Stochastic Modeling and Simulation of Multiscale Biochemical Systems

Minghan Chen

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Application

Young Cao, Chair
Layne T. Watson, Co-chair
John J. Tyson
Adrain Sandu
Hye Won Kang

May 9, 2019
Blacksburg, Virginia

Keywords: *Caulobacter* cell cycle model, hybrid stochastic simulation algorithm, stochastic parameter optimization

Stochastic Modeling and Simulation of Multiscale Biochemical Systems

Minghan Chen

ABSTRACT

Numerous challenges arise in modeling and simulation as biochemical networks are discovered with increasing complexities and unknown mechanisms. With the improvement in experimental techniques, biologists are able to quantify genes and proteins and their dynamics in a single cell, which calls for quantitative stochastic models for gene and protein networks at cellular levels that match well with the data and account for cellular noise.

This dissertation studies a stochastic spatiotemporal model of the *Caulobacter crescentus* cell cycle. A two-dimensional model based on a Turing mechanism is investigated to illustrate the bipolar localization of the protein PopZ. However, stochastic simulations are often impeded by expensive computational cost for large and complex biochemical networks. The hybrid stochastic simulation algorithm is a combination of differential equations for traditional deterministic models and Gillespie's algorithm (SSA) for stochastic models. The hybrid method can significantly improve the efficiency of stochastic simulations for biochemical networks with multiscale features, which contain both species populations and reaction rates with widely varying magnitude. The populations of some reactant species might be driven negative if they are involved in both deterministic and stochastic systems. This dissertation investigates the negativity problem of the hybrid method, proposes several remedies, and tests them with several models including a realistic biological system.

As a key factor that affects the quality of biological models, parameter estimation in stochastic models is challenging because the amount of empirical data must be large enough to obtain statistically valid parameter estimates. To optimize system parameters, a quasi-Newton algorithm for stochastic optimization (QNSTOP) was studied and applied to a stochastic

budding yeast cell cycle model by matching multivariate probability distributions between simulated results and empirical data. Furthermore, to reduce model complexity, this dissertation simplifies the fundamental cooperative binding mechanism by a stochastic Hill equation model with optimized system parameters. Considering that many parameter vectors generate similar system dynamics and results, this dissertation proposes a general α-β-γ rule to return an acceptable parameter region of the stochastic Hill equation based on QN-STOP. Different objective functions are explored targeting different features of the empirical data.

Stochastic Modeling and Simulation of Multiscale Biochemical Systems

Minghan Chen

GENERAL AUDIENCE ABSTRACT

Modeling and simulation of biochemical networks faces numerous challenges as biochemical networks are discovered with increased complexity and unknown mechanisms. With improvement in experimental techniques, biologists are able to quantify genes and proteins and their dynamics in a single cell, which calls for quantitative stochastic models, or numerical models based on probability distributions, for gene and protein networks at cellular levels that match well with the data and account for randomness.

This dissertation studies a stochastic model in space and time of a bacterium's life cycle—*Caulobacter*. A two-dimensional model based on a natural pattern mechanism is investigated to illustrate the changes in space and time of a key protein population. However, stochastic simulations are often complicated by the expensive computational cost for large and sophisticated biochemical networks. The hybrid stochastic simulation algorithm is a combination of traditional deterministic models, or analytical models with a single output for a given input, and stochastic models. The hybrid method can significantly improve the efficiency of stochastic simulations for biochemical networks that contain both species populations and reaction rates with widely varying magnitude. The populations of some species may become negative in the simulation under some circumstances. This dissertation investigates negative population estimates from the hybrid method, proposes several remedies, and tests them with several cases including a realistic biological system.

As a key factor that affects the quality of biological models, parameter estimation in stochastic models is challenging because the amount of observed data must be large enough to obtain valid results. To optimize system parameters, the quasi-Newton algorithm for stochastic

optimization (QNSTOP) was studied and applied to a stochastic (budding) yeast life cycle model by matching different distributions between simulated results and observed data. Furthermore, to reduce model complexity, this dissertation simplifies the fundamental molecular binding mechanism by the stochastic Hill equation model with optimized system parameters. Considering that many parameter vectors generate similar system dynamics and results, this dissertation proposes a general α-β-γ rule to return an acceptable parameter region of the stochastic Hill equation based on QNSTOP. Different optimization strategies are explored targeting different features of the observed data.

Dedication

This dissertation is dedicated to my beloved parents and siblings, my teachers, and my friends.

Acknowledgments

This dissertation would not have been possible without the guidance and support from all of those with whom I have had the pleasure to work during my PhD period. I am especially indebted to my supervisor, Dr. Young Cao, who has been supportive of my career goals and who has provided me a great deal about both scientific research and life in general. I am grateful to my co-advisor, Dr. Layne Watson, and my committee member, Dr. John Tyson, whose expertise was invaluable in exploring the methodology and research on quasi-Newton stochastic optimization algorithm and cell cycle models, respectively. I would like to thank Dr. Adrain Sandu and Dr. Hye Won Kang for their insightful suggestions and professional guidance on my research.

I would like to acknowledge my colleagues Mansooreh Ahmadian, Rachael Xu, Jing Cui, particularly, Fei Li and Shuo Wang, who have helped me a lot when I first joined in the lab. I also wish to express my heartful thanks to my friends, especially David and Aaron, who were of great support in deliberating over my problems and findings, as well as providing happy distraction to rest my mind outside of my research.

Nobody has been more important to me in the pursuit of the Ph.D. degree than the members of my family. I would like to thank my parents, whose love and support are with me in whatever I pursue. Most importantly, I wish to thank my sisters, Sandy and Emily, and my brother, Michael, for their sympathetic ear and unconditional help in life.

Contents

List of Figures

List of Tables

List of Abbreviations

CME Chemical Master Equation

DAE Differential Algebraic Equation

ODE Ordinary Differential Equation

PDE Partial Differential Equation

QNSTOP Quasi-Newton Stochastic Optimization Algorithm

RDMS Reaction Diffusion Master Equation

SDE Stochastic Differential Equation

SQSSA Stochastic Quasi-Steady-State Approximation

SSA Stochastic Simulation Algorithm

SSRFT Second Slow Reaction Firing Time

ssSSA Slow-Scale Stochastic Simulation Algorithm

Chapter 1

Overview

Traditional deterministic models, often represented as ordinary differential equations (ODEs) or differential-algebraic equations (DAEs), are widely used in biological systems to model and simulate average dynamics in cells. Yet different from average cell behavior, genetically identical cells show great variations in cell sizes, division times, and protein populations. Deterministic models fail to capture these stochastic features inside cells and cannot correctly represent cell dynamics in many situations [39, 107]. The Chemical Master Equation (CME) describes the time evolution of the probability of a system's states with a set of linear differential equations, thus governing the resulting stochastic process [51, 88]. The CME has no analytic solutions for all but the simplest systems and generally results in a linear system of very high or infinite dimension, which in the latter case is often unsolvable. Though there are several methods to approximate the CME solution [91, 135, 136, 137], the CME and its approximations face difficulty in handling large and complex systems where the dimension of system state is huge (n^N for n copies of N species). The stochastic simulation algorithm, often referred to as the Gillespie's algorithm or stochastic simulation algorithm (SSA) [49], is a major stochastic simulation method for "well-stirred" chemical reaction systems. As a Monte Carlo method, SSA numerically simulates each reaction firing over time to obtain a sample trajectory, which can be used to estimate the probability solution of the chemical master equation [53] or other system properties. Based on the same physicochemical assumptions as the CME, the SSA is considered accurate and numerically correct.

Realistic biochemical networks of a single cell usually have large discrepancies in the populations between mRNAs and proteins, in the rate constants between reactions and diffusions. As the SSA relies heavily on computation efficiency, much effort has been focused on improving the efficiency of the algorithm [16, 48, 87, 117]. However, the SSA is still prohibitively expensive for systems with fast reactions and large populations. To accelerate the SSA and utilize the multiscale feature, several approximation strategies have been proposed [17, 19, 55, 105]. In particular, Haseltine and Rawlings [55] proposed a hybrid method (hereafter referred as the HR hybrid method) that models fast reactions and large population species detereministically or as Langevin equations and the rest by the SSA. In the application of the HR hybrid method, differential equations are more commonly used to simulate the fast group instead of Langevin equations. The HR hybrid method approximates the chemical master equation well for a much greater region in system parameter space than the slow-scale SSA (ssSSA) and the stochastic quasi-steady-state assumption (SQSSA) methods [28]. The accuracy analysis for a linear chain reaction system showed that the HR hybrid method is accurate if the scale difference between fast and slow reactions is above a certain threshold, regardless of population scales.

However, populations of some reactant species might be driven negative if they are involved in both deterministic and stochastic subsystems [24]. This phenomenon is called the negativity problem, which often appears in stochastic simulation of reaction-diffusion systems particularly when low density species are distributed in a well-meshed space. Since diffusion is often modeled as continuous deterministic equations in the HR hybrid method for efficiency reasons, the average population inside each voxel or bin would be less than one if the total population of species is less than the number of voxels. Thus any consumption of those low population species in the stochastic domain will lead to a temporary negative population, and that needs special handling to avoid unrealistic simulation results.

Another important and difficult part of modeling is the estimation of system parameters, such as reaction rate constants in the model. Usually in a chemical reaction model a small portion of the parameters may have rough estimates derived from experiments, but most parameters are estimated by fitting model results to limited and error prone observations. For stochastic models, parameter estimation is even more challenging as the amount of empirical data must be large enough to obtain statistically valid parameter estimates. Many approaches have been developed to estimate parameters for stochastic biochemical systems [9, 71, 81, 89, 100, 106, 132, 134], One recent approach, the quasi-Newton stochastic optimization algorithm (QNSTOP) has been successfully applied to various stochastic optimization problems [27, 29, 102]. Belonging to the class of quasi-Newton methods for stochastic optimization problems, QNSTOP synthesizes ideas of stochastic approximation and response surface methodology, and the state-of-art numerical optimization methods (e.g., secant updates, trust regions). Originally developed by Castle [20], QNSTOP was later modified and adapted to (deterministic) global optimization problems. Amos et al. [5] combined the two versions into a single algorithm and computer code, QNSTOP, for deterministic and stochastic optimization problems.

With the development of new experimental techniques and more molecular data, biochemical networks expand quickly in both sizes and complexity. There is an increasing need for model reduction (using simple functions for complex dynamics) of many biochemical networks, in addition to the effort put in improving simulation efficiency. Take the cooperative binding as an example, which appears in a wide range of biochemical and physiological processes, such as multisite molecules [11, 61], transcription factors [1, 8, 70, 101], multimeric enzymes [21, 47], and drug-receptor relationships [54]. First introduced to formulate the curves of ligands binding to the enzyme or receptor, the Hill equation has been used to model the complex cooperative binding process for a long time. The equation is a nonlinear

function of the concentration of ligands and the Hill coefficient defines the degree of cooperativity of ligand binding. The Hill equation is simple and requires little prior knowledge of the binding mechanism, and thus is widely used in biological modeling. Using QNSTOP and the parameter estimation techniques we developed, in this thesis we also aim to evaluate the accuracy of the Hill function laws when they are applied to cooperative binding systems.

This dissertation presents four projects in the study on stochastic modeling and simulation of multiscale biochemical network, as shown in Fig. 1.1. Given any biological problem, the research methodology can be described in three steps: first establish the mathematical model, then set up the stochastic simulation, and last analyze simulated results. Chapter 3 illustrates the application of stochastic modeling and simulation to the *Caulobacter crescentus* cell cycle []. To improve the simulation efficiency, Chapter 4 introduces and analyzes the HR hybrid method and the corresponding negativity problem. To validate simulation results with empirical data, Chapter 5 investigates the parameter optimization of a budding yeast cell cycle model. Meanwhile, to reduce model complexity, Chapter 6 studies the stochastic Hill equation for fundamental cooperative binding process.

The *Caulobacter crescentus* cell cycle is representative of asymmetric division in prokaryotes. As an interesting biological organism that exhibits spatial patterns in a single cell, *Caulobacter crescentus* has become an important model organism for fundamental research on cell cycle regulation, differentiation, and asymmetric cell division. Unlike the budding yeast cells that have been studied extensively [4, 23], recent examination of the *Caulobacter* cell cycle has revealed many unknown mechanisms and necessitates a wider investigation. Experiments reveal that the cell cycle progression and cell differentiation is controlled by many elaborate molecular mechanisms that regulate the production, interaction, and localization of a host of proteins [73, 75]. At cell division, the landmark protein PopZ is located at the old ends of the newborn cells, and later in the cell cycle PopZ adopts a bipolar pat-

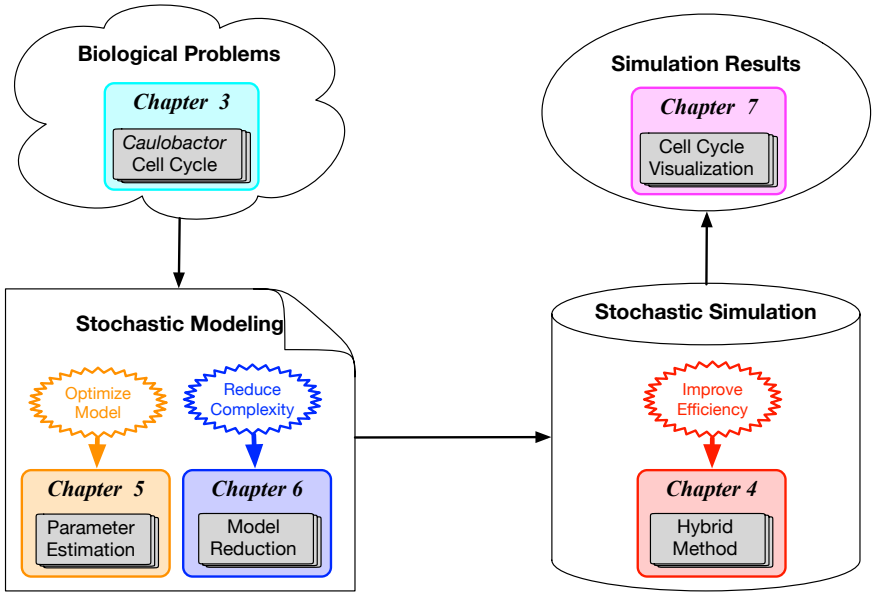


Figure 1.1: Stochastic modeling and simulation of multiscale biochemical network.

tern in the predivisional cell. The polar localization of PopZ plays a determining role in the intracellular location of certain key cell cycle regulators and in tethering the replicated chromosome to the two ends of the cell [25]. To study the mechanism of PopZ bipolarity, Chapter 3 proposes a model of spatiotemporal organization in two spatial dimensions, based on a Turing mechanism of pattern formation in coordination with chromosome replication and segregation. PopZ patterns are explored for domains of different shapes and different locations of *popZ* genes. Both deterministic and stochastic simulations capture the observed variations in the location and timing of PopZ polymerization.

Chapter 4 studies the negativity problem of the HR hybrid method, analyzes and tests with several models including a linear chain system, a nonlinear reaction system, and a realistic biological cell cycle system. As a benchmark, the second slow reaction firing time is used to measure the effect of negative populations on the accuracy of the HR hybrid method. The analysis demonstrates that usually the error caused by negative populations is negligible compared with approximation errors of the HR hybrid method itself, and sometimes nega-

tivity phenomena may even improve the accuracy. For systems where negative population species are involved in nonlinear reactions or some species are highly sensitive to negative populations, the system stability will be influenced and may lead to system failure when using the HR hybrid method. In those circumstances, three remedies are studied for the negativity problem.

Chapter 5 investigates the quasi-Newton algorithm for parameter optimization in a stochastic model of the budding yeast cell cycle. The cell cycle model from Oguz et al. [95] contains 52 stochastic parameters. The 'best' stochastic parameters in [95] were found by comparing simulated values and observed values for the means and variances of certain cell-cycle observables. Instead matching summary statistics, QNSTOP is used here to directly match empirical and simulated joint probability distributions of the pair (mass at birth, duration of of G1 phase) from the empirical and simulated cell colonies for both mother and daughter budding yeast cells. Results and predictions for the budding yeast model match well some summary statistics and one-dimensional distributions from empirical data, but do not match well the empirical joint distributions. The nature of the mismatch provides insight into the weakness in the stochastic model.

Chapter 6 simplifies the fundamental cooperative binding mechanism by a stochastic Hill equation model with optimized system parameter. However, traditional parameter optimization methods have been focusing on finding the best system parameter value, while in most circumstances there are many parameter values and numerous combinations of multi-dimensional parameters that generate similar system dynamics and results. Chapter 6 proposes a general α-β-γ rule to return an acceptable parameter region for the stochastic Hill equation based on a quasi-Newton stochastic optimization (QNSTOP) algorithm. Different objective functions were investigated targeting different features of the empirical data, among which the approximate maximum log-likelihood method is recommended for gen-

eral use. Numerical results demonstrate that if fed with appropriate parameter values, the stochastic Hill equation model depicts the basic cooperative binding process well except the initial stage.

Chapter 2

Background

2.1 Chemical Master Equation

Consider a well-mixed system of N distinct species and M reaction channels with $\hat{N}$ possible states. The chemical master equation [51] that describes the probabilistic time evolution of the system dynamics is

$$\frac{\partial}{\partial t}P(X;t) = P(X;t)A, \tag{2.1}$$

where $X = [x_1, x_2, \ldots, x_{\hat{N}}]$ is all possible state vectors x_i at any time t, and $P(X;t)$ represents the probabilities of those state vectors at time t, and A is the state reaction matrix [91], given by

$$A_{ij} = \begin{cases} -\sum_{\mu=1}^{M} a_\mu(x_j), & \text{for } i = j, \\ a_\mu(x_i), & \text{for } i \text{ such that } x_j = x_i + v_\mu, \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

where v_μ is the stoichiometric transition vector for reaction channel μ.

From equation (2.1), the solution is

$$P = P_0 e^{At}, \tag{2.3}$$

where P_0 is the initial probability of all the possible state, and the transition probability matrix can be calculated by

$$\mathcal{T} = e^{A\tau}, \tag{2.4}$$

where τ is the period of time the system has evolved from a previous time [91].

2.2 Stochastic Simulation Algorithms

Consider a well-stirred system of N chemical species $\{S_1,S_2,\ldots,S_N\}$ and M chemical reactions $\{R_1,R_2,\ldots,R_M\}$ within a constant volume. The instantaneous state of the chemical system at time t is denoted by a vector $\vec{x}(t) = (x_1(t),x_2(t),\ldots,x_N(t))^T$, where $x_j(t)$ is the number of S_j molecules at time t. The propensity function for reaction R_j is defined as $a_j(\vec{x}(t))$, where

$$a_j(\vec{x}(t))dt = \mathrm{P}\left[R_j \text{ fires within the time interval } [t, t+dt)\right].$$

The stoichiometric matrix of this system is defined as $V = [\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_M]$, whose elements $\vec{v}_j = (v_{j1},v_{j2},\ldots,v_{jN})^T$ for $j = 1,2,\ldots,M$ are state-change vectors. If reaction R_j fires, the system state $\vec{x}$ is changed by $\vec{v}_j$. The SSA procedure is presented as follows [49, 50]:

(1) At time t, compute propensities $a_j(\vec{x}(t))$ for $j = 1,2,\ldots,M$, and the summation

$$a_0(\vec{x}(t)) = \sum_{j=1}^{M} a_j(\vec{x}(t)).$$

(2) Generate two random numbers u_1 and u_2 which satisfy the uniform distribution $\mathrm{U}(0,1)$.

(3) Calculate a time increment τ according to the equation

$$\tau = -\frac{\ln(u_1)}{a_0(\vec{x}(t))}.$$

(4) Select a reaction with index μ which satisfies

$$\sum_{j=1}^{\mu} a_j(\vec{x}(t)) \leq u_2 a_0(\vec{x}(t)) < \sum_{j=1}^{\mu+1} a_j(\vec{x}(t)).$$

(5) Update the system time $t = t + \tau$ and the system state $\vec{x}(t) = \vec{x}(t - \tau) + \vec{v}_\mu$.

(6) Return to step (1) if stopping condition is not reached.

Although the SSA is a fundamental method for stochastic simulation, its computational cost may become very high when the size of a biochemical system increases. Since the SSA simulates every reaction event and the time increment τ is often very small due to the large total propensity $a_0(\vec{x})$, there is a high demand for efficient simulation methods.

2.3 Hybrid Stochastic Simulation Algorithm

2.3.1 HR hybrid method

The hybrid stochastic simulation algorithm studied in this work was first proposed by Haseltine and Rawlings (HR) [55]. Given a system of N species $\{\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_N\}$ and M reactions $\{\mathrm{R}_1, \mathrm{R}_2, \ldots, \mathrm{R}_M\}$, these M reactions are partitioned into two subsets: the fast reaction group G_f and the slow reaction group G_s. The dynamics of species in G_f are formulated by ODEs, and the reactions in G_s are simulated by the SSA. Let $\vec{x}(t) = (x_1, x_2, \ldots, x_N)^T$ be the system state at time t. $a_j(\vec{x}(t))$ for $j = 1, 2, \ldots, M$ are propensity functions. Define the state-change vector $\vec{v}_j = (v_{j1}, v_{j2}, \ldots, v_{jN})^T$ for reaction j. If reaction j fires, the system state is changed by $\vec{v}_j$. Let τ be the jump interval of the next slow (stochastic) reaction, and μ be its reaction index. Set $t = 0$. The hybrid method simulates the system as follows [55, 129]:

(1) Generate two random numbers u_1 and u_2 satisfying the uniform distribution $\mathrm{U}(0,1)$.

(2) Numerically integrate the ODE system and solve the integral equation

$$\int_t^{t+\tau} a_{tot}(\vec{x}(s))ds + \ln(u_1) = 0, \tag{2.5}$$

where $a_{tot}(\vec{x}(s))$ is the summation of propensities of reactions in G_s at time s and $t+\tau$ is the time when a slow reaction will fire.

(3) Stop the numerical integration and update the system time $t = t + \tau$.

(4) Select a slow reaction with index μ according to the inequalities

$$\sum_{j=1}^{\mu} a_j(\vec{x}(t)) \leq u_2 a_{tot}(\vec{x}(t)) < \sum_{j=1}^{\mu+1} a_j(\vec{x}(t)),$$

where $a_j(\vec{x}(t))$ is the propensity of reaction j in G_s.

(5) Update the system state $\vec{x}(t) = \vec{x}(t) + \vec{v}_\mu$.

(6) Return to step (1) if the stopping condition is not reached.

Our implementation is slightly different in step (2). Suppose that the ODE system is given by

$$\frac{d\vec{x}(s)}{ds} = f(\vec{x}(s)). \tag{2.6}$$

We simply add a variable z and an equation

$$\frac{dz}{ds} = a_{tot}(\vec{x}(s)), \text{ with initial value at } t\text{: } z(t) = \ln(u_1), \tag{2.7}$$

where $\ln(u_1)$ is negative and a_{tot} is nonnegative. During the simulation, each step starts at *current time* t and numerically integrates ODEs (2.6) and (2.7). When $z(t+\tau) = 0$, the ODE

integration stops. Then τ is the solution to Eq. (2.5) and the system time automatically proceeds to $t = t + \tau$. This integration process can be conveniently handled by standard ODE solvers combined with root-finding algorithms. Note that since z is an integration variable, one may easily choose to omit it from the error control mechanism. Adding this extra variable does not significantly affect the efficiency of ODE solvers [98].

2.3.2 Stochastic quasi-steady-state approximation

Quasi-steady-state assumption (QSSA) is an approximation method to reduce model complexity for differential equations in the deterministic regime. As an effective way to reduce expensive cost of fast dynamics, Rao and Arkin [105] extended the QSSA to stochastic simulation and proposed the stochastic quasi-steady-state approximation (SQSSA). Targeting transitory intermediate species, the SQSSA is suitable for biochemical networks containing chain-style reactions, e.g., enzyme-substrate models. For example, Kim and Sontag [67] and Kang et al. [65] derived the stochastic model of enzyme kinetics based on the QSSA and showed that the conditions for the stochastic QSSA are mostly consistent with the deterministic QSSA.

The SQSSA algorithm first separates system species into intermediate species and primary species. Reactions are correspondingly partitioned into two groups based on the two groups of species. Then the steady states for the intermediate species group are calculated, while the SSA is used for the primary species group. The SQSSA procedure is similar to the SSA except for the following steps [105]:

(0) Before step (1), at time t, first generate the steady state of the intermediate species.

(1) Based on the steady state, compute propensities $a_{tot}(\vec{x}(t))$ for reactions associated with the primary species, instead of $a_0(\vec{x}(t))$.

Note that $a_0(\vec{x}(t))$ is replaced with $a_{tot}(\vec{x}(t))$ through the whole simulation. Thus the SQSSA avoids generating exact realizations of those highly reactive intermediate species that contribute most to computational cost.

2.3.3 Slow-scale stochastic simulation algorithm

The slow-scale stochastic simulation algorithm (ssSSA) [19] was proposed to improve simulation efficiency for multiscale systems. Based on an observation that fast reactions are expensive in computation but less important for system dynamics change, the ssSSA assumes partial equilibrium for those fast reactions. The algorithm partitions the reactions into fast and slow subsets. For chain reaction systems, the partition strategy of the ssSSA is similar to that of the HR hybrid method, which puts fast reactions and fast species together, denoted as G_f, and the rest into the slow scale reaction group, G_s. But the ssSSA simulates the steady state of fast subsystem and uses the SSA to model the slow subsystem. In implementation, the ssSSA shares the same steps (2)-(6) with the SSA. The first two steps are as follows:

(0) At time t, first compute the steady state of G_f.

(1) Compute propensities $a_{tot}(\vec{x}(t))$ for reactions in G_s, instead of $a_0(\vec{x}(t))$.

Generally speaking, partitioning in the SQSSA is based on species and partitioning in the ssSSA is based on reactions. If we assume all species involved in fast reactions are in steady-state, then the ssSSA and the SQSSA are very similar. On the other hand, the ssSSA is easier to implement, whereas the SQSSA may be applied to more applications.

2.4 Quasi-Newton Stochastic Optimization Algorithm

QNSTOP is a class of quasi-Newton methods developed for stochastic optimization [20], where the objective function $f(X)$ is a random variable, and X is contained in a box $L \leqq X \leqq U$. Next, we summarize the essential steps of QNSTOP here. Further details on the algorithm and implementation can be found in Ref. [5].

- In each iteration k, construct a quadratic model

$$\widehat{m}_k(X - X_k) = \hat{f}_k + \hat{g}_k^T (X - X_k) + \frac{1}{2} (X - X_k)^T \hat{H}_k (X - X_k)$$

  centered at X_k, where $\hat{g}_k$ is the gradient vector and $\hat{H}_k$ is the Hessian matrix. Note that $\hat{f}_k$ is generally not $f(X_k)$, which is stochastic.

- QNSTOP uses an ellipsoidal region centered at the current iterate $X_k \in \mathbb{R}^n$ with radius τ_k,

$$E_k(\tau_k) = \left\{ X \in \mathbb{R}^n : (X - X_k)^T W_k (X - X_k) \leq \tau_k^2 \right\},$$

  where W_k is a symmetric, positive definite scaling matrix, satisfying $W_k \in W_\gamma$,

$$W_\gamma = \left\{ W \in \mathbb{R}^{n \times n} : W = W^T, \det(W) = 1, \gamma^{-1} I_n \preceq W \preceq \gamma I_n \right\}$$

  for some $\gamma \geq 1$, where I_n is the $n \times n$ identity matrix, and $A \preceq B$ means $B - A$ is positive semidefinite. The elements of the set W_γ are valid scaling matrices that control the shape of the ellipsoidal design regions with eccentricity constrained by γ.

- Then QNSTOP estimates the gradient based on a set of N uniformly sampled design sites $\{X_{k1}, \ldots, X_{kN}\} \subset E_k(\tau_k) \cap \Theta$ ($\Theta = [L, U]$, which is the feasible set of parameters defined initially.) For the Hessian matrix, QNSTOP uses either a variation of the SR1

(symmetric, rank one) quasi-Newton update (stochastic f) or the unconstrained BFGS quasi-Newton update (global optimization of deterministic f).

- For the next iteration, by utilizing an ellipsoidal trust region concentric with the design region for controlling step length, X_{k+1} is updated as

$$X_{k+1} = \left(X_k - \left[\hat{H}_k + \mu_k W_k \right]^{-1} \hat{g}_k \right)_\Theta ,$$

where μ_k is the Lagrange multiplier of a trust region subproblem, and $(\cdot)_\Theta$ denotes projection onto the feasible set $\Theta = [L, U]$.

- Finally, the experimental design region $E_k(\tau_k)$ is updated to approximate a confidence set by updating the scaling matrix W_k. The updated scaling matrix is given by

$$W_{k+1} = \left(\hat{H}_k + \mu_k W_k \right)^T V_k^{-1} \left(\hat{H}_k + \mu_k W_k \right),$$

where V_k is the covariance matrix of $\nabla \widehat{m}_k(X_{k+1} - X_k)$. For numerical stability, W_{k+1} is constrained (by modifying its eigenvalues) to satisfy the constraints $\gamma^{-1} I_n \preceq W_{k+1} \preceq \gamma I_n$ and $\det(W_{k+1}) = 1$, so $W_\gamma \ni W_{k+1}$.

Chapter 3

Two-dimensional Model of Bipolar PopZ Polymerization in *Caulobacter crescentus*

3.1 Introduction

Caulobacter, an oligotrophic bacterium that lives in aquatic environments, divides asymmetrically into two different types of daughter cells. One daughter is a 'stalked' cell with a tubular stalk structure anchoring the cell to a substratum. The other is a mobile 'swarmer' cell with a flagellum allowing it to swim away from the place of its birth. These two different morphological and functional forms enable *Caulobacter* populations to thrive in nutrient-poor habitats, by limiting competition between stalked cells and swarmer cells and by allowing swarmer cells to find new, nutrient rich environments [99]. After sensing nutritional cues, the swarmer cell will differentiate into a stalked cell by ejecting its flagellum, retracting its pili, and growing a stalk at the original flagellar pole [120].

The dimorphic cell cycle of *Caulobacter* is presented in Fig. 3.1. The swarmer stage is a G1 (pre-replicative) phase of the cell cycle. At the swarmer-to-stalked transition, the cell enters S phase (DNA synthesis). The stalked cell then goes through a morphological transition:

one half of the cell retains the stalk structure, while the other half develops a flagellum at the 'new' pole of the cell. This bipolar morphology is called the predivisional stage of the cell cycle.

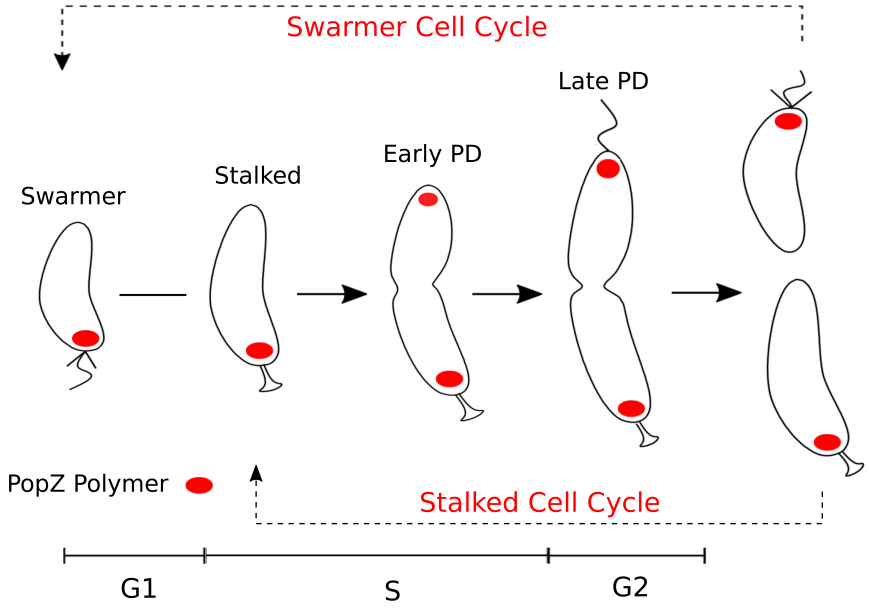


Figure 3.1: *Caulobacter* cell cycle.

As an interesting biological organism that exhibits spatial pattern in a single cell, *Caulobacter crescentus* has become an important model organism for fundamental research on cell cycle regulation, differentiation and asymmetric cell division. How *Caulobacter* regulates the asymmetric localization of proteins may provide insights on other similar biological systems. Experiments on *Caulobacter* reveal the elaborate details of protein localization in space and time [31]. These spatiotemporal changes govern cell shape [22, 68], chromosome segregation [13, 58] and cell differentiation [73]. In *Caulobacter* cells, the protein PopZ was identified as a potential landmark protein [14]. PopZ locates at the old pole of the swarmer cell and begins to accumulate at the new pole when chromosome replication and segregation is initiated in the stalked cell. Hence, predivisional cells have a peak of PopZ polymerization at each end [38]. While the dynamic localization pattern of PopZ is clearly observed, the mechanism

behind this pattern is still under debate.

Experiments indicate that PopZ forms polymer structures and that overexpression of PopZ can lead to cell division defects [14, 38]. PopZ mRNA drives the synthesis of PopZ monomers, which then form polymers at specific locations in the cell, determined in part by the location of pre-existing PopZ polymers. PopZ polymers can also depolymerize to monomers. The kinetics of PopZ polymerization along with diffusion of PopZ monomers underlie the self-organizing structures of PopZ polarization. The biochemistry of PopZ polymerization is similar to an Activator/Substrate-Depletion (A/SD) mechanism of Turing pattern formation [124]. For example, small *Caulobacter* cells are observed to have only one focus of PopZ while long cells have two or more foci [38]. This property to form peaks of PopZ polymerization separated by a characteristic distance is consistent with a Turing pattern. The fact that PopZ monomers diffuse much faster than polymers also satisfies the condition on the ratio of diffusion constants for the activator and substrate of an A/SD Turing mechanism. Based on these considerations, we develop a two-dimensional (2D) Turing-type model of PopZ polymerization, coupled with the segregation of replicated chromosomes, in order to study the hypothesis that a Turing pattern-formation mechanism may be responsible for the observed bipolar distribution of PopZ in *Caulobacter* cells.

Turing patterns depend on several factors, such as initial conditions, reaction terms and domain geometry [92]. The sensitivity of patterns to initial conditions varies in different systems, and experiments show that initial conditions influence the phase of patterns [92]. In general, a 2D Turing system forms a pattern of spots when extended in both x and y directions. When extension in the y direction is limited, a striped pattern will form, just like the corresponding one dimensional (1D) model [120]. Certain conditions on the reaction terms must also be satisfied for a system to generate Turing patterns [92]. We will demonstrate that our model, based on Turing patterns and chromosome dynamics,

can explain experimental observations and make predictions concerning the self-organized, bipolar distributions of landmark proteins in bacterial cells.

This chapter is organized as follows: in Section 3.2 we give a literature review; in Section 3.3 we present the mathematical model; in Section 3.4 we present the results of our main model and discuss different variations of the model under different settings; and in Section 3.5 we draw some conclusions about our 2D model.

3.2 Literature Review

Protein localization plays a significant role in many cell events and at all levels of biological organization. Besides *Caulobacter*, bacteria such as *Escherichia coli* and *Bacillus subtilis* exhibit dynamic localization of specific proteins during their cell division cycles [122, 133]. Modern methods, like fluorescence microscopy, have accelerated the identification and quantification of protein localization [118, 131]. The flood of new data requires more rigorous models of cell cycle regulatory networks to explain the self-assembly mechanisms of protein localization.

Several hypotheses have been proposed to explain the initial unipolar localization and later bipolar localization of landmark proteins [72]. One hypothesis is that membrane curvature at the poles of cells results in bipolar distribution of proteins [62, 104]. Another explanation posits the existence of scaffolds at the ends of a cell. Whenever proteins are close to the ends, they will be ensnared by the scaffold and kept in that region [110]. However, these explanations have problems of their own. For instance, "why are proteins observed to form peaks at poles even without curvature cues" and "what causes scaffolds to accumulate at the poles of a cell in the first place"?

A better explanation is Turing's mechanism of self-assembling pattern formation, which can explain many natural patterns in biology such as leopard spots, fish strips, spaced rows of alligator teeth, and so forth. As first proposed by Turing [124], patterns such as spots form as a result of the interplay of chemical reactions and diffusion which finally attain a steady state with heterogeneous spatial patterns of chemical concentration under certain conditions. There are two basic mechanisms that can exhibit Turing pattern: Activator/Inhibitor-Production (A/IP) and Activator/Substrate-Depletion (A/SD). For two species A/SD systems, one species must be self-activating and slowly diffusing, and the other fast-diffusing substrate is depleted during the production of autocatalytic activator [116].

Turing's mechanism can only form stripes in 1D systems, while in 2D situations, possible patterns vary from stripes, spots, squares, rhombi and hexagons. Generally the form of the interaction kinetics plays a major role in what pattern will be obtained. Cubic interactions tend to favor stripes while quadratic interactions tend to produce spots [92]. Analysis near bifurcation, referred to as weakly nonlinear stability analysis, was used to study the conditions on parameters for the steady state spatially heterogeneous solutions of these patterns [42, 93] and the corresponding spatial characteristics such as wavelength [138]. Meanwhile, the effect of growth is evident in the production of spatial heterogeneity. Some systems fail to generate stable pattern sequences within a growing domain. Yet, a frequency-doubling sequence of patterns can be robustly realized under uniform exponential growth [32], which is the growth law used in our model. Other growth laws, like linear or logistic growth, will eventually break down resulting from small system perturbations [86].

3.3 Mathematical model

3.3.1 A basic reaction-diffusion model

Our model includes a species called *popZ* gene that is confined to one discrete location in the cell. We start our model with the following biochemical reactions:

$$\emptyset \xrightarrow{gene} mRNA \longrightarrow \emptyset, \tag{3.1}$$

$$\emptyset \xrightarrow{mRNA} Monomer \longrightarrow \emptyset, \tag{3.2}$$

$$Monomer \rightleftharpoons Polymer \longrightarrow \emptyset, \tag{3.3}$$

$$Monomer + 2Polymers \longrightarrow 3Polymers. \tag{3.4}$$

There are three reacting-diffusing species in mechanism (1-4): *popZ* mRNA, PopZ monomer and PopZ polymer. The symbol $\emptyset$ in the above equations represents the synthesis or degradation of species. The synthesis of mRNA is driven by the *popZ* gene. Initially, there is only one gene, located at the old end of a cell. At about 50 min into the cell cycle, the *popZ* gene is replicated and the new copy is rapidly translocated to a position close to the new end of the cell.[1] We assume that the two genes are fixed in these places for simplification. Consequently, *popZ* mRNA is produced at the location of the *popZ* gene(s) (reaction (1)), and the mRNA molecules can subsequently diffuse throughout the cell. The production of PopZ monomers is dictated by the position of *popZ* mRNA (reaction (2)). Monomers can be incorporated into polymers either spontaneously (reaction (3)) or with the help of two polymers (reaction (4)). Polymers can depolymerize, and both monomers and polymers can

[1] Our simulation results (not shown) illustrated that omitting the fast drifting process of the duplicated gene does not affect the final pattern.

be degraded. Monomeric and polymeric PopZ diffuse with much different rates. Obviously, the *popZ* gene(s) will affect both the abundance and localization of PopZ protein. In order to study the influence of *popZ* gene replication and segregation, we will analyze simulation results with the genes located at different positions in the cell.

During cell growth, the width of a cell denoted by W, remains fixed, while cell length, denoted by $L(t)$, increases with time. The growth rate $\mu = dL/dt > 0$. Each point in the 2D domain can be represented by a pair of coordinates (x, y), where $0 \leq x \leq L(t)$ and $0 \leq y \leq W$. With appropriate diffusion constants for PopZ monomers and polymers, the corresponding 1D model [120] forms peaks of PopZ polymerization at both ends of the 1D cell, matching the observed patterns in *Caulobacter* cells. However, the 1D model is based on the assumption that proteins are uniformly distributed in the y direction, which may not be true. To make the model more realistic, we extend the model to two spatial dimensions and study the 2D patterns of PopZ localization.

In general, a 2D reaction-diffusion system for one species can be formulated as a partial differential equation

$$\frac{\partial u}{\partial t} = f(u) + D\Delta u, \tag{3.5}$$

where $u(t, x, y)$ is the concentration of the species at location (x, y) at time t, $f(u)$ is the reaction terms for this species (may involve other species), D is the diffusion rate for this species, and Δ is the Laplace operator. In 2D, $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$. This model can also be generalized to a corresponding compartment-based stochastic model [41] with appropriate discretization of space.

To solve equation (3.5), we need to specify initial and boundary conditions. Assuming molecules can neither enter nor leave the cell, we implement zero-flux boundary conditions,

defined by

$$(\mathbf{n}\cdot\nabla)u = 0, := \mathbf{r} := on := \partial B;$$

where ∂B is the closed boundary of system domain B and $\mathbf{n}$ is the unit outward normal to ∂B. The initial condition $u(0,\mathbf{r})$ will be given according to different scenarios.

For our PopZ model, the PDEs are thus given by

$$\begin{aligned}
\frac{\partial p}{\partial t} &= -k_{dp}p - k_{dpl}p + k_{dnv}m + k_a p^2 m + d_p\Delta p,\\
\frac{\partial m}{\partial t} &= k_{sm}r - k_{dm}m + k_{dpl}p - k_{dnv}m - k_a p^2 m + d_m\Delta m,\\
\frac{\partial r}{\partial t} &= k_{sr} - k_{dr}r + d_r\Delta r,
\end{aligned} \tag{3.6}$$

where p denotes the concentration of PopZ polymer, m denotes the concentration of PopZ monomer, and r denotes the concentration of mRNA. The definitions and values of the parameters in the model are listed in Table 3.1. We use the same parameter values that can generate bipolarity in 1D model [120]. In the table, k_{dnv} is the synthesis rate of polymer in reaction (3), and k_a is the synthesis rate in reaction (4).

Table 3.1: Parameters of the PopZ model.

Name	Value(min^{-1})	Description
k_{sr}	0.5	Gene synthesis rate of mRNA
k_{dr}	0.2	mRNA degradation rate
k_{sm}	26	mRNA synthesis rate of PopZ
k_{dm}	0.05	Monomer degradation rate
k_{dp}	0.05	Polymer degradation rate
k_{dpl}	0.1	Polymer depolymerization rate
k_{dnv}	12	Synthesis rate of polymer
k_a	12	Synthesis rate of polymer
d_r	$0.05\mu\text{m}^2$	mRNA diffusion rate
d_m	$10\mu\text{m}^2$	Monomer diffusion rate
d_p	$0.001\mu\text{m}^2$	Polymer diffusion rate

Note that equations (3.6) imply that mRNA can be produced anywhere in the cell. This

assumption is fine if we do not consider gene location. However, later we will show that gene location plays an important role in the final pattern of PopZ bipolarity. Thus we need to consider gene location in our model. To conveniently model that, we will first discretize the 2D domain.

3.3.2 Domain discretization and gene location

Discretization of a spatial domain is an important step in transferring continuous equations into discrete compartments. In the deterministic regime, with a finely spaced lattice the PDEs (3.6) can be converted into a set of ODEs using central differencing (the method of lines). In the stochastic regime, the lattice allows us to define system variables as populations of different species in each grid element and to formulate the system as a discrete stochastic model.

The domain discretization size directly affects the accuracy of the numerical solution of PDEs. It is well known that, in the deterministic case, sufficiently small compartment size can minimize numerical errors and thus deliver accurate simulations of the spatial variation of proteins. However, this is not true in the stochastic domain. Unlike the deterministic simulation where a finer mesh will give a more accurate result, the stochastic model (simulated by Gillespie's stochastic simulation algorithm (SSA)) may lead to large numerical errors if the compartment size is too small. This situation becomes more delicate when the system contains second-order (bimolecular) reactions or higher order reactions in the stochastic model [40]. Thus, discretization may lead to incorrect results when it is not handled with care.

In our system, there is one trimolecular reaction term which can affect the performance of stochastic simulations. Since the width of the cell is about half of its length, we start with

n grid points in the y direction and $2n$ grid points in the x direction. We tried different discretization sizes, from 10×20 (10 grids in y direction, 20 grids in x direction), to 20×40, and gradually increased to 80×160. We chose 50×100 as an appropriate discretization size for our model as it is delicate enough to catch species distribution within acceptable simulation time cost. As the cell grows, the length of each compartment increases while the width remains the same, and the total number of compartments remains the same. That is, we assume that new cell wall material is added uniformly along the x axis. Suppose each compartment grows exponentially with time:

$$\frac{dl}{dt} = \mu l; \quad \frac{dw}{dt} = 0.$$

where l is the length of each compartment (changing with time) and w is the width of each compartment (constant in time). In this case, the diffusion part in the system is described by

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} = \dfrac{u_{i-1,j} + u_{i+1,j} - 2u_{i,j}}{l^2}, \ 1 < i < 100, \\ \dfrac{\partial^2 u}{\partial y^2} = \dfrac{u_{i,j-1} + u_{i,j+1} - 2u_{i,j}}{w^2}, \ 1 < j < 50, \end{cases}$$

where (i, j) refers to the location of bin (i in length and j in width), $u_{i,j}$ is the concentration of species u at bin (i, j). Other diffusion expressions will be explained later with different domains. At the domain boundary,

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} = \dfrac{u_{2,j} - u_{1,j}}{l^2}, \ i = 1, \\ \dfrac{\partial^2 u}{\partial x^2} = \dfrac{u_{99,j} - u_{100,j}}{l^2}, \ i = 100, \\ \dfrac{\partial^2 u}{\partial y^2} = \dfrac{u_{i,2} - u_{i,1}}{w^2}, \ j = 1, \\ \dfrac{\partial^2 u}{\partial y^2} = \dfrac{u_{i,49} - u_{i,50}}{w^2}, \ j = 50, \end{cases}$$

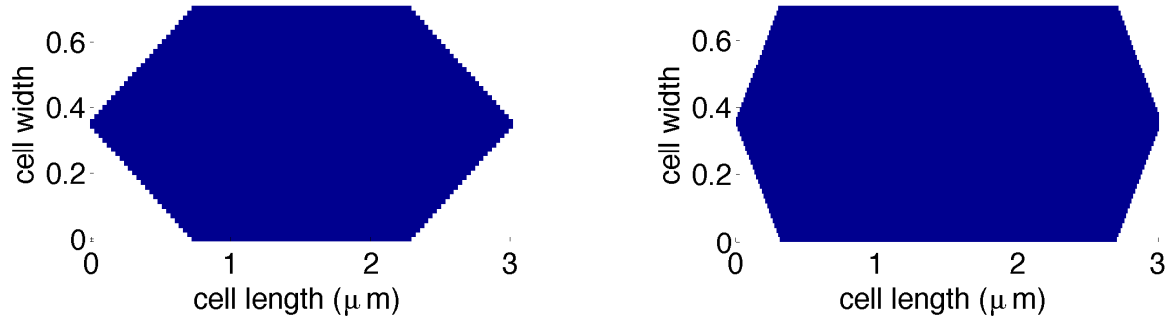
After discretizing the cell in this way, we assume each *popZ* gene is located within a specific compartment regardless of compartment growth. We assume this compartment is located in the center of the y axis, and the compartment of the second gene is placed symmetrically to the compartment of the first gene along the x axis. mRNA can only be produced in the compartment where a gene is located but can freely diffuse throughout the cell. The propensity function for mRNA synthesis will be a constant in the compartment where a *popZ* gene is located, rather than increasing linearly with compartment size.

3.3.3 Domain shape

Another factor that affects Turing pattern formation is domain shape. Hence, we may expect that the shape of a bacterial cell will play a significant role in protein polarization. The cells of *Caulobacter crescentus* are crescent shaped. In our study, we explored three different simplifications of the crescent shape: a rectangle, and a rectangle with two different types of triangular ends, as shown in Fig. 3.2. The two types of triangle end domains are: **A)** the length l of every compartment grows exponentially with time; **B)** only compartments in the middle rectangular part grow lengthwise with time, so the triangular end pieces do not change shape as the cell grows.

The triangular end shapes in Fig. 3.2 have a common slope of $2w/l$ (i.e., two compartments in the width direction for each compartment in the length direction). For type A triangles, the slope changes as the cell grows, because l increases in the triangular regions. For type B triangles, the slope is constant.

To match with experimental data, the total cell length at birth is set as 1.3 μm, and the cell grows to a total length of 3 μm at the end of cell cycle. Cell width remains 0.7 μm during the whole process. In type B cells, the diffusion rates between compartments must



(a) Type A triangle end (b) Type B triangle end

Figure 3.2: Two different cell shapes at the end of the cell cycle, when the cell is 3 μm in length.

be handled carefully because compartments in the middle rectangular region are increasing in size, while compartments in the end triangular regions maintain a constant size. At the boundary between triangular and rectangular regions, the diffusion flux is proportional to the inverse of the distance between centers of neighboring compartments. If l is the compartment size in the rectangular region and l_0 is the compartment size in the triangular region, the rate of one molecule jumping from rectangular region to triangular region will be given by $\frac{D}{h^2}$, where $h = \frac{l+l_0}{2}$.

Take species u in the compartment i as an example (Fig. 3.3), the Laplacian operator can be written as

$$\frac{\partial^2 u}{\partial x^2} = \frac{4u_{i-1,j}}{(l+l_0)^2} + \frac{u_{i+1,j}}{l^2} - \frac{4u_{i,j}}{(l+l_0)^2} - \frac{u_{i,j}}{l^2}.$$

where the four terms correspond to the sequential arrows in Fig. 3.3.

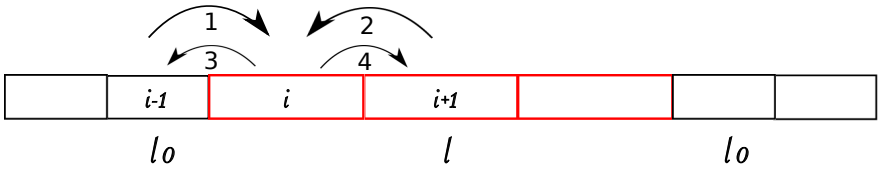


Figure 3.3: Four jump situations for compartments of different sizes. Red bins have the same length l, black bins' length is l_0.

We will explore pattern features on the three domain types through both deterministic and stochastic simulations.

3.4 Results and Discussion

In this section, we will first present model results based on our main model. Then we will present and discuss results related to other model settings. By comparing simulations, we will demonstrate that the settings in our main model lead to a good match with observed phenomenon in the *Caulobacter* cell cycle.

3.4.1 Main model results

The deterministic model was simulated in MATLAB using the ode15s solver. The stochastic simulation was written in C++ using Gillespie's stochastic simulation algorithm [49]. The spatiotemporal distribution plots were generated by Matlab.

Results presented in this section employ the domain with type B triangular ends (constant end shape) with *popZ* genes located at the center in the y direction and at the two boundaries between triangle and rectangle regions in the x direction. Specifically, the original copy of the *popZ* gene is located at 90% of the cell length from the start of the cell cycle, and at 50 min into the cell cycle, a new copy of the *popZ* gene appears at 10% of the cell length. For stochastic simulations we plot the average behavior of 100 simulations. The values shown in the following plots are all scaled to species populations for easy comparison. The non-integer population numbers are caused by taking the average of 100 stochastic simulations.

In Fig. 3.4e, f, PopZ polymers first appear at the old end of the cell, then PopZ polymerizes at the new end at about 75 min into the cell cycle, which is consistent with the 1D result.

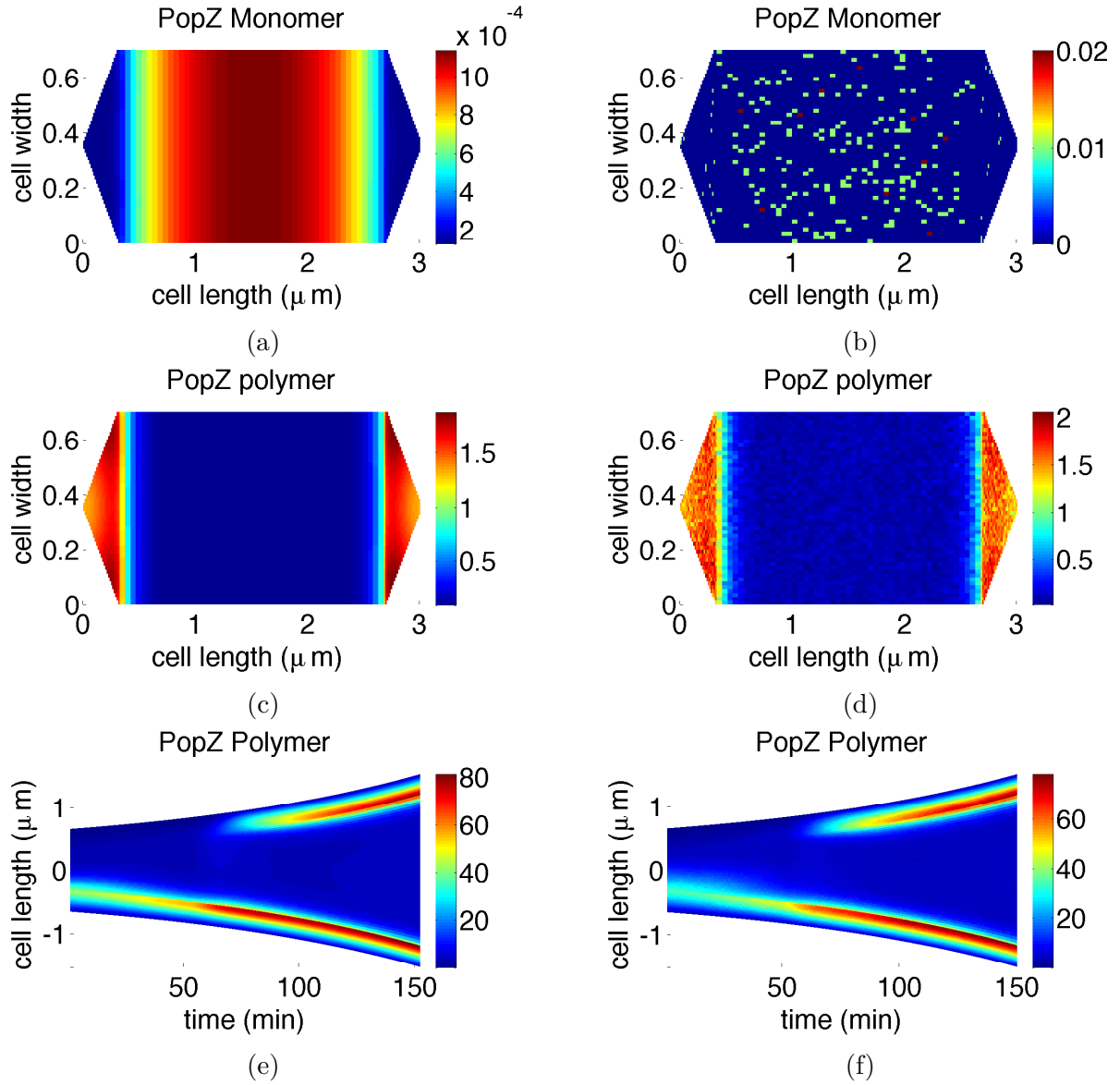


Figure 3.4: Results from the main model. Deterministic simulation (a, c, e) and stochastic simulation (b, d, f). The distribution of PopZ monomers (a, b) and PopZ polymers (c, d) at the end of the cell cycle. (e, f) space-time plot of PopZ polymer amount along the long axis of the cell (i.e., the sum of all polymers at a given location along the long axis).

Monomers accumulate at places with lower polymer concentrations (Fig. 3.4a, b). Generally, monomers only appear in the rectangular region and polymers dominate the triangular regions. Polymer levels at the two ends are nearly symmetric at the end of cell cycle (Fig. 3.4c, d). There are only a few monomers (under 10 molecules), much lower than the number of polymers (2000 - 3000 molecules). The difference of PopZ monomer in each bin between the deterministic and stochastic models can be narrowed down by increasing the number of stochastic simulation. Note that monomers are uniformly distributed across the width of the cell, while the polymer distribution has a small variation. The pattern of polymer inside the triangular regions appears to be annular, with slightly less polymer at the tips of the cell.

Bipolarity is observed in both deterministic and stochastic simulations. Although individual stochastic simulations show a great deal of variability, the average behavior of 100 stochastic simulations is nearly identical to the deterministic simulation.

3.4.2 Discussion

Effects of domain shape

To study the effects of different domain shapes, we plot the results of deterministic simulations on the rectangular cell shape and on the cell shape with type A triangular ends. Except for the different cell shapes, all other settings are the same as for the main model. (Stochastic simulations generate similar results.)

In Fig. 3.5e, f, for both idealized cell shapes PopZ polymerizes at the new pole at about 75 min into the cell cycle, and achieves a bipolar distribution by the end of the cell cycle. Thus PopZ bipolarity can be achieved in all three domain shapes. Yet, certain features of the patterns are different. Particularly in Fig. 3.5c, d, the total population of PopZ polymers at the old pole (to the right) is lower than that at the new pole (to the left). This asymmetry

is different from the symmetric distribution of PopZ polymers in the domain with type B triangular ends (Fig. 3.4c, d).

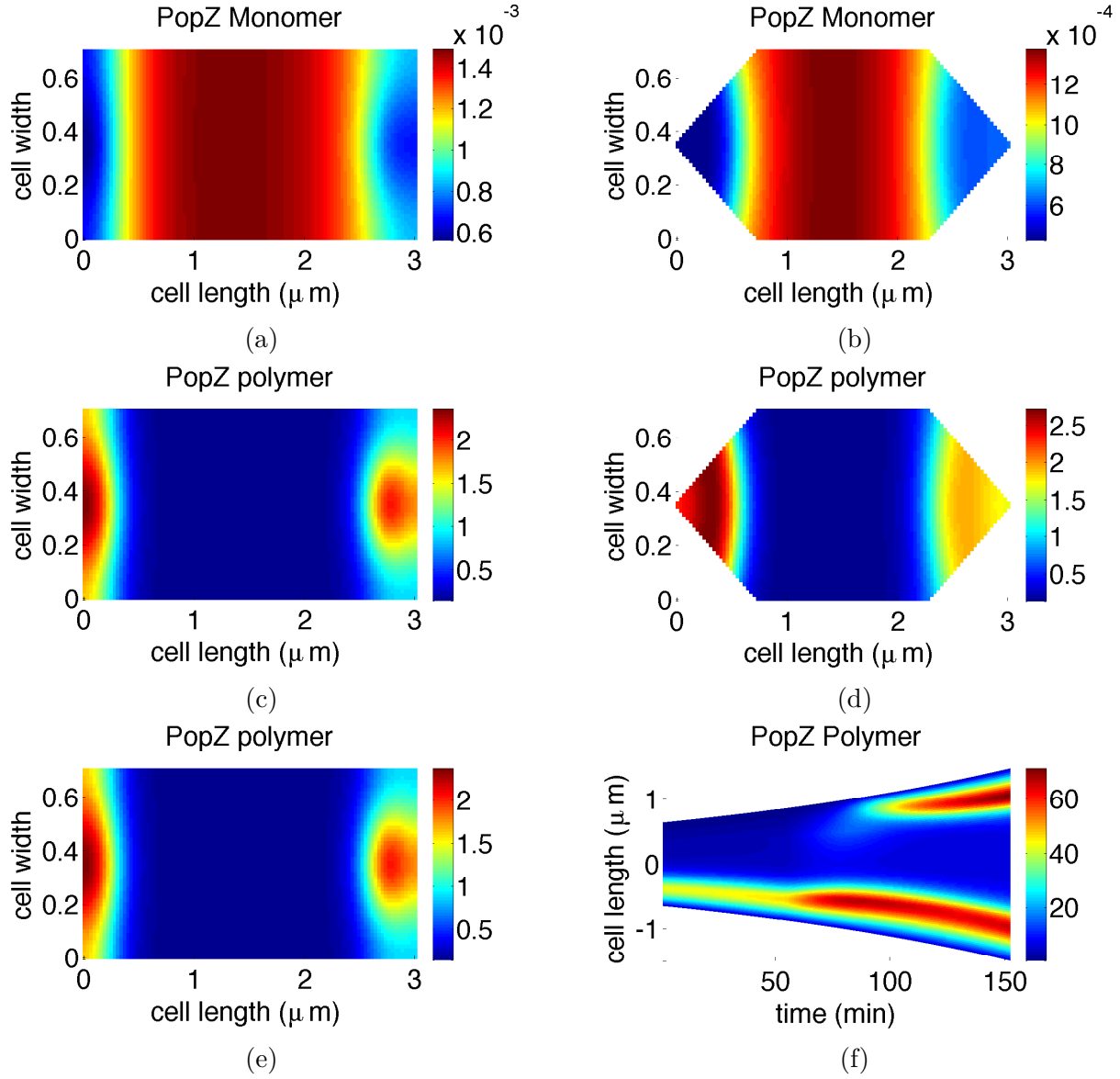


Figure 3.5: Deterministic simulations for a rectangular cell (a, c, e) and for a cell with type A triangular ends (b, d, f). (a, b, c, d) PopZ monomer and polymer distributions at the end of the cell cycle. (e, f) space-time plots for PopZ polymer distribution.

Effect of gene presence

Can the self-organizing model reproduce PopZ bipolarity by itself, in the absence of genetic encoding of PopZ monomer production? Here we consider the case without *popZ* gene(s), in order to study the self-organizing features of the monomer-polymer interaction. In this case, PopZ monomers are synthesized uniformly in the cell with a rate $k_{sm} = 0.09357$. The corresponding reaction is then changed to

$$\emptyset \xrightarrow{k_{sm}} Monomer \longrightarrow \emptyset.$$

We run simulations on all three domain types and find that they all produce patterns of PopZ polymerization. The observed patterns depend on initial conditions. A rectangular domain can generate a stripe pattern when the initial condition is uniform in the y direction (Fig. 3.6b). However, small perturbations in the y direction will lead to a spot pattern (Fig. 3.6a). Note that the spots can form either at the upper or lower edge.

For a domain with type A triangles, if the initial conditions at the two ends are the same, then the final pattern will be symmetric (Fig. 3.6d). Otherwise, it is asymmetric (Fig. 3.6c). But type B domains show less dependence on initial conditions (Fig. 3.6e, f).

Without *popZ* gene(s), the initial conditions of the model become a relevant factor in the bipolar distribution of PopZ. Initial conditions are known to determine the phase of patterns in Turing mechanisms. This conclusion only applies to the deterministic regime. It doesn't apply to stochastic simulations because the propensities of diffusion are much larger than those of chemical reactions. No matter where the molecules are initially placed, they quickly diffuse to other places in the cell before the reactions have much chance to establish a pattern.

Because resulting patterns can depend sensitively on initial conditions, we have explored

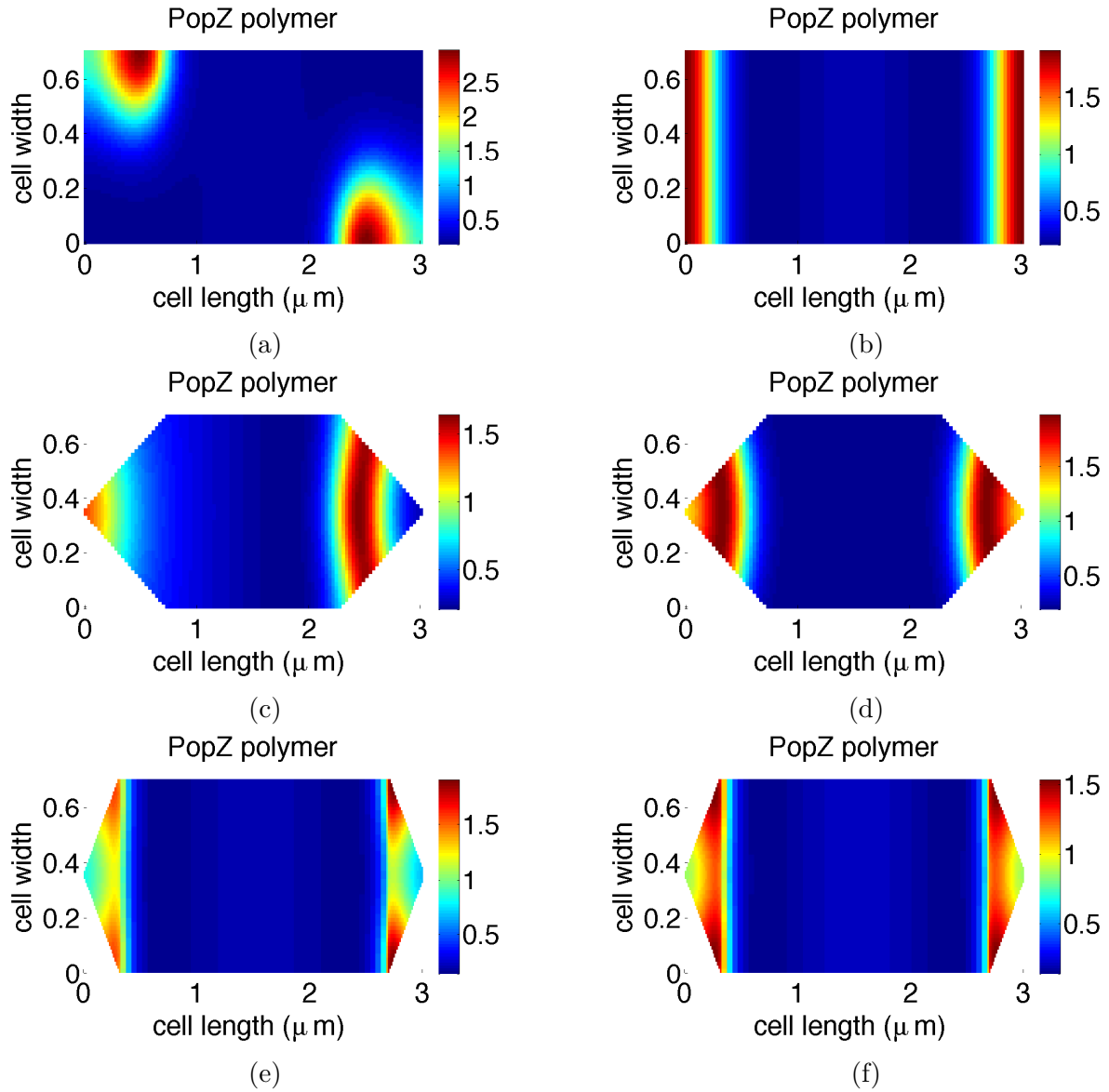


Figure 3.6: Deterministic simulations without genes: in the rectangle domain (a, b), the type A domain (c, d), and the type B domain (e, f). The left and right columns differ in the initial conditions of the simulations (see text).

four different initial distributions of PopZ polymer for the deterministic models: polymer initially concentrated at the old pole of the cell (which is the initial condition for our main model), at the new pole, at the center of cell, and at both poles. For the rectangular domain and the type A domain, the final distribution of PopZ polymers is very sensitive to initial conditions. In particular, if polymers are initially concentrated at the center of cell, then the peak stays in the center throughout the cell cycle (not shown). On the other hand, for the type B domain, PopZ is always polymerized in the triangular regions by the end of the cell cycle. Fig. 3.7 shows the dynamics in this case for the four different initial conditions under consideration. If polymers are initially present at both poles of the cell, they will remain so (Fig. 3.7d). If polymers are first present at one pole, then a second peak will form at the other pole later in the cell cycle (Fig. 3.7a, b). If polymers are initially concentrated in the center of the cell, the central peak will eventually break down and new peaks will form at the two ends (Fig. 3.7c).

Effect of gene location

We know that the positions of the *popZ* genes play an important role in PopZ polarization, as mentioned earlier. In order to study the effects of gene location, we ran simulations with different gene locations, ranging from 10%, 90% to 20%, 80% and 30%, 70% of the final length of a cell.

Both stochastic and deterministic simulations generate similar distributions of PopZ polymer. In the type B domain, polymers always concentrate in the triangular ends, even when the locations of the *popZ* genes is shifted towards the center of the cell, see Fig. 3.8 and Fig. 3.9. Polymers appear at the new pole at 75 min in all cases (Fig. 3.8c, d and Fig. 3.9c, d). However, even though polymers always accumulate in the triangular regions, the peak deviates further away from the tip of the triangle (Fig. 3.8a, b and Fig. 3.9a, b). When

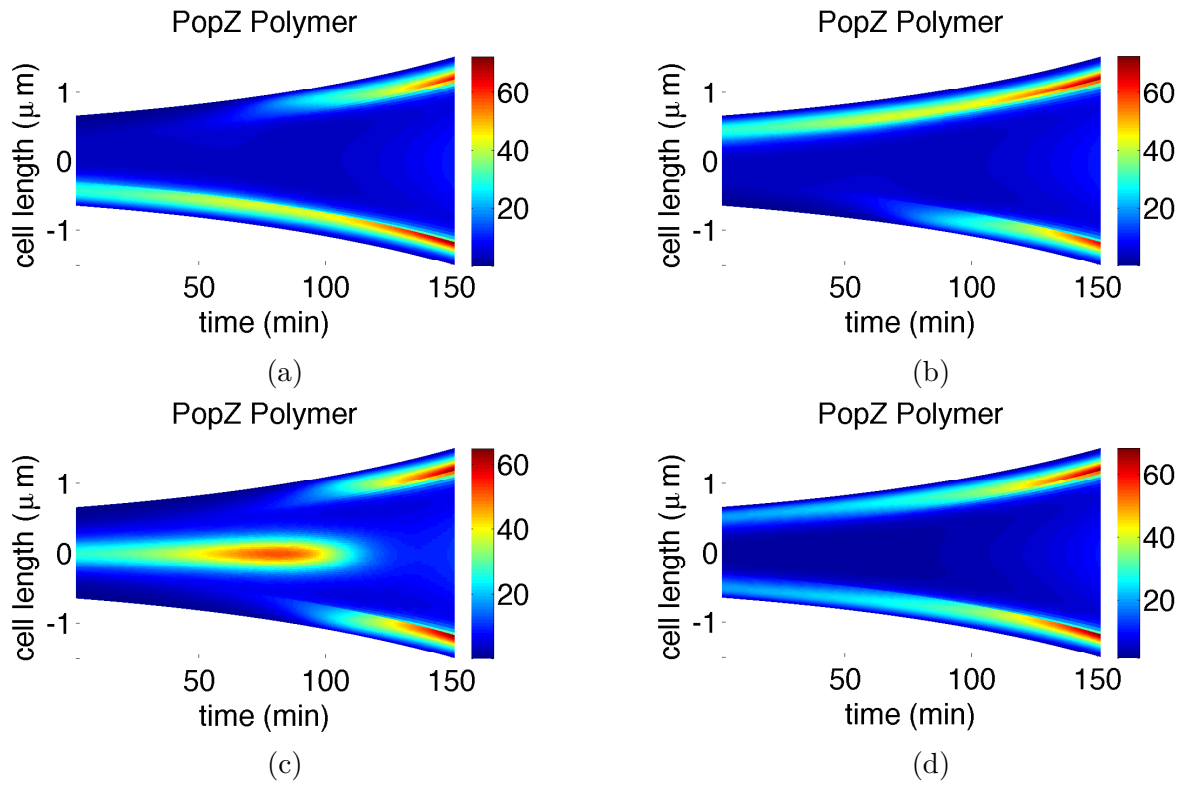


Figure 3.7: Deterministic simulations of PopZ polymerization on a type B domain, for the four different initial conditions proposed in the text.

the *popZ* genes are located at 30% and 70% of cell length, there is a weak peak of PopZ polymerization in the center of the cell, which is not clear in these plots. To see this effect more clearly, we divide the cell length into three parts: 0∼0.5μm (left), 0.5∼2.5μm (middle), 2.5∼3μm (right) and measure the total amount of PopZ polymers in each part.

As is evident in Fig. 3.10, more polymers fall in the middle part of the cell as the *popZ* genes move further to the center of the cell. Thus gene position does have an impact on the bipolar pattern of PopZ polymerization in this domain. However, the effects are less severe in this case compared with the other two domain types.

For the rectangular domain, a region of intense PopZ polymerization at the old pole moves with the gene's location (Fig. 3.11a, c). PopZ polymerization at the new pole is weaker and

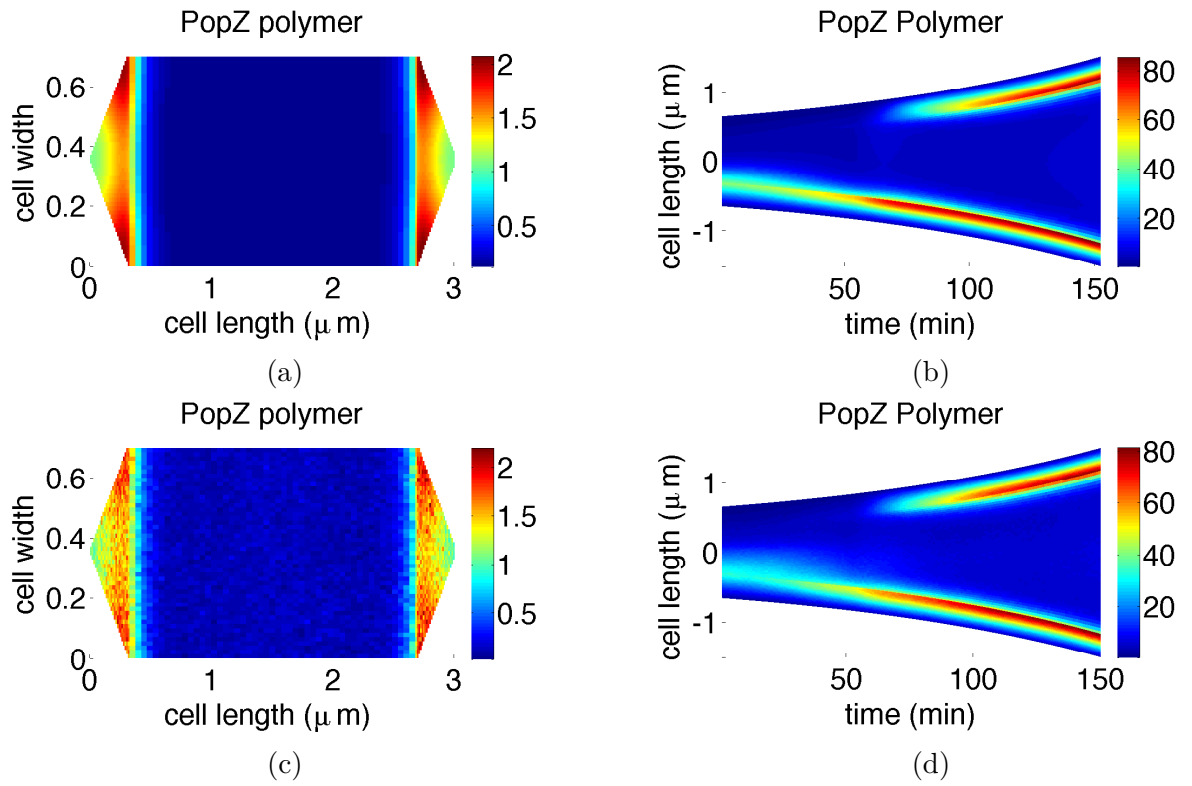


Figure 3.8: Deterministic (a, b) and stochastic (c, d) simulations for genes located at 20% and 80% of cell length in a type B domain.

more diffuse. For domains with type A triangular ends, the peak around the original gene also moves towards the center of the cell with the gene, while the peak at the new pole peak forms near the tip of the triangle (Fig. 3.11b, d). Clearly, gene position affects the bipolar pattern of PopZ polymerization on these domains.

Trimolecular reaction

As discussed in Section 3.3.2, discretization size affects the accuracy of SSA, especially for mechanisms involving bimolecular and trimolecular reactions. In our model, we have one trimolecular reaction in forming polymer from monomer (reaction (4)). The propensity for reaction (4) should be formulated as $\frac{cm_{ij}p_{ij}(p_{ij}-1)}{2}$, where m_{ij} and p_{ij} represent the populations

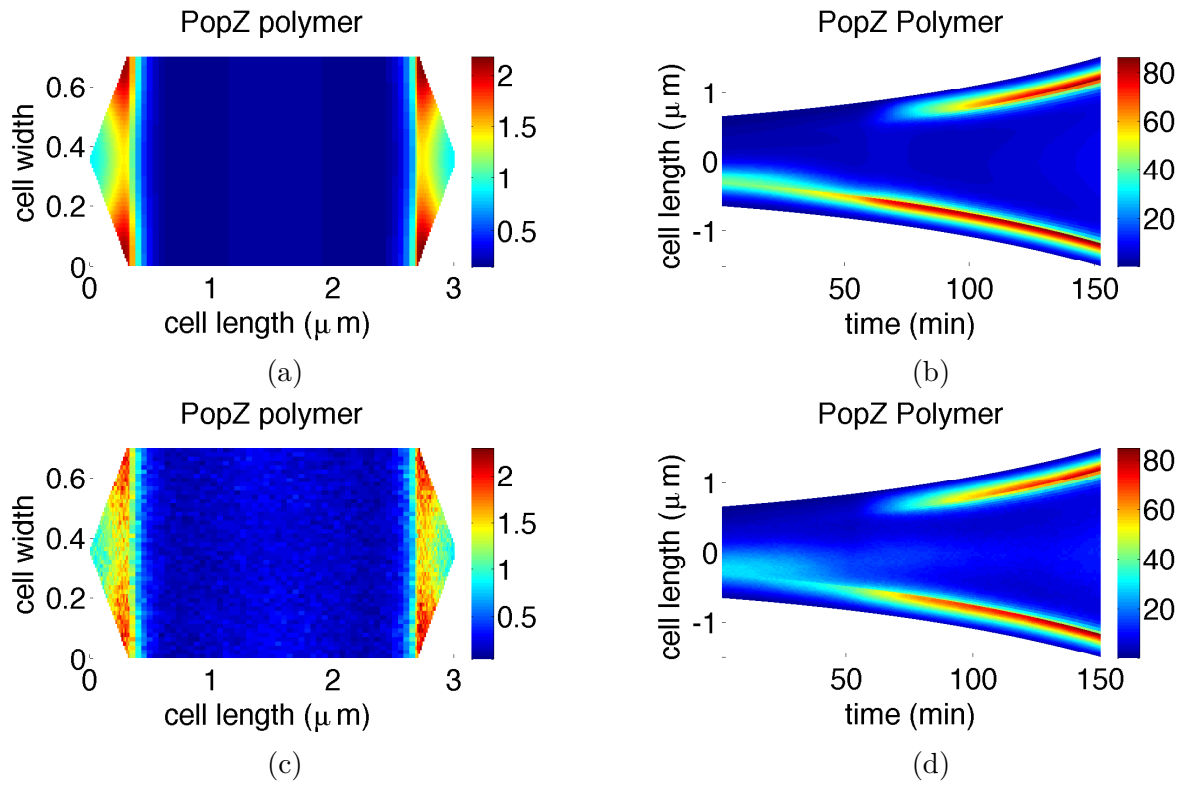


Figure 3.9: Deterministic (a, c) and stochastic (b, d) simulations for genes located at 30% and 70% of cell length in a type B domain.

of monomers and polymers in the compartment $(i,\ j)$. Trimolecular reactions are considered improbable in Gillespie's SSA. One may try to replace reaction (4) by a bimolecular reaction

$$Monomer + Polymer \longrightarrow 2Polymer, \tag{3.7}$$

with reaction rate given by $km_{ij}p_{ij}^2$. For deterministic simulations, these two reaction types generate similar results. But in stochastic simulations, they may make a big difference. Reaction (4) implies that there must be two PopZ polymers and one PopZ monomer in the same compartment for the reaction to fire, while reaction (3.7) requires only one PopZ polymer and one PopZ monomer. When the compartment size is small, reaction (3.7) is more likely to fire than reaction (4).

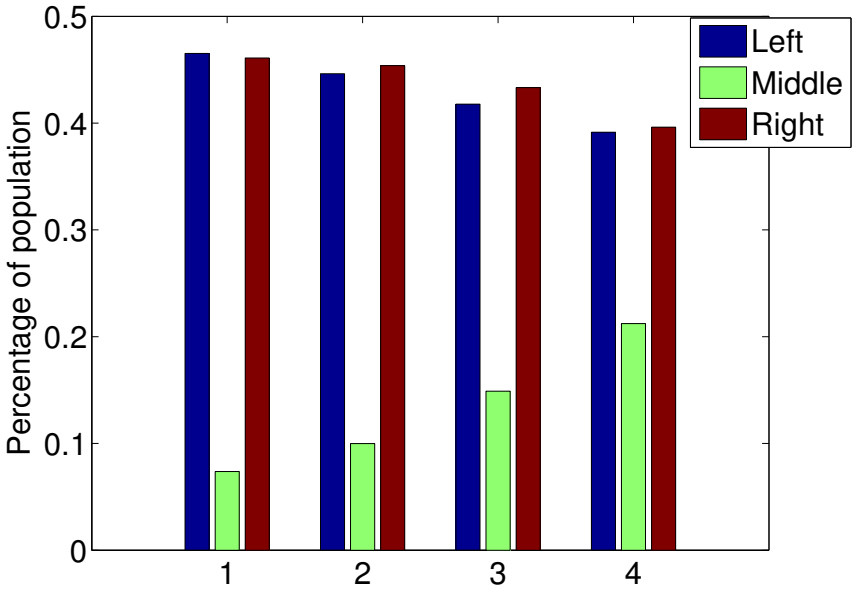


Figure 3.10: Stochastic simulations of polymer distribution in the left, middle and right portions of a type B cell at the end of the cell cycle. Groups 1, 2, 3, 4 correspond to gene locations at 10%, 90%, 20%, 80%, 30%, 70% and 40%, 60% of cell length.

Fig. 3.12 shows the results of stochastic simulations based on reaction (3.7). For the type A domain, the PopZ polymer distribution is diffuse, covering nearly two thirds of the cell (Fig. 3.12a, c). The rectangular domain gives similar results (not shown). The type B domain, on the other hand, still shows a clear bipolar distribution of PopZ polymers (Fig. 3.12b, d). Hence, we conclude that for type B domains, which is our preferred cell shape, either reaction (4) or reaction (7) gives satisfactory results.

Noise

So far we have plotted the average behavior of 100 stochastic simulations. For individual stochastic runs, variations in the distributions of PopZ polymers across the width of a cell are obvious, especially for simulations on the rectangular domain. Fig. 3.13 illustrates this effect by plotting a single stochastic simulation on a rectangular domain and on a type B

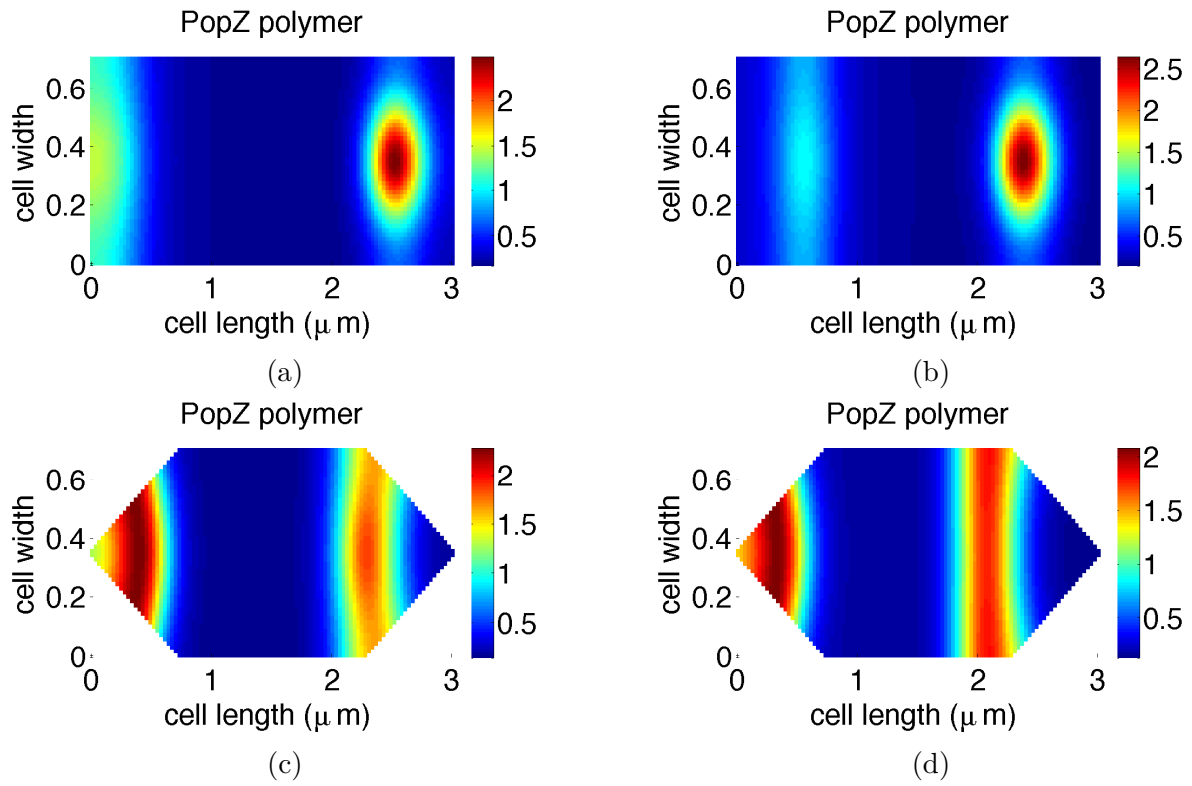
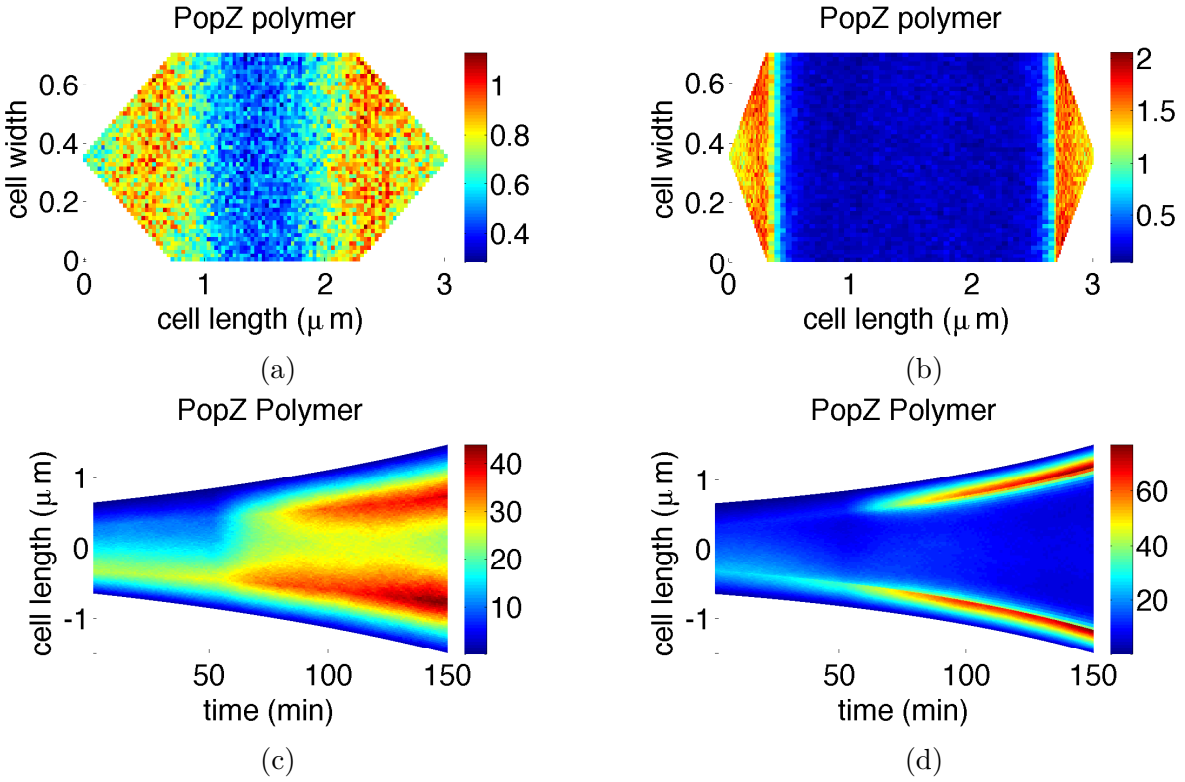


Figure 3.11: Deterministic simulations of PopZ polymer distributions at the end of cell cycle for the rectangular domain (a, c) and for the type A domain (b, d). The *popZ* genes are located at 20% and 80% (a, b) or at 30% and 70% (c, d).

domain.

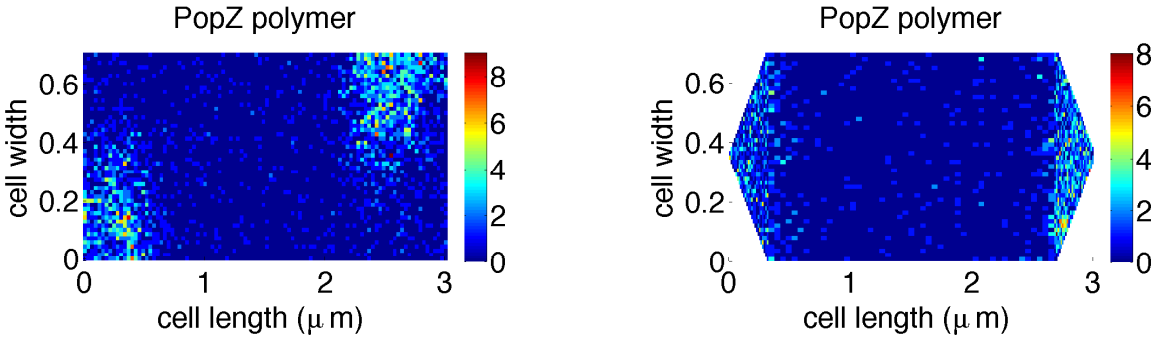
On the rectangular domain, PopZ polymers are more likely to accumulate at the corners of the rectangle (Fig. 3.13a). Whether PopZ polymerizes at an upper or lower corner is arbitrary and probably results from stochastic fluctuations. For the domain with type B triangular ends, PopZ reliably polymerizes within the triangular ends (Fig. 3.13b). This "corner" effect may be related to the observation in real *Caulobacter* cells for PopZ to polymerize in regions of high curvature of the cell wall.

Besides the corner preference, we also find that polymers sometimes accumulate in the center as well as the poles in all types of domains (see, e.g., Fig. 3.14). This tendency is especially prevalent in rectangular and type A domains, which seems to be a stochastic effect because



(a) (b) (c) (d)

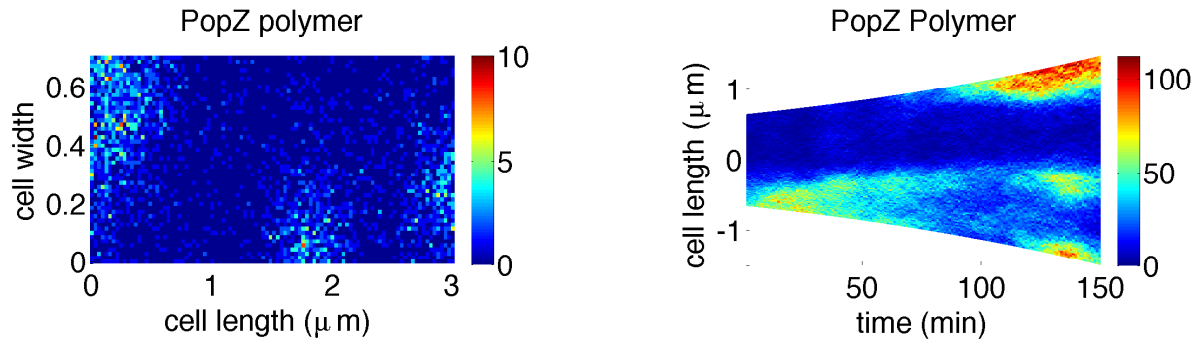
Figure 3.12: Stochastic simulations using reaction (7) instead of reaction (4) for domains with triangular ends of type A (a, c) and type B domain (b, d).



(a) (b)

Figure 3.13: One stochastic simulation of corner preference for the rectangular domain (a) and the type B domain (b).

it disappears in the mean behavior.



(a) (b)

Figure 3.14: An example of a central peak of PopZ polymerization on a rectangular domain. (a) population distribution of PopZ polymer. (b) space-time plot of PopZ polymer.

In order to quantify this tendency to form central peaks, we calculated the percentage of cells forming central peaks based on 100 stochastic simulations on each type of domain; see Fig. 3.15. The frequency of central peak formation is only 2% for type B domains, followed by (5%) for rectangular domains. Type A domains form central peaks quite frequently (11%).

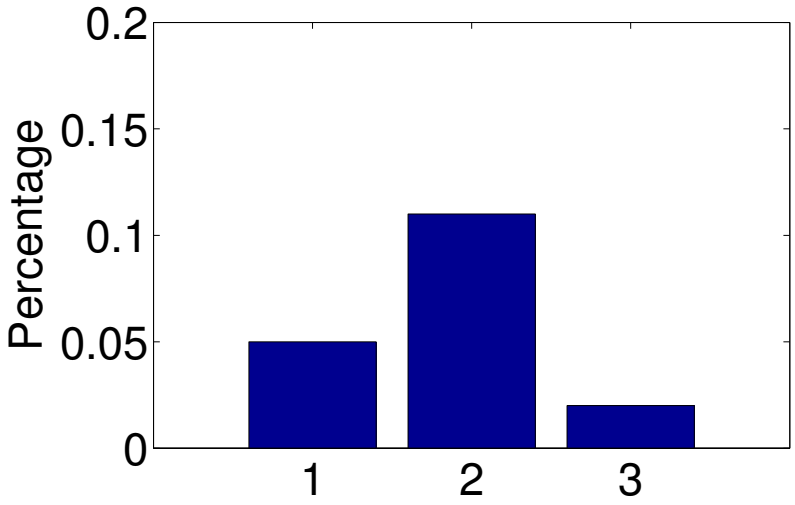


Figure 3.15: Percentage of cells exhibiting a central peak of PopZ polymerization. Groups 1, 2, 3 correspond to rectangular domains and domains with type A and type B triangular ends, respectively.

3.5 Conclusions

To model bipolar patterns of PopZ polymerization in *Caulobacter* cells, we propose a reaction-diffusion mechanism in two spatial dimensions, based on a Turing-instability for pattern formation, coupled with spatial localization of the *popZ* gene before and after chromosome replication. Under this mechanism, PopZ drives its own spatiotemporal distribution by a self-assembly process, where the locations of the *popZ* genes are found to be necessary for robust bipolarity of PopZ polymerization. We studied pattern features for different domain shapes and different gene locations. We conclude that gene location is important in forming a stable bipolar pattern. We also explored the pattern formation without participation of the *popZ* gene, in which case the initial distribution of PopZ polymers becomes a determining factor in the final pattern. In stochastic simulations, the way space is discretized and the trimolecular reaction is simulated are also important factors explored in this study. As more precise observations of PopZ polymerization in *Caulobacter* cells become available, we intend to make further revisions and improvements to the model.

Chapter 4

Analysis and Remedy of Negativity Problem in Hybrid Stochastic Simulation Algorithm and its Application

4.1 Introduction

The stochastic simulation algorithm (SSA), also often called Gillespie's algorithm [49, 50], is a major stochastic simulation method for simulating stochastic effects in biochemical networks. Although the SSA is quite reliable in numerous applications in computational biology, the algorithm is computationally intensive and inefficient for systems with fast reactions or large populations. Though many optimizations have been proposed to improve the efficiency of the algorithm [6, 16, 48, 80, 87, 117], the essential idea of simulating each reacting event in a dynamical system makes it unpromising for large and complex biochemical systems compared to traditional deterministic methods.

To avoid the expensive computational cost of the SSA, researchers began studying approximation strategies. One well-known approximation is the τ-leap method [52], which approx-

imates many reaction events in an interval of τ instead of simulating each reaction. As biological networks at single cell levels usually have large scale discrepancies in populations of species such as mRNAs and proteins, as well as rate constants among different reactions, research is increasingly focused on hybrid methods targeting multiscale systems that contain species populations or reaction rates with widely varying scales [18, 34, 55, 105]. One branch of the hybrid method is the piecewise deterministic Markov process [34, 45, 64], which mixes the deterministic evolution with random jumps. Under the SSA branch, one hybrid method is to combine the τ-leap algorithm and the SSA for multiscale features among species populations [18]. Species and corresponding reactions are partitioned into two sets based on their populations, one simulated by the SSA and the other simulated by the τ-leap method. In a multiscale system, fast reactions can reach partial equilibrium or quasi-steady-state under certain conditions. Hybrid methods, like the slow-scale SSA method (ssSSA) [17, 19] and the stochastic quasi-steady-state SSA method (SQSSA) [105], were proposed based on this property. The ssSSA partitions the system into fast reaction and slow reaction sets, assuming partial equilibrium for the fast reactions, while simulating the slow reactions with the SSA. Similarly, the SQSSA first separates intermediate species and their corresponding reactions from the system, then assumes that the separated subsystem is at a steady state and simulates the rest of the system with the SSA. But both methods have limitations on parameter space to ensure the system validity [115, 123].

For general cases where fast reactions do not always reach a steady state or partial equilibrium, Haseltine and Rawlings [55] proposed a hybrid method (hereafter referred to as the HR hybrid method), which modeled part of the system by continuous dynamics (ordinary differential equations (ODEs) or Langevin equations), while keeping the rest discrete. The idea of the HR hybrid method was further explored, improved, and extended to several hybrid methods [77, 84, 85, 113]. Salis et al. [113] partitioned the system into fast and slow

reaction groups, and modeled the fast group by Langevin equations and the slow group by the SSA. Later, Liu et al. [84] improved the efficiency of the HR hybrid method by a different partitioning strategy: reactions that have both low-density reactants and small reaction rates were put into the slow reaction subsystem and all the other reactions into the fast reaction subsystem. Wang et al. [129] optimized the implementation efficiency for the HR hybrid method and compared the efficiencies of the hybrid method coupled with three traditional ODE solvers RADAU5, DASSL, and DLSODAR. Lecca et al. [77] further divided the system into three sets: fast reactions, moderate reactions, and slow reactions, where the simulation of moderate reactions can be switched between stochastic and deterministic processes based on the reaction firing time during the system evolution. For spatial models or domains, hybrid methods were introduced under reaction-diffusion systems, where diffusion was approximated by differential equations to improve simulation efficiency [30, 85, 109].

Simulation tools, e.g., Hy3S [114] and MoBioS [77], and software like COPASI [60] included the HR hybrid method and provided users with different simulation choices and implementation rules. As to the application to complex biochemical models, Wang et al. [3, 128] used the HR hybrid method to model a budding yeast cell cycle. The method largely reduced the simulation time and the results matched well with experimental data on cell cycle properties and prototypes of most mutant cells. To mathematically analyze the accuracy of the HR hybrid method, Chen et al. [28] used the next reaction time of the slow reaction event as the accuracy benchmark and showed that the HR hybrid method is accurate in linear chain systems under certain conditions (either large populations of reactants in the fast subsystem or large scale differences of reaction rates between fast reactions and slow reactions). This work also demonstrated that the HR hybrid method is valid for a much greater region in system parameter space than those for the ssSSA and the SQSSA methods.

However, in the HR hybrid method framework, populations of some reactant species may

become negative if they are involved in both deterministic and stochastic systems. Take system (4.1) as an example. If reaction rate constants satisfy $f_1 \gg k_c$ and $b_1 \gg k_c$, the system can be divided into two groups: the fast reaction group and the slow reaction group, containing the reversible and irreversible reactions, respectively.

$$\mathrm{S}_1 \underset{b_1}{\overset{f_1}{\rightleftharpoons}} \mathrm{S}_2 \xrightarrow{k_c} \mathrm{S}_3. \tag{4.1}$$

Assume that this system has two S_1 molecules at the beginning, and the system parameters are $f_1 = 1, b_1 = 9, k_c = 0.01$. Then, compared with the slow system, the fast system can be considered at equilibrium, which gives $x_1 = 1.8$ and $x_2 = 0.2$, where x_i denotes the mean population of species S_i. Thus, when a slow reaction fires, x_2 is reduced to -0.8. Only after a certain period of time, S_2 may become nonnegative again through the reaction $\mathrm{S}_1 \to \mathrm{S}_2$.

Negative populations may also appear in stochastic simulations of reaction-diffusion systems, especially when low-density species are distributed in a well-meshed space. For example, in a one-dimensional spatial model of the *Caulobactor* cell cycle [78], the spatial domain is divided into 50 equally spaced bins. Since diffusion happens much faster than chemical reactions in the cell, diffusion events are modeled as continuous deterministic equations whereas chemical reactions are modeled by the SSA. In the initial stage of the cell cycle, protein DivKp has a low population (< 50). Thus the average population of DivKp inside each bin is < 1. Note that the mean population of a species is a real number if it is involved in the fast subsystem. As illustrated in Fig. 4.1, if there is a degradation reaction of DivKp firing in the ith bin, its population would become negative. Therefore, any consumption of those low population species in the spatial stochastic domain may lead to a negative population. The phenomenon of species' populations becoming negative, as shown in the above two examples, is called the negativity problem for the HR hybrid method.

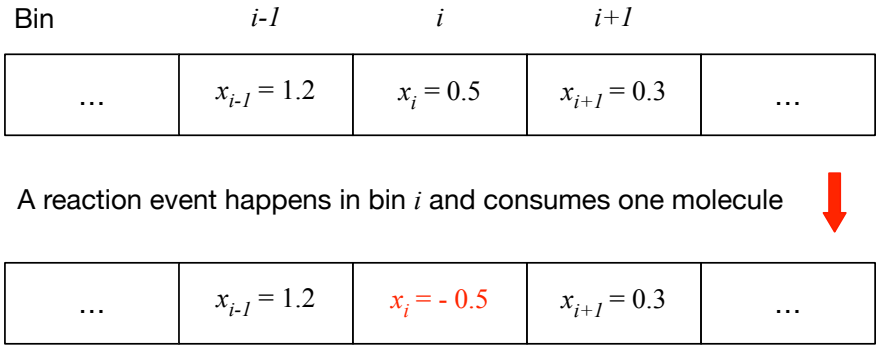


Figure 4.1: An example of a negativity phenomenon in a reaction diffusion system. x_i denotes the population of DivKp in the ith bin.

This chapter is organized as follows. In the Methods section, we present the theoretical derivation of the second exit time of the chemical master equation (CME), the HR hybrid method, and three proposed remedies for the negativity problem. The Results and Discussion section analyzes the potential negativity effects on the accuracy of linear chain systems for a simple case ($n = 2$) and a complex case ($n = 10$). We test three remedies on three examples: a closed linear chain system, a nonlinear system, and a realistic biological system. Summary and conclusions are given last.

4.2 Methods

4.2.1 Second slow reaction firing time

Our prior work [28] analyzed the accuracy of the HR hybrid method by studying the next slow reaction firing time (NSRFT, also called the first exit time). Since the negative population problem mostly emerges after a slow reaction, the first exit time does not show the influence of negative population. So, we further extend that work and study the second slow reaction firing time (SSRFT, which can also be referred to as the second exit time). The SSRFT

reflects the influence of a (possible) negative population on the firing of slow reactions. With this analysis, we hope to gain certain insight on the impact of the negativity problem on algorithm accuracy. In the HR hybrid method we assume reactants become negative only after a slow reaction happens, which is after the first exit time. Negative populations may also arise, with a much smaller probability, in the numerical integration of ODEs. That is not our focus here.

We use the same linear chain reaction network in Refs. [28, 127] as a study example, shown below.

$$\mathrm{S}_1 \underset{b_1}{\overset{f_1}{\rightleftharpoons}} \mathrm{S}_2 \underset{b_2}{\overset{f_2}{\rightleftharpoons}} \cdots \underset{b_{n-1}}{\overset{f_{n-1}}{\rightleftharpoons}} \mathrm{S}_n \xrightarrow{k_c} \mathrm{S}_{n+1}. \tag{4.2}$$

A particle can exit the reversible chain system through S_n with reaction rate k_c. In most cases, k_c is comparably less than reaction rates f_i and b_i for $1 \leq i \leq n-1$. In many applications, the reversible chain reactions can be considered as a fast subsystem and the irreversible reaction (exit to S_{n+1}) as a slow subsystem. With this partitioning strategy, if $x_n < 1$, then S_n will become negative whenever a slow reaction fires.

4.2.2 SSRFT for the CME

While the first exit time (NSRFT) denotes the time when the next slow reaction fires in the linear chain system (4.2), the second slow reaction firing time (SSRFT) is the time period from the system start to the second time the slow reaction fires.

Recapping the derivation of NSRFT in Ref. [28], the SSRFT can be considered as the probability that two independent events (NSRFT) happen in a time interval $[0,\ t]$. In system (4.2), $\vec{x}(t) = (x_1(t),\ x_2(t),\ \ldots,\ x_{n+1}(t))^T$ represents the system state at time t. If

there is only one particle in the system, we denote the probability of $x_i = 1$ as

$$p_i(t) = \mathrm{P}[x_i(t) = 1], \quad \text{for } i = 1, \ldots, n+1.$$

and the probability vector for species $\mathrm{S}_1, \mathrm{S}_2, ..., \mathrm{S}_n$ as

$$\mathbb{P}(t) = [p_1(t), \ldots, p_n(t)]^T$$

In the chemical master equation system, we have

$$\frac{d\mathbb{P}}{dt} = -A\mathbb{P}, \tag{4.3}$$

where A is stoichiometric matrix,

$$A = \begin{bmatrix} f_1 & -b_1 & 0 & 0 & \cdots & 0 \\ -f_1 & b_1 + f_2 & -b_2 & 0 & \cdots & 0 \\ 0 & -f_2 & b_2 + f_3 & -b_3 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -f_{n-2} & b_{n-2} + f_{n-1} & -b_{n-1} \\ 0 & \cdots & 0 & 0 & -f_{n-1} & b_{n-1} + k_c \end{bmatrix}.$$

As there is only one particle all the time in the system, we have $\sum_{i=1}^{n+1} p_i(t) = 1$. And $x_{n+1}(t) = 1$ if and only if the first exit time $T_1 \leq t$. Thus

$$\mathrm{P}[T_1 \leq t] = p_{n+1}(t) = 1 - \sum_{i=1}^{n} p_i(t) = 1 - \vec{e}^T \mathbb{P}(t),$$

where $\vec{e} = [1, \ldots, 1]^T$. Given an initial condition $\vec{e}_j$ (a vector with the jth element equal to

1 and all other elements equal to 0), the NSRFT is (see Ref. [28])

$$q_j(T_1) = \mathrm{P}[T_1 > t] = \vec{e}^T e^{-At}\vec{e}_j = 1 - p_{n+1}(t) = 1 - \int_0^t k_c p_n(s)ds.$$

In a general case where there are m particles ($\vec{x}_0 = [m_1, m_2, \ldots, m_n]^T$, $m = \sum_{i=1}^n m_i$) in this linear system, as particles are independent of each other, the NSRFT is

$$q(T_1) = \prod_{j=1}^{n} q_j^{m_j}(T_1).$$

Based on similar analysis to NSRFT, the second slow reaction firing time can be written as

$$\mathrm{P}[T_2 \le t] = 1 - \prod_{j=1}^{n} q_j^{m_j} \; - \sum_{i=1, m_i \neq 0}^{n} C_{m_i}^1 (1 - q_i) q_i^{m_i - 1} \prod_{j=1, j\neq i}^{n} q_j^{m_j}. \tag{4.4}$$

For a simple case where $n = 2$ and the initial condition: $m_1 = 2$ and $m_2 = 0$, we have

$$\mathrm{P}[T_2 \le t] = \left(1 - q_1(t)\right)^2. \tag{4.5}$$

4.2.3 SSRFT for the HR hybrid method

In the HR hybrid method, define the state vector in the fast subsystem as $\vec{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$. The fast subsystem is modeled as a linear ODE system, denoted as

$$\frac{d\vec{x}(t)}{dt} = -\tilde{A}\vec{x}(t). \tag{4.6}$$

$\tilde{A}$ is a $n \times n$ matrix given by

$$\tilde{A} = A - k_c \vec{e}_n \vec{e}_n^T,$$

where only the last elements of matrices $\tilde{A}$ and A are different, $\tilde{A}(n,n) = b_{n-1}$, $A(n,n) = b_{n-1} + k_c$.

Denote T_1 as the NSRFT, we have

$$\int_0^{T_1} k_c x_n(t)dt = \int_0^{T_1} \vec{e}_n^T e^{-\tilde{A}t}\vec{x}_0 dt = r,$$

where r is a unit exponential random number. In order to compare it with the CME result, we have to change the exponential random number to a unit uniform random number u by the relation $u = 1 - e^{-r}$. The above equation can be written as

$$\mathrm{P}[T_1 \le t] = 1 - e^{-\int_0^{T_1} k_c \vec{e}_n^T e^{-\tilde{A}t}\vec{x}_0 dt} = u.$$

And the density function of the NSRFT is

$$p(T_1) = k_c \vec{e}_n^T e^{-\tilde{A}t}\vec{x}_0 e^{-\int_0^{T_1} k_c \vec{e}_n^T e^{-\tilde{A}t}\vec{x}_0 dt}. \tag{4.7}$$

The SSRFT for the HR hybrid method can be considered as the next slow reaction firing time with a different initial condition $\vec{x}_{T_1}$ (the system state after the first exit time T_1),

$$\int_{T_1}^{T_2} k_c x_n(t)dt = \int_{T_1}^{T_2} \vec{e}_n^T e^{-\tilde{A}(t-T_1)}\vec{x}_{T_1} dt = r,$$

where $\vec{x}_{T_1} = e^{-\tilde{A}T_1}\vec{x}_0 - \vec{e}_n$. Thus

$$p(T_2|T_1) = (k_c \vec{e}_n^T e^{-\tilde{A}(T_2-T_1)}\vec{x}_{T_1} - k_c \vec{e}_n^T \vec{x}_{T_1}) e^{-\int_{T_1}^{T_2} k_c \vec{e}_n^T e^{-\tilde{A}(t-T_1)}\vec{x}_{T_1} dt}. \tag{4.8}$$

Therefore, the SSRFT is

$$p(T_2) = \int_0^\infty p(T_1)p(T_2|T_1)dT_1. \tag{4.9}$$

Below we present three strategies to handle the negativity problem and compare the corresponding impact on the SSRFT.

4.2.4 SSRFT for Remedy I: Zero-Population

For the negativity problem in the HR hybrid method, one simple treatment is to immediately change any negative value to zero. So we name this strategy the **Zero-Population** remedy: after a slow reaction happens in the stochastic subsystem, detect whether any corresponding reactants become negative, if so set them as zero and continue the simulation, otherwise continue without modification.

To check the effect of the Zero-Population remedy, we study the second slow reaction firing time of the system under this rule. Since the rule takes effect after the first slow reaction, only the conditional density function $p(T_2|T_1)$ is different from the HR hybrid method when $x_n < 0$. In this scenario, the initial condition of the second slow reaction firing time in linear chain system (4.2) is

$$\vec{x}_{T_1} = e^{-\tilde{A}T_1}\vec{x}_0 - \vec{e}_n, \quad \vec{x}_{T1}(n) = \max(0, \vec{x}_{T1}(n)). \tag{4.10}$$

The conditional density function of the second exit time in the Zero-Population remedy is similar to Eq. (4.8), just replace the initial condition with $\vec{x}_{T1}(n)$ in Eq. (4.10).

4.2.5 SSRFT for Remedy II: Zero-Reaction

While avoiding the negative effect, the Zero-Population remedy changes the conservation law in the system. For example, the total amount of all species in the closed system (4.2) should always be m, but simply changing the negative population $(-m_\delta)$ to zero causes the total population to increase to $m + m_\delta$. In order to follow the law of conservation, which is important in many practical applications, one idea is to scale down a reaction when the slow reaction happens with a reactant population less than one. Take the reaction $\mathrm{X} \to \mathrm{Y}$ as an example. Suppose the population of reactant X is less than one $(0 < m_X < 1)$, then instead of consuming one molecule of X, scale the reaction by a ratio of m_X, which produces m_X molecule of Y. However, this method breaks the natural discrete feature of slow reaction firing and can cause significant errors for stochastic models.

Alternatively, one can simply set all related reaction propensities as zero when a negative population appears. So we have the second remedy named the **Zero-Reaction** rule: set all reaction propensities involving negative species as zero in corresponding subsystems until the negative species become nonnegative.

After the first slow reaction, the system status is $\vec{x}_{T_1} = e^{-\tilde{A}T_1}\vec{x}_0 - \vec{e}_n$. For $x_n < 0$, reaction rates for all reactions that S_n participated in the fast subsystem are treated as zero, and the corresponding coefficient matrix for the ODEs is denoted as $\hat{A}$. The propensity of the slow reaction, which S_n participated in, is also set to zero. Since system (4.2) has only one slow reaction, no reaction in the slow subsystem will fire for negative x_n. The fast subsystem keeps running until $x_n = 0$. We denote τ as the time period of x_n evolving from negative to zero in the ODE system:

$$\vec{e}_n^T e^{-\hat{A}\tau}\vec{x}_{T_1} = 0, \tag{4.11}$$

where the above equation has a unique solution τ. So after time $T_1 + \tau$, the system is back

to normal, with the system state $\vec{x}_\tau = e^{-\hat{A}\tau}\vec{x}_{T_1}$.

Thus, the conditional density function of the SSNFT under the Zero-Reaction remedy is

$$p(T_2|T_1) = (k_c\vec{e}_n^T e^{-\tilde{A}(T_2-T_1-\tau)}\vec{x}_\tau - k_c\vec{e}_n^T\vec{x}_\tau)e^{-\int_{T_1+\tau}^{T_2} k_c\vec{e}_n^T e^{-\tilde{A}(t-T_1-\tau)}\vec{x}_\tau dt}. \tag{4.12}$$

4.2.6 SSRFT for Remedy III: Zero-Time

Another remedy for the negativity problem is called the **Zero-Time** rule: whenever a species' population become negative, pause the system and run a separate virtual ODE system G'_f that only contains reactions related to the negative species. When the species in the virtual system recovers to a nonnegative state, restart the original hybrid system with the updated system state. Because the system recovers under existing reaction systems, the conservation law is obeyed.

In system (4.2), the system state is still $\vec{x}_{T_1} = e^{-\tilde{A}T_1}\vec{x}_0 - \vec{e}_n$ after the first slow reaction. For $x_n < 0$, build a separate ODE system that only contains the last reversible reaction,

$$\mathrm{G}'_f : \mathrm{S}_{n-1} \underset{b_{n-1}}{\overset{f_{n-1}}{\rightleftharpoons}} \mathrm{S}_n. \tag{4.13}$$

Run the G'_f system until $x_n = 0$, which costs time ρ from the below equation.

$$\vec{e}_2^T e^{-B\rho}\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix} = 0, \quad B = \begin{bmatrix} f_{n-1} & -b_{n-1} \\ -f_{n-1} & b_{n-1} \end{bmatrix} \tag{4.14}$$

Since the recovery time ρ is not counted in system evolution, we have the conditional density

function of the second exit time under the Zero-Time remedy as

$$p(T_2|T_1) = (k_c \vec{e}_n^T e^{-\tilde{A}(T_2-T_1)} \vec{x}_\rho - k_c \vec{e}_n^T \vec{x}_\rho) e^{-\int_{T_1}^{T_2} k_c \vec{e}_n^T e^{-\tilde{A}(t-T_1)} \vec{x}_\rho dt}, \tag{4.15}$$

where $\vec{x}_\rho$ is the system state after running the G'_f system for ρ.

4.3 Results and Discussion

4.3.1 Theoretical analysis of SSRFT

A simple case ($n = 2$)

This subsection examines the SSRFT of the linear system (4.2) with $n = 2$, as shown in (4.1).

We first check conditions that may cause the negativity problem, such as parameters and initial conditions. In system (4.1), after the first slow reaction, we have

$$\vec{x}_{T_1} = e^{-\tilde{A}T_1} \vec{x}_0 - \vec{e}_2, \quad \tilde{A} = \begin{bmatrix} f_1 & -b_1 \\ -f_1 & b_1 \end{bmatrix},$$

thus the population of S_2 at time T_1 is

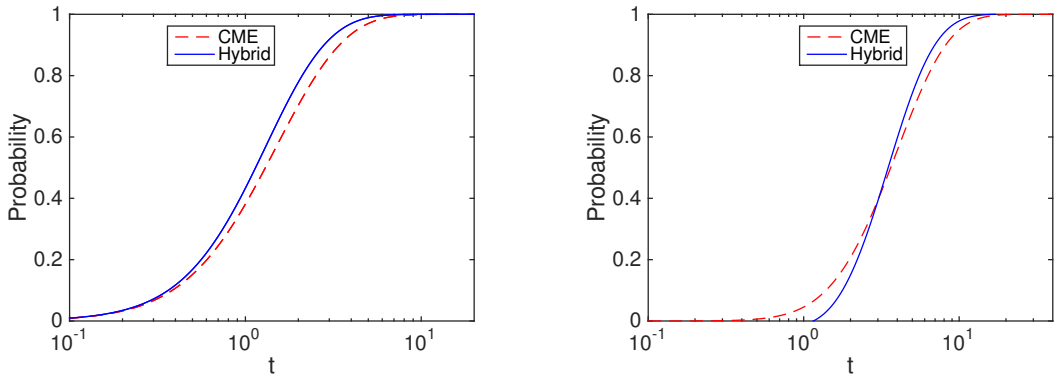
$$\vec{x}_{T_1}(2) = \frac{(m_2 b_1 - m_1 f_1) e^{-(b_1+f_1)T_1} + (m_1 + m_2) f_1}{b_1 + f_1} - 1.$$

The first slow reaction firing may happen from time 0 to ∞. To ensure that the population of S_2 is nonnegative for all possible T_1, we should have $\vec{x}_{T_1}(2) \geq 0$ for all T_1. We solve this inequality and get

$$m_1 + m_2 \geq \frac{f_1 + b_1}{f_1} \quad \text{and} \quad m_2 \geq 1. \tag{4.16}$$

For the parameter set $f_1 = b_1 = 1$, the population of S_2 may become negative when the initial condition satisfies $m_2 = 0$ (Assume $m_1 + m_2 \geq 2$, so a second slow reaction is possible). For the parameter set $f_1 = 1, b_1 = 10$, the population of S_2 may become negative when the initial condition satisfies $m_2 = 0$ or $m_1 + m_2 < 11$.

Fig. 4.2 presents the cumulative distribution functions (CDFs) of NSRFT and SSRFT from both the CME and the HR hybrid method, respectively. The model parameters and initial conditions are chosen so that the negativity problem may arise. For the NSRFT, when t is small, the two methods are close enough. When $t \in [1, 10]$, the HR hybrid method has the first slow reaction firing earlier than the CME. In this time interval, the mismatch between the two methods comes from the error of the hybrid method, rather than the negativity problem. For the SSRFT, the CDFs of two methods have an intersection at around $t^* = 2$, where before this point the HR hybrid method tends to fire the second slow reaction later than the CME. This difference shows the impact of the negativity problem. After the first slow reaction, x_2 becomes negative, and the system needs extra time to recover, which causes a delay for the SSRFT in the HR hybrid method.



(a) NSRFT (b) SSRFT

Figure 4.2: Cumulative probability distributions of NSRFT and SSRFT in the linear system (4.1) of the CME and the HR hybrid method. Parameters used in this example are: $f_1 = b_1 = k_c = 1$. The initial condition is $m_1 = 2, m_2 = 0$.

For comparison, we calculate the relative error

$$e_r = \frac{|T_c - T_h|}{T_c}, \tag{4.17}$$

where T_c and T_h are the second slow reaction firing times of the CME and the HR hybrid method, respectively. Here we sample the mean SSRFT as T_c and T_h, i.e., $P(T_2 < T_c) = 0.5$, $P(T_2 < T_h) = 0.5$.

When we increase b_1 from 1 to 10 and set the initial condition as $m_1 = m, m_2 = 0$, then S_2 has a low population and a great chance to become negative when the slow reaction fires. In Fig 4.3, the acceptable region for $e_r < 0.01$ of the SSRFT is surprisingly larger than that of the NSRFT. But this can be well explained. First, the NSRFT of the HR hybrid method is faster than that of the CME from the previous work [28] and the example in Fig. 4.2a. If there is no negativity problem, then the SSRFT of the hybrid method should be even faster than that of the CME as error accumulates over two slow reactions. However, when there is a negative species, e.g., S_2 in this case, the system slows down to recover from its negative state. The negativity recovery time, to some extent, reduces the numerical error of the hybrid method in the linear chain system. This runs like a way of coordination between two subsystems: when a species becomes negative caused by a slow reaction, i.e., the slow subsystem runs faster than expected, then the slow subsystem has to wait longer for the next slow reaction to fire again.

For the Zero-Reaction and Zero-Time remedies, though they have a smaller acceptable parameter space compared with the HR hybrid method, they still have a larger parameter region than the NSRFT. Note that in the linear chain system, the Zero-Time rule is similar to the Zero-Reaction rule as G'_f is the fast subsystem and time τ, ρ are comparably small, but they are different for other cases, such as when G'_f is different from the fast subsystem

or when there are more slow reactions in the slow subsystems. For the Zero-Population remedy, the acceptable parameter space is only half of the other methods. All three remedies converge to the contour line of the HR hybrid method when $m \approx 11$, satisfying one of the nonnegative requirements (4.16). On the other hand, when the total population is large (or $m \geq \frac{f_1+b_1}{f_1}$), a negative population appears in a shorter time window, whereas Ref. [28] has demonstrated that the error of the HR hybrid method becomes smaller with larger populations. This leads us to a conjecture regarding the HR hybrid method for the linear chain reaction system: the error caused by negative populations is negligible compared to the original error of the hybrid method. In other words, the negativity effect is substantial only when the method is already problematic in accuracy. Sometimes it acts as a positive sign to reduce the numerical error of the HR hybrid method.

For general cases that include both negative and nonnegative situations, we calculate the mean relative error based on all possible initial conditions $\vec{x}_0$, which follows the steady state distribution of the fast subsystem subject to f_1 and b_1. Fig. 4.4 illustrates the acceptable system parameter region ($e_r < 0.01$) for $b_1 = 1$ and $b_1 = 10$. In both cases, the SSRFT keeps the same pattern but with improved accuracy horizontally resulting from the negativity phenomenon and decreased accuracy vertically due to the accumulative method error. Since nonnegative situations occur much more frequently than negative situations (where the initial condition must be either $m_2 = 0$ or $m < 11$), the three proposed remedies do not make a difference in the acceptable region of the HR hybrid method.

A larger system ($n = 10$)

If we increase the length of the linear chain system to $n = 10$, it is hard to calculate the derived formulas (4.4,4.9) for the second exit time. Instead, we run simulations of each method and collect samples for the NSRFT and the SSRFT. For each pair of (m, k_c),

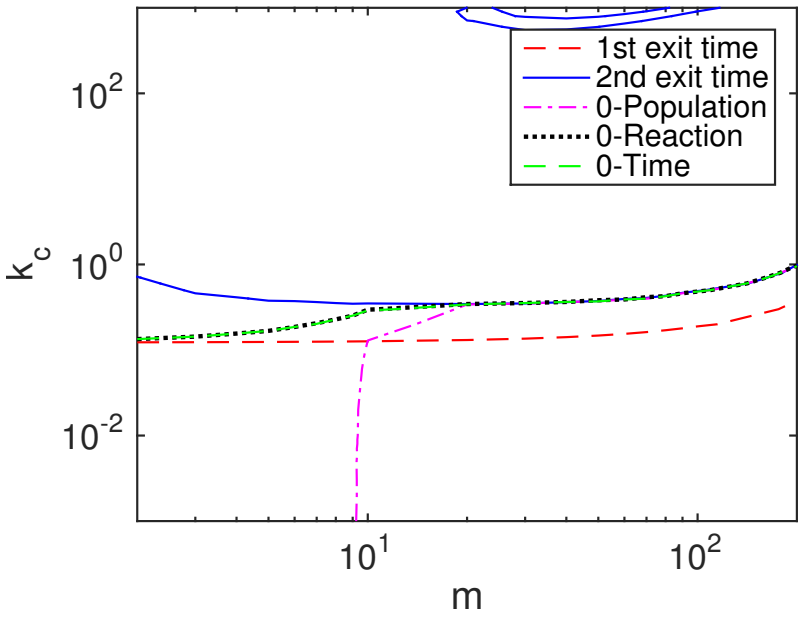
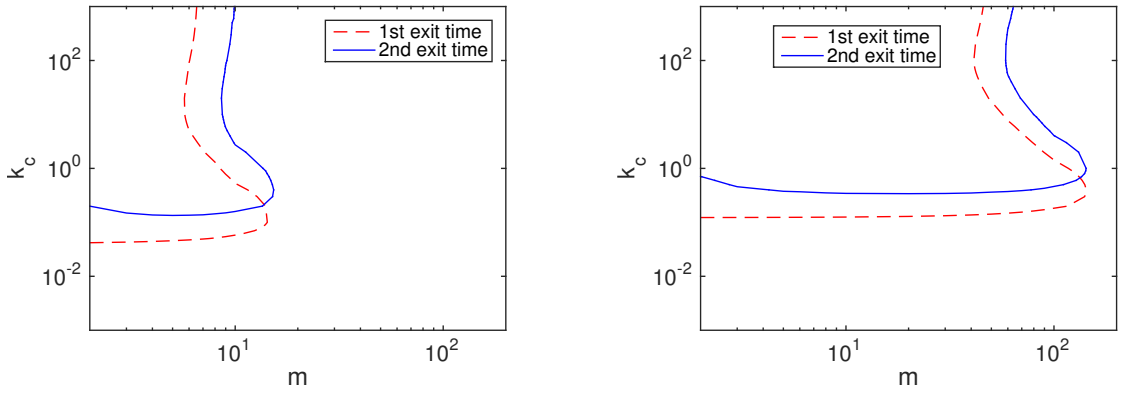


Figure 4.3: Contour plot of relative error e_r in the linear system (4.1) with parameters $f_1 = 1, b_1 = 10$. The initial condition is set to $m_1 = m, m_2 = 0$. Regions below each line have a relative error less than 1%. For the Zero-Population rule, the bottom right region is the acceptable parameter space.



(a) $b_9 = 1$ (b) $b_9 = 10$

Figure 4.4: Contour plot of the average relative error $e_r = 0.01$ in the linear system (4.1) with different k_c and m values. The remaining parameter is $f_1 = 1$. Acceptable parameter pairs for the HR hybrid method can be chosen from the bottom and right regions.

the reported NSRFT and SSRFT are the mean values from one million simulation results. Consider that all the contour plots in this subsection are processed to make the outline smooth.

First look at one negative case where the initial condition is $m_1 = m, m_i = 0$ $(i = 2, 3, ..., 10)$. In Fig. 4.5, the shape of the acceptable region ($e_r < 0.01$) is similar to the case when $n = 2$. For the Zero-Reaction rule, the accuracy is similar to that of the original HR hybrid method, except for the top right region; while the Zero-Population rule is only accurate for large m and small k_c. So the observation that negativity does not influence the accuracy still holds for large linear chain systems, if the Zero-Reaction rule is applied. We do not consider the Zero-Time rule which is much less efficient than the other two remedies in this larger system.

For general cases, we randomly sample initial conditions for each (m, k_c) pair. And the probability distribution of $\vec{x}_0$ satisfies the steady state of the reversible reactions controlled by f_i and b_i. Fig. 4.6 exhibits the same pattern as Fig. 4.4 for both cases $b_9 = 1$ and $b_9 = 10$. The acceptable parameter space for the second exit time is smaller when $n = 10$, which is similar to the first exit time discussed in Ref. [28]. Thus, for large linear chain systems, the negativity problem is insignificant for the hybrid method.

4.3.2 Numerical experiments

The previous subsection studied the accuracy based on the first and the second slow reaction times. In this section, we apply the HR hybrid method and three remedies to different systems and compare statistics.

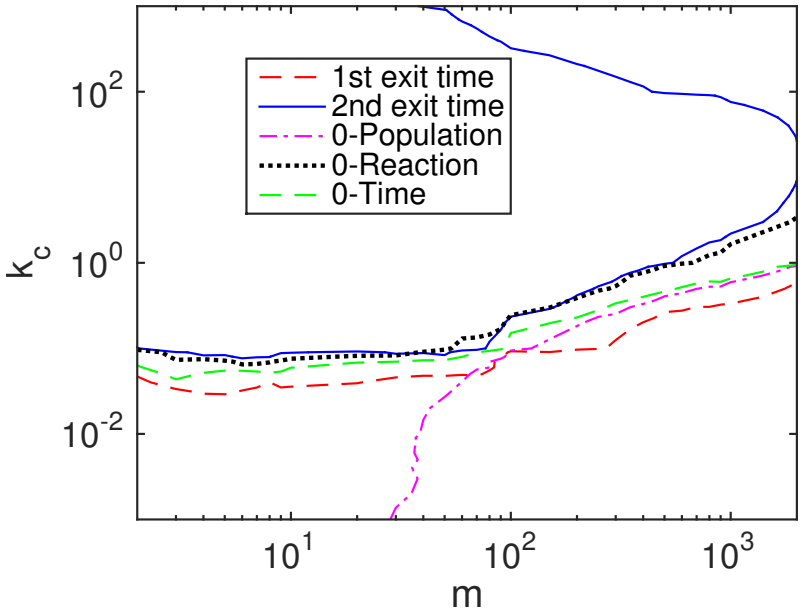
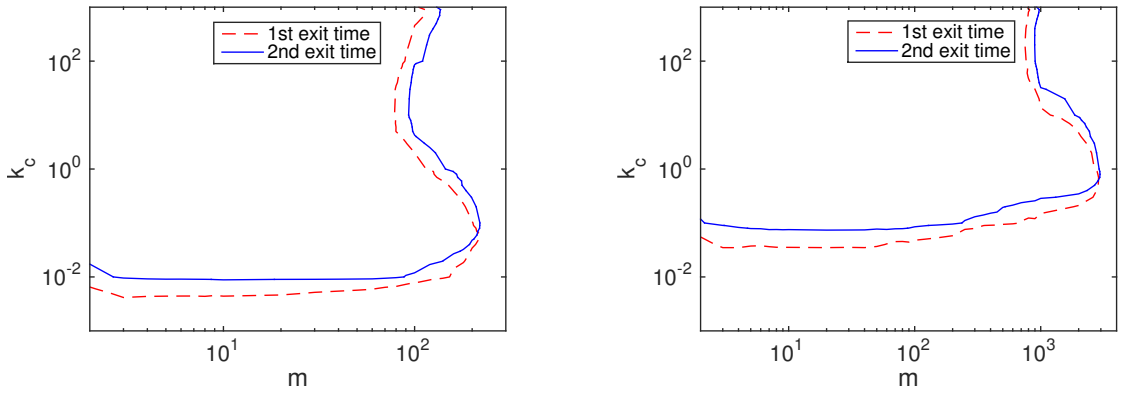


Figure 4.5: Contour plot of relative error of e_r in the linear chain system (4.2) with parameters $f_i = b_i = k_c = 1, b_9 = 10$. The chain length is $n = 10$ and the initial condition is set to $m_1 = m, m_i = 0$ $(i = 2, 3, ..., 10)$. Regions below each line have a relative error less than 1%. Note that the HR hybrid method has an extra top right region of acceptable parameter pairs.



(a) $b_9 = 1$ (b) $b_9 = 10$

Figure 4.6: Contour plot of the average relative error $e_r = 0.01$ in the linear chain system (4.2) with different k_c and m values. The chain length is $n = 10$. Acceptable parameter pairs for the HR hybrid method can be chosen from the bottom and right parts.

A closed linear chain system

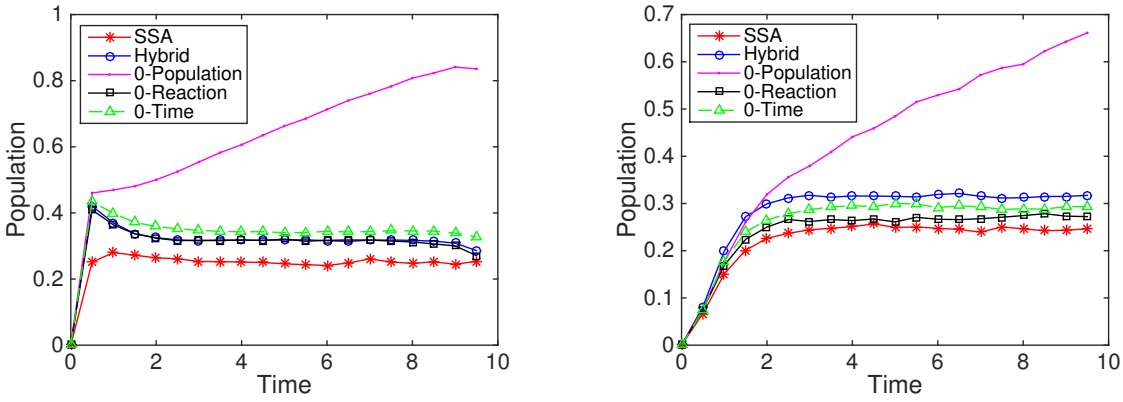
The first example shown below is a similar linear chain system with an extra reaction $S_3 \xrightarrow{k_s} S_1$ that forms a closed system. We divide the system into two parts, the ODE system contains the reversible reaction, while the SSA system takes the remaining reactions involving S_3.

$$S_1 \underset{b_1}{\overset{f_1}{\rightleftharpoons}} S_2 \xrightarrow{k_c} S_3 \xrightarrow{k_s} S_1. \tag{4.18}$$

We choose model parameters and initial conditions so that negative populations occur frequently in the system. Fig. 4.8 shows the average evolution of species S_2 and S_3 from different simulation methods and rules over $10,000$ simulations. Under the Zero-Population rule, the populations of S_2 and S_3 keep increasing with time while the other methods reach a steady state. The evolution from the Zero-Reaction rule is slightly closer to the results of SSA than the other methods. We then look at the final distributions of species S_2 and S_3 based on $10,000$ simulations, shown in Fig. 4.7. In system (4.18), x_2 is negative for 30% of the time if using the original HR hybrid method. With the Zero-Time rule, x_2 is always nonnegative, while the Zero-Reaction rule does not change the distribution of S_2 much. For the final distribution of S_2, though the differences between methods are fairly close ($< 10\%$), the Zero-Reaction rule also works better than others. The results from the Zero-Population rule are much different from the SSA results, and the final distribution will shift further to the right if we run the system longer.

The above example demonstrates that the system can finally recover from negative populations without using any remedies. To test an extreme case, suppose there is another species Y highly sensitive to S_2, shown below

$$S_1 \underset{b_1}{\overset{f_1}{\rightleftharpoons}} S_2 \xrightarrow{k_c} S_3, \quad \emptyset \xrightarrow{k_1[S_2]} Y \xrightarrow{k_2} \emptyset \tag{4.19}$$



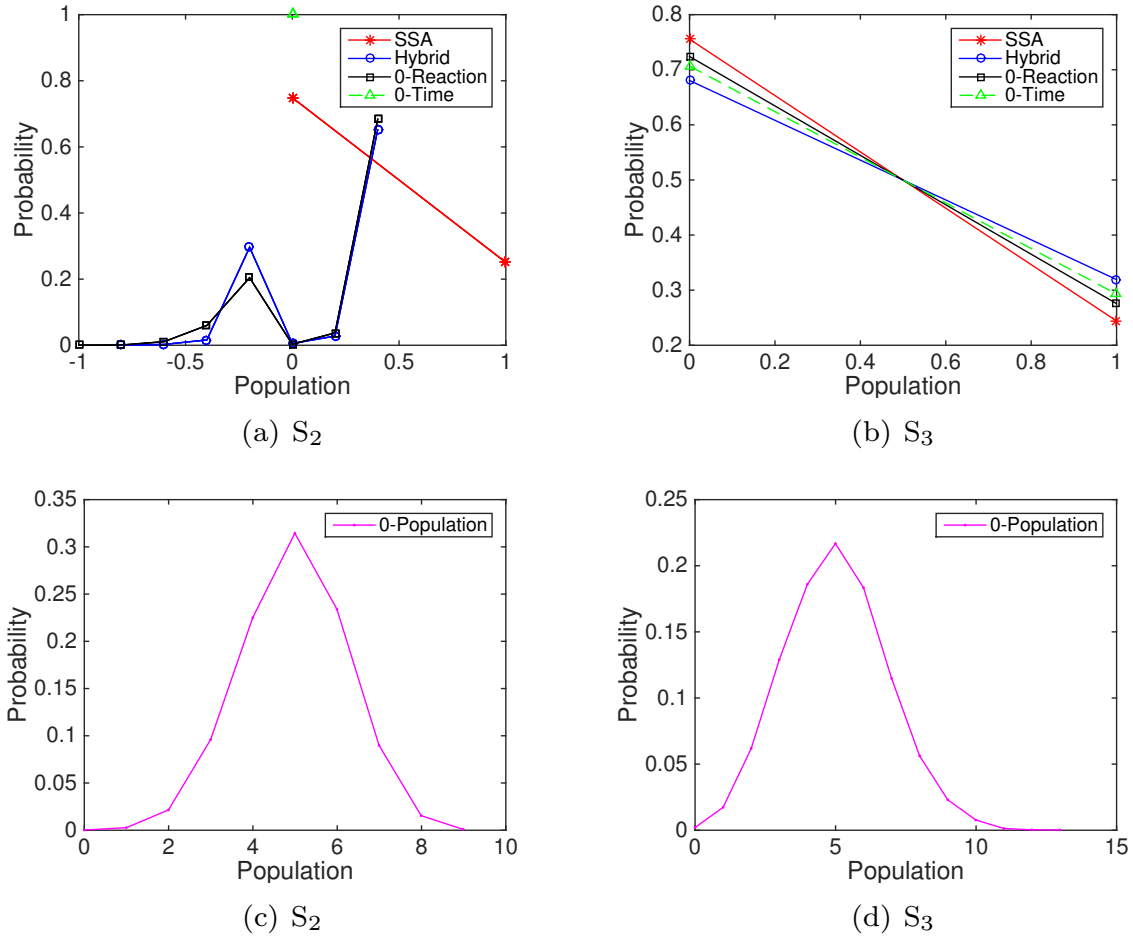
(a) S_2 (b) S_3

Figure 4.7: Final distributions of species S_2 and S_3 in the closed linear system (4.18) from the SSA, the HR hybrid method, and three remedies (Zero-Population, Zero-Reaction, and Zero-Time) based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 1$ and the remaining species populations are zero.

The first part is a simple linear chain system with $n = 2$. A particle can exit the reversible chain system through S_2 with reaction rate k_c. The other part is a birth and death process of species Y. S_2 acts as the enzyme activating the synthesis of Y. Following the same partition strategy used above, we isolate the irreversible reaction in a slow subsystem as both the rate constant k_c and quantity of S_2 are small. All the remaining reactions are put into a fast subsystem.

By setting $f_1/b_1 = 0.1$, S_2 maintains a low population and has a high chance to become negative when the slow reaction fires. Once $x_2 < 0$, the slow reaction propensity is negative, which ensures that the slow reaction will not fire until x_2 goes back to a positive value in the fast subsystem. So the negative population of S_2 has no effect on S_3 in this case. But for Y, it can provoke large fluctuations especially when $k_1[S_2] \gg k_2$. Fig. 4.9 shows one simulation result of S_2 and Y. When S_2 first drops to -0.7, Y is also driven negative and then becomes positive after one time unit, while under the Zero-Reaction rule, Y is always nonnegative. During the recovery period of Y, if there is another species Z dependent on Y and a species

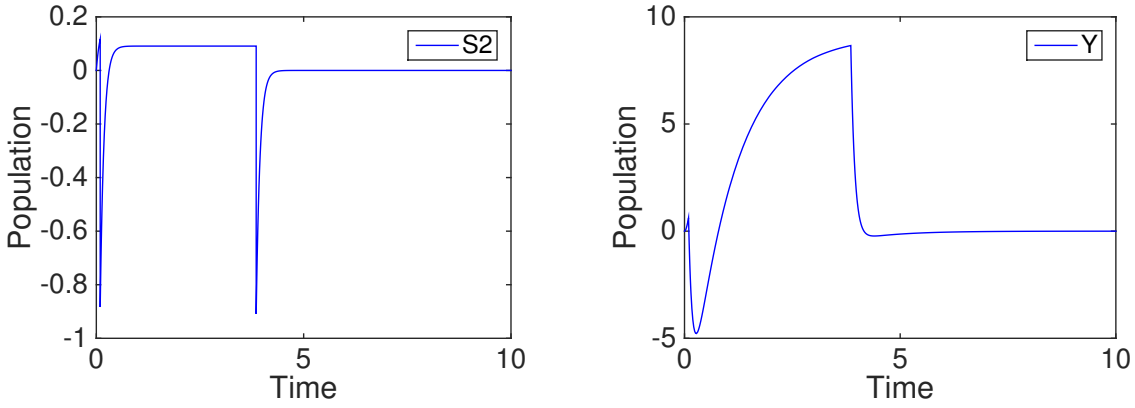
A dependent on Z, then Y may incur a cascade of negativity which significantly increases the simulation error. For situations where negative reactants heavily affect other species, a remedy is definitely needed to prevent a simulation failure.



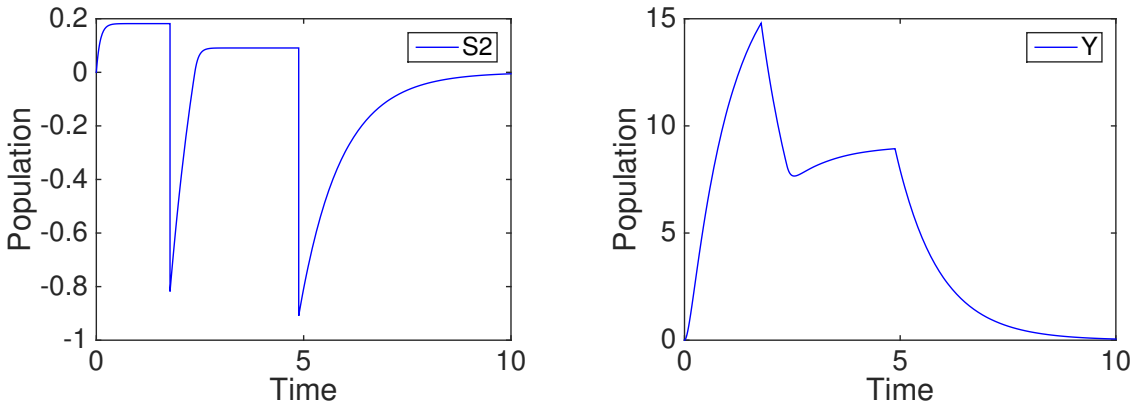
(a) S_2 (b) S_3

(c) S_2 (d) S_3

Figure 4.8: Evolution of species S_2 and S_3 in the closed linear system (4.18) from the SSA, the HR hybrid method, and three remedies (Zero-Population, Zero-Reaction, and Zero-Time) based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 1$ and the remaining species populations are zero.



(a) HR hybrid method



(b) Zero-Reaction remedy

Figure 4.9: Population trajectories of S_2 and Y in system (4.19) from one simulation of the HR hybrid method and the Zero-Reaction remedy The parameters are $f_1 = 1, b_1 = 10, k_c = 1, k_1 = 100, k_2 = 1$. The initial condition is $m_1 = 2$ and the remaining species populations are zero.

A closed nonlinear system

From the previous studies, we found that the HR hybrid method works fine for linear systems even with a high frequency of negative populations. Here we want to examine the effect of negative values on nonlinear systems, e.g., bimolecular reactions. Slightly modifying reactions involving S_2 into bimolecular reactions in system (4.18), we generate a new nonlinear

system shown below.

$$\mathrm{S}_1 \underset{b_1}{\overset{f_1}{\rightleftharpoons}} 2\mathrm{S}_2 \xrightarrow{k_c} \mathrm{S}_3 \xrightarrow{k_s} \mathrm{S}_1. \tag{4.20}$$

Similarly, we partition the system into groups: the fast group containing the reversible reactions and the slow group containing the remaining reactions. For the bimolecular reaction $\mathrm{S}_2 + \mathrm{S}_2 \xrightarrow{k_c} \mathrm{S}_3$ in the slow subsystem, the propensity is $a_{bi} = k_c x_2(x_2 - 1)$. When $x_2 < 0$, a_{bi} is positive which can potentially cause the reaction to fire and further decrease the value of x_2. Thus, the SSA system becomes unstable when $x_2 < 0$.

For the fast ODE system, we have

$$\begin{aligned}\frac{dx_1}{dt} &= -f_1 x_1 + b_1 x_2^2 \\ \frac{dx_2}{dt} &= 2f_1 x_1 - 2b_1 x_2^2\end{aligned}$$

The Jacobian matrix is

$$J = \begin{bmatrix} -f_1 & 2b_1 x_2 \\ 2f_1 & -4b_1 x_2 \end{bmatrix}$$

The determinant is

$$\begin{aligned}|\lambda I - J| &= \begin{vmatrix} \lambda + f_1 & -2b_1 x_2 \\ -2f_1 & \lambda + 4b_1 x_2 \end{vmatrix} \\ &= \lambda(\lambda + f_1 + 4b_1 x_2)\end{aligned}$$

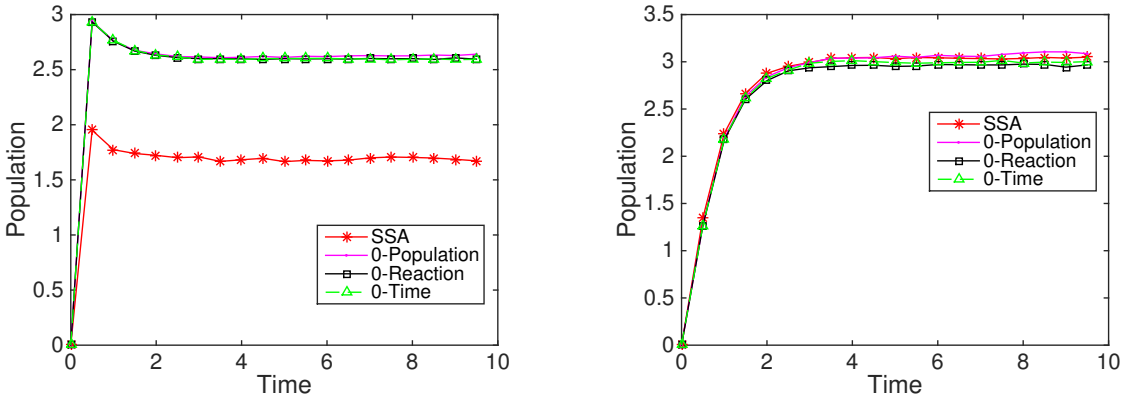
The two eigenvalues of the ODE system are $\lambda_1 = 0, \lambda_2 = -f_1 - 4b_1 x_2$. For $x_2 > 0$, $\lambda_2 < 0$, so the fast system is stable. But in the HR hybrid method, x_2 could be negative under certain

conditions. Let $\frac{dx_2}{dt} = 2f_1x_1 - 2b_1x_2^2 = 0$, species S_2 has the two equilibrium points:

$$x_2^* = -a \pm \sqrt{a^2 + am}, \quad a = \frac{f_1}{4b_1} \tag{4.21}$$

where m is the initial total population. One equilibrium point is positive, thus is a stable point in the system. But for the other point, since $x_2^* = -a - \sqrt{a^2 + am} < -2a < -\frac{f_1}{4b_1}$, $\lambda_2 > 0$, thus is an unstable point. So the HR hybrid method fails in this nonlinear system when the population of S_2 gets smaller than $-\frac{f_1}{4b_1}$ by chance.

The above analysis is consistent with our simulation results. In our experiments, a simulation is considered a failure when the ODE solver is unable to meet integration tolerances with the smallest step length, or when one of the species' population reaches an extremely abnormal value, for example if a species' population becomes abnormally large (e.g., 1000) or below the negative value of the total population ($-m$). With an initial condition $m_1 = 10$ and $m_2 = 0$, even if x_2 has a probability less than 1% to be negative (see Fig. 4.11a), the system still suffers a significant error and breaks down after certain simulation time when using the original HR hybrid method. Particularly, 191 simulations of the original HR hybrid method failed among the total $10,000$ trials (each trial runs from time $t = 0$ to $t = 10$). In Fig. 4.10, while the evolution of S_3 from the three rules is pretty close to that of the SSA, there is an approximate one molecule difference in the S_2 population between the remedy rules and the SSA, which mainly comes from the method error rather than the influence of negative value of x_2. The final distributions of species S_3 from the Zero-Reaction and Zero-Time rules are close to the bell shape of the SSA results. Although the Zero-Population remedy did not fail in the simulation, the results are quite erroneous. Note that in Fig. 4.10, the population of S_2 and S_3 from the Zero-Population rule does increase slowly with time. If we run the system to a much larger time (e.g., $t = 1000$), S_3 can reach 30, as shown in the final distribution of Fig. 4.11d.

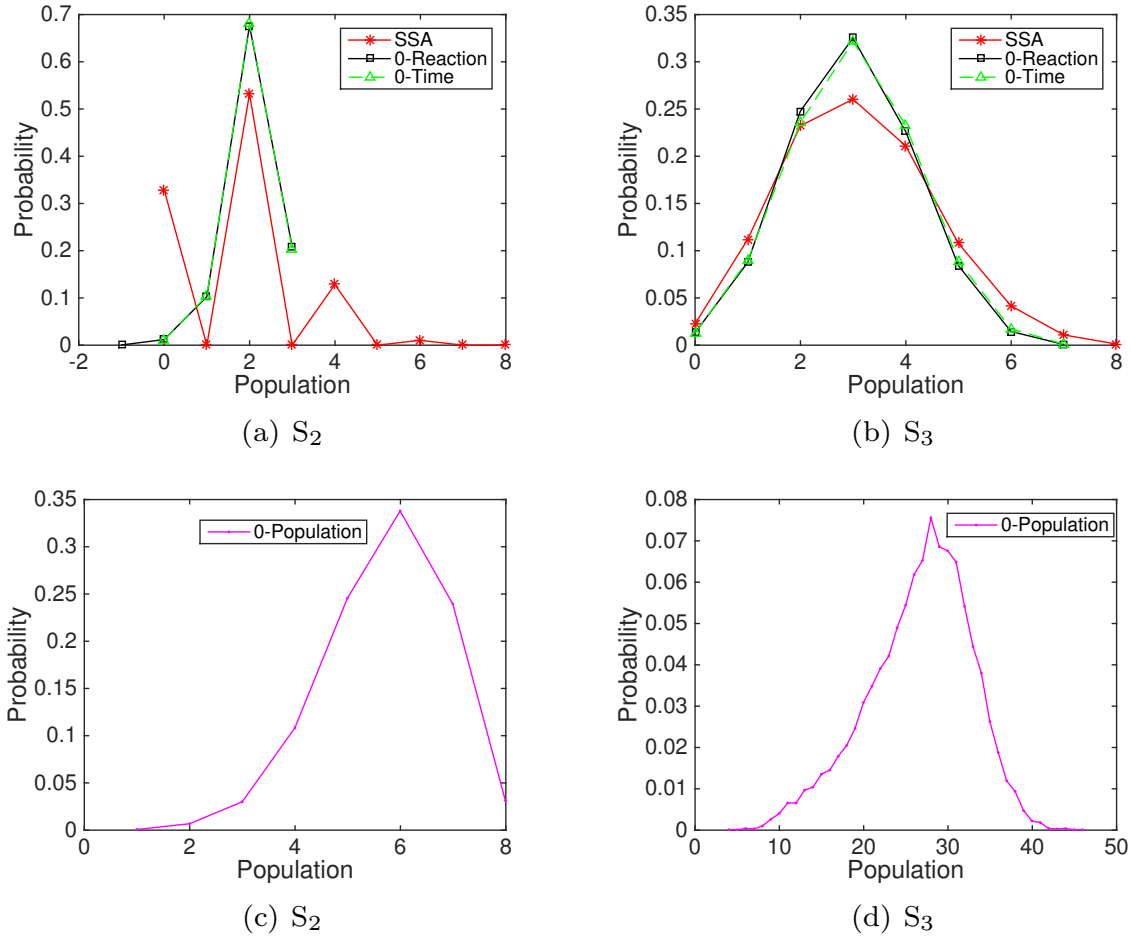


(a) S_2 (b) S_3

Figure 4.10: Evolution of species S_2 and S_3 in the nonlinear system (4.20) from the SSA and three remedies (Zero-Population, Zero-Reaction, and Zero-Time) based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 10$ and the remaining species populations are zero.

When decreasing the total population to $m = 3$, only the Zero-Reaction rule still works and generates stable results similar to the SSA except the approximate one molecule difference in S_2 population, see Fig. 4.12 and Fig. 4.13. The Zero-Time rule failed because the separate G'_f system (which is the fast subsystem in this case) is unstable. Among the $10,000$ simulations where the system was simulated from time $t = 0$ to $t = 10$, the original HR hybrid method failed in 2468 trials, while the Zero-Time rule failed in 1302 trials.

In general, the system stability will be affected if negative species are involved in nonlinear reactions where either the corresponding reacting terms in the ODE system or the corresponding propensities in the SSA system are still positive. Through the above comparison, the Zero-Reaction rule shows its ability to avoid the instability of nonlinear systems caused by negative values and at the same time keeps the accuracy of HR hybrid method. In application, the Zero-Reaction rule is easy and efficient to implement.



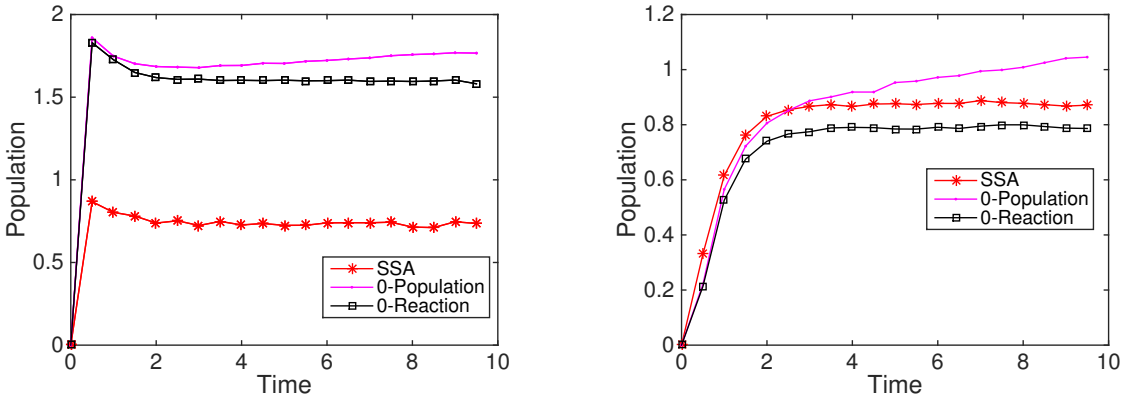
(a) S_2 (b) S_3

(c) S_2 (d) S_3

Figure 4.11: Final distributions of species S_2 and S_3 in the nonlinear system (4.20) from the SSA, three remedies (Zero-Population, Zero-Reaction, and Zero-Time) based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 10$ and the remaining species populations are zero.

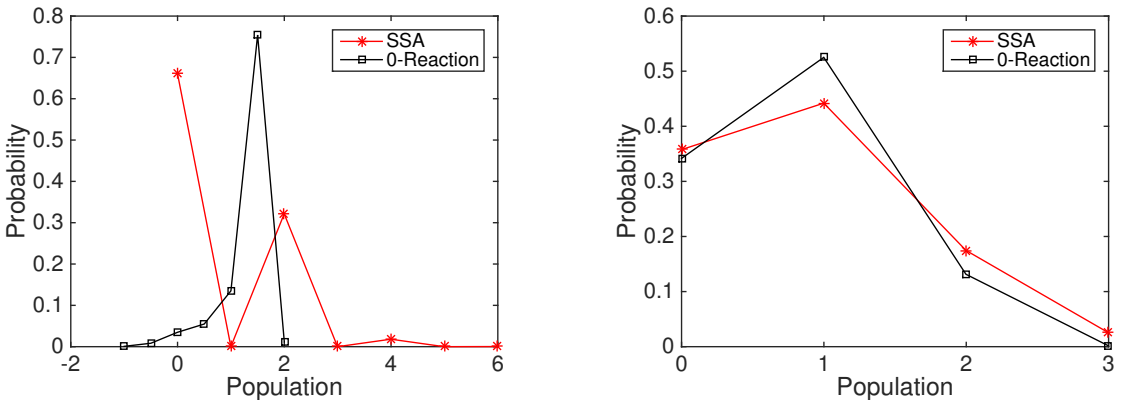
Caulobactor crescentus cell cycle model

Caulobactor crescentus is a bacteria that lives in freshwater like streams and lakes. It has an asymmetrical division that produces two morphologically different daughter cells, which makes it an important study organism for cell cycle modeling. Li et al. [78] studied the stochastic spatiotemporal model of a response-regulator network in the cell cycle. The stochastic model focused on the bistable switch of PleC that functioned as both kinase



(a) S_2 (b) S_3

Figure 4.12: Evolution of species S_2 and S_3 in the nonlinear system (4.20) from the SSA, the HR hybrid method, and three remedies (Zero-Population, Zero-Reaction, and Zero-Time) based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 3$ and the remaining species populations are zero.



(a) S_2 (b) S_3

Figure 4.13: Final distributions of species S_2 and S_3 in nonlinear system (4.20) from the SSA, the Zero-Reaction remedy based on $10,000$ simulations. The parameters are $f_1 = b_1 = k_c = k_s = 1$. The initial condition is $m_1 = 3$ and the remaining species populations are zero.

and phosphatase and successfully captured the viability of mutant cells. But the stochastic simulation took three days for a single run.

To improve the efficiency, we applied the HR hybrid method and compared different partitioning strategies. In the stochastic model, there are 141 reactions involving 45 species, in which eight proteins and their corresponding mRNAs are diffusive. The rod cell shape was modeled as $50 \times 1 \times 1$ cubics. In a test run, protein diffusion took 99.826% of the total number of reactions, putting it in the ODE system would greatly decrease the time cost. The catalytic reactions CtrA $\rightleftharpoons$ CtrAp took 0.148%. Compared to the diffusion of proteins, the diffusion of mRNAs only occupied 0.024%, while the remaining 140 reactions took 0.002%.

The computational cost of the hybrid method is proportional to the number of slow reaction firings, but is also affected by the size of the ODE subsystem. Based on the firing number of different reactions in the SSA simulation, we investigated three partitioning strategies as shown in Table 4.1. In Strategy I, only diffusion events of eight proteins are simulated by the ODE subsystem, the remaining events are put into the SSA subsystem. While Strategy II further partitions catalytic reactions of CtrA into the ODE subsystem, the size of the ODE subsystem does not change (reactants and products of the catalytic reaction are diffusive and already included in the ODE subsystem). But the average slow reaction firing time is one order of magnitude less than Strategy I, which decreases the time cost by approximately a factor of ten. Strategy III partitions the system by species type: mRNA reactions are all in the SSA subsystem and protein reactions are in the ODE subsystem. This strategy greatly reduces the probability of negativity problems. On the other hand, although Strategy III has the least interruption by slow reactions (every $6e^{-5}$ min), its size for the ODE subsystem increases to 50 (bin number)$\times$37 (types of species). This is quite a large ODE system, which imposes a high computational burden on the ODE solver. Overall, Strategy II is the most efficient partitioning strategy for the HR hybrid method in this *Caulobacter* cell cycle model,

the time cost for a single cell cycle simulation is significantly reduced to one hour from three days!

Table 4.1: Comparison of different partition strategies and complexities on the PleC model of the Caulobactor crescentus cell cycle.

		Strategy I	**Strategy II**	**Strategy III**
ODE	**Reactions**	Protein diffusion	Protein diffusion, catalytic reaction	Diffusion, reactions involving proteins
	System Size	50×8 equations	50×8 equations	50× 37 equations
SSA	**Reactions**	mRNA diffusion, all 141 reactions	mRNA diffusion, rest 140 reactions	mRNA diffusion, synthesis, degradation
	Firing Interval	$1e^{-6}$	$2e^{-5}$	$6e^{-5}$
	Time	9.5h	1h	4h

However, as mentioned in the second example in the introduction, the negativity problem appears when species density is low. Fig. 4.14 summarizes the total time of negativity state for each species during one cell cycle ($\sim$ 120 min). Protein DivKp has a negative value for almost 10% of a cycle period, followed by proteins CckA, DivL(free), CtrAp, and DivJ(free), which are negative for less than 0.1% of the total time. Fig. 4.15 shows the average population trajectories of four negative species from the SSA and the HR hybrid method using Strategy II. We can see that the hybrid method matches well with the SSA except for a slight difference in DivJ(free). All species with negative values have a period of a low population during the cell cycle. The scarce density of DivKp (nearly zero for the initial 30 min) results in a high occurrence of negative value. Yet the negativity problem in this model has no significant impact on simulation accuracy because the diffusion of proteins happens much faster than chemical reactions (at least one order of magnitude faster). Whenever a bin has a negative population resulting from a slow reaction firing, proteins in neighboring bins (with positive populations) quickly diffuse to the negative bin in the ODE system and make it positive before any chemical reaction happens. The HR hybrid method does not even need

a remedy rule for the negativity problem in this case. But in general, the Zero-Reaction rule is recommended since the added computational cost is minimal but the potential impacts of the negativity problem can be avoided.

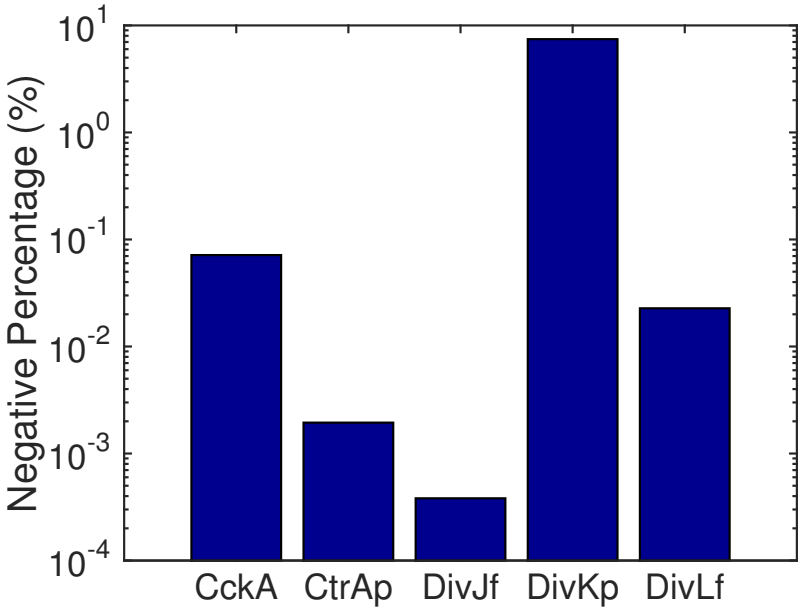


Figure 4.14: The percentage of the cell cycle time where species have negative populations.

4.4 Conclusions

This chapter presents an analysis on the negativity problem of the HR hybrid stochastic simulation algorithm. Based on the second slow reaction firing time, the error caused by negative populations is shown to be negligible compared to the approximation error of the method itself. In the linear chain system, the negativity phenomenon actually helps to increase the method's accuracy. But for nonlinear systems, negative values may lead to system failure. Three remedies for negativity are proposed and studied in the context of SSRFT where Zero-Time and Zero-Reaction rules have acceptable accuracy. Particularly,

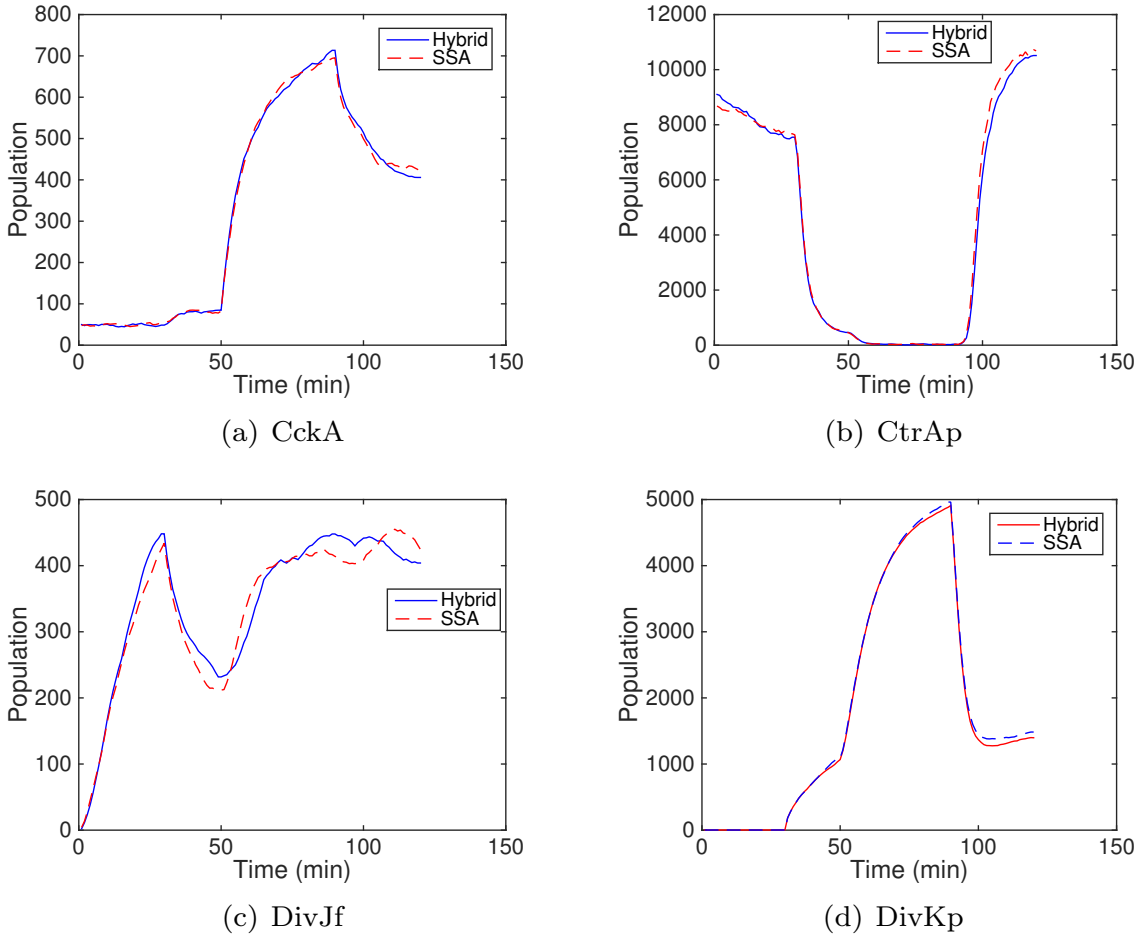


Figure 4.15: The mean population trajectories of negative species in the *Caulobacter* cell cycle model from over 48 simulations. Note that the shown population of each species at each time point are the summation of the population over 50 bins in the domain.

the Zero-Reaction remedy can handle both extreme negative cases and nonlinear systems whereas the other two methods may fail. Without any remedy for negative populations, the HR hybrid method may still be successfully applied to a real biological network and significantly improves the efficiency via an optimized partition strategy. Overall, we conclude that the negativity phenomenon does not influence the biochemical network unless the negative species are involved in nonlinear reactions that generate positive reacting terms or propensities. In general, the Zero-Reaction remedy is recommended due to easy implementation and

minimal additional computational cost.

Chapter 5

Quasi-Newton Stochastic Optimization Algorithm for Parameter Estimation of a Stochastic Model of the Budding Yeast Cell Cycle

5.1 Introduction

A fundamental challenge of molecular systems biology is to build accurate dynamical models of the molecular mechanisms underlying various aspects of cell physiology, e.g., cellular chemotaxis or the regulation of cell growth and division. Typically, these models are expressed in terms of differential equations, i.e., the models are 'deterministic', and their validity is assessed by comparison of model simulations to the observed (average) properties of large populations of cells responding to various experimental conditions. In recent years, however, cell biologists are increasingly able to measure the behavior and molecular constitution of single cells as they go about their business in space and time. As might be expected, the specific behavior of any given cell may be quite different than the average behavior of a

population of cells, reflecting molecular variability between cells. Regardless of the source(s) of such variability, which may be at the levels of DNA, mRNA, protein, and/or signaling molecules, deterministic models of the behavior must be converted into realistic stochastic models to deal with the variability of responses from one cell to another.

An important and difficult aspect of any modeling project is estimation of the kinetic constants ('model parameters') that appear in any dynamical model (deterministic or stochastic) of a molecular regulatory process. The parameters (e.g., rates of gene expression, rate constants for mRNA and protein degradation, rates of association and dissociation of molecular complexes, etc.) are estimated by comparison of model simulations to relevant experimental measurements of molecular turnover in cells. For deterministic models the problem is difficult enough, because any reasonably complete model will have dozens of molecular species and many dozens of undetermined parameters, but the available experimental data is often quite extensive, and there exist powerful algorithms for fitting deterministic simulations to experimental data points. For stochastic models the problem is considerably more difficult, because a stochastic model adds many more parameters to the deterministic model on which it is based, and the specific sorts of data required to estimate these 'stochastic' parameters is often difficult to obtain experimentally. Furthermore, stochastic simulations generate statistical distributions of observables, and these distributions must be compared to experimentally observed distributions, and the parameters estimated by optimization of an objective function that is a random variable. Algorithms for such stochastic optimization problems are still being developed and assessed.

This chapter presents results on optimization of the parameters in a stochastic model of cell cycle regulation in budding yeast. Section 5.2 describes the model briefly. Section 5.3 states the mathematical optimization problem precisely. Section 5.4 outlines a new quasi-Newton algorithm (QNSTOP) for stochastic optimization, and Section 5.5 presents the results of

using QNSTOP to fit the stochastic cell cycle model to observed distributions of cell cycle observables (mass at birth, cell cycle time, duration of G_1 phase). Section 5.6 discusses some biological implications of the results, and conclusions are drawn in Section 7.

5.2 Stochastic Cell Cycle Model

The cell cycle model used in this chapter was developed originally by Teeraphan Laomettachit and is described in full in his Ph.D. thesis [74]. The deterministic version of the model uses a set of nonlinear differential algebraic equations (DAEs) to track the temporal evolution of 26 variables (proteins governing progression through the budding yeast cell cycle). These equations involve 126 parameters (kinetic constants) that are estimated by fitting simulations of the model to the observed phenotypes of 119 budding yeast strains.

The initial determination of the 'best' parameter values was done 'by hand' as follows. Starting with an initial 'basal' parameter vector X_{basal}, simulate the sequence of cell cycle events in 'wild-type' cells growing in glucose and in galactose, making sure that the cells are viable under both conditions. Then simulate the phenotypes of 117 mutant strains of budding yeast growing in either glucose or galactose. Each mutant strain is characterized by a set of genetic changes (e.g., gene A is knocked out and gene B is overexpressed two-fold). The strain is simulated by appropriate changes to the basal parameter vector (e.g., the rate constant for synthesis of protein A from gene A is set to zero, and the rate constant for synthesis of protein B from gene B is set to twice the basal value). Each mutant strain has an observed phenotype: viable or inviable; if viable then there is some observed birth size relative to wild-type cells; if inviable then the cell is stuck at some particular stage of the cell cycle. The simulated phenotype of each mutant strain is compared to the observed phenotype, and the basal parameter vector is scored accordingly. Then the basal parameter

vector is modified, the simulations are repeated and rescored, and the process is repeated until no further improvement seems to be possible.

Surprisingly, despite the immensity of the parameter space, a good modeler can make significant improvements to the basal parameter vector by hand in a few weeks, and the process is necessary (from the modeler's point of view) in order to understand the vagaries of the model with respect to the experimental data. In the process, the modeler often makes slight 'tweaks' to the underlying molecular model (the DAEs) in order to get better agreement between the model and the mutant phenotypes.

Once the deterministic model (from [74]) was fitted as well as possible to the data set (the phenotypes of 110 of 119 strains correctly simulated), it was converted to a stochastic model in order to explore the observed variability of cell cycle progression among single cells (wild-type and mutant strains). The conversion was made in two steps. First, the dimensionless variables of the DAE model (call them $z_i(t)$, $i = 1, \ldots, 26$) had to be converted into numbers of molecules of species i per cell $= c_i * z_i(t)$, where c_i is the 'characteristic concentration' of species i. Then each of the differential equations of the system of DAEs was converted into a stochastic differential equation of the Langevin type by adding two random noise terms to the right hand side. The first noise term had the usual form of a birth-death process for the protein species, and the second term was designed specifically to model the effects of mRNA fluctuations on noisy protein expression; see Laomettachit's thesis [74]. These two steps introduced 52 new 'stochastic' parameters into the model: 22 characteristic concentrations, and 30 parameters describing the coupling between mRNA expression and protein synthesis.

Laomettachit estimated these stochastic parameters by hand, as well. From experimental estimates of the average numbers of protein molecules per cell for each cell cycle gene, he could estimate the 22 characteristic concentrations. From reasonable guesses about mRNA dynamics in budding yeast cells, he could estimate the 30 other parameters. These estimates

gave quite acceptable agreement with the limited amount of statistical data at his disposal for the distributions of cell cycle observables in populations of wild-type cells.

The Laomettachit model of the budding yeast cell cycle was further examined by Oguz et al. [95], who explored the utility of differential evolution (DE) as a tool for characterizing the parameter space of the model. These authors started from an intermediate stage of Laomettachit's search (a basal parameter vector that accounted correctly for the phenotypes of only 72 of 119 strains). They found that DE could quickly improve the score (i.e., the number of phenotypes correctly simulated) of the basal parameter vector, but could not improve on the score that Laomettachit achieved by hand. That is to say, 92.5% (110/119) seems to be about the best fit that Laomettachit's deterministic model can achieve. In a later publication, Oguz et al. [96] applied DE to the stochastic version of Laomettachit's model. They held the 126 deterministic parameters fixed at the values determined by the mutant phenotypes, and they estimated the 52 stochastic parameters by DE. The objective function in this case was constructed by comparing simulated values and observed values for the means and variances of certain cell-cycle observables: total cycle time and duration of G_1 phase of the cell cycle, for mother cells and daughter cells. The purpose of this chapter was not so much to estimate the stochastic parameters of the model as to use the parametrized model to study the synchronization of cell division in budding yeast populations by external perturbations; see [96] for details.

5.3 The Mathematical Problem

As explained in the previous section, stochastic models of the cell cycle are necessary to explain the observed variability in cell cycle progression among individual cells. Estimating the parameters in a stochastic cell cycle model is challenging, both mathematically and em-

pirically. Obtaining accurate and useful data from individual cells is difficult, and very little such data exists in the literature. Regardless of what criterion is minimized to estimate the model parameters, the mathematical problem is a stochastic optimization problem, meaning that the objective function $\theta(x)$ itself is a random variable. To further complicate matters, the random noise in the objective function is not additive, i.e., the objective function is not of the form (deterministic $\theta(x)$) + (random noise). The randomness is buried deep in the simulation model, and has no simple representation at the output level of the simulation model.

For a real colony of cells and a simulated colony, several properties (e.g., mass at birth m_B and duration of G_1 phase T_{G_1}) can be observed. It is common practice to compute statistics (e.g., mean, variance) of these observables and then to estimate the simulation model parameters by minimizing the difference (measured somehow) between the empirical colony's statistics and the simulated colony's statistics. For example, both Laomettachit [74] and Oguz et al. [96] approximated the scatter plot of the two-dimensional joint distribution of m_B versus T_{G_1} by a dogleg (continuous piecewise linear function with two line segments), and then estimated stochastic model parameters by matching the slopes of the line segments in the two (empirical and model predicted) doglegs. Matching these statistics is certainly a *necessary* condition for the correctness of the model, but such summary statistics do not capture all the available information. What one really wants to do, for example, is match the empirical colony's distribution of m_B with the simulated colony's distribution of m_B. Even better, match the distributions for all the observables simultaneously, or even match the joint distributions. The proposal here is to do exactly that—for both mother and daughter budding yeast cells, match the joint distributions of the pair (mass at birth, duration of of G_1 phase) from the empirical and simulated cell colonies.

Postponing until later the details of obtaining (approximations of) these distributions, let

$p(i)$ and $q(i)$ denote the probability mass (after discretization of the probability density) functions of the empirical and of the simulated colony's observable, respectively. There are several standard, well-justified ways to compare distributions. From an information theoretic perspective comes the Kullback-Leibler divergence

$$d_{\mathrm{KL}}(p,q) = \sum_i p(i) \log_2 \left(\frac{p(i)}{q(i)} \right),$$

which is nonnegative and zero if and only if $p = q$, but is not a metric. Another criterion from statistics is the Hellinger distance

$$d_H(p,q) = \left(\sum_i \left(\sqrt{p(i)} - \sqrt{q(i)} \right)^2 \right)^{1/2},$$

which is a metric. Depending on how discretization (one- or two-dimensional histograms) is done, some of the simulation probabilities $q(i)$ might be zero (a histogram interval or box has no points in it), which makes d_{KL} infinite. d_H is better behaved in such cases. Both d_{KL} and d_H were tried for this work, but only results for d_H are reported.

Let $X \in \mathbb{R}^n$ be the vector of parameters to be estimated in the stochastic cell cycle model. Let $p(i)$ and $q(i)$ be the probability mass functions of the observable (e.g., m_B or the pair (m_B, T_{G_1}) in the bin with index i) from the empirical cell colony and from the simulated cell colony, respectively. $p(i)$ is constant, but $q(i)$ is a random variable determined by a stochastic simulation. The objective function is the random variable

$$f(X) = d_H(p,q),$$

and the stochastic optimization problem to be solved is

$$\min_{L \leqq X \leqq U} f(X),$$

where $[L, U]$ is a box in $\mathbb{R}^n$ defining the feasible set (allowable values for the model parameters X).

The approach taken here, aptly described as simulation-based parameter estimation, has a long history in statistics, which is discussed, with historical references, in Castle's Ph.D. thesis [20]. QNSTOP is the name of a class of quasi-Newton methods originally developed by Castle [20] for stochastic optimization problems. Subsequent work modified the original algorithm significantly to produce a variant that was applicable to (deterministic) global optimization problems. Since these two variants had considerable overlap, they were combined into a single algorithm and computer code, also called QNSTOP, with two operating modes, global and stochastic, described in Amos et al. [5]. Computational experience since then and considerations of computational efficiency and numerical stability resulted in further significant changes, such as using a different quasi-Newton update rule for the first iteration, changing the update rules to avoid small denominators, and numerous other changes from [5], reflected in the detailed algorithm description and Fortran code in the supplementary files. The current version of the algorithm and computer code (cf. subroutine QNSTOPS in supplementary files), outlined in the next section, is used here.

5.4 Quasi-Newton Algorithm for Stochastic Optimization

QNSTOP is a class of quasi-Newton methods developed for stochastic optimization that can also be used for deterministic global optimization, with certain modifications. Deterministic global optimization and stochastic optimization are the "usage modes" of the class QNSTOP referred to in the algorithm summary below. For brevity, only the essential steps are outlined here. In iteration k, QNSTOP computes the gradient vector $\hat{g}_k$ and Hessian matrix $\hat{H}_k$ of a quadratic model

$$\widehat{m}_k(X - X_k) = \hat{f}_k + \hat{g}_k^T (X - X_k) + \frac{1}{2} (X - X_k)^T \hat{H}_k (X - X_k)$$

of the objective function f centered at X_k, where $\hat{f}_k$ is generally not $f(X_k)$. The next iterate is

$$X_{k+1} = \left(X_k - \left[\hat{H}_k + \mu_k W_k \right]^{-1} \hat{g}_k \right)_\Theta ,$$

where μ_k is the Lagrange multiplier of a trust region subproblem, W_k is a symmetric, positive definite scaling matrix, and $(\cdot)_\Theta$ denotes projection onto the feasible set $\Theta = [L, U]$.

To estimate the gradient, QNSTOP uses an ellipsoidal design region centered at the current iterate $X_k \in \mathbb{R}^n$. Let

$$W_\gamma = \left\{ W \in \mathbb{R}^{n \times n} : W = W^T, \det(W) = 1, \gamma^{-1} I_n \preceq W \preceq \gamma I_n \right\}$$

for some $\gamma \geq 1$ where I_n is the $n \times n$ identity matrix. The elements of the set W_γ are valid scaling matrices that control the shape of the ellipsoidal design regions with eccentricity

constrained by γ. Let the ellipsoidal design regions, with radius τ_k, be given by

$$E_k(\tau_k) = \left\{ X \in \mathbb{R}^n : (X - X_k)^T W_k (X - X_k) \leq \tau_k^2 \right\}$$

where $W_k \in W_\gamma$.

In each iteration, QNSTOP chooses a set of N uniformly sampled design sites $\{X_{k1}, \ldots, X_{kN}\} \subset E_k(\tau_k) \cap \Theta$. Let $Y_k = (y_{k1}, \ldots, y_{kN})^T$ denote the N-vector of responses modeled by the linear model $y_{ki} = \hat{f}_k + X_{ki}^T \hat{g}_k + \epsilon_{ki}$ where ϵ_{ki} accounts for lack of fit. $\hat{g}_k$ is then the least squares estimate of the linear model gradient.

Depending on the context, QNSTOP either constrains the Hessian matrix update to satisfy

$$-\eta I_n \preceq \hat{H}_k - \hat{H}_{k-1} \preceq \eta I_n$$

for some $\eta \geq 0$, using a variation of the SR1 (symmetric, rank one) quasi-Newton update, or uses the unconstrained BFGS quasi-Newton update

$$\hat{H}_k = \hat{H}_{k-1} - \frac{\hat{H}_{k-1} s_k s_k^T \hat{H}_{k-1}}{s_k^T \hat{H}_{k-1} s_k} + \frac{\nu_k \, \nu_k^T}{\nu_k^T s_k},$$

where $s_k = X_k - X_{k-1}$, $\nu_k = \hat{g}_k - \hat{g}_{k-1}$.

QNSTOP utilizes an ellipsoidal trust region concentric with the design region for controlling step length. In one usage mode, the trust region ellipsoid radius ρ_k is taken equal to the design ellipsoid radius τ_k, and the optimization problem

$$\min_{X \in E_k(\rho_k)} \; \hat{g}_k^T (X - X_k) + \frac{1}{2} (X - X_k)^T \hat{H}_k (X - X_k)$$

is solved for X_{k+1} and μ_k related by

$$X_{k+1} = X(\mu_k) = X_k - \left[\hat{H}_k + \mu_k W_k\right]^{-1} \hat{g}_k.$$

In another usage mode, μ_{k-1} is directly updated to μ_k, giving $X_{k+1} = X(\mu_k)$ as above. If necessary, X_{k+1} is projected back into the feasible set Θ.

Finally, the experimental design region $E_k(\tau_k)$ is updated to approximate a confidence set by updating the scaling matrix W_k. The updated scaling matrix is given by

$$W_{k+1} = \left(\hat{H}_k + \mu_k W_k\right)^T V_k^{-1} \left(\hat{H}_k + \mu_k W_k\right),$$

where V_k is the covariance matrix of $\nabla \widehat{m}_k(X_{k+1} - X_k)$. For numerical stability, W_{k+1} is constrained (by modifying its eigenvalues) to satisfy the constraints $\gamma^{-1} I_n \preceq W_{k+1} \preceq \gamma I_n$ and $\det(W_{k+1}) = 1$, so $W_\gamma \ni W_{k+1}$.

Algorithm summary: It is generally desirable to run QNSTOP from multiple start points, and the algorithm described below is repeated for each start point.

Step 0 (initialization): Given a function evaluation budget $\tilde{B}$ per start point and operating mode (choices of quasi-Newton update, ways of updating the ellipsoidal design region radii τ_k and ellipsoidal trust region radii ρ_k, etc.), set values for $\tau_0 > 0$, $\gamma \geq 1$, $\eta \geq 0$, N, X_0, $k := 0$, $W_0 := \hat{H}_0 := I_n$.

Step 1 (regression experiment): Depending on the usage mode, compute the design ellipsoid radius τ_k. Uniformly sample $\{X_{k1}, \ldots, X_{kN}\} \subset E_k(\tau_k) \cap \Theta$. Observe the response vector $Y_k = (y_{k1}, \ldots, y_{kN})^T$. Compute $\hat{g}_k$ by linear regression.

Step 2 (secant update): If $k > 0$, compute the model Hessian matrix $\hat{H}_k$ using either the BFGS or SR1 variant update, depending on the usage mode.

Step 3 (update iterate): Compute μ_k depending on the usage mode, solve $[\hat{H}_k+\mu_k W_k]s_k = -\hat{g}_k$ for the step s_k, and compute $X_{k+1} = \big(X_k + s_k\big)_\Theta$.

Step 4 (update subsequent design ellipsoid): Compute a new scaling matrix $W_{k+1} \in W_\gamma$.

Step 5: If $(k+2)(N+1)+1 < \tilde{B}$ then increment k by 1 and go to **Step 1**. Otherwise, the algorithm terminates. (f is also observed at each ellipsoid center X_k.)

For efficiency, a hardware specific parallel version of the portable serial code QNSTOPS (cf. supplementary files) was actually used. All QNSTOP parameters are optional arguments and default to reasonable values in the computer code, and extensive tuning is not generally necessary; the few nondefault values used are reported with the results later.

5.5 Numerical Results and Discussion

The budding yeast stochastic cell cycle model in [96], called 'Laomettachit's stochastic model' here, has 52 parameters that are exclusive to the stochastic aspects of the model, of which some are chosen to be equal to others, leaving 44 independent variables (parameters) to be determined by some mathematical procedure (here, solving a stochastic optimization problem). The parameter names follow a pattern: the species $\star$ is denoted by an index 1, 2, ..., 10, referring to species Cln3, Bck2, Cln2, CKI, Clb5, Clb2, Swi5, Cdc20, Pds1, and POLO, respectively. The parameters $k_{tr\star}$, $k_{dm\star}$, $m_{min\star}$ for species $\star$ are, respectively, translation rate, mRNA degradation rate, and minimum number of mRNA molecules. This accounts for 30 parameters. The remaining 22 parameters c_x, where x is the species name, are the characteristic concentrations of the above ten species and 12 other species: Whi5, SBF, Cdh1, APCP, Clb14, Net1, PPX, Esp1, Cdc15, Tem1, MEN, Mcm1. These characteristic concentrations are introduced to convert the dimensionless concentrations of the species in

the deterministic version of Laomettachit's model into numbers of molecules per cell for each species in the stochastic version of the model. Since some of the species in the model bind with each other to form stoichiometric complexes, the characteristic concentrations of such binding partners must be identical. Therefore, as in Oguz et al. [96], making the eight assignments $c_{\mathrm{SBF}} \equiv c_{\mathrm{Whi5}}$, $c_{\mathrm{Clb2}} \equiv c_{\mathrm{Clb5}} \equiv c_{\mathrm{CKI}}$, $c_{\mathrm{APCP}} \equiv c_{\mathrm{Cdc20}}$, $c_{\mathrm{Net1}} \equiv c_{\mathrm{Cdc14}}$, $c_{\mathrm{Esp1}} \equiv c_{\mathrm{Pds1}}$, and $c_{\mathrm{MEN}} \equiv c_{\mathrm{Tem1}} \equiv c_{\mathrm{Cdc15}}$ leaves 44 independent parameters defining the vector X.

Table 5.1 lists the 52 stochastic parameters in Laomettachit's model. The nominal vector X_0 defines the search box $[L, U]$, where the bounding interval for the ith component $(X)_i$ of X is $\left[(1/\varphi_i)(X_0)_i, \varphi_i(X_0)_i\right]$ and each factor φ_i is either 2 or 5. The 'best [96] vector' is the best estimate of the parameter vector found by [96] using differential evolution.

Table 5.1: List of parameters in stochastic budding yeast cell cycle model.

Parameter	Nominal value	Best [96] value	$[L, U]$
k_{tr1}	0.22	0.3870	[0.044,1.1]
k_{dm1}	0.7	2.9459	[0.14,3.5]
m_{min1}	1.0	5.0	[0.2,5.0]
k_{tr2}	0.22	0.6166	[0.044,1.1]
k_{dm2}	0.7	0.6033	[0.14,3.5]
m_{min2}	4.0	17.0	[0.8,20.0]
k_{tr3}	0.22	0.0761	[0.044,1.1]
k_{dm3}	0.7	2.9502	[0.14,3.5]
m_{min3}	1.0	2.0	[0.2,5.0]
k_{tr4}	0.22	0.2024	[0.044,1.1]
k_{dm4}	0.7	1.4652	[0.14,3.5]

Continued on next page

Table 5.1 – *Continued from previous page*

Parameter	Nominal value	Best [96] value	$[L, U]$
m_{min4}	4.0	1.0	[0.8,20.0]
k_{tr5}	0.22	0.6878	[0.044,1.1]
k_{dm5}	0.7	0.1975	[0.14,3.5]
m_{min5}	4.0	8.0	[0.8,20.0]
k_{tr6}	0.22	0.6974	[0.044,1.1]
k_{dm6}	0.7	1.6668	[0.14,3.5]
k_{min6}	4.0	15.0	[0.8,20.0]
k_{tr7}	0.22	0.8867	[0.044,1.1]
k_{dm7}	0.7	2.4182	[0.14,3.5]
m_{min7}	4.0	16.0	[0.8,20.0]
k_{tr8}	0.22	0.7344	[0.044,1.1]
k_{dm8}	0.7	3.4411	[0.14,3.5]
m_{min8}	4.0	6.0	[0.8,20.0]
k_{tr9}	0.22	0.6737	[0.044,1.1]
k_{dm9}	0.7	1.2706	[0.14,3.5]
m_{min9}	4.0	9.0	[0.8,20.0]
k_{tr10}	0.22	0.4258	[0.044,1.1]
k_{dm10}	0.7	0.1469	[0.14,3.5]
m_{min10}	4.0	5.0	[0.8,20.0]
c_{Cln3}	10.0	19.0957	[5.0,20.0]
c_{Bck2}	10.0	16.3317	[5.0,20.0]
c_{Whi5}	22.0	21.8688	[11.0,44.0]

Continued on next page

Table 5.1 – *Continued from previous page*

Parameter	Nominal value	Best [96] value	$[L, U]$
c_{SBF}	22.0	21.8688	[11.0,44.0]
c_{Cln2}	45.0	84.2260	[22.5,90.0]
c_{CKI}	80.0	101.9969	[40.0,160.0]
c_{Clb5}	80.0	101.9969	[40.0,160.0]
c_{Clb2}	80.0	101.9969	[40.0,160.0]
c_{Swi5}	57.5	50.4561	[28.75,115.0]
c_{Cdc20}	100.0	93.1338	[50.0,200.0]
c_{Cdh1}	100.0	59.4664	[50.0,200.0]
c_{APCP}	100.0	93.1338	[50.0,200.0]
c_{Cdc14}	14.0	20.2049	[7.0,28.0]
c_{Net1}	14.0	20.2049	[7.0,28.0]
c_{PPX}	100.0	81.0649	[50.0,200.0]
c_{Pds1}	3.3	2.3993	[1.65,6.6]
c_{Esp1}	3.3	2.3993	[1.65,6.6]
c_{Cdc15}	8.0	8.7958	[4.0,16.0]
c_{Tem1}	8.0	8.7958	[4.0,16.0]
c_{MEN}	8.0	8.7958	[4.0,16.0]
c_{POLO}	100.0	155.2614	[50.0,200.0]
c_{Mcm1}	100.0	183.1687	[50.0,200.0]

The empirical data from Di Talia et al. [35] includes mass at birth, duration of G_1 phase,

and cell cycle time of both mother and daughter budding yeast cells. Using the Hellinger distance to measure the difference between the empirical data distribution and the simulated data distribution requires approximating the continuous distributions by (one- or two-dimensional) histograms. For example, Fig. 5.1 shows the histogram box boundaries for the joint distribution of the (scaled) pair (mass at birth, duration of G_1 phase) for mother cells. The strategy is to define rectangles (or intervals in one dimension) that roughly evenly divide the empirical data points and such that every rectangle (or interval) contains some data points. The 122 data points yield 17 bins (divided by black lines in Fig. 5.1). The particular discretization has no effect on the optimization algorithm. Fig. 5.2 shows the one-dimensional histogram for the empirical data of daughter cell cycle times. Here the 97 data points are divided into 10 bins. Given the sparsity and accuracy of the data, and the stated goal for how to discretize the continuous distributions, the result is a histogram shape as in Fig. 5.2.

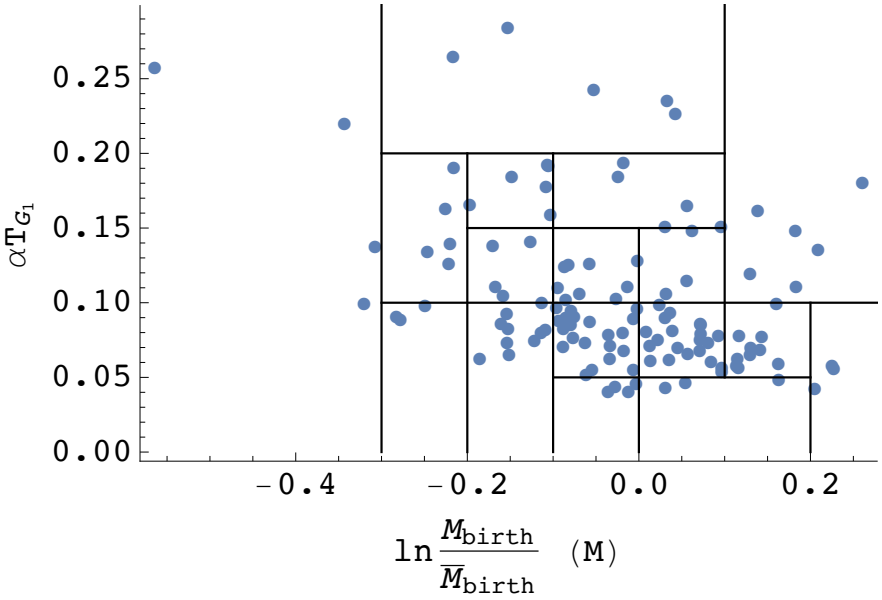


Figure 5.1: Discretization for empirical correlations of mass at birth and scaled duration of G_1 phase of mother cells. The x-axis is ln (individual mass/mean mass), where the mean is of all mother cells.

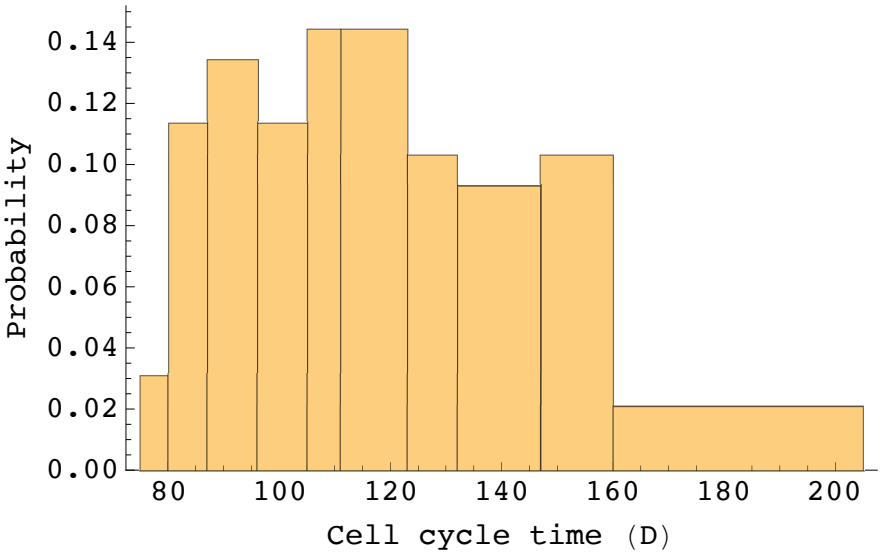


Figure 5.2: Discretization for empirical daughter cell cycle time.

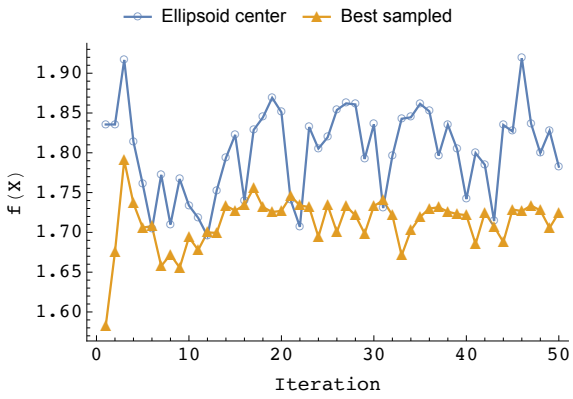
Altogether there are eight distributions being matched (and eight Hellinger distances $d_{H,i}$, $i = 1, \ldots, 8$): joint pair (mass at birth, duration of G_1 phase) for mothers (17 boxes), joint pair (mass at birth, duration of G_1 phase) for daughters (18 boxes), mass at birth for mothers (12 intervals), mass at birth for daughters (10 intervals), G_1 duration for mothers (9 intervals), G_1 duration for daughters (11 intervals), cell cycle time for mothers (10 intervals), and cell cycle time for daughters (10 intervals). There are thus a total of 97 discrete probabilities being matched (one for each bin/box/interval) using 44 degrees of freedom (the independent stochastic cell cycle model parameters X), which is a well-posed problem. The objective function is

$$f(X) = \sum_{i=1}^{8} d_{H,i}(p, q),$$

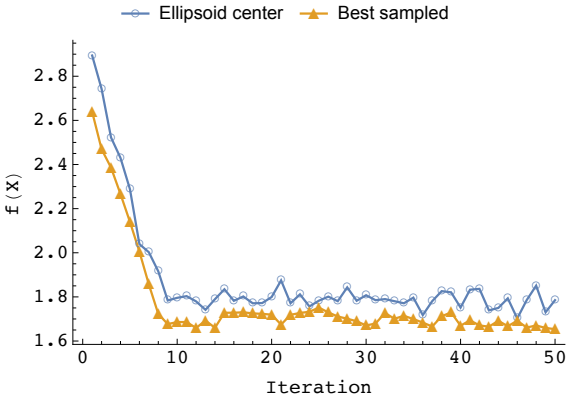
where p, q were described earlier. Trying different weights on the $d_{H,i}$ in the sum had little effect on the final results, and hence results for different weights are not reported here.

Nondefault values for the input arguments to the computer code QNSTOPS are described next. MODE is 'G' for global optimization, 'S' for stochastic optimization; N is the number

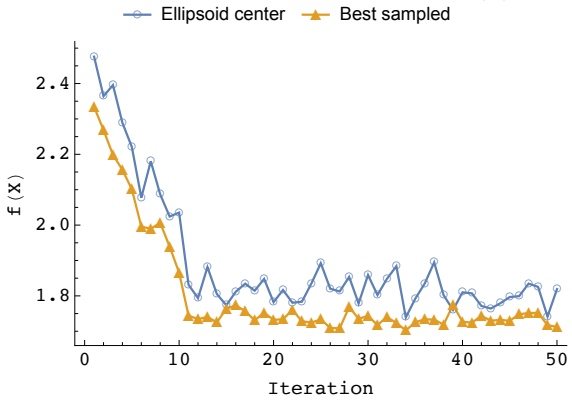
of design ellipsoid sample points (from the statistical rule of thumb that at least $1.5n$ data points are needed to estimate n parameters); XI is the initial start point; $[L, U]$ is the feasible box; TAU is the initial design ellipsoid radius τ; GAIN, relevant only for MODE 'G', defines the decay factor such that the design ellipsoid radius at iteration k is $\tau_k = \text{GAIN}/(\text{GAIN} + k - 1) \cdot \text{TAU}$.



(a) Low box corner



(b) Upper box corner



(c) Best value in [96]

Figure 5.3: Execution trace of QNSTOP for three start points from Table 5.1. The x-axis shows the iteration number, and the y-axis is the objective function value. For each iteration, the ellipsoid center (circles) and best sampled (triangles) objective function values are reported.

Using MODE = 'G' (global optimization), TAU = 20 (5% of the search box diameter), GAIN = 10, $N = 64$, Fig. 5.3 shows the iteration histories starting from three points chosen from the box $[L, U]$ in Table 1 (lower box corner, upper box corner, and the best value in

Oguz's model). The Hellinger distance starting from the upper corner point shows a clear descent from ≈ 2.9 to ≈ 1.75 in 13 iterations. Starting from the best point in [96], the Hellinger distance decreases from ≈ 2.5 to ≈ 1.75 in 15 iterations and then oscillates around that value. The same oscillation happens starting from the lower corner point, which suggests that ≈ 1.75 is the best objective function value, and that every point in the box near the corner L has about the same objective function value ≈ 1.8. Dozens of other different start points in the box $[L, U]$ produced similar best function values (QNSTOP can automatically generate a Latin hypercube design of start points including a given start point XI). Note that the best objective function values (≈ 1.75) are not particularly small in the (summed) Hellinger distance measure, meaning that the empirical data is not being matched especially well, although the average Hellinger distance of $\approx 1.75/8 = 0.21875$, or $|p(i) - q(i)| \approx 0.04$ on average, is not bad. (The maximum Hellinger distance is $\sqrt{2} \approx 1.414$.)

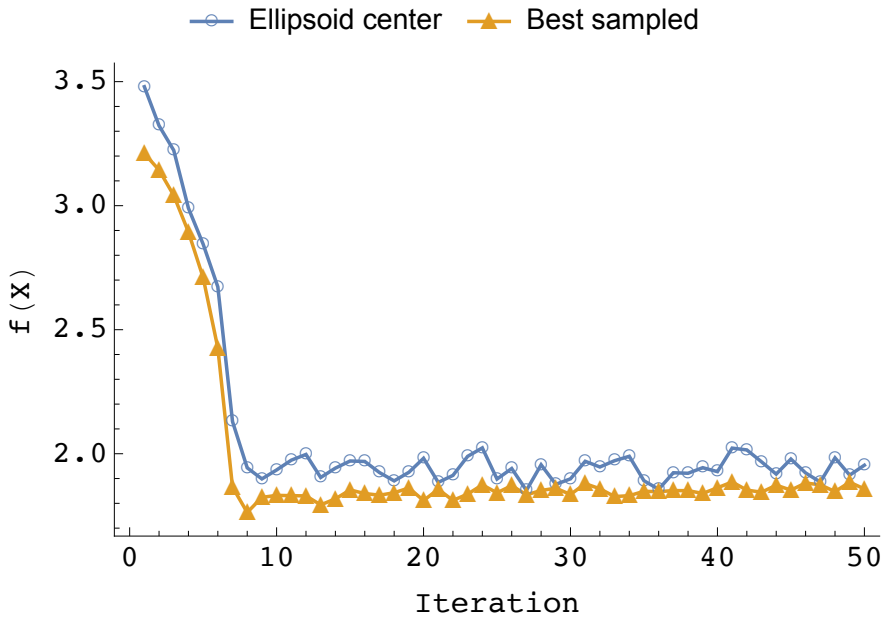


Figure 5.4: Execution trace of QNSTOP starting at the upper corner of the larger box $[(1/2)L, 2U]$. The x-axis shows the iteration number, and the y-axis is the objective function value. For each iteration, the ellipsoid center (circles) and best sampled (triangles) objective function values are reported.

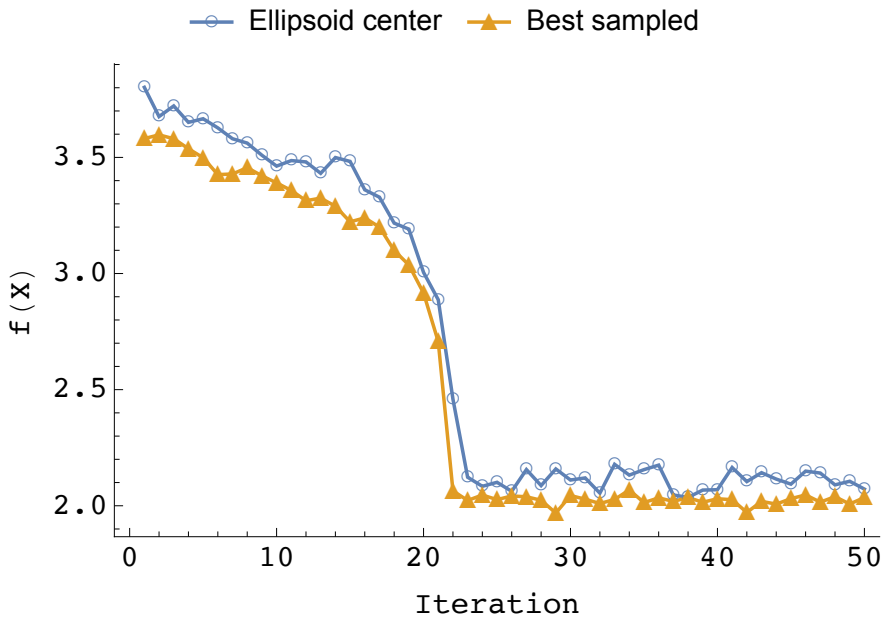


Figure 5.5: Execution trace of QNSTOP starting at the upper corner of the larger box $[(1/4)L, 4U]$. The x-axis shows the iteration number, and the y-axis is the objective function value. For each iteration, the ellipsoid center (circles) and best sampled (triangles) objective function values are reported.

To demonstrate that the stochastic parameters in Laomettachit's stochastic cell cycle model are not entirely arbitrary, and that QNSTOP can make progress on stochastic optimization problems, consider an enlarged search box $[(1/2)L, 2U]$ with the start point XI taken as the upper bound corner of this box. This start point is far away from the best point in [96] and has a much larger objective function value. The initial design ellipsoid radius TAU is also changed to 5% of the diameter of the new box, and GAIN = 10. A larger value for GAIN causes the ellipsoid radii to decrease more slowly, which is advantageous when starting far away from the optimum point. The execution trace in Fig. 5.4 drops rapidly to near 1.9 in less than 10 iterations, and stays around that value, apparently a local minimum. Fig. 5.5 shows the execution trace of QNSTOP from an even worse starting point (upper bound corner) in the much larger box $[(1/4)L, 4U]$, with the initial TAU adjusted as for Fig. 5.4, and GAIN = 10. The plot shows a downward trend and drops sharply around 20 iterations

to approach ≈ 2.1, apparently another local minimum.

As QNSTOP iterates, the design ellipsoid (in which samples are taken to build a quadratic model of the objective function) radius τ_k decreases. Fig. 5.3, showing the objective function value at the ellipsoid center and at the best sampled point inside that ellipsoid, thus gives a good indication of the variability of the stochastic objective function values within that ellipsoid. Observe that this variability shows little change with respect to the iteration number, meaning that the inherent simulation variance for a fixed parameter vector is roughly comparable to the variance within the (small) design ellipsoid.

Table 5.2: Individual Hellinger distances between empirical distributions and simulated distributions using the best point from Table 5.1 and the best point found by QNSTOP.

	Table 1	QNSTOP
$d_{H,1}$	0.57	0.44
$d_{H,2}$	0.37	0.22
$d_{H,3}$	0.16	0.09
$d_{H,4}$	0.19	0.12
$d_{H,5}$	0.45	0.31
$d_{H,6}$	0.37	0.22
$d_{H,7}$	0.19	0.10
$d_{H,8}$	0.18	0.15
$f(X)$	2.48	1.65

Table 5.2 shows the individual Hellinger distances $d_{H,i}(p, q)$ comprising the objective function $f(X)$, and that the best point (from all runs) found by QNSTOP is considerably better than that found by differential evolution in [96]. For completeness, Table 5.3 reports that best point X found by QNSTOP. In summary, QNSTOP performs well on this stochastic budding yeast cell cycle model, quickly finding the best known Hellinger distance even from a poor starting point, and significantly improving the result from differential evolution in [96]. From

Table 5.3: Best parameter vector for the budding yeast cell cycle found by QNSTOP.

Parameter	Value	Parameter	Value
k_{tr1}	0.6470	k_{tr2}	0.4938
k_{dm1}	0.8598	k_{dm2}	1.4749
m_{min1}	0.2085	m_{min2}	9.0806
k_{tr3}	0.4768	k_{tr4}	0.6377
k_{dm3}	2.1048	k_{dm4}	1.4175
m_{min3}	3.3014	m_{min4}	12.2215
k_{tr5}	0.5411	k_{tr6}	0.4676
k_{dm5}	1.9824	k_{dm6}	1.5821
m_{min5}	8.7150	m_{min6}	10.7990
k_{tr7}	0.5430	k_{tr8}	0.59856
k_{dm7}	1.3543	k_{dm8}	1.6878
m_{min7}	9.7407	m_{min8}	12.3070
k_{tr9}	0.5941	k_{tr10}	0.5638
k_{dm9}	2.0224	k_{dm10}	1.8554
m_{min9}	12.2770	m_{min10}	8.7718
c_{Cln3}	11.0180	c_{Bck2}	13.0380
c_{Whi5}	25.612	c_{Cln2}	59.8760
c_{CKI}	94.6380	c_{Swi5}	67.5470
c_{Cdc20}	123.9300	c_{Cdh1}	121.8000
c_{Cdc14}	20.1910	c_{PPX}	110.2900
c_{Pds1}	3.9074	c_{Cdc15}	11.0270
c_{POLO}	126.2300	c_{Mcm1}	125.5000

very distant starting points, QNSTOP converges to a (not globally optimal) local minimum point, which is not unexpected behavior.

From different starting points, the objective function values converge to near the same value, but is the same true of the parameter vectors X? For two parameter vectors $\hat{X}$ and $\tilde{X}$, consider the normalized error vector $E(\hat{X}, \tilde{X})$ defined by

$$E(\hat{X}, \tilde{X})_i = |\hat{X}_i - \tilde{X}_i|/(U_i - L_i).$$

For all the starting points, final points $\hat{X}$, and best found point $\tilde{X}$, $\text{median}_i\ E(\hat{X}, \tilde{X})_i < 0.12$; for all starting points, there were no more than three components $E(\hat{X}, \tilde{X})_i > 0.25$, and the 80th percentile of the $E(\hat{X}, \tilde{X})_i$ was always less than 0.2. Let X_L, X_U be the final points found starting from the lower, upper corners of the bounding box, respectively. The objective function along the line segment between X_L and X_U fluctuates around its optimum value ≈ 1.75, indicating that X_L and X_U are not isolated local minimum points, but are essentially the same parameter vector with respect to model behavior. Given the complexity of the stochastic model, and the sparsity of the empirical data, this agreement among final points from different runs of QNSTOP is about as good as can be expected.

5.6 Implications for the Cell Cycle Model

Mathematically, Table 5.2 shows how well the distributions of the various cell cycle observables (mass at birth, etc.) are being captured by Laomettachit's stochastic cell cycle model. The smallest Hellinger distances are associated with the distributions of birth masses for mother and daughter cells, $d_{H,3}$ and $d_{H,4}$, and with the cycle time distributions for mother and daughter cells, $d_{H,7}$ and $d_{H,8}$. The histograms of daughter cell cycle times (Fig. 5.6)

show how good the fit is between the model and the data in this particular case. The major discrepancies are in the tails of the distribution. In contrast, the distribution of G_1 durations for mother cells is not a good match: $d_{H,5} = 0.31$ in Table 5.2, and the histograms in Fig. 5.7 show clearly that the model overestimates the time spent by mother cells in G_1 phase of the cell cycle. This discrepancy points to a 'structural' problem of the model: the 'G_1-stabilizing' proteins in the model (Cdh1 and CKI) seem to be too active in mother cells, delaying the exit of mother cells from G_1 into S phase. On the other hand, the time spent by daughter cells in G_1 phase is not nearly so discrepant, $d_{H,6} = 0.22$ in Table 5.2, suggesting that the structural problem is related to some subtle difference between mother cells and daughter cells, which has escaped modelers' attention so far.

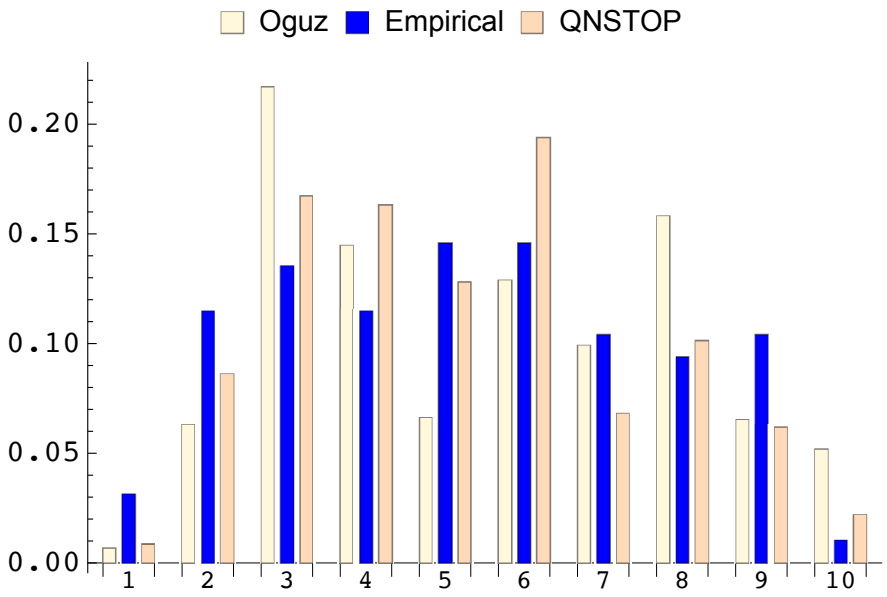


Figure 5.6: Comparison of histograms of the cell cycle time for daughter cells, from the simulation using the best point from Table 5.1 (Oguz), from the empirical data, and from the simulation using the best point found by QNSTOP.

The other data that are poorly matched by the model are the joint distributions of (mass at birth, duration of G_1 phase) for mother and daughter cells. The Hellinger distances from QNSTOP are $d_{H,1} = 0.44$ and $d_{H,2} = 0.22$, respectively, which are clear improvements over

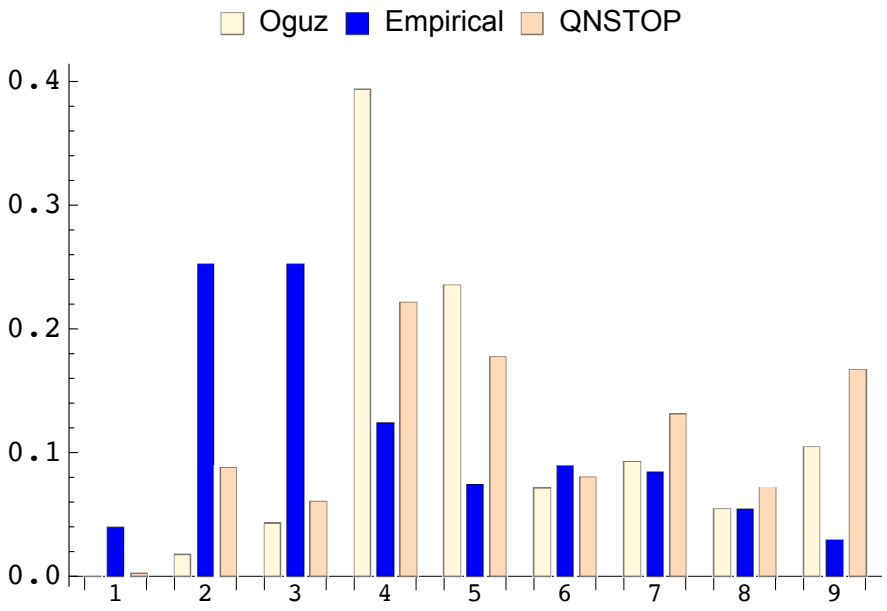
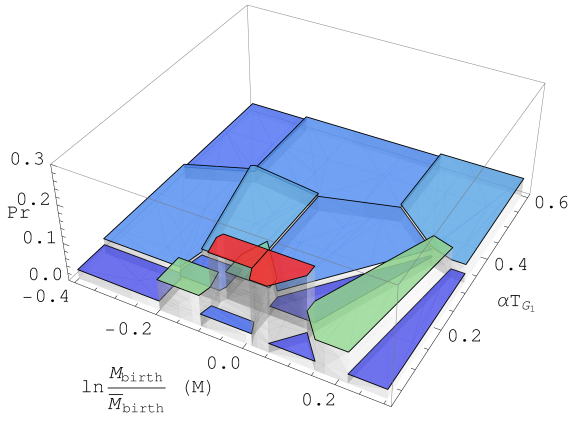
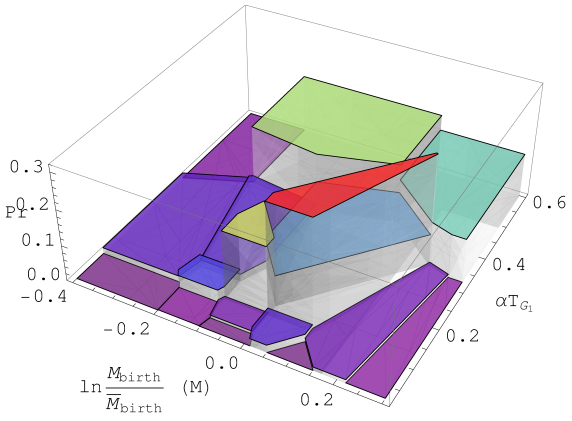


Figure 5.7: Comparison of histograms of G_1 duration for mother cells, from the simulation using the best point from Table 5.1 (Oguz), from the empirical data, and from the simulation using the best point found by QNSTOP.

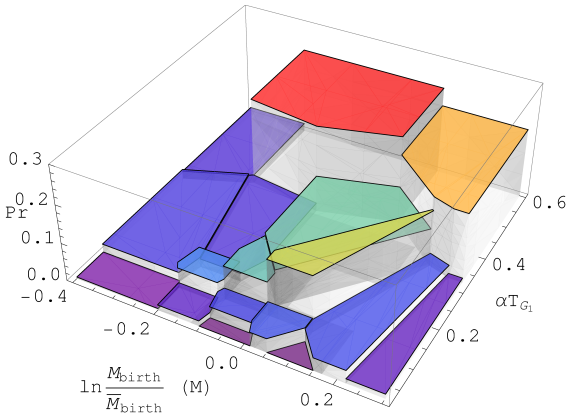
the best point from Table 5.1; nonetheless, the Hellinger distances are hard to interpret. Fig. 5.8 and 5.9 contain histograms of these joint distributions from the empirical data, from the simulation using the best point from Table 5.1, and from the simulation using the best point found from QNSTOP. Each simulation produces about 1,000 data points, compared to about 100 empirical data points. From these histograms it is evident that the major discrepancies between the model and the empirical joint distributions are in one specific region of the joint distribution: the region where T_{G_1} is large ($T_{G_1} > 0.2\alpha = 26$ min) and m_B is not too much different from the mean mass of mother cells at birth ($-0.3 < \ln(m_B/\overline{m_B}) < 0.1$). In this case, the model is clearly overestimating the number of cells (both mothers and daughters) that spend a long time in G_1 phase, which is complementary to the 'structural' problem noted above. The model underestimates the number of cells with short G_1 durations and overestimates the number of cells with long G_1 durations.



(a) Empirical data



(b) Best point from Table 5.1



(c) Best point found by QNSTOP

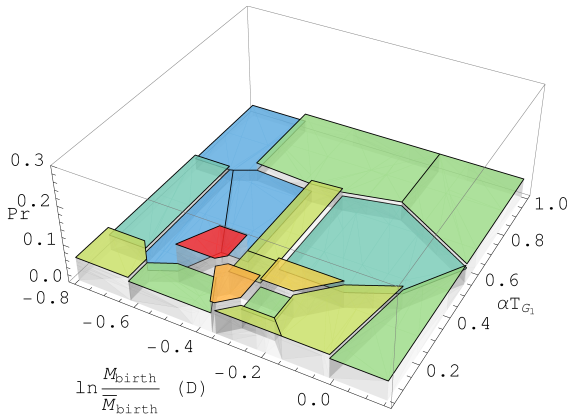
Figure 5.8: Two-dimensional histogram of the joint distribution of the pair (mass at birth, duration of G_1 phase) for mother cells from the empirical data, from the simulation using the best point from Table 5.1, and from the simulation using the best point found by QNSTOP. The polygons in this display correspond to the rectangles in Fig. 5.1, because the plotting program partitions the horizontal plane into a Voronoi diagram based on the centers of each of the rectangles in Fig. 5.1. The height of each polygon is the relative frequency of data points lying in the corresponding rectangle.

5.7 Conclusions

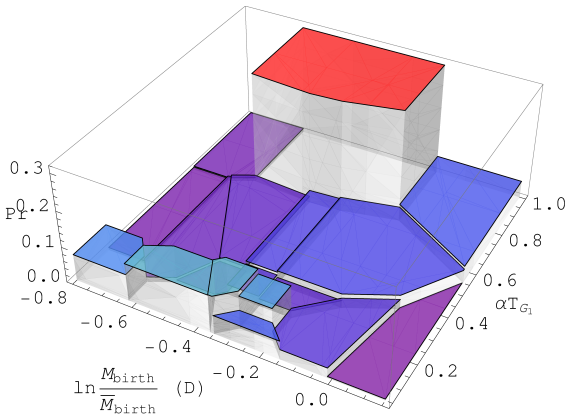
As observed in the Introduction, to understand fully the molecular basis of many aspects of cell physiology requires the construction of detailed mathematical models that take into account the intricate interactions among the genes, mRNAs, and proteins involved in regulating each process. Deterministic models, expressed as sets of nonlinear differential equations describing the temporal and spatial interactions of these molecules, are appropriate for understanding the average behavior of large populations of cells. On the other hand, to get at the statistical variability of how individual cells behave requires stochastic models that accurately describe cell-to-cell variability. Stochastic differential equations (SDEs) are often used for this purpose.

In either case—deterministic or stochastic models—the modeler is faced with a daunting task of estimating dozens of parameters (rate constants) by fitting model simulations to experimental observations. The parameter estimation problem is difficult enough for a deterministic model, because of the high dimension of the parameter space of any reasonably complete, molecular-level model of some aspect of cell physiology, and because of the general paucity of accurate and pertinent experimental data. For stochastic models, parameter estimation is more difficult indeed because one must compare statistical distributions (computed and observed) and vary the parameter values to optimize the fit. The computations are more expensive (typically hundreds or thousands of replica simulations to approximate the probability distribution function), and relevant experimental distributions of sufficient quality are rare indeed.

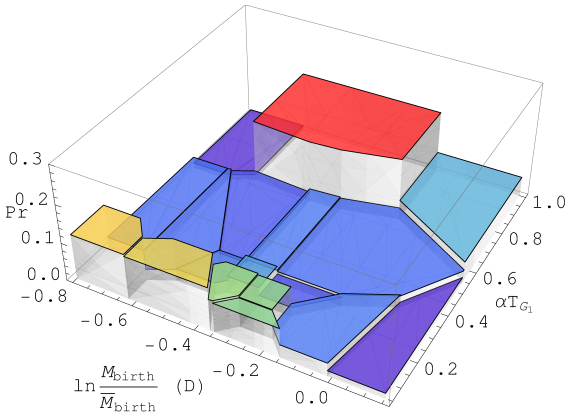
This chapter tested the efficacy of a quasi-Newton method for stochastic optimization (QN-STOP) to estimate the parameters in a system of SDEs that model the molecular interactions governing progression through the cell division cycle in budding yeast. The model has 44



(a) Empirical data



(b) Best point from Table 5.1



(c) Best point found by QNSTOP

Figure 5.9: Two-dimensional histogram of the joint distribution of the pair (mass at birth, duration of G_1 phase) for daughter cells from the empirical data, from the simulation using the best point from Table 5.1 and the best point found by QNSTOP. As in Fig. 5.8 the polygons correspond to the rectangles in a partition of the daughter cell data (not shown here), similar to that for the mothers in Fig. 5.1, and the height of each polygon is the relative frequency of data points lying in the corresponding rectangle.

independent parameters that determine the random fluctuations in molecular populations, and these fluctuations determine the variability from cell to cell of certain observable properties, such as cell cycle time, time spent in G_1 phase of the cell cycle, and cell size at birth. Di Talia et al. [35] have collected data on the distributions of these observables, and on the joint distribution of the pair (mass at birth, G_1 duration). Budding yeast cells divide asymmetrically into a large 'mother' cell and a small 'daughter' cell, so Di Talia measured separate distributions for mother-cell and daughter-cell populations. Hence, Di Talia provides sample data sets from eight different distributions.

QNSTOP can efficiently find a globally (but occasionally only locally) optimal stochastic parameter vector X by minimizing the sum of Hellinger distances $f(X) = \sum_{i=1}^{8} d_{H,i}(p, q)$ between the observed and computed probability mass functions p and q, respectively, for each of the eight different distributions. QNSTOP's fit to these distributions is considerably better than the 'best' fit found in an earlier publication [96], which used a differential evolution algorithm on an objective function that was a sum of squares of deviations between summary statistics (means and standard deviations) for the eight empirical distributions: $f(X) = 1.65$ for QNSTOP, $f(X) = 2.48$ for differential evolution. Presumably, QNSTOP is doing a better job because it is a more efficient algorithm than differential evolution and because it is using all of the information in the full distributions rather than just the summary statistics. A major conclusion of this chapter is that matching summary statistics and even marginal distributions does not in practice imply that the joint distributions match.

A few conclusions about the model can be drawn from the best parameter vector found by QNSTOP (Table 5.3). First of all, fluctuations in protein levels in the stochastic model are most sensitively dependent on the parameters $m_{min,i}$. Genes with smaller values of this parameter display larger fluctuations in protein levels. For Oguz's best parameter vector (Table 5.1), the noisiest gene expression is attributable to *CKI* and *PDS1*. For QNSTOP's

best parameter vector, *CLN3* is, by far, the noisiest gene, which seems quite reasonable because Cln3 protein abundance is quite low in budding yeast cells and Cln3-dependent kinase activity is known to play a major role in the G_1-to-S phase transition. Secondly, in QNSTOP's best parameter vector, all mRNAs (except for *CLN3* mRNA) have degradation rate constants in the range $1.3 - 2.1\ \text{min}^{-1}$, which corresponds to half lives in the range 0.33 – 0.51 min. These values seem to be quite smaller than what one might expect (say, 5 min half life), but rapid turn over of mRNAs seems to be necessary to limit the magnitude of protein-level fluctuations in the stochastic model. Notice that *CLN3* mRNA has a noticeably longer half life (1.25 min) than any of the other mRNAs in the model, presumably because it is fluctuations in *CLN3* mRNA numbers that plays the most important role in determining the noisiness of the model's behavior. The fact that the model requires rapid turnover of mRNA species in order to fit the observed probability distributions of cell cycle observables suggests that the way molecular noise is incorporated into the model may be oversimplified. More elaborate models, which incorporate mRNA bursting, mRNA processing, mRNA transport, etc., will have to be explored in later publications.

Chapter 6

Finding Acceptable Parameter Regions of Stochastic Hill Equations for Cooperative Binding Mechanisms

6.1 Introduction

Cooperative binding appears in a wide range of biochemical and physiological processes, such as multisite molecules [11, 61], transcription factors [1, 8, 70, 101], multimeric enzymes [21, 47], and drug-receptor relationships [54]. In particular, cooperative binding occurs when the binding of the first ligand molecule influences the binding affinity of the second or more ligand molecules. The cooperativity is considered positive if the previous ligand binding increases the next binding affinity (more likely to bind), and considered negative if the previous binding decreases the next binding affinity (less likely to bind).

There are various cooperative binding mechanisms, of which one basic sequential binding scheme is

$$B_0 \underset{b_1}{\overset{f_1[A]}{\rightleftharpoons}} B_1 \underset{b_2}{\overset{f_2[A]}{\rightleftharpoons}} B_2 \ldots \underset{b_{n-1}}{\overset{f_{n-1}[A]}{\rightleftharpoons}} B_{n-1} \underset{b_n}{\overset{f_n[A]}{\rightleftharpoons}} B_n. \tag{6.1}$$

B is the receptor with n binding sites available for ligand molecules. B_0 represents the free form. The fully saturated B_n is formed in sequential steps, with intermediate stages

increasing by one ligand molecule until n ligand molecules are bound to B. Assume that all binding sites are the same and there is no preference over the selection of binding sites. b_i $(i = 1, 2,\ldots,n)$ denotes the dissociation rate of B_i. The binding process is activated by an enzyme A, thus the association rate of B_i is formulated as the mass action $f_i[A]$, where $[A]$ denotes the quantity (population or concentration) of the enzyme A. In the stochastic regime, $[A]$ represents the enzyme population. Assume that the population of A is constant and there are enough ligand molecules for binding in system (6.1). The cooperative binding scheme (6.1) can also be interpreted as a processive multisite protein phosphorylation process, where B represents the substrate, ligand molecules refer to phosphatase, and enzyme A is the kinase. Kinetic models of multisite phosphorylation are usually described as the elementary reactions [12, 112, 121]. Cooperative binding, described in a mathematical way, is a binding process where the number of occupied binding sites has a nonlinear relation to the ligand's concentration [119].

The Hill equation was first introduced to model the observed curves of ligand binding to the receptor [59]. The equation is a nonlinear function of the concentration of ligand, and the Hill coefficient defines the degree of cooperativity of ligand binding. Assuming ligand binding happens simultaneously, the Hill equation is reliable only when the cooperativity of binding sites is extremely positive [130]. Other complex models have been proposed to describe different cooperativity of ligand binding [2, 69, 90, 97]. For example, the Adair equation assumes that the ligand binding is formed in a sequential order with one ligand molecule, two, etc., until fully saturated [2]. The Koshland-Némethy-Filmer (KNF) model further distinguishes binding complexes into two conformations: active or inactive [69]. However, the Hill equation requires little prior knowledge of the binding mechanism and is much simpler than the other proposed models, thus is widely used in biochemical networks to model fast signal response and complex binding processes.

As the Hill equation has been extensively used and thoroughly studied in traditional deterministic models (differential equations), the unexplored discrete stochastic representation is getting more and more attention from researchers with the presence of low population levels and molecular noise. Previous studies have discovered that the sigmoidal behavior of the Hill equation dynamics may reduce to a linear function in the stochastic regime, especially under the reaction-diffusion master equation framework [26]. This work restricts attention to a homogeneous domain.

With the increasing need for model reduction of complex biochemical networks, this work considers a stochastic Hill equation system for the basic binding reaction scheme (6.1) using the stochastic simulation algorithm (SSA) [49]:

$$B_0 \underset{k_d}{\overset{k_a \dfrac{[A]^\sigma}{k_m^\sigma + [A]^\sigma}}{\rightleftharpoons}} B_n, \tag{6.2}$$

where k_a and k_d are the forward and reverse reaction rates, respectively, k_m is the dissociation rate, and σ is the Hill coefficient, a real number with $\sigma \leq n$. This work, instead of validating the stochastic results with the deterministic results, optimizes the stochastic Hill equation for modeling the cooperative binding process. As biologists are able to quantify species population at molecular levels with improved experimental techniques [103], the empirical data from experiments provide good resources for validating stochastic models and optimizing stochastic parameters.

The challenges of solving the inverse problem of parameter estimation for modeling biological systems are manifold. One major difficulty is related to the large scale of the biological networks, which adds many unknown parameters that are hard or impossible to measure in experiments. The nonlinear nature of the models involves a nonconvex problem with multiple (locally) optimum points and local optimization methods may be trapped at local

optimum points. Global optimization methods can be very expensive in high dimensions, and global optimality is often not guaranteed [10]. Moreover, most of the efforts in solving such problems thus far have been focused on deterministic models, particularly estimating the parameters of models formulated by nonlinear differential equations [89]. That is because prior to the emergence of single cell study, which makes the variability between individual cells measurable, most existing models in systems biology were deterministic. Parameter estimation for deterministic models of biological systems is a challenging task. Various approaches have been adopted to estimate parameters of such dynamical models. One common approach is based on numerical optimization methods. Various local and global optimization methods have been used to identify kinetic parameters of biochemical pathways, including gradient based methods, direct search, simulated annealing, and evolutionary algorithms as well as hybrid or sequential methods of local and global optimization. Mendes et al. compared several of these optimization-based methods for estimating the parameters of biochemical pathways [89].

Parameter estimation in stochastic models is even more challenging as the amount of empirical data must be large enough to obtain statistically valid parameter estimates. Two well-known approaches for stochastic optimization problems are stochastic approximation (SA) and response surface methodology (RSM). The class of quasi-Newton methods for stochastic optimization extends state-of-the-art numerical optimization methods (e.g., secant updates, trust regions) [20], which also can be used for deterministic global optimization with minor variations [5, 37], has been successfully applied to various stochastic optimization problems, such as cell cycle models [29], bistable models [27], and biomechanics problems [102]. An alternative approach is probabilistic Bayesian inference [83]. Since the exact Bayesian method is computationally intractable for a realistically large model, a variant of Markov Chain Monte Carlo can be used for sampling [108] and approximate Bayesian computation [83] are

preferred. The Kalman filter and its variants have been applied to solve this problem as well [82]. Recently, machine learning techniques have been tailored by sparsity-promoting methods to identify not only the parameters but also the structure of both ordinary differential equations [15] and partial differential equations [111].

Yet, most parameter optimization methods only return a single best parameter vector, regardless of the fact that there are many parameter vectors that could generate similar system dynamics and characteristics. Take multisite protein phosphorylation as an example. Bistable phenomena occur when model parameters are inside the bistability parameter region. The whole bistability region may be considered acceptable if the goal is to model bistability, rather than to find the best fit. This work focuses on finding an acceptable parameter region, where parameter vectors in the region are good alternatives to the best parameter vector. In other words, the system parameters are given by a region rather than a single point.

The goal is to find an acceptable parameter region so that the evolutionary population of B_n in the stochastic Hill equation system (6.2) matches well with the simulated empirical data from the cooperative binding scheme (6.1). In Section 6.2, three objective functions measuring different features of the empirical data and different system sizes are investigated. Section 6.3 then presents the proposed α-β-γ rule for searching the acceptable parameter regions based on QNSTOP. Numerical results and detailed analyses are given in Section 6.4.

6.2 Objective Functions

This section presents three different objective functions emphasizing different aspects of the empirical data. In particular, we propose a general simulation-based objective function that can be applied to large biochemical networks.

6.2.1 Minimum distance area

In the reaction binding scheme (6.1), suppose $D = [x_1, x_2, \ldots, x_m]$ is a sequence of the molecular population of B_n collected after every time τ (denoted as $[t_1, t_2, \ldots, t_m]$), where m is the data size. This subsection will consider the population difference between the empirical and simulation results over time, called the distance area. Given the empirical data and the simulated data as two vectors, the p-norm is one traditional way to measure the vectors' difference. The problem is stochastic and the time series data can be quite noisy, and outliers have more influence for $p > 1$, hence the 1-norm is used to measure the distance. Define the distance area as the objective function,

$$f_d(\theta) = \int |p(t) - q(t)|dt \approx \sum_{i=1}^{m} |x_i - y_i|\tau, \tag{6.3}$$

where θ is the model parameters, $p(t)$ and $q(t)$ are the trajectory functions of empirical and simulated populations in continuous domains. For discrete time series data, x_i and y_i represent the population of B_n from empirical and simulated data, respectively. The stochastic optimization problem to be solved is

$$\min_{\theta \in \Theta} f_d(\theta), \tag{6.4}$$

where Θ is a set in $\mathbb{R}^n$ defining the feasible set (allowable values for the model parameter vector θ).

6.2.2 Maximum log-Likelihood

The minimum distance area, similar to other traditional optimization methods, builds objective functions based on 'mean' measurements from stochastic simulations, hence cannot

reflect the intrinsic noise in stochastic models. To capture the stochastic fluctuations, measure the transition probability that a system jumps from one state to the next state after a certain time step. The likelihood function of time series data can then be factorized into the product of transition probabilities. For convenience, the logarithm of the likelihood, which changes a product to a summation, is used. The logarithm of the likelihood of the observed data D is

$$\log \mathcal{L}(\theta|D) = \log \left(\prod_{i=2}^{m} \mathcal{T}_{x_{i-1},x_i} \right) = \sum_{i=2}^{m} \log \mathcal{T}_{x_{i-1},x_i} \tag{6.5}$$

where $\theta \in \mathbb{R}^n$ is model parameters and $\mathcal{T}$ is the transition matrix. Specifically, $\mathcal{T}_{x_{i-1},x_i}$ is the transition probability that the system changes from state x_{i-1} to state x_i. Note that we take the logarithm of the likelihood because the transition probability matrix is usually very close to zero. A larger value of log-likelihood indicates a higher similarity between the empirical data and simulation data with parameter vector θ.

The objective function is

$$f_l(\theta) = -\log \mathcal{L}(\theta|D), \tag{6.6}$$

and the stochastic optimization problem to be solved is

$$\min_{\theta \in \Theta} f_l(\theta), \tag{6.7}$$

where Θ is a set in $\mathbb{R}^n$ defining the feasible set (allowable values for the model parameter vector θ).

When a system has a finite number of states, then we can calculate $\mathcal{T}$ directly from Eq. (2.4). When a system is small and has an infinite number of states, the finite state projection (FSP) method [91] projects the infinite state vector X to a finite state vector, approximating the CME solution with an error ϵ. Accordingly, A and $\mathcal{T}$ are approximated by $\hat{A}$ and $\hat{\mathcal{T}}$,

respectively. Fox et al. [44] proved that the FSP-derived likelihood converges monotonically to the exact likelihood value.

6.2.3 Approximate maximum log-likelihood

In the above maximum log-likelihood method, the approximation of the transition matrix $\mathcal{T}$ is only tractable for small systems. It is difficult or nearly impossible to solve the CME for large systems. To overcome this limitation, this subsection proposes a general use objective function that is applicable to large and/or complex biochemical models, called approximate maximum log-likelihood.

The transition probability of system state going from x_{i-1} to x_i is

$$\mathcal{T}_{x_{i-1},x_i} = \Pr(x_i|x_{i-1}). \tag{6.8}$$

Thus, the logarithm of the likelihood of the observed data D is

$$\log \mathcal{L}(\theta|D) = \log\left(\prod_{i=2}^{m} \Pr(x_i|x_{i-1})\right) = \sum_{i=2}^{m} \log \Pr(x_i|x_{i-1}), \tag{6.9}$$

where $\Pr(x_i|x_{i-1})$ can be approximated by simulation (an example will be shown later in (6.13)). The objective function is

$$f_p(\theta) = -\log \mathcal{L}(\theta|D). \tag{6.10}$$

The stochastic optimization problem to be solved is

$$\min_{\theta\in\Theta} f_p(\theta), \tag{6.11}$$

where Θ is a set in $\mathbb{R}^n$ defining the feasible set.

Algorithm 1 summarizes the essential steps of the approximate maximum log-likelihood method. In Line 6, by simulating the system several times from time t_{i-1} to t_i with initial system state $B_n = x_{i-1}$, we get a set S_i of simulation results for B_n at time t_i. $\Pr(x_i|x_{i-1})$ can be approximated from the distribution of S_i. For example, assuming x_i (sampled in S_i) follows a normal distribution

$$x_i \sim N(\mu, \sigma^2), \tag{6.12}$$

where μ and σ^2 are the mean and variance of x_i, we can approximate the probability as

$$\Pr(x_i|x_{i-1}) \approx \Pr(x_i - 0.5 < \mu + \sigma z_i < x_i + 0.5), \quad z_i = \frac{x_i - \mu}{\sigma} \sim N(0,1). \tag{6.13}$$

Note that the empirical data could be multiple species' evolutionary population, then x_i in $\Pr(x_i|x_{i-1})$ becomes a vector referring to multiple species' population. For line 4, if the simulation results do not include $B_n = x_i$, then choose one that is the closest to x_i.

Algorithm 1 Approximate maximum log-likelihood

Input: Empirical data $D = [x_1, x_2, \ldots, x_m]$ represents the evolutionary population of species B_n

Output: $-\sum_{i=2}^{m} \log \Pr(x_i|x_{i-1})$

1: **Initialization** $i = 2$, system state x_1.
2: **while** $i \leq m$ **do**
3: **if** $i = 2$ **then** Go to line 6
4: **else** Initialize the system with the simulation result where $B_n = x_{i-1}$ at time t_{i-1}
5: **end if**
6: Simulate the system q times from time t_{i-1} to t_i giving the list $S_i = [s_1, s_2, , s_q]$ of simulation results for B_n at time t_i.
7: Construct $\Pr(x_i|x_{i-1})$ based on the distribution of S_i
8: $i = i + 1$
9: **end while**

6.3 Acceptable Parameter Region

As mentioned before, for most systems, especially those with multidimensional parameters, there are many possible parameter combinations that produce similar system dynamics and behaviours. This section introduces the α-β-γ rule to find the acceptable parameter regions of the stochastic Hill equation. Parameter values sampled in the returned acceptable region should give close system results to the best parameter values. In particular, we applied the α-β-γ rule to QNSTOP, which has been used to find the best system parameter values of various stochastic problems similar to other optimization methods.

6.3.1 α-β-γ Rule

Based on the fact that QNSTOP defines an ellipsoidal design region for each iteration, we can utilize this ellipsoid to define the acceptable parameter region.

For an ellipsoid E, define $f(E) = \{f(x) \mid x \in E\}$. To accept a parameter region, intuitively, most parameters sampled from the region should have relatively small objective values, and close to the minimum objective function value $\min f(E)$ of the ellipsoidal region. After scrutinizing all possible distributions of $f(E)$, we define a stable ellipsoidal region E as follows:

Definition 6.1. E is a min-stable region if

$$\Pr\big[f(\theta) \leq (1+\alpha)\min f(E)\big] \geq \beta, \quad \theta \in E,\ \alpha \in (0,\infty),\ \beta \in (0,1].$$

In the above definition, α measures how close the objective function values are to the minimum value $\min f(E)$, β controls the percentage of parameter values that generate close minimum objective function values. α and β can be assigned with different values depend-

ing on the problem. For any parameter region E, if α is fixed, define the percentage of points with objective function values no larger than $(1+\alpha)\min f(E)$ as the **region stability**, which is the value of $\Pr\big[f(\theta) \leq (1+\alpha)\min f(E)\big]$.

As QNSTOP designs a specific ellipsoid region for each iteration, the minimum objective function values found may vary dramatically between iterations. We don't want to accept the ellipsoidal region E_1 of the first iteration even if E_1 is min-stable, because $\min f(E_1)$ is usually much larger than the minimum objective value f_{min} found over all iterations and all starting points (if QNSTOP is run with multiple starting points). Thus, to ensure that the parameter region is globally min-stable, we need to choose those min-stable regions whose local minimum objective function values are close to the global minimum. The acceptable parameter region is defined as a union of min-stable ellipsoids with local minimum objective function values close to f_{min}:

Definition 6.2. An acceptable region is $R = \bigcup_{k\in\mathcal{B}} E_k$ where

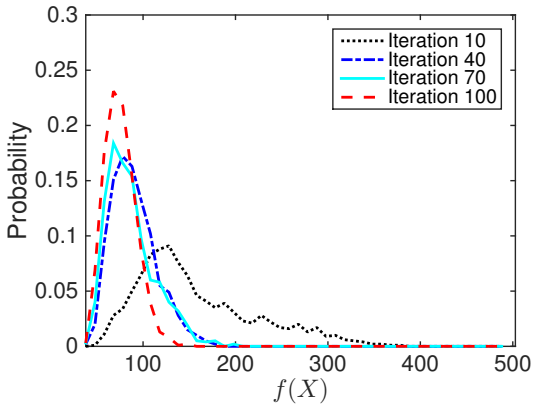
$$\mathcal{B} = \big\{k \mid E_k \text{ is min-stable and } \min f(E_k) \leq (1+\gamma) f_{min}\big\}, \quad \gamma \in [0, \infty).$$

In the above definition, k is the iteration number, γ controls how close $\min f(E_k)$ is to the global minimum, and γ may vary according to the problem. Choosing values for α, β, γ is referred to as an α-β-γ rule.

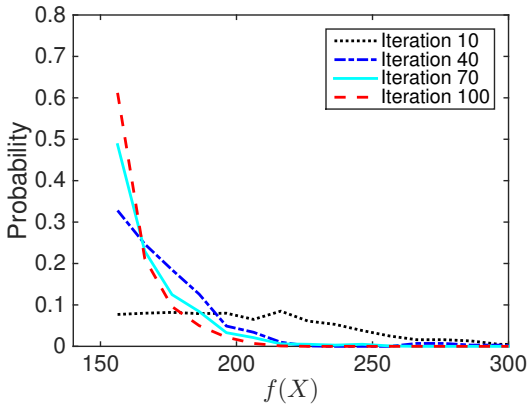
6.3.2 Analysis

Values for α, β, γ, derive from analyzing the objective function values in parameter regions. Fig. 6.1 shows the distributions of the three objective function values over several iterations. For iteration 10, $f(X)$ is almost a uniform distribution, with values spread out over the

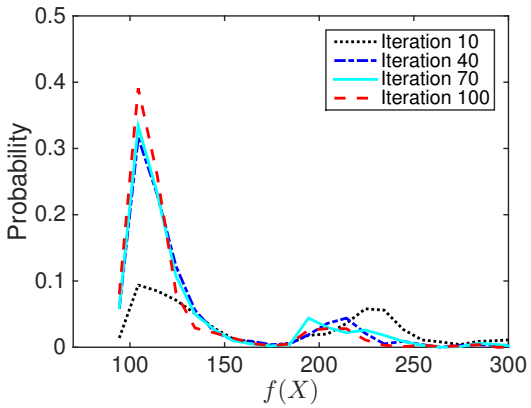
domain. As the iteration continues, more objective function values are clustering around the minimum value f_{min}. The distribution of $f(X)$ gradually forms a peak around f_{min}. While these features hold for all three methods, the maximum log-likelihood has the highest percentage of the small objective values. Based on the distributions, the values of α for the three methods are chosen as 0.5 (minimum distance area), 0.2 (maximum log-likelihood), 0.3 (approximate maximum log-likelihood), respectively. In Fig. 6.2, the corresponding regions for parameters k_a, k_d shrink with iterations.



(a) Minimum distance area



(b) Maximum log-likelihood



(c) Approximate maximum log-likelihood

Figure 6.1: Distribution of objective function values from three methods (minimum distance area, maximum log-likelihood, and approximate maximum log-likelihood) based on 1000 sampled points inside the ellipsoidal regions for iterations 10, 40, 70, and 100.

Figure 6.3 shows that for all three methods, the average region stability over 100 start-

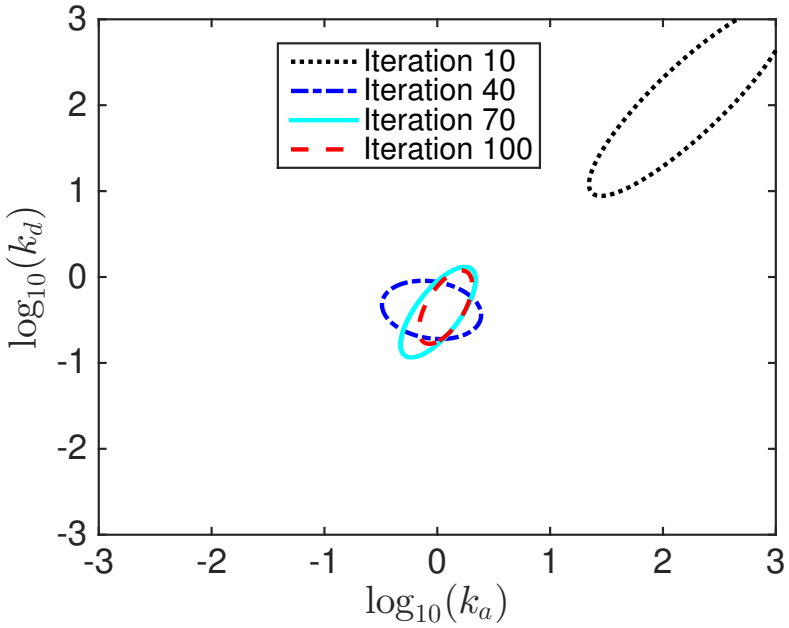


Figure 6.2: QNSTOP ellipsoids at iterations 10, 40, 70, and 100 from the maximum log-likelihood method.

ing points increases with iteration, eventually reaching one if the iteration number is large enough. The maximum log-likelihood has the highest region stability compared with the other two objective functions. For the maximum log-likelihood method, we set $\alpha = \gamma = 0.2$ and $\beta = 0.8$ based on the region stability. In this way, at least 80% of the sampled points from the acceptable region have objective function values within 20% relative error of the minimum value. We call this criterion the 80%-20% rule. For the other two methods, we set $\alpha = \gamma = 0.3$, $\beta = 0.7$ for approximate maximum log-likelihood, and $\alpha = \beta = \gamma = 0.5$ for minimum distance area.

6.4 Results

This section first discusses the input parameters, empirical data, and experimental setup. The results are divided into two parts, the first of which demonstrates the two parameter

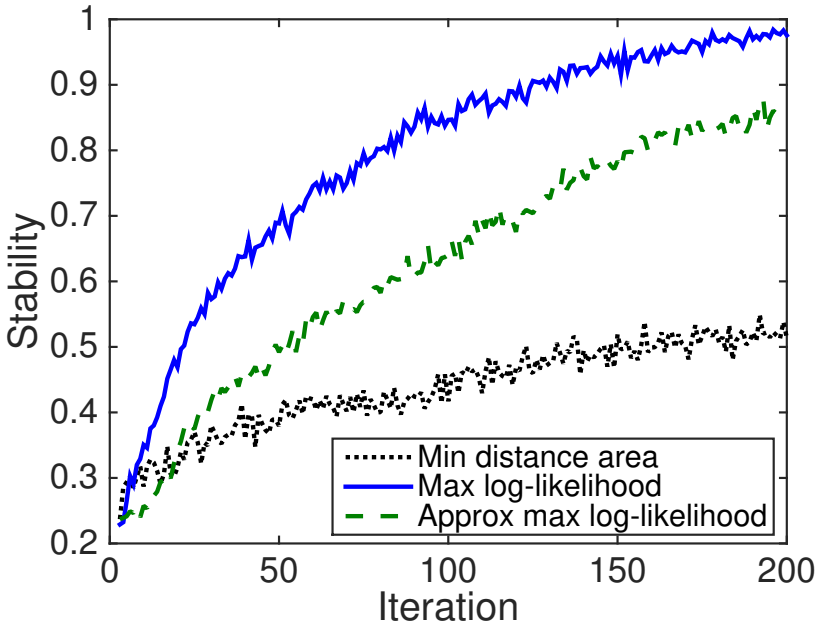


Figure 6.3: Region stability from iteration 1 to 200 based on 100 starting points from minimum distance area ($\alpha = 0.5$), maximum log-likelihood ($\alpha = 0.2$), and approximate maximum log-likelihood ($\alpha = 0.3$).

case (the acceptable parameter region is a two-dimensional graph), and the second part compares the three objective functions and studies the full parameter set by two-dimensional projections of the acceptable parameter region.

6.4.1 Experimental setup

Empirical data. This work assumes the sequential binding system (6.1) as the ground truth. In particular, consider the system size as $n = 4$, the population level of enzyme A as $[A] = 1000$, and the reaction rates as $f_i = 0.0025$, $b_i = 1$ for $i = 1, 2, 3, 4$. We use the stochastic simulation algorithm to simulate the system and sample one population trajectory of B_n with a time step τ ($t_1 = \tau$, $t_2 = 2\tau$, ..., $t_m = m\tau$) as a single set D of empirical data. Since both systems (6.1) and (6.2) stabilize at a steady state after a certain time, if the

empirical data D contains more steady state points than transition points, then the system parameters will be optimized in a way that minimizes the difference of steady states but overlooks the transition dynamics before the system stabilizes, and vice versa. In order to not bias either the stable state or the transition dynamics, the sampled data points are split equally in describing the two properties.

The stochastic Hill equation system (6.2) has an initial condition: $[B_0] = 100$ and $[B_n] = 0$, thus there are at most 101 system states. Table 6.1 lists the bounds for each parameter in the system.

Table 6.1: Parameter boundary in the stochastic Hill equation system.

Parameter	k_a	k_d	k_m	σ
$[L, U]$	$[0.001, 1000]$	$[0.001, 1000]$	$[0.001, 10^6]$	$[0.001, 10]$
$\log_{10}([L, U])$	$[-3, 3]$	$[-3, 3]$	$[-3, 6]$	$[-3, 1]$

Note that parameters k_a and k_d (rate constants of association and dissociation) are more sensitive in controlling the system's time scale than the other two parameters. Assuming k_m and σ are fixed values, while region $[0.001, 1]$ for parameters k_a and k_d occupies 0.0001% of the entire search box, the system time scale varies by three orders of magnitude. In Fig. 6.4, the final population of B_n initially grows linearly (in logarithm) with k_a/k_d and then levels off (B_n is fully saturated), which indicates that when $k_a > 1$, $k_d = 1$. The objective function value would not change much as the population of B_n at the stable state is always around 100. Thus, the decimal region $[L, 1]$ ($L < 1$ is the lower bound), while sensitive, is minimized or overlooked when the upper bound U is much larger than one ($U \gg 1$). We refer to this phenomenon as **decimal parameter sensitivity lost**, which can affect the optimization performance, especially for systems that are sensitive to the $[0, 1]$ domain. To avoid this problem, simply use the logarithms of the parameters in the simulation. So the bounds for $(\log_{10}(k_a), \log_{10}(k_d))$ are written as $[-3, 3]$.

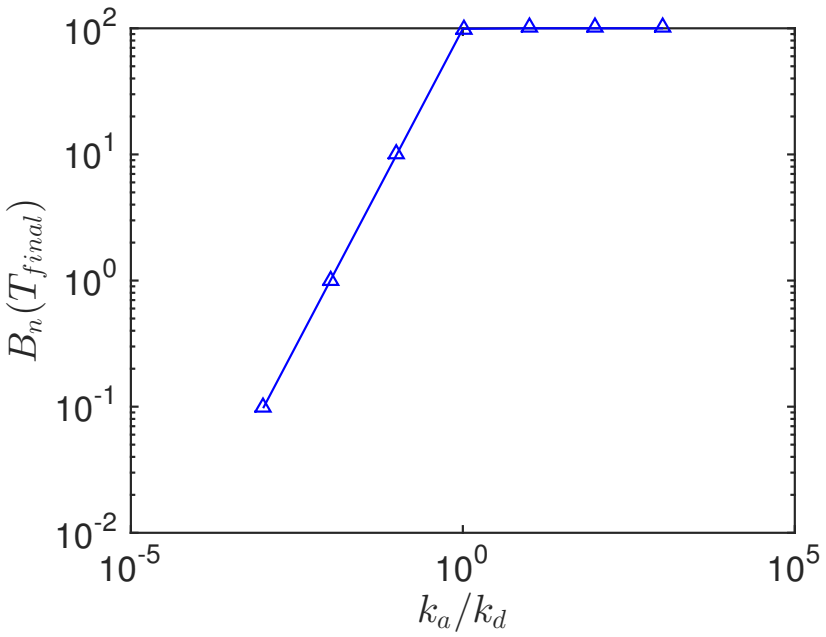


Figure 6.4: The population of B_n at stable state under the stochastic Hill equation system with respect to different k_a/k_d values, where $k_d = 1$.

6.4.2 Two parameter case

This subsection studies the two parameter case, in which only k_a and k_d are unknown while $\sigma = 2$ and $k_m = 100$ in the stochastic Hill equation system. QNSTOP parameters are: total iterations = 100, sample points N = 10, initial ellipsoid radius TAU = 0.85 (one-tenth of the searching box diameter), ellipsoid decay factor GAIN = 35, MODE = 'G'. There are $m = 50$ empirical data points collected at time step $\tau = 0.2$ from one SSA simulated trajectory of the B_n population. The α-β-γ acceptable region is defined by $\alpha = 0.2$, $\beta = 0.8$, $\gamma = 0.2$, which indicates that 80% of the sampled values should have objective function values within 20% relative error of the local minimum, and the local minimum is within 20% relative error of the global minimum.

Fig. 6.5 shows the maximum log-likelihood objective function values (f_l) over the entire k_a, k_d domain. Any (k_a, k_d) pairs sampled in the dark blue region in the middle of the graph

are acceptable parameters for the stochastic Hill equation system (6.2).

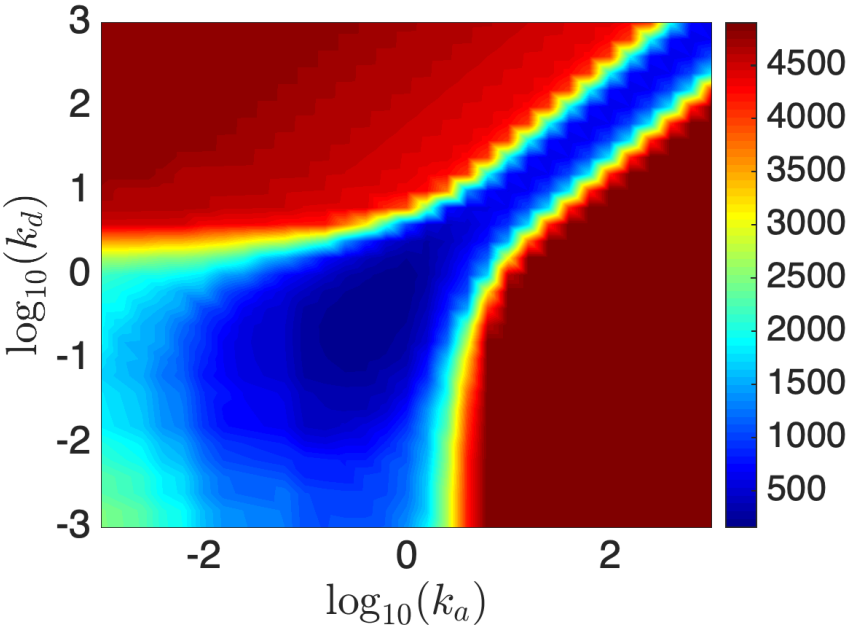


Figure 6.5: Exhaustive search over $(\log_{10}(k_a), \log_{10}(k_d)) \in [-3, 3]$, plotting the values of the maximum log-likelihood objective function.

Influence of starting points. Fig. 6.6 shows the execution traces of QNSTOP and the corresponding acceptable parameter regions from different starting points: lower boundary point L, upper boundary point U, and center point $(L + U)/2$. From the execution traces of all three starting points, the best sampled objective function value $f_l(X)$ decreases fast in the first 40 iterations and then oscillates around 200. The worst sampled objective function values also oscillates around 200 after 80 iterations. The acceptable parameter regions, even though there are subtle differences in the region size and number of acceptable ellipsoids, are pretty close and are located in the range of $[-1, 0.5]$ for both k_a and k_d. Since QNSTOP allows multiple starting points, the rest of the chapter will show the acceptable parameter regions collected from ten randomly selected starting points.

Influence of empirical data. To check the robustness of our algorithm, we vary the empirical data and the data size. Fig. 6.7 illustrates the acceptable regions from the same

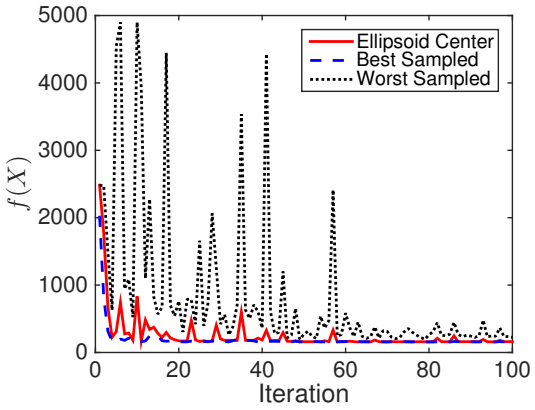
dataset but with different numbers of data points. Fig. 6.7(a) uses the population of B_n at time step $\tau = 1$, 10 data points total, while Fig. 6.7(b),(c) have 50 and 200 data points, respectively. The acceptable region increases with the data size m.

Fig. 6.8 shows the results from three different empirical datasets (three different population trajectories of B_n). Even with different empirical data, the acceptable parameter regions are consistent in size and shape. Thus, the parameter optimization of the stochastic Hill equation system is more sensitive to the empirical data size than the data content variation.

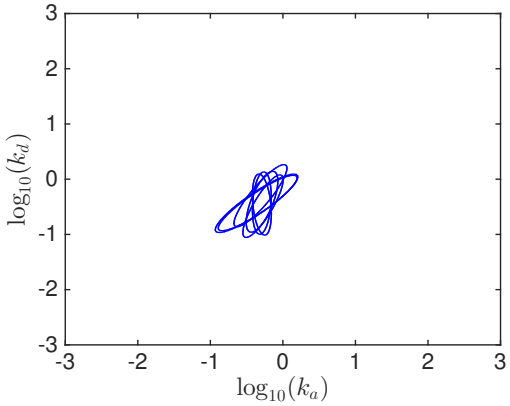
6.4.3 Full parameter case

This subsection studies the full parameter vector, in which all four parameters k_a, k_d, σ, and k_m are unknown in the stochastic Hill equation system. Simulations use the same parameter input except the initial ellipsoid radius TAU = 1.3 and sampling points N=20. For the empirical data, we use the same dataset from one SSA simulated trajectory of B_n population where $m = 50$, $\tau = 0.2$. The values of α-β-γ defining the acceptable regions are set differently according to the objective functions.

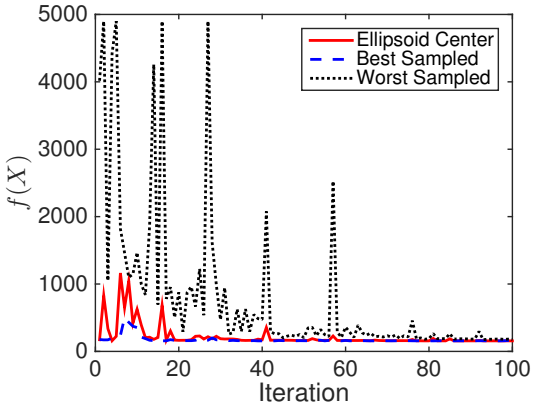
Influence of objective functions. Fig. 6.9 shows the results of the three objective functions: maximum log-likelihood, minimum distance area, and approximate maximum log-likelihood. For all three methods, the acceptable regions for the (k_a, k_d) pair are an ellipsoid shape centered at the middle of the domain, though the size has subtle differences. While for the (k_m, σ) pair, the acceptable regions are all over the domain, shown in Fig. 6.10a, indicating that the system is not sensitive to parameters k_m, σ. Note that in this Hill equation system, where the population level of enzyme A is fixed, the acceptable regions of (k_m, σ) pair are very different if considering multiple population levels of enzyme A. Fig. 6.10b shows the result for an objective function that sums over 11 population levels of enzyme A,



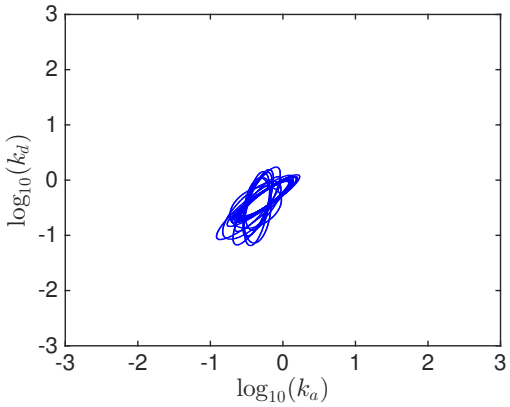
(a) Execution trace of QNSTOP



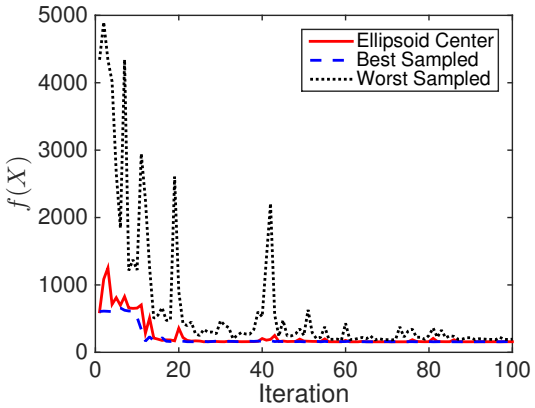
(b) Acceptable region



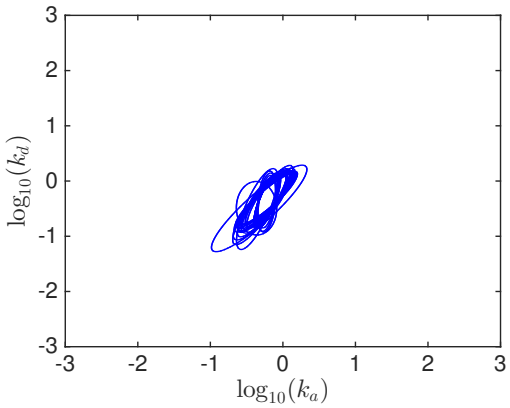
(c) Execution trace of QNSTOP



(d) Acceptable region

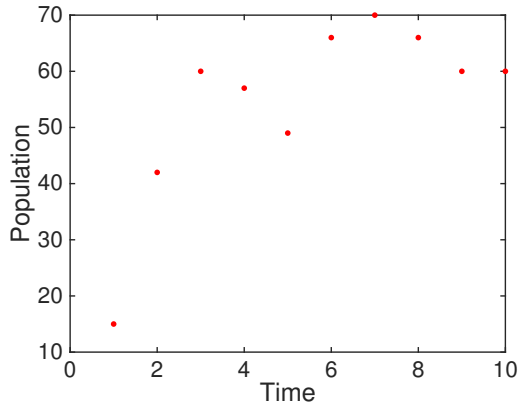


(e) Execution trace of QNSTOP

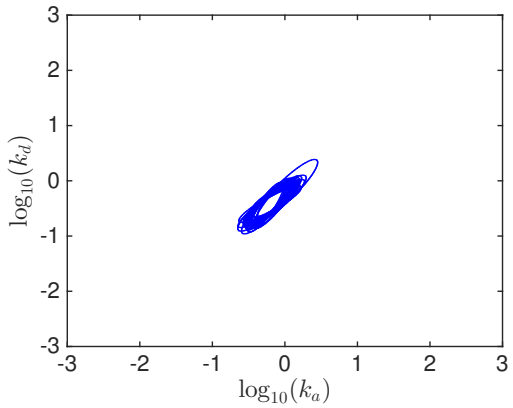


(f) Acceptable region

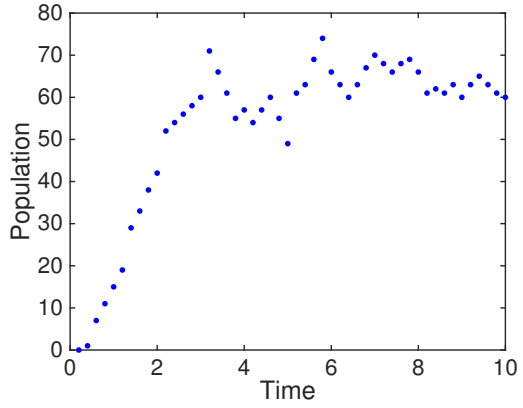
Figure 6.6: Optimization results of maximum log-likelihood with different starting points: (a, b) the lower box corner $(\log_{10}(k_a), \log_{10}(k_d)) = (-3, -3)$; (c, d) box center $(\log_{10}(k_a), \log_{10}(k_d)) = (0, 0)$; (e, f) the upper box corner $(\log_{10}(k_a), \log_{10}(k_d)) = (3, 3)$.



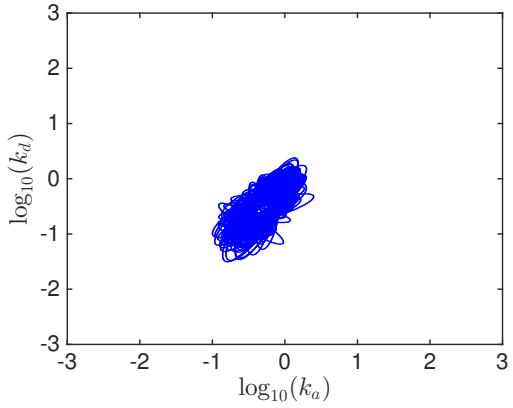
(a) $\tau = 1, n = 10$



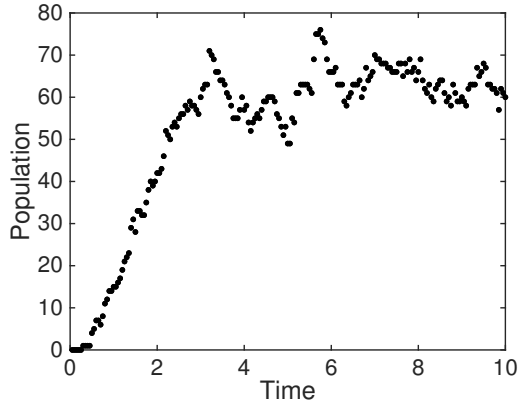
(b) Acceptable region



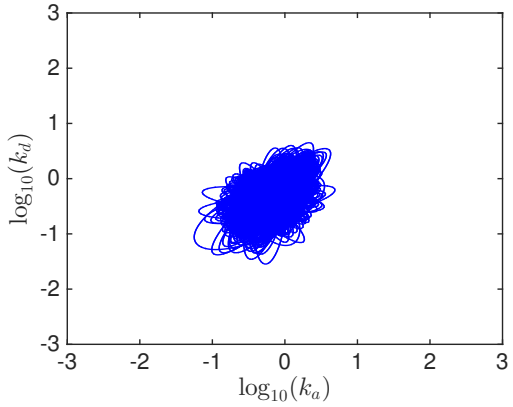
(c) $\tau = 0.2, n = 50$



(d) Acceptable region

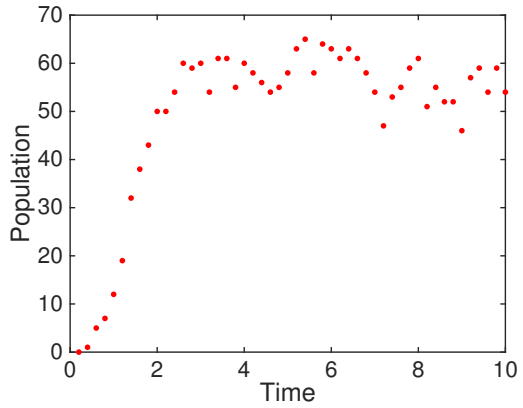


(e) $\tau = 0.05, n = 200$

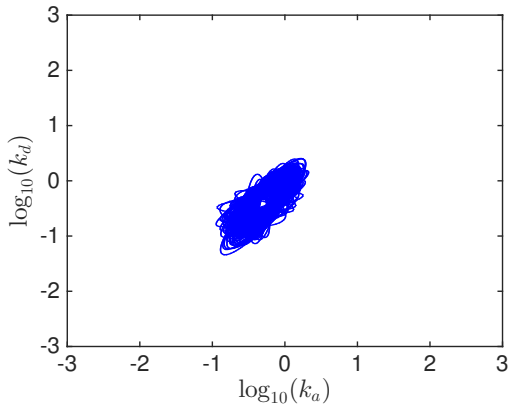


(f) Acceptable region

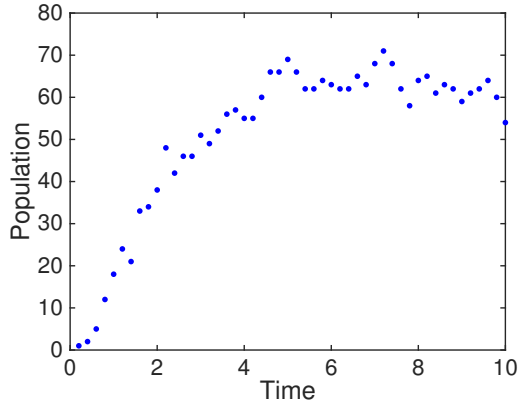
Figure 6.7: Optimization results of maximum log-likelihood with different time steps τ from one set of empirical data: (a, b) $\tau = 1$; (c, d) $\tau = 0.2$; (e, f) $\tau = 0.05$. The acceptable region of each method is the union of results from 20 starting points.



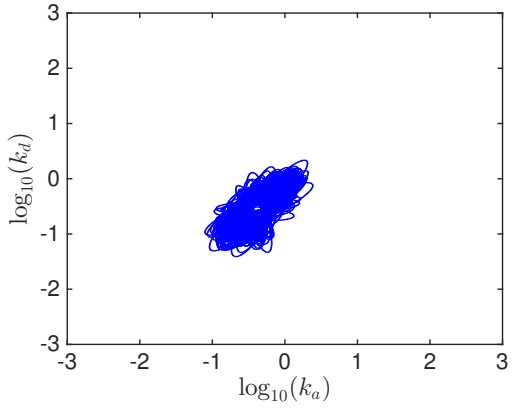
(a) Dataset A



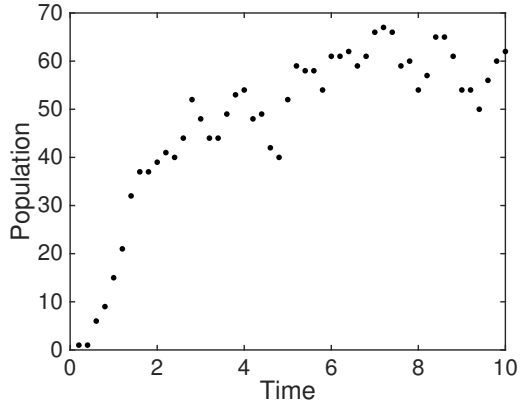
(b) Acceptable region



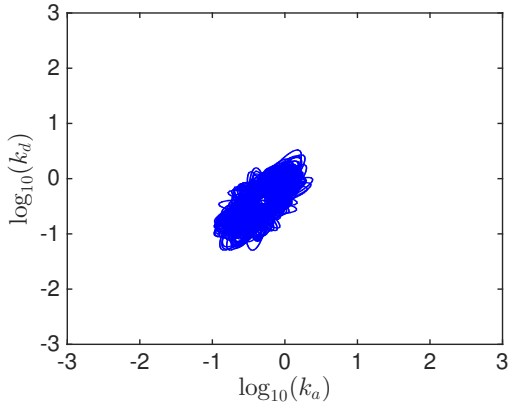
(c) Dataset B



(d) Acceptable region



(e) Dataset C



(f) Acceptable region

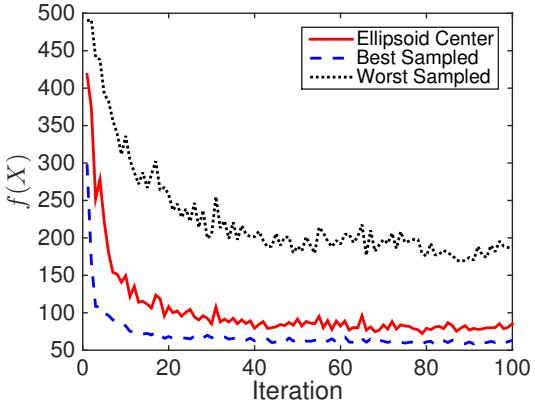
Figure 6.8: Optimization results of maximum log-likelihood with different empirical data: three sets of empirical data, which all contain 50 data points, collected every 0.2 time unit. The acceptable region of each method is the union of results from 20 starting points.

where $[A]$ = 150, 300, 400, 520, 620, 800, 1050, 1400, 2100, 5000, 20000. (This summed objective function will be explored in future work).

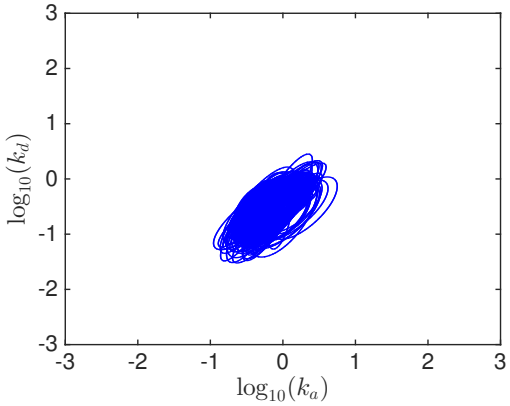
To compare the stochastic Hill equation system with the sequential binding scheme, sample 100 points from the returned acceptable parameter region. Fig. 6.11 shows the average population evolution of B_n with the sampled parameter values in the stochastic Hill equation. The average dynamics from all three methods match well with empirical data where $m = 50$. Fig. 6.12 further presents the population distributions of the stochastic Hill equation system based on the 100 parameter values sampled from the acceptable regions. Except for the significant difference at the initial stage of the B_n transition (time $t = 1$), the empirical data falls in the 25th-75th percentile range for other stages (time $t = 2$, $t = 3$, $t = 6$, $t = 8$, $t = 10$) of maximum log-likelihood and approximate maximum log-likelihood. Note that the range of minimum distance area samples is much smaller than that for the other two methods.

6.5 Conclusion

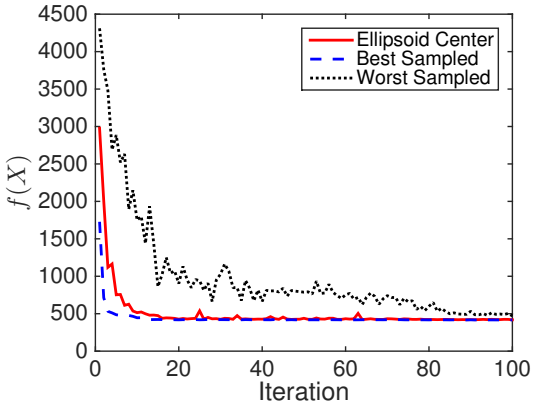
This chapter formulates a stochastic Hill equation model for the fundamental cooperative binding process. To match with the simulation-based empirical data, we explored three different objective functions that emphasize different problem sizes and different features of the empirical data, among which the approximate maximum log-likelihood method works well and can be applicable to large complex biochemical networks. In particular, we proposed an α-β-γ rule to find acceptable parameter regions instead of single best parameter values. Results demonstrated that the optimized stochastic Hill equation can be used to model the switch behavior and the steady state of the fundamental cooperative binding process, while it cannot capture the initial transition period. QNSTOP and this simple rule can be extended and applied to other stochastic models as well.



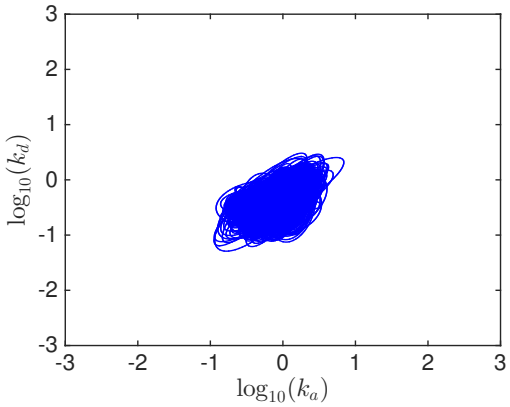
(a) Average execution trace of QNSTOP



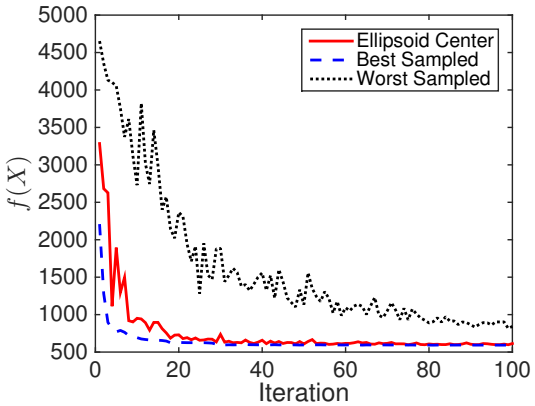
(b) Acceptable region



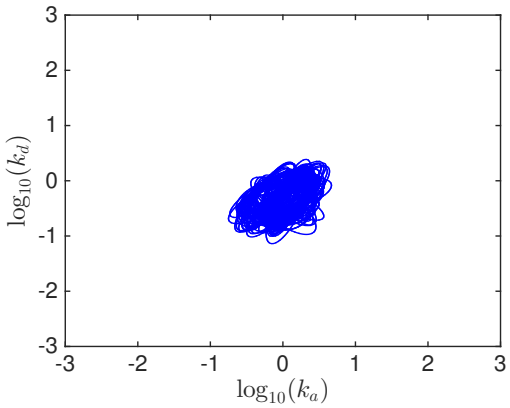
(c) Average execution trace of QNSTOP



(d) Acceptable region

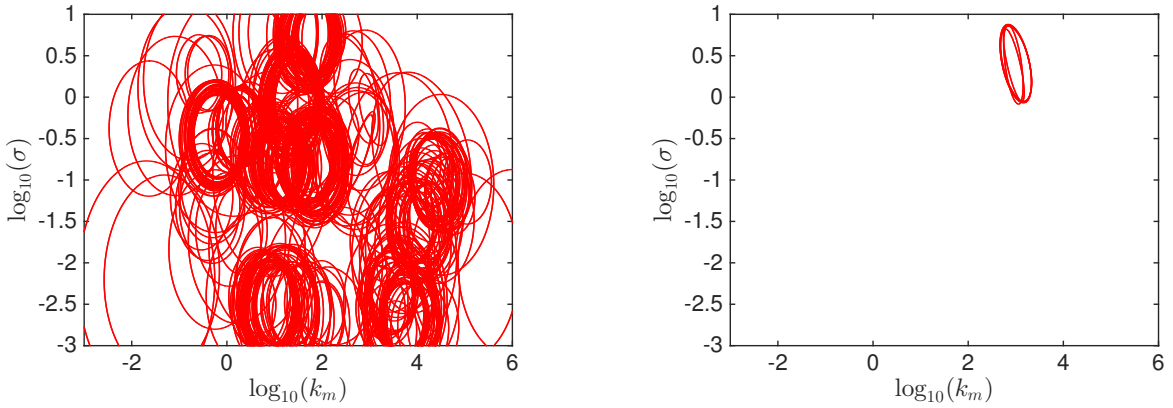


(e) Average execution trace of QNSTOP



(f) Acceptable region

Figure 6.9: Average execution traces of QNSTOP based on 20 starting points and acceptable parameter region projected to two-dimensional domains. (a, b) minimum distance area, (c, d) maximum log-likelihood, (e, f) approximate maximum log-likelihood. The acceptable region of each method is the union of results from 20 starting points.



(a) Acceptable region for k_m, σ

(b) Acceptable region for k_m, σ

Figure 6.10: Acceptable parameter regions projected to two-dimensional domains for maximum log-likelihood method. (a) The population of enzyme A is fixed at a single value. (b) 11 population levels of enzyme A are considered in the stochastic Hill equation system.

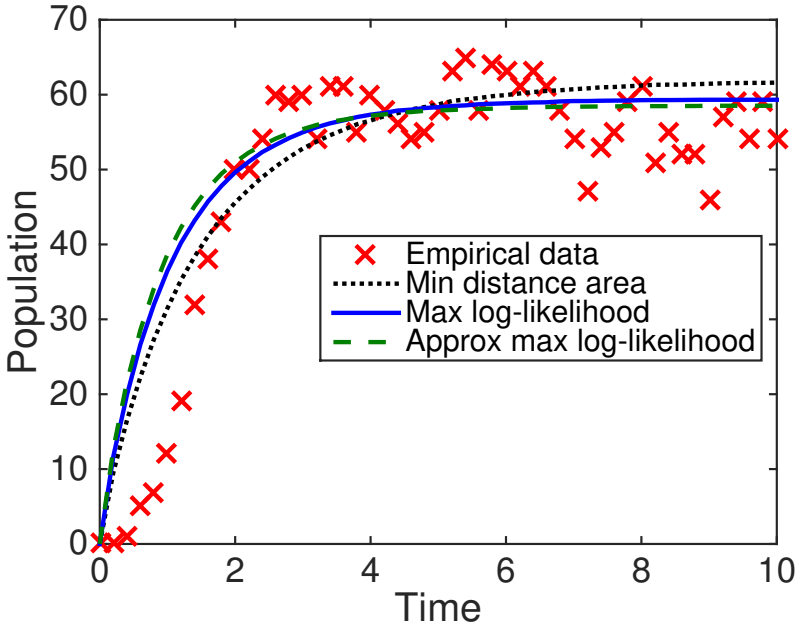
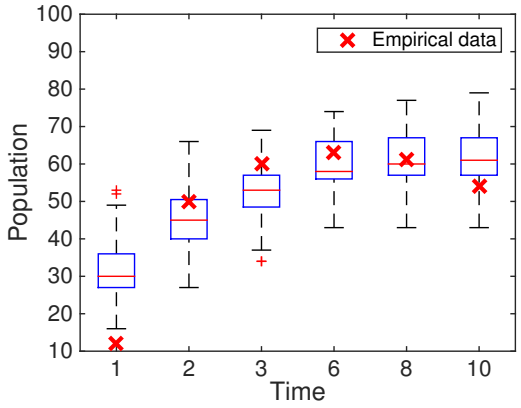
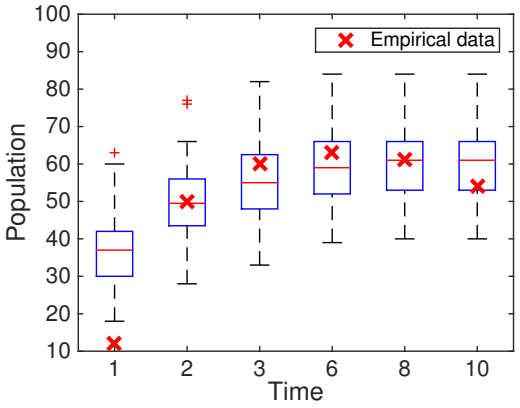


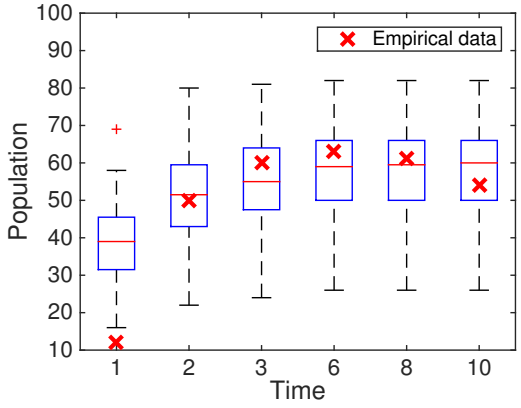
Figure 6.11: Average population evolution of B_n in the stochastic Hill equation system (6.2) from minimum distance area, maximum log-likelihood, and approximate maximum log-likelihood, compared with the empirical data.



(a) Minimum distance area



(b) Maximum log-likelihood



(c) Approximate maximum log-likelihood

Figure 6.12: Population distributions of B_n in the stochastic Hill equation system (6.2) at time $t = 1, 2, 3, 6, 8, 10$, corresponding to 10%, 20%, 30%, 60%, 80%, 100% of the total simulation time, based on 100 points sampled in each acceptable parameter region from minimum distance area, maximum log-likelihood, and approximate maximum log-likelihood.

Chapter 7

Outlook

This dissertation presented a series of theoretical studies and applications that contribute to the field of stochastic modeling and simulation. In the study of Caulobacter cell cycle, Chapter 3 showed that the Turing mechanism, along with the gene position, contributes to protein PopZ's bipolar location. As stochastic simulations are often impeded by expensive computational cost for large and complex biochemical networks, Chapter 4 studied the HR hybrid method which significantly improved the simulation efficiency and proposed several strategies for the negative population phenomenon. To further validate simulation results with empirical data, a quasi-Newton algorithm for stochastic optimization (QNSTOP) was used to optimize system parameters of a stochastic budding yeast cell cycle model in Chapter 5. Finally, to cope with increasing model complexity, Chapter 6 demonstrates the model reduction of the fundamental cooperative binding scheme by a optimized stochastic Hill equation.

7.1 Improvement on HR hybrid method

In Chapter 4, while applying the HR hybrid method to the PleC model of the *Caulobacter crescentus* cell cycle, the simulation time for one cell cycle can be reduced to several hours from the original three days. Based on different partitioning strategies, the time cost varies from about nine hours to one hour. It is easy to see that the partitioning strategy plays

an important role in the simulation efficiency. Sometimes put all fast reactions or large-population species into the fast subsystem is not a good strategy as the ODE solver slows down with a large ODE system size. In order to find the best partitioning strategy for a specific application, researchers may need to define different partition strategies, to set up corresponding simulations, and to test and compare the results, which involves a substantial amount of work.

Meanwhile, as species populations vary with time in most biological systems, the related reactions can be considered slow in certain time period and fast at other time during the simulation. Based on time-varying system states and accuracy requirement, an automatic partition system can help maximize the efficiency of hybrid method. In particular, the partitioning strategy may involve an automatic switch feature where the species and reactions can be put into the different subsystems based on the dynamical state (e.g., population scale) in the evolution. Such an automatic switch has to take the overhead of the ODE solver into account and will need careful study. In our future research, the automatic partitioning and switch mechanism will be studied under the framework of hybrid methods.

7.2 Spatial Stochastic Algorithm

In recent years, stochastic modeling and simulation for spatiotemporal biological systems, particularly reaction-diffusion systems, have captured more and more attention. Several algorithms and tools [7, 56, 76, 125] to model and simulate reaction-diffusion systems have been proposed. These methods can be categorized into two theoretical frameworks: the spatially and temporally continuous Smoluchowski modeling framework [126] and the compartment-based modeling framework, formulated as the spatially discretized reaction-diffusion master equation (RDME) [46, 94]. The Smoluchowski framework [36, 66, 126] stores the exact

position of each molecule and is mathematically fundamental, whereas the RDME is coarse-grained and better suited for large scale simulations [43]. In RDME, the spatial domain is discretized into small compartments. Within each compartment, molecules are considered "well-stirred". Under the RDME scheme, diffusion is modeled as continuous time random walk on mesh compartments, while reactions fire only among molecules in the same compartment.

Yet reaction-diffusion systems presents great challenges regarding accuracy and efficiency, especially when nonlinear reactions are involved. It has been proved that the RDME of bimolecular reactions in 3D domain becomes incorrect and yields nonphysical results when the discretization size approaches microscopic scale [40, 57, 63]. Previous work on trimolecular reaction models in the compartment-based framework also illustrated the accuracy error in a 1D domain [79]. For highly nonlinear reactions, e.g., the Hill function, study revealed that when the compartment size is small enough, the sigmoidal behavior of Hill function dynamics reduces to a linear function of the input signal and discretization size [26]. In general, to establish a well-rounded spatial stochastic modeling and simulation algorithm, there are many problems requiring a resolution via appropriate strategies; finally, unforeseen issues awaiting further exploration may also exist. The long term plan is to develop spatial stochastic simulation algorithms for two-dimensional and three-dimensional biological models that maintain great tradeoff between a high accuracy and a relatively low computational complexity.

7.3 Cell Cycle Visualization

While much attention has been focused on various mathematical modeling and computational simulations, it is hard for people without a biological background to interpret simula-

tion results and to understand the cell cycle dynamics. Moreover, those new spatial models necessitate a visualization tool to illustrate the spatiotemperal simulation results. Fig. 7.1 presents an example of the PopZ dynamics during the cell cycle. Based on the D3 toolkit, this Web-based animation of *Caulobacter crescentus* cell division provides an interactive visualization interface for people to view the spatiotemporal dynamics of species in cells and to check simulation results with experimental observations. It has been extended to the PleC model of the *Caulobacter crescentus* and budding yeast cell cycle [33]. The current project only works with static data (previously generated), so future work is to integrate the simulation process into the visualization so that people can check different molecular behaviors by tuning system parameters. The visualization application provides a significant modeling tool in addition to traditional graphs.

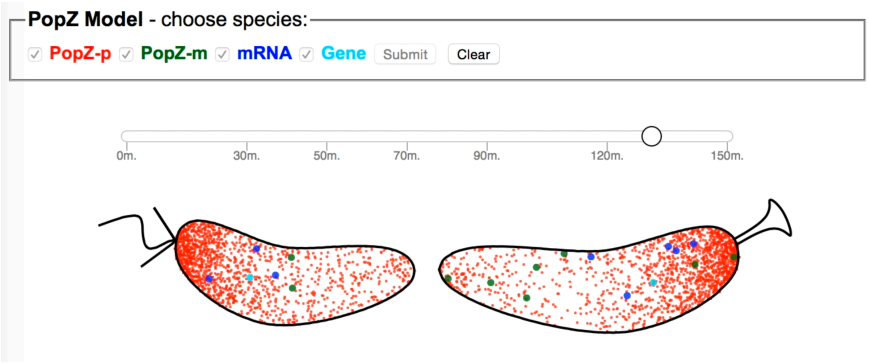


Figure 7.1: Visualization of *Caulobacter crescentus* cell cycle based on simulation results.

Bibliography

[1] Gary K Ackers, Alexander D Johnson, and Madeline A Shea. Quantitative model for gene regulation by lambda phage repressor. *Proceedings of the National Academy of Sciences*, 79(4):1129–1133, 1982.

[2] GS Adair. The hemoglobin system VI. The oxygen dissociation curve of hemoglobin. *Journal of Biological Chemistry*, 63(2):529–545, 1925.

[3] Mansooreh Ahmadian, Shuo Wang, John Tyson, and Young Cao. Hybrid ode/ssa model of the budding yeast cell cycle control mechanism with mutant case study. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 464–473. ACM, 2017.

[4] Mansooreh Ahmadian, John Tyson, and Yang Cao. A stochastic model of size control in the budding yeast cell cycle. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 589–590. ACM, 2018.

[5] Brandon D Amos, David R Easterling, Layne T Watson, William I Thacker, Brent S Castle, and Michael W Trosset. Algorithm XXX: QNSTOP—quasi-Newton algorithm for stochastic optimization. Dept of Computer Sci TR-14-2, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2014.

[6] David F Anderson. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of chemical physics*, 127 (21):214107, 2007.

[7] Steven S Andrews and Dennis Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical biology*, 1(3):137, 2004.

[8] Hironori Aramaki, Hiroyuki Kabata, Shuso Takeda, Hiroshi Itou, Hideki Nakayama, and Nobuo Shimamoto. Formation of repressor-inducer-operator ternary complex: negative cooperativity of d-camphor binding to CamR. *Genes to Cells*, 16(12):1200–1207, 2011.

[9] Maksat Ashyraliyev, Johannes Jaeger, and Joke G Blom. Parameter estimation and determinability analysis applied to Drosophila gap gene circuits. *BMC Systems Biology*, 2(1):83, 2008.

[10] Maksat Ashyraliyev, Yves Fomekong-Nanfack, Jaap A Kaandorp, and Joke G Blom. Systems biology: Parameter estimation for biochemical models. *The FEBS journal*, 276(4):886–902, 2009.

[11] Y Sudhakar Babu, John S Sack, Trevor J Greenhough, Charles E Bugg, Anthony R Means, and William J Cook. Three-dimensional structure of calmodulin. *Nature*, 315 (6014):37, 1985.

[12] Debashis Barik, William T Baumann, Mark R Paul, Bela Novak, and John J Tyson. A model of yeast cell-cycle regulation based on multisite phosphorylation. *Molecular systems biology*, 6(1):405, 2010.

[13] Cara C Boutte, Jonathan T Henry, and Sean Crosson. ppGpp and polyphosphate modulate cell cycle progression in Caulobacter crescentus. *Journal of bacteriology*, 194 (1):28–35, 2012.

[14] Grant R Bowman, Luis R Comolli, Jian Zhu, Michael Eckart, Marcelle Koenig, Kenneth H Downing, WE Moerner, Thomas Earnest, and Lucy Shapiro. A polymeric

protein anchors the chromosomal origin/ParB complex at a bacterial cell pole. *Cell*, 134(6):945–955, 2008.

[15] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[16] Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics*, 121 (9):4059–4067, 2004.

[17] Yang Cao, Dan Gillespie, and Linda Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395–411, 2005.

[18] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Avoiding negative populations in explicit Poisson tau-leaping. *The Journal of chemical physics*, 123(5):054104, 2005.

[19] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of chemical physics*, 122(1):014116, 2005.

[20] Brent S Castle. *Quasi-Newton methods for stochastic optimization and proximity-based methods for disparate information fusion*. PhD thesis, Indiana University, Bloomington, IN, 2012.

[21] Jean-Pierre Changeux. The feedback control mechanisms of biosynthetic L-threonine deaminase by L-isoleucine. *Cold Spring Harb Symp Quant Biol*, 26:313–318, 1961.

[22] Godefroid Charbon, Matthew T Cabeen, and Christine Jacobs-Wagner. Bacterial intermediate filaments: in vivo assembly, organization, and dynamics of crescentin. *Genes and Development*, 23(9):1131–1144, 2009.

[23] Katherine C Chen, Attila Csikasz-Nagy, Bela Gyorffy, John Val, Bela Novak, and John J Tyson. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular biology of the cell*, 11(1):369–391, 2000.

[24] Minghan Chen and Yang Cao. Analysis and remedy of negativity problem in hybrid stochastic simulation algorithm and its application. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 585–586. ACM, 2018.

[25] Minghan Chen, Fei Li, Kartik Subramanian, John Tyson, and Yang Cao. Two-dimensional model of bipolar popz polymerization in caulobacter crescentus. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 37–46. ACM, 2015.

[26] Minghan Chen, Fei Li, Shuo Wang, and Young Cao. Stochastic modeling and simulation of reaction-diffusion system with Hill function dynamics. *BMC systems biology*, 11(3):21, 2017.

[27] Minghan Chen, Yang Cao, and Layne T Watson. Parameter estimation of stochastic models based on limited data. *ACM SIGBioinformatics Record*, 7(3):3, 2018.

[28] Minghan Chen, Shuo Wang, and Yang Cao. Accuracy analysis of hybrid stochastic simulation algorithm on linear chain reaction systems. *Bulletin of mathematical biology*, pages 1–29, 2018.

[29] Minghan Chen, Brandon Amos, Layne T Watson, John Tyson, Yang Cao, Cliff Shaffer, Michael Trosset, Cihan Oguz, and Gisella Kakoti. Quasi-Newton stochastic optimization algorithm for parameter estimation of a stochastic model of the budding yeast cell cycle. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1): 301–311, 2019.

[30] KH Chiam, Chee Meng Tan, Vipul Bhargava, and Gunaretnam Rajagopal. Hybrid simulations of stochastic reaction-diffusion processes for modeling intracellular signaling pathways. *Physical Review E*, 74(5):051910, 2006.

[31] Justine Collier and Lucy Shapiro. Spatial complexity and control of a bacterial cell cycle. *Current opinion in biotechnology*, 18(4):333–40, 2007.

[32] Edmund J Crampin, Eamonn A Gaffney, and Philip K Maini. Reaction and diffusion on growing domains: Scenarios for robust pattern formation. *Bulletin of Mathematical Biology*, 61(6):1093–1120, 1999.

[33] Jing Cui. Visualization of the budding yeast cell cycle. Master's thesis, Virginia Tech, Blacksburg, VA, 2017.

[34] Mark HA Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):353–376, 1984.

[35] Stefano Di Talia, Jan M Skotheim, James M Bean, Eric D Siggia, and Frederick R Cross. The effects of molecular noise and size control on variability in the budding yeast cell cycle. *Nature*, 448(7156):947, 2007.

[36] Masao Doi. Stochastic theory of diffusion-controlled reaction. *Journal of Physics A: Mathematical and General*, 9(9):1479, 1976.

[37] David R Easterling, Layne T Watson, Michael L Madigan, Brent S Castle, and Michael W Trosset. Parallel deterministic and stochastic global minimization of functions with very many minima. *Computational Optimization and Applications*, 57(2): 469–492, 2014.

[38] Gitte Ebersbach, Ariane Briegel, Grant J Jensen, and Christine Jacobs-Wagner. A self-associating protein critical for chromosome attachment, division, and polar organization in Caulobacter. *Cell*, 134(6):956–68, 2008.

[39] Michael B Elowitz, Arnold J Levine, Eric D Siggia, and Peter S Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[40] Radek Erban and S Jonathan Chapman. Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions. *Physical Biology*, 6(4):046001, 2009.

[41] Radek Erban, Jonathan Chapman, and Philip Maini. A practical guide to stochastic simulations of reaction-diffusion processes. *arXiv preprint arXiv:0704.1908*, 2007.

[42] Bard Ermentrout. Stripes or spots? Nonlinear effects in bifurcation of reaction—diffusion equations on the square. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1891):413–417, 1991.

[43] David Fange, Otto G Berg, Paul Sjöberg, and Johan Elf. Stochastic reaction-diffusion kinetics in the microscopic limit. *Proceedings of the National Academy of Sciences*, 107 (46):19820–19825, 2010.

[44] Zachary Fox, Gregor Neuert, and Brian Munsky. Finite state projection based bounds to compare chemical master equation models using single-cell data. *The Journal of chemical physics*, 145(7):074101, 2016.

[45] Uwe Franz, Volkmar Liebscher, and Stefan Zeiser. Piecewise-deterministic Markov processes as limits of Markov jump processes. *Advances in Applied Probability*, 44(3): 729–748, 2012.

[46] CW Gardiner, KJ McNeil, DF Walls, and IS Matheson. Correlations in stochastic theories of chemical reactions. *Journal of Statistical Physics*, 14(4):307–331, 1976.

[47] John C Gerhart and Arthur B Pardee. The enzymology of control by feedback inhibition. *J biol Chem*, 237891, 1962.

[48] Michael A Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A*, 104(9):1876–1889, 2000.

[49] Daniel 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.

[50] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[51] Daniel T Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425, 1992.

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

[53] Daniel T Gillespie. Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem*, 58:35–55, 2007.

[54] Sylvain Goutelle, Michel Maurin, Florent Rougier, Xavier Barbaut, Laurent Bourguignon, Michel Ducher, and Pascal Maire. The Hill equation: a review of its capabilities in pharmacological modelling. *Fundamental & clinical pharmacology*, 22(6): 633–648, 2008.

[55] Eric L Haseltine and James B Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of chemical physics*, 117 (15):6959–6969, 2002.

[56] Johan Hattne, David Fange, and Johan Elf. Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, 21(12):2923–2924, 2005.

[57] Stefan Hellander, Andreas Hellander, and Linda Petzold. Reaction-diffusion master equation in the microscopic limit. *Physical Review E*, 85(4):042901, 2012.

[58] Jonathan T Henry and Sean Crosson. Chromosome replication and segregation govern the biogenesis and inheritance of inorganic polyphosphate granules. *Molecular biology of the cell*, 24(20):3177–3186, 2013.

[59] Archibald Vivian Hill. The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *J Physiol*, 40:4–7, 1910.

[60] Stefan Hoops, Seven Sahle, Ralph Gauges, Christine Lee, Jügen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. COPASI—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[61] CY Huang, V Chau, PB Chock, JH Wang, and RK Sharma. Mechanism of activation of cyclic nucleotide phosphodiesterase: Requirement of the binding of four Ca2+ to calmodulin for activation. *Proceedings of the National Academy of Sciences of the United States of America*, 78(2):871–874, 1981.

[62] Kerwyn Casey Huang, Ranjan Mukhopadhyay, and Ned S Wingreen. A curvature-mediated mechanism for localization of lipids to bacterial poles. *PLoS computational biology*, 2(11):e151, 2006.

[63] Samuel A Isaacson. The reaction-diffusion master equation as an asymptotic approximation of diffusion to a small target. *SIAM Journal on Applied Mathematics*, 70(1): 77–111, 2009.

[64] Tobias Jahnke and Michael Kreim. Error bound for piecewise deterministic processes modeling stochastic reaction systems. *Multiscale Modeling & Simulation*, 10(4):1119–1147, 2012.

[65] Hye-Won Kang, Wasiur R KhudaBukhsh, Heinz Koeppl, and Grzegorz A Rempała. Quasi-steady-state approximations derived from the stochastic model of enzyme kinetics. *Bulletin of mathematical biology*, pages 1–34, 2019.

[66] Joel Keizer. Nonequilibrium statistical thermodynamics and the effect of diffusion on chemical reaction rates. *The Journal of Physical Chemistry*, 86(26):5052–5067, 1982.

[67] Jae Kyoung Kim and Eduardo D Sontag. Reduction of multiscale stochastic biochemical reaction networks using exact moment derivation. *PLoS computational biology*, 13 (6):e1005571, 2017.

[68] Jin Seob Kim and Sean X Sun. Morphology of Caulobacter crescentus and the mechanical role of crescentin. *Biophysical Journal*, 96(8):L47–L49, 2009.

[69] DE Koshland Jr, G Nemethy, and D Filmer. Comparison of experimental binding data and theoretical models in proteins containing subunits. *Biochemistry*, 5(1):365–385, 1966.

[70] Tino Krell, Wilson Terán, Obdulio López Mayorga, Germán Rivas, Mercedes Jiménez, Craig Daniels, Antonio-Jesús Molina-Henares, Manuel Martínez-Bueno, María-Trinidad Gallegos, and Juan-Luis Ramos. Optimization of the palindromic order of the TtgR operator enhances binding cooperativity. *Journal of molecular biology*, 369(5):1188–1199, 2007.

[71] Niels Rode Kristensen, Henrik Madsen, and Sten Bay Jørgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, 2004.

[72] Géraldine Laloux and Christine Jacobs-Wagner. How do bacteria localize proteins to the cell pole? *Journal of Cell Science*, 127(1):11–19, 2014.

[73] Hubert Lam, Jean-Yves Matroule, and Christine Jacobs-Wagner. The asymmetric spatial distribution of bacterial signal transduction proteins coordinates cell cycle events. *Developmental cell*, 5(1):149–159, 2003.

[74] Teeraphan Laomettachit. *Mathematical modeling approaches for dynamical analysis of protein regulatory networks with applications to the budding yeast cell cycle and the circadian rhythm in cyanobacteria*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2011.

[75] Keren Lasker, Thomas H Mann, and Lucy Shapiro. An intracellular compass spatially coordinates cell cycle modules in Caulobacter crescentus. *Current opinion in microbiology*, 33:131–139, 2016.

[76] Nicolas Le Novère and Thomas Simon Shimizu. STOCHSIM: modelling of stochastic biomolecular processes. *Bioinformatics*, 17(6):575–576, 2001.

[77] Paola Lecca, Fabio Bagagiolo, and Marina Scarpa. Hybrid deterministic/stochastic simulation of complex biochemical systems. *Molecular BioSystems*, 13(12):2672–2686, 2017.

[78] Fei Li, Kartik Subramanian, Minghan Chen, John J Tyson, and Yang Cao. A stochastic spatiotemporal model of a response-regulator network in the Caulobacter crescentus cell cycle. *Physical biology*, 13(3):035007, 2016.

[79] Fei Li, Minghan Chen, Radek Erban, and Yang Cao. Reaction time for trimolecular reactions in compartment-based reaction-diffusion models. *The Journal of chemical physics*, 148(20):204108, 2018.

[80] Hong Li and Linda Petzold. Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. Dept. of computer sci., University of California Santa Barbara, Santa Barbara, CA, 2006.

[81] Gabriele Lillacci and Mustafa Khammash. Parameter estimation and model selection in computational biology. *PLoS computational biology*, 6(3):e1000696, 2010.

[82] Xin Liu and Mahesan Niranjan. State and parameter estimation of the heat shock response system using Kalman and particle filters. *Bioinformatics*, 28(11):1501–1507, 2012.

[83] Xin Liu and Mahesan Niranjan. Parameter estimation in computational biology by approximate Bayesian computation coupled with sensitivity analysis. *arXiv preprint arXiv:1704.09021*, 2017.

[84] Zhen Liu, Yang Pu, Fei Li, Clifford A Shaffer, Stefan Hoops, John J Tyson, and Yang Cao. Hybrid modeling and simulation of stochastic effects on progression through the eukaryotic cell cycle. *The Journal of chemical physics*, 136(3):034105, 2012.

[85] Wing-Cheong Lo, Likun Zheng, and Qing Nie. A hybrid continuous-discrete method for stochastic reaction-diffusion processes. *Royal Society open science*, 3(9):160485, 2016.

[86] Philip K Maini, Thomas E Woolley, Ruth E Baker, Eamonn A Gaffney, and S Seirin Lee. Turing's model for biological pattern formation and the robustness problem. *Interface focus*, 2(4):487–496, 2012.

[87] James M McCollum, Gregory D Peterson, Chris D Cox, Michael L Simpson, and Nagiza F Samatova. The sorting direct method for stochastic simulation of biochemical

systems with varying reaction execution behavior. *Computational biology and chemistry*, 30(1):39–49, 2006.

[88] Donald A McQuarrie. Stochastic approach to chemical kinetics. *Journal of applied probability*, 4(3):413–478, 1967.

[89] Pedro Mendes and Douglas Kell. Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics (Oxford, England)*, 14(10):869–883, 1998.

[90] Jacque Monod, Jeffries Wyman, and Jean-Pierre Changeux. On the nature of allosteric transitions: a plausible model. *J Mol Biol*, 12(1):88–118, 1965.

[91] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of chemical physics*, 124(4): 044104, 2006.

[92] JD Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Springer-Verlag, Berlin, 1993.

[93] MR Myerscough and JD Murray. Analysis of propagating pattern in a chemotaxis system. *Bulletin of mathematical biology*, 54(1):77–94, 1992.

[94] Gregoire Nicolis and Ilya Prigogine. *Self-organization in nonequilibrium systems: From dissipative structures to order through fluctuations*. A Wiley-Interscience Publication. J. Wiley and sons, New York, London, Sydney, 1977. ISBN 0-471-02401-5.

[95] Cihan Oguz, Teeraphan Laomettachit, Katherine C Chen, Layne T Watson, William T Baumann, and John J Tyson. Optimization and model reduction in the high dimensional parameter space of a budding yeast cell cycle model. *BMC systems biology*, 7 (1):53, 2013.

[96] Cihan Oguz, Alida Palmisano, Teeraphan Laomettachit, Layne T Watson, William T Baumann, and John J Tyson. A stochastic model correctly predicts changes in budding yeast cell cycle dynamics upon periodic expression of CLN2. *PLoS one*, 9(5):e96726, 2014.

[97] Linus Pauling. The oxygen equilibrium of hemoglobin and its structural interpretation. *Proceedings of the National Academy of Sciences of the United States of America*, 21 (4):186, 1935.

[98] Linda R Petzold. Description of DASSL: A differential/algebraic system solver. Technical report, Sandia National Labs., Livermore, CA (USA), 1982.

[99] Jeanne S Poindexter. The caulobacters: ubiquitous unusual bacteria. *Microbiological reviews*, 45(1):123–79, 1981.

[100] Suresh Kumar Poovathingal and Rudiyanto Gunawan. Global parameter estimation methods for stochastic biochemical systems. *BMC bioinformatics*, 11(1):414, 2010.

[101] Mark Ptashne, Andrea Jeffrey, Alexander D Johnson, Russell Maurer, Barbara J Meyer, Carl O Pabo, Thomas M Roberts, and Robert T Sauer. How the λ repressor and cro work. *Cell*, 19(1):1–11, 1980.

[102] Nicholas R Radcliffe, David R Easterling, Layne T Watson, Michael L Madigan, and Kathleen A Bieryla. Results of two global optimization algorithms applied to a problem in biomechanics. In *Proceedings of the 2010 Spring Simulation Multiconference*, page 86. Society for Computer Simulation International, 2010.

[103] Arjun Raj and Alexander van Oudenaarden. Single-molecule approaches to stochastic gene expression. *Annual review of biophysics*, 38:255–270, 2009.

[104] Kumaran S Ramamurthi and Richard Losick. Negative membrane curvature as a cue for subcellular localization of a bacterial protein. *Proceedings of the National Academy of Sciences of the United States of America*, 106(32):13541–13545, 2009.

[105] Christopher V Rao and Adam P Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption:Application to the gillespie algorithm. *The Journal of chemical physics*, 118(11):4999–5010, 2003.

[106] Stefan Reinker, Rachel M Altman, and Jens Timmer. Parameter estimation in stochastic biochemical reactions. *IEE Proceedings-Systems Biology*, 153(4):168–178, 2006.

[107] Matthew L Robb and Vahid Shahrezaei. Stochastic cellular fate decision making by multiple infecting lambda phage. *PLoS one*, 9(8):e103636, 2014.

[108] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Science & Business Media, 2013.

[109] Diego Rossinelli, Basil Bayati, and Petros Koumoutsakos. Accelerated stochastic and hybrid methods for spatial simulations of reaction-diffusion systems. *Chemical Physics Letters*, 451(1-3):136–140, 2008.

[110] David Z Rudner and Richard Losick. Protein subcellular localization in bacteria. *Cold Spring Harbor Perspectives in Biology*, 2(4):a000307, 2010.

[111] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[112] Carlos Salazar and Thomas Höfer. Multisite protein phosphorylation–from molecular mechanisms to kinetic models. *The FEBS journal*, 276(12):3177–3198, 2009.

[113] Howard Salis and Yiannis Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *The Journal of chemical physics*, 122(5): 054103, 2005.

[114] Howard Salis, Vassilios Sotiropoulos, and Yiannis N Kaznessis. Multiscale Hy3S: Hybrid stochastic simulation for supercomputers. *BMC bioinformatics*, 7(1):93, 2006.

[115] Kevin R Sanft, Daniel T Gillespie, and Linda R Petzold. Legitimacy of the stochastic Michaelis-Menten approximation. *IET systems biology*, 5(1):58–69, 2011.

[116] Lee A Segel and Julius L Jackson. Dissipative structure: An explanation and an ecological example. *Journal of Theoretical Biology*, 37(3):545–559, 1972.

[117] Alexander Slepoy, Aidan P Thompson, and Steven J Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The journal of chemical physics*, 128(20):05B618, 2008.

[118] Oleksii Sliusarenko, Jennifer Heinritz, Thierry Emonet, and Christine Jacobs-Wagner. High-throughput, subpixel-precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. *Molecular microbiology*, 80(3):612–627, 2011.

[119] Melanie I. Stefan and Nicolas Le Novère. Cooperative binding. *PLoS computational biology*, 9(6):e1003106, 2013.

[120] Kartik Subramanian, Fei Li, Mark R Paul, Yang Cao, and John J Tyson. Spatiotemporal model of PopZ localization in Caulobacter crescentus. *Submitted to PLoS Computational Biology*.

[121] Thapanar Suwanmajo and Jeevanithya Krishnan. Mixed mechanisms of multi-site phosphorylation. *Journal of The Royal Society Interface*, 12(107):20141405, 2015.

[122] Martin Thanbichler and Lucy Shapiro. MipZ, a spatial regulator coordinating chromosome segregation with cell division in Caulobacter. *Cell*, 126(1):147–162, 2006.

[123] Philipp Thomas, Arthur V Straube, and Ramon Grima. Communication: limitations of the stochastic quasi-steady-state approximation in open biochemical reaction networks, 2011.

[124] Alan Mathison Turing. The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52(1-2):153–197, 1990.

[125] Jeroen S. van Zon and Pieter Rein ten Wolde. Green's-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space. *The Journal of Chemical Physics*, 123(23):234910, 2005.

[126] Marian Von Smoluchowski. Zur kinetischen theorie der brownschen molekularbewegung und der suspensionen. *Annalen der physik*, 326(14):756–780, 1906.

[127] Shuo Wang and Yang Cao. The abridgement and relaxation time for a linear multi-scale model based on multiple site phosphorylation. *PLoS ONE*, 10(8):e0133295, 2015.

[128] Shuo Wang, Mansooreh Ahmadian, Minghan Chen, John Tyson, and Young Cao. A hybrid stochastic model of the budding yeast cell cycle control mechanism. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 261–270. ACM, 2016.

[129] Shuo Wang, Minghan Chen, Layne T Watson, and Yang Cao. Efficient implementation of the hybrid method for stochastic simulation of biochemical systems. *Journal of Micromechanics and Molecular Physics*, 02(02):1750006, 2017.

[130] James N Weiss. The Hill equation revisited: uses and misuses. *The FASEB Journal*, 11(11):835–841, 1997.

[131] John N Werner, Eric Y Chen, Jonathan M Guberman, Angela R Zippilli, Joseph J Irgon, and Zemer Gitai. Quantitative genome-scale analysis of protein localization in an asymmetric bacterium. *Proceedings of the National Academy of Sciences*, 106(19): 7858–7863, 2009.

[132] Darren J Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in bioinformatics*, 8(2):109–116, 2007.

[133] Juliane Winkler, Anja Seybert, Lars Kö nig, Sabine Pruggnaller, Uta Haselmann, Victor Sourjik, Matthias Weiss, Achilleas S Frangakis, Axel Mogk, and Bernd Bukaub. Quantitative and spatio-temporal features of protein aggregation in Escherichia coli and consequences on protein quality control and cellular ageing. *The EMBO Journal*, 29(5):910–923, 2010.

[134] Christoph Zechner, Jakob Ruess, Peter Krenn, Serge Pelet, Matthias Peter, John Lygeros, and Heinz Koeppl. Moment-based inference predicts bimodality in transient gene expression. *Proceedings of the National Academy of Sciences*, 109(21):8340–8345, 2012.

[135] Jingwei Zhang, Layne T Watson, and Yang Cao. Adaptive aggregation method for the chemical master equation. In *2008 8th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–6. IEEE, 2008.

[136] Jingwei Zhang, Layne T Watson, Christopher A Beattie, and Yang Cao. Radial basis function collocation for the chemical master equation. *International Journal of Computational Methods*, 7(03):477–498, 2010.

[137] Jingwei Zhang, Layne T Watson, and Yang Cao. A modified uniformization method for the solution of the chemical master equation. *Computers & Mathematics with Applications*, 59(1):573–584, 2010.

[138] Mei Zhu and JD Murray. Parameter domains for spots and stripes in mechanical models for biological pattern formation. *Journal of Nonlinear Science*, 5(4):317–336, 1995.