Adaptive Time-stepping in the Numerical Solution of the Reaction-Diffusion Master Equation

by

Jill Marie Anderson Padgett

Bachelor of Education, University of Toronto, 2012

Bachelor of Science, Computer Science, South Dakota School of Mines & Technology, 2007

Bachelor of Science, Applied and Computational Mathematics, South Dakota School of Mines & Technology, 2007

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Applied Mathematics

Toronto, Ontario, Canada, 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

Adaptive Time-stepping in the Numerical Solution of the Reaction-Diffusion Master Equation

Master of Science 2015

Jill Marie Anderson Padgett

Applied Mathematics

Ryerson University

Stochastic modeling and simulation are essential tools for studying cellular processes. The dynamics of spatially heterogeneous biochemical systems with species in low amounts is governed by a discrete, stochastic model, the Reaction-Diffusion Master Equation (RDME). The Inhomogeneous Stochastic Simulation Algorithm (ISSA) is an exact numerical method for the RDME, but is prohibitively slow as it simulates every chemical reaction and diffusion event. To overcome this difficulty, an approximate strategy, the tau-leaping scheme, was developed that steps over multiple reactions and diffusion events. Mathematical models of biochemical systems are often prone to stiffness, thus computationally challenging. In this thesis, we propose an adaptive time-stepping scheme for the tau-leaping method for the RDME. This strategy is compared to the ISSA, for several models of interest. The numerical results show that the proposed adaptive technique significantly speeds-up the simulation, while maintaining excellent accuracy of the numerical solution.

Acknowledgements

I would like to express my sincerest thanks and gratitude to the individuals who assisted with this thesis. First of all, I would like to thank my supervisor Prof. Silvana Ilie; her ongoing support and guidance were integral to the completion of both the research and this document. Next, I would like recognize my family, in particular my husband Neil, for the constant encouragement and positive support provided to me. Afterwards, I would like to thank Steve Kanellis for all of his technical support. Finally, to my examining committee members, Dr. Jean-Paul Pascal, Dr. Katrin Rohlf, and Dr. Dejan Delic, I want to express my great appreciation for the time set aside to read the thesis and attend the thesis defense. Additionally, I would like to acknowledge the Ontario Graduate Scholarship program's financial support of this work.

Contents

List of Figures

List of Appendices

Chapter 1

Introduction

The area of Computational and Systems Biology has seen a rise in popularity in recent years because advancements in genetic and molecular biology have led to a rapid influx of genomic data [11]. Mathematical modeling and simulations provide an invaluable tool for analyzing these complex biological systems under a variety of experimental conditions [11, 52]. Among the approaches used to model biochemical systems are ordinary differential equations, partial differential equations, and stochastic master equations.

Biochemical computer simulations test hypothesis with *in silico* experiments. The aim of such simulations is to test the validity of underlying model assumptions by predicting the biochemical system's dynamics. Biochemical modeling and simulation development is an iterative process. Starting with an existing biochemical system, scientists and mathematicians determine which mathematical model is most appropriate, then develop an associated computer model. The output of such a model is then compared to the existing experimental observations. If inconsistencies between the simulation and experimental data are observed, this may indicate incomplete knowledge of the biochemical system under consideration or a potential issue with the simulation. Thus, more accurate measurements or refinements to either the model or the simulation are required. Once the simulation passes validation, it can be used to test sensitivity to biological parameters, to provide predictions to be tested by *in vitro* and *in vivo* studies, or to infer relationships that may not be observable experimentally [36].

Cellular processes are typically modeled as systems of chemical reactions. The mathematical model that accurately represents the system under consideration depends on the molecular population size and the nature of the biochemical reactions in the system. When all molecular or *species* populations occur in large amounts, the deterministic biochemical model of the reaction-rate equations can effectively model the biochemical kinetics. The reaction-rate equations are based on the *Law of Mass Action* which gives a relationship between the reaction rates and the molecular concentration. On the level of a single cell, the assumption that all molecular species have large amounts is often unjustified. Certain key reactants like DNA or RNA occur in such low numbers that discrete models are required [52]. These small population numbers lead to random fluctuation within the cellular system which have been observed experimentally [13, 5]. Consequently, discrete, stochastic mathematical models are required to reproduce the dynamics of these cellular systems.

A biological application that has greatly benefited from stochastic modeling is gene regulatory networks [29, 31]. *Gene expression* is defined as the "production of an observable molecular product (RNA or protein)" [35]. When considering the pathway from DNA to protein in a eukaroytic cell, gene expression can be regulated at six different stages including: *transcriptional control*, how often a gene is transcribed into RNA, *RNA processing control*, the splicing and processing of RNA transcripts into mRNA, and *translational control*, selecting which mRNA are translated into proteins. Transcriptional controls are paramount for most genes as they prevent the synthesis of unnecessary intermediates. However, transcriptional controls also aid in the transcription of genes. Transcriptional control is provided by *gene regulatory proteins* which bind to specific DNA sequences to influence the transcription of the gene. Gene regulatory proteins are often produced from separate genes, but can be produced by the gene they regulate. This network of gene regulatory proteins and the genes they govern form a *gene regulatory network* [35]. More complex gene regulatory networks describe the regulatory interactions between DNA, RNA, proteins, and small molecules [11]. Low molecular numbers are inherent to gene regulatory networks and certain cellular events, like the start and finish of transcription, may have timing fluctuation. Thus, discrete and stochastic mathematical models will generally provide the most accurate representation of the system's behaviour.

Arkin et. al. [2] presented the first successful application of discrete, stochastic biochemical modeling to an in vivo experiment. The study focused on the phage λ lysis-lysogeny genetic regulatory network. In their experiment, *Escherichia coli* cells were infected with λ-phage, a virus that induces either lysis[1] or lysogeny [2]. They found that the fraction of cells selecting the lysogenic pathway predicted from the stochastic kinetic model was consistent with experimental observations.

The discrete stochastic model that Arkin et. al. [2] used is known as the *Chemical Master Equation* (CME) [41, 22]. The CME is the most refined mathematical model of homogeneous or *well-stirred* biochemical kinetics and is too complex to solve analytically for most biochemical systems of practical interest. Daniel Gillespie proposed the first exact numerical simulation algorithm for the CME [20, 21]. Known as the *Stochastic Simulation Algorithm* (SSA) or Gillespie's algorithm, this Monte Carlo method simulates individual realizations of the biochemical system's state which agree with the statistics of the CME. Other exact CME simulation algorithms include the *First Reaction Method* [20] and Gibson and Bruck's *Next Reaction Method* [19].

All of these exact simulation algorithms require the simulation of every chemical reaction, one at a time. When biochemical systems include many fast reactions or large population numbers in one or more species, the average time between reactions decreases to a point where it is no longer computationally feasible to simulate individual reactions [25]. To improve computation efficiency, Gillespie [23] proposed tau-leaping methods, approximate discrete stochastic strategies designed to improve simulation times by sacrificing a certain level of simulation detail. This technique uses a pre-determined step size τ to leap over many reactions using Poisson random variables. A poor choice of τ can lead to inaccuracy in the final solution or negative numbers in some molecular species. As such, a series of τ selection procedures have been proposed [26, 8, 9] as well as replacing the Poisson random variables with binomial random variables [51, 10]. Gillespie [23] also proposed the Chemical Langevin Equation, a continuous stochastic method, as the bridge model between the

[1] the breakdown of the cell membrane

[2] the integration of the virus's nucleic acid into the host's DNA

CME and the deterministic reaction-rate equations. Additionally, hybrid methods, which combine two or more of discrete stochastic, continuous stochastic, and continuous deterministic models, have been proposed to improve simulation time for the CME [28, 43, 47, 46, 48, 54].

Moreover, many critical cellular processes such as chemotaxis [50], intracellular signal pathways [3], and cell division [30] rely on the heterogeneous distribution of the eukaryotic cell. Thus, mathematical models for these biochemical systems must include a spatial component to accurately replicate the underlying system's behaviour. Fange and Elf's [16] study of the Min system in *Escherichia coli* illustrates the requirement to include spatial characteristics. They found that experimental observations of two mutant strains only correlated with the results of the simulation model that incorporated randomness and discreteness along with the spatial characteristics of the process. By partitioning the biochemical system's volume into subvolumes or *voxels* and allowing diffusion to occur between the voxels, we can extend the Chemical Master Equation to the Reaction-Diffusion Master Equation (RDME) model [18]. This popular mesoscopic model accurately describes the kinetics of heterogeneous biochemical systems and is generally intractable for real-life applications.

Similar to the CME, a number of exact numerical simulation algorithms have been developed to solve the RDME including: Stundzia and Lumsden's extension of the SSA [49], Elf and Ehrenberg's extension of the Next Reaction Method to the *Next-Subvolume Method* [12], *MesoRD* [16], and [34]. All of these methods are prohibitively slow in general [25], because they simulate every reaction and diffusion event. Furthermore, as spatial discretization becomes more refined, diffusive transport events can dominate the system dynamics [25]. To overcome these limitations, approximation techniques like tau-leaping have been extended to solve the RDME [40, 45, 37]. Hybrid methods have also been proposed to improve simulation time for the RDME [45, 38, 17].

A biochemical system is considered *stiff* if explicit numerical methods require very small step sizes to maintain solution stability. Mathematical models of heterogeneous biochemical systems are particularly prone to stiffness; as they generally contain a combination of fast and slow reactions leading to multiple time-scales. Additionally, as voxel size decreases, the number of diffusion events increases dramatically; thus drastically increasing the computational cost. Stiffness is computationally expensive—adaptive time-stepping and approximate techniques can significantly improve simulation efficiency for heterogeneous biochemical systems.

The tau-leaping methods are the most accurate of the approximation techniques for both the CME and the RDME. Adaptivity plays a critical role for approximating the solution of the CME and RDME. Using the leaping strategies, adaptivity attempts to improve simulation time by picking the largest value of τ that is compatible with the *leap condition*[3]. Gillespie [23] proposed the first adaptive τ-selection procedure which imposed a uniform bound in the leap condition and only considers the mean of the propensity functions. Petzold and Gillespie [26] proposed an improvement to Gillespie's adaptive τ-selection procedure by considering bounds on both the mean and the variance of the propensity functions. However, these methods fail to account for the potential order of magnitude differences in the propensity functions. Cao et. al. [9] provided the next incremental improvement to Petzold and Gillespie's τ-selection method by bounding the

[3] The condition stating that the propensity functions remain relatively constant on the time interval τ.

relative changes in the propensity functions, thus accounting for propensity functions of differing magnitudes. Cao et. al. [9] then proposed a second, and now more widely used, τ-selection scheme which bounds the relative changes in the molecular populations and considers the mean and variance of the molecular species. Nonetheless, Cao et. al.'s τ-selection procedure still has the potential to violate the leap condition in a single species, over one step.

As biochemical systems and models increase in size and complexity, it becomes computationally very expensive to complete biochemical simulation runs of even short time durations using exact methods [17]. Therefore, there exists a need for faster, but accurate approximation techniques. In this thesis, we are proposing a new adaptive τ-stepping strategy for the tau-leaping method for numerically simulating the Reaction-Diffusion Master Equation [42]. We base our variable time-stepping technique on an integral controller from control theory [27]. This method has been shown to be very efficient for deterministic models, and has also been successfully applied to adapt the step size in the numerical solution of Stochastic Differential Equations [6, 33]. Our variable time-stepping scheme aims to minimize the computational cost while guaranteeing the desired accuracy. For the models tested we observed a two orders of magnitude speed-up over exact methods.

To guarantee the solution accuracy, we employ *post-leap checking*. For each time-step, we verify that the numerical solution satisfies the leap condition with some pre-defined tolerance [23, 9], before attempting the next step. We may also reject occasionally, but store leap steps which fail the leap condition. To avoid biasing the solution statistics, we use the post-leaping checking technique described by Anderson [1] to condition the solution trajectory based on the rejected, but stored solution steps. This technique guarantees that the leap condition is always satisfied. As such, we see excellent agreement in the simulation results from our proposed numerical method when compared to the simulation results from an exact inhomogeneous stochastic simulation algorithm for the RDME, for models with molecular populations bounded away from zero. Our new and reliable adaptive tau-leaping strategy can be used as a standalone method to numerically solve the RDME. Our variable step size scheme can also be effectively integrated into hybrid methods which use tau-leaping schemes for numerically solving the RDME.

Finally, we provide an overview of this thesis. In Chapter 2, we introduce the mathematical models and simulation methods for biochemical systems. We first consider homogeneous biochemical systems and discuss the mathematical theory associated with Chemical Master Equation, the Stochastic Simulation Algorithm, and the Poisson tau-leaping algorithm. We then extend the mathematical theory to heterogeneous biochemical systems and describe the Reaction-Diffusion Master Equation, the Inhomogeneous Stochastic Simulation Algorithm, and the spatial Poisson tau-leaping algorithm. In Chapter 3, we present our proposed adaptive spatial Poisson tau-leaping method. In this chapter, we outline the mechanics of accuracy estimation, time adaptivity, and step rejection; the three key features used to maintain solution accuracy. Next, in Chapter 4, we test the numerical simulation algorithm on three biochemical models to demonstrate the efficiency and accuracy of the new method. Lastly, in Chapter 5 we discuss the implications of our work and the future directions of research.

Chapter 2

Background

Continuous deterministic models of biochemical systems have been successfully used to model cellular processes [52]; however when these molecular systems include species in low numbers or are prone to noise amplification, deterministic models fail to accurately reproduce the system's dynamics [53], because stochasticity may dominate the system's behaviour. The Chemical Master Equation, a homogeneous discrete, stochastic model can successfully reproduce these types of system dynamics. Additionally, certain critical cellular processes like the cell cycle [30] or intra-cellular signaling [3] depend on the heterogeneous layout of the eukaryotic cell. The Reaction-Diffusion Master Equation, a heterogeneous, discrete stochastic model, incorporates spatial heterogenity by dividing the system's volume into voxels. Each voxel is treated as a homogeneous system and the species can freely diffuse between the voxels.

2.1 Chemical Master Equation and methods

Consider a biochemical system of constant volume Ω at thermodynamic equilibrium with N chemical species $\{S_1, S_2, \ldots, S_N\}$ and M chemical reactions $\{R_1, R_2, \ldots, R_M\}$. We assume this system is *well-stirred*, i.e. the molecules of each chemical species are spread uniformly throughout the volume Ω. These assumptions allow us to define the *state vector*, $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_N(t)]^{\mathrm{T}}$, where each $X_i(t)$ is the number of molecules of species S_i at time t. The *state vector* tracks the evolution of the species in the biochemical system through time.

Each reaction R_j has an associated *stochastic rate constant*, k_j, *state-change vector*, $\boldsymbol{\nu}_j$, and *propensity function*, $a_j(\mathbf{x})$, where $\mathbf{X}(t) = \mathbf{x}$ is the current system state at time t. The *state-change vector* $\boldsymbol{\nu}_j = (\nu_{1j}, \ldots, \nu_{Nj})^{\mathrm{T}}$, where each ν_{ij} is the net change in species S_i caused by reaction R_j. If the system is in state $\mathbf{x}$ and one reaction R_j occurs, then the new system state is $\mathbf{x} + \boldsymbol{\nu}_j$. The set of state-change vectors $\{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots, \boldsymbol{\nu}_M\}$ can also be represented as a *stoichiometric matrix* $\nu = [\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots, \boldsymbol{\nu}_M]$.

The physical meaning of the *propensity function*, $a_j(\mathbf{x})$, is that $a_j(\mathbf{x})dt$ gives the *probability* that one reaction R_j occurs in the next infinitesimal time interval $[t, t + dt)$ given that the current system state is $\mathbf{X}(t) = \mathbf{x}$. Thus, $a_j(\mathbf{x})$ is the rate at which reaction R_j occurs in the system.

The form of the propensity function $a_j(\mathbf{x})$ follows *mass-action kinetics*, it is based on the order of the reaction R_j and the principles of the kinetic theory. Zeroth-order reactions have the form $\emptyset \overset{k_j}{\rightarrow} S_n$ with $a_j(\mathbf{x}) = k_j$. First-order reactions have the form $S_n \overset{k_j}{\rightarrow}$ "products" with $a_j(\mathbf{x}) = k_j x_n$. Second-order reactions have either the form $S_n + S_m \overset{k_j}{\rightarrow}$ "products" with $a_j(\mathbf{x}) = k_j x_n x_m$ or the form $2S_n \overset{k_j}{\rightarrow}$ "products" with $a_j(\mathbf{x}) = k_j \frac{x_n(x_n-1)}{2}$. Third-orders reactions have 4 possible forms—one example is $2S_n + S_m \overset{k_j}{\rightarrow}$ "products" with $a_j(\mathbf{x}) = k_j \frac{x_n(x_n-1)x_m}{2}$. For most biochemical system simulations, third- and higher-order reactions are typically modeled as a series of first- and second-order reactions.

2.1.1 Derivation of the Chemical Master Equation

The well-stirred biochemical system described previously may be modeled by a finite state, continuous-time Markov process. A *finite state, continuous-time Markov process* [44] is a stochastic process with the property that the future states are independent of the past states given the present state. This can be represented formally as

$$P(X(t+dt) = x|\{X(t) = x(t)|t \in [0,t]\}) = P(X(t+dt) = x|X(t) = x(t)), \quad \forall\, t \in [0,\infty),\ x \in S$$

where S is the state space of the stochastic process X. Below, we shall derive a set of differential equations whose solution gives the full transitional probability kernel for the biochemical system's dynamics. This set of differential equations is known as the Chemical Master Equation (CME).

For any pair of states i and j, the *transient probability*, $p_{ij}(t) = P\{X(t+s) = j|X(s) = i\}$, denotes that the process is currently in state i and will be in state j at a time t later. Let v_i be the rate at which the process makes a transition while in state i. Let p_{ij} be the probability that this transition brings the process to state j. Then, the *instantaneous transition rate*, $q_{ij} = v_i p_{ij}$, is the rate at which the process when in state i moves to state j. Based on the definition of q_{ij}, we can derive the following relationships,

$$v_i = \sum_j v_i p_{ij} = \sum_j q_{ij} \tag{2.1}$$

and

$$p_{ij} = \frac{q_{ij}}{v_i} = \frac{q_{ij}}{\sum_j q_{ij}}. \tag{2.2}$$

Therefore, once the instantaneous transition rates are specified, the parameters of the continuous-time Markov process can be found using equations (2.1) and (2.2).

Now, we will prove a series of lemmas prior to introducing the *Kolmogorov's forward equations* which are the basis for the CME. These results are attributed to Ross [44].

Lemma 2.1.1. The following property applies

$$\lim_{h \to 0} \frac{1 - p_{ii}(h)}{h} = v_i. \tag{2.3}$$

Proof. Since we assume the amount of time until the next transition is exponentially distributed, we know the probability of 1 event occurring in time h is $v_i h + o(h)$ and the probability of 2 or more events occurring in time h is $o(h)$. We note that $p_{ii}(h)$ is the probability that the process starts in state i at time zero and remains in state i at time h. Thus, $1 - p_{ii}(h)$, the probability that the process is in state i at time 0 and will not be in state i at time h is

$$1 - p_{ii}(h) = v_i h + o(h).$$

Dividing through by h, then taking the limit of both sides yields

$$\lim_{h \to 0} \frac{1 - p_{ii}(h)}{h} = v_i.$$

□

Lemma 2.1.2. The following holds

$$\lim_{h \to 0} \frac{p_{ij}(h)}{h} = q_{ij} \qquad \text{when } i \neq j. \tag{2.4}$$

Proof. We note that $p_{ij}(h)$, the probability that the process goes from state i to state j in time h, equals the probability that a transition occurs multiplied by the probability that the transition brings the process into state j. Thus,

$$p_{ij}(h) = v_i h p_{ij} + o(h),$$
$$p_{ij}(h) = h q_{ij} + o(h).$$

Dividing through by h, then taking the limit of both sides yields

$$\lim_{h \to 0} \frac{p_{ij}(h)}{h} = q_{ij}.$$

□

Lemma 2.1.3. *Chapman-Kolmogorov equations*
For all $s \geq 0$, $t \geq 0$, the equation below is valid

$$p_{ij}(t+s) = \sum_{k=0}^{\infty} p_{ik}(t) p_{kj}(s). \tag{2.5}$$

Proof. For the process to move from state i to state j in time $t + s$, the process must be somewhere at time

t, therefore

$$\begin{aligned} p_{ij}(t+s) &= P\{X(t+s)=j|X(0)=i\} \\ &= \sum_{k=0}^{\infty} P\{X(t+s)=j \cap X(t)=k|X(0)=i\} \\ &= \sum_{k=0}^{\infty} P\{X(t+s)=j|X(t)=k \cap X(0)=i\}P\{X(t)=k|X(0)=i\} \\ &= \sum_{k=0}^{\infty} P\{X(t+s)=j|X(t)=k\}P\{X(t)=k|X(0)=i\} \\ &= \sum_{k=0}^{\infty} p_{jk}(s)p_{ki}(t). \end{aligned}$$

□

Theorem 2.1.4. *Kolmogorov's forward equations*
For all states i, j, and times $t \geq 0$,

$$p'_{ij}(t) = \sum_{k\neq j} q_{kj}p_{ik}(t) - v_j p_{ij}(t). \tag{2.6}$$

Proof. Starting from equation (2.5), we have

$$\begin{aligned} p_{ij}(t+h) - p_{ij}(t) &= \sum_{k=0}^{\infty} p_{ik}(t)p_{kj}(h) - p_{ij}(t) \\ &= \sum_{k\neq j} p_{ik}(t)p_{kj}(h) + p_{ij}(t)p_{jj}(h) - p_{ij}(t) \\ &= \sum_{k\neq j} p_{ik}(t)p_{kj}(h) - p_{ij}(t)(1-p_{jj}(h)). \end{aligned}$$

Dividing through by h, then taking the limit of both sides yields

$$\lim_{h\to 0} \frac{p_{ij}(t+h)-p_{ij}(t)}{h} = \lim_{h\to 0}\left[\sum_{k\neq j} p_{ik}(t)\frac{p_{kj}(h)}{h} - p_{ij}(t)\left[\frac{1-p_{jj}(h)}{h}\right]\right].$$

Since the process has a finite number of system states, we can exchange the limit and the summation, then apply Lemmas 2.1.1 and 2.1.2 to arrive at

$$p'_{ij}(t) = \sum_{k\neq j} q_{kj}p_{ik}(t) - v_j p_{ij}(t).$$

□

To derive the Chemical Master Equation (CME), we begin with Kolmogorov's forward equations (2.6) and consider a biochemical system that goes from state $i = \mathbf{x}_0$ at t_0 to state $j = \mathbf{x}$ at t. Thus,

$$p'_{\mathbf{x}_0\,\mathbf{x}}(t) = \sum_{k \neq \mathbf{x}} q_{k\,\mathbf{x}} p_{\mathbf{x}_0\,k}(t) - v_{\mathbf{x}} p_{\mathbf{x}_0\,\mathbf{x}}(t).$$

Given that the system is in state $\mathbf{x}$, $k = \mathbf{x} - \boldsymbol{\nu}_j$ for $j = 1, \ldots, M$ are the only possible transitions that bring the system to this particular state. It follows that the instantaneous transition rate $q_{k\,\mathbf{x}}$ becomes $q_{(\mathbf{x}-\boldsymbol{\nu}_j)\,\mathbf{x}}$. Then, we can conclude based on the definition of the propensity function $a_j(\mathbf{x})$ that $q_{(\mathbf{x}-\boldsymbol{\nu}_j)\,\mathbf{x}} = a_j(\mathbf{x} - \boldsymbol{\nu}_j)$.

Furthermore, given that the system is in state $\mathbf{x}$, $\mathbf{x} + \boldsymbol{\nu}_j$ for $j = 1, \ldots, M$ are the only possible transitions that bring the system from this particular state. We can conclude based on equation (2.1) and the definition of $a_j(\mathbf{x})$, that $v_{\mathbf{x}} = \sum_{j=1}^{M} q_{\mathbf{x}\,(\mathbf{x}+\boldsymbol{\nu}_j)} = \sum_{j=1}^{M} a_j(\mathbf{x})$.

Finally, letting $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ be the probability that $\mathbf{X}(t)$ is equal to $\mathbf{x}$, given that $\mathbf{X}(t_0) = \mathbf{x}_0$, we have $P(\mathbf{x}, t|\mathbf{x}_0, t_0) = p_{\mathbf{x}_0\,\mathbf{x}}(t)$. Therefore, by combining these results, we arrive at the following equation,

$$\begin{aligned} \frac{\partial P(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\partial t} &= \mathcal{M} P(\mathbf{x}, t|\mathbf{x}_0, t_0) \\ &= \sum_{j=1}^{M} \left[a_j(\mathbf{x} - \boldsymbol{\nu}_j) P(\mathbf{x} - \boldsymbol{\nu}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x}) P(\mathbf{x}, t|\mathbf{x}_0, t_0) \right], \end{aligned} \tag{2.7}$$

which represents the Chemical Master Equation [41, 22]. It governs the dynamics of well-stirred biochemical kinetics. Model (2.7) is a system of linear ordinary differential equations (ODEs) with one equation for each possible system state. This model is *homogeneous*, discrete and stochastic and is intractable for most biochemical systems.

2.1.2 Relationship between the stochastic and deterministic biochemical kinetic models

The classical deterministic biochemical kinetic model of the reaction-rate equations (RRE) may be derived under certain assumptions from the stochastic biochemical kinetic model (see: Wilkinson [55] for more details). We start by deriving a set of differential equations by considering the expected value of

$$\begin{aligned} \frac{\partial}{\partial t} E(\mathbf{X}(t)) &= \frac{\partial}{\partial t} \sum_{\mathbf{x} \in \mathcal{S}} \mathbf{x} P(\mathbf{x}, t|\mathbf{x}_0, t_0), \\ &= \sum_{\mathbf{x} \in \mathcal{S}} \mathbf{x} \frac{\partial}{\partial t} P(\mathbf{x}, t|\mathbf{x}_0, t_0), \end{aligned}$$

where $\mathcal{S}$ is the state space of the process. Applying equation (2.7) yields

$$\begin{aligned}\frac{\partial}{\partial t}E(\mathbf{X}(t)) &= \sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x}\frac{\partial}{\partial t}P(\mathbf{x},t|\mathbf{x}_0,t_0) \\ &= \sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x}\left[\sum_{j=1}^{M}\left[a_j(\mathbf{x}-\boldsymbol{\nu}_j)P(\mathbf{x}-\boldsymbol{\nu}_j,t|\mathbf{x}_0,t_0) - a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right]\right] \\ &= \sum_{j=1}^{M}\left[\sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x}a_j(\mathbf{x}-\boldsymbol{\nu}_j)P(\mathbf{x}-\boldsymbol{\nu}_j,t|\mathbf{x}_0,t_0) - \sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x}a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right].\end{aligned}$$

In the previous equation, we are summing over every possible system state $\mathbf{x}$. Consequently, we can exchange $\mathbf{x}+\boldsymbol{\nu}_j$ for $\mathbf{x}$ because $\mathbf{x}+\boldsymbol{\nu}_j \in \mathcal{S}$. Thus

$$\frac{\partial}{\partial t}E(\mathbf{X}(t)) = \sum_{j=1}^{M}\left[\sum_{\mathbf{x}\in\mathcal{S}}(\mathbf{x}+\boldsymbol{\nu}_j)a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0) - \sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x}a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right].$$

From the definition of the expected value we can rewrite the equation as

$$\frac{\partial}{\partial t}E(\mathbf{X}(t)) = \sum_{j=1}^{M}\left[E((\mathbf{X}(t)+\boldsymbol{\nu}_j)a_j(\mathbf{X}(t))) - E(\mathbf{X}(t)a_j(\mathbf{X}(t)))\right].$$

Then, applying linearity of expectation yields

$$\begin{aligned}\frac{\partial}{\partial t}E(\mathbf{X}(t)) &= \sum_{j=1}^{M} E(\boldsymbol{\nu}_j a_j(\mathbf{X}(t))) \\ &= \sum_{j=1}^{M} \boldsymbol{\nu}_j E(a_j(\mathbf{X}(t))).\end{aligned}$$

Now, let $E(\mathbf{X}(t)) = \mathbf{y}(t)$ be the expected value of the system state at time t. Then,

$$\frac{d}{dt}\mathbf{y}(t) = \sum_{j=1}^{M} \boldsymbol{\nu}_j E(a_j(\mathbf{X}(t))).$$

If

$$E(a_j(\mathbf{X}(t))) = a_j E(\mathbf{X}(t)), \tag{2.8}$$

then

$$\frac{d}{dt}\mathbf{y}(t) = \sum_{j=1}^{M} \boldsymbol{\nu}_j E(a_j(\mathbf{X}(t))) = \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(E(\mathbf{X}(t))) = \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{y}(t))$$

and the evolution of the expected value of the system state, which was modeled by the stochastic biochemical kinetic model satisfies the deterministic continuous model of the RRE.

Equation (2.8) is valid for the following forms of the propensity function $a_j(\mathbf{X}(t))$:

- $a_j(\mathbf{X}(t)) = k_j$, since

$$E(a_j(\mathbf{X}(t))) = E(k_j) = k_j = a_j(\mathbf{X}(t)),$$

- $a_j(\mathbf{X}(t)) = k_j x_i$, as

$$E(a_j(\mathbf{X}(t))) = E(k_j x_i) = k_j E(x_i) = a_j(E(\mathbf{X}(t))).$$

Therefore, if the biochemical system is only comprised of zero- or first-order reactions, the RRE may be derived by taking the expected value of the stochastic model of the CME.

Equation (2.8) is not necessarily true for the propensity function $a_j(\mathbf{X}(t)) = k_j x_i x_l$, because

$$E(a_j(\mathbf{X}(t))) = E(k_j x_i x_l) = k_j E(x_i x_l) \neq k_j E(x_i) E(x_l),$$

as the populations of species S_i and S_l are not independent in general.

In the thermodynamic limit, defined by Gillespie [24] as "the limit in which the species populations X_i and the system volume Ω all approach infinity, but in such a way that the species concentrations X_i/Ω stay constant", the stochastic effects become negligible compared to the deterministic term. Thus, in biochemical stochastic models where the number of molecules of the species is generally greater than 1000, we are often able to reduce the stochastic model to that of the deterministic reaction-rate equations.

2.1.3 Derivation of the Stochastic Simulation Algorithm

Since the CME (2.7) is intractable for most biochemical systems, numerical simulation algorithms have been developed to compute its solution. One commonly used simulation approach is the Stochastic Simulation Algorithm (SSA) or Gillespie's algorithm [20, 21]. This exact algorithm simulates each reaction in the system and the time when it occurs. Each simulation generates a possible realization of the stochastic process $\mathbf{X}$, not the entire probability distribution.

Assuming that the system is in state $\mathbf{X}(t) = \mathbf{x}$ at time t, we are interested in the probability that (a) the next reaction will take place in the infinitesimal time interval $[t+\tau, t+\tau+d\tau)$ and (b) will move the system to state $\mathbf{x}+\boldsymbol{\nu}_j$ because the jth reaction fired. Let $p(\mathbf{x}+\boldsymbol{\nu}_j, \tau, |\mathbf{x}, t)dt$ denote this probability. It follows that

$$\begin{aligned} p(\mathbf{x}+\boldsymbol{\nu}_j, \tau, |\mathbf{x}, t)dt = {} & P(\text{Next reaction in } [t+\tau, t+\tau+d\tau)|\mathbf{x}, t) \\ & \times P(\mathbf{x}+\boldsymbol{\nu}_j|\text{Next reaction in } [t+\tau, t+\tau+d\tau), \mathbf{x}, t). \end{aligned} \tag{2.9}$$

The following propositions, according to Wilkinson [55], will be used to justify our derivation of the SSA.

Proposition 2.1.5. If $X_i \sim Exp(\lambda_i)$, $i = 1, 2, \ldots, n$, are independent random variables, then

$$X_0 \equiv \min_i \{X_i\} \sim Exp(\lambda_0), \text{ where } \lambda_0 = \sum_{i=1}^{n} \lambda_i.$$

Proposition 2.1.6. If $X_i \sim Exp(\lambda_i)$, $i = 1, 2, \ldots, n$, are independent random variables, let j be the index of the smallest in value of the X_i. Then j is a discrete random variable with probability mass function (PMF)

$$\pi_i = \frac{\lambda_i}{\lambda_0}, \quad i = 1, 2, \ldots, n, \text{ where } \lambda_0 = \sum_{i=1}^{n} \lambda_i.$$

First, consider $P(\text{Next reaction in } [t+\tau, t+\tau+d\tau)|\mathbf{x}, t)$. For $j = 1, 2, \ldots, M$, the time until reaction R_j occurs, assuming no other reactions have occurred in the interim, is an $Exp(a_j(\mathbf{x}))$ random variable. Therefore, we can apply the result from Proposition 2.1.5 to find the minimum time to the next reaction, τ, as $Exp(a_0(\mathbf{x}))$ where $a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x})$. Since an exponential random variable has the form $Exp(\lambda) = \lambda e^{-\lambda x}$, the probability that the next reaction will take place in $[t+\tau, t+\tau+d\tau)$[1] is

$$P(\text{Next reaction in } [t+\tau, t+\tau+d\tau)|\mathbf{x}, t) = a_0(\mathbf{x}) e^{-a_0(\mathbf{x})\tau} d\tau. \tag{2.10}$$

Next, consider $P(\mathbf{x} + \boldsymbol{\nu}_j|\text{Next reaction in } [t+\tau, t+\tau+d\tau), \mathbf{x}, t)$. This can be rewritten as

$$P\left(\mathbf{X}(t+\tau+d\tau) = \mathbf{x} + \boldsymbol{\nu}_j | [\mathbf{X}(t+\tau) = \mathbf{x}] \cap [\mathbf{X}(t+\tau+d\tau) \neq \mathbf{x}]\right)$$
$$= \frac{P\left(\mathbf{X}(t+\tau+d\tau) = \mathbf{x} + \boldsymbol{\nu}_j | \mathbf{X}(t+\tau) = \mathbf{x}\right)}{P\left(\mathbf{X}(t+\tau+d\tau) \neq \mathbf{x} | \mathbf{X}(t+\tau) = \mathbf{x}\right)} = \frac{a_j(\mathbf{x}) d\tau}{\sum_{j=1}^{M} a_j(\mathbf{x}) d\tau}.$$

Thus,

$$P(\mathbf{x} + \boldsymbol{\nu}_j|\text{Next reaction in } [t+\tau, t+\tau+d\tau), \mathbf{x}, t) = \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})}. \tag{2.11}$$

From Proposition 2.1.6, we recognize equation (2.11) as the probability mass function for the specific reaction R_j that has occurred during the time interval $[t+\tau, t+\tau+d\tau)$.

Substituting equation (2.10) and equation (2.11) into equation (2.9) yields

$$p(\mathbf{x} + \boldsymbol{\nu}_j, \tau, |\mathbf{x}, t) = \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})} a_0(\mathbf{x}) e^{-a_0(\mathbf{x})\tau}. \tag{2.12}$$

The function $p(\mathbf{x} + \boldsymbol{\nu}_j, \tau, |\mathbf{x}, t)$ is the joint density function for two independent random variables: the **next reaction index**, j, and the **time to next reaction**, τ. Equation (2.12) provides the mathematical

[1] $P(\tau < T < \tau + d\tau) = \int_{\tau}^{\tau+d\tau} f_T(y)dy \simeq f_T(\tau)d\tau$ for small $d\tau$.

basis for the Stochastic Simulation Algorithm (SSA) [20]. Because the **next reaction index**, j, and the **time to next reaction**, τ, are independent random variables, these two values can be simulated separately.

Stochastic Simulation Algorithm

1. Initialize the time $t = t_0$ and system state $\mathbf{X}(t_0) = \mathbf{x}_0$.
2. Generate two independent uniform $U(0,1)$ random numbers ξ_1 and ξ_2.
3. Evaluate $a_i(\mathbf{X}(t))$ for $i = 1, \dots, M$ and $a_0 = \sum_{i=1}^{M} a_i(\mathbf{X}(t))$.
4. Calculate the time to the next reaction $\tau = \dfrac{1}{a_0} \ln\left(\dfrac{1}{\xi_1}\right)$.
5. Calculate the next reaction index, j, as the smallest integer satisfying $\sum_{i=1}^{j} a_i(\mathbf{X}(t)) > \xi_2 a_0$.
6. Set $\mathbf{X}(t+\tau) = \mathbf{X}(t) + \boldsymbol{\nu}_j$ and update t to $t+\tau$
7. Return to step 2.

The SSA provides an exact method for numerically solving the CME (2.7), but can be prohibitively slow in practice, as often some biochemical reactions occur on a very fast time-scale.

2.1.4 Derivation of the Poisson tau-leaping algorithm

Poisson tau-leaping is one approximate method for numerically solving the CME (2.7), that aims to reduce the computational cost of the stochastic simulation with an insufficient effect on the accuracy. Like the SSA, each simulation yields a single state vector, not the entire probability distribution.

The SSA assumes that the time-step τ is so small that only one reaction will occur during each time-step. If this requirement is relaxed, the exact state of the system at time $t+\tau$ is found by the following equation, proven by Kurtz [15]:

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}_j \left(\int_t^{t+\tau} a_j(\mathbf{X}(s)) ds \right), \tag{2.13}$$

where $\mathcal{P}_j(\cdot)$ is a Poisson random variable with parameter $\int_t^{t+\tau} a_j(\mathbf{X}(s))ds$. If we assume that τ is small enough such that all of the propensity functions $a_j(\mathbf{X}(t))$, for $j = 1, \dots, M$, remain relatively constant on the interval $t \le s \le t+\tau$, i.e. that relatively few chemical reactions fire, then $a_j(\mathbf{X}(s)) \simeq a_j(\mathbf{X}(t))$. Therefore,

$$\begin{aligned}\int_t^{t+\tau} a_j(\mathbf{X}(s))ds &\simeq \int_t^{t+\tau} a_j(\mathbf{X}(t))ds \\ &= a_j(\mathbf{X}(t)) \int_t^{t+\tau} ds \\ &= a_j(\mathbf{X}(t))\tau.\end{aligned}$$

Thus, we can approximate

$$\mathcal{P}_j\left(\int_t^{t+\tau} a_j(\mathbf{X}(s))ds\right) \simeq \mathcal{P}_j(a_j(\mathbf{X}(t))\tau).$$

By substituting this equation into equation (2.13), we arrive at

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}_j(a_j(\mathbf{X}(t))\tau), \tag{2.14}$$

known as the tau-leaping method [23]. Equation (2.14) provided the mathematical basis for the Poisson tau-leaping algorithm.

Tau-leaping Algorithm

1. Initialize the time $t = t_0$ and system state $\mathbf{X}(t_0) = \mathbf{x}_0$.
2. Evaluate $a_j(\mathbf{X}(t))$ for $j = 1 \ldots M$.
3. Generate M independent Poisson random numbers p_j, from $\mathcal{P}_j(a_j(\mathbf{X}(t))\tau)$.
4. Set $\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum\limits_{j=1}^{M} \boldsymbol{\nu}_j p_j$ and update t to $t+\tau$ using either a fixed or variable τ.
5. Return to step 2.

By allowing several chemical reactions to occur at each time-step, the Poisson tau-leaping algorithm can significantly reduce the time it takes to numerically solve the CME (2.7).

2.1.5 Method for calculating a variable tau

When implementing the Poisson tau-leaping strategy, we want to guarantee that τ is chosen small enough so that the *Leap Condition* is satisfied: all propensity functions $a_j(\mathbf{X}(t))$ remain relatively constant on the time interval $[t, t+\tau)$. The *Leap Condition* is the name generally given to the assumption that reduces equation (2.13) to equation (2.14). One commonly used approach for selecting τ employs a *pre-leap check*, meaning that for every leap step, we first apply a set of criteria designed to select the optimal τ, then generate the new system state based on that τ. The Poisson tau-leaping algorithm presented in the previous section uses this approach.

The first method in the literature designed for generating a τ that obeys the leap condition was proposed by Gillespie [23], which was then improved by Petzold [26]. For some small prescribed error tolerance $\epsilon > 0$, we want to find a τ prior to each leap step such that the following relationship is satisfied,

$$|a_j(\mathbf{X}(t+\tau)) - a_j(\mathbf{X}(t))| \leq \max\{\epsilon a_j(\mathbf{X}(t)), k_j\} \quad \text{for } j = 1, \ldots, M, \tag{2.15}$$

where k_j is the stochastic rate constant and is the smallest amount propensity function a_j can change. Petzold [26] outlines a τ selection procedure which guarantees that equation (2.15) is satisfied. This method is not always accurate; moreover, the intermediate calculations required prior to each leap tend to be very time consuming.

As such, Cao et. al. [9] developed a different τ selection procedure which bounds the relative changes in the molecular populations, $X_i(t)$, instead of the propensity functions, $a_j(\mathbf{X}(t))$. Thus, we now want to satisfy the following relationship

$$|X_i(t+\tau) - X_i(t)| \leq \max\left\{\epsilon_i X_i(t), 1\right\}, \tag{2.16}$$

where $i \in I_{rs}$ and I_{rs} denotes the set of indices of all reactant species. This condition requires that the relative change in X_i is bounded by ϵ_i, such that X_i will never change by an amount less than 1. Even when $X_i(t)$ contains a few molecules, the relationship $|X_i(t+\tau) - X_i(t)| \leq 1$ permits $X_i(t)$'s participation in one reaction.

The relationship between ϵ_i and the user prescribed error tolerance ϵ is

$$\epsilon_i = \frac{\epsilon}{g_i}, \tag{2.17}$$

where $g_i = g_i(X_i(t))$ is defined as follows. Let HOR(i) denote the *highest order of reaction* in which species S_i appears as a reactant.

(i) If HOR(i) $= 1$, take $g_i = 1$.

(ii) If HOR(i) $= 2$, take $g_i = 2$, except if any second-order reaction requires two S_i molecules instead take

$$g_i = \left(2 + \frac{1}{X_i(t) - 1}\right)$$

(iii) If HOR(i) $= 3$, take $g_i = 3$, except if any third-order reaction requires two S_i molecules instead take

$$g_i = \frac{3}{2}\left(2 + \frac{1}{X_i(t) - 1}\right),$$

except if any third-order reaction requires three S_i molecules instead take

$$g_i = \left(3 + \frac{1}{X_i(t) - 1} + \frac{2}{X_i(t) - 2}\right).$$

It is important to note that g_i will remain constant throughout the simulation, except where several molecules of species S_i participate in a given reaction.

The length of τ is given by [9]

$$\tau = \min_{i \in I_{rs}} \left\{ \frac{\max\{\epsilon X_i(t)/g_i, 1\}}{|\hat{\mu}_i(\mathbf{X}(t))|}, \frac{(\max\{\epsilon X_i(t)/g_i, 1\})^2}{(\hat{\sigma}_i(\mathbf{X}(t)))^2} \right\}, \tag{2.18}$$

where

$$\hat{\mu}_i(\mathbf{X}(t)) = \sum_{j=1}^{M} \nu_{ij} a_j(\mathbf{X}(t)),$$
$$\hat{\sigma}_i(\mathbf{X}(t))^2 = \sum_{j=1}^{M} (\nu_{ij})^2 a_j(\mathbf{X}(t)).$$

The computation of τ using equation (2.18) is relatively expensive. Furthermore, it does not guarantee that the leap condition is satisfied over the step τ in general.

2.2 Reaction-Diffusion Master Equation and methods

The Reaction-Diffusion Master Equation (RDME) [18] is a very popular mesoscopic model that extends the CME and accounts for the spatially inhomogeneous layout of the eukaryotic cell.

The biochemical system's volume is partitioned into K subvolumes or *voxels* $\{\Omega_1, \ldots, \Omega_K\}$ of edge length h, where each voxel Ω_k is treated as a well-stirred system. When extending the CME to the RDME, the following changes occur within the system. First, the M chemical reactions R_j are replaced by KM reactions, R_{jk}, where R_{jk} is reaction R_j in voxel Ω_k. The N-dimensional *state vector* $\mathbf{X}(t)$ becomes a $N \times K$-dimensional *state vector* $\mathbf{X}(t) = [\mathbf{X}_1(t), \ldots, \mathbf{X}_K(t)]$, where for each voxel Ω_k, $\mathbf{X}_k(t) = [X_{1k}(t), \ldots, X_{Nk}(t)]^{\mathrm{T}}$ and each $X_{ik}(t)$ is the number of molecules of species S_i in voxel Ω_k at time t. The propensity function $a_j(\mathbf{x})$ for reaction R_j now becomes $a_{jk}(\mathbf{x}_k)$ for reaction R_{jk}, where $\mathbf{X}_k(t) = \mathbf{x}_k$ is the current state of the system for voxel Ω_k at time t. The *state-change vector* $\boldsymbol{\nu}_j$ for reaction R_j becomes $\boldsymbol{\nu}_{jk}$ for reaction R_{jk}. Thus, if voxel Ω_k is in state $\mathbf{x}_k$ and one reaction R_{jk} occurs, the new system state for voxel Ω_k is $\mathbf{x}_k + \boldsymbol{\nu}_{jk}$.

For the RDME, chemical species S_i is allowed to diffuse from voxel Ω_k to voxel Ω_l so long as voxel Ω_k and voxel Ω_l are in contact (see Figure 2.1). This diffusion is modeled as a *uni-molecular diffusive transfer reaction*, R^{d}_{ikl}. If B is the total number of shared surfaces of adjacent voxels, then there will be $2BN$ possible diffusive transfer reactions in the system. Similar to the CME, each diffusion R^{d}_{ikl} has an associated *stochastic diffusion rate constant*, d_i, a diffusion *state-change vector*, $\boldsymbol{\nu}^{\mathrm{d}}_{ikl}$, and a diffusion *propensity function*, $a^{\mathrm{d}}_{ikl}(\mathbf{x})$. The diffusive transfer reaction has the form

$$X_{ik} \xrightarrow{d_i} X_{il},$$

where the rate constant $d_i = \frac{D_i}{h^2}$ with diffusion coefficient, D_i, and length of the voxel, h. The diffusion *state-change vector* $\boldsymbol{\nu}^{\mathrm{d}}_{ikl}$ decreases $X_{ik}(t)$ by 1 and increases $X_{il}(t)$ by 1. If the system is in state $\mathbf{x}$ and one diffusion R^{d}_{ikl} occurs, then the new system state is $\mathbf{x} + \boldsymbol{\nu}^{\mathrm{d}}_{ikl}$. As in the CME, the physical meaning of the diffusion *propensity function*, $a^{\mathrm{d}}_{ikl}(\mathbf{x})$, is that $a^{\mathrm{d}}_{ikl}(\mathbf{x})dt$ gives the *probability* that one S_i molecule diffuses from voxel Ω_k to voxel Ω_l in the next infinitesimal time interval $[t, t+dt)$ given that the current system state $\mathbf{X}(t) = \mathbf{x}$. The diffusion propensity function has the form $a^{\mathrm{d}}_{ikl}(\mathbf{x}) = d_i x_{ik}$, where the rate constant $d_i = \frac{D_i}{h^2}$ with diffusion coefficient, D_i, and length of the voxel, h.

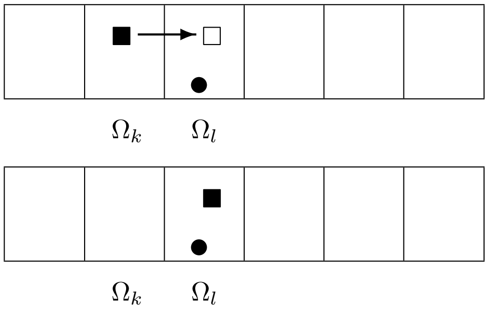


Figure 2.1: Example of diffusion in the Reaction-Diffusion Master Equation

2.2.1 Derivation of the Reaction-Diffusion Master Equation

Like the CME, the system described above may be modeled by a finite state, continuous-time Markov process. The system of differential equations whose solution gives the full transitional probability kernel for the biochemical system's dynamics with diffusion is known as the Reaction-Diffusion Master Equation [18].

For a biochemical system which only includes reactions, the master equation is

$$\begin{aligned}\frac{\partial P(\mathbf{x},t|\mathbf{x}_0,t_0)}{\partial t} &= \mathcal{M}P(\mathbf{x},t|\mathbf{x}_0,t_0)\\ &= \sum_{k=1}^{K}\sum_{j=1}^{M}\left[a_{jk}(\mathbf{x}_k-\boldsymbol{\nu}_{jk})P(\mathbf{x}_k-\boldsymbol{\nu}_{jk},t|\mathbf{x}_{k0},t_0)-a_{jk}(\mathbf{x}_k)P(\mathbf{x}_k,t|\mathbf{x}_{k0},t_0)\right].\end{aligned}$$

Note that for $K = 1$ this equation reduces to the CME in equation (2.7).

For a biochemical system which only includes diffusions, the master equation is

$$\begin{aligned}\frac{\partial P(\mathbf{x},t|\mathbf{x}_0,t_0)}{\partial t} &= \mathcal{D}P(\mathbf{x},t|\mathbf{x}_0,t_0)\\ &= \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{i=1}^{N}\left[a^{\mathrm{d}}_{ikl}(\mathbf{x}-\boldsymbol{\nu}^{\mathrm{d}}_{ikl})P(\mathbf{x}-\boldsymbol{\nu}^{\mathrm{d}}_{ikl},t|\mathbf{x}_0,t_0)-a^{\mathrm{d}}_{ikl}(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right].\end{aligned}$$

In the limit the diffusion of species S_i is governed by Einstein's diffusion equation,

$$\frac{\partial p(\mathbf{r},t)}{\partial t} = D_i \ \bigtriangledown^2_{\mathbf{r}}\, p(\mathbf{r},t),$$

where $\mathbf{r}$ is the position, p is the position probability density function of the S_i molecule and D_i is its diffusion coefficient [25].

Thus, for a biochemical system which includes both reactions and diffusions, this system can be considered a much larger dimensional Chemical Master Equation. The full RDME [18] is the sum of all of the reactions and diffusions over the set of voxels $\{\Omega_k\}$,

$$\begin{aligned}
\frac{\partial P(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\partial t} &= \\
&\sum_{k=1}^{K} \Bigg[\sum_{j=1}^{M} \left[a_{jk}(\mathbf{x}_k - \boldsymbol{\nu}_{jk}) P(\mathbf{x}_k - \boldsymbol{\nu}_{jk}, t|\mathbf{x}_{k0}, t_0) - a_{jk}(\mathbf{x}_k) P(\mathbf{x}_k, t|\mathbf{x}_{k0}, t_0)\right] \\
&\quad + \sum_{l=1}^{K} \sum_{i=1}^{N} \left[a^{\mathrm{d}}_{ikl}(\mathbf{x} - \boldsymbol{\nu}^{\mathrm{d}}_{ikl}) P(\mathbf{x} - \boldsymbol{\nu}^{\mathrm{d}}_{ikl}, t|\mathbf{x}_0, t_0) - a^{\mathrm{d}}_{ikl}(\mathbf{x}) P(\mathbf{x}, t|\mathbf{x}_0, t_0)\right] \Bigg] \\
&= \sum_{k=1}^{K} \sum_{j=1}^{M} \left[a_{jk}(\mathbf{x}_k - \boldsymbol{\nu}_{jk}) P(\mathbf{x}_k - \boldsymbol{\nu}_{jk}, t|\mathbf{x}_{k0}, t_0) - a_{jk}(\mathbf{x}_k) P(\mathbf{x}_k, t|\mathbf{x}_{k0}, t_0)\right] \\
&\quad + \sum_{k=1}^{K} \sum_{l=1}^{K} \sum_{i=1}^{N} \left[a^{\mathrm{d}}_{ikl}(\mathbf{x} - \boldsymbol{\nu}^{\mathrm{d}}_{ikl}) P(\mathbf{x} - \boldsymbol{\nu}^{\mathrm{d}}_{ikl}, t|\mathbf{x}_0, t_0) - a^{\mathrm{d}}_{ikl}(\mathbf{x}) P(\mathbf{x}, t|\mathbf{x}_0, t_0)\right] \\
&= \mathcal{M} P(\mathbf{x}, t|\mathbf{x}_0, t_0) + \mathcal{D} P(\mathbf{x}, t|\mathbf{x}_0, t_0).
\end{aligned} \tag{2.19}$$

Model (2.19) governs the dynamics of *heterogeneous* biochemical kinetics. As with the CME, the RDME model (2.19) is a system of linear ODEs with one equation for each possible system state. It is discrete and stochastic and is also intractable for most biochemical systems. Moreover, the RDME is much more challenging to simulate numerically than the CME of a biochemical system without diffusion.

2.2.2 Derivation of the Inhomogeneous Stochastic Simulation Algorithm

Analogous to the CME, numerical simulation algorithms have been developed to solve the RDME. One common exact algorithm used is an extension of the SSA, and it is called the Inhomogeneous Stochastic Simulation Algorithm (ISSA). The ISSA simulates every reaction, diffusion, and the time at which it occurs. As before, each simulation yields a single state vector, not the full probability distribution.

To implement the ISSA, we partition the system's volume Ω into K voxels using a uniform n-dimensional Cartesian mesh. For now, we assume that $n = 3$ when describing the partitioned system and associated ISSA, but n can easily be reduced to $n = 1$ or $n = 2$. To facilitate simulation, we will consider the voxel index k for Ω_k as a vector $\mathbf{k} \in \mathbf{K} = \{[k_1, k_2, k_3] \,|\, k_1 = 1, \dots K_x,\ k_2 = 1, \dots, K_y,\ k_3 = 1, \dots, K_z\}$, where k_1 corresponds to the x-axis, k_2 corresponds to the y-axis, and k_3 corresponds to the z-axis.

When modeling diffusion, we treat all diffusions as unimolecular chemical reactions and scale the diffusion constant D_i by $d_i = \frac{D_i}{h^2}$ where h is the length of the voxel. We allow the diffusive species S_i to flow freely throughout the volume Ω but treat the volume Ω's boundaries as reflective. By dividing the volume Ω into a uniform Cartesian mesh, we can easily determine all voxels $\Omega_{\mathbf{l}}$ that are in contact with voxel $\Omega_{\mathbf{k}}$. Let $\mathcal{E} = \{[1,0,0], [-1,0,0], [0,1,0], [0,-1,0], [0,0,1], [0,0,-1]\}$ be the set of possible diffusion directions. Then, any adjacent voxel $\Omega_{\mathbf{l}}$ must have $\mathbf{l} = \mathbf{k} + \varepsilon$, $\varepsilon \in \mathcal{E}$. By iterating over the set $\mathcal{E}$, we can account for all directions of diffusion from any voxel $\Omega_{\mathbf{k}}$. If voxel $\Omega_{\mathbf{k}}$ is located in the interior of the volume Ω, any species S_i in voxel $\Omega_{\mathbf{k}}$ is allowed to diffuse in all 6 directions. If voxel $\Omega_{\mathbf{k}}$ is located on the boundary of the volume Ω, any species S_i in voxel $\Omega_{\mathbf{k}}$ is only allowed to diffuse in the directions that keep it within the volume Ω.

From this set of assumptions, we will now redefine our diffusion propensity function as $a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{x})$, where e is the index that indicates the direction of diffusion from voxel $\Omega_{\mathbf{k}}$. The propensity function $a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{x}) = 0$, if species S_i does not diffuse or if e is a diffusion direction that takes species S_i outside the volume Ω.

In the ISSA, the **time to next reaction**, τ, is calculated similarly to the SSA, taking into account all the reaction and diffusive events. However, the **next reaction index**, j, in the SSA now becomes a **next event index** in the ISSA. This means that at every time-step in the ISSA, either a reaction has occurred with *next reaction index*, $j\mathbf{k}$, where j indicates the reaction R_j and $\mathbf{k}$ indicates the voxel $\Omega_{\mathbf{k}}$ or a diffusion has occurred with *next diffusion index*, $i\mathbf{k}e$, where i indicates the species S_i, $\mathbf{k}$ indicates the voxel $\Omega_{\mathbf{k}}$, and e indicates the direction of diffusion.

Inhomogeneous Stochastic Simulation Algorithm [49]

1. Initialize the time $t = t_0$ and system state $\mathbf{X}(t_0) = \mathbf{x}_0$.
2. Generate two independent uniform $U(0, 1)$ random numbers ξ_1 and ξ_2.
3. Evaluate $a_{j'\mathbf{k}'}(\mathbf{X}_{\mathbf{k}'}(t))$ for all $j' = 1, \ldots, M$ and $\mathbf{k}' \in \mathbf{K}$ and evaluate $a^{\mathrm{d}}_{i'\mathbf{k}'e'}(\mathbf{X}(t))$ for all $i' = 1, \ldots, N$, $\mathbf{k}' \in \mathbf{K}$, and $e' = 1, .., |\mathcal{E}|$.
4. Evaluate
$$a_0 = \sum_{j'=1}^{M} \left[\sum_{k_1'=1}^{K_x} \sum_{k_2'=1}^{K_y} \sum_{k_3'=1}^{K_z} a_{j'\,k_1'\,k_2'\,k_3'}(\mathbf{X}_{\mathbf{k}'}(t)) \right] + \sum_{e'=1}^{|\mathcal{E}|} \sum_{i'=1}^{N} \left[\sum_{k_1'=1}^{K_x} \sum_{k_2'=1}^{K_y} \sum_{k_3'=1}^{K_z} a^{\mathrm{d}}_{i'\,k_1'\,k_2'\,k_3'\,e'}(\mathbf{X}(t)) \right].$$
5. Calculate the time to the next reaction $\tau = \dfrac{1}{a_0} \ln\left(\dfrac{1}{\xi_1}\right)$.
6. Calculate next event index as either (a) or (b) where
   (a) the next reaction index, $j\mathbf{k}$, is the smallest quadruple (j, k_1, k_2, k_3) satisfying
$$\sum_{j'=1}^{j} \left[\sum_{k_1'=1}^{k_1} \sum_{k_2'=1}^{k_2} \sum_{k_3'=1}^{k_3} a_{j'\,k_1'\,k_2'\,k_3'}(\mathbf{X}_{\mathbf{k}'}(t)) \right] > \xi_2 a_0.$$
   (b) the next diffusion index, $i\mathbf{k}e$, is the smallest quintuple (i, k_1, k_2, k_3, e) satisfying
$$\sum_{j'=1}^{M} \left[\sum_{k_1'=1}^{K_x} \sum_{k_2'=1}^{K_y} \sum_{k_3'=1}^{K_z} a_{j'\,k_1'\,k_2'\,k_3'}(\mathbf{X}_{\mathbf{k}'}(t)) \right] + \sum_{e'=1}^{e} \sum_{i'=1}^{i} \left[\sum_{k_1'=1}^{k_1} \sum_{k_2'=1}^{k_2} \sum_{k_3'=1}^{k_3} a^{\mathrm{d}}_{i'\,k_1'\,k_2'\,k_3'\,e'}(\mathbf{X}(t)) \right] > \xi_2 a_0.$$
7. If a reaction occurred, set $\mathbf{X}_{\mathbf{k}}(t+\tau) = \mathbf{X}_{\mathbf{k}}(t) + \boldsymbol{\nu}_{j\mathbf{k}}$ or if a diffusion occurred, set $\mathbf{X}(t+\tau) = \mathbf{X}(t) + \boldsymbol{\nu}^{\mathrm{d}}_{i\mathbf{k}e}$ and update t to $t + \tau$.

8. Return to step 2.

Given that the number of possible reactions and diffusions in the ISSA is now $KM + 2BN$, where B is the total number of shared surfaces of adjacent voxels, compared to the M reactions in the SSA, we can see that solving the RDME using this algorithm is even more prohibitively slow than solving the CME with the SSA.

2.2.3 Derivation of the spatial Poisson tau-leaping algorithm

One algorithm used to improve the simulation time for the RDME is to extend the Poisson tau-leaping algorithm, used to numerically solve the CME, to the spatial Poisson tau-leaping algorithm for the RDME. Like the well-stirred case, the spatial Poisson tau-leaping method is an approximate algorithm which allows several chemical reactions and diffusions to occur during a given time-step τ. Again, assuming all of the reaction propensity functions $a_{jk}(\mathbf{x}_k(t))$ and diffusion propensity functions $a^{\mathrm{d}}_{ikl}(\mathbf{x}(t))$ remain relatively constant during the interval $t \leq s \leq t + \tau$, the number of reactions and diffusions that occur is well approximated by Poisson random variables with parameters $a_{jk}(\mathbf{x}_k(t))\tau$ and $a^{\mathrm{d}}_{ikl}(\mathbf{x}(t))\tau$, respectively.

Like for the ISSA, we partition the volume Ω into K voxels using a uniform n-dimensional mesh and follow the same indexing for voxel $\Omega_{\mathbf{k}}$. We treat all diffusions as a type of unimolecular chemical reaction with the same diffusion behaviour as outlined for the ISSA. We will retain the definition of the diffusion propensity function described in the ISSA as $a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{x})$, where e is the index indicating the direction of diffusion from voxel $\Omega_{\mathbf{k}}$. Thus, the number of molecules of species S_i that diffuse from voxel Ω_k in direction e is $\mathcal{P}_{i\mathbf{k}e}\left(a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{X}(t))\tau\right)$.

Spatial Poisson tau-leaping algorithm

1. Initialize the time $t = t_0$ and system state $\mathbf{X}(t_0) = \mathbf{x}_0$.

2. Evaluate $a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t))$ for all $j = 1, \ldots, M$ and $\mathbf{k} \in \mathbf{K}$ and evaluate $a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{X}(t))$ for all $i = 1, \ldots, N$, $\mathbf{k} \in \mathbf{K}$, and $e \in \mathcal{E}$.

3. Generate KM independent Poisson random numbers $p_{j\mathbf{k}}$, from $\mathcal{P}_{j\mathbf{k}}(a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t))\tau)$, and $K|\mathcal{E}|N$ independent Poisson random numbers $p_{i\mathbf{k}e}$, from $\mathcal{P}_{i\mathbf{k}e}\left(a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{X}(t))\tau\right)$.

4. Set

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{\mathbf{k}\in\mathbf{K}} \sum_{j=1}^{M} \boldsymbol{\nu}_{j\mathbf{k}} \omega_{\mathbf{k}} p_{j\mathbf{k}} + \sum_{\mathbf{k}\in\mathbf{K}} \sum_{e\in\mathcal{E}} \sum_{i=1}^{N} \boldsymbol{\nu}^{\mathrm{d}}_{i\mathbf{k}e} p_{i\mathbf{k}e}$$

   where $\omega_{\mathbf{k}}$ is a $1 \times K$ vector with 1 in the **k**th position and 0's elsewhere.

5. Update t to $t + \tau$ using either a fixed or variable τ.

6. Return to step 2.

Chapter 3

Adaptivity

Most biochemical systems include a combination of slow and fast reactions leading to multiple time-scales. These multiple time-scales in conjugation with diffusive systems with short voxel lengths lead to very small step sizes. Small step sizes then generate mathematically stiff problems and long simulation run times. Adaptive methods are known to drastically improve simulation efficiency. We are proposing a new adaptive tau-leaping method with post-leap checking to improve the simulation time of the Reaction-Diffusion Master Equation (see also Padgett & Ilie [42]).

3.1 Accuracy criteria

Our proposed adaptive tau-leaping algorithm aims to satisfy the accuracy criteria while improving the computational cost of the simulation. As we shall see below the accuracy criteria is based on guaranteeing that the leap condition is obeyed.

3.1.1 Theoretical foundation

We first present the justification for partitioning the species into two distinct sets, then introduce the formulae used to ensure our accuracy criteria is satisfied. The following sections consider the homogeneous system.

Partition the molecular populations $X_i(t)$

The Cao et. al. [9] leap condition (2.16) can be reposed as

$$\frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)} \leq \max\left\{\epsilon_i, \frac{1}{X_i(t)}\right\}, \tag{3.1}$$

for any $1 \leq i \leq N$ so long as $X_i(t) \neq 0$. The relationship between the tolerance ϵ and the small parameter ϵ_i is given in equation (2.17). Thus, there exists two potential bounds for the change in the molecular population X_i: the local relative change

$$\frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)} \leq \epsilon_i \tag{3.2}$$

and the local absolute change

$$|X_i(t_n + \tau_n) - X_i(t_n)| \leq 1. \tag{3.3}$$

When the population of $X_i(t)$ is sufficiently large, we find that equation (3.2) provides effective bounding for the local relative change of $X_i(t)$. We denote the set of species with large population numbers, X_L, where

$$X_L = \{X_i(t) \,|\, \epsilon_i X_i(t) > 1, i = 1, \dots, N\}.$$

However, if the population of $X_i(t)$ is too small, $\frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)} \nleq \epsilon_i$ unless the number of $X_i(t)$ does not change from time t to time $t + \tau$, effectively halting any reaction R_j with the net change of species S_i, $|\nu_{ij}| \geq 1$. We denote the set of species with small population numbers, X_S, where

$$X_S = \{X_i(t) \,|\, \epsilon_i X_i(t) \leq 1, i = 1, \dots, N\}.$$

The set X_S is bounded using equation (3.3) so $X_i(t)$ may still participate in a reaction during τ. The *critical threshold* where $X_i(t)$ switches between X_S and X_L is determined by considering

$$\frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)} = \epsilon_i.$$

If we assume the smallest possible net change of $X_i(t)$ on the interval $[t, t+\tau)$ is 1 and apply equation (2.17), we arrive at the formula for the *critical threshold*, c_i,

$$\begin{aligned} \frac{1}{X_i(t)} &= \frac{\epsilon}{g_i}, \\ c_i = X_i(t) &= \frac{g_i}{\epsilon}. \end{aligned} \tag{3.4}$$

Local change computations

The previous analysis demonstrates the need for two distinct local change calculations to guarantee our leap condition is satisfied. Once each $X_i(t)$ has been classified as having either a large or a small population, we can apply the following local change formulae.

Large population species, X_L For each $X_i(t) \in X_L$, e_i, the local relative change for $X_i(t)$, is calculated as

$$e_i = \frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)}. \tag{3.5}$$

We also define e_L, the global relative change for X_L, as

$$e_L = \max_{X_i(t) \in X_L} \{e_i\}.$$

Small population species, X_S For each $X_i(t) \in X_S$, E_i, the local absolute change for $X_i(t)$, is calculated as

$$E_i = |X_i(t+\tau) - X_i(t)|. \tag{3.6}$$

We also define E_S, the global absolute change for X_S, as

$$E_S = \max_{X_i(t) \in X_S} \{E_i\}.$$

Accuracy criteria and rejections

The Cao et. al. [9] method for τ-selection (also presented in Section 2.1.5) may still calculate τ so big that it violates the leap condition for one or more species. To overcome this limitation, Anderson [1] outlines the *post-leap checking* Poisson tau-leaping strategy for well-stirred biochemical systems which guarantees that the leap condition is satisfied for all $X_i(t)$. His method selects τ, generates a new potential system state, and then verifies that the potential system state satisfies the leap condition. If the leap condition is satisfied, the leap is accepted, otherwise the leap is rejected.

We consider the sets X_L and X_S separately when validating the leap condition.

Large population species For all $X_i(t) \in X_L$, we use equation (3.5) to calculate the local relative change. To satisfy the leap condition imposed on X_L, every local relative change e_i must satisfy

$$e_i = \frac{|X_i(t+\tau) - X_i(t)|}{X_i(t)} \leq \epsilon_i. \tag{3.7}$$

Small population species For all $X_i(t) \in X_S$, equation (3.6) is used to determine the local absolute change. To satisfy the leap condition imposed on X_S, every local absolute change E_i must obey

$$E_i = |X_i(t+\tau) - X_i(t)| \leq Z_i, \tag{3.8}$$

where $Z_i = \max_j \{|\nu_{ij}|\}$ is the consumption rate of species S_i. Typically, $Z_i = 1$, as only one molecule of species S_i is produced or consumed by a given reaction.

Should any of the elements in X_L fail the leap condition (3.7) or any of the elements in X_S fail the leap condition (3.8), we reject the entire leap step.

3.1.2 Application to heterogenous systems

When applying the previous analysis to our proposed spatial Poisson tau-leaping algorithm, we update the definitions and equations to reflect the discretization of the volume Ω into voxels $\Omega_{\mathbf{k}}$. These updated equations help maintain our solution accuracy and provide input to our stepping strategy.

For every time-step t_n, we divide the set $\{X_{i\mathbf{k}}\}$ into two sets X_L and X_S. The set of species with large population numbers X_L is now

$$X_L = \{X_{i\mathbf{k}}(t_n) | X_{i\mathbf{k}}(t_n) > c_i, i = 1, \ldots, N, \mathbf{k} \in \mathbf{K}\} \tag{3.9}$$

and the set of species with small population numbers X_S is now

$$X_S = \{X_{i\mathbf{k}}(t_n) | X_{i\mathbf{k}}(t_n) \leq c_i, i = 1, \ldots, N, \mathbf{k} \in \mathbf{K}\}, \tag{3.10}$$

where the critical threshold c_i comes from equation (3.4). Afterwards, we calculate the local changes and validate the leap condition as follows.

Large population species, X_L

To calculate the local relative change for all $X_{i\mathbf{k}}(t_n) \in X_L$, we update equation (3.5) to

$$e_{i\mathbf{k}} = \frac{|X_{i\mathbf{k}}(t_n + \tau_n) - X_{i\mathbf{k}}(t_n)|}{X_{i\mathbf{k}}(t_n)}. \tag{3.11}$$

We also update e_L, the global relative change for X_L, as

$$e_L = \max_{X_{i\mathbf{k}}(t_n) \in X_L} \{e_{i\mathbf{k}}\}. \tag{3.12}$$

Then, we use the local relative changes calculated with equation (3.11) to validate our updated leap condition. Every local relative change $e_{i\mathbf{k}}$ is required to satisfy

$$e_{i\mathbf{k}} = \frac{|X_{i\mathbf{k}}(t_n + \tau_n) - X_{i\mathbf{k}}(t_n)|}{X_{i\mathbf{k}}(t_n)} \leq \epsilon_i, \tag{3.13}$$

otherwise, we reject the entire leap step τ_n.

Small population species, X_s

To calculate the local absolute change for all $X_{i\mathbf{k}}(t_n) \in X_S$, we change equation (3.6) to

$$E_{i\mathbf{k}} = |X_{i\mathbf{k}}(t_n + \tau_n) - X_{i\mathbf{k}}(t_n)|. \tag{3.14}$$

and E_S, the global absolute change for X_S, becomes

$$E_S = \max_{X_{i\mathbf{k}}(t_n) \in X_S} \{E_{i\mathbf{k}}\}. \tag{3.15}$$

Afterwards, we use the local absolute changes calculated with equation (3.14) to validate our updated leap condition. Every local absolute change $E_{i\mathbf{k}}$ must satisfy

$$E_{i\mathbf{k}} = |X_{i\mathbf{k}}(t_n + \tau_n) - X_{i\mathbf{k}}(t_n)| \leq Z_i, \tag{3.16}$$

where $Z_i = \max_j \{|\nu_{ijk}|\}$ is the consumption rate of species S_i. We reject the entire leap step τ_n, if equation (3.16) fails for any $X_{i\mathbf{k}}(t_n) \in X_S$.

3.2 Step rejection

Step rejection in the *post-leap checking* tau-leaping scheme guarantees that the leap condition is always satisfied; however, naïve step rejection may skew the statistics of the numerical solution [1]. To overcome this issue, we employ the post-leap check technique proposed by Anderson [1] to reduce bias in the sample trajectories. Anderson's tau-leap procedure for well-stirred biochemical systems stores every rejected step. Our new adaptive tau-leaping strategy relaxes this storage requirement, and we see excellent accuracy in our numerical results. We first outline the theoretical framework and then describe in mathematical detail the methods used to generate a sample trajectory for the heterogeneous biochemical system.

3.2.1 Theoretical foundation

Unlike *pre-leap check* tau-leaping strategies, the methods used to generate the sample trajectory for our proposed adaptive tau-leaping scheme vary according to the satisfaction of the leap condition and any *restrictions* placed on the system. We use a *Poisson bridge* to handle these *restrictions*, that is the rejected and stored leap steps.

Poisson bridge

In [1], Anderson proposed the following theorem. We use its conclusions to generate reaction and diffusion events in the biochemical system after a step was rejected.

Theorem 3.2.1. Let $Y(t)$ be a Poisson process with intensity λ, and let $0 \leq s < u < t$. Then, conditioned on $Y(s)$ and $Y(t)$, $Y(u) - Y(s)$ has a binomial $(Y(t) - Y(s), r)$ distribution where $r = \dfrac{(u-s)}{(t-s)}$.

Proof. Without loss of generality, we suppose that $s = 0$ and $Y(0) = 0$. Let $Y(t) = N$ and $0 < u < t$. Then

$$\begin{aligned} P(Y(u) = j | Y(t) = N) & \\ &= \frac{P(Y(u) = j \cap Y(t) = N)}{P(Y(t) = N)} \\ &= \frac{P(Y(t) - Y(u) = N - j)P(Y(u) = j)}{P(Y(t) = N)} \\ &= \frac{(\lambda(t-u))^{N-j} e^{-\lambda(t-u)}}{(N-j)!} \frac{(\lambda u)^j e^{-\lambda u}}{j!} \frac{N!}{(\lambda t)^N e^{-\lambda t}} \\ &= \binom{N}{j} \left(1 - \frac{u}{t}\right)^{N-j} \left(\frac{u}{t}\right)^j . \end{aligned}$$

□

For the simulated biochemical system, the total number of firings of any reaction or diffusion event in the time interval $[t_0, t)$ is modeled as a Poisson process. We can denote this total number of firings as ρ_t. Assume the simulation is at time s and we know the total number of firings ρ_s in the time interval $[t_0, s)$. Also, assume that for some future time t, where $s < t$, we know the number of firings ρ_t (i.e. the leap step was rejected, but stored). To leap to time u where $s < u < t$, we generate the number of events (reaction or diffusion) that occurs in the time interval $[s, u)$ as a binomial $B\left(\rho_t - \rho_s, \frac{(u-s)}{(t-s)}\right)$ random variable. Thus, by conditioning on the rejected future trajectory, Theorem 3.2.1 provides the mechanism to guarantee that the correct solution path is followed.

Trajectory generation

Now, we present an overview of the process used to generate the sample trajectory in our adaptive tau-leaping algorithm. We have omitted the details of propensity function calculations, the next potential system state $\mathbf{X}^*(\cdot)$ calculation, the time-step τ adaptation, the leap step validation, and the restriction storage as the details for each of these topics are presented in other sections of the thesis. Finally, recall that the biochemical system under consideration is effectively modeled by a Markov process. As long we know the current state of the process, the future states do not depend on the past ones. This fact is used to justify why not all rejected leap steps are saved.

Suppose that the simulation is in state $\mathbf{X}(t)$ at time t with a pre-determined time-step of length τ, and that no restrictions have been placed on the sample trajectory. Also, suppose that we know the total number of firings of all reaction and diffusion events by time t. We attempt a leap using Poisson random variables to generate the number of reaction and diffusion events that occur in the system in the time interval $[t, t+\tau)$. After calculating the new system state, $\mathbf{X}^*(t+\tau)$, we validate the leap condition. If the leap condition is satisfied, we accept the leap, set $\mathbf{X}(t+\tau) = \mathbf{X}^*(t+\tau)$, and update the time t to $t = t + \tau$. Afterwards, we attempt another leap.

If the leap condition is not satisfied, we reject the leap. Failure of the leap condition indicates that the pre-determined time-step τ was too large. Thus, a new τ_1 is selected, such that $\tau_1 < \tau$. A restriction is

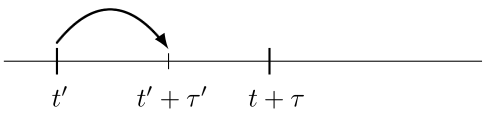


Figure 3.1: Leap before the restriction in place at $t + \tau$.

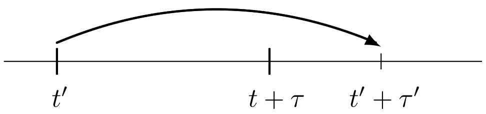


Figure 3.2: Leap past the restriction in place at $t + \tau$.

created on the sample trajectory at time $t + \tau$; and we store the total number of firings for all reaction and diffusion events that have occurred by time $t + \tau$. Next, we attempt another leap using the time-step τ_1. Since a restriction exists, we use binomial random variables, conditioned on the stored numbers for the reaction and diffusion events at time t and at time $t + \tau$, to generated the number of firings of reaction and diffusion events during the time interval $[t, t + \tau_1)$. We calculate the next potential system state $\mathbf{X}^*(t + \tau_1)$, before validating the leap condition. If the leap condition fails again, we select a new $\tau_2 < \tau_1 < \tau$, but do not store the reaction and diffusion events that have taken place by time $t + \tau_1$. This simplification is justified by the Markov property of the biochemical system. We leap using binomial random variables with τ_2 and the stored events at time t and at time $t + \tau$. We may repeat the process of selecting $\tau_i < \tau_{i-1} < \ldots < \tau$ and generating new leaps using binomial random variables until we find a potential system state $\mathbf{X}^*(t + \tau_i)$ that satisfies leap condition.

Once the leap condition has been satisfied and a restriction exists at time $t+\tau$, we must compare the final time of the current leap step, $t' + \tau'$ to time of the restriction to determine the appropriate leap generation method. If the final time of the current leap step falls before the restriction, i.e. $t'+\tau' < t+\tau$ (see Figure 3.1), we use binomial random variables as described in the previous paragraph to generate the leap for the time interval $[t', t'+\tau')$. When $t+\tau < t'+\tau'$, we have leapt past the restriction (see Figure 3.2). We have already established the number of reaction and diffusion events that occurred in the time interval $[t', t + \tau)$, so we attempt a leap for the remaining time interval $[t + \tau, t' + \tau')$ using Poisson random variables. We add the number of events that happened in $[t', t + \tau)$ and $[t + \tau, t' + \tau')$ and use the total in the calculation of the next potential system state $\mathbf{X}^*(t' + \tau')$. Afterwards, we verify the leap condition. For $t + \tau < t' + \tau'$, the outcome of the leap check governs the restriction(s) on the sample trajectory. If the leap condition is met, we can eliminate the restriction in place at time $t + \tau$. If the leap condition was not met, we create another restriction on the sample trajectory at the future time $t' + \tau'$.

Our proposed adaptive spatial tau-leaping strategy combines the mathematics presented in the next section with conceptual framework described above. Using this scheme, we see very good agreement of the results of the approximate algorithm when compared to the exact algorithm.

3.2.2 Application to heterogeneous systems

Our adaptive tau-leaping algorithm requires the following information: a series of times, t, and the total number of reactions $R_{j\mathbf{k}}$ and diffusions $R^d_{i\mathbf{k}e}$ that have occurred by each time t in the series. The *stored times vector*, $\mathbf{T}$, contains one or more times t sorted in ascending order. For each $t \in \mathbf{T}$, we have fixed the total number of firings for all reactions and diffusions in the time interval $[0, t)$ in the sample trajectory.

For a given time t, $\rho^t_{j\mathbf{k}}$ is the total number of firings of reaction $R_{j\mathbf{k}}$ and $\rho^t_{i\mathbf{k}e}$ is the total number of firings of diffusion $R^d_{i\mathbf{k}e}$ that have occurred in the time interval $[0, t)$. We define $\mathcal{R}_t$ as the collection of reactions $\{\rho^t_{j\mathbf{k}}\}^{j=1...M}_{k\in\mathbf{K}}$ and diffusions $\{\rho^t_{i\mathbf{k}e}\}^{i=1...N}_{k\in\mathbf{K},e\in\mathcal{E}}$ that occurred during the time interval $[0, t)$. Then, for all $t \in \mathbf{T}$, the *stored reaction-diffusion matrix*, $\mathcal{R} = \{\mathcal{R}_t\}$ tracks the total number of all reaction and diffusion events that occurred by those specific times in the sample trajectory.

The size of $\mathcal{R}$ depends on the partitioning of the volume Ω. For a 1-d mesh, $\mathcal{R}$ has size $K_x|\mathbf{T}|(M + i|\mathcal{E}|)$. For a 2-d mesh, $\mathcal{R}$ has size $K_xK_y|\mathbf{T}|(M + i|\mathcal{E}|)$. For a 3-d mesh, $\mathcal{R}$ has size $K_xK_yK_z|\mathbf{T}|(M + i|\mathcal{E}|)$.

Generation of the next potential system state

For a system in state $\mathbf{X}(t_n)$ at time t_n, the creation of the next potential system state $\mathbf{X}^*(t_n + \tau_n)$ depends on the current time, t_n, the future time, $t_n + \tau_n$, and the stored times vector, $\mathbf{T}$. One of three methods may be used to generate the number of reaction and diffusion events in the time interval $[t_n, t_n + \tau_n)$. We decide among these methods based on the criteria below:

- If $\mathbf{T} = t_n$, then apply **Unrestricted path**.
- If $\mathbf{T}(I - 1) < t_n + \tau_n < \mathbf{T}(I)$ with I an index into the vector $\mathbf{T}$, then apply **Poisson bridge path**.
- If $t_n < \mathbf{T}(end) < t_n + \tau_n$, then apply **Partial Poisson path**.

Afterwards, we calculate the next potential system state $\mathbf{X}^*(t_n + \tau_n)$ as

$$\mathbf{X}^*(t_n + \tau_n) = \mathbf{X}(t_n) + \sum_{\mathbf{k}\in\mathbf{K}}\sum_{j=1}^{M} \boldsymbol{\nu}_{j\mathbf{k}}\omega_{\mathbf{k}}\alpha_{j\mathbf{k}} + \sum_{\mathbf{k}\in\mathbf{K}}\sum_{e\in\mathcal{E}}\sum_{i=1}^{N} \boldsymbol{\nu}^{\mathrm{d}}_{i\mathbf{k}e}\alpha_{i\mathbf{k}e}, \tag{3.17}$$

where $\alpha_{j\mathbf{k}}$ is the number of firings of reaction $R_{j\mathbf{k}}$, $\alpha_{i\mathbf{k}e}$ is the number of firings of diffusion $R^d_{i\mathbf{k}e}$ in $[t_n, t_n+\tau_n)$, and $\omega_{\mathbf{k}}$ is a $1 \times K$ vector with 1 in the $\mathbf{k}$th position and 0's elsewhere.

Unrestricted path When $\mathbf{T} = t_n$, no restrictions exist on the sample trajectory. For all $j = 1, \ldots, M$ and $\mathbf{k} \in \mathbf{K}$, we generate the number of reactions that fired in the time interval $[t_n, t_n + \tau_n)$ as

$$\alpha_{j\mathbf{k}} = \mathcal{P}_{j\mathbf{k}}(a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t))\tau). \tag{3.18}$$

For all $i = 1, \ldots, N$, $\mathbf{k} \in \mathbf{K}$, and $e \in \mathcal{E}$, we generate the number of diffusions that fired in the time interval $[t_n, t_n + \tau_n)$ as

$$\alpha_{i\mathbf{k}e} = \mathcal{P}_{i\mathbf{k}e}(a^{\mathrm{d}}_{i\mathbf{k}e}(\mathbf{X}(t))\tau). \tag{3.19}$$

Poisson bridge path When $t_n + \tau_n < \mathbf{T}(\text{end})$, we have stored one or more rejected leap steps. Theorem 3.2.1 justifies the use of binomial random variables to generate the number of reaction and diffusion events that occur when restrictions exist on the sample trajectory.

We find index I, such that $\mathbf{T}(I-1) < t_n + \tau_n < \mathbf{T}(I)$. To calculate r, we set $u = t_n + \tau_n$, $s = \mathbf{T}(I-1)$, and $t = \mathbf{T}(I)$. Thus, $r = \dfrac{(t_n + \tau_n) - \mathbf{T}(I-1)}{\mathbf{T}(I) - \mathbf{T}(I-1)}$. For all $j = 1, \ldots, M$ and $\mathbf{k} \in \mathbf{K}$, the number of reactions that fired in the time interval $[t_n, t_n + \tau_n)$ is found as

$$\alpha_{j\mathbf{k}} = B_{j\mathbf{k}}\left(\rho_{j\mathbf{k}}^{\mathbf{T}(I)} - \rho_{j\mathbf{k}}^{\mathbf{T}(I-1)}, r\right) + \left(\rho_{j\mathbf{k}}^{\mathbf{T}(I-1)} - \rho_{j\mathbf{k}}^{t_n}\right). \tag{3.20}$$

For all $i = 1, \ldots, N$, $\mathbf{k} \in \mathbf{K}$, and $e \in \mathcal{E}$, the number of diffusions that fired in the time interval $[t_n, t_n + \tau_n)$ is found as

$$\alpha_{i\mathbf{k}e} = B_{i\mathbf{k}e}\left(\rho_{i\mathbf{k}e}^{\mathbf{T}(I)} - \rho_{i\mathbf{k}e}^{\mathbf{T}(I-1)}, r\right) + \left(\rho_{i\mathbf{k}e}^{\mathbf{T}(I-1)} - \rho_{i\mathbf{k}e}^{t_n}\right). \tag{3.21}$$

Note that $B_{\{\cdot\}}(\cdot, r)$ is the binomial distribution and $\left(\rho_{\{\cdot\}}^{\mathbf{T}(I-1)} - \rho_{\{\cdot\}}^{t_n}\right)$ accounts for the reaction and diffusion events that may have occurred in the time interval $[t_n, \mathbf{T}(I-1))$.

Partial Poisson path When $t_n < \mathbf{T}(end) < t_n + \tau_n$, we require that the sample trajectory passes through the state $\mathbf{X}(\mathbf{T}(end))$ before reaching the future state $\mathbf{X}(t_n + \tau_n)$. Therefore, we only need to generate reaction and diffusion events in the time interval $[\mathbf{T}(end), t_n + \tau_n)$.

We calculate the change in time $\triangle\tau = (t_n + \tau_n) - \mathbf{T}(end)$. Then, for all $j = 1, \ldots, M$ and $\mathbf{k} \in \mathbf{K}$, we create the number of reactions that fired in the time interval $[t_n, t_n + \tau_n)$ using

$$\alpha_{j\mathbf{k}} = \mathcal{P}_{j\mathbf{k}}(a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t))\triangle\tau) + \left(\rho_{j\mathbf{k}}^{\mathbf{T}(end)} - \rho_{j\mathbf{k}}^{t_n}\right). \tag{3.22}$$

For all $i = 1, \ldots, N$, $\mathbf{k} \in \mathbf{K}$, and $e \in \mathcal{E}$, we create the number of diffusions that fired in the time interval $[t_n, t_n + \tau_n)$ using

$$\alpha_{i\mathbf{k}e} = \mathcal{P}_{i\mathbf{k}e}(a_{i\mathbf{k}e}(\mathbf{X}(t))\triangle\tau) + \left(\rho_{i\mathbf{k}e}^{\mathbf{T}(end)} - \rho_{i\mathbf{k}e}^{t_n}\right). \tag{3.23}$$

Note that $\left(\rho_{\{\cdot\}}^{\mathbf{T}(end)} - \rho_{\{\cdot\}}^{t_n}\right)$ corresponds the reaction and diffusion events that occurred in the time interval $[t_n, \mathbf{T}(end))$.

Acceptence of the leap step

When the leap condition is satisfied, we update the time and the system state. We change the time t_n to time $t_n + \tau_n$ and set the next system state $\mathbf{X}(t_n + \tau_n) = \mathbf{X}^*(t_n + \tau_n)$. We must also update the stored times vector, $\mathbf{T}$, and stored reaction-diffusion matrix, $\mathcal{R}$.

We start by saving the information for the sample trajectory at $t_n + \tau_n$. We create a new collection

$\mathcal{R}_{t_n+\tau_n}$ for time $t_n + \tau_n$. For each reaction total $\rho_{j\mathbf{k}}^{t_n+\tau_n} \in \mathcal{R}_{t_n+\tau_n}$, we set

$$\rho_{j\mathbf{k}}^{t_n+\tau_n} = \rho_{j\mathbf{k}}^{t_n} + \alpha_{j\mathbf{k}}, \tag{3.24}$$

where $\alpha_{j\mathbf{k}}$ is calculated using equation (3.18), (3.20), or (3.22), and for each diffusion total $\rho_{i\mathbf{k}e}^{t_n+\tau_n} \in \mathcal{R}_{t_n+\tau_n}$, we have

$$\rho_{i\mathbf{k}e}^{t_n+\tau_n} = \rho_{i\mathbf{k}e}^{t_n} + \alpha_{i\mathbf{k}e}, \tag{3.25}$$

where $\alpha_{i\mathbf{k}e}$ is calculated using equation (3.19), (3.21), or (3.23).

When $\mathbf{T}(end) < t_n + \tau_n$, the Markov property of the biochemical system allows us to forget all sample trajectory information prior to $t_n + \tau_n$. Thus, we assign $\mathbf{T} = t_n + \tau_n$ and assign $\mathcal{R} = \mathcal{R}_{t_n+\tau_n}$. If $\mathbf{T}(I-1) < t_n + \tau_n < \mathbf{T}(I)$, we remove all times $t \in \mathbf{T}$ where $t \leq \mathbf{T}(I-1)$. We also eliminate all corresponding collections $\mathcal{R}_t \in \mathcal{R}$. Afterwards, we insert the time $t_n + \tau_n$ at the beginning of $\mathbf{T}$ and the new collection $\mathcal{R}_{t_n+\tau_n}$ at the beginning of $\mathcal{R}$.

Rejection of the leap step

If the leap condition is not satisfied, we reject the updated system. We may also need to update the stored time vector, $\mathbf{T}$, and the stored reaction-diffusion matrix, $\mathcal{R}$. If $t_n + \tau_n \leq \mathbf{T}(end)$, no new information is required to condition the sample trajectory. However, if $\mathbf{T}(end) < t_n + \tau_n$, an additional restriction has been set on the sample trajectory. We create a new collection $\mathcal{R}_{t_n+\tau_n}$. For every reaction total $\rho_{j\mathbf{k}}^{t_n+\tau_n} \in \mathcal{R}_{t_n+\tau_n}$, we have

$$\rho_{j\mathbf{k}}^{t_n+\tau_n} = \rho_{j\mathbf{k}}^{t_n} + \alpha_{j\mathbf{k}},$$

where $\alpha_{j\mathbf{k}}$ is found using equation (3.18) or (3.22), and for every diffusion total $\rho_{i\mathbf{k}e}^{t_n+\tau_n} \in \mathcal{R}_{t_n+\tau_n}$, we have

$$\rho_{i\mathbf{k}e}^{t_n+\tau_n} = \rho_{i\mathbf{k}e}^{t_n} + \alpha_{i\mathbf{k}e},$$

where $\alpha_{i\mathbf{k}e}$ is determined using equation (3.19) or (3.23). Afterwards, we add the time $t_n + \tau_n$ to the end of $\mathbf{T}$ and its corresponding $\mathcal{R}_{t_n+\tau_n}$ to the end of $\mathcal{R}$.

3.3 New stepping strategy

The goal of our new stepping strategy is to minimize the number of rejected leap steps while selecting τ large enough to see improvements in simulation time compared to the ISSA. When developing our new adaptive stepping strategy, we consider the sets X_L and X_S separately. We derive the method for selecting an appropriate time-step h_{n+1}^1 for the large population set X_L and an appropriate time-step h_{n+1}^2 for the small population set X_S, then select the new time-step h_{n+1} as the minimum of h_{n+1}^1 and h_{n+1}^2.

3.3.1 Theoretical foundation

The method we use for adapting the length of the current time-step, h_n, is based on an *integral controller* from control theory. This method is valid for deterministic models [27]; however, has shown excellent results when applied to our stochastic models. By attempting to maintain the local change below a threshold, ϵ, the method aims to minimize the computational cost, while satisfying the accuracy criteria.

Step selection for X_L

We use the following method to select a time-step for the large population set X_L. Let $\mathbf{Y}(t_n)$ be the current solution of a system at time t_n. At every time t_n, we define the following system of equations

$$\begin{cases} e_n^* = \left\| \frac{\mathbf{Y}(t_n+h_n)-\mathbf{Y}(t_n)}{\mathbf{Y}(t_n)} \right\|_\infty \\ t_{n+1} = t_n + h_n \end{cases}$$

where e_n^* is the maximum relative local change corresponding to the next solution $\mathbf{Y}(t_n+h_n)$ at time t_n+h_n. For every t_n, we want to ensure that the relative local change e_n^* is strictly less than a prescribed variation tolerance level ϵ, i.e.,

$$\left\| \frac{\mathbf{Y}(t_n + h_n) - \mathbf{Y}(t_n)}{\mathbf{Y}(t_n)} \right\|_\infty < \epsilon.$$

Assume that in the asymptotic regime, the following relationship exists between the relative local change e_n^* and the current time-step h_n

$$e_n^* = \phi_n h_n^\gamma, \tag{3.26}$$

where γ is a parameter and ϕ_n is the principal error function. The function ϕ_n may be derived from the Itô-Taylor series expansion of the solution. As it is generally difficult to get an exact formulation of the principle error function ϕ_n, we focus on adjusting the length of the time-step h_n to control the relative local change e_n^*.

For the next step t_{n+1}, the largest relative local change e_{n+1}^* we are willing to introduce to the system is ϵ. Thus, from equation (3.26) we get

$$e_{n+1}^* = \phi_{n+1} h_{n+1}^\gamma = \epsilon. \tag{3.27}$$

We can combine equations (3.26) and (3.27) as

$$\frac{e_n^*}{\epsilon} = \frac{\phi_n h_n^\gamma}{\phi_{n+1} h_{n+1}^\gamma}. \tag{3.28}$$

For deterministic problems, often $\phi_n \simeq \phi_{n+1}$. Thus, solving equation (3.28) for h_{n+1}, we have the following stepping strategy [27],

$$h_{n+1} = h_n \left(\frac{\epsilon}{e_n^*} \right)^{\frac{1}{\gamma}}. \tag{3.29}$$

If the current relative local change $e_n^* > \epsilon$, equation (3.29) decreases the length of h_{n+1} to reduce the variation accruing in the system. If the current relative local change $e_n^* < \epsilon$, equation (3.29) increases the length of h_{n+1} to help shorten the overall simulation duration.

In practice equation (3.29) may cause numerical solution instability when rapid oscillations occur in consecutive time-steps. Thus, we can introduce both a minimal step decrease factor, f_{min}, and maximal step increase factor, f_{max}, to dampen these oscillations. To decrease the number of rejected steps, the prescribed variation tolerance level ϵ is often replaced by the safety factor $\theta_1 \epsilon$ where $\theta_1 \leq 1$.

Combining these assumptions yields the stepping strategy for the set X_L,

$$h_{n+1}^1 = h_n \min \left(f_{max}, \max \left(f_{min}, \left(\frac{\theta_1 \epsilon}{e_n^*} \right)^{\frac{1}{\gamma}} \right) \right), \tag{3.30}$$

where $1 < f_{max} \leq 2$ and $0 < f_{min} < 1$. This stepping strategy has been successfully applied to stochastic continuous models [6]. We apply this technique for the first time to stochastic discrete models; and we found it to be very efficient.

Step selection for X_S

We use an analogous derivation method to select a time-step for the small population set X_S. At each time t_n, we define the following system of equations

$$\begin{cases} E_n^* = \|\mathbf{Y}(t_n + h_n) - \mathbf{Y}(t_n)\|_\infty \\ t_{n+1} = t_n + h_n \end{cases}$$

where E_n^* is the absolute local change corresponding to the next solution $\mathbf{Y}(t_n + h_n)$ at time $t_n + h_n$.

Now for time t_n, we want the absolute local change E_n^* to be less than or equal to the minimum consumption of a species S_i, which we take as 1. Thus

$$\|\mathbf{Y}(t_n + h_n) - \mathbf{Y}(t_n)\|_\infty \leq 1.$$

Like equation (3.26), a similar relationship exists between the absolute local error E_n^* and the current time-step h_n,

$$E_n^* = \varphi_n h_n^\gamma, \tag{3.31}$$

where φ_n is the principle error function. Using equation (3.31) and following the same procedure used to derive equation (3.28), we have the stepping strategy [27],

$$h_{n+1} = h_n \left(\frac{1}{E_n^*}\right)^{\frac{1}{\gamma}}. \tag{3.32}$$

This time, to reduce the number of rejected steps, we replace 1 with the safety factor θ_2, where $\theta_2 \leq 1$. Thus, the stepping strategy for the set X_S is

$$h_{n+1}^2 = h_n \min\left(f_{max}, \max\left(f_{min}, \left(\frac{\theta_2}{E_n^*}\right)^{\frac{1}{\gamma}}\right)\right), \tag{3.33}$$

where $1 < f_{max} \leq 2$ and $0 < f_{min} < 1$.

3.3.2 Application to heterogeneous systems

Calculation of the initial τ

In our adaptive tau-leaping algorithm, we determine the initial time-step τ using the method presented in [45] which extends the Cao et. al. [9] τ-selection process outlined in Section 2.1.5 to spatially heterogeneous systems.

We calculate the time-step τ according to

$$\tau = \min_{\mathbf{k} \in \mathbf{K}} \{\tau_{\mathbf{k}}\}. \tag{3.34}$$

For each voxel $\mathbf{k}$, we have

$$\tau_{\mathbf{k}} = \min_{i \in I_{rs}} \left\{ \frac{\max\{\epsilon X_{i\mathbf{k}}(t)/g_i, 1\}}{|\hat{\mu}_{i\mathbf{k}}(\mathbf{X}(t))|}, \frac{(\max\{\epsilon X_{i\mathbf{k}}(t)/g_i, 1\})^2}{(\hat{\sigma}_{i\mathbf{k}}(\mathbf{X}(t)))^2} \right\},$$

where

$$\hat{\mu}_{i\mathbf{k}}(\mathbf{X}(t)) = \sum_{j=1}^{M} \nu_{ij\mathbf{k}} a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t)) + \sum_{e=1}^{|\mathcal{E}|} (-1) a_{i\mathbf{k}e}(\mathbf{X}(t)),$$

$$\hat{\sigma}_{i\mathbf{k}}(\mathbf{X}(t))^2 = \sum_{j=1}^{M} (\nu_{ij\mathbf{k}})^2 a_{j\mathbf{k}}(\mathbf{X}_{\mathbf{k}}(t)) + \sum_{e=1}^{|\mathcal{E}|} a_{i\mathbf{k}e}(\mathbf{X}(t)).$$

Calculation of the subsequent τ_n

We use the following parameters in equations (3.30) and (3.33) to adapt all subsequent time-steps τ_n. We take $\gamma = 1$. We let $\theta_1 = 0.8$, $\theta_2 = 1$, $f_{min} = 0.5$, $f_{max} = 1.5$, and $\epsilon = \max_i \{\epsilon_i\}$. These specific values were found from extensive simulation testing.

Equation (3.30), used to adapt the time-step τ_{n+1}^1 for the large population set X_L, requires an estimate of relative local change. We get this value $e_n^* = e_n$ from equation (3.12). Thus, the equation used to adapt the time-step τ^1_{n+1} is

$$\tau^1_{n+1} = \tau_n \min\left(1.5, \max\left(0.5, \frac{0.8\epsilon_i}{e_n}\right)\right). \tag{3.35}$$

Likewise, equation (3.33), used to adapt the time-step τ^2_{n+1} for the small population set X_S, needs an estimate of absolute local change. The value $E^*_n = E_n$ comes from equation (3.15). Thus, the equation used to adapt the time-step τ^2_{n+1} is

$$\tau^2_{n+1} = \tau_n \min\left(1.5, \max\left(0.5, \frac{1}{E_n}\right)\right). \tag{3.36}$$

Once τ^1_{n+1} and τ^2_{n+1} have been calculated according to equations (3.35) and (3.36), we choose the next step size τ_{n+1} as

$$\tau_{n+1} = \min\left(\tau^1_{n+1}, \tau^2_{n+1}\right). \tag{3.37}$$

3.4 New adaptive tau-leaping algorithm

Below we propose a new variable time-stepping algorithm for the spatial Poisson tau-leaping scheme for the Reaction-Diffusion Master Equation [32].

Algorithm

```
0. (Parameters) Set tolerance $\epsilon$, control factors $\theta_1$, $\theta_2$, and increase and decrease factors $f_{min}$, $f_{max}$.
   Calculate critical thresholds $c_i$ using equation (3.4).
1. (Initialize) Set $\mathbf{X} \leftarrow \mathbf{x}_0$ at $t \leftarrow 0$ and the initial step $\tau$ using equation (3.34).
2. WHILE ($t < T$) do 3–6.
3. (State Generation)
   3.1. IF no future restriction exists:
        a. Generate Poisson distributions:
           Compute $\alpha_{i\mathbf{k}}$ using equation (3.18).
           Compute $\alpha_{i\mathbf{k}e}$ using equation (3.19).
        b. Compute $\mathbf{X}^*$ using equation (3.17).
   3.2. ELSE IF future restriction(s) exist AND step after restriction, $\mathbf{T}(end) < t + \tau$:
        a. Generate partial Poisson distributions:
           Compute $\alpha_{i\mathbf{k}}$ using equation (3.22).
           Compute $\alpha_{i\mathbf{k}e}$ using equation (3.23).
        b. Compute $\mathbf{X}^*$ using equation (3.17).
```

- 3.3. ELSE (future restriction(s) exist AND step in between restrictions, $\mathbf{T}(I-1) < t+\tau < \mathbf{T}(I)$):
  - a. Generate binomial distributions:
    Compute $\alpha_{i\mathbf{k}}$ using equation (3.20).
    Compute $\alpha_{i\mathbf{k}e}$ using equation (3.21).
  - b. Compute $\mathbf{X}^*$ using equation (3.17).

4. (Leap Validation)
   - a. Partition $\{X_{i\mathbf{k}}(t)\}$ into X_L and X_S using equations (3.9) and (3.10).
   - b. For X_L, validate leap condition using equation (3.13).
   - c. For X_S, validate leap condition using equation (3.16).

5. (Leap Acceptance) IF accuracy criteria is met, accept step.
   - a. Update $t \leftarrow t+\tau$ and set $\mathbf{X} \leftarrow \mathbf{X}^*$.
   - b. Create $\mathcal{R}_{t+\tau}$ using equations (3.24) and (3.25).
   - c1. IF $\mathbf{T}(end) < t+\tau$, clear future restriction(s).
     Set $\mathbf{T} \leftarrow t+\tau$ and $\mathcal{R} \leftarrow \mathcal{R}_{t+\tau}$.
   - c2. ELSE, $(\mathbf{T}(I-1) < t+\tau < \mathbf{T}(I))$.
     Remove all $t' \leq \mathbf{T}(I-1)$ and $\mathcal{R}'_t \in \mathcal{R}$. Add $t+\tau$ to $\mathbf{T}$ and $\mathcal{R}_{t+\tau}$ to $\mathcal{R}$.
   - d. Update τ using equation (3.37).

6. (Leap Rejection) IF accuracy criteria is not met, reject step.
   - a. IF $\mathbf{T}(end) < t+\tau$, create new future restriction.
     Create $\mathcal{R}_{t+\tau}$ using equations (3.24) and (3.25). Add $t+\tau$ to $\mathbf{T}$ and $\mathcal{R}_{t+\tau}$ to $\mathcal{R}$.
   - b. Update τ using equation (3.37).

This τ-selection procedure is shown to produce results in excellent agreement with those generated with the exact ISSA. The numerical experiments demonstrate the efficiency and the accuracy of the new method.

Chapter 4

Numerical results

We exhibit the advantages of the new variable time-stepping strategy for the spatial tau-leaping method on three models of heterogeneous biochemical systems of practical interest. Poisson tau-leaping approaches work best when the number of molecules in the model remains bounded away from zero throughout the simulation. Consequently, these models have been integrated over a time interval where the molecular populations obey this criteria.

The model systems used to demonstrate the effectiveness of our proposed tau-leaping algorithm contain a few species each. Thus, the species names, e.g. U and V, have replaced the general S_n notation in chemical reaction and diffusion descriptions. The species names have also replaced the general x_n notation in the propensity functions. Furthermore, this naming lets us drop the subscript indicating species type; $U_{\mathbf{k}}$ is the number of species U in voxel $\Omega_{\mathbf{k}}$. We have also removed the voxel $\mathbf{k}$ subscripts from the reaction propensity functions and both the voxel $\mathbf{k}$ and diffusion direction e subscripts from the diffusion propensity functions.

For each model, we demonstrate the accuracy of the proposed method by comparing the numerical results of the exact ISSA to the numerical results of the approximate adaptive scheme, for a sequence of tolerances. We have also provided the mean number of attempted, accepted, and rejected steps, the ratio of the mean relative change to the tolerance, and the speed up over the ISSA using data collected from 10,000 simulations for each tolerance ϵ. The relative change values considered are the maximum relative change values on the individual trajectories. We have calculated the speed-up of the new variable tau-leaping scheme by taking the ratio of the time taken by the ISSA compared to the computational time required by the new method

$$\text{speed-up} = \frac{\text{time for ISSA}}{\text{time for adaptive tau-leaping}}.$$

4.1 Turing patterns model

The first model under consideration, the Turing patterns model, was studied in [7]. A unique feature of this model is the stable Turing patterns that appear in the steady-state of this reaction-diffusion system. This one-dimensional model contains two species U and V, where the volume Ω is divided into $K = 40$ voxels of edge length $h = \frac{1}{40}$.

For the ith voxel, the reactions in this model are

$$\emptyset \underset{k_2}{\overset{k_1}{\rightleftarrows}} U_i, \qquad \emptyset \overset{k_3}{\rightarrow} V_i, \qquad 2U_i + V_i \overset{k_4}{\rightarrow} 3U_i,$$

and the diffusions are

$$U_1 \underset{d_u}{\overset{d_u}{\rightleftarrows}} U_2 \underset{d_u}{\overset{d_u}{\rightleftarrows}} \dots \underset{d_u}{\overset{d_u}{\rightleftarrows}} U_K, \qquad V_1 \underset{d_v}{\overset{d_v}{\rightleftarrows}} V_2 \underset{d_v}{\overset{d_v}{\rightleftarrows}} \dots \underset{d_v}{\overset{d_v}{\rightleftarrows}} V_K.$$

The propensity functions for this model are then

$$\begin{aligned} a_1(t) &= k_1 h, & a_4(t) &= \frac{k_4}{h^2} U_i(t)(U_i(t)-1)V_i(t), \\ a_2(t) &= k_2 U_i(t), & a_5(t) &= a_6(t) = d_u U_i(t), \\ a_3(t) &= k_3 h, & a_7(t) &= a_8(t) = d_v V_i(t), \end{aligned}$$

with rate constants $\boldsymbol{k} = [4 \times 10^3, 2, 1.2 \times 10^4, 6.25 \times 10^{-8}]$ and $\boldsymbol{D} = [10^{-3}, 10^{-1}]$. The standard form of the propensity function for the third-order reaction $2U_i + V_i \overset{k_4}{\rightarrow} 3U_i$ includes the fraction $\frac{1}{2}$. This fraction has been absorbed into the rate constant k_4. The initial conditions used are $U_{i0} = 25$ and $V_{i0} = 75$ per voxel.

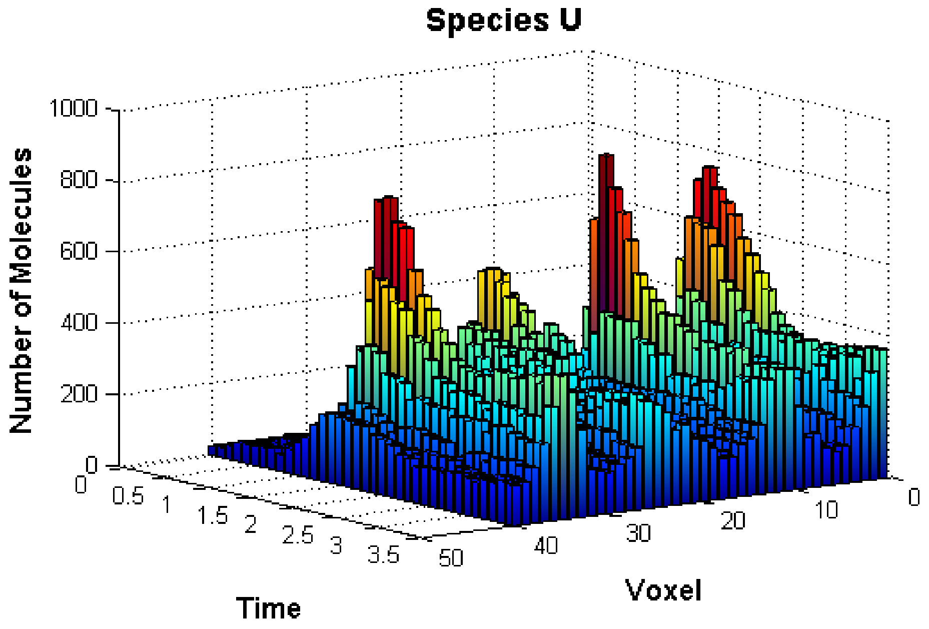
Finally, the stoichiometric matrix per voxel for this model is

$$\nu = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

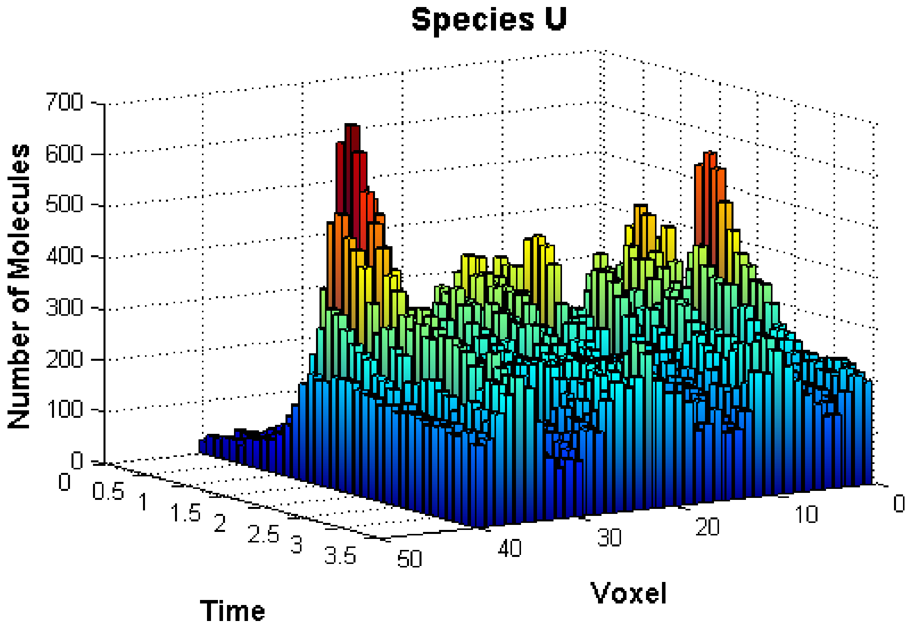
For the time interval $t = [0, 3.5]$, we simulated the dynamics of this model on $10,000$ trajectories using the ISSA and on $10,000$ trajectories each using the proposed adaptive tau-leaping scheme with the following accuracy criteria, $\epsilon = [0.2, 0.4, 0.8]$.

Figure 4.1 shows two different sample trajectories of the Turing patterns model for species U in the time interval $t = [0, 3.5]$. Figure 4.1a was generated using the exact ISSA and Figure 4.1b was generated using the proposed adaptive spatial Poisson tau-leaping method with tolerance $\epsilon = 0.4$. Figure 4.2 provides the corresponding sample trajectories for species V. Figure 4.2a was created using the exact ISSA and Figure 4.2b was created using the adaptive spatial Poisson tau-leaping method with tolerance $\epsilon = 0.4$. The bar height in these figures indicates the number of molecules of a species in a particular voxel at a specific time with the final voxel distribution displayed at the front. The stable Turing patterns are visible in both the ISSA and adaptive tau figures for species U. We also observe similar system behaviour with the location of the peaks and the species distribution at the final time $t = 3.5$.

In Table 4.1, we record the mean number of attempted steps for the new adaptive tau-leaping method and the relative local changes produced, along with the mean number of accepted and rejected steps, for the sequence of tolerances 0.2, 0.4, and 0.8. This table also shows the speed-up of the proposed variable time-stepping algorithm over the ISSA. Note that for these tolerances the new strategy is one to two orders of magnitude faster than the ISSA. Additionally, the ratio between the number of rejected steps and the

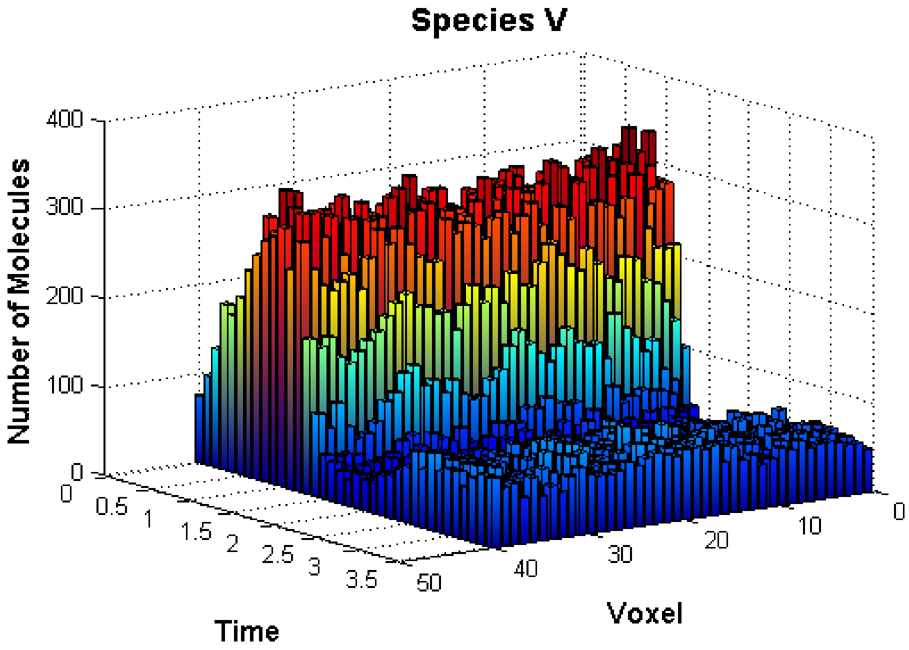


(a) ISSA

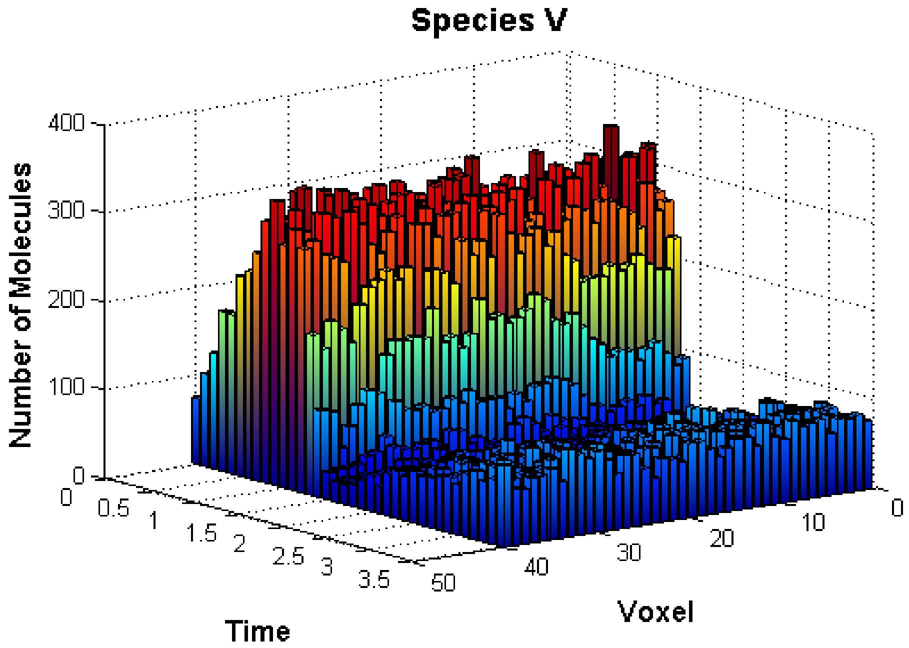


(b) Adaptive tau-leaping algorithm, tolerance $\epsilon = 0.4$

Figure 4.1: Turing patterns model: sample trajectories for species U



(a) ISSA



(b) Adaptive tau-leaping algorithm, tolerance $\epsilon = 0.4$

Figure 4.2: Turing patterns model: sample trajectories for species V

ϵ	Attempted steps	Accepted steps	Rejected steps	Relative change	Speed-up over ISSA
8×10^{-1}	7203	5835	1368	1.00	97.79
4×10^{-1}	25803	20705	5098	1.00	41.65
2×10^{-1}	126785	96171	30614	1.00	10.63

Table 4.1: The Turing patterns model: the number of steps taken by the adaptive tau-leaping scheme and speed-up over the ISSA.

total number of attempted steps is between 19% and 24% for the tolerance tried.

Figure 4.3 compares the histograms over 10,000 trajectories obtained using the variable tau-leaping algorithm with tolerances 0.2, 0.4, and 0.8 and the exact ISSA at time $t = 3.5$. The data for these figures was collected by calculating the total number of molecules of each species across all voxels $\Omega_{\mathbf{k}}$ at the end of the simulation. The agreement is excellent, illustrating the accuracy of the proposed adaptive method.

4.2 Production-annihilation model

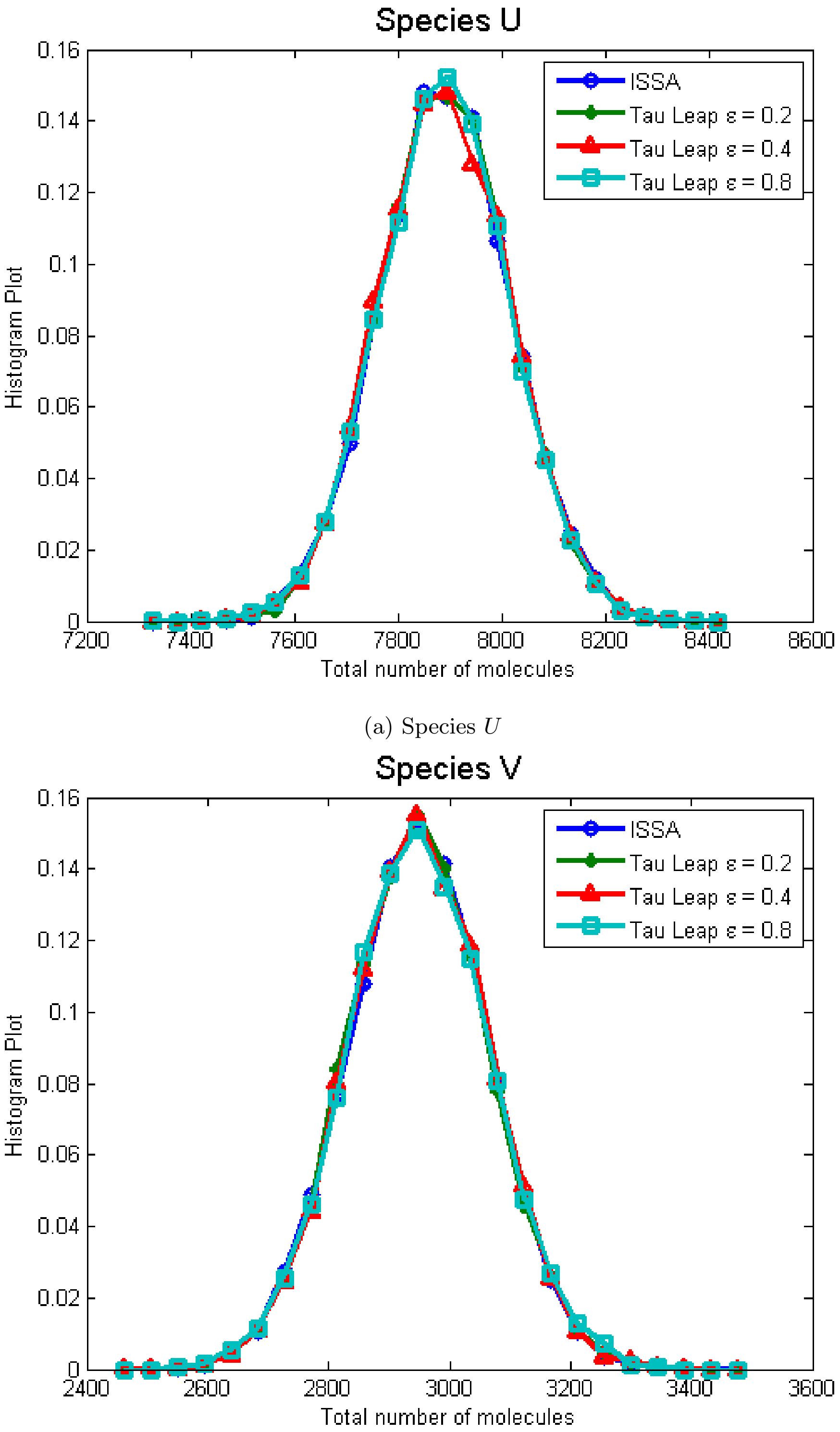
The second model analyzed, the production-annihilation model, is the extension of a one-dimensional system in [14] to a two-dimensional system. This two-dimensional system includes species A and B and the volume Ω is divided into $K = 10 \times 10$ voxels of edge length $h = \frac{1}{10}$. One aspect to note about the model is that the production of both species is restricted to a subset of the system's volume.

The reactions in the i, jth voxel are

$$\begin{aligned} &A_{i,j} + A_{i,j} \xrightarrow{k_1} \emptyset, \qquad && \emptyset \xrightarrow{k_3} A_{i,j} \quad \text{for } i = 1, 2, ..., 9,\ j = 1, 2, ..., 9, \\ &A_{i,j} + B_{i,j} \xrightarrow{k_2} \emptyset, \qquad && \emptyset \xrightarrow{k_4} B_{i,j} \quad \text{for } i = 4, 5, ..., 10,\ j = 4, 5, ..., 10 \end{aligned}$$

and the diffusions include

$$\begin{aligned} &A_{i,j} \xrightarrow{d} A_{i\pm1,j}, \qquad && B_{i,j} \xrightarrow{d} B_{i\pm1,j}, \\ &A_{i,j} \xrightarrow{d} A_{i,j\pm1}, \qquad && B_{i,j} \xrightarrow{d} B_{i,j\pm1}. \end{aligned}$$



(a) Species U

(b) Species V

Figure 4.3: Turing patterns model: histograms for ISSA and adaptive tau algorithm at $t = 3.5$, tolerances $\epsilon = 0.2, 0.4, 0.8$.

ϵ	Attempted steps	Accepted steps	Rejected steps	Relative change	Speed-up over ISSA
4×10^{-1}	5104	4437	667	0.99998	138.67
2×10^{-1}	22234	18803	3431	1.00	38.45
10^{-1}	149354	130503	18851	1.00	6.09

Table 4.2: The production-annihilation model: the number of steps taken by the adaptive tau-leaping scheme and speed-up over the ISSA.

Therefore, the propensity functions are

$$\begin{aligned} a_1(t) &= k_1 A_{i,j}(t)(A_{i,j}(t)-1), & a_4(t) &= k_4, \\ a_2(t) &= k_2 A_{i,j}(t) B_{i,j}(t), & a_5(t) &= a_6(t) = a_7(t) = a_8(t) = dA_{i,j}(t), \\ a_3(t) &= k_3, & a_9(t) &= a_{10}(t) = a_{11}(t) = a_{12}(t) = dB_{i,j}(t), \end{aligned}$$

with rate constants $\boldsymbol{k} = [0.2, 9 \times 10^{-3}, 2000, 100]$ and $D = 0.9$. The standard form of the propensity function for the second-order reaction $A_{i,j} + A_{i,j} \xrightarrow{k_1} \emptyset$ includes the fraction $\frac{1}{2}$. This fraction has been absorbed into the rate constant k_1. The initial conditions per voxel are $A_{i,j0} = 500$ and $B_{i,j0} = 100$.

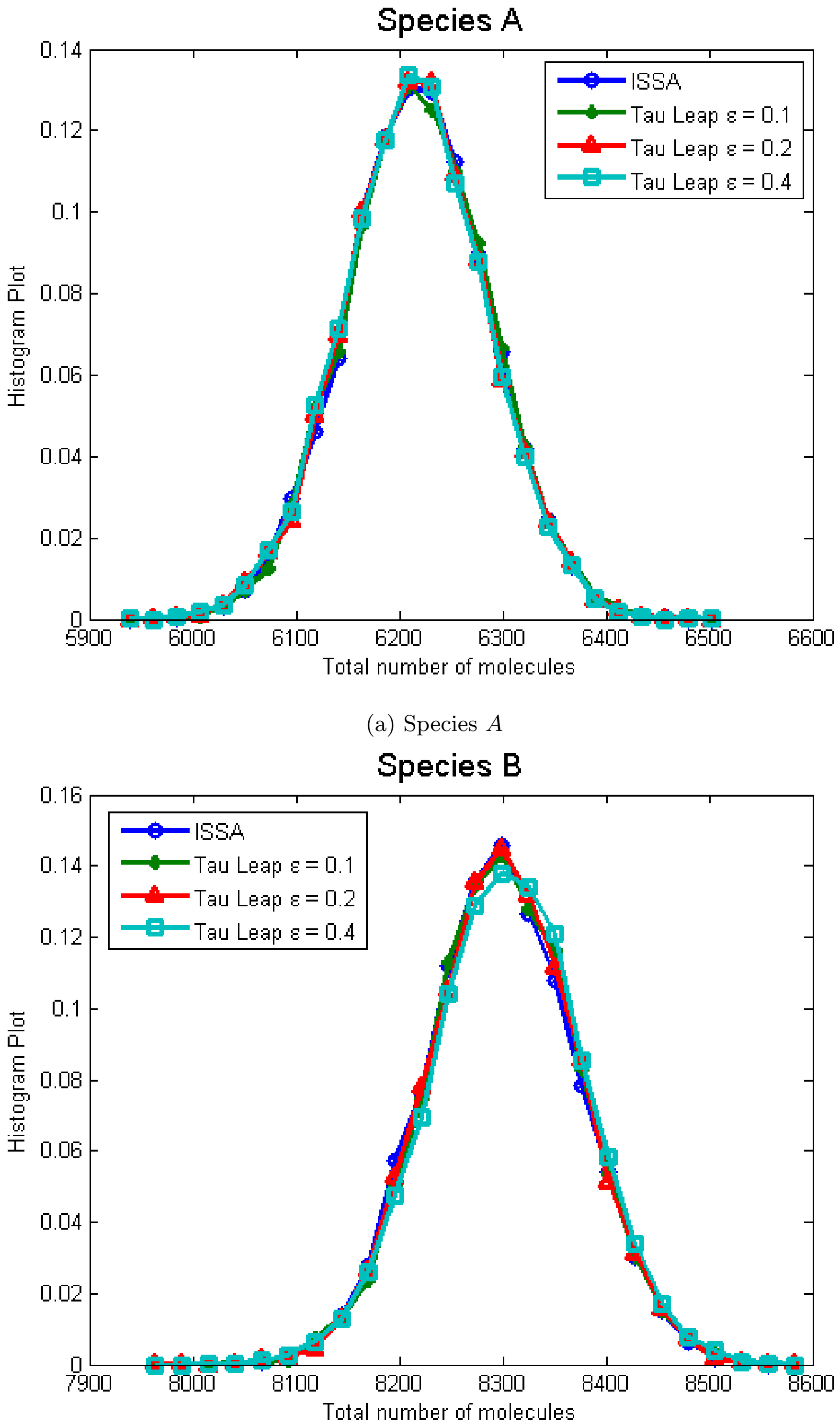
The stoichiometric matrix per voxel is

$$\nu = \begin{bmatrix} -2 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$

We have integrated the system in the time interval $t = [0, 1]$ with the variable step tau-leaping algorithm for $\epsilon = [0.1, 0.2, 0.4]$ and the ISSA, each over 10,000 trajectories.

Table 4.2 presents the mean number of attempted steps along with the mean accepted and the mean rejected steps for the variable step size tau-leaping strategy. Additionally, it records the ratio of the relative local change to the given tolerance and the speed-up of the proposed algorithm over the ISSA for the following sequence of imposed tolerances $\epsilon = [0.1, 0.2, 0.4]$. For this particular model, we see speed-ups of between 98 to 138 times. We also find that the ratio of the number of rejected steps to the total number of steps taken in the adaptive algorithm is quite low, ranging from 12% to 15%.

Figure 4.4 plots the histograms for each of the species at time $t = 1$, for the adaptive tau-leaping algorithm with tolerances 0.1, 0.2, and 0.4 and for the exact ISSA. Again, the data for these figures was collected by adding the total number of molecules of each species across all voxels $\Omega_{\mathbf{k}}$ at the end of the simulation. Again, these figures show that our adaptive strategy has exceptional accuracy.



(a) Species A

(b) Species B

Figure 4.4: Production-annihilation model: histograms for ISSA and adaptive tau algorithm at $t = 1$, tolerances $\epsilon = 0.1, 0.2, 0.4$.

4.3 Brusselator model

The final model tested, the Brusselator model, is found in [39]. This system is quite stiff, thus simulation efficiency depends on a variable time-step. The model has species U and V and divides the two-dimensional volume Ω into $K = 8 \times 8$ voxels of edge length $h = 1$.

Species U and V participate in the following reactions in the i, jth voxel

$$\begin{aligned} &\emptyset \xrightarrow{k_1} U_{i,j}, && 2U_{i,j} + V_{i,j} \xrightarrow{k_3} 3U_{i,j}, \\ &U_{i,j} \xrightarrow{k_2} V_{i,j}, && U_{i,j} \xrightarrow{k_4} \emptyset, \end{aligned}$$

and the following diffusions

$$\begin{aligned} &U_{i,j} \xrightarrow{d_u} U_{i\pm1,j}, && V_{i,j} \xrightarrow{d_v} V_{i\pm1,j}, \\ &U_{i,j} \xrightarrow{d_u} U_{i,j\pm1}, && V_{i,j} \xrightarrow{d_v} V_{i,j\pm1}. \end{aligned}$$

The propensity functions are

$$\begin{aligned} &a_1(t) = k_1 h^2, && a_4(t) = k_4 U_{i,j}(t), \\ &a_2(t) = k_2 U_{i,j}(t), && a_5(t) = a_6(t) = a_7(t) = a_8(t) = dU_{i,j}(t), \\ &a_3(t) = \frac{k_3}{h^4} U_{i,j}(t)(U_{i,j}(t) - 1)V_{i,j}(t), && a_9(t) = a_{10}(t) = a_{11}(t) = a_{12}(t) = dV_{i,j}(t), \end{aligned}$$

with rate constants $\boldsymbol{k} = [7000, 40, 5.0 \times 10^{-5}, 10]$ and $\boldsymbol{D} = [1, 1]$. The standard form of the propensity function for the third-order reaction $2U_{i,j} + V_{i,j} \xrightarrow{k_3} 3U_{i,j}$ includes the fraction $\frac{1}{2}$. This fraction has been absorbed into the rate constant k_3. The initial conditions per voxel are $U_{i,j0} = 100$ and $V_{i,j0} = 50$.

Lastly, the stoichiometric matrix per voxel is

$$\nu = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}.$$

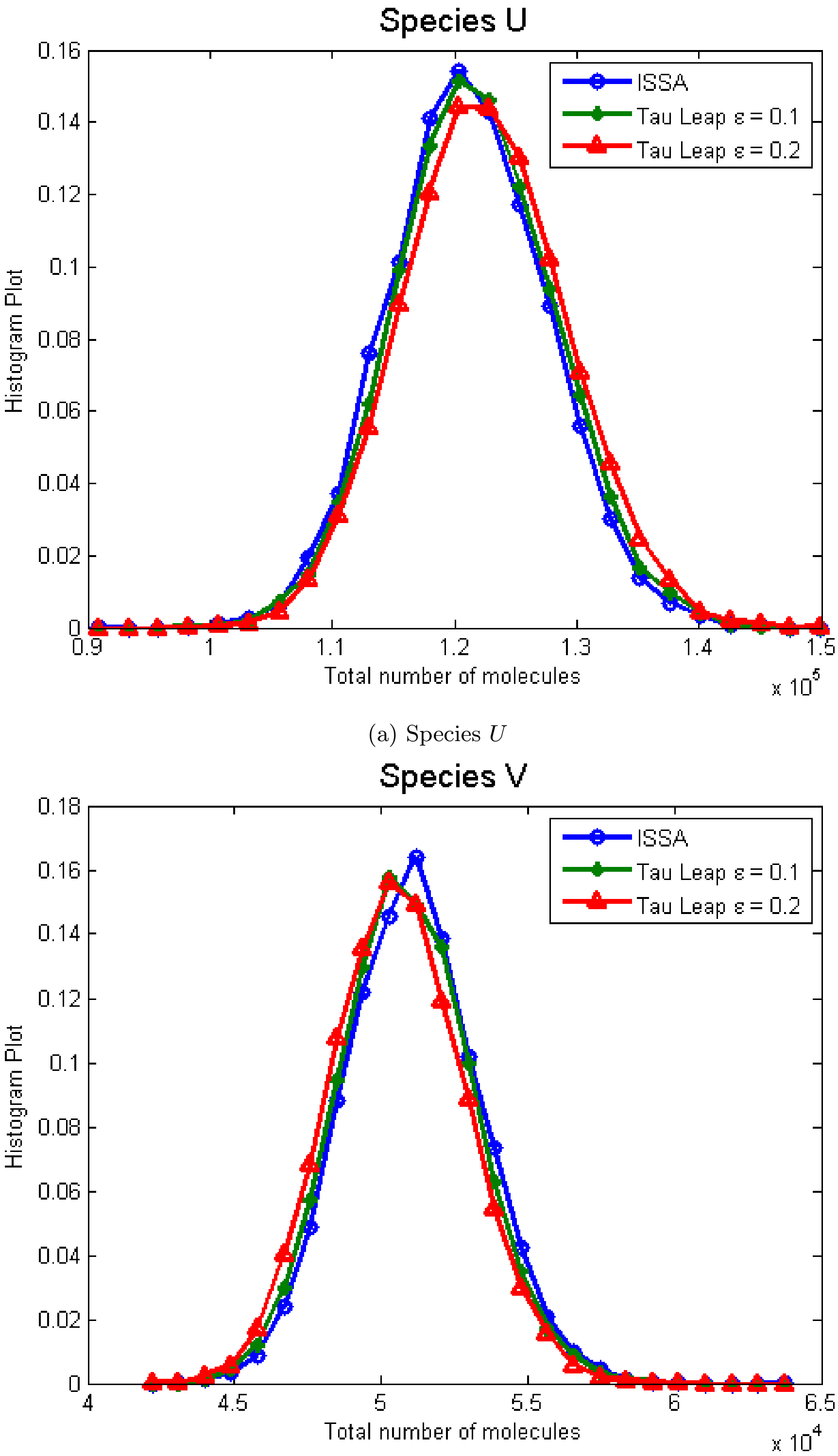
We follow the same simulation process as the first two models, integrating over the time duration $t = [0, 1]$ and with tolerances $\epsilon = [0.1, 0.2, 0.4]$.

Table 4.3 characterizes the results of our simulation of the Brusselator model. For the sequence of tolerances: $\epsilon = [0.1, 0.2, 0.4]$, we give the accuracy achieved by the variable step size tau-leaping scheme, the mean number of attempted, accepted, and rejected steps, and the speed-up over the ISSA. The ratio of steps rejected to total steps attempted is less than 16%, for the tolerances used. Moreover, the speed-up of our method over the ISSA is significant, from 78 to 387 times.

We observe in Figure 4.5 the match of the results of our adaptive tau-leaping scheme for tolerance $\epsilon = 0.1$ and $\epsilon = 0.2$ with results of the ISSA. As before, the molecular number totals were found by adding the total number of molecules of each species across all voxels $\Omega_{\mathbf{k}}$, at the end of the simulation. This further confirms that our proposed method gives very accurate results for a much lower computational cost.

ϵ	Attempted steps	Accepted steps	Rejected steps	Relative change	Speed-up over ISSA
4×10^{-1}	626	555	70	0.99687	387.61
2×10^{-1}	2099	1830	269	0.99948	211.74
10^{-1}	8724	7291	1433	1.00	78.10

Table 4.3: The Brusselator model: the number of steps taken by the adaptive tau-leaping scheme and speed-up over the ISSA.



(a) Species U

(b) Species V

Figure 4.5: Brusselator model: histograms for ISSA and adaptive tau algorithm at $t = 1$, tolerances $\epsilon = 0.1, 0.2$.

Chapter 5

Conclusion

Mathematical modeling and simulation of biochemical systems provide unique insight into the system dynamics. As cellular biology experimental techniques become more refined at the level of the single cell, these mathematical models and simulation will play an ever increasing role in understanding and predicting system dynamics under a variety of conditions. Additionally, these models and systems will help clarify the complex dependencies that exist in the various pathways in the cell [4]. Thus, the need for more efficient, but still accurate mathematical simulation algorithms will continue to grow.

This thesis analyzed an inhomogeneous, discrete stochastic model of biochemical kinetics, that of the Reaction-Diffusion Master Equation (RDME). This model accurately describes the dynamics of biochemical systems with some species in low numbers and whose system behaviour is dependent on spatial layout of the system. The study of biochemical systems with low numbers and spatial dependence is crucial for understanding cellular behaviour.

We have proposed a new adaptive time-stepping scheme targeting the spatial Poisson tau-leaping strategy for the RDME. The adaptive time-stepping scheme is based on an integral controller and helps address stiffness that is inherit to biochemical systems which include diffusion. We have also incorporated post-leap checking to address solution accuracy. Post-leap checking guarantees that the leap-condition is always satisfied. Combining time adaptivity with post-leap checking has allowed us to create an accurate and efficient simulation algorithm targeting the RDME. Our proposed algorithm achieves excellent accuracy while being orders of magnitude faster than the ISSA for the three tested models.

Our future work will consider a number of improvements to the presented algorithm including developing an effective technique for negative numbers. We also plan to apply our variable time-stepping strategy to semi-implicit tau-leaping schemes to improve the simulation time for very stiff biochemical problems.

Appendix A

Probability distributions

Definition A.0.1. The random variable X has a *uniform distribution* over range $[a, b]$, written $X \sim U(a, b)$, if it has probability density function (PDF)

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases}$$

Definition A.0.2. The random variable X has an *exponential distribution* with parameter λ, written $X \sim Exp(\lambda)$, if it has PDF

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Definition A.0.3. The random variable X has a *Poisson distribution* with parameter λ, written $X \sim \mathcal{P}(\lambda)$, if it has probability mass function (PMF)

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad \text{if } k = 0, 1, 2, \ldots$$

Definition A.0.4. The random variable X has a *binomial distribution* with parameters n and p, written $X \sim B(n, p)$, if it has PMF

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad \text{if } k = 0, 1, 2, \ldots, n$$

Appendix B

Poisson Process

Definition B.0.5. A stochastic process $\{N(t), t \geq 0\}$ is said to be a *counting process* if $N(t)$ represents the total number of "events" that occur by time t.

Definition B.0.6. The counting process $\{N(t), t \geq 0\}$ is said to be a *Poisson process having rate* λ, $\lambda > 0$, if

1. $N(0) = 0$.

2. The process has independent increments.

3. The number of events in any interval of length t is Poisson distributed with mean λt. That is, for all $s, t \geq 0$

$$P\{N(t+s) - N(s) = n\} = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \quad n = 0, 1, \ldots$$

References

[1] D.F. Anderson. Incorporating postleap checks in tau-leaping. *Journal of Chemical Physics*, 128(5):054103, 2008.

[2] A. Arkin, J. Ross, and H. McAdams. Stochastic kinetic analysis of development pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149:1633–1648, 1998.

[3] N.N. Batada, L.A. Shepp, and D.O. Siegmund. Stochastic model of protein-protein interaction: Why signaling proteins need to be colocalized. *Proceedings of the National Academy of Sciences*, 101(17):6445–6449, 2004.

[4] M.R. Bennett and J. Hasty. Microfluidic devices for measuring gene network dynamics in single cells. *Nature Reviews Genetics*, 10:628–638, 2009.

[5] W.J. Blake, M. Kærn, C.R. Cantor, and J.J. Collins. Noise in eukaryotic gene expression. *Nature*, 422:633–637, 2003.

[6] P.M. Burrage, R. Herdiana, and K. Burrage. Adaptive stepsize based on control theory for stochastic differential equations. *Journal of Computational and Applied Mathematics*, 170, 2004:317–336.

[7] Y. Cao and R. Erban. Stochastic Turing patterns: Analysis of compartment-based approaches. *Bulletin of Mathematical Biology*, 76(12):3051–3069, 2014.

[8] Y. Cao, D.T. Gillespie, and L. Petzold. Avoiding negative populations in explicit poisson tau-leaping. *Journal of Chemical Physics*, 123(5):054104, 2005.

[9] Y. Cao, D.T. Gillespie, and L. Petzold. Efficient step size selection for the tau-leaping simulation method. *Journal of Chemical Physics*, 124:044109, 2006.

[10] A. Chatterjee, D.G. Vlachos, and M.A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *Journal of Chemical Physics*, 122(24):024112, 2005.

[11] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[12] J. Elf and M. Ehrenberg. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Systems Biology*, 1(2):230–236, 2004.

[13] M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[14] R. Erban, S.J. Chapman, and P.K. Maini. A practical guide to stochastic simulations of reaction-diffusion processes. *arXiv:0704.1908v2*, [q-bio.SC], 2007.

[15] S.N. Ethier and T.G. Kurtz. *Markov Processes: Characterization and Convergence.* New York: Wiley, 1986.

[16] D. Fange and J. Elf. Noise-induced min phenotypes in *E. coli. PLoS Computational Biology*, 2(6):0637–0648, 2006.

[17] L. Ferm, A. Hellander, and P. Lötstedt. An adaptive algorithm for simulation of stochastic reaction-diffusion processes. *Journal of Compuational Physics*, 229:343–360, 2010.

[18] C.W. Gardiner, K.J. McNeil, D.F. Walls, and I.S. Matheson. Correlations in stochastic theories of chemical reactions. *Journal of Statistical Physics*, 14(4):307–331, 1976.

[19] M.A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

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

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

[22] D.T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A:404–425*, 188, 1992.

[23] D.T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[24] D.T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58:35–55, 2007.

[25] D.T. Gillespie, A. Hellander, and L.R. Petzold. Perspective: Stochastic algorithms for chemical kinetics. *Journal of Chemical Physics*, 138(17):170901, 2013.

[26] D.T. Gillespie and L.R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *Journal of Chemical Physics*, 119(16):8229–8234, 2003.

[27] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems (2nd Revised Edition)*, volume 1. Springer, Berlin, 2009.

[28] E.L. Haseltine and J.B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *Journal of Chemical Physics*, 117(15):6959–6969, 2002.

[29] J. Hasty, D. McMillen, F. Isaacs, and J.J. Collins. Computational studies of gene regulatory networks: *in numero* molecular biology. *Nature Reviews Genetics*, 2:268–279, 2001.

[30] M. Howard and A.D. Rutenberg. Pattern formation inside bacteria: fluctuations due to the low copy number of proteins. *Physical Review Letters*, 90(12):128102, 2003.

[31] T. Ideker, O. Ozier, B. Schwikowski, and A.F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(Suppl. 1):S233–S240, 2002.

[32] S. Ilie and J.M.A. Padgett. Adaptive time-stepping in the numerical solution of the reaction-diffusion master equation. *Book of Abstracts of the 2015 AMMCS-CAIMS Congress*, 393, 2015.

[33] S. Ilie and A. Teslya. An adaptive stepsize method for the chemical Langevin equation. *Journal of Chemical Physics*, 136(18):184101, 2012.

[34] S.A. Isaacson and C.S. Peskin. Incorporating diffusion in complex geometries into stochastic chemical kinetic simulations. *SIAM Journal of Scientific Computing*, 28(1):47–74, 2006.

[35] A. Johnson, B. Alberts, J. Lewis, K. Roberts, and P. Walter. *Molecular Biology of the Cell, 5th edition.* Garland Science, 2008.

[36] H. Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.

[37] W. Koh and K.T. Blackwell. An accelerated algorithm for discrete stochastic simulation of reaction-diffusion systems using gradient-based diffusion and tau-leaping. *Journal of Chemical Physics*, 134(15):154103, 2011.

[38] S. Lampoudi, D.T. Gillespie, and L.R. Petzold. The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *Journal of Chemical Physics*, 130(09):094104, 2009.

[39] F. Liu, M.A. Blätke, M. Heiner, and M. Yang. Modelling and simulating reaction-diffusion systems using coloured Petri nets. *Computers in Biology and Medicine*, 53:297–308, 2014.

[40] T.T. Marquez-Lago and K. Burrage. Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *Journal of Chemical Physics*, 127(10):104101, 2007.

[41] D.A. McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413–478, 1967.

[42] J.M.A. Padgett and S. Ilie. An adaptive tau-leaping method for stochastic simulations of reaction-diffusion systems. *submitted July 2015.*

[43] C.V. Rao and A.P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *Journal of Chemical Physics*, 118(11):4999-5010, 2003.

[44] S.M. Ross. *Introduction to Probability Models, 10th Edition.* Elsevier Inc., 2010.

[45] D. Rossinelli, B. Bayati, and P. Koumoutsakos. Accelerated stochastic and hybrid methods for spatial simulations of reaction-diffusion systems. *Chemical Physics Letters*, 451:136–140, 2008.

[46] H. Salis and Y.N. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *Journal of Chemical Physics*, 122(05):054103, 2005.

[47] H. Salis and Y.N. Kaznessis. An equation-free probabilistic steady-state approximation: Dynamic application to the stochastic simulation of biochemical reaction networks. *Journal of Chemical Physics*, 123(21):214106, 2005.

[48] A. Samant and D.G. Vlachos. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: A multiscale Monte Carlo algorithm. *Journal of Chemical Physics*, 123(14):144114, 2005.

[49] A.B. Stundzia and C.J. Lumsden. Stochastic simulation of coupled reaction-diffusion processes. *Journal of Computational Physics*, 127:196–207, 1996.

[50] R. Thar and M. Kühl. Bacteria are not too small for spatial sensing of chemical gradients: an experimental evidence. *Proceedings of the National Academy of Sciences*, 100(10):5748–5753, 2003.

[51] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *Journal of Chemical Physics*, 121(21):10356–10364, 2004.

[52] T.E. Turner, S. Schnell, and K. Burrage. Stochastic approaches for modeling in vivo reactions. *Computational Biology and Chemistry*, 28:165–178, 2004.

[53] J. Vilar, H. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Sciences*, 99(9):5988–5991, 2002.

[54] E. Weinan, D. Liu, and E. Vanden-Eijnden. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *Journal of Chemical Physics*, 123(19):194107, 2005.

[55] D.J. Wilkinson. *Stochastic Modeling for Systems Biology, Second Edition.* CRC Press, 2012.