the efficiency of K leaping relative to τ leaping due to the ordering of reactions.

Other higher-order leaping methods [12, 13, 116]. Another proposed extension to the τ-leaping approach are the so-called Poisson Runge-Kutta (PRK) methods of Burrage and Tian [12] (also see Burrage et al. [13]). Like Rathinam et al. [92], Burrage and Tian [12] were motivated to develop higher-order leaping methods by Gillespie's work on the midpoint τ-leaping approach [47]. As discussed above, the idea behind the estimated-midpoint approach is to estimate the state of the system at the time $t+\tau/2$ and then approximate

$$K_\mu(\tau|\mathbf{x_t}, t) \approx \mathcal{P}_\mu\left(\widetilde{a}_\mu(t+\tau/2)\tau\right), \tag{3.48}$$

where the estimated midpoint propensities $\widetilde{a}_\mu(t+\tau/2)$ are calculated (in the case of τ-independent propensity constants) as

$$\begin{aligned}\widetilde{a}_\mu(t+\tau/2) &= a_\mu(t) + \langle \Delta a_\mu(\tau/2)\rangle \\ &= a_\mu(t) + \frac{\tau}{2}\sum_{\nu=1}^{M} f_{\mu\nu}(t) a_\nu(t).\end{aligned} \tag{3.49}$$

Here, the second equality follows from Eq. (3.18). We see in Eq. (3.49) that the estimated-midpoint method is an explicit approach since all of the required terms

are evaluated at the current time t.

Burrage and Tian [12] proposed a generalization of this scheme based on S intermediate approximation points, i.e.,

$$K_\mu(\tau|\mathbf{x_t},t) \approx \mathcal{P}_\mu\left(\sum_{s=1}^{S}\phi_s\widetilde{a}_\mu(t+\theta_s\tau)\tau\right), \tag{3.50}$$

$$\sum_{s=1}^{S}\phi_s = 1,$$

$$\theta_s \leq 1.$$

As in Eq. (3.49), the intermediate propensities are estimated as

$$\widetilde{a}_\mu(t+\theta_s\tau) \approx a_\mu(t) + \theta_s\tau\sum_{\nu=1}^{M} f_{\mu\nu}(t)a_\nu(t). \tag{3.51}$$

This is a straightforward extension of τ leaping that retains the original explicit approach ($S=1$, $\phi=1$, $\theta=0$) and the estimated-midpoint approach ($S=1$, $\phi=1$, $\theta=1/2$) as special cases.

In Ref. [12], Burrage and Tian advanced two new higher-order approaches that they termed the Heun and R2 PRK methods. Both are two-stage methods ($S=2$). The Heun method considers the propensity at the current time t and the estimated propensity at the end of the step, $t+\tau$, giving equal weight to each. This means that $\phi_1=\phi_2=1/2$, $\theta_1=0$ and $\theta_2=1$. The R2 method considers the propensity at the current time t ($\theta_1=0$) and the estimated propensity at the time $t+2\tau/3$ ($\theta_2=2/3$), giving added weight to the latter. Specifically, $\phi_1=1/4$ and $\phi_2=3/4$. These choices are optimal in the sense that they minimize the local error for a two-stage PRK method [12].

When applied to a simple example system, Burrage and Tian [12] showed that the higher-order methods (midpoint, Heun and R2) are significantly more accurate

than the explicit τ-leaping approach for a given error control parameter ϵ. The R2 method performed the best, followed by the midpoint method and then the Heun. These results are consistent with the local error characteristics of the corresponding ODE methods.

Another method that can be considered to be a higher-order leaping approach is the "unbiased" τ-leaping method of Xu and Cai [116]. These authors propose solving approximate chemical master equations for the mean and variance of each $K_\mu(\tau|\mathbf{x_t}, t)$ and then using them to generate Poisson [if $\langle K_\mu(\tau|\mathbf{x_t}, t)\rangle < 10$] or Gaussian [if $\langle K_\mu(\tau|\mathbf{x_t}, t)\rangle \geq 10$] random deviates for the reaction firings. The overhead associated with such a procedure is obviously high. However, Xu and Cai [116] show through various example systems that the procedure produces much more accurate results than either the simple explicit τ-leaping approach or the estimated-midpoint method.

Miscellaneous leaping methods [1, 70, 91, 94, 113, 114]. Additional leaping methods that have been proposed in the literature include Anderson's post-leap checking procedure [1] (see Sec. 3.4.3), the spatial τ-leaping approaches of Marquez-Lago and Burrage [70] and Rossinelli et al. [94] (see Sec. 3.6.2), the controllable approximative stochastic (COAST) algorithm of Wagner et al. [114], the reversible-equivalent-monomolecular τ-leaping (REMM-τ) approach of Rathinam and El Samad [91], and the particle-based τ-leaping method of Vlachos [113].

The COAST algorithm [114] is particularly interesting because it bears strong similarities to the PLA and appears to be based on the same basic principles. As in the PLA, reactions in COAST are dynamically classified during the course of a simulation into various categories and the numbers of firings are subsequently

determined by sampling from appropriate probability distributions based on the classifications. The categories in COAST correspond to the ES, Langevin and deterministic classifications of the PLA (i.e., the Poisson classification is omitted—see below) and ES reactions are treated using a modified version of the FRM, again similar to the PLA. Wagner et al. [114] also present a τ-selection procedure that is reminiscent of the RB τ-selection approach of Sec. 3.4.1 in that characteristic time intervals τ_μ^{COAST} are calculated for each reaction in the system and $\tau = \min\{\tau_\nu^{\text{COAST}}\}$, $\nu \in \{1 \ldots M\}$.

A primary difference between the COAST algorithm and the PLA is that Wagner et al. [114] assume that the base-level description of reaction dynamics is binomial in nature rather than Poissonian. As such, a central theme of Ref. [114] is deriving the conditions under which a binomial distribution can be well approximated by a Gaussian. This is why the Poisson classification is omitted in the algorithm. It is unclear why the authors make this assumption and we can only speculate that they were influenced by the works on binomial τ leaping [25, 109].

Other differences between COAST and the PLA include how ES reactions are handled, the details of the τ-selection procedure and how the species updates are performed. With regard to ES reactions, the procedure in COAST is to calculate successive values of τ_μ^{ES} for each ES reaction, updating the reactant populations after each firing, until $\sum_{j=1}^{J} \tau_{\mu,j}^{\text{ES}} > \tau$. The number of firings is then $k_\mu(\tau) = J - 1$. This means that multiple ES reactions can fire during τ and each reaction can fire multiple times. This is contrary to the situation in the PLA where an iterative procedure is employed to ensure that at most one ES reaction fires in τ (see Sec. 3.3.2). The τ-selection approach proposed by Wagner et al. [114] involves calculating the number of times, l_μ, that each reaction can fire before the leap condition is vio-

lated and then calculating the mean time $\left\langle \tau_\mu^{\mathrm{COAST}}(l_\mu) \right\rangle$ it takes for these firings to occur. τ is then set to the minimum of these. The procedure for calculating l_μ is reminiscent of the procedures proposed for calculating $n_\mu(t)$ in the binomial τ-leaping methods [67, 83, 109] and takes into account the number of different reaction channels that the reactants participate in. Finally, during the population update phase in COAST, the reactions are ordered from smallest $\left\langle \tau_\mu^{\mathrm{COAST}}(l_\mu) \right\rangle$ to largest. It is not entirely clear why this is done; perhaps it pertains to the speed of the reactions, fastest to slowest. The populations of the *reactant species only* are then updated by looping over all reactions. Not until all reactant populations have been updated are the product populations updated. Again, it is not entirely clear why this approach is necessary although the authors state that "...this splitting of updates is in accordance with the assumption that no particle reacts twice in $[t_0, T)$" [114]. In the PLA, such a splitting of updates is unnecessary.

The REMM-τ method of Rathinam and El Samad [91] is based on an entirely different premise than all other leaping methods discussed up until now. Indeed, it is the opinion of the author of this dissertation that it is probably better characterized as a model reduction approach (see Sec. 3.6.3) than a leaping method. Nonetheless, it is an intriguing method that deserves mention here. The basic idea of the REMM-τ approach is to partition a system of reactions into multiple small subsystems and then assume that each subsystem behaves independently of all others. The validity of this assumption increases with decreasing τ. The subsystems should be of a type, or "motif," that can be analyzed either exactly or approximately. In Ref. [91], Rathinam and El Samad focus exclusively on reversible pairs of elementary reactions, either monomolecular or bimolecular. They derive exact solutions for the elementary monomolecular reactions $S_1 \rightleftharpoons S_2$ and $S \rightleftharpoons \emptyset$ and then derive conditions under which various elementary bimolecular re-

actions (e.g., $S_1+S_2 \rightleftharpoons S_3$, etc.) can be approximated by a monomolecular reaction and, hence, analyzed using the exact solutions. Moreover, non-reversible reactions can be thought of as reversible reaction pairs with a backwards rate constant equal to zero. Thus, the entire reaction network is recast as a set of independent, reversible, monomolecular reaction pairs (hence the name "reversible-equivalent-monomolecular" τ leaping) and the system is evolved by approximating the state changes due to each pair during τ.

The main results of Ref. [91] can be summarized as follows. For the elementary reversible reaction pair $S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2$, the system state at time $t+\tau$ can be written as

$$X_i(t+\tau) = X_i(t) - \Delta X_{i\to j}(\tau) + \Delta X_{j\to i}(\tau), \;\; i = 1, 2, \tag{3.52}$$

where $\Delta X_{i\to j}(\tau)$ represents the number of molecules identified as S_i at time t that are *observed to be* of type S_j at time $t+\tau$. Rathinam and El Samad [91] show that, in isolation,

$$\Delta X_{i\to j}(\tau) \sim \mathcal{B}\left(X_i(t), p_i(\tau)\right), \tag{3.53}$$

$$p_i(\tau) \equiv \frac{c_i}{c_1+c_2}\left(1 - e^{-(c_1+c_2)\tau}\right), \tag{3.54}$$

where '$\sim$' denotes "distributed according to." Thus, after selecting a value of τ, one simply generates two binomial random deviates and updates the system according to Eq. (3.52). For the elementary birth-death process $S \underset{c_2}{\overset{c_1}{\rightleftharpoons}} \emptyset$, Rathinam and El Samad [91] show that

$$\Delta X_{S\to\emptyset}(\tau) \sim \mathcal{B}\left(X(t), 1 - e^{-c_1\tau}\right), \tag{3.55}$$

$$\Delta X_{\emptyset\to S}(\tau) \sim \mathcal{P}\left(\frac{c_2}{c_1}\left(1 - e^{-c_1\tau}\right)\right). \tag{3.56}$$

Thus, in this case, one generates one binomial[19] and one Poisson random deviate

[19]Notice that Eq. (3.55) follows directly from Eqs. (3.53) and (3.54) with $c_2 = 0$.

and updates the system according to

$$X(t+\tau) = X(t) - \Delta X_{S\to\emptyset}(\tau) + \Delta X_{\emptyset\to S}(\tau). \tag{3.57}$$

Rathinam and El Samad [91] also consider five different types of reversible, elementary, bimolecular reaction pairs and show how they can be approximated as one of these two types of monomolecular reaction pairs. The above results can then be applied to those reactions with some modifications to the basic parameters (e.g., c_1 and c_2). We refer the reader to Ref. [91] for further details.

The basic idea of the REMM-τ approach, i.e., proceeding by coarse time intervals and describing the reaction dynamics in terms of probability distributions rather than simulating each and every reaction firing, is certainly in the same spirit as all other leaping methods discussed in this dissertation. However, as mentioned above, we believe that the method is probably better characterized as a model reduction approach than a leaping method. Our reasoning is that the philosophy of identifying small network motifs that can be considered largely independent of the rest of the network and then treating them in some approximate way is the same that underlies model reduction methods based on rapid-equilibrium assumptions. A well-known example is the Michaelis-Menten approximation, where a set of three reactions is collapsed into one effective reaction with modified rate parameters (see Sec. 3.6.3). Often, the species involved participate in other reactions in the system as well. However, the motif is considered to act independently of the rest of the network because the dynamics are usually much faster than the other reactions in which the species participate.

This is essentially the same situation that the REMM-τ method was designed for. Indeed, Rathinam and El Samad [91] advertise the approach as a way to handle "small number and stiff" reaction networks, i.e., fast reaction subsystems

involving small numbers of molecules, and discuss extending it beyond the simple reversible-reaction-pair motif. They do not provide a procedure for selecting τ nor do they discuss how to deem whether the independence assumption is valid, only that it is expected to hold for small values of τ. In truth, the validity of the assumption likely relies on a rapid-equilibrium condition, again emphasizing the connection to model reduction. However, regardless of which category of approach the REMM-τ method falls in, it is an important advancement in that it addresses what is perhaps the biggest shortcoming of the leaping methods, the inability to handle small-number-and-stiff reaction subnetworks (as also pointed out by Cao et al. [21]). We discuss this issue further in Sec. 3.6.3 and provide an example of this shortcoming and the use of model reduction to overcome it in Sec. 4.2.3.

Finally, Vlachos [113] recently proposed a particle-based version of τ leaping with a focus on surface science applications. The idea is quite simple: describe the system in terms of the basic processes that can take place (e.g., adsorption, desorption, site-to-site hopping), calculate a τ and determine the set of firings $\{k_\nu(\tau)\}$ as in standard τ leaping, and then randomly apply the firings to individual particles in the system. In essence, the only difference between this approach and standard τ leaping is the latter application to individual particles.

Interestingly, an analogous situation exists between Gillespie's SSA [41, 42] and a method known as the N-fold way of Bortz et al. [10], another highly-cited method in the kMC literature. For all practical purposes, the SSA and the N-fold way are the same method. However, the N-fold way was developed within the context of Ising spin system models where the states of lattice sites change with rates dependent upon their local environments. Each lattice site is essentially a particle and complex models can contain thousands to millions of such sites. Instead of

looping through all sites in the system, Bortz et al. [10] proposed grouping sites into categories of which there are usually far fewer than individual sites. An example is the category $\{\uparrow;\downarrow\downarrow\downarrow\downarrow\}$, which corresponds to a lattice site in state $\uparrow$ (e.g., up-spin) surrounded by four lattice sites in the $\downarrow$ state (i.e., a two-dimensional lattice with first-nearest neighbor interactions). Essentially, $\{\uparrow;\downarrow\downarrow\downarrow\downarrow\}$ is a "species" with a population and we can represent transitions between states within the language of chemical kinetics as, e.g.,

$$\{\uparrow;\downarrow\downarrow\downarrow\downarrow\} \xrightarrow{c_{\uparrow;\downarrow\downarrow\downarrow\downarrow}} \{\downarrow;\downarrow\downarrow\downarrow\downarrow\}.$$

The approach then is to count up the number of sites within each category, determine the type and time of the next transition using Eqs. (2.14) and (2.15) (i.e., identical to the DM variant of the SSA), and then randomly select which site will undergo the transition.

Given the obvious connection between the SSA and the N-fold way and the fact that τ-leaping can be seen as an extension to the SSA, the particle-based approach of Vlachos [113] is a natural extension to τ leaping. However, a major complication that must be addressed in any practical application of the method is the possibility of conflicts in the selection of particles. Basically, it is possible to choose particles to undergo transitions that are no longer feasible because of the firing of another transition in the same simulation step. For example, when considering site-to-site hopping, one may determine that multiple hopping events will occur in the next simulation step. One randomly chooses a particle and moves it to a neighboring site. Now, assume that the next particle chosen happens to be adjacent to the site just occupied by the previous migrating particle. Obviously, the newly chosen particle cannot hop to that site. However, when calculating the hopping rates at the beginning of the step the site was unoccupied and hence contributed to the rate. Simply disallowing an attempted hop to that site may,

therefore, skew the statistics of the process and introduce error. There is thus a fundamental complication here that does not arise in the N-fold approach because of its one-reaction-at-a-time nature. Simply put, in the τ-leaping framework we are sacrificing knowledge of the order of events for the sake of efficiency and, hence, we cannot say that one process occurs prior or subsequent to any other. Allowing for *attempted* moves and then disallowing them if they are no longer feasible while still including them amongst the firing count (i.e., introducing null events) may be the solution to the problem but further investigation is necessary. Vlachos addresses this issue briefly at the end of Ref. [113] but his arguments are speculative and non-rigorous.

Comparisons to partitioned leaping [56]. The methods discussed above that are most closely related to the PLA are the explicit τ-leaping approach of Gillespie, Petzold and co-workers [18, 20, 47, 50] and the COAST algorithm of Wagner et al. [114]. It is our opinion that the PLA is superior to both methods for the following reasons. First, the PLA utilizes the entire theoretical multiscale framework laid out by Gillespie in Refs. [46] and [47]. The explicit τ-leaping method lacks this feature and thus does not capitalize on the advantages of a multiscale approach. Specifically, it cannot treat reactions at the less computationally-demanding Langevin and deterministic levels and it cannot force continuum simulations for comparison purposes. The COAST algorithm is a multiscale approach but Wagner et al. [114] do not utilize the rigorous theoretical foundation developed by Gillespie, instead choosing to derive their own framework based on a binomial distribution-based description of reaction dynamics. As stated above, it is unclear why they chose this path and whether various aspects of their approach are theoretically sound (i.e., the treatment of ES reactions, the τ-selection approach and

the population update procedure).

More importantly, however, the PLA is significantly simpler in construction and much easier to implement than both the explicit τ-leaping method and COAST. In particular, the SSA is naturally incorporated into the PLA because of its analogous relationship to the FRM (and NRM by extension). In the explicit τ-leaping algorithm, incorporation of the SSA is somewhat forced due to the desire by the authors to utilize the DM variant. The incorporation is somewhat more natural in COAST although the procedure proposed in Ref. [114] is less rigorous than the iterative approach employed in the PLA (see Sec. 3.3.2). Furthermore, again, various aspects of the COAST algorithm are confusing and not well motivated. Conversely, the basic principles of the PLA are simple and straightforward, being based on the well-formulated ideas of Gillespie [46, 47] and drawing on an analogy with the well-known FRM/NRM variants of the SSA [40–42].

Cao et al. [18] have also added additional machinery, and hence complexity, to the explicit τ-leaping method in order to deal with the problem of negative populations. Negative populations are *mostly* avoided in the PLA by the construction of the algorithm (see below for further discussion). Reactions with small reactant populations are *automatically* detected by the PLA and classified as ES, thus preventing them from firing multiple times in τ. This is analogous to the critical/non-critical approach proposed in Ref. [18] but without the need for the added machinery presented there. Overall, we believe that the PLA represents the ideal explicit leaping approach and is the natural progeny to the foundational ideas laid out in Refs. [46] and [47].

The implicit τ-leaping methods proposed by Gillespie, Petzold and co-workers [24, 92, 93] and the higher-order PRK methods of Burrage and Tian [12, 13]

offer important direction and motivation for future enhancements to the PLA. In principle, it should be relatively straightforward to implement implicit and Runge-Kutta variants of the PLA based on Eqs. (3.41), (3.44) and (3.50). As discussed above, each of these [including Eq. (3.50)] have analogous forms at the SDE and ODE levels. The higher-order approach proposed by Xu and Cai [116] is likely too computationally intensive to employ as a practical simulation method. However, it may provide a good benchmark against which the accuracy of the envisioned PLA extensions can be compared.

Similarly, it should be relatively straightforward to implement a K-leaping [3, 14] variant of the PLA, although it will require some thought to ascertain the appropriate procedure for including the continuous-stochastic and continuous-deterministic descriptions into the framework. As explained, the only true advantage to the K-leaping approach is the potential to reduce computational effort by ordering reactions during the allocation phase. Whether this has a significant impact on the practicability of the PLA will require further investigation.

The REMM-τ approach of Rathinam and El Samad [91] offers another intriguing possibility for extending the PLA. We can envision an approach similar in spirit to the hybrid explicit/implicit τ-leaping method of Cao et al. [21] but that uses the methods presented in Ref. [91] [e.g., Eqs. (3.52)–(3.57)] for describing the dynamics of fast reversible reaction pairs rather than the implicit τ-leaping methods. This is a more theoretically-sound approach that avoids the *ad hoc* downshifting strategy proposed in [92] and is also capable of handling the small-number-and-stiff reactions that the method of Cao et al. [21] cannot.

Furthermore, it should be possible to derive the conditions under which the binomial and Poisson descriptions in Eqs. (3.53), (3.55) and (3.56) can be well-

approximated by Gaussian distributions and Dirac delta functions, just as is done in the PLA. This opens the door to the possibility of a multiscale model reduction framework that can be integrated with the multiscale leaping framework of the PLA, an exciting prospect. However, two critical developments will be necessary to make this approach practicable. First, a procedure for determining the time interval over which the model reduction is valid (analogous to the τ-selection procedures of Sec. 3.4) will need to be developed. Rathinam and El Samad [91] provide no such guidance and we are unaware of any such theory. Secondly, the theoretical framework presented in Ref. [91] will need to expanded to motifs other than reversible reaction pairs, ideally to the general case. A recent article by Sinitsyn et al. [106] may provide the necessary theoretical foundation for this. In Sec. 3.6.3, we discuss further the possibility of integrating model reduction methods with the PLA.

It should also be possible to develop a particle-based variant of the PLA in the spirit of Vlachos [113], provided that the problem of conflicts in the selection of particles can be resolved. This would be an important development, expanding the applicability of the PLA to a much wider range of problems. While Vlachos' focus was on surface science applications [113], particle-based kMC approaches are utilized in a wide variety of disciplines, including the ever-important field of computational biology. For example, Yang et al. [117] recently proposed a "rule-based" implementation of the SSA that applies transitions to complex biological molecules. Because the numbers of states that the molecules can exist in are combinatorially large, molecules and molecular complexes are treated as particles (or "agents") rather than as populations. A particle-based version of the PLA would thus prove particularly useful in the simulation of such models.

Extending the PLA to handle spatially-inhomogeneous reaction-diffusion systems in the same vein as the methods proposed by Marquez-Lago and Burrage [70] and Rossinelli et al. [94] is another important area of future investigation. In Sec. 3.6.2, we address this issue in detail, discussing the shortcomings present in the methods of Refs. [70] and [94] and the challenges associated with implementing an accurate and efficient spatial leaping approach.

Finally, as explained above, the motivation behind the development of the binomial τ-leaping methods [25, 67, 83, 84, 109] was to avoid the occurrence of negative populations. Doing so requires replacing the Poisson description of the reaction dynamics by a binomial description and including various non-trivial procedures for calculating the associated parameters of the binomial distributions. However, as also explained above, the PLA mostly prevents the occurrence of negative populations simply by its construction, i.e., through the inclusion of the ES classification. Furthermore, Anderson's post-leap checking procedure [1] (Sec. 3.4.3) is a rigorous and relatively simple approach for ensuring satisfaction of the leap condition for all reactions in a system and, hence, also avoids negative populations by definition. As such, integrating post-leaping checking into the PLA to handle those rare instances where negative populations do arise is, in the opinion of the author of this dissertation, a much more simple and preferable approach for dealing with the problem (see Sec. 3.6.1 for further discussion regarding integrating post-leap checking into the PLA). Thus, one could argue that the PLA coupled with post-leap checking essentially renders the binomial τ-leaping methods *obsolete*.

This is a somewhat provocative statement and it may, in fact, be the case that certain aspects of the binomial τ-leaping methods will prove important and useful in future extensions of the PLA and other leaping methods. Indeed, it is argued by

various authors that binomial τ-leaping is more accurate and more efficient than its Poisson-based counterparts [25, 67, 83, 84, 109], at least in some cases. The idea of grouping reactions in a leaping algorithm, as proposed by Pettigrew and Resat [84], is also intriguing. Thus, further investigation into these issues is certainly warranted. However, from the perspective of avoiding negative populations, we stand by our statement that the use of binomial random deviates in τ-leaping is unnecessary and unwarranted in light of the developments of the PLA and post-leap checking, especially considering the added complexity and overhead associated with doing so.

3.6 Future directions

3.6.1 Improved τ selection by intermixing pre-leap calculations and post-leap checks

By far, the most computationally expensive aspect of the PLA, and leaping algorithms in general, is the procedure for selecting the time step τ. In Sec. 3.4, we described three procedures for accomplishing this that have been proposed in the literature to date: the pre-leap RB τ-selection procedure (Sec. 3.4.1) [20, 47, 50, 56], the pre-leap SB approach (Sec. 3.4.2) [20, 56] and post-leaping checking (Sec. 3.4.3) [1]. RB τ selection is, generally speaking, the most computationally expensive of the pre-leap approaches.[20] This is because (i) there is added cost associated with calculating the quantities $f_{\mu\nu}(t)$ [Eq. (3.9)] for use in, e.g., Eq. (3.27) [contrast with the simpler quantities for SB τ selection in Eq. (3.30)], and (ii) there are often far

[20] As we will see in Chapter 4, however, this is not always true.

more reactions in a system than there are species; the RB approach requires a τ_μ^{leap} calculation for each reaction while the SB approach requires a T_i^{leap} calculation for each species.

The post-leap checking procedure of Anderson [1] can, in principle, be quite computationally expensive as well since every time the leap condition Eq. (3.20) is violated one binomial random number must be generated for each reaction in the system (see Sec. 3.4.3). The potential benefit of the approach, therefore, in terms of computational speed, is if one can minimize the number of violations of Eq. (3.20) realized during the course of a simulation. There would still be some overhead associated with the process of checking post-leap but it would almost certainly be small, if not negligible, compared to the costs of random number generation or pre-leap calculations.[21]

The trick, then, is to develop a low-cost strategy that minimizes the frequency with which the leap condition is violated while concurrently maximizing the average size of the time steps taken (obviously, we can avoid violating the leap condition by choosing very small time steps but then we would suffer in terms of efficiency). In [1], Anderson proposed a simple strategy along these lines (see Sec. 3.4.3). In a direct comparison with the τ-leap method of Cao et al. [20], implemented using the SB τ-selection formula Eq. (3.30), for a simple model system (investigated in Sec. 4.1.1 of this dissertation), Anderson showed that for a given level of accuracy his post-leap checking procedure outperformed SB τ leaping in terms of speed [1]. However, he also showed that for a given ϵ, post-leap checking was slower than τ leaping, though significantly more accurate.

[21] It might also be possible to reduce the cost of checking even further by sorting reactions based on their propensity to violate Eq. (3.20), similar to the procedure adopted by McCollum et al. [75] in their optimized DM.

In general, these results indicate the potential advantages associated with post-leap checking. They also suggest, however, that the approach may not be sufficient on its own; violations of the leap condition occurred often enough to make the method only marginally faster than SB τ leaping [1]. Thus, we propose developing a new τ-selection strategy based primarily on post-leap checking but with pre-leap calculations mixed in intermittently. The basic idea would be to monitor the propensity changes in relation to the leap condition and to perform a pre-leap calculation whenever the algorithm appears to be struggling to find optimal time steps. This would occur if the leap condition were being violated frequently or, conversely, if it were being overly satisfied (i.e, unnecessarily small time steps). The pre-leap calculation would act to rectify these problems by putting the algorithm "back on track," as it were. We note that Anderson actually follows this strategy to some extent in that in the algorithm proposed in [1] the initial value of τ is calculated pre-leap using Eq. (3.30). Thus, our idea is to extend this approach to steps in the simulation other than the initial.

It is unclear whether this strategy will have significant impact when applied to simple systems under "normal" conditions, e.g., the example systems considered in Chapter 4. However, we do believe that it will prove to be of great importance in more advanced settings, such as extensions to spatial domains (Sec. 3.6.2) or when considering time-varying environmental quantities. We anticipate that in such situations the simple procedure proposed by Anderson [1] might prove insufficient. Yet, it will be infeasible to employ a τ-selection procedure based purely on pre-leap calculations. Thus, intermixing an occasional expensive pre-leap calculation with multiple inexpensive post-leap checks might prove to be the optimal approach. Work in this direction will be undertaken in the near future.

3.6.2 Spatial extension

The most natural extension to the PLA is one that introduces spatial extent and diffusion. In both materials science [85, 95] and biology [15, 32, 111, 118], situations in which the reactive environment cannot be considered homogeneous and well-mixed are common. In Sec. 2.5, we discussed approaches used in such situations at the exact-stochastic level. Thus, a spatial PLA (SPLA) implementation should naturally extend these methods.

In keeping with the schematic of Fig. 2.4, we envision discretizing an inhomogeneous reaction environment into numerous homogeneous subvolumes loosely coupled through diffusive transport. The SPLA will then simulate multiple reaction firings and diffusion events at each simulation step, representing a significant improvement over spatial-SSA methods such as the NSM (Sec. 2.5), which fire only one reaction or diffusion event at each step.

However, the implementation of such an approach is not trivial. First, as explained in Sec. 3.3 of this dissertation, a particularly attractive feature of the PLA is its seamless transition to the FRM/NRM variants of the SSA (Sec. 2.4) in the limit of small populations. In the spatial case, we would like a similar transition to occur, perhaps to the NSM. However, there is a fundamental difficulty in achieving this: it is not, in principle, permissible to arbitrarily group reactions in leaping algorithms as is done in spatial-SSA methods. This is because the leap condition Eq. (3.4), on which the leaping approach is based, applies to individual reactions (and diffusion events), not groups. Basically, there is no guarantee that a negligible change in a_0^l will correspond to negligible changes in the individual values of a_μ^l that comprise the sum. Indeed, in many, if not most, cases it will not. Thus, it is very possible that the leap condition will be violated for individual

events in the group even when it is satisfied for the group as a whole.

Second, as explained in Sec. 3.6.1, τ selection is the most computationally-expensive aspect of the PLA. In the case of RB τ selection (Sec. 3.4.1), one τ_μ^{leap} calculation is required for each reaction in the system; in SB τ selection (Sec. 3.4.2), one T_i^{leap} calculation is required for each species. For spatially-discretized systems, this problem is intensified. In general, each subvolume is assigned local copies of each reaction and species, essentially multiplying the numbers of each by a factor of L. Furthermore, events describing diffusive transport are added to each subvolume, further increasing the number of events that must be considered. Thus, it is not hard to imagine situations where the computational load associated with τ selection renders the method infeasible.

We note that to date two attempted applications of spatial τ leaping have been proposed in the literature, those of Marquez-Lago and Burrage [70] and Rossinelli et al. [94]. The method in Ref. [70] is a leaping analogue of the NSM. It is based on grouping reactions and diffusion events by subvolume and calculating a leap time interval for the group. The global τ is then chosen as the smallest over all subvolumes. However, as explained above, τ selection cannot be applied to groups as a whole. Thus, an individual τ-selection calculation must still be performed for each event or species in the system. As a result, the reduction in computational effort seen in spatial-SSA implementations such as the NSM is largely absent in the method of Marquez-Lago and Burrage. Furthermore, the authors neglect diffusion incoming into subvolumes in the τ-selection procedure and employ an outdated τ-selection approach [47]. The approach of Rossinelli et al. [94] is somewhat similar but differs in spirit in that it does not attempt to emulate a spatial-SSA approach. In fact, there is no mechanism for transitioning to a spatial-SSA method in the

limit of small populations. A more serious shortcoming is that reactions and diffusion events are considered *independently* of each other in τ selection. There is no justifiable reason for doing this. Indeed, the firings of reactions are intimately tied to the rates at which entities diffuse into and out of subvolumes, and *vice versa*.

Work is currently ongoing to develop a spatial version of the PLA that overcomes each of these shortcomings. Specifically, both incoming and outgoing diffusion are accounted for in the τ-selection process, reactions and diffusion events are considered together in τ selection and, being built upon the PLA, the method segues to an exact-stochastic method for small populations. The approach can be seen as a straightforward and accurate implementation of spatial leaping against which future enhancements can be tested. The approach does not, however, address the fundamental problem of the high cost of τ selection. We believe that incorporating post-leap checking [1] (Secs. 3.4.3 and 3.6.1) might provide a partial solution. Grouping events and performing a single τ-selection calculation on the group would also be a great benefit. Work is currently underway to determine how this might be accomplished in a theoretically-sound manner.

3.6.3 Dynamic model reduction

The stated goal of leaping methods such as the PLA is to address the general problem of timescale disparity in population-dynamical systems [46, 47]. However, the characteristic timescale of a reaction is dependant upon two factors: the populations of the reactant species and the rate parameters. Leaping methods specifically target only the former, the effects of population size. As explained in Sec. 3.1, the basis of the leaping methodology is to advance forward in time by intervals τ over which many reaction firings can occur without the propensities changing "appre-

ciably." This is a tacit statement regarding population size; reactant populations must be large in order for a single reaction firing to have a negligible impact on its propensity.

It is not uncommon, however, for timescale disparities to arise because of disparities in rate constants. Moreover, fast reactions can often be associated with small reactant populations. For example, in cellular biology, the binding and unbinding of transcription factors to genes, of which there is usually but one copy (never more than a few), occurs much more rapidly than other processes in the cell, such as transcription and translation [61, 69, 73, 74, 89, 90, 100]. In materials fabrication, the diffusion of adatoms on a surface is often orders of magnitude faster than the deposition rate, especially at low coverage [71, 72]. In situations such as these, the PLA, and leaping algorithms in general, will "bog down" computationally. Especially in the case of small populations, a small number of firings, or even a single one, will significantly alter the propensity, and if the rate constants are large, then the time interval over which this will occur, i.e., the time step, will be small. In Chapter 4 (Sec. 4.2.3), we consider an example system under just such conditions.

A commonly-used strategy in chemical kinetics to overcome problems like this is *model reduction*. There are various flavors of model reduction, but most rely on some form of rapid-equilibrium assumption [89]. The basic idea is to remove explicit consideration of fast reactions by collapsing them down into a smaller set of slow reactions with *effective* rate expressions. A well-known example of this in biology is the Michaelis-Menten mechanism, where a "substrate" S is converted into a product P through the action of an enzyme E. The full set of reactions

describing the process is

$$E + S \underset{c_{-1}}{\overset{c_1}{\rightleftharpoons}} \{E{\cdot}S\} \xrightarrow{c_{\rm cat}} E + P,$$

where the rate parameters are given in units of $[time^{-1}]$. By assuming that the enzyme-substrate complex $\{E{\cdot}S\}$ is in quasi-equilibrium, this set of reactions can be reduced to a single one,

$$S \xrightarrow{a_\mu} P,$$

with an effective rate expression

$$a_\mu = \frac{c_{\rm cat} E_T X_S}{C_M + X_S},$$

where $E_T \equiv X_E + X_{\{E\cdot S\}}$ and $C_M \equiv (c_{-1} + c_{\rm cat})/c_1$.

Removing, in this way, the fast, reversible binding/unbinding reactions $E+S \rightleftharpoons \{E{\cdot}S\}$ from explicit consideration and incorporating their effects into a single, effective reaction $S \rightarrow P$ can greatly accelerate the simulation process, especially for large, complex reaction networks that contain numerous such subnetworks. In the case of the PLA, and leaping algorithms in general, the primary improvement is in the calculation of *larger time steps*. This is an important point. Reducing the number of reactions to be explicitly considered certainly helps, but this is *not* the primary advantage to using model reduction in the context of a leaping algorithm. Rather, model reduction acts to remove the aspects of a network that constitute the primary bottleneck in τ selection. Leaping and model reduction are thus *complementary* approaches, the former specifically handling timescale disparities associated with disparate species populations while the latter targets those due to disparities in rate constants. Integrating the PLA with some form of model reduction is thus an area of fundamental importance and great future interest.

In Chapter 4 (Secs. 4.2.2 and 4.2.3), we illustrate the advantages of such an integration using simple model reductions similar to that outlined above. However,

there is a fundamental shortcoming to such an approach: the model reduction is *static*, i.e., it is imposed at the outset of a simulation and remains in effect for the duration. Species populations change in time, however, and, in principle, rate parameters can change as well (i.e., if they are dependent on environmental quantities such as volume or temperature). It is entirely possible, therefore, for the conditions under which the reduction is valid to be violated at some point during the course of a simulation due to the time evolution of the system. Moreover, the process can be cyclic, with the conditions being met and then violated and then met again, and so on. Static model reductions either ignore such situations or simply disallow them. Thus, it is advantageous and preferable to employ a model reduction scheme that is *dynamic*, automatically reducing out fast reactions when the conditions permit but reinserting them when the opposite is true.

Recently, a number of methods implementing strategies along these lines have been proposed [30, 97, 98]. Although differing in various implementation details, the basic idea is the same. First, the system is partitioned into fast and slow subsets. The full system is then evolved until such time that the fast subsystem is deemed to be in quasi-equilibrium. During the equilibration phase, data is collected for the species involved in the fast reactions and probability distribution functions are generated for each. The fast reactions are then removed from explicit consideration and the system is advanced to the time of the next slow reaction firing. The slow subsystem is updated and the populations in the fast subsystem are drawn from the probability distributions generated during equilibration. The fast reactions are then updated and the system re-equilibrated. The process is repeated as necessary. In Fig. 3.2, we provide a simple schematic illustrating this procedure.

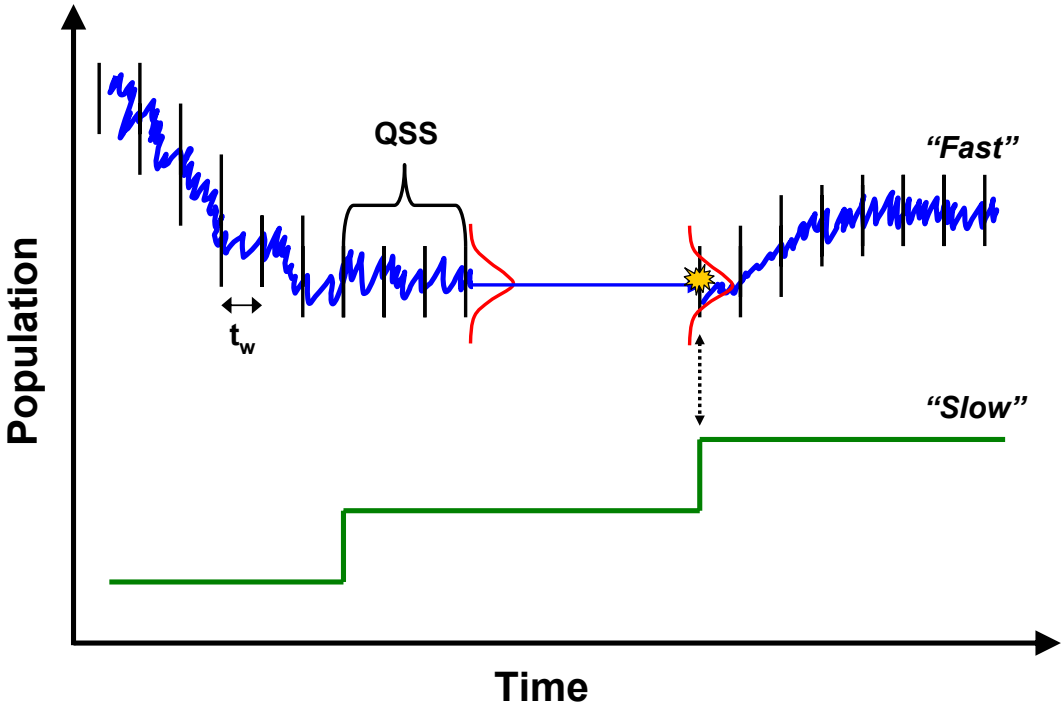

Figure 3.2: Simple illustration of the "brute force" dynamic model reduction schemes of Refs. [30, 97, 98]. Illustrated here are time courses for two species, one belonging to the "fast" subsystem and the other to the "slow." A sampling window of size t_w is defined and data is collected for the fast subsystem until it is deemed to be in quasi-steady-state (QSS). Probability distributions are then generated for all fast species and the system is advanced to the time of the next slow reaction firing. At that point, populations for the fast species are drawn from the generated distributions and the procedure is repeated. Note that slow reactions can fire during the equilibration phase. This may or may not affect the equilibration process.

The approaches in Refs. [30, 97, 98] can be thought of as "brute force" dynamic model reduction schemes. They are somewhat clumsy and difficult to implement and rely on a heuristic partitioning into fast and slow subsystems. Nevertheless, they illustrate in a straightforward way the general approach that we wish to integrate with the PLA. We seek a scheme where reactions are included or excluded "on the fly" based on the conditions present within the system. Reactions that are

retained are simulated using the PLA with appropriately-modified rate expressions that account for the effects of the reactions (and species) that are excluded.

We note that the field of model reduction is an extremely active one, with multitudes of approaches having been proposed to treat complex systems in, e.g., biology [11, 16, 17, 19, 54, 63, 78, 88, 105] and combustion science [110], to name just a few. It might be the case, therefore, that we can accomplish our goal by simply perusing the literature. Alternatively, we might develop a novel method based on aspects of existing methods. In this regard, the methods of Cao et al. [16, 17, 19], Goutsias [54] and Rathinam and El Samad [91] (see Sec. 3.5) look particularly promising. Work in this direction is currently underway and will be expanded upon in the near future.

Chapter 4

Select Applications of Partitioned Leaping

In this chapter, we present results of various applications of the partitioned-leaping algorithm (PLA), which was introduced in Chapter 3 of this dissertation. We begin in Sec. 4.1 by examining various simple systems that exemplify, in a general way, the utility of the PLA. We then proceed in Sec. 4.2 to consider prototypical biochemical networks, systems that are more complex than those in Sec. 4.1 but are still simple enough to understand from an intuitive standpoint. These are particularly useful in ascertaining the strengths and limitations of the method.

In all cases, we compare the performance of the PLA to the SSA in terms of computational efficiency and accuracy. In some of the cases, we also compare to deterministic predictions in order to quantify the extent of stochastic effects. PLA simulations are performed with the parameters '≈ 1'$=3$ and '$\gg 1$'$= 100$ and using both the reaction-based (RB) and species-based (SB) τ-selection procedures of Secs. 3.4.1 and 3.4.2, respectively. For the RB calculations, Eq. (3.27) is used, while for the SB we use Eq. (3.30). Furthermore, SSA and deterministic simulations are performed by manipulating the classification parameters of the PLA. Specifically, setting '≈ 1'$=\infty$ forces all reactions to be classified at the ES level at all steps of a simulation. The algorithm then *becomes* an implementation of the NRM variant of the SSA. Similarly, setting '$\gg 1$'$= 0$ forces a fully deterministic description, and the algorithm becomes a simple forward Euler method for solving ordinary differential equations (ODEs).[1]

[1]Note that we must also turn rounding off when performing deterministic simulations. Recall that the standard implementation of the PLA involves rounding the numbers of reactions firings,

4.1 Simple systems

We begin by considering three simple systems that exemplify the utility of the PLA. The first is the "decaying-dimerizing" reaction set, a toy network that has been utilized numerous times by Gillespie and co-workers [20, 21, 47, 50, 92], as well as others [3, 14, 83, 97, 109], to demonstrate the utility of various leaping and hybrid algorithms. Next, we consider a simple model of clustering that is motivated by problems in various fields, including materials and atmospheric sciences, polymer chemistry and biology. Finally, we investigate a biologically-inspired model system that includes a crude description of gene expression dynamics along with protein-protein interactions.

4.1.1 Decaying-dimerizing

The decaying-dimerizing reaction set is comprised of the following four reactions:

$$\begin{aligned} R_1: &\quad S_1 \xrightarrow{c_1} \emptyset, \\ R_2: &\quad 2S_1 \xrightarrow{c_2} S_2, \\ R_3: &\quad S_2 \xrightarrow{c_3} 2S_1, \\ R_4: &\quad S_2 \xrightarrow{c_4} S_3, \end{aligned} \tag{4.1}$$

where $\emptyset$ represents a sink. Very simply, species S_1 either degrades or is converted into S_3 through the unstable dimer S_2. In all that follows, we use the values of the rate constants and initial populations from Ref. [47]: $c_1 = 1.0$, $c_2 = 0.002$, $c_3 = 0.5$, $c_4 = 0.02$ (all in arbitrary units of $[time^{-1}]$), $X_1(0) = 10^5$, $X_2(0) = 0$ and $X_3(0) = 0$.

In Fig. 4.1, we show an example time course for this system. In Fig. 4.1(a), we

$\{k_\nu\}$, before updating the species populations (Sec. 3.3.2).

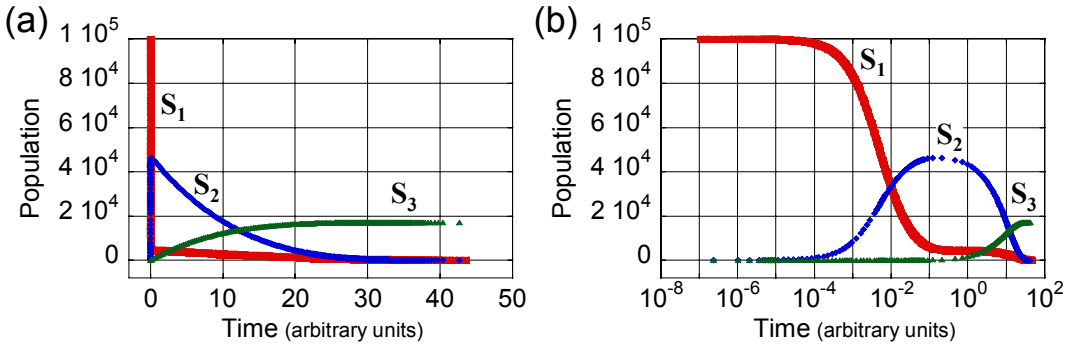

Figure 4.1: Example decaying-dimerizing time course. Initial populations are $X_1(0) = 10^5$, $X_2(0) = 0$ and $X_3(0) = 0$. In (a), the time course is shown on standard axes, while in (b) the time axis (x-axis) is given on a log scale, providing visual access to the short-time behavior of the initial transient.

see that the system experiences an initial transient whereby approximately one-half of the S_1 entities are converted into S_2. Most of the action occurs within the first 0.1 time units, as evident in Fig. 4.1(b) where the time axis is shown on a log scale for convenience. Upon completion of the transient, the populations of both S_1 and S_2 steadily decrease while that for S_3 increases. Exhaustion of all reactant entities occurs by $\sim$45 time units.

To illustrate the practical advantage of using the PLA to simulate this system, we present in Fig. 4.2 results of a step and timing analysis comparing the performance of the PLA to the SSA. We consider both the SB (Sec. 3.4.2) and RB (Sec. 3.4.1) τ-selection variants of the PLA with $\epsilon = 0.01$, 0.03 and 0.05. In Fig. 4.2(a), we see that the PLA requires far fewer simulation steps for each run than does the SSA, by about an order of magnitude at $\epsilon = 0.01$ and nearly three orders of magnitude at $\epsilon = 0.05$. We also see that the PLA-RB consistently requires fewer steps than the PLA-SB, which is consistent with the tighter constraint implicit in the SB τ calculation (see Sec. 3.4.2). Interestingly, however, we see in Fig. 4.2(b) that both τ-selection procedures perform almost identically in terms

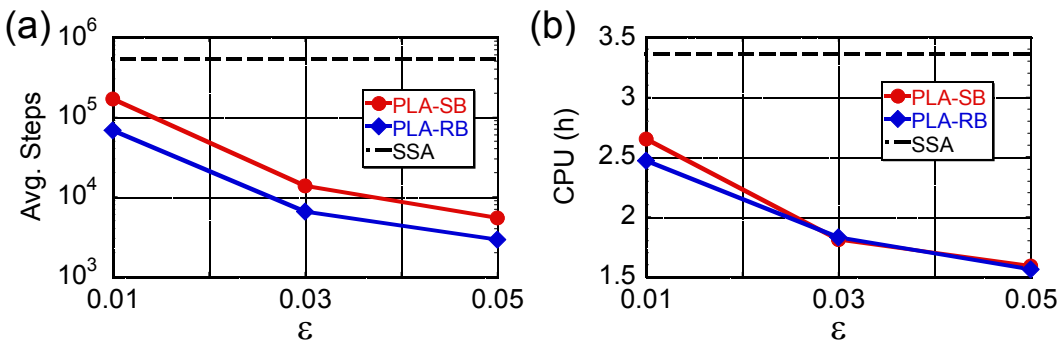

Figure 4.2: Average numbers of steps from (a) and total CPU times for (b) 10 000 PLA and SSA simulation runs of the decaying-dimerizing reaction set. PLA results are shown for both the RB and SB τ-selection variants (Secs. 3.4.1 and 3.4.2, respectively) for various values of ϵ. All simulations were run on a 3.60 GHz Pentium Xeon processor.

of CPU run time. This is because each SB τ calculation is computationally less expensive than each RB calculation (Sec. 3.4.2).

In Fig. 4.3, we plot the classifications achieved for each reaction at each step of typical PLA-RB and PLA-SB simulations with $\epsilon = 0.03$. These plots can be cross-referenced with Fig. 4.4, which shows the elapsed time at each simulation step of these runs, and the time course in Fig. 4.1, to understand how the PLA operates and why it outperforms the SSA.

In Fig. 4.4(a), four regimes of system behavior are identified. The first, the initial transient, is magnified in Fig. 4.4(b). We see from this plot that around 1/10th of the total simulation steps, between 1000 and 1400, are spent traversing the first 0.1 time units. Cross-referencing with Fig. 4.3, we see that the classifications for reactions R_1, R_2 and R_3 fluctuate during this period between ES and coarser descriptions, up to Langevin in the case of R_2. R_4, however, remains at the ES level throughout.[2] Regime II is characterized by extensive leaping of all reactions. In an

[2]For the purposes of classification, the only difference between R_3 and R_4 is the value of the

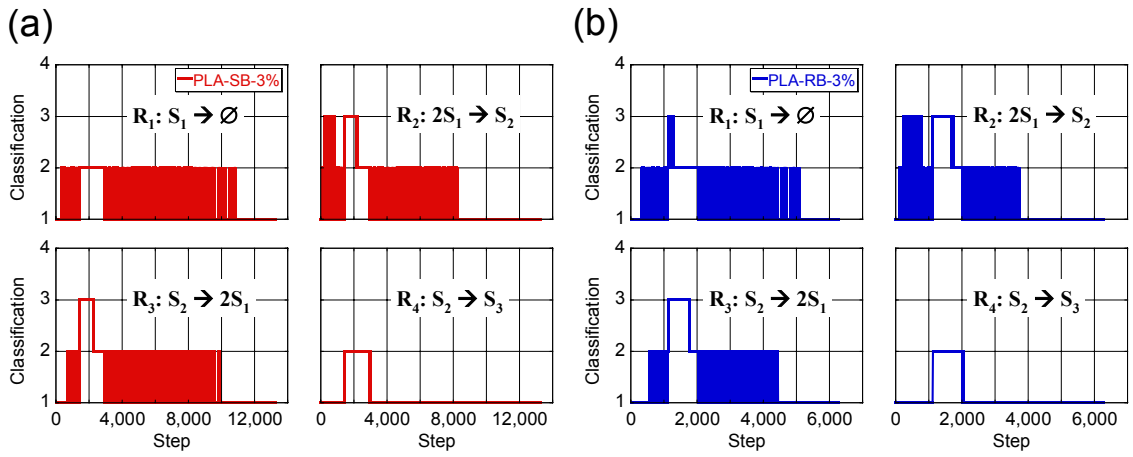

Figure 4.3: Classifications achieved for each reaction of the decaying-dimerizing reaction set at each step of a typical PLA-SB (a) and PLA-RB (b) simulation run with $\epsilon = 0.03$. The classifications are: (1) exact stochastic, (2) Poisson, (3) Langevin, (4) deterministic. The two sets of classifications look quite similar but close inspection reveals slightly coarser classifications in (b), explaining the differences seen between the methods in the timing analysis of Fig. 4.2.

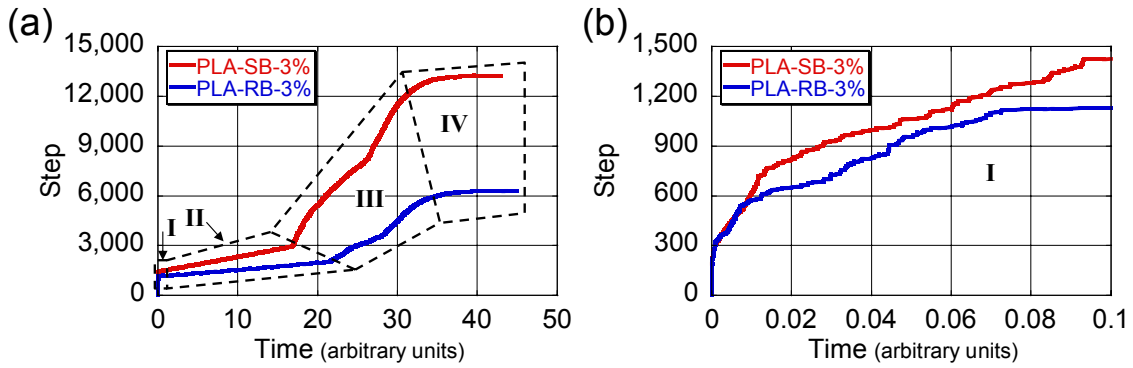

Figure 4.4: Elapsed times at each step of the PLA-SB and PLA-RB simulations of Fig. 4.3. In (a), four regimes of system behavior are identified: the initial transient (I), the large-population regime (II), the transition range (III), and the small-population terminus (IV). A magnified view of regime I is shown in (b) for convenience.

approximately equal number of steps to regime I, we see in Fig. 4.4(a) that about 20 time units are traversed during this period. This is because the populations of S_1 and S_2 reach their highest points (Fig. 4.1), leading to non-fluctuating coarse classifications for all reactions (Fig. 4.3). As the population levels fall, however, the system moves into regime III, characterized by a return to fluctuating classifications and, hence, an increase in the slopes in Fig. 4.4(a). This behavior arises because the populations of S_1 and S_2 fall to low enough levels where not much leaping can take place but are still large enough so that the time intervals between reaction firings are relatively short. Basically, the system is transitioning from conditions where coarse descriptions dominate to those where a fully ES description is necessary. This conflict persists until the reactant populations fall below a few hundred, at which point the conditions call for a fully ES treatment (Fig. 4.3). Thus, in regime IV, the PLA automatically segues into the SSA (NRM variant) and the final 10 time units or so are traversed in only a handful of simulation steps because of the large time intervals between successive reaction firings.

Finally, in Fig. 4.5, we quantify the accuracy of the PLA through comparisons to the SSA. Following Gillespie and Petzold [50], we perform 10 000 simulation runs up until time $t = 12$ using both the PLA and the SSA and compare histograms of the distributions of all three species. We generate smoothed histograms using Eq. (F.3) and quantify their differences using the histogram distance, D, of Eq. (F.4) and the SSA self distance, $D_{\text{SSA}}^{\text{self}}$, of Eq. (F.5). Very simply, if $D < 2 \times \left\langle D_{\text{SSA}}^{\text{self}} \right\rangle$ [Eq. (F.11)], then the PLA histogram cannot be statistically distinguished from the corresponding SSA histogram, indicating maximal accuracy (see Appendix F).

rate constant. Thus, the fact that different classifications are seen for these reactions illustrates how population size alone is not sufficient for determining appropriate levels of description within a reaction network.

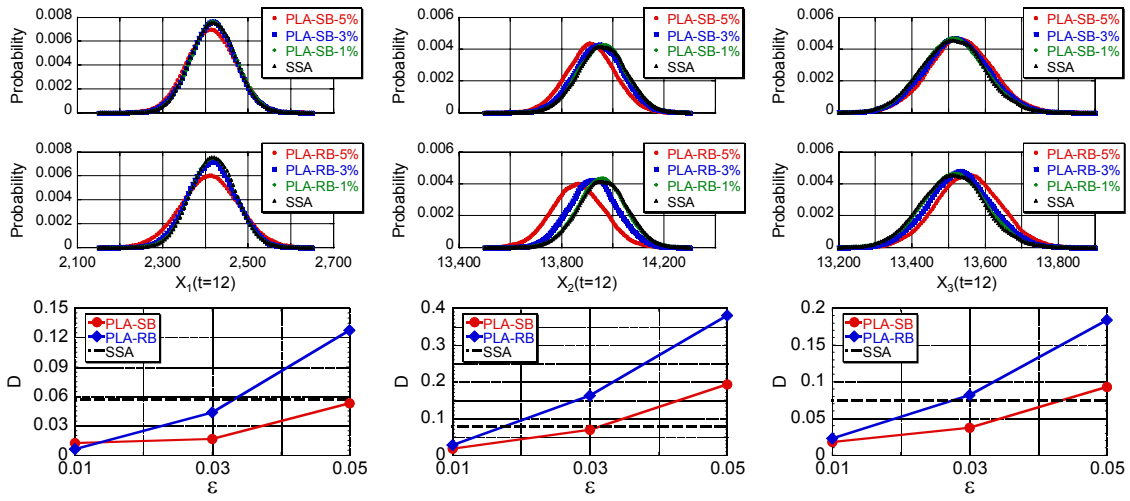

Figure 4.5: Accuracy analysis for simulations of the decaying-dimerizing reaction set. (*top row*): Smoothed histograms for all three species at $t = 12$ obtained from 10 000 simulation runs of the SSA and the PLA-SB with $\epsilon = 0.01$, 0.03 and 0.05. (*middle row*): Same as the top row but using the PLA-RB. (*bottom row*): Histogram distances quantifying the differences between the PLA results and the SSA results. The dashed lines denote *twice* the expected values of the SSA self distances [Eq. (F.11)]. Points that lie below this threshold cannot be statistically distinguished from the corresponding SSA histogram [56].

The results in Fig. 4.5 indicate good accuracy on the part of the PLA, though statistically significant deviations are seen at larger values of ϵ. In general, we see that the PLA-SB results are more accurate than the PLA-RB at given values of ϵ. This is largely as expected, however, given that the SB τ-selection procedure implicitly imposes a tighter constraint on the propensity changes than does the RB procedure (Sec. 3.4.2). It is also interesting to note how deviations from the SSA manifest themselves differently for the different species. For S_1, the distributions broaden but the mean is maintained. Conversely, the distributions for S_2 and S_3 largely maintain their shape but shift to the left for S_2 and to the right for S_3. We also see that deviations seem to be more pronounced for S_2 than for the other

species. All in all, these results indicate that error propagation in the PLA, and leaping algorithms in general, is a non-trivial matter and that a trade-off exists between accuracy and efficiency (cf. Fig. 4.2), as one might expect.

4.1.2 Simple clustering[3]

Clustering phenomena arise in a variety of settings, from defect clustering in materials to droplet coalescence in clouds to chemical polymerization and biological oligomerization. The process of clustering is inherently multiscale: small numbers of large clusters generally coexist within a system of large numbers of small clusters. As such, clustering provides an ideal setting in which to demonstrate the utility of the PLA when applied to systems with large disparities in species populations.

Here, we consider a very simple model of clustering comprised of the following nine reactions:

$$\begin{aligned} R_1: \quad 2S_1 &\xrightarrow{c_1=c} S_2, \\ R_2: \quad S_1 + S_2 &\xrightarrow{c_2=c} S_3, \\ R_3: \quad S_1 + S_3 &\xrightarrow{c_3=c} S_4, \\ &\vdots \\ R_9: \quad S_1 + S_9 &\xrightarrow{c_9=c} S_{10}. \end{aligned} \tag{4.2}$$

For simplicity, we have neglected dissociation reactions and we assume that monomers (S_1) are the only mobile species in the system (i.e., larger clusters cannot interact with each other).

[3]The content of this subsection has been adapted from Ref. [56].

Furthermore, we confine the multiscale effects to variations in the species populations *alone* by choosing equivalent propensity constants for all reactions, i.e., $c_i = c$ for all $i = 1, \ldots, 9$. However, in this example, we want to investigate the performance characteristics of the PLA over a wide range of system sizes. Thus, we choose "deterministic" rate constants, k_i, that are on a per molar basis and then calculate the appropriate propensity constants by dividing by $N_A\Omega$, Avogadro's number times the system volume (see Table B.1 of Appendix B). For R_1, we choose $k_1 = 3 \times 10^6$ M^{-1} s^{-1} and for all other reactions $k_i = 6 \times 10^6$ M^{-1} s^{-1}, $i = 2, \ldots, 9$.[4] We set the initial monomer concentration $[S_1](0) = 1.66 \times 10^{-6}$ M (all other species begin with zero concentration) and consider various system volumes Ω ranging from 10^{-15} to 10^{-9} l. This corresponds to initial monomer populations $X_1(0) = 10^3$ to 10^9 and propensity constants $c = 10^{-2}$ to 10^{-8} s^{-1}. All simulations are run until the consumption of all monomers is complete.

In Fig. 4.6, we show an example time course obtained for a system volume $\Omega = 10^{-9}$ l. In Fig. 4.6(a), we see a steady decline in the monomer population accompanied by offset increases in the populations of clusters up to size 10. Consumption of all monomers is complete by ~ 6 s, at which point the cluster populations vary by as many as four orders of magnitude. In Fig. 4.6(b), the time axis is shown on a log scale so that the short-time behavior of the system can be visualized.

In Fig. 4.7, we compare average numbers of steps and total CPU times required for PLA and SSA simulations of this system at all system sizes considered. In Fig. 4.7(a), we see that the numbers of steps for PLA and SSA simulations are virtually identical at the smallest volumes considered. The utility of the PLA is

[4]Because R_1 involves two entities of the same species, k_1 must be one-half the value of all other rate constants in order for all the c_i to be equivalent (see Table B.1 of Appendix B).

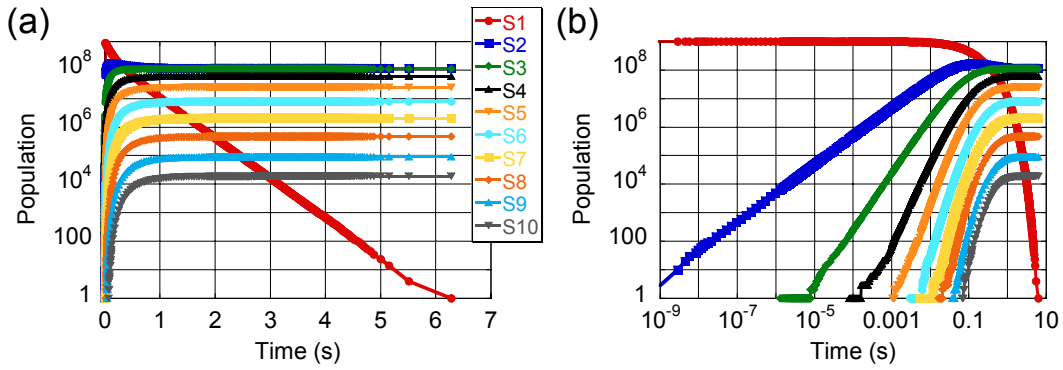

Figure 4.6: Example time course of the simple clustering model for $\Omega = 10^{-9}$ l. In (a), the time course is shown as a semi-log plot, while in (b) both axes are shown on a log scale in order to provide visual access to the short-time behavior of the system.

clearly demonstrated at larger volumes, however. It is also interesting to note how the computational expense of the PLA actually *decreases* for volumes $\gtrsim 10^{-11}$ l.[5] Furthermore, we see that the PLA-SB consistently requires fewer simulation steps than the PLA-RB for given ϵ. This is a curious result and contrary to what was seen in Fig. 4.2 of the preceding subsection. This illustrates that the performance characteristics of the PLA are non-trivial and can be system specific (more on this below). In Fig. 4.7(b), the trends in the CPU times generally coincide with those of the simulation steps. The only exception is the elevated expense of the PLA-RB

[5]We envision two competing effects: (i) system size, which increases the computational load by increasing the population levels, and (ii) leaping, which decreases the computational load by allowing multiple reaction firings at each simulation step. For small systems, leaping effects are minimal but the computational cost in Fig. 4.7 is low because the populations are small. As the system size increases leaping effects increase, but initially they are not significant enough to overcome the effects of system size. Thus, we see an increase in computational effort in Fig. 4.7 up to $\Omega = 10^{-11}$ l, albeit to a lesser extent than for the SSA. Above this point, however, the computational cost begins to decrease. We interpret this as being due to the effects of leaping overtaking those of system size. Of course, this cannot continue indefinitely and, thus, we see a plateauing of the cost at very large volumes.

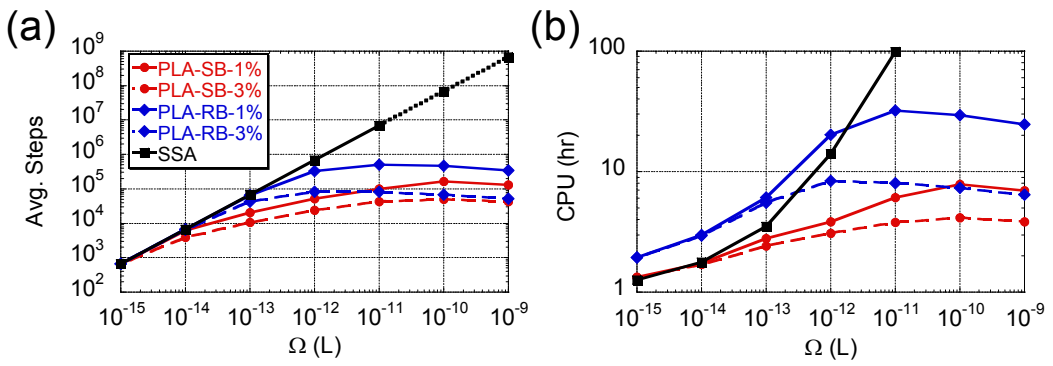

Figure 4.7: Average numbers of steps from (a) and CPU times for (b) 10 000 PLA and SSA simulation runs of the simple clustering model at various system volumes. PLA results are shown for both the RB and SB τ-selection variants with $\epsilon = 0.01$ (solid lines) and 0.03 (dashed lines). Note that in (a) the SSA values at 10^{-10} and 10^{-9} l are extrapolations (not based on actual data). As such, no CPU times are plotted for these volumes in (b). All simulations were run on a 1.80 GHz Athlon processor.

relative to the SSA even at small volumes where the numbers of steps are identical. This indicates a significant computational overhead associated with RB τ selection. We see that the PLA-SB, however, does not suffer from this shortcoming.

In Fig. 4.8, we show the classifications achieved for each reaction of the simple clustering model at each step of a typical PLA-RB-3% (i.e., $\epsilon = 0.03$) simulation for a system size $\Omega = 10^{-9}$ l. These show leaping in action and illustrate the multiscale nature of the network. The reactions involving the smallest cluster sizes (e.g., R_1–R_4) experience extensive amounts of leaping throughout much of the simulation, with the classifications fluctuating between exact stochastic (ES), Poisson, Langevin and, at times, deterministic. Reactions involving larger cluster sizes, however, experience much less leaping because of the smaller populations of these species (Fig. 4.6). Reactions R_7–R_9, for example, experience only small amounts of leaping up into the Langevin regime.

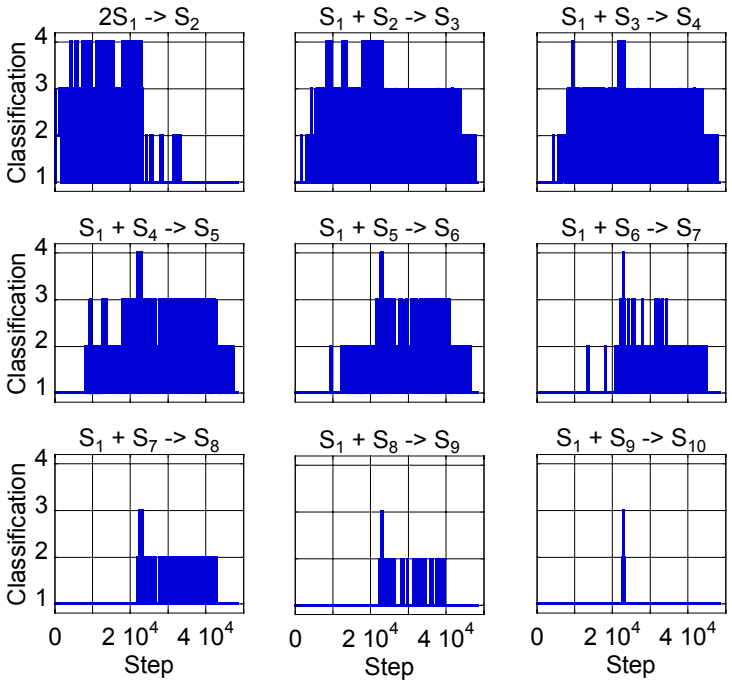

Figure 4.8: Classifications for each reaction of the simple clustering model at each step of a typical PLA-RB-3% simulation at $\Omega = 10^{-9}$ l $[X_1(0) = 10^9]$. Classifications are: (1) exact stochastic, (2) Poisson, (3) Langevin, (4) deterministic.

In Fig. 4.9, we show the elapsed time at each step of the simulation depicted in Fig. 4.8. As before, this allows us to relate the classifications seen in Fig. 4.8 to the time course in Fig. 4.6. In Fig. 4.9(a), we see behavior reminiscent of that in Fig. 4.4 of the preceding subsection: (i) an initial period in which much of the computational effort is spent traversing a small amount of simulated time [a magnified view is shown in Fig. 4.9(b) for convenience], (ii) a period of extensive leaping associated with large species populations, (iii) a period of transition from coarse to finer-level descriptions, and (iv) a small-population period where a large amount of time is traversed in a small number of steps. As before, this is a stark illustration of the capabilities of the PLA. When possible, the algorithm fires multiple reactions simultaneously in order to accelerate the simulation. It maintains accuracy, however, by doing so only when the conditions permit.

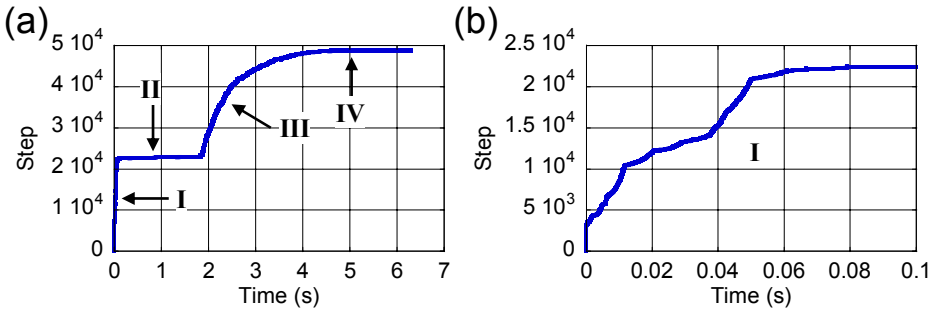

Figure 4.9: Elapsed time at each step of the PLA-RB simulation in Fig. 4.8. Similar to Fig. 4.4, we see four regimes of system behavior (*see text*). A magnified view of the initial regime is shown in (b) for convenience.

This accuracy is illustrated in Fig. 4.10, where we plot histogram distances for the final populations (i.e., at time $t = t_f$) of selected cluster sizes (S_2, S_5, S_8 and S_{10}) for a system volume $\Omega = 10^{-9}$ l. These results are assumed to be representative of the full spectrum of cluster sizes at all volumes considered. We see from these plots that the PLA achieves good accuracy, though statistically significant deviations are seen in some cases, particularly for $\epsilon = 0.03$. Interestingly, we also see that the PLA-RB is consistently more accurate than the PLA-SB. This is contrary to what was seen previously (Fig. 4.5) but is consistent with the smaller time steps [i.e, larger number of steps—Fig. 4.7(a)] realized by the PLA-RB in this case.

4.1.3 Stochastic gene expression[6]

The role and consequences of stochasticity in biological systems is a subject of great current interest [2, 34, 37, 61, 73, 74, 89, 90]. In cellular systems, the primary source of "intrinsic" stochastic noise is gene expression, where the small numbers of regulatory molecules involved in the process result in proteins being produced in

[6]The content of this subsection has been adapted from Ref. [56].

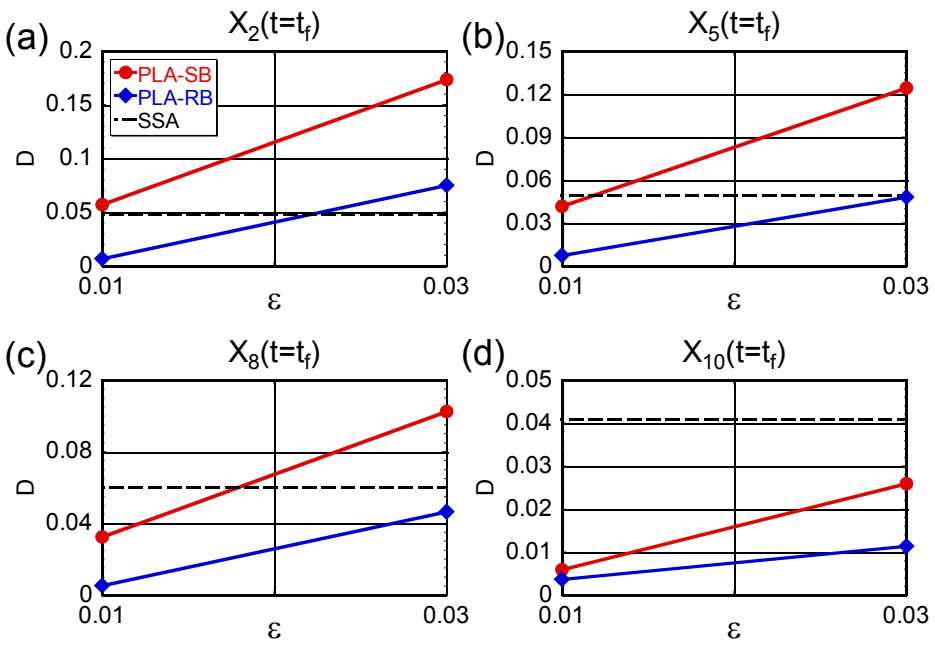

Figure 4.10: Histogram distances comparing the results of PLA simulations of the simple clustering model to those of SSA simulations. Results are shown for the final populations (at $t = t_f$) of selected cluster sizes for $\Omega = 10^{-9}$ l. Dashed lines denote twice the expected values of the SSA self distances [Eq. (F.11)]. All histograms were obtained from 10 000 PLA or SSA simulation runs.

"bursts" rather than continuously [61, 73, 74]. Other cellular processes, however, such as metabolism, often involve large numbers of molecules and, as such, are commonly treated deterministically. Nevertheless, it has been shown that stochastic fluctuations in gene expression can quantitatively affect these dynamics [87]. In principle, therefore, they cannot not be considered independently of each other. However, fully stochastic treatments of biological systems containing both gene expression dynamics and metabolic processes are generally infeasible [35]. There is great motivation, therefore, to develop multiscale simulation methods (such as the PLA) that are capable of handling systems containing both large- and small-number dynamics.

Here, we apply the PLA to a simple biologically-inspired model system that contains a crude description of gene expression along with protein-protein interactions. The network is as follows:

$$\begin{aligned} R_1: &\quad G &\xrightarrow{c_1}& \quad G^*, \\ R_2: &\quad G^* &\xrightarrow{c_2}& \quad G + nP, \\ R_3: &\quad P + Q &\xrightarrow{c_3}& \quad \{P{\cdot}Q\}, \\ R_4: &\quad \{P{\cdot}Q\} &\xrightarrow{c_4}& \quad R + Q, \\ R_5: &\quad R &\xrightarrow{c_5}& \quad \emptyset. \end{aligned} \tag{4.3}$$

The first two reactions constitute the gene expression part of the network, where the single gene G spontaneously converts into an active conformation G^* that produces proteins P in bursts of n. The third and fourth reactions constitute the protein-protein enzymatic part of the network where P interacts with Q to form an enzyme-substrate complex $\{P{\cdot}Q\}$ that subsequently produces R and reconstitutes Q. The final reaction models the degradation of R.

Rate constants for the five reactions are chosen as: $k_1 = k_2 = 750$ s^{-1}, $k_3 = 6{\times}10^8$ M^{-1} s^{-1}, $k_4 = 100$ s^{-1} and $k_5 = 50$ s^{-1}.[7] We set the initial enzyme concentration $[Q](0) = 1.66{\times}10^{-7}$ M and define the burst parameter $n = 0.2X_Q(0)$, where $X_Q(0)$ is the initial *population* of Q.[8] Investigations are carried out for system sizes ranging from 10^{-15} to 10^{-7} l, corresponding to initial enzyme populations $X_Q(0)$ ranging from 10^2 to 10^{10}. In all cases, the system begins with a single entity of G and null populations of G^*, $\{P{\cdot}Q\}$ and R. All simulations are

[7] For the first-order reactions we have $c_i = k_i$, $i = 1, 2, 4, 5$. For the second-order reaction we have $c_3 = k_3/N_A\Omega$ (see Table B.1).

[8] Our reason for doing this is primarily for convenience, to exemplify the utility of the PLA. Nevertheless, by allowing the number of proteins produced per expression event to change we are effectively varying the degree of "translational efficiency" [61].

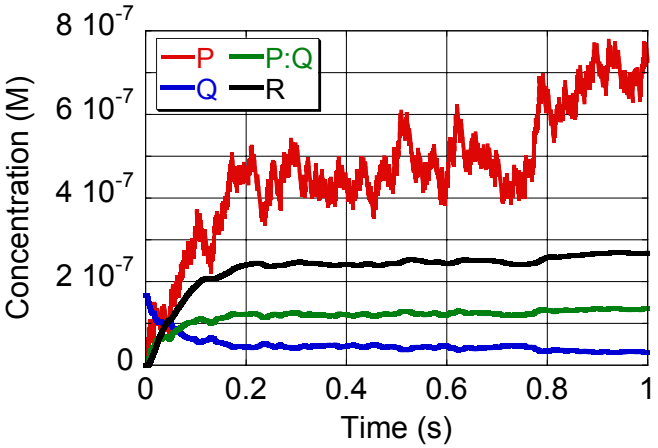

Figure 4.11: Example time course of the simple gene expression system for a system volume $\Omega = 10^{-11}$ l. Despite having a larger population, the dynamics of P are significantly noisier than those for Q, $\{P{\cdot}Q\}$ and R. This is because the dynamics of P are intimately linked to those of G, of which there is but a single copy.

run until $t = 1$ s.

In Fig. 4.11, we show an example time course for this system obtained for a system volume $\Omega = 10^{-11}$ l. This plot illustrates the stochastic nature of the gene expression dynamics, apparent in the noisy time evolution of the gene product P. Smoother trajectories are seen for the species Q, $\{P{:}Q\}$ and R which, interestingly, have smaller populations than P. This is a clear illustration, therefore, of the shortcoming of using the population size alone as a metric for predicting levels of stochasticity. Though the population of P is large in this case, the dynamics are noisier than might be expected because the behavior of P is intimately tied to that of the gene G, of which there is but a single copy. Thus, the dynamics of P are, in fact, small-number in nature.

In Fig. 4.12, we present results of a step and timing analysis comparing the performance of the PLA to the SSA. Here, as before, we see that the PLA achieves significant computational savings relative to the SSA, particularly at large volumes.

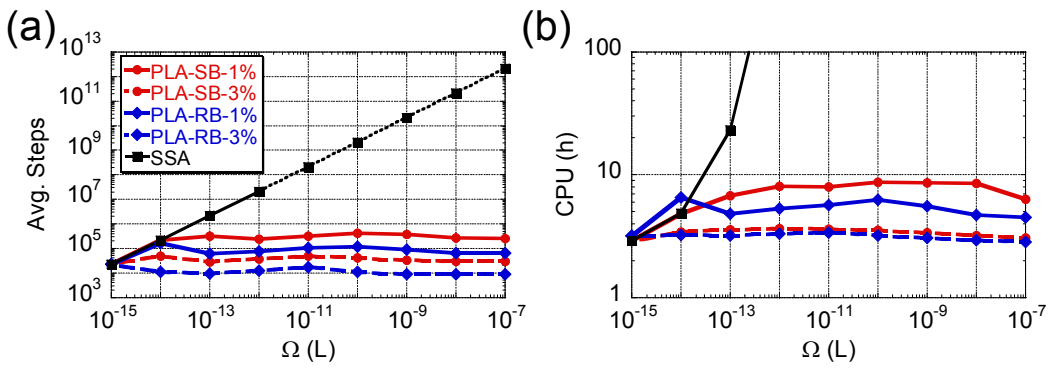

Figure 4.12: Average numbers of steps from (a) and CPU times for (b) 10 000 PLA and SSA simulation runs of the simple gene expression model at various system volumes. PLA results are shown for both the RB and SB τ-selection variants with $\epsilon = 0.01$ (solid lines) and 0.03 (dashed lines). Note that in (a) the SSA values for $\Omega > 10^{-12}$ l are extrapolations (not based on actual data). As such, no CPU times are plotted for these volumes in (b). All simulations were run on a 1.80 GHz Athlon processor.

We also see that the PLA-RB consistently requires fewer steps and less CPU time than does the PLA-SB for given ϵ. This is in direct contrast to the results of the preceding subsection (Fig. 4.7) though generally consistent with those of Sec. 4.1.1 (Fig. 4.2), except for the CPU times. Indeed, in the course of three simple example systems, we have now seen cases where the PLA-RB requires: (i) fewer steps than the PLA-SB but an equivalent amount of CPU time, (ii) more steps and more CPU time, and (iii) fewer steps and less CPU time. This system-specific behavior is particularly intriguing and is a subject deserving of further inquiry in the future.

In Fig. 4.13, we take a closer look at the step requirements of the PLA when applied to this system. Here, we plot not only the average numbers of steps required for different variants of the PLA but also the "envelopes" within which 95% of all the data points lie. Doing so reveals some interesting behavior that is not apparent in Fig. 4.12. Specifically, we see in Fig. 4.13 that the distributions in the numbers of steps differ greatly with system size. The lower bound on the

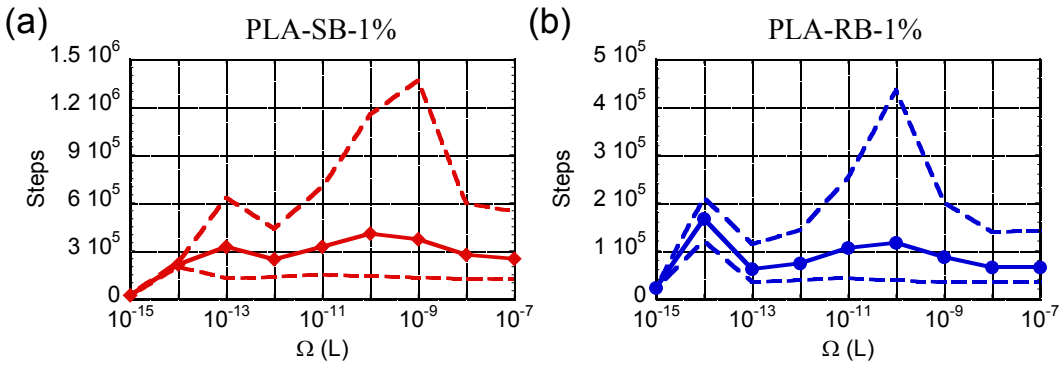

Figure 4.13: Averages (solid lines) and 95% envelopes (dashed lines) on the numbers of simulation steps required for PLA-SB-1% (a) and PLA-RB-1% (b) simulations at various values of Ω. For $10^{-11} \leq \Omega \leq 10^{-9}$ l, the stochastic nature of the gene expression dynamics results in a significant portion of simulations requiring a large number of steps.

distributions remains essentially constant for $\Omega \geq 10^{-13}$ l but the upper bound varies dramatically, reaching its largest value at around 10^{-10} l. Basically, the stochastic nature of the gene expression dynamics gives rise to a wide variety of time-evolution trajectories, particularly for $10^{-11} \leq \Omega \leq 10^{-9}$ l, a portion of which require many more simulation steps to complete than others. The ability of the PLA to adapt and correctly capture the dynamics of such a "variable fate" system is a particular strength of the method. Bistable switching and other forms of variable-fate decision making are common to many biological systems and understanding the role that stochasticity plays in this process is a subject of great current interest [2, 26, 58, 89].

To illustrate this adaptive capability of the PLA, we present in Fig. 4.14 classification profiles for reaction R_3 ($P+Q \rightarrow \{P \cdot Q\}$) obtained from typical PLA-RB-3% simulations at three different system sizes.[9] At $\Omega = 10^{-14}$ l, we see that the classi-

[9]Similar behavior is seen for reactions R_4 and R_5 while reactions R_1 and R_2 are always classified as ES because the populations of G and G^* can obviously never exceed unity (data not

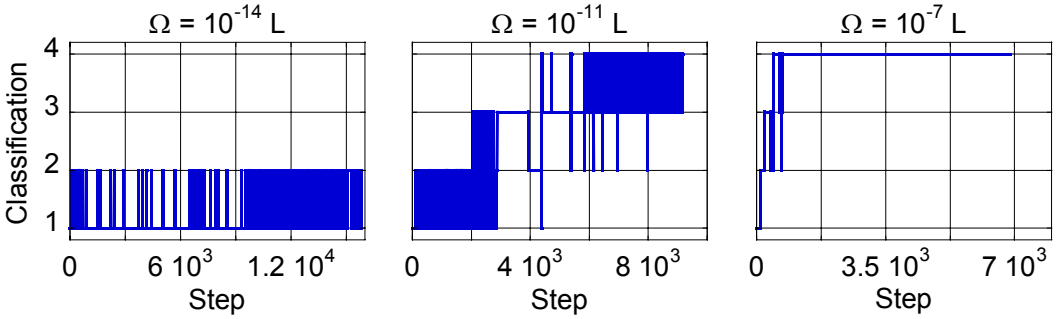

Figure 4.14: Classifications achieved for reaction R_3 of the simple gene expression model during typical PLA-RB-3% simulation runs at three different system volumes. The abrupt switch from fine to coarse classifications seen at $\Omega = 10^{-11}$ l is the cause of the high variability in numbers of simulations steps seen in Fig. 4.13 at similar volumes.

fications never reach beyond the Poisson level. At $\Omega = 10^{-11}$ l, we see a coarsening of the classifications approximately midway through the simulation. This occurs because of the firing of multiple successive gene expression bursts in this particular run. It is this type of behavior that leads to the large variability in the numbers of steps seen in Fig. 4.13 at similar volumes. Finally, at $\Omega = 10^{-7}$ l, deterministic status is achieved quickly and maintained almost exclusively throughout. This explains the narrower distributions seen at large volumes in Fig. 4.13.

Finally, in Fig. 4.15, we quantify the accuracy of the PLA via histogram distances for the species P, Q, $\{P{\cdot}Q\}$ and R obtained at $t = 1$ s for a system volume $\Omega = 10^{-13}$ l. Again, we assume these results to be indicative of all system sizes considered. In this case, we see exceptional accuracy, with all histogram distances falling well below the SSA self-distance thresholds.

shown).

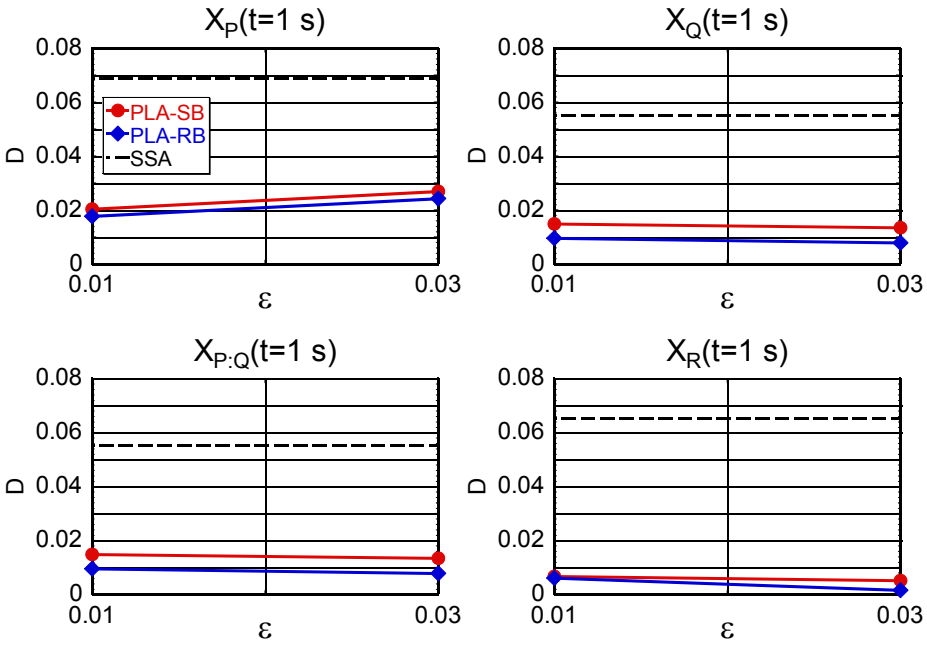

Figure 4.15: Histogram distances comparing the results of PLA simulations of the simple gene expression model to those of SSA simulations. Results are shown for the populations of P, Q, $\{P{\cdot}Q\}$ and R at $t = 1$ s for $\Omega = 10^{-13}$ l. Dashed lines denote twice the expected values of the SSA self distances [Eq. (F.11)]. All histograms were obtained from 10 000 PLA or SSA simulation runs.

4.2 Prototypical biochemical networks[10]

In this section, we use the PLA to systematically investigate the effects of stochasticity in two model biochemical reaction networks. The systems that we consider are a core model for calcium oscillations in hepatocytes (liver cells) introduced by Kummer et al. [66] (Sec. 4.2.2) and the three-gene "repressilator" of Elowitz and Leibler [33] (Sec. 4.2.3). These systems are relatively simple, yet they differ from those of Sec. 4.1 in that they are not "toy" problems. They contain non-trivial features that are ubiquitous to biochemical systems, such as enzyme catalysis and feedback control. Moreover, both systems emit large-amplitude oscillations which

[10]The content of this section has been adapted from Ref. [57].

give rise to the kinds of wide disparities in species populations that leaping algorithms are specifically designed to cope with [47, 56]. All in all, these networks provide an ideal testbed for investigating the practical utility of the PLA in computational systems biology.

Our investigation involves using the PLA to probe behavioral changes that arise in these systems due to changes in various system properties. In particular, we investigate the transition from stochastic to deterministic behavior that accompanies increases in the system volume in the calcium-oscillations model and increases in the gene-protein binding and unbinding rate constants in the repressilator. The salient feature of our investigation is that we are able to ascertain, in a systematic way, the performance characteristics of the PLA over a wide spectrum of conditions. Thus, we identify cases where leaping proves particularly beneficial, where it "bogs down," and various points in between.

We begin in Sec. 4.2.1 by discussing the details of our investigation, including the particulars of the PLA implementation, the time series analysis tool and the statistical tests employed in this work. We then present in Secs. 4.2.2 and 4.2.3 the models and results for the calcium-oscillations and repressilator systems, respectively. We conclude in Sec. 4.2.4 with a discussion of these results and their implications for future applications of the PLA.

4.2.1 Computational details

As previously, all PLA simulations reported in this section were performed with the parameters '≈ 1'$= 3$ and '$\gg 1$'$= 100$ and utilizing both the RB and SB τ-selection formulas of Eqs. (3.27) and (3.30), respectively. In the case of SB τ-selection,

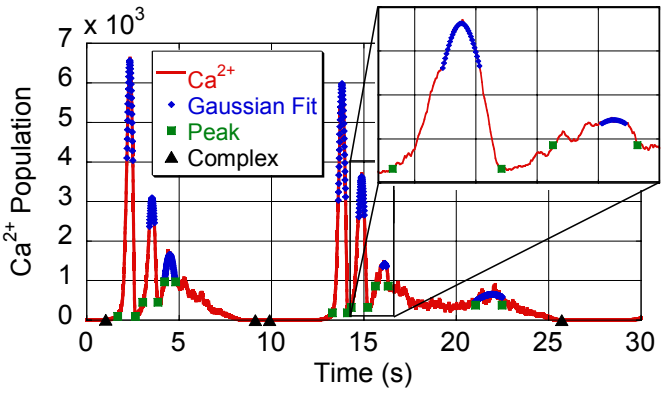

Figure 4.16: Example calcium-oscillations time course and the Gaussian fits obtained using the peak-analysis software employed in this work. Results are for a system volume $\Omega = 10^{-21}$ l. Also shown are peak and peak-complex bracket points identified by the fitting algorithm. Notice that at this small volume stochastic effects lead to the identification of a fourth peak in the second peak complex. (*Inset*) Blown-up view of the second and third peaks in the second peak complex. Squares correspond to where fitting began, diamonds to where fitting concluded.

derivations of the g_i values used for the Michaelis-Menten and Adair reactions of Tables 4.1 and 4.3, respectively, are given in Appendix D.

In order to account for the noisy time-evolution trajectories generated by the PLA, we use in-house time-domain peak-analysis software. Borrowing ideas from the automated identification of peaks in mass spectral data [62, 115], the software identifies "significant" peaks within a time series and fits Gaussians to the data in order to wash out the noise. An example calcium-oscillations time series and the Gaussian fits achieved using the peak-analysis software are shown in Fig. 4.16.

Using this tool, we collect large amounts of peak amplitude and peak-to-peak distance data from simulated time series and perform various statistical analyses. We calculate averages and variances from long-time PLA and deterministic simu-

lation runs[11] and perform z-tests on the differences in means and F-tests on the ratios of variances [77]. We also calculate coefficients of variation (COVs), defined as the ratio of the standard deviation to the mean [61], in order to quantify the relative *importance* of the noise. Finally, we put the data into the form of smoothed histograms and calculate histogram distances, D, and self distances, $D_{\text{Ref}}^{\text{self}}$ [23, 56] (Appendix F), so as to account for any particulars in the *shapes* of the distributions (e.g., long tails, bimodal features, etc.). We do all of this for various system properties (i.e., volumes, telegraph factors) in order to quantify changes in the system behavior and identify points of transition to determinism.

4.2.2 Calcium oscillations

Intracellular calcium is an important second messenger for the functioning of many cell types, both in plants and in animals. It is involved in a multitude of functions during the lifetime of a cell, including fertilization, development and death [7]. The dynamics of intracellular calcium are not smooth and continuous, however. Rather, they are driven by small numbers of receptors and ion channels that can give rise to highly stochastic behavior. Indeed, experiments have shown that calcium waves are triggered by elementary stochastic events known as "blips" and "puffs" [36]. Incorporating stochasticity into models of calcium oscillations is thus of high interest.

[11] Obviously, deterministic simulations should exhibit zero variance in their results. However, due to sampling and curve-fitting inaccuracies we do see slight variations. It is these variations that we use as the criteria for determining when a system attribute has converged to the deterministic limit. Clearly, if the PLA results show equal or less variation than the deterministic results then we can deem that the property has converged to determinism.

The network

Many theoretical models have been proposed to describe the oscillatory dynamics of intracellular calcium [36, 102]. Kummer et al. [66] proposed a model for calcium oscillations in hepatocytes (liver cells) that displays a rich variety of behaviors. The model features self-enhanced activation of the G_α subunit of the receptor complex and is able to capture many aspects of experimentally-observed behavior that eluded previous models. The authors also presented a simplified version of the model that displays the same basic behaviors as the full model, thus emphasizing the "core" mechanisms driving the oscillations [66].

In Table 4.1, we show the Kummer et al. [66] core model for calcium oscillations in hepatocytes. Note that the model is in a reduced form, with degradation processes described in terms of Michaelis-Menten kinetics. Reaction 2, which is the prime feature of this model, describes the agonist-initiated (e.g., ATP) autocatalytic activation of the G_α subunit. The parameter k_2 thus amounts to the product of the second-order association constant and the agonist concentration and is a primary determinant of the system behavior. Kummer et al. showed that with increasing k_2 the system behavior transitions from simple Ca^{2+} spiking oscillations, to complex oscillations, to chaotic behavior and, finally, to an elevated steady state [65, 66].

We also see in Table 4.1 that the model contains various feedback loops which drive the oscillatory behavior of the network. Specifically, PLC^* and Ca are created autocatalytically in reactions 5 and 7, respectively, through the action of G_α. In reactions 3 and 4, however, G_α is degraded enzymatically by the actions of PLC^* and Ca, respectively. Thus, in the correct parameter range, increased levels of G_α lead to increased levels of PLC^* and Ca which, in turn, lead to increased

Table 4.1: Kummer et al. [66] core model for calcium oscillations in hepatocytes. '$\emptyset$' represents a source or a sink and $k_2 = 2.85$ s^{-1} puts the system into the "periodic-bursting" regime. G_α represents the activated α-subunit of the intracellular receptor-bound G-protein, PLC^* the activated form of phospholipase C, and Ca cytosolic calcium ions. Note that to perform stochastic simulations all parameters must be devoid of molar units (M). Parameters with molar units are thus multiplied by $N_A\Omega$ (Avogadro's number $\times$ system volume) prior to runtime.

Reaction	Rate Expression	Parameter Value(s)
1. $\emptyset \rightarrow G_\alpha$	k_1	$k_1 = 0.212$ M s^{-1}
2. $G_\alpha \rightarrow 2G_\alpha$	$k_2[G_\alpha]$	$k_2 = 2.85$ s^{-1}
3. $G_\alpha + PLC^* \rightarrow PLC^*$	$k_3[G_\alpha][PLC^*]/(K_4 + [G_\alpha])$	$k_3 = 1.52$ s^{-1}, $K_4 = 0.19$ M
4. $G_\alpha + Ca \rightarrow Ca$	$k_5[G_\alpha][Ca]/(K_6 + [G_\alpha])$	$k_5 = 4.88$ s^{-1}, $K_6 = 1.18$ M
5. $G_\alpha \rightarrow G_\alpha + PLC^*$	$k_7[G_\alpha]$	$k_7 = 1.24$ s^{-1}
6. $PLC^* \rightarrow \emptyset$	$k_8[PLC^*]/(K_9 + [PLC^*])$	$k_8 = 32.24$ M s^{-1}, $K_9 = 29.09$ M
7. $G_\alpha \rightarrow G_\alpha + Ca$	$k_{10}[G_\alpha]$	$k_{10} = 13.58$ s^{-1}
8. $Ca \rightarrow \emptyset$	$k_{11}[Ca]/(K_{12} + [Ca])$	$k_{11} = 153.0$ M s^{-1}, $K_{12} = 0.16$ M

Initial conditions: $[G_\alpha] = [PLC^*] = [Ca] = 0.01$ M

degradation of G_α, which leads to decreased levels of PLC^* and Ca, and so on and so forth.

In Ref. [65], Kummer et al. compared the deterministic behavior of this model to results of stochastic simulations performed using the SSA. The goal was to determine points of transition to determinism for various dynamical regimes of the model (e.g., "periodic spiking," "periodic bursting," "chaos") and to provide general insight as to when a deterministic treatment is applicable and when a stochastic approach is necessary. SSA simulations were performed at various system sizes (with fixed concentrations) and the point of transition to determinism was estimated via *visual* comparison of stochastic and deterministic time courses. Visual inspection was necessary because of the high computational expense of the SSA [65].

Here, we extend the analysis of Kummer et al. [65] for the "periodic-bursting" regime, a main focus of Ref. [65]. The regime is characterized by complex Ca^{2+} oscillations comprised of three-peak complexes (see below), behavior that is reminiscent of that seen experimentally in hepatocytes stimulated by ATP [29, 65, 66]. Using the PLA and the peak-analysis tool described in Sec. 4.2.1, we collect large amounts of peak amplitude and peak-to-peak distance data at various system volumes and *quantify* the relationship between stochasticity and system size, something that was not feasible in Ref. [65] because of the limitations of the SSA. This allows us to pinpoint, from a statistical perspective, the points of transition to determinism. As we shall see, these differ, to some extent, from those reported in [65].

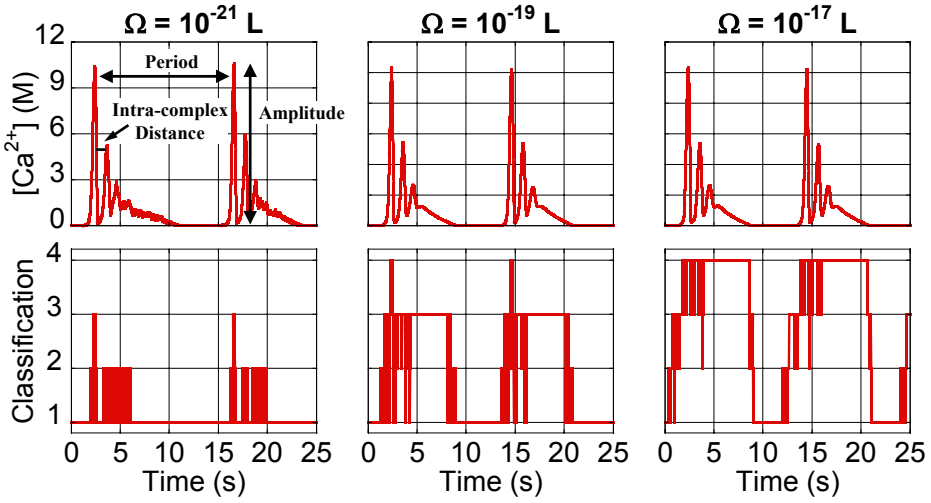

Figure 4.17: Example periodic-bursting Ca^{2+} time courses and associated classifications for $G_\alpha + Ca \rightarrow Ca$ (Table 4.1, reaction 4) obtained using the PLA-SB with $\epsilon = 0.03$ at three different system volumes. Classifications are: (1) exact stochastic, (2) Poisson, (3) Langevin, (4) deterministic. Also shown (*top-left panel*) are the three system attributes investigated: First-peak amplitudes, first-to-second intra-complex distances and first-to-first inter-complex periods.

Statistical analysis

The periodic-bursting regime of the Kummer et al. [66] calcium-oscillations model (Table 4.1) is characterized by large-amplitude complex oscillations in which the Ca^{2+} repeating unit is a three-peak complex. In Fig. 4.17, we show example time courses at three different system volumes spanning four orders of magnitude obtained using the PLA. Also shown in Fig. 4.17 are the classifications achieved along the time courses for the reaction $G_\alpha + Ca \rightarrow Ca$ (Table 4.1, reaction 4). The classifications range from 1–4, with 1 being the finest level of description (exact stochastic) and 4 the coarsest (deterministic).

The plots in Fig. 4.17 starkly illustrate why this system is ideally suited for treatment via the PLA: *The classifications oscillate in time along with the reactant*

species populations. When the Ca^{2+} population is small we see that the reaction is classified at the exact-stochastic level, while coarser descriptions are employed when the population is large (similar behavior is seen for other reactions in the system as well—data not shown). As a result, the PLA is able to accurately capture stochastic effects that arise in this system when the species populations become small without suffering from the characteristic inefficiency of the SSA when the populations become large.

This is evident in Fig. 4.18, where we show results of a step and timing analysis comparing the performance of the PLA (SB variant with $\epsilon = 0.03$) to the SSA. As expected, we see a linear increase in the computational expense of the SSA with increasing system size (see Fig. 4.18, caption) [41, 42, 49]. The PLA, on the other hand, exhibits more complex behavior, with the expense initially remaining constant, then increasing slightly, going through a maximum at $\sim\Omega = 10^{-18}$ l and then dropping off sharply before finally leveling off. Interestingly, this behavior is similar to that seen for the simple example systems of Secs. 4.1.2 (Fig. 4.7) and 4.1.3 (Fig. 4.12), which were specifically constructed to showcase the strengths of the algorithm. Most importantly, however, is that Fig. 4.18 clearly illustrates that for all but the smallest system size considered the PLA *far* outperforms the SSA, by as many as eight orders of magnitude in simulation steps at $\Omega = 10^{-15}$ l. It is these types of accelerations that make quantifying stochastic effects in this system possible, something that was unachievable in Ref. [65] because of the limitations of the SSA.

Our statistical results are shown in Fig. 4.19. In all cases, we compare results obtained from both PLA and SSA simulations to deterministic predictions for the three attributes considered, namely, first-peak amplitudes, first-to-second

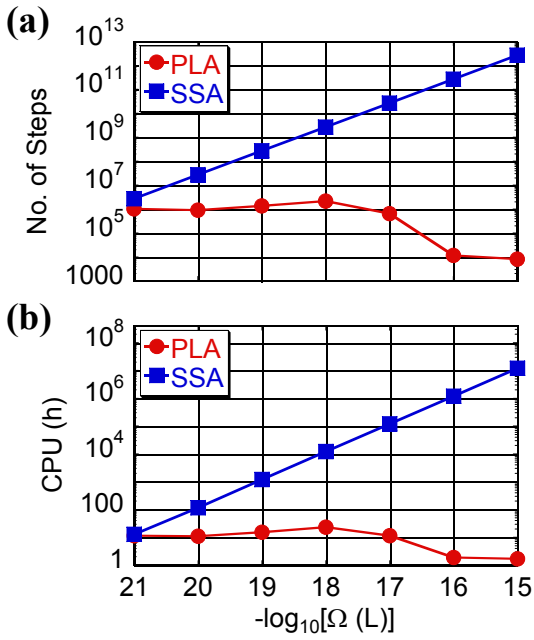

Figure 4.18: Average numbers of steps from (*a*) and total CPU times for (*b*) 10 000 PLA-SB-3% and SSA simulation runs of 20 s for the Kummer et al. [66] core model for calcium oscillations (Table 4.1). SSA values at $\Omega = 10^{-20}$ and 10^{-19} l are based on 1000 and 100 simulations runs, respectively. SSA values at $\Omega \geq 10^{-18}$ l are extrapolations (not based on actual data). Note that the PLA steps and CPU times go through maxima at $\sim \Omega = 10^{-18}$ l. Similar behavior was observed for the example systems of Secs. 4.1.1–4.1.3. Also note that in the case of the SSA, the linear relationship between computational expense and system size [41, 42, 49], which has the form $y = mx$, with m being the slope (the y-intercept is zero since, obviously, a system of zero size requires zero computational effort), appears here as a line with a slope of unity and y-intercept of $\log_{10}(m)$. All simulations were performed on a 3.60 GHz Pentium Xeon processor.

intra-complex distances, and first-to-first inter-complex periods (see Fig. 4.17, *top-left panel*). In the case of the SSA, we were only able to obtain data for the three smallest system sizes considered because of the computational expense of the method.

In Figs. 4.19*a*–4.19*c*, we compare averages and modes obtained from the PLA and SSA to deterministic predictions. The results are shown as percent deviations from determinism. In all cases, we see small yet statistically significant deviations from determinism at small volumes and, in the case of the PLA, a rapid convergence to the deterministic limit with increasing system size. Close inspection reveals that full convergence is achieved for all attributes by $\Omega = 10^{-18}$ l. It is also clear in Figs. 4.19*a* and 4.19*b* that there are discrepancies between the PLA results and the SSA results. The discrepancies are small, however, on the order of 1% or less in all cases, and decrease with decreasing ϵ (data not shown). Interestingly, there are virtually no discrepancies between the PLA results and the SSA results in Fig. 4.19*c*, the inter-complex periods. We cannot at present explain why the PLA achieves greater accuracy for this attribute over the others. Understanding the sources of error in the algorithm and attenuating them is an area of current interest. Suffice it to say for now that the PLA achieves very good to excellent accuracy for these quantities.

In Figs. 4.19*d* and 4.19*e*, we consider the *distributions* of the attributes. Figure 4.19*d* shows data for standard deviations, a simple point statistic, while Fig. 4.19*e* considers the shapes of the distributions through the histogram distance [23, 56] (Appendix F). In Fig. 4.19*d*, we see almost perfect correspondence between the PLA and the SSA results. In Fig. 4.19*e*, however, we see discrepancies in the histogram distances for the amplitude and the intra-complex distance (see

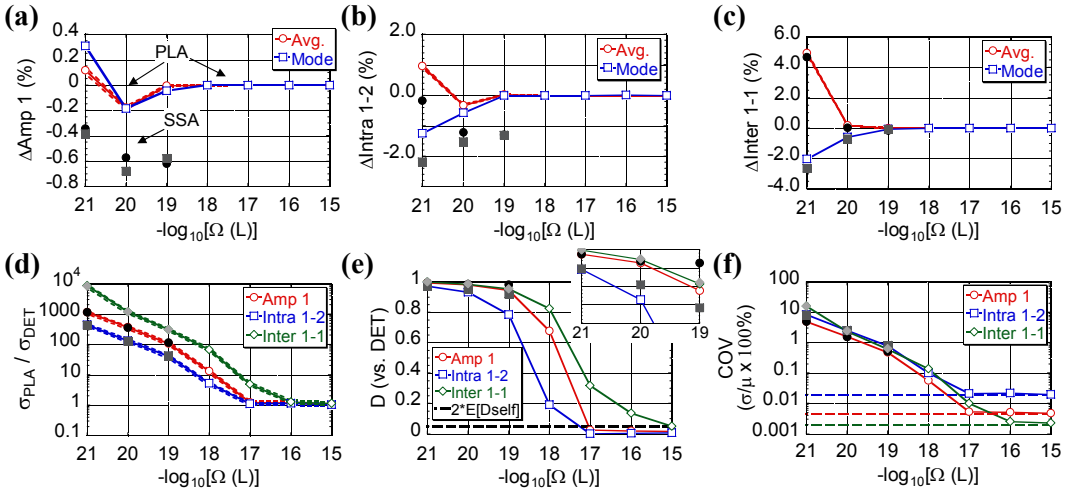

Figure 4.19: Statistical results for the Ca^{2+} periodic-bursting regime. Results of PLA and SSA simulations are compared to deterministic (DET) predictions. PLA results are shown as colored symbols (circle, square, diamond) connected by lines. SSA results are shown as disconnected symbols in shades of grey. PLA and SSA points designated with the same symbol correspond to the same quantity. All PLA and deterministic values are based on over 10 000 collected data points. For the SSA, over 10 000 data points were collected for $\Omega = 10^{-21}$ and 10^{-20} l and ~ 1000 were collected for $\Omega = 10^{-19}$ l. No SSA results are given for $\Omega \geq 10^{-18}$ l due to computational expense. (*a*)–(*c*): Deviations from determinism, shown as percentages [({PLA or SSA}−DET)÷DET×100%], for averages and modes of Ca^{2+} first-peak amplitudes (Amp 1), first-to-second intra-complex distances (Intra 1-2), and first-to-first inter-complex periods (Inter 1-1), respectively (see Fig. 4.17, *top-left panel*). Dashed lines denote 95% confidence intervals on the PLA averages [difficult to see in (b) and (c)]. Note that long-tailed distributions lead to averages and modes on opposite sides of the deterministic predictions at small volumes in (b) and (c). (*d*): Ratios of standard deviations ({PLA or SSA}÷DET) for the three attributes in (a)–(c). Dashed lines denote 80% confidence intervals (because of the relative weakness of the F-test [77]). (*e*): Histogram distances ({PLA or SSA} vs. DET) for the three attributes in (a)–(c). The dashed line denotes *twice* the deterministic self distance ($2 \times \left\langle D_{\mathrm{Det}}^{\mathrm{self}} \right\rangle$—Appendix F). The self distances for all three attributes are essentially identical in this case. (*f*): Coefficients of variation (COVs) obtained from PLA and SSA simulations, shown as percentages (standard deviation÷average×100%), for the three attributes in (a)–(c). Deterministic limits are given as dashed lines.

inset). Taken together along with Figs. 4.19*a* and 4.19*b*, this indicates that the PLA is accurately capturing the shapes of the distributions but they are shifted slightly relative to those obtained with the SSA.

As far as convergence to determinism, both Figs. 4.19*d* and 4.19*e* give the same result: *the different attributes converge to the deterministic limit at different rates and with different transition points*. The intra-complex distance converges the fastest, followed by the peak amplitude and finally the inter-complex period. The amplitude and intra-complex distance statistically converge to the deterministic limit at $\Omega = 10^{-17}$ l while the period converges at 10^{-15} l. These convergence points differ from those for the averages by one to three orders of magnitude (cf. Figs. 4.19*a*–4.19*c*) and indicate a persistence of noise in this system at volumes much larger than expected based on the analysis of Ref. [65].

Finally, in Fig. 4.19*f* we consider the relative "importance" of the noise through the coefficient of variation (COV). The idea is that even if noise in an attribute is significant from a statistical perspective it might be so subtle as to be of little practical import. For example, in this case we see that for $\Omega \geq 10^{-20}$ l the COVs for all attributes are less than a few percent (the discrepancies between the PLA and the SSA seen in Figs. 4.19*a* and 4.19*b* are virtually indiscernible on this scale). The noise effects clearly persist up until 10^{-17} l (as seen in Figs. 4.19*d* and 4.19*e* as well) but it seems unlikely that in any realistic setting, e.g, an embedding within a larger "whole-cell" model, they would be of much practical consequence. Whether or not this is true (it is debatable [99, 100]), it is certainly the case that it would be difficult, if not impossible, to perceive these effects visually. This explains, therefore, why Kummer et al. [65] reported the stochastic-to-deterministic transition point for this model to be at $\sim 10^{-20}$ l (tens of thousands of Ca^{2+} ions).

Their claim that a deterministic treatment is justified for volumes larger than this is thus largely corroborated by our results.

RB *vs.* SB τ selection

In Fig. 4.20, we expand the step analysis of Fig. 4.18*a* in order to compare the performance of the PLA-SB when applied to this system to that of the PLA-RB. Our reason for doing so is to illustrate a curious occurrence that takes place at large volumes. For $\Omega \leq 10^{-17}$ l, we see in Fig. 4.20 the behavior that we expect: the PLA-SB requires more steps than the PLA-RB for given ϵ. This is consistent with the tighter constraint implicit in the SB τ-selection procedure (Sec. 3.4.2), as discussed previously, and coincides with the behavior seen for the decaying-dimerizing (Fig. 4.2) and simple gene expression (Fig. 4.12) models of Secs. 4.1.1 and 4.1.3, respectively. However, at larger volumes we see a reversal of fortunes. The cost curves *cross*, indicating that the SB procedure begins producing larger time steps than the RB procedure. This counterintuitive behavior is consistent with that seen for the simple clustering model of Sec. 4.1.2 (Fig. 4.7). Thus, in addition to the differential behavior seen for the simple systems of Secs. 4.1.1–4.1.3, we now have a case where the behavior *switches* based on the conditions under which the network is being considered. This further emphasizes the non-trivial nature of τ selection in the PLA. We will revisit this issue in the subsequent subsection and hypothesize as to its origin.

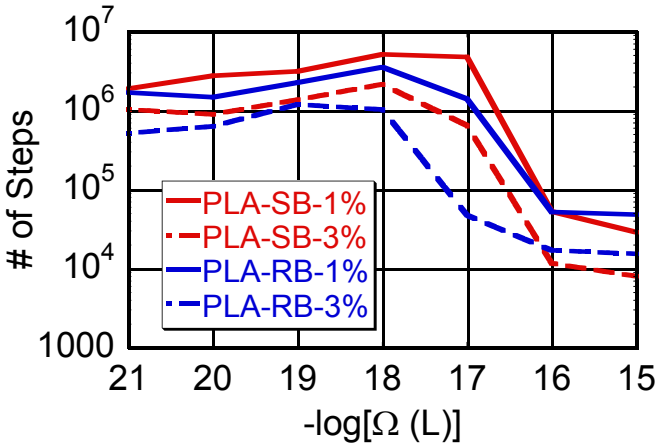

Figure 4.20: Detailed step analysis comparing the performance of the PLA-SB to the PLA-RB for simulations of the Kummer et al. [66] calcium-oscillations model of Table 4.1. Results are shown for $\epsilon = 0.01$ (solid lines) and 0.03 (dashed lines). At $\Omega = 10^{-16}$ l, the cost curves cross, indicating a reversal in the relative performances of the methods.

4.2.3 Repressilator

Synthetic biology is a relatively new and rapidly growing scientific field [5, 58, 59, 108]. In analogy with electrical circuit design, synthetic biologists attempt to use their knowledge of fundamental biological principles to design and construct artificial biological "circuits" that confer novel function unto their host. In this way, one can isolate and control specific aspects of a biological process and circumvent the immense complexity of natural biological systems, providing a means by which current theoretical understanding can be tested and scrutinized. Moreover, the long-term goal is to develop protocols for logical control. One can envision a time when microorganisms are "programmed" at the genetic level to carry out important functions, such as cleaning up oil spills or delivering tumor-suppressing drugs to specific locations within the body [58, 59].

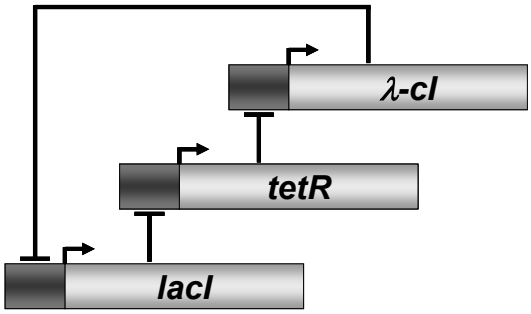

Figure 4.21: Schematic diagram of the repressilator. Each gene (*lacI*, *tetR*, *λ-cI*) produces a protein which binds to the operator site of the promoter driving expression of the next gene in the sequence, thus repressing it. Within the correct region of parameter space the repressilator oscillates, a so-called "ring oscillator" [58].

The network

Numerous artificial biological circuits have been constructed in bacteria and demonstrated to perform as designed. One such network is the repressilator, a three-gene synthetic genetic regulatory network developed by Elowitz and Leibler [33]. Each gene in the repressilator produces a protein which represses the next gene in the sequence; the protein product of the last gene represses the first gene, thus closing the loop. This construct is known in microelectronics as a "ring oscillator" [58]. As implemented experimentally in *Escherichia coli* [33], the repressilator consists of the genes *lacI*, *tetR*, and *λ-cI*; LacI protein represses *tetR*, etc. (Fig. 4.21).[12]

Under the right conditions, i.e., within the correct region of parameter space, the repressilator oscillates, acting as a biological clock. However, determining

[12]Biological convention is to denote genes in italicized font and beginning with a lower-case letter (e.g., *lacI*), mRNA transcripts by the same name, also beginning with a lower-case letter, but unitalicized (e.g., lacI), and proteins by the same name, unitalicized, but beginning with a capital letter (e.g., LacI).

the conditions for oscillation is nontrivial and theoretical modeling was employed to identify the appropriate design criteria [33]. Once functional, a particularly interesting experimental observation was the significant fluctuations in amplitude and period exhibited by the circuit. Natural oscillators, such as circadian clocks, do not exhibit such variability [51–53] and subsequent modeling indicated that Nature must employ some form of regulatory control in order to overcome this problem [4, 112]. The repressilator thus succeeded in providing valuable insight regarding the design principles underlying an important biological process.

The extensive use of modeling in the design and analysis of the repressilator, as well as the highly stochastic behavior exhibited by the network, motivates our investigation using the PLA. In Table 4.2, we show the basic form of one-third of the repressilator model (all three genes are considered equivalent). This corresponds to the "stochastic" model of Elowitz and Leibler [33]. Here, all reactions are treated as elementary using simple mass-action kinetics (i.e., rates are directly proportional to the reactant population levels—see Appendix B). Each gene is assumed to have two binding sites for repressor protein, with binding occurring sequentially, and the unbound gene transcribes mRNA 1000 times faster than the singly- or doubly-bound gene. mRNA also translates protein autocatalytically and mRNA and protein degrade with half-lives of 120 and 600 s, respectively [33].

We also include in Table 4.2 various multiplicative factors: a "telegraph factor" γ, an "RNA factor" η and a "protein factor" ρ [103, 104]. These factors allow us to control and tune the various sources of noise in the system. For example, increasing η increases the rates of gene transcription, resulting in larger mRNA populations and less mRNA-related "shot noise," i.e., noise arising from the fact that the system is comprised of discrete numbers of interacting entities (in electrical circuits,

Table 4.2: Basic form of one-third of the full repressilator model [33] (all three genes are considered equivalent). gX represents one of the three repressilator genes (*lacI*, *tetR* or λ-*cI*) and pR the corresponding repressor protein (LacI for *tetR*, etc.). mX and pX represent the mRNA and protein products of gX, respectively. All reactions are treated using simple mass-action kinetics (Appendix B) and all parameters with inverse molar units (M^{-1}) are divided by $N_A\Omega$ prior to runtime. k_1, k_2 are rate constants for forward repressor binding while k_{-1}, k_{-2} are for the reverse reactions. Also shown (see text for explanation) are the "telegraph factor" γ, the "RNA factor" η, and the "protein factor" ρ (equivalent for all genes) [103, 104]. Here, we set $\eta = \rho = 1000$ and vary $10^{-4} \leq \gamma \leq 1$.

Reaction		Parameter Value
$\vec{1}, \vec{2}.$	$gX + pR \rightleftharpoons \{gX{\cdot}pR\}$	$k_1 = 10^9\boldsymbol{\gamma}/\boldsymbol{\rho}$ $M^{-1}s^{-1}$ $k_{-1} = 224.0\boldsymbol{\gamma}$ s^{-1}
$\vec{3}, \vec{4}.$	$\{gX{\cdot}pR\} + pR \rightleftharpoons \{gX{\cdot}pR_2\}$	$k_2 = 10^9\boldsymbol{\gamma}/\boldsymbol{\rho}$ $M^{-1}s^{-1}$ $k_{-2} = 9.0\boldsymbol{\gamma}$ s^{-1}
5.	$gX \rightarrow gX + mX$	$k_3 = 0.5\boldsymbol{\eta}$ $\mathbf{s^{-1}}$
6.	$\{gX{\cdot}pR\} \rightarrow \{gX{\cdot}pR\} + mX$	$k_4 = 5{\times}10^{-4}\boldsymbol{\eta}$ $\mathbf{s^{-1}}$
7.	$\{gX{\cdot}pR_2\} \rightarrow \{gX{\cdot}pR_2\} + mX$	$k_5 = 5{\times}10^{-4}\boldsymbol{\eta}$ $\mathbf{s^{-1}}$
8.	$mX \rightarrow mX + pX$	$k_6 = 0.167\boldsymbol{\rho}/\boldsymbol{\eta}$ $\mathbf{s^{-1}}$
9.	$mX \rightarrow \emptyset$	$k_7 = \ln(2)/120$ s^{-1}
10.	$pX \rightarrow \emptyset$	$k_8 = \ln(2)/600$ s^{-1}

Initial conditions:

$[mTetR] = 3.8$ μM; $[mCI] = 8.1$ μM; $[mLacI] = 0.15$ μM;

$[pTetR] = 0.22$ mM; $[pCI] = 2.4$ mM; $[pLacI] = 0.20$ mM;

$gTetR = gCI = gLacI = 1$ (molecule);

All $\{gX{\cdot}pR\}$ and $\{gX{\cdot}pR_2\} = 0$.

shot noise arises from discrete numbers of charge carriers; in optical devices, from discrete numbers of photons) [103, 104]. The translation rate is divided by η, however, thus cancelling out the effect of increased mRNA levels on the protein production rates. Protein-related shot noise is controlled similarly through the protein factor ρ while the amount of "telegraph noise," i.e., that associated with the random switching between the ON and OFF states of the genes (reminiscent of an electronic telegraph transmitting Morse code) [103, 104], is controlled through the parameter γ.

Here, we focus primarily on the telegraph factor γ. We do so because the performance of the leaping algorithm is strongly affected by this parameter: at small values the method performs exceptionally well but falters as it is increased, approaching the performance of the SSA (see subsequent subsection). With the system volume $\Omega = 1.4 \times 10^{-15}$ l (the volume of a typical *E. Coli* cell) and η and ρ set to high values (i.e., 1000) in order to dampen the mRNA- and protein-related noise sources, we investigate how the system behavior changes for $10^{-4} \leq \gamma \leq 1$. We thus observe how the actual values of the gene-protein binding and unbinding rate constants, as opposed to simply their ratios, affect the overall dynamical behavior of the system as well as the performance of the PLA.

We also find it convenient to investigate a reduced form of the repressilator model obtained by applying the "partial equilibrium assumption" (PEA) to the first four reactions in Table 4.2. Assuming each reversible reaction to be in rapid equilibrium, simple algebra leads to effective rate expressions of the *Adair* form [28] for mRNA production from the free, singly-bound and doubly-bound genes (see Appendix E for derivations). These expressions are strictly valid in the limit $\gamma \to \infty$. Doing so reduces the 30 reactions of Table 4.2 to 18 in Table 4.3. Note that

Table 4.3: Basic form of one-third of the *reduced* repressilator model. Parameter values are the same as in Table 4.2. The *Adair* functional forms [28] describing mRNA production are similar to the well-known Hill forms, but are formally correct for $\gamma \to \infty$ (see Appendix E).

Reaction		Rate Expression
1.	$gX \to gX + mX$	$k_3 K_1 K_2 / f([pR])$
2.	$\{gX{\cdot}pR\} \to \{gX{\cdot}pR\} + mX$	$k_4 K_2 [pR] / f([pR])$
3.	$\{gX{\cdot}pR_2\} \to \{gX{\cdot}pR_2\} + mX$	$k_5 [pR]^2 / f([pR])$
4.	$mX \to mX + pX$	$k_6 [mX]$
5.	$mX \to \emptyset$	$k_7 [mX]$
6.	$pX \to \emptyset$	$k_8 [pX]$

$K_i \equiv k_{-i}/k_i$, $(i = 1, 2)$

$f([pR]) \equiv K_1 K_2 + K_2 [pR] + [pR]^2$

the reduced model in Table 4.3 differs from the "deterministic" model of Elowitz and Leibler [33] in that the expressions in Table 4.3 are directly derivable from the reactions in Table 4.2 via application of the PEA while those in Ref. [33] are not.

Statistical analysis

Our analysis of the repressilator focuses on behavioral changes that arise when the intermittent rates of switching between the transcriptional ON and OFF states of the genes are varied. The parameter that controls this is the telegraph factor γ. From an intuitive standpoint, we expect to observe large deviations from determinism at small values of γ and a convergence towards deterministic behavior with increasing γ because of the "averaging out" of the states of the genes [61]. Moreover, by making the RNA and protein factors, η and ρ, large we minimize

the effects of shot noise. However, we cannot eliminate it completely, and thus we expect to encounter some residual effects. Finally, we also expect that the PLA simulations will begin to bog down as γ is increased because of the growing disparities between the gene-protein binding and unbinding rates and the rates of all other reactions in the system [56].

In Figs. 4.22–4.25, these expectations are realized. In Fig. 4.22, we show example time courses for TetR protein (taken as representative of the system behavior) that illustrate how "deviant effects" [99] arise in the repressilator at small values of γ. With $\gamma = 10^{-4}$, we see in Fig. 4.22 that the true behavior of the system, as captured by both the PLA and the SSA, differs markedly from that predicted deterministically. Rather than emitting smooth and regular oscillations, the system produces large-amplitude intermittent "bursts" of (mRNA and) protein production. This is a direct consequence of the slow stochastic switching between the ON and OFF states of the genes and is consistent with gene-expression behavior often observed in eukaryotes [8, 61]. Note that due to stochasticity the PLA and SSA traces differ from each other. As we shall see, however, they are virtually identical from a statistical perspective.

In Fig. 4.23, we present results of our statistical analyses of the repressilator. At various values of γ, as well as at the Adair limit ($\gamma \to \infty$), we compare the stochastic behavior of the system, as captured by both the PLA and the SSA, to deterministic predictions. In Figs. 4.23*a* and 4.23*b*, we consider averages and modes for the TetR-protein peak amplitude and period, respectively. In both cases, the PLA and SSA results coincide almost perfectly and show large deviations from determinism at small values of γ and a convergence towards the deterministic limit with increasing γ. Close inspection of the PLA results reveals that statistical

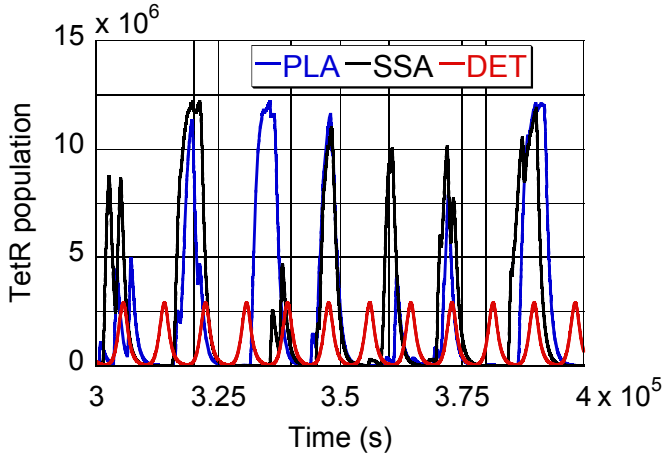

Figure 4.22: Example time courses (TetR protein) illustrating "deviant effects" [99] in the repressilator at small values of γ. With $\gamma = 10^{-4}$ and $\eta = \rho = 1000$, stochastic realizations (PLA and SSA) differ markedly from the deterministic prediction.

convergence to the deterministic limit is achieved for both attributes by $\gamma = 1$. It is also evident from these plots that the behavior of the full model (Table 4.2) approaches that of the reduced model (Table 4.3) with increasing γ, as we would expect.

In Figs. 4.23*c* and 4.23*d*, we consider the distributions of the amplitude and the period. Again, we look at ratios of standard deviations and histogram distances and again we see a convergence towards determinism with increasing γ. However, in this case the deterministic limit is never reached; even at the Adair limit we see considerable deviation from determinism. Furthermore, we see very good correspondence between the PLA and the SSA results. In fact, the only significant differences that we see are the small discrepancies in the histogram distances at the Adair limit in Fig. 4.23*d*. This is interesting in light of the discrepancies seen between the PLA and the SSA in Figs. 4.19*a* and 4.19*b* for the calcium-oscillations model which also contains reduced reaction types (see Table 4.1). Though mere speculation at this point, this suggests that reduced reactions might be to blame

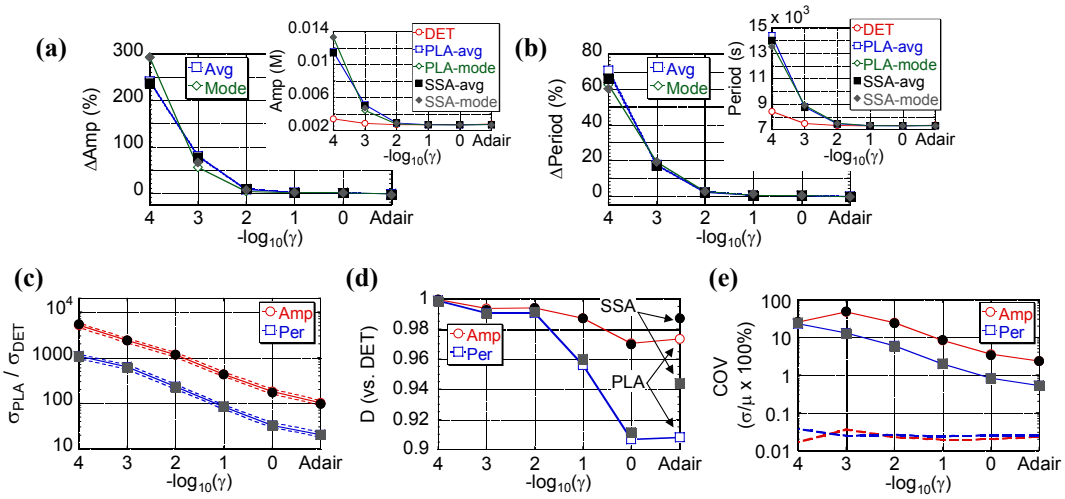

Figure 4.23: Statistical results for the repressilator. At various values of the telegraph factor γ, and at the Adair limit ($\gamma \to \infty$), results of both PLA and SSA simulations are compared to deterministic (DET) predictions. As in Fig. 4.19, PLA results are shown as colored symbols connected by lines, SSA results are shown as disconnected symbols in grey scale, and PLA and SSA points designated with the same symbol correspond to the same quantity. All PLA and deterministic values are based on over 10 000 collected data points while all SSA values are based on $\sim$1000 collected data points. Note that the only clearly discernible discrepancies between the PLA and the SSA results are the histogram distances in (d) at the Adair limit. (*a*),(*b*): Averages and modes for the TetR-protein peak amplitude and period, respectively. In the main plots, results are given as percent deviations from determinism (95% confidence intervals on the PLA averages are difficult to see). In the insets, results are shown in absolute form, illustrating the dependencies of the amplitude and period on γ. (*c*): Ratios of standard deviations ({PLA or SSA}$\div$DET) for the TetR-protein peak amplitude and period. Dashed lines denote 80% confidence intervals. (*d*): Histogram distances ({PLA or SSA} vs. DET). Note that the self distances are off the chart. (*e*): Coefficients of variation, given as percentages, obtained from both PLA and SSA simulations. In principle, the deterministic limits (dashed lines) vary with γ [see (a) and (b), *insets*], though here they are very nearly constant.

for the various inaccuracies that we see in the PLA results in Figs. 4.19 and 4.23. We plan to investigate this issue further in the future.

In Fig. 4.23*e*, we consider the noise strength through the COV. Here, as in Figs. 4.23*c* and 4.23*d*, we see almost perfect agreement between the PLA and the SSA results and an incomplete convergence towards the deterministic limit with increasing γ. It is clear, therefore, that significant shot noise effects persist in this system even as $\gamma \to \infty$. Moreover, it is interesting to note the elevated levels of noise in the amplitude as compared to the period. We see an approximately order-of-magnitude difference in the COVs for these two attributes at all values of $\gamma > 10^{-4}$ and at the Adair limit. Contrast this with Fig. 4.19*f*, which shows no appreciable difference between the COVs for the amplitude and the period in the calcium-oscillations model. This is an example of the type of fine-level insight that we can garner by using the leaping algorithm.

It is clear from Figs. 4.23*c*–4.23*e* that the repressilator never behaves in a fully deterministic manner under the conditions that we consider. However, it is also clear that the behavior does approach that of the reduced model with increasing γ. Therefore, in Fig. 4.24 we quantify this convergence to the Adair limit by repeating the statistical tests of Figs. 4.23*c* and 4.23*d* but using the PLA (or SSA) results for the reduced model, rather than the deterministic results at each γ, as our reference. The results clearly confirm the (near) convergence of the system behavior to the Adair limit at $\gamma = 1$.

Finally, in Fig. 4.25 we present results of a step and timing analysis comparing the performance of the PLA to the SSA for simulations of both the full (Table 4.2) and reduced (Table 4.3) repressilator models. For the full model, we see the convergence in computational expense of the PLA and the SSA that we anticipated

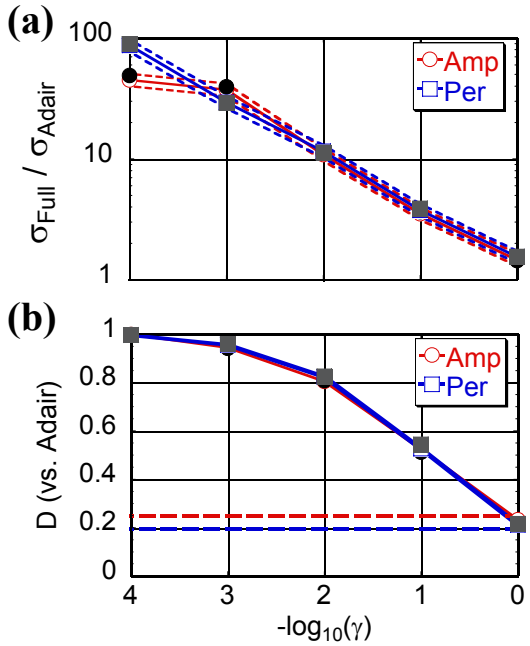

Figure 4.24: Convergence of the full repressilator model to the Adair limit with increasing γ. At each value of γ, PLA and SSA results of the full model (Table 4.2) are compared to PLA and SSA results, respectively, of the reduced model (Table 4.3). In all cases, the PLA and SSA values (colored and grey-scale symbols, respectively) coincide almost perfectly. (a): Ratios of standard deviations (full÷reduced) for the TetR-protein peak amplitude and period. Dashed lines denote 80% confidence intervals. (b): Histogram distances (full vs. reduced). Dashed lines denote twice the *Adair* self distances.

[56]. In Fig. 4.25a, the numbers of steps required for PLA and SSA simulations converge asymptotically with increasing γ. In Fig. 4.25b, we see a similar trend for the CPU times, although interestingly the curves here cross at $\gamma=1$ because each PLA step is more computationally expensive than each SSA step. Also of note is that both plots indicate that the expense of the SSA *decreases* with increasing γ while the opposite is true for the PLA. This is because the protein (and mRNA) populations, which are the prime bottleneck for the SSA, tend to be larger at small values of γ (cf. Figs. 4.22 and 4.23a). Leaping algorithms are not affected by population sizes, having been developed specifically to cope with this problem [46, 47].

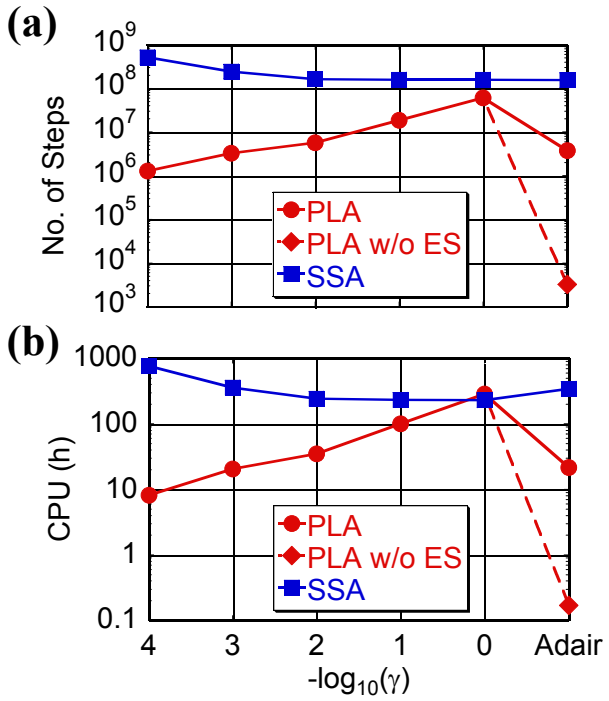

Figure 4.25: Average numbers of steps (a) and total CPU times (b) required for 1000 PLA and SSA simulation runs of 30 000 s of the full (Table 4.2) and reduced (Table 4.3) repressilator models. All SSA points are based on 100 simulation runs (due to computational expense). Note that the CPU curves in (b) cross at $\gamma = 1$ because each PLA step is more expensive than each SSA step. At the Adair limit, results are given for PLA simulations that both include and exclude the exact-stochastic (ES) classification (see footnote 13). All simulations were performed on a 3.60 GHz Pentium Xeon processor.

Hence, we see that when stochastic effects in this system are most pronounced (small γ) the PLA far outperforms the SSA.

In Ref. [56], it was posited that large disparities in rate constants would prove to be prime hindrances for leaping algorithms. This is confirmed in Fig. 4.25 by the declining performance of the PLA with increasing γ. It is for exactly this reason that we consider the reduced model of Table 4.3. In Figs. 4.23 and 4.24, we have seen that the behavior of the full model approaches that of the reduced model

with increasing γ. Now, in Fig. 4.25 we see that the performance of the PLA is greatly enhanced by the model reduction. Depending on exactly how we choose to implement the PLA,[13] we can achieve gains of between one and four orders of magnitude in both simulation steps and run times. Additionally, it is important to note that reducing the model has very little effect on the performance of the SSA. In fact, we see in Fig. 4.25 that while the numbers of simulation steps required for the SSA remain virtually unchanged upon reducing the model, the CPU time actually *increases* by $\sim 50\%$ because of the higher complexity rate expressions in Table 4.3 which impose additional computational burdens on the algorithm. Our results indicate, therefore, that there is a distinct advantage to using model reduction in conjunction with leaping which is absent with regards to the SSA.

RB *vs.* SB τ selection

As in Sec. 4.2.2, we expand here in Fig. 4.26 the step analysis of Fig. 4.25*a* in order to compare the performance of the PLA-SB to the PLA-RB when applied to both

[13] We found that significant speed-ups could be achieved in the PLA simulations of the reduced repressilator model (Table 4.3) if we removed the ES classification. The problem lies in the iterative τ-selection procedure (Sec. 3.3.2) designed to account for randomness in the ES reactions. In this specific case, we experienced an unexpected "classification cascade," whereby reactions classified as ES led to a reduced τ, which then led to more ES reactions (via reclassification), which further reduced τ, and so on and so forth. Removing the ES classification eliminated this problem with no noticeable effect on the accuracy. However, this cannot be done in all cases. Removing the ES classification when simulating the full model led to numerous instances of negative populations, particularly for the species gX, $\{gX{\cdot}pR\}$ and $\{gX{\cdot}pR_2\}$, which can have populations of only zero or unity. These required costly reversals which significantly increased the run time. Further investigation of this issue is warranted and will be undertaken in the near future.

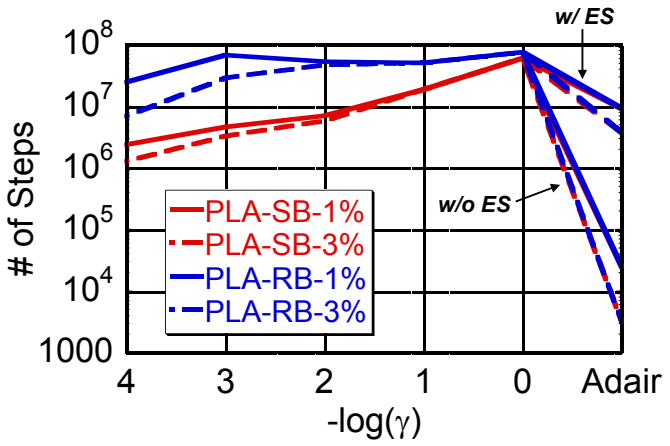

Figure 4.26: Detailed step analysis comparing the performance of the PLA-SB to the PLA-RB for simulations of the full (Table 4.2) and reduced (Table 4.3) repressilator models. Results are shown for $\epsilon = 0.01$ (solid lines) and 0.03 (dashed lines). For the reduced model, results are shown with and without the ES classification included in the PLA implementation. For $\gamma < 1$, the PLA-RB requires far more simulation steps than the PLA-SB, an unexpected result.

the full and reduced repressilator models of Tables 4.2 and 4.3, respectively. For the full model, we see significant differences in the performances of the two methods for both values of ϵ considered. The PLA-RB requires *far* more simulation steps than does the PLA-SB for $\gamma < 1$, by as many as an order of magnitude. Again, this is counterintuitive given the tighter constraint implicit in the SB τ-selection procedure (Sec. 3.4.2). Furthermore, for the reduced model, we see that both τ-selection procedures perform essentially identically, both with and without the ES classification included (see footnote 13). This suggests, therefore, that the unexpected behavior of the PLA-RB is somehow related to the binding/unbinding reactions in Table 4.2 (reactions 1–4) which are reduced out of the model in Table 4.3.

Based on this fact, we believe that we can explain the origin of this unexpected behavior, which is also seen for the simple clustering model of Sec. 4.1.2 (Fig. 4.7)

and the calcium-oscillations model of Sec. 4.2.2 at large Ω (Fig. 4.18). It appears to be due to the fact that variations in the propensities (the focus of RB τ selection) of second-order (or higher) reactions tend to be greater than the variations in the populations (the focus of SB τ selection) of the associated reactant species. This is particularly true when the population of one species is very small and the other is very large. For example, in the case of the repressilator, the gene-protein binding/unbinding reactions of Table 4.2 describe the interactions between a *single* gene and a potentially very large pool of proteins. In a given τ, one can expect that the populations of the gene species gX, $\{gX{\cdot}pR\}$ and $\{gX{\cdot}pR_2\}$ will fluctuate between zero and unity. The protein populations, on the other hand, can reach levels on the order of millions (Fig. 4.22) with fluctuations on the order of hundreds or thousands. Thus, taken together, the propensities of the binding/unbinding reactions can be expected to fluctuate, in a given τ, between *zero and the order of millions*, a much larger variation than for either of the individual reactant species populations.

In terms of τ selection, what this means is that much more variability is associated with the random variable $\Delta a_\mu(\tau_\mu^{\rm leap})$ [Eq. (3.12)], which is the focus of RB τ selection, than $\Delta X_i(T_i^{\rm leap})$ [Eq. (3.29)], which is the focus of SB τ selection. The variability in these quantities manifests in the 'σ^2' terms [Eqs. (3.28) and (3.32)] of the associated τ-selection formulas [Eqs. (3.27) and (3.30)]. Thus, increased variability in $\Delta a_\mu(\tau_\mu^{\rm leap})$ means that values of $\tau_\mu^{\rm leap}$ will, with increasing frequency, be based on the second term in Eq. (3.27) rather than the first. Our intuitive expectation that RB τ selection should result in larger time steps than SB τ selection is based on "average" considerations. It is these "variance" effects that lead to the unexpectedly small time steps for the PLA-RB simulations seen here and in Secs. 4.1.2 and 4.2.2.

These effects are most pronounced for the repressilator because of the presence of but a single gene copy, the extreme lower limit on the population scale. This results in extreme variability in the propensities of the binding/unbinding reactions, especially when the protein populations are large. The effect is not as pronounced for the simple clustering model of Sec. 4.1.2 because the populations of the larger clusters grow in time, reducing the variabilities in the associated propensities. Finally, the anomalous behavior seen for the calcium-oscillations model of Sec. 4.2.2 at large Ω is due to the fact that the differences in the populations of the species G_α, PLC^* and Ca, which oscillate out of phase with each other, reach extreme levels at large volumes (the troughs are always around zero but the peaks increase with increasing volume). The propensities of the second-order Michaelis-Menten reactions that govern the interactions between these species (Table 4.1) thus experience increased variability with increasing volume.

4.2.4 Discussion

The networks that we have considered in this section provide tangible examples of the potential utility of the PLA in computational systems biology. For both the calcium-oscillations model (Sec. 4.2.2) and the repressilator (Sec. 4.2.3), we have observed orders-of-magnitude accelerations relative to the SSA (Figs. 4.18 and 4.25) that have made quantifying stochastic effects in these systems possible. In the calcium-oscillations case, this gave us access to *subtle* effects of stochasticity that would have been indiscernible otherwise (Fig. 4.19). For the repressilator, we actually saw the greatest gains in situations where the stochastic effects were *most* prevalent (small γ—Fig. 4.23). This is a particularly intriguing result. Gene regulation is a common feature of many biological models, and our results indicate

a great potential advantage to using leaping in cases of slow transcription-factor binding and unbinding, such as is common in eukaryotes [8, 61].

Another critical aspect of this investigation has been our ability to identify conditions under which the leaping algorithm does *not* perform particularly well. We have seen that the PLA clearly falters when applied to the full repressilator model (Table 4.2) with large telegraph factor γ (Fig. 4.25). Intuitively, it is easy to understand why this is. As explained in Chapter 3, the basic strategy underlying all leaping algorithms is to allow, at each simulation step, as many reaction firings as possible without the reaction rates in the system changing "appreciably" [46, 47, 49]. However, in the case of the repressilator, there is only a single copy of each gene. Thus, only a single binding/unbinding event is possible at each simulation step; one firing changes the binding/unbinding rates from either finite values to zero or vice versa, which is obviously appreciable. When γ is small, this is not a problem because the time interval between successive binding and unbinding events is large enough so that many transcription, translation and degradation reactions can fire. When γ is large, however, this is no longer the case. The numbers of reaction firings become limited due to the high frequency of binding and unbinding, and in the extreme limit the effect is such that the performance of the algorithm approaches that of the SSA (i.e., one reaction firing per step—Fig. 4.25). We can generalize this observation by saying that small reaction subnetworks (pairs of reversible reactions in this case) that have small populations and large rate constants are prime bottlenecks for the PLA.

Fortunately, our results also illustrate how one can surmount such problems. By applying a simple rapid-equilibrium assumption to the first four reactions of Table 4.2, we were able to recover the behavior of the full model for $\gamma \geq 1$ (Fig. 4.24)

at significantly reduced computational cost (Fig. 4.25; see footnote 13). This includes accurately capturing stochastic effects associated with finite numbers of mRNAs and proteins. Interestingly, we have also shown that reducing the model has little effect on the performance of the SSA (Fig. 4.25). Thus, the chief benefit to using model reduction in this case is *not* in reducing the number of reactions that have to be considered, but rather in increasing the size of the time step that can be traversed at each simulation step. This is a different perspective on the issue than is usual and strongly suggests that leaping and model reduction should be viewed, not as alternative approaches to the problem of timescale separation (as is common), but as *complementary*. As discussed in Sec. 3.6.3 of Chapter 3, integrating leaping with advanced model-reduction schemes (e.g., [11, 19, 30, 54, 78, 97, 98, 105]) is an area of great future interest. As a final note, we did observe some (small) disagreement between the PLA and the SSA results (Figs. 4.19a, 4.19b and 4.23d) which may be due to the inclusion of the reduced reaction types. This is an issue that will be investigated further in the future.

Finally, our investigation has also helped to uncover the origins of some unexpected behavior regarding the performance of the RB τ-selection procedure (Sec. 3.4.1) in relation to the SB approach (Sec. 3.4.2). Specifically, it was believed that the RB approach would always produce larger time steps than the SB, though it was recognized that each RB step is generally more computationally expensive than each SB step. However, it is now evident that this is not always the case. In particular, when considering reactions involving multiple interacting species that have large disparities in their populations, the size of the RB time step can depend more so on the *variability* in the reaction rates than on the mean behaviors. This is much less of an issue in the SB case because the variability in the species populations is generally much less pronounced than for the reaction rates. This

provides further evidence regarding the superiority of the SB τ-selection procedure over the RB procedure, though further inquiry into the matter is warranted.[14]

[14]Remember that SB τ-selection can only be applied in the special case that the rate constants are time independent (see Sec. 3.4.2). However, there are many situations (e.g., growing cells, temperature ramps) where this requirement will not hold. Thus, understanding the shortcomings of RB τ selection is crucial if we are to develop efficient variants applicable in all cases.

Chapter 5

Conclusions and the Road Ahead

The central contribution of this dissertation is the introduction in Chapter 3 of the partitioned-leaping algorithm, a novel multiscale simulation approach that is based firmly in Markov process theory and is generally easy to implement by non-experts. The PLA was developed by merging the ideas underlying Gillespie's τ-leaping approach (Sec. 3.1) with the FRM/NRM variants of the SSA (Sec. 2.4). This results in an intuitive and seamless integration of exact-stochastic simulation into the multiscale framework, a significant achievement. The main elements of the algorithm are: (i) τ selection, (ii) reaction classification, (iii) generation of reaction firings, and (iv) system update. We have provided in Sec. 3.4 the theoretical foundation for τ selection and presented three different computational strategies. These include the reaction-based and species-based τ-selection variants and the post-leap checking procedure. The classification of reactions is simple and straightforward, being based on the criteria presented by Gillespie for approximating a Poisson distribution by a Gaussian distribution and, subsequently, by a Dirac delta function. Reaction firings are generated based on the reaction classifications. Exact-stochastic reactions are handled using the methods of the FRM/NRM. Poisson and Langevin-type reactions require the generation of Poisson and Gaussian random deviates, respectively, while deterministic reactions simply require multiplying the reaction propensity $a_\mu(t)$ by τ.

In Sec. 3.3.2, we discussed some technical issues that must be taken into account when implementing the PLA. These include rounding the firings of Langevin- and deterministic-type reactions and applying an iterative approach to account for the randomness associated with the firing times of ES reactions. These issues only

slightly complicate the approach and add minimal computational overhead. The iterative procedure does, however, lead to a "classification cascade" problem in some cases which can significantly hinder the approach, as alluded to briefly in Sec. 4.2.3. This issue will be further investigated in the near future. Once implemented, application of the PLA requires nothing more than a system description, with associated rate parameters, and the definition of three model-independent parameters: ≈ 1, $\gg 1$ and $\ll 1$ (i.e., ϵ). These parameters are intuitively simple and straightforward; typical values are 3, 100 and 0.01–0.05, respectively. The algorithm will then automatically and dynamically determine during the course of a simulation the appropriate level at which to treat each reaction in the system. For systems with large disparities in species populations, or those in which populations oscillate between low and high levels, the method can achieve significant computational savings relative to the SSA while still capturing essential stochastic effects associated with small-population species.

Chapter 2 of this dissertation provides a detailed treatment of exact-stochastic simulation approaches. Besides being essential foundational material for the development of the PLA, the chapter also serves as a review/tutorial of the subject at a level accessible to advanced undergraduates or beginning graduate students in chemical engineering and associated disciplines. Our presentation goes well beyond the standard treatment, considering the general case of both intrinsic and extrinsic sources of noise and time-dependent rate parameters. The traditional kMC formulations are shown as special cases of the more general theory. We also discuss various extensions and optimizations of the approach, some of which can be incorporated into future enhancements of the PLA. These include optimization strategies in the DM based on ordering reactions from largest propensity to smallest and spatial SSA variants that utilize the general principle of grouping reactions

and choosing over and within the groups using a SSA method of one's choosing.

In Chapter 3, we introduced the PLA, provided a review of related leaping and multiscale simulation approaches and provided thoughts on various possible future extensions to the method (see below for further discussion). We followed this in Chapter 4 by applying the PLA to various example systems that demonstrate the utility of the approach and also elucidate some shortcomings. We considered various toy problems inspired by chemistry, biology and materials science and then advanced to more realistic biological problems that highlight some of the challenges that remain in developing a practical simulation approach based on the PLA. Our primary conclusions are that (i) the PLA can achieve orders-of-magnitude accelerations relative to exact-stochastic methods for systems with large disparities in species populations and similar rate parameters, and (ii) the method bogs down when faced with large disparities in rate parameters (i.e., stiffness). The latter effect is particularly prevalent for cases involving small-population species (e.g., genes) that participate in large-rate-parameter processes (e.g., transcription factor binding/unbinding). We demonstrated, however, that the problem can be overcome by utilizing a model reduction strategy that removes explicit consideration of the fast processes. In Sec. 4.2.3, we applied a simple rapid-equilibrium reduction approach as a proof-of-principle demonstration. In Sec. 3.6.3, we discussed more advanced model reduction strategies that could be combined with the PLA. Importantly, we showed that the primary benefit of integrating model reduction with the PLA lies in increasing the size of the time step τ, as opposed to reducing the number of reactions being considered.

The material presented in Secs. 3.5 and 3.6 of this dissertation provides a practical guide for extensions and modifications to the PLA that can be under-

taken in the short- and mid-term. For example, higher-order Runge-Kutta and implicit formulations of the PLA can be implemented relatively quickly as they are straightforward extensions to the method. These will be particularly important in extending the utility of the approach to more realistic systems. Incorporating post-leap checking into the PLA is another area of great importance that can be accomplished relatively easily in the short-term, as is extending the method to spatially-inhomogeneous systems. Progress has been made recently with respect to the latter with promising results. However, challenges remain, particularly with regard to the high expense of τ selection. As emphasized in Sec. 3.6.1, in fact, τ selection is by far the most computationally expensive aspect of the PLA, and all leaping methods in general, and developing improved τ-selection procedures will be the innovation that will have the greatest impact in the mid-term. Future τ-selection procedures will likely include some form of post-leap checking, perhaps coupled with the RB and SB τ-selection variants presented in Secs. 3.4.1 and 3.4.2, respectively, or some yet-to-be developed alternative approach. Finally, integrating some form of dynamic model reduction with the PLA is a critical mid-term task. Many practical systems in biology and materials science, as demonstrated in Sec. 4.2.3, contain fast processes involving small numbers of interacting entities. Devising a strategy for handling such situations will thus be crucial for bringing to bear on systems of practical import the full power of the PLA.

In the long-term, we see the PLA as being but one component of a larger modeling and simulation framework for analyzing large-scale population-dynamical models of physical systems in both time and space. We hope to merge the method with modern model building and specification platforms at the front end, such as rule-based languages that have been recently introduced to succinctly describe complex biochemical networks, and at the back end with stochastic parameter estimation

and model checking techniques that are prevalent in computer science. Adaptive meshing techniques will also be critically important for spatial simulations, as will visualization methods for providing an intuitive understanding of system behavior. In sum, the PLA is a small yet significant innovation in the field of computational simulation methodologies that moves us one step closer to the ultimate goal of understanding the internal mechanisms by which many chemical, biological and materials systems operate.

APPENDIX A

Some useful functions and theorems

Dirac delta function.

$$\textit{Definition (two parts).} \quad \begin{array}{ll} (i) & \delta(x-a) = 0 \quad for \quad x \neq a, \\ (ii) & \displaystyle\int_{-\infty}^{\infty} \delta(x-a)dx = 1 \end{array} \tag{A.1}$$

$$\textit{Sifting property.} \quad \int_{-\infty}^{\infty} f(x)\delta(x-a)dx = f(a) \tag{A.2}$$

Heaviside step function.

$$\textit{Definition.} \quad H(x) = \begin{cases} 0 & for \quad x \leq 0 \\ 1 & for \quad x > 0 \end{cases} \tag{A.3}$$

Binomial distribution.

$$\textit{Random variable}: \quad \mathcal{B}(n,p) \tag{A.4}$$

$$\textit{Distribution}: \quad \Pr\{\mathcal{B}(n,p) = k\} = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k} \tag{A.5}$$

$$\textit{Mean}: \quad np \tag{A.6}$$

$$\textit{Variance}: \quad np(1-p) \tag{A.7}$$

Poisson distribution.

$$\textit{Random variable}: \quad \mathcal{P}(at) \tag{A.8}$$

$$\textit{Distribution}: \quad \Pr\{\mathcal{P}(at) = k\} = \frac{(at)^k e^{-at}}{k!} \tag{A.9}$$

$$\textit{Mean and variance}: \quad at \tag{A.10}$$

Gaussian (normal) distribution.

$$\textit{Random variable}: \quad \mathcal{N}(m, \sigma^2) \tag{A.11}$$

$$\textit{Distribution}: \quad \Pr\{x < \mathcal{N}(m, \sigma^2) < x + dx\} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) dx \tag{A.12}$$

$$\textit{Mean}: \quad m \tag{A.13}$$

$$\textit{Variance}: \quad \sigma^2 \tag{A.14}$$

Linear combination theorem [44]. *For any set of random variables* $\{X_1, \ldots, X_N\}$ *and any set of constants* $\{c_1, \ldots, c_N\}$,

$$\left\langle \sum_{i=1}^{N} c_i X_i \right\rangle = \sum_{i=1}^{N} c_i \left\langle X_i \right\rangle, \tag{A.15}$$

$$\operatorname{var}\left\{ \sum_{i=1}^{N} c_i X_i \right\} = \sum_{i=1}^{N} c_i^2 \operatorname{var}\{X_i\} + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_i c_j \operatorname{cov}\{X_i, X_j\}. \tag{A.16}$$

Random variable transformation (RVT) theorem [44] (single-variable version). *If the random variable* X *has the density function* $P(x)$*, and if the random variable* Y *is defined via* $Y = f(X)$*, then the density function* $Q(y)$ *of* Y *is given by*

$$Q(y) = \int_{-\infty}^{\infty} dx P(x) \delta(y - f(x)). \tag{A.17}$$

APPENDIX B

Mass-action kinetics

The simplest and most common type of interaction rule used in chemical kinetics (population dynamics in general) is that based on the "law of mass action." This law states, very simply, that the rate (propensity, in the stochastic jargon) of a chemical reaction is directly proportional to the "amount(s)" of the reactant species present in the system. Often, amounts are thought of in terms of concentrations, particularly when considering dynamics in the deterministic limit. It is important to recognize, however, that the law of mass action is not synonymous with determinism. In fact, it is better viewed from a probabilistic perspective, as Gillespie did in his seminal work on the SSA [41].

In Ref. [41], Gillespie cast the SSA in terms of elementary reactions with interaction probabilities obeying the law of mass action. In particular, the original version of the fundamental hypothesis, Eq. (2.1), was posited as

$$c_\mu(t)dt = c_\mu\left(\mathbf{\Theta}(t)\right)dt \quad \equiv \quad \begin{array}{l}\text{probability, at time } t\text{, that a } \textit{particular} \text{ reactant entity, or set of entities, of reaction } R_\mu \text{ will react within the next infinitesimal time interval } dt.\end{array} \tag{B.1}$$

The difference between this and Eq. (2.1) is the emphasis placed above on 'particular.' With this definition, the probability that *any* reactant entity (as opposed to a particular entity), or set of entities, of reaction R_μ will react in dt is then $c_\mu(t)dt$ multiplied by the number of combinations of potential R_μ reactant interactions. The latter quantity, denoted as $h_\mu(t) = h_\mu(\mathbf{X}(t))$ [41], can be determined for differ-

Table B.1: Propensity constants and degeneracies for different elementary reaction types [41]. Reactions up to third order are shown. Propensity constants are given as functions of the associated deterministic *rate* constants k_μ. N_A is Avogadro's number (6.022×10^{23} molecules/mol) and Ω is the system volume. The first reaction is a creation event with $\emptyset$ representing a source. Although third-order reactions are (virtually) impossible in nature, they are included here for completeness and to better illustrate the combinatorics underlying the degeneracy formulas.

Reaction	$c_\mu(\mathbf{\Theta}(t))$	$h_\mu(\mathbf{X}(t))$
$\emptyset \xrightarrow{k_\mu}$ *products*	$k_\mu \times N_A\Omega$	1
$S_i \xrightarrow{k_\mu}$ *products*	k_μ	X_i
$2S_i \xrightarrow{k_\mu}$ *products*	$2k_\mu/N_A\Omega$	$X_i(X_i-1)/2$
$S_i + S_j \xrightarrow{k_\mu}$ *products*	$k_\mu/N_A\Omega$	X_iX_j
$3S_i \xrightarrow{k_\mu}$ *products*	$6k_\mu/(N_A\Omega)^2$	$X_i(X_i-1)(X_i-2)/6$
$2S_i + S_j \xrightarrow{k_\mu}$ *products*	$2k_\mu/(N_A\Omega)^2$	$X_i(X_i-1)X_j/2$
$S_i + S_j + S_k \xrightarrow{k_\mu}$ *products*	$k_\mu/(N_A\Omega)^2$	$X_iX_jX_k$

ent reaction types from simple combinatorics (Table B.1). We term this quantity the reaction "degeneracy." The propensity can thus be written in this case as

$$a_\mu\left(\mathbf{X}(t),\mathbf{\Theta}(t)\right) = c_\mu(\mathbf{\Theta}(t)) \times h_\mu(\mathbf{X}(t)). \tag{B.2}$$

This is, in fact, a statement of the law of mass action. The number of ways in which R_μ reactant entities can interact, h_μ, provides the "mass" dependence while c_μ is the proportionality constant.

Written this way, Eq. (B.2) is clearly reminiscent of the phenomenological expression for the reaction *rate* of deterministic chemical kinetics. Indeed, we will term c_μ the "propensity constant" in keeping with the analogy. Gillespie pointed this out in [41], noting that the expressions in both cases are essentially identical except for a trivial factor of volume and small differences when multiple instances of

the same species are involved. For example, consider the monomolecular dimerization $2S_i \rightarrow products$. In counting the number of ways that this reaction can occur, one must (i) not allow entities to interact with themselves, and (ii) not double count equivalent interactions (i.e., molecule x reacting with molecule y is the same as y reacting with x). Thus, in Table B.1 we see that $h_\mu = X_i(X_i - 1)/2$ (contrast this with the bimolecular reaction $S_i + S_j \rightarrow products$, where $h_\mu = X_i X_j$). In the deterministic case, the expression for the rate of this reaction is $r_\mu = k_\mu [S_i]^2$, where k_μ is the rate constant and $[\cdot]$ denotes concentration. The relationship between X_i and $[S_i]$ is trivial, i.e., $X_i = [S_i] \times (N_A \Omega)$, with N_A being Avogadro's number and Ω the volume. In the limit of large X_i, we have $X_i(X_i - 1) \rightarrow X_i^2$. Thus, we see that $a_\mu \rightarrow r_\mu$ as $X_i \rightarrow \infty$ with $c_\mu = 2k_\mu/(N_A \Omega)$ (note that the factor of $1/2$ in h_μ has been absorbed into k_μ and must be accounted for when converting k_μ to c_μ). Similar formulas for the propensity constants of other elementary reactions are shown in Table B.1.

What Gillespie did in [41], therefore, is to provide a physical explanation for the long-standing *phenomenological* rate expressions commonly used in chemical kinetics. Rather than being a simple proportionality constant, for example, Gillespie showed that the rate constant is, in fact, an interaction probability [Eq. (B.1)]. Since the probability of, e.g., an encounter between two objects, clearly increases as the populations of those objects increases, the law of mass action becomes, in hindsight, intuitively obvious. From an educational point of view, this is extremely enlightening.

The consequence, however, of introducing the SSA in this way, both in [41] and in the highly-cited article [42], has been, as mentioned in Sec. 2.1, that there appears to be a common misperception in the literature today that the SSA is only

applicable to elementary reaction types that obey the law of mass action (this is not true in every case, but it is prevalent). The truth, however, is that the SSA is much more general than this. The algorithm is, by construction, indifferent as to the functional forms of the transition probabilities a_μ, so long as they are Markovian in nature. This means that common reduction methods, such as Michaelis-Menten, can be used in concert with the SSA. It is important to recognize, however, that the conditions under which certain assumptions hold (rapid equilibrium in the case of Michaelis-Menten) may be different in the small-number case (the realm of the SSA) than in the deterministic limit. This must be carefully considered in order to obtain meaningful results when applying the SSA in such situations.

APPENDIX C

Inversion generating technique

The inversion generating technique is a commonly-used method for obtaining random samples from either discrete or continuous probability functions [44]. In the continuous case, given a probability density function $P(x)$, the procedure involves calculating the associated *probability distribution function* $F(x)$ (also known as the cumulative distribution function), defined as

$$F(x) \equiv \int_{-\infty}^{x} P(x')\mathrm{d}x'. \tag{C.1}$$

A representative sample x from $P(x)$ can then be obtained by setting $F(x)=r$, a unit uniform random number on $[0,1)$, and inverting the expression to solve for x. Symbolically, the inversion procedure is written as

$$F^{-1}(r) = x. \tag{C.2}$$

The procedure works because $F(x)$ lies on $[0,1)$ [i.e., $F(-\infty)=0$ and $F(\infty)=1$] and thus maps $P(x)$ onto this interval. To understand this, consider Fig. C.1. In the top row we have plotted two normalized Gaussian probability density functions, both with mean 10. The standard deviation for the narrow Gaussian on the left is 0.5 while that for the wider one on the right is 2.0. In the bottom row we have plotted the associated distribution functions $F(x)$. The blue dotted lines illustrate the idea behind the inversion generating technique. Upon generating a random number r on $[0,1)$, the same range as $F(x)$, the value of x associated with $F(x)=r$ is then determined. For the distribution on the left, these values will lie close to the mean of the associated Gaussian because of the steep transition from $F(x)=0$ to 1. More variability will be seen for the values obtained from the distribution on the right, however, consistent with the broader density function.

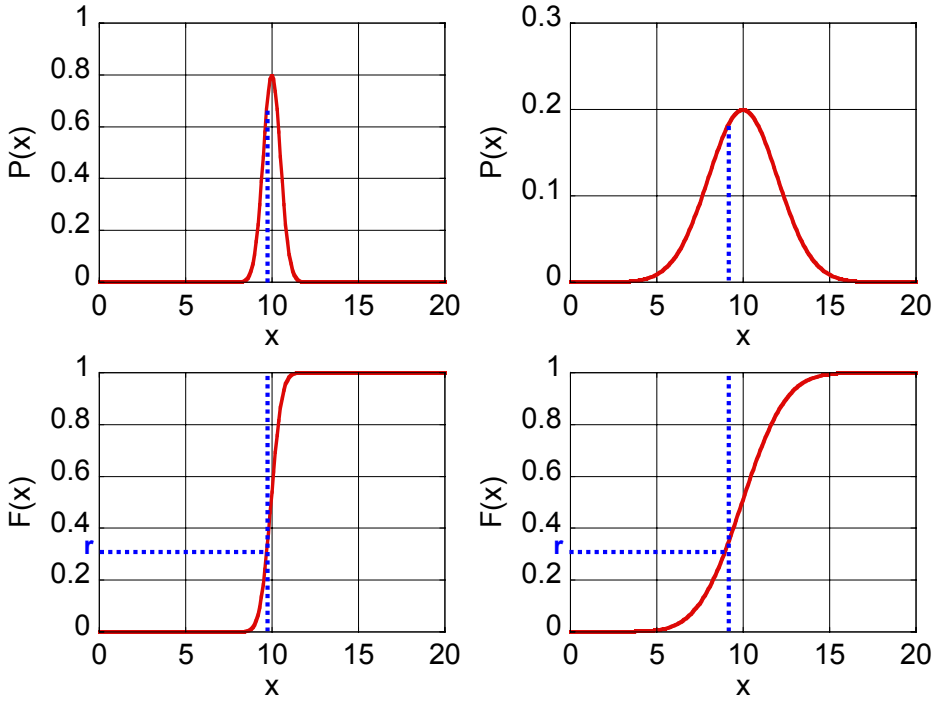

Figure C.1: Graphical illustration of the inversion generating technique. Gaussian probability density functions $P(x)$ are shown along with their associated distribution functions $F(x)$. (*Left column*): mean=10, standard deviation=0.5; (*right column*): mean=10, standard deviation=2.0. The blue dotted lines illustrate the concept behind the inversion generating technique. Upon drawing a random number r on $[0, 1)$, the value of x associated with $F(x) = r$ is determined. For the narrow Gaussian on the left, most of the values will lie close to the mean, while larger deviations will be seen for the Gaussian on the right.

Application of the inversion generating technique obviously requires being able to evaluate the integral Eq. (C.1). In general, this can be done numerically, iteratively adjusting the upper bound x until a value is found at which the integral equals r (within a small tolerance) [96]. Ideally, however, the integral can be evaluated analytically, yielding a simple expression for x from Eq. (C.2). This happens to be the case for the τ-independent propensity functions of Chapter 2.

Consider Eq. (2.12), the continuous probability function governing next-

reaction times τ within the direct method of the SSA. In the τ-independent case, we have $a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) = a_0(\mathbf{x_t}, \theta_{\mathbf{t}}) = a_0(t)$, and hence

$$\begin{aligned} P_1(\tau|\mathbf{x_t}, t)d\tau &= a_0(t) \exp\left[-\int_0^\tau a_0(t)d\tau'\right] d\tau \\ &= a_0(t) \exp\left[-a_0(t)\tau\right] d\tau. \end{aligned} \tag{C.3}$$

We can then obtain an analytical expression for τ that is consistent with this expression by evaluating the distribution function,

$$\begin{aligned} F_1(\tau|\mathbf{x_t}, t) &\equiv \int_0^\tau P_1(\tau'|\mathbf{x_t}, t)d\tau' \\ &= 1 - \exp\left[-a_0(t)\tau\right] \end{aligned} \tag{C.4}$$

[note that the lower integration bound is set to zero because $P_1(\tau|\mathbf{x_t}, t) = 0$ for $\tau < 0$]. Setting Eq. (C.4) equal to $r \in [0, 1)$ and rearranging gives

$$\tau = -\frac{\ln(1-r)}{a_0(t)} = -\frac{\ln(r)}{a_0(t)}, \tag{C.5}$$

where the second equality notes that $1-r$ is, itself, a random number on $[0, 1)$.

Equation (C.5) is used extensively throughout the literature. It is important to recognize, however, that it is formally valid *only* in the case of τ-independent propensities. Since the species populations $\mathbf{X}(t)$ remain constant between successive reaction firings, τ dependency can only manifest itself via time-varying environmental quantities $\mathbf{\Theta}(t)$, such as volume or temperature. (Note that in some cases, concentrations of "fast" species, treated as continuous and deterministic, can be included in $\mathbf{\Theta}(t)$ in order to account for their effects on reactions involving "slow" species [96].)

There is also an integer version of the inversion generating technique that is a simple extension of the continuous version [44]. Instead of an integration, the

distribution function $F(n)$, $n \in \mathbb{Z}$, the set of all integers, is defined as

$$F(n) \equiv \sum_{n'=-\infty}^{n} P(n'). \tag{C.6}$$

A random sample n of $P(n)$ is then the value of n that satisfies the double inequality

$$F(n-1) \leq r < F(n), \tag{C.7}$$

with r again is a unit-uniform random number on $[0, 1)$.

As a simple example, consider Eq. (2.13), the discrete probability function governing next-reaction types μ within the direct method of the SSA,

$$P_2(\mu|t+\tau) = a_\mu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \, / a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \, . \tag{C.8}$$

Note that in this case, whether the propensities are τ dependent or not is irrelevant. Using Eqs. (C.6) and (C.7), a sample value μ is then the integer that satisfies

$$\sum_{\nu=1}^{\mu-1} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) \leq r \times a_0(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}) < \sum_{\nu=1}^{\mu} a_\nu(\mathbf{x_t}, \theta_{\mathbf{t}+\tau}). \tag{C.9}$$

Again, this expression is widely used in the literature.

APPENDIX D

Calculation of $g_{\mu i}(t)$ for various reaction types in SB τ selection

The species-based τ-selection procedure presented in Sec. 3.4.2 of this dissertation was first proposed by Cao et al. in Ref. [20]. In that work, the authors considered only elementary reaction types (i.e., those obeying the law of mass action—see Appendix B) and derived $g_{\mu i}(t)$ expressions for reactions up to third order. In this Appendix, we will reproduce these derivations along with some simplifications that we proposed in Ref. [56] and also derive expressions for some non-elementary reaction types that we considered in Ref. [57] and Sec. 4.2 above.

Elementary 1st order. In the species-based τ-selection procedure of Sec. 3.4.2, the scaling factor $g_i(t)$ [Eq. (3.29)], associated with species S_i, is determined by calculating individual factors $\{g_{\nu i}(t)\}$ for S_i in each reaction (i.e., $\nu = 1 \ldots M$), and then choosing $g_i(t)$ as the largest of these [Eq. (3.34)]. Our job here, therefore, is to determine expressions for $g_{\mu i}(t)$ for various types of reactions.

The simplest type of reaction to consider is the elementary first-order reaction

$$S_i \xrightarrow{c_\mu} products,$$

which has the propensity

$$a_\mu = c_\mu X_i.$$

The general procedure for determining $g_{\mu i}(t)$ is given by Eqs. (3.35)–(3.39) of Sec. 3.4.2. Thus, we begin by deriving an expression for $\Delta a_\mu / a_\mu$ of the form given

in Eq. (3.35), which, in this case, is easy to do,[1]

$$\frac{\Delta a_\mu}{a_\mu} = \frac{da_\mu}{dX_i}\frac{\Delta X_i}{a_\mu} = \frac{\Delta X_i}{X_i}.$$

In terms of Eq. (3.35), this means that $\alpha_{\mu i} = 1$ and $\{\alpha_{\mu j} = 0\}_{j \neq i}$. We now obtain an expression for $|\Delta a_\mu| / a_\mu$ using Eq. (3.36), which is trivial in this case,

$$\frac{|\Delta a_\mu|}{a_\mu} = \frac{|\Delta X_i|}{X_i} = \epsilon_\mu.$$

Following Eq. (3.37), we must now place a constraint on $|\Delta X_i| / X_i$ that is tighter than, or equal to, that placed on $|\Delta a_\mu| / a_\mu$. In this case, it is obvious from the above expression that the constraint is the same. Thus, the scaling factor $\zeta_{\mu i} = 1$ in Eq. (3.37). This choice is intuitively obvious, but one could also say that it is required by the condition in Eq. (3.38). Finally, using Eq. (3.39), we get the result

$$g_{\mu i} = 1. \tag{D.1}$$

In the following examples, we will follow this same procedure, though less verbosely. Where complications and/or subtleties arise we will address them, and where simplifications can be made we will suggest them.

Elementary bimolecular 2nd order. For a second-order elementary reaction involving the interaction between two different molecular species

$$S_i + S_j \xrightarrow{c_\mu} products,$$

we have

$$a_\mu = c_\mu X_i X_j.$$

[1] In all that follows, we will continue this practice of expressing Δa_μ in terms of derivatives of a_μ with respect to species populations, multiplied by the changes in the species populations.

Following the same procedure as above, we can write $\Delta a_\mu/a_\mu$ as

$$\frac{\Delta a_\mu}{a_\mu} \approx \frac{\partial a_\mu}{\partial X_i}\frac{\Delta X_i}{a_\mu} + \frac{\partial a_\mu}{\partial X_j}\frac{\Delta X_j}{a_\mu} = \frac{\Delta X_i}{X_i} + \frac{\Delta X_j}{X_j},$$

which means that $\alpha_{\mu i}=\alpha_{\mu j}=1$. Note that we are neglecting here the correlations between ΔX_i and ΔX_j [20]. Using Eq. (3.36), we then write

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \frac{|\Delta X_i|}{X_i} + \frac{|\Delta X_j|}{X_j} = \epsilon_\mu.$$

Thus, if we constrain $|\Delta X_i|/X_i = |\Delta X_j|/X_j = \epsilon_\mu/2$, then we assure that $|\Delta a_\mu|/a_\mu \lesssim \epsilon_\mu$. This means that $\zeta_{\mu i}=\zeta_{\mu j}=2$ and, hence,

$$g_{\mu i} = g_{\mu j} = 2. \tag{D.2}$$

Elementary monomolecular 2nd order. For a second-order reaction involving the interaction between two entities of the same species

$$2S_i \xrightarrow{c_\mu} products,$$

we have

$$a_\mu = c_\mu X_i(X_i - 1)/2,$$

which leads to

$$\frac{\Delta a_\mu}{a_\mu} = \frac{da_\mu}{dX_i}\frac{\Delta X_i}{a_\mu} = \left(2 + \frac{1}{X_i - 1}\right)\frac{\Delta X_i}{X_i},$$
$$\frac{|\Delta a_\mu|}{a_\mu} = \left(2 + \frac{1}{X_i - 1}\right)\frac{|\Delta X_i|}{X_i} = \epsilon_\mu.$$

This means that $\alpha_{\mu i}$ is the term in the parentheses in the above expressions and, since there is only a single term, $\zeta_{\mu i}=1$. Thus,

$$g_{\mu i}(t) = 2 + \frac{1}{X_i(t) - 1}. \tag{D.3}$$

This is our first example where $g_{\mu i}$ is time dependent and, hence, would need to be calculated at each step of a leaping simulation. However, the expression in Eq. (D.3) has clear upper and lower bounds,

$$\begin{aligned} g_{\mu i}(X_i = 2) &= 3, \\ g_{\mu i}(X_i \to \infty) &= 2. \end{aligned}$$

The lower bound is at $X_i = 2$ because the reaction cannot fire if the population is less than this. The fact that this range is so small led us to argue in Ref. [56] that it is not worth the effort to calculate $g_{\mu i}(t)$ at every simulation step. Thus, for simplicity, we suggested that one simply choose the upper bound,

$$g_{\mu i}(t) = g_{\mu i} = 3. \tag{D.4}$$

This simplifies the τ-selection procedure significantly with very little loss in efficiency.

Elementary trimolecular 3rd order. Although extremely rare in nature, third-order reactions are often considered for completeness [20] and because they have long been used as approximations for more complex mechanisms, especially in systems that exhibit oscillations [41].

If we consider a third-order reaction of the type

$$S_i + S_j + S_k \xrightarrow{c_\mu} \textit{products},$$

and assume that it obeys the law of mass action (Appendix B), i.e.,

$$a_\mu = c_\mu X_i X_j X_k,$$

then we have

$$\frac{\Delta a_\mu}{a_\mu} \approx \frac{\partial a_\mu}{\partial X_i}\frac{\Delta X_i}{a_\mu} + \frac{\partial a_\mu}{\partial X_j}\frac{\Delta X_j}{a_\mu} + \frac{\partial a_\mu}{\partial X_K}\frac{\Delta X_K}{a_\mu} = \frac{\Delta X_i}{X_i} + \frac{\Delta X_j}{X_j} + \frac{\Delta X_K}{X_K},$$

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \frac{|\Delta X_i|}{X_i} + \frac{|\Delta X_j|}{X_j} + \frac{|\Delta X_K|}{X_K} = \epsilon_\mu.$$

Thus, we have $\alpha_{\mu i} = \alpha_{\mu j} = \alpha_{\mu k} = 1$, and if we constrain $|\Delta X_i| / X_i = |\Delta X_j| / X_j = |\Delta X_K| / X_K = \epsilon_\mu / 3$, then we assure that $|\Delta a_\mu| / a_\mu \lesssim \epsilon_\mu$. This means that $\zeta_{\mu i} = \zeta_{\mu j} = \zeta_{\mu k} = 3$ and

$$g_{\mu i} = g_{\mu j} = g_{\mu k} = 3. \tag{D.5}$$

Elementary bimolecular 3rd order. If we consider a third-order reaction that involves only two interacting species

$$S_i + 2S_j \xrightarrow{c_\mu} products,$$

and obeys the law of mass action

$$a_\mu = c_\mu X_i X_j (X_j - 1)/2,$$

then we have

$$\frac{\Delta a_\mu}{a_\mu} \approx \frac{\partial a_\mu}{\partial X_i}\frac{\Delta X_i}{a_\mu} + \frac{\partial a_\mu}{\partial X_j}\frac{\Delta X_j}{a_\mu} = \frac{\Delta X_i}{X_i} + \left(2 + \frac{1}{X_j - 1}\right)\frac{\Delta X_j}{X_j},$$

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \frac{|\Delta X_i|}{X_i} + \left(2 + \frac{1}{X_j - 1}\right)\frac{|\Delta X_j|}{X_j} = \epsilon_\mu.$$

This means that $\alpha_{\mu i} = 1$ and $\alpha_{\mu j}$ is the quantity in parentheses in the above expressions. With regards to the scaling factors $\zeta_{\mu i}$ and $\zeta_{\mu j}$, we have an interesting situation. Since the above expression is comprised of two terms, one's first instinct might be to set $\zeta_{\mu i} = \zeta_{\mu j} = 2$. However, being a third-order reaction, Cao et al. [20] proposed instead that $\zeta_{\mu i} = 3$ and $\zeta_{\mu j} = 3/2$. This is the first example in which determining these scaling factors is not immediately obvious. Using these quantities, we have

$$g_{\mu i} = 3, \tag{D.6}$$

$$g_{\mu j}(t) = \frac{3}{2}\left(2 + \frac{1}{X_j(t) - 1}\right). \tag{D.7}$$

Now, again we have a situation where there are clear upper and lower bounds on the latter of these expressions,

$$\begin{aligned} g_{\mu j}(X_j = 2) &= 9/2, \\ g_{\mu j}(X_j \to \infty) &= 3. \end{aligned}$$

Thus, for simplicity, we suggested in [56] that one use the upper bound

$$g_{\mu j}(t) = g_{\mu j} = 9/2. \tag{D.8}$$

Elementary monomolecular 3rd order. The last elementary reaction that we consider is the third-order reaction involving three entities of the same species,

$$3S_i \xrightarrow{c_\mu} products,$$

which has the propensity

$$a_\mu = c_\mu X_i(X_i - 1)(X_i - 2)/6.$$

Following the same procedure as above, we have

$$\begin{aligned} \frac{\Delta a_\mu}{a_\mu} = \frac{da_\mu}{dX_i}\frac{\Delta X_i}{a_\mu} &= \left(3 + \frac{1}{X_i - 1} + \frac{2}{X_i - 2}\right)\frac{\Delta X_i}{X_i}, \\ \frac{|\Delta a_\mu|}{a_\mu} &= \left(3 + \frac{1}{X_i - 1} + \frac{2}{X_i - 2}\right)\frac{|\Delta X_i|}{X_i} = \epsilon_\mu. \end{aligned}$$

$\alpha_{\mu i}$ is thus the quantity in parentheses and, since there is only one term, $\zeta_{\mu i} = 1$. Therefore,

$$g_{\mu i}(t) = 3 + \frac{1}{X_i(t) - 1} + \frac{2}{X_i(t) - 2}. \tag{D.9}$$

Again, this expression has clear upper and lower bounds,

$$\begin{aligned} g_{\mu i}(X_i = 3) &= 11/2, \\ g_{\mu i}(X_i \to \infty) &= 3. \end{aligned}$$

The lower bound is at $X_i = 3$ because the reaction cannot fire if the population is less than this. In Ref. [56], we argued, for simplicity, that one simply choose the upper bound,

$$g_{\mu i}(t) = g_{\mu i} = 11/2. \tag{D.10}$$

Michaelis-Menten.[2] Enzyme-catalyzed reactions are ubiquitous throughout biology [28]. Perhaps the most well-known of these is the Michaelis-Menten mechanism, where a substrate S is converted into a product P though the action of an enzyme E. The transformation involves the three elementary reactions,

$$E + S \underset{c_{-1}}{\overset{c_1}{\rightleftharpoons}} \{E{\cdot}S\} \xrightarrow{c_{\rm cat}} E + P$$

(note that we have written these in terms of *propensity constants*, i.e., the rate parameters have units of $[time^{-1}]$). The Michaelis-Menten approximation is to assume that the enzyme-substrate complex $\{E{\cdot}S\}$ is in quasi-equilibrium, meaning that the rate (or probability) of its production equals its rate of consumption, i.e.,

$$c_1 X_E X_S = (c_{-1} + c_{\rm cat}) X_{\{E\cdot S\}}. \tag{D.11}$$

This is known as the quasi-steady state assumption (QSSA).

The rate (propensity) of product formation can then be written as

$$a_P = c_{\rm cat} X_{\{E\cdot S\}} = \frac{c_{\rm cat}}{C_M} X_E X_S, \tag{D.12}$$

where the "Michaelis constant"

$$C_M \equiv \frac{c_{-1} + c_{\rm cat}}{c_1}. \tag{D.13}$$

If we then write the *total* enzyme population as

$$E_T \equiv X_E + X_{\{E\cdot S\}}, \tag{D.14}$$

[2]The material in this subsection has been adapted from Ref. [57].

we can derive an expression for X_E by substituting Eq. (D.14) into Eq. (D.11) and rearranging, which allows us to rewrite Eq. (D.12) as

$$a_P = \frac{c_{\text{cat}} E_T X_S}{C_M + X_S}. \tag{D.15}$$

"1st-order." If we assume that the total enzyme population E_T is constant (which is the usual approach), then we can think of the transformation of substrate S into product P as a simple first-order reaction

$$S \xrightarrow{a_\mu = a_P} P,$$

with an *effective* propensity given by Eq. (D.15).[3]

In determining the scaling factor $g_{\mu S}$ for species S in this "reduced" reaction type, we follow the same approach as previously.

$$\frac{\Delta a_\mu}{a_\mu} = \frac{da_\mu}{dX_S}\frac{\Delta X_S}{a_\mu} = \frac{\Delta X_S}{X_S} - \frac{\Delta X_S}{C_M + X_S} = \left(\frac{C_M}{C_M + X_S}\right)\frac{\Delta X_S}{X_S},$$

$$\frac{|\Delta a_\mu|}{a_\mu} = \left(\frac{C_M}{C_M + X_S}\right)\frac{|\Delta X_S|}{X_S} = \epsilon_\mu.$$

This means that $\alpha_{\mu S}$ is the quantity in parentheses in the above expressions and $\zeta_{\mu S} = 1$ because there is only a single term. Thus, in general, we have

$$g_{\mu S}(t) = \frac{C_M}{C_M + X_S(t)}. \tag{D.16}$$

However, it is also evident that

$$\begin{aligned} g_{\mu S}(X_S = 1) &= \frac{C_M}{C_M + 1}, \\ g_{\mu S}(X_S \to \infty) &= 0. \end{aligned}$$

[3]Reactions 6 and 8 of Table 4.1 are of this type, with the total enzyme population E_T absorbed into the rate parameters k_8 and k_{11}, respectively.

Thus, we can assure that $|\Delta a_\mu| / a_\mu \leq \epsilon_\mu$ by choosing the upper bound,

$$g_{\mu S}(t) = g_{\mu S} = \frac{C_M}{C_M + 1}. \tag{D.17}$$

Furthermore, if $C_M \gg 1$, we can make the further simplification

$$g_{\mu S} \approx 1, \tag{D.18}$$

which is, interestingly, the same as for the corresponding elementary first-order reaction [cf., Eq. (D.1)].

It is important to recognize that this is the first example in which we have a $g_{\mu i}$ value that can be less than unity. When this is the case, it means that to effect a $(\epsilon \times 100)\%$ change in a_μ requires a *larger* change in X_i (X_S in this case). In fact, in this case, we have shown above that when X_i is very large a *much* larger change in X_i is required, to the point where, in the limit as $X_i \to \infty$, changing X_i has virtually no effect on a_μ. This means that, in principle, a large time step can be taken in such circumstances [cf., Eq.(3.30)]. However, by imposing Eq. (D.17) or (D.18), we are negating this possibility. It is not as clear as in previous examples, therefore, whether or not these simplifications should be used. Obviously, this will depend on the value of C_M and how X_S changes in time. In general, the appropriateness of using these simplifications will have to be determined on a case-by-case basis.

"2nd-order." If one assumes that (i) the total enzyme population E_T is *not* a constant, and (ii) that the lifetime of the enzyme-substrate complex $\{E{\cdot}S\}$ is nil (i.e., $X_{\{E\cdot S\}} = 0$), then the Michaelis-Menten mechanism can be thought of as a second-order reaction

$$E + S \xrightarrow{a_\mu = a_P(E_T = X_E)} E + P,$$

with an effective propensity

$$a_\mu = \frac{c_{\text{cat}} X_E X_S}{C_M + X_S}. \tag{D.19}$$

This is just Eq. (D.15) with $E_T = X_E$.[4]

Following the same procedure as above, we then have

$$\frac{\Delta a_\mu}{a_\mu} \approx \frac{\partial a_\mu}{\partial X_S}\frac{\Delta X_S}{a_\mu} + \frac{\partial a_\mu}{\partial X_E}\frac{\Delta X_E}{a_\mu} = \left(\frac{C_M}{C_M + X_S}\right)\frac{\Delta X_S}{X_S} + \frac{\Delta X_E}{X_E},$$

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \left(\frac{C_M}{C_M + X_S}\right)\frac{|\Delta X_S|}{X_S} + \frac{|\Delta X_E|}{X_E} = \epsilon_\mu.$$

Thus, $\alpha_{\mu S}$ is the quantity in parentheses in the above expressions and $\alpha_{\mu E} = 1$. Moreover, we choose $\zeta_{\mu S} = \zeta_{\mu E} = 2$ because there are two terms in the above expression and because the reaction is effectively of second order. This gives us

$$g_{\mu S}(t) = \frac{2C_M}{C_M + X_S(t)}, \tag{D.20}$$

$$g_{\mu E} = 2. \tag{D.21}$$

Again, however, $g_{\mu S}(t)$ has clear upper and lower bounds,

$$g_{\mu S}(X_S = 1) = \frac{2C_M}{C_M + 1},$$

$$g_{\mu S}(X_S \to \infty) = 0.$$

Thus, in the right circumstances, we can choose

$$g_{\mu S}(t) = g_{\mu S} = \frac{2C_M}{C_M + 1}. \tag{D.22}$$

Moreover, if $C_M \gg 1$, we can choose

$$g_{\mu S} \approx g_{\mu E} = 2, \tag{D.23}$$

which corresponds to those for the associated elementary second-order bimolecular reaction [cf., Eq. (D.2)]. Again, whether or not to use the simplifications in Eqs. (D.22) and (D.23) needs to be determined on a case-by-case basis.

[4]Reactions 3 and 4 of Table 4.1 are of this type, with the enzyme E being PLC^* and Ca, respectively.

Hill. Another type of enzymatic process that is common in biological systems is *gene transcription*. Transcription is the process by which messenger RNA (mRNA) molecules are synthesized from DNA sequences, or genes, through interaction with RNA polymerase. In general, transcription is driven by the action of "transcription factors," proteins that bind to the DNA and either enhance or inhibit the rate of mRNA production.

In modeling gene transcription, a common approach is to collapse the complicated machinery of the process down into a simplified view in terms of an enzymatic reaction

$$g \xrightarrow{a_\mu(X_p)} g + m,$$

where g is the gene, m is mRNA, and X_p is the population of the transcription-factor protein p. The propensity for the reaction then takes the functional form

$$a_\mu(X_p) = \frac{\bar{c} X_p^h}{C_{0.5}^h + X_p^h}, \tag{D.24}$$

which is known as the *Hill equation* [28]. $\bar{c}$ and $C_{0.5}$ are analogous to c_{cat} (really $c_{\text{cat}} E_T$) and C_M, respectively, of the Michaelis-Menten reaction discussed above.[5] Indeed, the above expression reduces to the Michaelis-Menten form when $h = 1$. h is known as the Hill coefficient and can, in general, take on any real value, i.e., $h \in \mathbb{R}$ [28]. If $h > 1$, then the action of the protein p is to enhance the rate of mRNA production, which is known as *positive cooperativity*. If $h < 1$, then p inhibits transcription, which is known as *negative cooperativity*.

Since the propensity of this reaction depends on the population of the protein p, our job here is to derive an expression for $g_{\mu p}(t)$. To do this, we again follow

[5] $C_{0.5}$ is labeled as such because when $X_p = C_{0.5}$, $a_\mu = 0.5\bar{c}$. When $h > 0$, $\bar{c}$ is the maximum possible value of a_μ [i.e., $X_p^h/(C_{0.5}^h + X_p^h) \to 1$ as $X_p \to \infty$]. Thus, $C_{0.5}$ is often referred to as the population (concentration) of half-maximal velocity.

the same procedure as previously. First, we have

$$\frac{\Delta a_\mu}{a_\mu} = \frac{da_\mu}{dX_p}\frac{\Delta X_p}{a_\mu} = h\frac{\Delta X_p}{X_p} - h\frac{X_p^h}{C_{0.5}^h + X_p^h}\frac{\Delta X_p}{X_p} = h\left(\frac{C_{0.5}^h}{C_{0.5}^h + X_p^h}\right)\frac{\Delta X_p}{X_p},$$

$$\frac{|\Delta a_\mu|}{a_\mu} = |h|\left(\frac{C_{0.5}^h}{C_{0.5}^h + X_p^h}\right)\frac{|\Delta X_p|}{X_p} = \epsilon_\mu.$$

This means that $\alpha_{\mu p}$ is $|h|$ times the term in parentheses and $\zeta_{\mu p}=1$ since there is only a single term. Thus,

$$g_{\mu p}(t) = \frac{|h|\, C_{0.5}^h}{C_{0.5}^h + X_p^h(t)}. \tag{D.25}$$

We also have two sets of upper and lower bounds, depending on whether $h>0$ or $h<0$,

$$\left.\begin{array}{rcc} g_{\mu p}(X_p = 1) & = & h\frac{C_{0.5}^h}{C_{0.5}^h+1} \\ g_{\mu p}(X_p \to \infty) & = & 0 \end{array}\right\} \quad if \quad h > 0,$$

$$\left.\begin{array}{rcc} g_{\mu p}(X_p = 1) & = & |h|\frac{1}{C_{0.5}^{|h|}+1} \\ g_{\mu p}(X_p \to \infty) & = & |h| \end{array}\right\} \quad if \quad h < 0.$$

Since $C_{0.5}$ is positive definite, when $h>0$, $C_{0.5}^h/(C_{0.5}^h+1)>0$. Thus, we can choose

$$g_{\mu p}(t) = g_{\mu p} = h\frac{C_{0.5}^h}{C_{0.5}^h + 1} \quad if \quad h > 0. \tag{D.26}$$

For the same reason, when $h<0$, $1/(C_{0.5}^{|h|}+1)<1$, meaning that we can choose

$$g_{\mu p}(t) = g_{\mu p} = |h| \quad if \quad h < 0. \tag{D.27}$$

We also see that if $C_{0.5}^h \gg 1$, then $C_{0.5}^h/(C_{0.5}^h+1) \to 1$, meaning that the upper bounds for $h>0$ and $h<0$ converge to the same value (obviously, $h=|h|$ when $h>0$). Thus, in this circumstance, we can use, regardless of whether h is positive or negative,

$$g_{\mu p} \approx |h| \quad if \quad C_{0.5}^h \gg 1. \tag{D.28}$$

Again, whether or not we can use these simplifications must be determined on a case-by-case basis.

Adair.[6] Although the Hill equation is widely used to model enzymatic processes in biology, including gene transcription, it is, in principle, a purely empirical and phenomenological equation [28], originally devised by Hill to fit data regarding the cooperative binding of oxygen to hemoglobin. As such, it cannot be derived. However, there do exist physically-based models of cooperative enzyme binding that reduce to the Hill equation in certain limits, thus providing a type of derivation. One such mechanism is that of Adair [28]. In Appendix E of this dissertation, we derive the Adair equation and discuss its relation to the Hill equation. Here, we simply present it and derive the exact and approximate expressions for $g_{\mu p}$.

We use the Adair equation when considering gene transcription from a set of protein-bound (and unbound) gene complexes $\{g, g{\cdot}p, g{\cdot}p_2, \ldots, g{\cdot}p_q\}$, where q is the total number of binding sites on the gene. The transcription reactions that we are concerned with are

$$\begin{aligned} g &\xrightarrow{a_\mu^{0,q}(X_p)} g+m, \\ g{\cdot}p &\xrightarrow{a_\mu^{1,q}(X_p)} g{\cdot}p+m, \\ &\vdots \\ g{\cdot}p_q &\xrightarrow{a_\mu^{q,q}(X_p)} g{\cdot}p_q+m. \end{aligned}$$

In general, we can think of the rates of mRNA production from n-bound genes containing q binding sites ($0 \leq n \leq q$). The Adair equation is then

$$a_\mu^{n,q}(X_p) = \frac{c_n^t g_T X_p^n \prod_{j=n}^{q-1} C_j}{\sum_{i=0}^{q} X_p^i \prod_{j=i}^{q-1} C_j}, \tag{D.29}$$

where g_T is the total number of genes (usually unity, not more than a few) and the C_j are dissociation constants,[7] i.e., $C_j \equiv c_{-j}/c_j$ $(j = 0, \ldots, q-1)$ for the reversible

[6] The material in this subsection has been adapted from Ref. [57].

[7] Note that dissociation constants are usually denoted as K_j and given in molar units. In our case, we are simply removing the molar units and defining $C_j \equiv K_j \times N_A \Omega$.

reaction

$$g{\cdot}p_j + p \underset{c_{-j}}{\overset{c_j}{\rightleftharpoons}} g{\cdot}p_{j+1}.$$

To clarify the connection to the Hill equation, consider

$$a_\mu^{2,2}(X_p) = \frac{c_2^t g_T X_p^2}{C_0 C_1 + C_1 X_p + X_p^2},$$

which is essentially a Hill equation with $h=2$, $\bar{c}=c_2^t g_T$ and $C_{0.5}=\sqrt{C_0 C_1}$ except for the second term in the denominator, which goes to zero as $X_p \to 0$ and is small compared to X_p^2 as $X_p \to \infty$. In other words, except for a small range of intermediate values of X_p, this expression gives very similar values to the associated Hill equation.[8]

Now, following the same procedure as above, we have[9]

$$\begin{aligned}\frac{|\Delta a_\mu^{n,q}|}{a_\mu^{n,q}} &= \left|\frac{da_\mu^{n,q}}{dX_p}\frac{\Delta X_p}{a_\mu^{n,q}}\right| \\ &= \left|n - q\frac{\sum_{i=1}^q \frac{i}{q} X_p^i \prod_{j=i}^{q-1} C_j}{\prod_{j=0}^{q-1} C_j + \sum_{i=1}^q X_p^i \prod_{j=i}^{q-1} C_j}\right| \frac{|\Delta X_p|}{X_p}.\end{aligned}$$

With $\zeta_{\mu p}^{n,q}=1$, this means that

$$g_{\mu p}^{n,q}(t) = \left|n - q\frac{\sum_{i=1}^q \frac{i}{q} X_p^i(t) \prod_{j=i}^{q-1} C_j}{\prod_{j=0}^{q-1} C_j + \sum_{i=1}^q X_p^i(t) \prod_{j=i}^{q-1} C_j}\right|. \tag{D.30}$$

[8]This is not necessarily true for larger values of q. Moreover, while there is an Adair equation for each protein-bound complex, Hill equations are usually the *only* equation used to describe the transcription dynamics. In other words, they attempt to account for transcription from all of the complexes simultaneously. This is why the Hill coefficient h, unlike n, is generally not an integer.

[9]We have pulled a factor of q out of the numerator in this equation in order to make more clear the value of $|\Delta a_\mu^{n,q}|/a_\mu^{n,q}$ in the infinite limit (see below).

We also have the upper and lower bounds[10]

$$\begin{aligned} g_{\mu p}^{n,q}(X_p = 1) &= \left| n - q \frac{\sum_{i=1}^{q-1} \frac{i}{q} \prod_{j=i}^{q-1} C_j + 1}{\sum_{i=0}^{q-1} \prod_{j=i}^{q-1} C_j + 1} \right|, \\ g_{\mu p}^{n,q}(X_p \to \infty) &= |n - q| \,. \end{aligned}$$

Analogous to before, we also have if all $\{C_j \gg 1\}$,[11]

$$g_{\mu p}^{n,q}(X_p = 1) \approx n.$$

In general, therefore, it may be possible, in the right circumstance, to use

$$g_{\mu p}^{n,q}(t) = g_{\mu p}^{n,q} = \max\left\{ g_{\mu p}^{n,q}(X_p = 1), |n - q| \right\}. \tag{D.31}$$

[10]For $g_{\mu p}^{n,q}(X_p = 1)$, we have changed the upper limit on the summations to avoid confusion (i.e., we have changed q to q–1 and added a '+1' to the end). Also, the value of $g_{\mu p}^{n,q}(X_p \to \infty)$ is as such because as $X_p \to \infty$ the terms in the summations at $i = q$ dominate (i.e., $X_p^q \gg X_p^{q-1} \gg X_p^{q-2}$, etc.), meaning that, in this limit, the numerator and denominator are approximately equal.

[11]By pulling a factor of q out of the numerator and leaving behind a factor of i/q within the summation, it becomes clear that the denominator will always exceed the numerator if all $\{C_j > 1\}$. Thus, in the limit that all $\{C_j \to \infty\}$, this term goes to zero.

APPENDIX E

The Adair reduction—Rapid equilibrium in gene expression[1]

As discussed briefly in Appendix D of this dissertation, gene transcription dynamics are often modeled in a coarse-grained manner using Hill equations. The Hill equation is phenomenological in nature, however, and cannot, in principle, be derived [28]. In this Appendix, we present the *Adair* reduction [28], a coarse-grained description of transcription dynamics that is derivable by applying the rapid-equilibrium assumption to a realistic reaction network that describes the transcription process. In the correct limits, the Adair equation reduces to the Hill equation, providing a type of derivation of the latter.

The specific problem that we are concerned with involves a gene g that has multiple binding sites, q in total, for the transcription-factor protein p. We assume that p binds as a monomer (i.e., it does not oligomerize in solution, which may not be realistic in many cases) and that successive binding and unbinding events can occur with different rates. The binding/unbinding reaction network thus looks like

$$\begin{aligned} g + p &\underset{c_{-0}}{\overset{c_0}{\rightleftharpoons}} g{\cdot}p, \\ g{\cdot}p + p &\underset{c_{-1}}{\overset{c_1}{\rightleftharpoons}} g{\cdot}p_2, \\ &\vdots \\ g{\cdot}p_{q-1} + p &\underset{c_{-(q-1)}}{\overset{c_{q-1}}{\rightleftharpoons}} g{\cdot}p_q. \end{aligned}$$

In principle, the unbound gene and each of the protein-bound gene complexes can transcribe mRNA at different rates.[2] The transcription process is often modeled

[1] The material in this Appendix is a generalized version of that presented in Ref. [57].

[2] If the action of p is to suppress transcription upon binding, then the rate of transcription from

as a simple enzymatic creation event. Thus, we consider the following reactions,

$$\begin{aligned} g &\xrightarrow{c_0^t} g + m, \\ g{\cdot}p &\xrightarrow{c_1^t} g{\cdot}p + m, \\ &\vdots \\ g{\cdot}p_q &\xrightarrow{c_q^t} g{\cdot}p_q + m. \end{aligned}$$

Obviously, the rate at which mRNA is produced strongly depends on the amount of time that the unbound gene and each of the gene complexes is populated, which, in turn, depends on the relative rates of gene-protein binding and unbinding.

The Adair approximation is to assume that each reversible binding and unbinding reaction pair is in equilibrium. This is known as the *partial equilibrium assumption* (PEA),[3] and allows us to derive effective expressions for the "occupancy probabilities" of each of the "gene species" $\{g, g{\cdot}p, \ldots, g{\cdot}p_q\}$. This means that we can remove explicit consideration of the gene-protein binding and unbinding reactions and only consider the transcription reactions with *effective* rate expressions that account for gene-protein binding and unbinding in a coarse-grained manner. These effective expressions are known as the Adair equations [28].

---

a bound complex may be significantly lower than that for the unbound gene but not necessarily zero. This phenomenon is known as transcriptional "leakage," and is often included in models of gene transcription (e.g., Ref. [33]).

[3]The PEA is different from the quasi-steady state assumption (QSSA) in that the PEA pertains to reaction rates while the QSSA pertains to species populations [16]. In particular, applying the PEA here to the reaction $g{+}p \underset{c_{-0}^b}{\overset{c_0^b}{\rightleftharpoons}} g{\cdot}p$ means that $c_0^b[g][p] = c_{-0}^b[g{\cdot}p]$. Conversely, in Appendix D, we used the QSSA to derive the Michaelis-Menten equation, which assumes that the population of the enzyme-substrate complex $\{E{\cdot}S\}$ does not change in time. This implies that the rate of its production via the reaction $E{+}S \xrightarrow{c_1} \{E{\cdot}S\}$ is equal to the rate of its consumption via the *two* reactions $\{E{\cdot}S\} \xrightarrow{c_{-1}} E{+}S$ and $\{E{\cdot}S\} \xrightarrow{c_{\rm cat}} E{+}P$.

In deriving the Adair equations, we begin by applying the PEA to the binding/unbinding reactions above, which gives us the q equations

$$\begin{aligned}
[g] &= C_0[g{\cdot}p]/X_p, \\
[g{\cdot}p] &= C_1[g{\cdot}p_2]/X_p, \\
&\vdots \\
[g{\cdot}p_{q-2}] &= C_{q-2}[g{\cdot}p_{q-1}]/X_p, \\
[g{\cdot}p_{q-1}] &= C_{q-1}[g{\cdot}p_q]/X_p.
\end{aligned}$$

Here, we have defined, for an n-bound gene, $C_n \equiv c_{-n}/c_n$ $(n = 0, \ldots, q-1)$ and $[\cdot]$ denotes the occupancy probability.[4] By successive substitution, we can then express each of these as functions of $[g{\cdot}p_q]$,

$$\begin{aligned}
[g] &= C_0C_1C_2 \ldots C_{q-1}[g{\cdot}p_q]X_p^{-q}, \\
[g{\cdot}p] &= C_1C_2 \ldots C_{q-1}[g{\cdot}p_q]X_p^{1-q}, \\
&\vdots \\
[g{\cdot}p_{q-2}] &= C_{q-2}C_{q-1}[g{\cdot}p_q]X_p^{-2}, \\
[g{\cdot}p_{q-1}] &= C_{q-1}[g{\cdot}p_q]X_p^{-1}.
\end{aligned}$$

In general, therefore, we can write

$$[g{\cdot}p_n] = [g{\cdot}p_q]X_p^n X_p^{-q} \prod_{j=n}^{q-1} C_j, \;\; n = 0, \ldots, q-1. \tag{E.1}$$

Now, the total number of genes, g_T, is

$$g_T = [g] + [g{\cdot}p] + \ldots + [g{\cdot}p_q] = \sum_{i=0}^{q} [g{\cdot}p_i]. \tag{E.2}$$

[4]Since there is often only a single copy of each gene, and never more than a few, we cannot really think in terms of "populations" of the gene species. Thus, for convenience, we use the notation $[\cdot]$. This is traditionally used to denote concentration, which is similar, in some sense, to probability. Note, however, that we retain the notation X_p for the population of p.

Substituting Eq. (E.1) into (E.2) and rearranging thus gives us an expression for $[g{\cdot}p_q]$,

$$[g{\cdot}p_q] = \frac{g_T}{\sum_{i=0}^{q} X_p^i X_p^{-q} \prod_{j=i}^{q-1} C_j}. \tag{E.3}$$

It is clear from the transcription reactions above that the rate of mRNA production from an n-bound gene with q binding sites is

$$a_\mu^{n,q} = c_n^t [g{\cdot}p_n]. \tag{E.4}$$

Thus, the Adair expression that we are looking for is obtained by substituting Eq. (E.3) into (E.1) and then Eq. (E.1) into (E.4). This gives us

$$a_\mu^{n,q}(X_p) = \frac{c_n^t g_T X_p^n \prod_{j=n}^{q-1} C_j}{\sum_{i=0}^{q} X_p^i \prod_{j=i}^{q-1} C_j}, \tag{E.5}$$

which is the same as Eq. (D.29) of Appendix D.

Let us consider a gene with two binding sites, i.e., $q=2$. In this circumstance, we have three gene species: g, $g{\cdot}p$ and $g{\cdot}p_2$. We have already shown in Appendix D that for $g{\cdot}p_2$ we can write

$$a_\mu^{2,2}(X_p) = \frac{c_2^t g_T X_p^2}{C_0 C_1 + C_1 X_p + X_p^2}, \tag{E.6}$$

which is reminiscent of a Hill equation with a Hill coefficient $h=2$ except for the second term in the denominator. For the unbound gene g, we have, similarly,

$$\begin{aligned} a_\mu^{0,2}(X_p) &= \frac{c_0^t g_T C_0 C_1}{C_0 C_1 + C_1 X_p + X_p^2} \\ &= \frac{c_0^t g_T X_p^{-2}}{(C_0 C_1)^{-1} + (C_0 X_p)^{-1} + X_p^{-2}}, \end{aligned} \tag{E.7}$$

which, written in the latter manner, is reminiscent of a *negative-cooperativity* Hill equation with a Hill coefficient $h=-2$. Thus, we see that the Adair formalism is able to capture the effects of both positive and negative cooperative binding. Interestingly, for $g{\cdot}p$ we have

$$a_\mu^{1,2}(X_p) = \frac{c_1^t g_T C_1 X_p}{C_0 C_1 + C_1 X_p + X_p^2}, \tag{E.8}$$

which is intermediate between Eqs. (E.6) and (E.7) and does not have an analogous Hill form. As alluded to in Appendix D, this is why the Hill coefficient h is generally not an integer. Hill equations are often used as approximate descriptions of the combined effects of Eqs. (E.6), (E.7) and (E.8).

APPENDIX F

Histogram smoothing, histogram distance and self distance[1]

For a set of N data points $\{x_1, x_2, \ldots, x_N\}$, the total number falling within a discrete interval $[x, x+\Delta)$ can be formally written as $\sum_{i=1}^{N} \int_x^{x+\Delta} \delta(x_i - x')dx'$, where $\delta(x_i - x')$ is the Dirac delta function (Appendix A) and the integral equals unity if x_i lies within $[x, x+\Delta)$ and zero otherwise. A "histogram density" can be obtained by dividing this quantity by $N\Delta$ and taking the limit as $\Delta \to 0$,

$$\widehat{h}(x) = \lim_{\Delta \to 0} \frac{1}{N\Delta} \sum_{i=1}^{N} \int_x^{x+\Delta} \delta(x_i - x')dx'. \tag{F.1}$$

The 'hat' in $\widehat{h}(x)$ signifies that this quantity is a statistical *estimator* of the true histogram density

$$h(x) = \lim_{N \to \infty} \widehat{h}(x). \tag{F.2}$$

A "smoothed" histogram can be obtained by approximating the delta function in (F.1) by a finite-width Gaussian $\kappa \exp\left(\frac{-(x_i - x')^2}{2\sigma^2}\right)$, where κ is a normalization constant and σ^2 is the (user-defined) variance. Substituting into (F.1), noting that to first order $\int_x^{x+\Delta} e^{-u^2} du = \Delta e^{-x^2}$, and normalizing, gives

$$\widehat{h}(x) \approx \frac{1}{\sqrt{2\pi}\sigma N} \sum_{i=1}^{N} \exp\left(\frac{-(x_i - x)^2}{2\sigma^2}\right). \tag{F.3}$$

All smoothed histograms presented in this dissertation have been obtained using this expression.[2]

[1] The material in this Appendix is adapted from Ref. [56].

[2] Equation (F.3) is also known as kernel density estimation or the Parzen window method [82].

In order to quantitatively compare results obtained via different simulation methods (i.e., PLA, SSA, etc.), we use the "histogram distance" discussed by Cao and Petzold [23]. We first define the quantity $\delta h_x \equiv h_1(x) - h_2(x)$. The histogram distance, D, is then defined as

$$D \equiv \frac{1}{2} \int_x |\delta h_x| dx, \tag{F.4}$$

where the factor of $1/2$ assures that D lies within [0,1), with 0 constituting a perfect fit and 1 a complete mismatch.

It is important to recognize that D is defined in Eq. (F.4) in terms of the *true* histogram densities $h_1(x)$ and $h_2(x)$. In practice, we only have their estimators and can thus only calculate an estimated value of D. As a result, a certain amount of statistical uncertainty is associated with the comparison of histograms. In order to quantify this uncertainty, Cao and Petzold [23] introduced the "self distance," D^{self}, which can be thought of as the distance between the estimator $\widehat{h}(x)$ and the *true* density $h(x)$. Expressions for the upper bounds on the mean and variance of D^{self} are presented in Ref. [23] in terms of the number of bin intervals K used to generate the histograms. However, since we are using Eq. (F.3) to generate histograms rather than a counting procedure, we must derive alternate expressions.

We begin by defining, as before, $\delta h_x^{\text{self}} \equiv \widehat{h}(x) - h(x)$. The self distance is then

$$D^{\text{self}} \equiv \frac{1}{2} \int_x |\delta h_x^{\text{self}}| dx. \tag{F.5}$$

Following Cao and Petzold [23], we then note that the number of points falling within the interval $[x, x + \Delta)$ is a binomial random variable $B(p_x, N)$, where p_x is the success probability. $\widehat{h}(x)$ can thus be written as

$$\widehat{h}(x) = \lim_{\Delta \to 0} B(p_x, N)/N\Delta = B(dp_x, N)/Ndx. \tag{F.6}$$

Since the mean $\langle B(p_x,N)\rangle = Np_x$ and the variance $\mathrm{var}\{B(p_x,N)\} = Np_xq_x$ ($q_x \equiv 1-p_x$), the mean and variance of $\widehat{h}(x)$ are dp_x/dx and dp_xdq_x/Ndx^2, respectively. With $dp_x = h(x)dx$ and $dq_x = 1-h(x)dx$, the mean and variance of $\delta h_x^{\mathrm{self}}$ are thus

$$\left\langle \delta h_x^{\mathrm{self}} \right\rangle = 0, \tag{F.7}$$

$$\mathrm{var}\{\delta h_x^{\mathrm{self}}\} = [h(x)dx][1-h(x)dx]/Ndx^2 \approx \widehat{h}(x)/Ndx, \tag{F.8}$$

where the last line utilizes the histogram density estimator $\widehat{h}(x)$ and assumes that $\widehat{h}(x)dx \ll 1$. For large N, $\delta h_x^{\mathrm{self}}$ can thus be approximated as a *normal random variable* with mean zero and variance $\widehat{h}(x)/Ndx$. This means that

$$\frac{\delta h_x^{\mathrm{self}}}{\sqrt{\widehat{h}(x)/Ndx}}$$

is approximately *standard normal* and

$$\frac{|\delta h_x^{\mathrm{self}}|}{\sqrt{\widehat{h}(x)/Ndx}}$$

is approximately *chi distributed* with one degree of freedom (DOF). Since the mean and variance of a chi random variable with one DOF are $\sqrt{2/\pi}$ and $(\pi-2)/\pi$, respectively, we have

$$\left\langle |\delta h_x^{\mathrm{self}}| \right\rangle \approx \sqrt{\frac{2}{N\pi}\frac{\widehat{h}(x)}{dx}}, \tag{F.9}$$

$$\mathrm{var}\left\{|\delta h_x^{\mathrm{self}}|\right\} \approx \left(\frac{\pi-2}{N\pi}\right)\frac{\widehat{h}(x)}{dx}. \tag{F.10}$$

Finally, using the identities [44]

$$\left\langle \textstyle\int_x f(x)dx \right\rangle = \displaystyle\int_x \langle f(x)\rangle\, dx,$$

$$\mathrm{var}\left\{\textstyle\int_x f(x)dx\right\} \le \left(\displaystyle\int_x \sqrt{\mathrm{var}\{f(x)\}}dx\right)^2,$$

we get

$$\left\langle D^{\mathrm{self}} \right\rangle \approx \frac{1}{2}\sqrt{\frac{2}{N\pi}}\int_x \sqrt{\widehat{h}(x)dx}, \tag{F.11}$$

$$\mathrm{var}\left\{D^{\mathrm{self}}\right\} \lesssim \frac{1}{4}\left(\frac{\pi-2}{N\pi}\right)\left(\int_x \sqrt{\widehat{h}(x)dx}\right)^2 \tag{F.12}$$

[note that the dx are within the square-roots in Eqs. (F.11) and (F.12)]. In practice, we calculate $\left\langle D_{\text{ref}}^{\text{self}} \right\rangle$, the mean self distance for a *reference* histogram, generally obtained using the SSA. This value then tells us that any histogram with a distance $D < 2\left\langle D_{\text{ref}}^{\text{self}} \right\rangle$ cannot be distinguished, statistically speaking, from the reference histogram.[3] Note that the expression for $\text{var}\{D^{\text{self}}\}$ is included here for completeness but it is of little practical value.

[3]The reason why we use $2\left\langle D_{\text{ref}}^{\text{self}} \right\rangle$ as the criterion for statistical indistinguishability is that two sample histograms can have identical self distances but from opposite sources. For example, one can be slightly taller and thinner, and the other slightly shorter and wider, than the true histogram. Thus, two sample histograms can be as dissimilar as *twice* the mean self distance and still be indistinguishable from the true histogram. In the Appendix of Ref. [56], it was incorrectly stated that two histograms can be considered distinct if they differ by only a single $\left\langle D_{\text{ref}}^{\text{self}} \right\rangle$.

BIBLIOGRAPHY

[1] D. F. Anderson. Incorporating postleap checks in tau-leaping. *J. Chem. Phys.*, 128:054103, 2008.

[2] A. P. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, 149:1633–1648, 1998.

[3] A. Auger, P. Chatelain, and P. Koumoutsakos. *R*-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *J. Chem. Phys.*, 125:084103, 2006.

[4] N. Barkai and S. Leibler. Biological rhythms – Circadian clocks limited by noise. *Nature*, 403:267–268, 2000.

[5] S. A. Benner and A. M. Sismour. Synthetic biology. *Nat. Rev. Genet.*, 6:533–543, 2005.

[6] D. Bernstein. Simulating mesoscopic reaction-diffusion systems using the Gillespie algorithm. *Phys. Rev. E*, 71:041103, 2005.

[7] M. J. Berridge, M. D. Bootman, and P. Lipp. Calcium – a life and death signal. *Nature*, 395:645–648, 1998.

[8] W. J. Blake, G. Balázsi, M. A. Kohanski, F. J. Isaacs, K. F. Murphy, Y. Kuang, C. R. Cantor, D. R. Walt, and J. J. Collins. Phenotypic consequences of promoter-mediated transcriptional noise. *Mol. Cell*, 24:853–865, 2006.

[9] J. L. Blue, I. Beichl, and F. Sullivan. Faster Monte Carlo simulations. *Phys. Rev. E*, 51:R867–R868, 1995.

[10] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *J. Comput. Phys.*, 17:10–18, 1975.

[11] R. Bundschuh, F. Hayot, and C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophys. J.*, 84:1606–1615, 2003.

[12] K. Burrage and T. Tian. Poisson Runge-Kutta methods for chemical reaction systems. In Y. Lu, W. Sun, and T. Tang, editors, *Third International Workshop on Scientific Computing and Applications*, Advances in Scientific Computing and Applications, pages 82–96. Science Press, 2004.

[13] K. Burrage, T. Tian, and P. Burrage. A multi-scaled approach for simulating chemical reaction systems. *Prog. Biophys. Mol. Biol.*, 85:217–234, 2004.

[14] X. Cai and Z. Xu. K-leap method for accelerating stochastic simulation of coupled chemical reactions. *J. Chem. Phys.*, 126:074102, 2007.

[15] E. Callaway. Bacteria's new bones. *Nature*, 451:124–126, 2008.

[16] Y. Cao, D. Gillespie, and L. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *J. Comput. Phys.*, 206:395–411, 2005.

[17] Y. Cao, D. T. Gillespie, and L. R. Petzold. Accelerated stochastic simulation of the stiff enzyme-substrate reaction. *J. Chem. Phys.*, 123:144917, 2005.

[18] Y. Cao, D. T. Gillespie, and L. R. Petzold. Avoiding negative populations in explicit Poisson tau-leaping. *J. Chem. Phys.*, 123:054104, 2005.

[19] Y. Cao, D. T. Gillespie, and L. R. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122:014116, 2005.

[20] Y. Cao, D. T. Gillespie, and L. R. Petzold. Efficient step size selection for the tau-leaping method. *J. Chem. Phys.*, 124:044109, 2006.

[21] Y. Cao, D. T. Gillespie, and L. R. Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *J. Chem. Phys.*, 126:224101, 2007.

[22] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121:4059–4067, 2004.

[23] Y. Cao and L. Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *J. Comput. Phys.*, 212:6–24, 2006.

[24] Y. Cao, L. R. Petzold, M. Rathinam, and D. T. Gillespie. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 121:12169–12178, 2004.

[25] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *J. Chem. Phys.*, 122:024112, 2005.

[26] A. Colman-Lerner, A. Gordon, E. Serra, T. Chin, O. Resnekov, D. Endy, C. G. Pesce, and R. Brent. Regulated cell-to-cell variation in a cell-fate decision system. *Nature*, 437:699–706, 2005.

[27] H. Conzelmann, D. Fey, and E. D. Gilles. Exact model reduction of combinatorial reaction networks. *BMC Syst. Biol.*, 2:78, 2008.

[28] A. Cornish-Bowden. *Fundamentals of Enzyme Kinetics, 3rd Ed.* Portland Press Ltd., London, U.K., 2004.

[29] C. J. Dixon, N. M. Woods, K. S. R. Cuthbertson, and P. H. Cobbold. Evidence for two Ca^{2+}-mobilizing purinoceptors on rat hepatocytes. *Biochem. J.*, 269:499–502, 1990.

[30] W. E, D. Liu, and E. Vanden-Eijnden. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *J. Chem. Phys.*, 123:194107, 2005.

[31] J. Elf and M. Ehrenberg. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *IEE Syst. Biol.*, 1:230–236, 2004.

[32] R. J. Ellis. Macromolecular crowding: obvious but underappreciated. *Trends Biochem. Sci.*, 26:597–604, 2001.

[33] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

[34] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 2002.

[35] D. Endy and R. Brent. Modelling cellular behaviour. *Nature*, 409:391–395, 2001.

[36] M. Falcke. Reading the patterns in living cells – the physics of Ca^{2+} signaling. *Adv. Phys.*, 53:255–440, 2004.

[37] N. Fedoroff and W. Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 2002.

[38] J. Feret, V. Danos, J. Krivine, R. Harmer, and W. Fontana. Internal coarse-

graining of molecular systems. *Proc. Natl. Acad. Sci. U.S.A*, 106:6453–6458, 2009.

[39] T. Fricke and D. Wendt. The Markoff automaton: A new algorithm for simulating the time-evolution of large stochastic dynamic systems. *Int. J. Mod. Phys. C*, 6:277–306, 1995.

[40] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104:1876–1889, 2000.

[41] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.

[42] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–2361, 1977.

[43] D. T. Gillespie. Monte Carlo simulation of random walks with residence time dependent transition probability rates. *J. Comput. Phys.*, 28:395–407, 1978.

[44] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists*. Academic, San Diego, 1992.

[45] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.

[46] D. T. Gillespie. The chemical langevin equation. *J. Chem. Phys.*, 113:297–306, 2000.

[47] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115:1716–1733, 2001.

[48] D. T. Gillespie. The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction. *J. Phys. Chem. A*, 106:5063–5071, 2002.

[49] D. T. Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.

[50] D. T. Gillespie and L. R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.*, 119:8229–8234, 2003.

[51] A. Goldbeter. Computational approaches to cellular rhythms. *Nature*, 420:238–245, 2002.

[52] D. Gonze and A. Goldbeter. Circadian rhythms and molecular noise. *Chaos*, 16:026110, 2006.

[53] D. Gonze, J. Halloy, and A. Goldbeter. Robustness of circadian rhythms with respect to molecular noise. *Proc. Natl. Acad. Sci. U.S.A*, 99:673–678, 2002.

[54] J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J. Chem. Phys.*, 122:184102, 2005.

[55] M. E. Gracheva, R. Toral, and J. D. Gunton. Stochastic effects in intercellular calcium spiking in hepatocytes. *J. Theor. Biol.*, 212:111–125, 2001.

[56] L. A. Harris and P. Clancy. A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics. *J. Chem. Phys.*, 125:144107, 2006.

[57] L. A. Harris, A. M. Piccirilli, E. R. Majusiak, and P. Clancy. Quantifying stochastic effects in biochemical reaction networks using partitioned leaping. *Phys. Rev. E*, 79:051906, 2009.

[58] J. Hasty, D. McMillen, and J. J. Collins. Engineered gene circuits. *Nature*, 420:224–230, 2002.

[59] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins. Computational studies of gene regulatory networks: *In Numero* molecular biology. *Nat. Rev. Genet.*, 2:268–279, 2001.

[60] J. Hattne, D. Fange, and J. Elf. Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, 21:2923–2924, 2005.

[61] M. Kærn, T. C. Elston, W. J. Blake, and J. J. Collins. Stochasticity in gene expression: From theories to phenotypes. *Nat. Rev. Genet.*, 6:451–464, 2005.

[62] A. J. Kearsley, W. E. Wallace, J. Bernal, and C. M. Guttman. A numerical method for mass spectral data analysis. *Appl. Math. Lett.*, 18:1412–1417, 2005.

[63] T. B. Kepler and T. C. Elston. Stochasticity in transcriptional regulation: Origins, consequences, and mathematical representations. *Biophys. J.*, 81:3116–3136, 2001.

[64] M. Koschorreck and E. D. Gilles. ALC: automated reduction of rule-based models. *BMC Syst. Biol.*, 2:91, 2008.

[65] U. Kummer, B. Drajnc, J. Pahle, A. K. Green, C. J. Dixon, and M. Marhl. Transition from stochastic to deterministic behavior in calcium oscillations. *Biophys. J.*, 89:1603–1611, 2005.

[66] U. Kummer, L. F. Olsen, C. J. Dixon, A. K. Green, E. Bornberg-Bauer, and G. Baier. Switching from simple to complex oscillations in calcium signaling. *Biophys. J.*, 79:1188–1195, 2000.

[67] A. Leier, T. T. Marquez-Lago, and K. Burrage. Generalized binomial τ-leap method for biochemical kinetics incorporating both delay and intrinsic noise. *J. Chem. Phys.*, 128:205107, 2008.

[68] H. Li and L. Petzold. Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. Technical report, Department of Computer Science, University of California, Santa Barbara, 2006. http://www.engr.ucsb.edu/_cse.

[69] N. Maheshri and E. K. O'Shea. Living with noisy genes: How cells function reliably with inherent variability in gene expression. *Annu. Rev. Biophys. Biomol. Struct.*, 36:413–434, 2007.

[70] T. T. Marquez-Lago and K. Burrage. Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *J. Chem. Phys.*, 127:104101, 2007.

[71] E. A. Mastny, E. L. Haseltine, and J. B. Rawlings. Stochastic simulation of catalytic surface reactions in the fast diffusion limit. *J. Chem. Phys.*, 125:194715, 2006.

[72] E. A. Mastny, E. L. Haseltine, and J. B. Rawlings. Two classes of quasi-steady-state model reductions for stochastic kinetics. *J. Chem. Phys.*, 127:094106, 2007.

[73] H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. U.S.A*, 94:814–819, 1997.

[74] H. H. McAdams and A. Arkin. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet.*, 15:65–69, 1999.

[75] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Biol. Chem.*, 30:39–49, 2006.

[76] D. A. McQuarrie. Stochastic approach to chemical kinetics. *J. Appl. Prob.*, 4:413–478, 1967.

[77] J. S. Milton and J. C. Arnold. *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences, 3rd Ed.* McGraw-Hill Inc., New York, N.Y., 1995.

[78] M. J. Morelli, R. J. Allen, S. Tănase-Nicola, and P. R. ten Wolde. Eliminating fast reactions in stochastic simulations of biochemical networks: A bistable genetic switch. *J. Chem. Phys.*, 128:045105, 2008.

[79] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124:044104, 2006.

[80] B. Munsky and M. Khammash. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *J. Comput. Phys.*, 226:818835, 2007.

[81] J. Pahle. Biochemical simulations: stochastic, approximate stochastic and hybrid approaches. *Brief. Bioinform.*, 10:53–64, 2009.

[82] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.

[83] X. Peng, W. Zhou, and Y. Wang. Efficient binomial leap method for simulating chemical kinetics. *J. Chem. Phys.*, 126:224109, 2007.

[84] M. F. Pettigrew and H. Resat. Multinomial tau-leaping method for stochastic kinetic simulations. *J. Chem. Phys.*, 126:084101, 2007.

[85] J. D. Plummer and P. B. Griffin. Challenges for predictive process simulation in sub 0.1 μm silicon devices. *Nucl. Instr. Meth. Phys. Res. B*, 102:160, 1995.

[86] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing, 2nd Ed.* Cambridge University Press, New York, NY, 1999.

[87] J. Puchałka and A. M. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophys. J.*, 86:1357–1372, 2004.

[88] C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *J. Chem. Phys.*, 118:4999–5010, 2003.

[89] C. V. Rao, D. M. Wolf, and A. P. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420:231–237, 2002.

[90] J. M. Raser and E. K. O'Shea. Noise in gene expression: Origins, consequences, and control. *Science*, 309:2010–2013, 2005.

[91] M. Rathinam and H. El Samad. Reversible-equivalent-monomolecular tau: A leaping method for "small number and stiff" stochastic chemical systems. *J. Comput. Phys.*, 224:897–923, 2007.

[92] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chem. Phys.*, 119:12784–12794, 2003.

[93] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Consistency and stability of tau-leaping schemes for chemical reaction systems. *Multiscale Model. Simul.*, 4:867–895, 2005.

[94] D. Rossinelli, B. Bayati, and P. Koumoutsakos. Accelerated stochastic and hybrid methods for spatial simulations of reaction-diffusion systems. *Chem. Phys. Lett.*, 451:136–140, 2008.

[95] S. Roy and A. Asenov. Where do the dopants go? *Science*, 309:388, 2005.

[96] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J. Chem. Phys.*, 122:054103, 2005.

[97] H. Salis and Y. N. Kaznessis. An equation-free probabilistic steady-state approximation: Dynamic application to the stochastic simulation of biochemical reaction networks. *J. Chem. Phys.*, 123:214106, 2005.

[98] A. Samant and D. G. Vlachos. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: A multiscale Monte Carlo algorithm. *J. Chem. Phys.*, 123:144114, 2005.

[99] M. S. Samoilov and A. P. Arkin. Deviant effects in molecular reaction pathways. *Nat. Biotechnol.*, 24:1235–1240, 2006.

[100] M. S. Samoilov, G. Price, and A. P. Arkin. From fluctuations to phenotypes: The physiology of noise. *Sci. STKE*, 2006:re17, 2006.

[101] T. P. Schulze. Efficient kinetic Monte Carlo simulation. *J. Comput. Phys.*, 227:2455–2462, 2008.

[102] S. Schuster, M. Marhl, and T. Höfer. Modelling of simple and complex calcium oscillations. From single-cell responses to intercellular signalling. *Eur. J. Biochem.*, 269:1333–1355, 2002.

[103] J. P. Sethna. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*. Oxford Univ. Press, Oxford, U.K., 2006. [Exercise (8.11)].

[104] J. P. Sethna and C. R. Myers. *Entropy, Order Parameters, and Complexity* computer exercises: Hints and software, 2004.

[105] T. Shibata. Reducing the master equations for noisy chemical reactions. *J. Chem. Phys.*, 119:6629–6634, 2003.

[106] N. A. Sinitsyn, N. Hengartner, and I. Nemenman. Adiabatic coarse-graining and simulations of stochastic biochemical networks. *Proc. Natl. Acad. Sci. U.S.A*, 106:10546–10551, 2009.

[107] A. Slepoy, A. P. Thompson, and S. J. Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *J. Chem. Phys.*, 128:205101, 2008.

[108] D. Sprinzak and M. B. Elowitz. Reconstruction of genetic circuits. *Nature*, 438:443–448, 2005.

[109] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121:10356–10364, 2004.

[110] A. S. Tomlin, T. Turanyi, and M. J. Pilling. Mathematical tools for the construction, investigation and reduction of combustion mechanisms. In M. J. Pilling, editor, *Low Temperature Combustion and Autoignition*, pages 293–437. Elsevier, Amsterdam, 1997.

[111] L. Trinkle-Mulcahy and A. I. Lamond. Nuclear functions in space and time: Gene expression in a dynamic, constrained environment. *FEBS Lett.*, 582:1960–1970, 2008.

[112] J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proc. Natl. Acad. Sci. U.S.A*, 99:5988–5992, 2002.

[113] D. G. Vlachos. Temporal coarse-graining of microscopic-lattice kinetic Monte Carlo simulations via τ leaping. *Phys. Rev. E*, 78:046713, 2008.

[114] H. Wagner, M. Möller, and K. Prank. COAST: Controllable approximative stochastic reaction algorithm. *J. Chem. Phys.*, 125:174104, 2006.

[115] W. E. Wallace, A. J. Kearsley, and C. M. Guttman. An operator-independent approach to mass spectral peak identification and integration. *Anal. Chem.*, 76:2446–2452, 2004.

[116] Z. Xu and X. Cai. Unbiased τ-leap methods for stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 128:154112, 2008.

[117] J. Yang, M. I. Monine, J. R. Faeder, and W. S. Hlavacek. Kinetic Monte Carlo method for rule-based modeling of biochemical networks. *Phys. Rev. E*, 78:031910, 2008.

[118] H.-X. Zhou, G. Rivas, and A. P. Minton. Macromolecular crowding and confinement: Biochemical, biophysical, and potential physiological consequences. *Annu. Rev. Biophys.*, 37:375–397, 2008.