DYNAMIC PARTITIONING OF STOCHASTIC NETWORKS OF MOLECULAR INTERACTIONS

BY

MARK R. GRIFFITH

B.S., University of Illinois at Urbana-Champaign, 2003

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois

ABSTRACT

The stochastic kinetics of a well-mixed chemical system, governed by the chemical master equation, can be simulated using the exact methods of Gillespie. However, these methods do not scale well as systems become more complex and larger models are built to include reactions with widely varying rates, since the computational burden of simulation increases with the number of reaction events. Continuous models may provide an approximate solution and are computationally less costly, but they fail to capture the stochastic behavior of small populations of macromolecules.

In this paper we present a hybrid simulation algorithm that dynamically partitions the system into subsets of continuous and discrete reactions, approximates the continuous reactions deterministically as a system of ordinary differential equations (ODE), and uses a Monte Carlo method for generating discrete reaction events according to a time-dependent propensity. Our approach to partitioning is novel in that we partition the system of reactions, in an online and dynamic manner, based on a threshold relative to the distribution of propensities in the discrete subset. We have implemented the hybrid algorithm in an extensible framework, utilizing two rigorous ODE solvers to approximate the continuous reactions, and use an example model to illustrate the accuracy and potential speedup of the algorithm when compared to exact stochastic simulation.

To my family for being the one constant in my life.

To anyone who ever believed or invested in me,

Thank you.

ACKNOWLEDGMENTS

There are a number of people who have contributed, in some manner, to the completion of this thesis. I am grateful to you all for your support and guidance.

First and foremost, I would like to thank my advisor, Professor William H. Sanders, for his technical advice and for all the opportunities I was afforded while working in his research group. Second, I would like to thank Tod Courtney for his invaluable contributions and dedication to this work. His experience and insight were critical, and his addition to the team was undoubtedly the catalyst to my success. I am thankful for the patience exhibited by them both, particularly during periods of frustration.

I would especially like to thank Jean Peccoud at Pioneer Hi-Bred International for his collaboration, and for providing the motivation and, in large part, the financial support for this work. His guidance and experience with problems in the biological domain were certainly valuable assets. I would also like to thank Leor Weinberger at Princeton University for his interactions, including the sharing of simulation code and models and personal discussions involving the Tat transactivation example. Thanks also to Kaustubh Joshi for his preliminary work in hybrid systems research, on which this thesis was based, as well as his continued encouragement and help.

In addition, other members of the PERFORM group deserve recognition. Jenny Applequist was instrumental in the editing and preparation of this thesis, and I owe a debt of gratitude to her for her assistance and general knowledge in this and all other matters. Also, thanks goes to Michael McQuinn for his contribution in data collection and analysis. Many of the figures that appear in this thesis were made possible because of his work. I would

also like to acknowledge all of the members of the PERFORM group, past and present, with whom I worked. Their moral and technical support, both inside and outside the office, are greatly appreciated.

Finally, I would be remiss not to mention several people who have helped me immeasurably in a nontechnical manner. I would like to thank my parents, to whom I am forever indebted, and my sister, Lisa, for always being there. I would like to thank Cassidy Groth for her love and support throughout, especially during stressful times, and my many other friends, without whom this would have been considerably more difficult.

TABLE OF CONTENTS

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER 1 INTRODUCTION 1
1.1 Introduction 1
1.2 System Specification 3

CHAPTER 2 THEORY 6
2.1 Dynamic Partitioning Scheme 6
2.2 Continuous Reactions 8
2.3 Discrete Reactions 9
2.4 Comparison with Related Work 10
2.4.1 Partitioning 10
2.4.2 Continuous reactions 11
2.4.3 Discrete reactions 12

CHAPTER 3 METHODS 13
3.1 Algorithm 13
3.2 Implementation 14
3.3 Validity of Confidence Intervals 15

CHAPTER 4 RESULTS AND DISCUSSION 18
4.1 HIV-1 Tat Transactivation Example 18
4.2 Cycle Test 22
4.3 Simple Crystallization Example 23
4.4 Viral Infection Example 25

CHAPTER 5 CONCLUSIONS AND FUTURE WORK 32

APPENDIX A TAT TRANSACTIVATION REACTIONS AND PARAMETERS . . 34

APPENDIX B SUMMARY OF RESULTS 36

REFERENCES 38

LIST OF TABLES

4.1 Model parameters for the cycle test model. 23
4.2 Model parameters for the simple crystallization system. 24
4.3 Model parameters for the viral infection example. 28

A.1 Model parameters for the Tat transactivation example with initial GFP concentrations corresponding to a *Dim* sort. 35
A.2 Model parameters for the Tat transactivation example with initial GFP concentrations corresponding to a *Mid* sort. 35

B.1 A summary of performance results for several benchmark models comparing the stochastic and hybrid simulation algorithms. Average number of discrete events and CPU time, in seconds, are reported per run. 37
B.2 Stochastic/hybrid relative reduction in simulation cost for the benchmarks above. 37

LIST OF FIGURES

4.1 A comparison of the time evolution of the mean (center line) and standard deviation ($\pm\sigma$) of GFP for discrete stochastic, hybrid, and deterministic ODE simulation. Results are based on 1000 trials. 20

4.2 The distribution of GFP molecules at time $t = 10^6$ s as computed by exact stochastic simulation (left) and hybrid simulation (right) for the *Mid* sort variation of the Tat transactivation model. Results are based on 10 000 simulations. 20

4.3 A comparison of the CPU time per run of the hybrid algorithm versus the baseline stochastic simulation, as the solution parameter γ is varied, for the *Dim* sort variation of the Tat transactivation model. Relative speedup is measured as the ratio of the computational run time of stochastic simulation to that of hybrid simulation. 22

4.4 A comparison of the running times for different values of Θ in the cycle test model using discrete stochastic simulation (diamonds) and hybrid simulation (squares). Results are based on 10 000 trials. 23

4.5 Comparison of the results of hybrid simulation (points) to those of exact stochastic simulation (lines) based on 10 000 trials of the crystallization model with $\Theta = 10^6$. 25

4.6 Errors in mean (top) and variance (bottom) of species A (left) and C (right) for 100 runs of the crystallization model with $\Theta = 10^6$ (diamonds), 10^7 (squares), 10^8 (triangles), and 10^9 ($\times$'s). Errors for species C are plotted with $\Theta = 10^8$ and 10^9 on the secondary axis. 26

4.7 A comparison of the running times for different values of $\Theta = \mathrm{A}_0$ in the crystallization model using discrete stochastic simulation (diamonds) and hybrid simulation (squares). 27

4.8 The distribution of the template species at time $t = 200$ days as computed by exact stochastic simulation. 28

4.9 The distribution of the template species at time $t = 200$ days as computed by hybrid simulation. 29

4.10 A comparison of the time evolution of the mean (center line) and standard deviation ($\pm\sigma$) of the template and genome species for discrete stochastic (solid lines), hybrid with $\gamma = 10$ (dashed lines), and deterministic ODE (points) simulation. 29

4.11 The reduction in computational expense of the hybrid algorithm over stochastic simulation for the viral infection model, as the solution parameter γ is varied. Relative speedup is measured as the ratio of the computational run time of stochastic simulation to that of hybrid simulation. 30

CHAPTER 1

INTRODUCTION

1.1 Introduction

A number of investigators have demonstrated a relationship between phenotypic variability and the noisy dynamics of populations of macromolecules controlling gene expression [1–8]. These observations led to a renewed interest in simulation algorithms suitable to modeling molecular noise. Specifically, many algorithms and tools for the stochastic solution of well-mixed chemical systems have been developed, beginning with the work of Gillespie [9]. That work outlined two variants of the stochastic simulation algorithm, the Direct method and the First Reaction method, which permit the Monte Carlo generation of stochastic trajectories of systems composed of coupled chemical reactions. However, as physical systems become larger and interacting molecular species appear in greater numbers, these algorithms often require unacceptably large computational resources. In particular, a system (e.g., a genetic network [10]) consisting of some species with low copy numbers (e.g., a gene) and some species with high copy numbers (e.g., proteins) could have reactions whose rates differ by several orders of magnitude. Moreover, the small populations lessen the accuracy of continuous approximations, while the large populations make discrete stochastic simulation impractical. To be of value to scientific research, methods must accommodate the stiffness of the networks of molecular interactions that regulate many aspects of the cell developmental and physiological processes.

Several recent efforts have been directed toward this end. Gibson and Bruck [11] created the Next Reaction variant of the stochastic simulation algorithm, a modification to the First Reaction method, by using special data structures and more efficient random number generation. Gillespie [12] approximated the system as a continuous-state Markov process, and then derived the chemical Langevin equation to specify how the trajectories of the state evolve continuously with stochastic fluctuations. Gillespie [13] presented the "τ-leap" method, an approximate technique for accelerating stochastic simulation, in which it is possible to eliminate the occurrence of some fast reactions by taking time steps that are larger than a single reaction. An improved procedure for selecting the value of τ has also been presented [14]. By applying the quasi-steady-state assumption to the Gillespie algorithm, Rao and Arkin [15] explored an approximation technique designed to reduce the computational load by simulating a simplified stochastic model.

All these methods are based on a monolithic representation of the chemical system; that is, all parts of the system are modeled either discretely or continuously. However, by describing different parts of the system with different appropriate representations, one is able to leverage the advantages of both discrete and continuous solutions. Specifically, a hybrid approach would be less computationally expensive than exact discrete-event simulation and more accurate than continuous approximations, while still preserving the stochastic nature of the model where it is important. A hybrid method also leads to models in which the sources of stochastic behavior are more transparent, as the nonstochastic components are abstracted away [16]. Takahashi et al. [17] developed an efficient method to simulate higher-order models consisting of components driven by different algorithms and timescales. Hybrid algorithms have also been proposed [18, 19] that partition the system into subsets of "fast" and "slow" reactions, approximate the "fast" reactions either deterministically as ODE or stochastically using the chemical Langevin equation, and simulate the "slow" reactions as stochastic events with time-dependent propensities.

This thesis expands upon the ideas developed by those researchers, and highlights a

unique partitioning approach. Instead of partitioning the system into "fast" and "slow" reactions based on reaction propensity alone, we note that, due to small population sizes, it may not be valid to approximate a "fast" reaction continuously. We have developed a unique partitioning algorithm that dynamically classifies reactions as discrete by default, and continuous only when it is both theoretically possible and practically beneficial. In our algorithm, we first partition reactions into subsets of continuous and discrete reactions based on which approximation is more valid for the population of reactants and products involved. We then restrict the continuous subset to those reactions whose propensity is some constant factor times larger than the propensity of the fastest reaction in the discrete subset. Thus, the reaction rates are partitioned not according to an absolute threshold, but rather to a relative threshold that updates dynamically throughout the course of the simulation. The goal of this approach is to guarantee that the reactions modeled continuously are indeed significantly faster than the reactions modeled discretely, thus ensuring that the computational gain of eliminating the "fast" reactions overcomes the cost associated with switching between the deterministic and stochastic regimes, as well as the overhead involved in partitioning. Results from an example model indicate that our algorithm reproduces an accurate solution and is faster than discrete stochastic simulation, so long as the reaction propensities in the discrete and continuous subsets vary by some orders of magnitude. In addition, there may be a tradeoff between the accuracy desired and the speedup provided by the algorithm.

1.2 System Specification

In general, the mass action law formalizes the notion that as the size of a system increases, the frequency of events within the system also increases. Specifically, with application to a system of chemical reactions, the mass action law dictates that the rate of reactions is proportional to the concentrations of the reacting species. In this context, consider a system

of r coupled biochemical reactions involving M species in a well-mixed volume, V. The i^{th} reaction R_i can be written as

$$\sum_{j=1}^{M} \alpha_{i,j} Y_j \xrightarrow{k_i} \sum_{j=1}^{M} \beta_{i,j} Y_j \qquad i = 1, 2, \ldots, r$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are, respectively, the number of molecules of species Y_j consumed and produced by reaction R_i, and k_i is the kinetic coefficient. The $r \times M$ stoichiometric matrix is defined by $\nu_{i,j} = \beta_{i,j} - \alpha_{i,j}$.

The state of the system at time t is given by $X(t)$, an M-vector of nonnegative integers representing the number of molecules of each species. For readability we will use X to mean $X(t)$, and X_j to mean the j^{th} component of X. The propensity of the i^{th} reaction as a function of the state vector X, denoted by $\lambda_i(X)dt$, is the probability that the reaction will occur in the interval $[t, t+dt)$. While these propensities may be computed using different rate laws, our implementation expands on the work by Joshi et al. [20] and focuses specifically on mass action kinetics, under which Gillespie has shown the propensity to be a function of the kinetic coefficient, the populations of the reactants, and a combinatorial term capturing the number of reacting configurations [9]. That is,

$$\lambda_i(X) = \frac{k_i}{V^{(\sum_{j=1}^{M} \alpha_{i,j})-1}} \prod_{j=1}^{M} \frac{X_j!}{(X_j - \alpha_{i,j})!} \tag{1.1}$$

The system can then be modeled using a continuous-time Markov chain (CTMC), where, given the system is in state X, each reaction R_i fires after an exponentially distributed random delay whose rate is given by $\lambda_i(X)$.

As the population size increases (i.e., as $X_j \to \infty,\ j = 1, \ldots, M$), under the mass action law, the dynamics of the system can be described by the following system of differential equations:

$$\lim_{X_1,\ldots,X_M \to \infty} \frac{dX_j}{dt} = \sum_{i=1}^{r} \nu_{i,j} k_i \prod_{k=1}^{M} [X_k]^{\alpha_{i,k}} \tag{1.2}$$

where $[X_k] = X_k/V$ is the concentration of species Y_k in the limit. A first-order correction for stochastic fluctuations can be introduced to obtain the chemical Langevin equation [12], but in this thesis, we choose to approximate the continuous dynamics using deterministic ordinary differential equations. This will be justified later in Section 2.4.2.

CHAPTER 2

THEORY

2.1 Dynamic Partitioning Scheme

The key to any hybrid simulation algorithm is its approach to partitioning. In our algorithm, the system of reactions is dynamically partitioned into two subsets, C and D, representing the continuous and discrete reactions respectively. We partition using a combination of population- and propensity-based approaches. In particular, our scheme chooses for C the largest set such that $\forall R_j \in C$, the following conditions are met:

$$X_i > \gamma * |\nu_{j,i}| \qquad \forall i, i = \{\text{reactant or product of } R_j\} \tag{2.1}$$

and

$$\lambda_j(X) \geq \Lambda * \lambda_{max} \tag{2.2}$$

where γ and Λ are nonnegative constants, and λ_{max} is the rate of the fastest reaction currently classified as discrete. Condition (2.2) is what makes our partitioning scheme unique. Unlike schemes in previous work, the reaction rate threshold in (2.2) is not absolute, but depends on the distribution of reaction propensities in the discrete regime. We conducted a detailed analysis of simulation traces for different classes of models that showed there must be a minimum separation between the reaction rates in the two regimes in order to overcome the cost of switching between a deterministic and a stochastic solution. A similar recommen-

dation has been made by Haseltine and Rawlings [18], but in an offline setting. Ideally, the distribution of reaction rates would be clustered into two distinct groups separated by a large gap, and hybrid simulation would classify the "fast" reactions as continuous and the "slow" reactions as discrete by choosing an appropriate threshold somewhere in the gap. For simple systems in which small populations participate in "slow" reactions and large populations participate in "fast" reactions, this may be accomplished by a naïve partitioning scheme based solely on reaction propensities. However, for more complex systems, that may not be possible, as there may exist reactions that must be classified as discrete (e.g., they involve species with low copy numbers) in order to preserve their stochastic behavior, but may, due to other reactant populations or kinetic constants, be "fast" and in some cases actually have larger propensities than the reactions labeled continuous. This is the situation we attempt to address in our partitioning scheme, and will investigate further in Chapter 4.

Thus condition (2.1) first divides the set of reactions into a tentative partitioning based on the populations of reactant and product species. Then condition (2.2) revises the partitioning in an iterative fashion so that the propensity of any continuous reaction will be at least Λ times larger than the propensity of the fastest discrete reaction. As a result of condition (2.2), therefore, the partitioning process iterates until a fixed point is reached. The following algorithm, which is a component of our general hybrid simulation algorithm outlined in Section 3.1, illustrates how the partitioning is done.

1. Set $\lambda_{max} = 0$, $C = D = \emptyset$
2. For each reaction R_j
   (a) If $X_i \leq \gamma * |\nu_{j,i}|$ for some reactant or product of R_j, then
      i. $D = D \cup \{R_j\}$
      ii. $\lambda_{max} = \max\{\lambda_j, \lambda_{max}\}$
   (b) else $C = C \cup \{R_j\}$

3. If $C = \emptyset$ or $D = \emptyset$ then stop. Reactions are either all discrete or all continuous.

4. Set $fxdpt = true$

5. For each $R_j \in C$

   (a) If $\lambda_j < \Lambda * \lambda_{max}$

      i. $C = C - \{R_j\}$

      ii. $D = D \cup \{R_j\}$

      iii. $\lambda_{max} = \max\{\lambda_j, \lambda_{max}\}$

      iv. Set $fxdpt = false$

6. If $fxdpt = false$ goto step 4.

2.2 Continuous Reactions

Given a system of reactions partitioned into subsets C and D, and under the assumption that rate laws such as mass action or Michaelis-Menten kinetics [21] apply, we approximate the subset of continuous reactions as a continuous Markov process and use the following system of ODE to model the evolution of the system as affected by only these reactions.

$$dX_i = \sum_{j:R_j \in C} \nu_{j,i} \lambda_j(X) dt \tag{2.3}$$

The approximation is accurate when conditions (2.1) and (2.2) are satisfied by the continuous reactions, which we dynamically evaluate to determine if a repartitioning of the system is necessary. In fact, in the thermodynamic limit as the system size tends to infinity, the chemical master equation produces the same process as the solution of the ODEs, and the approximation becomes exact.

ODEs are fairly simple computationally. They can be solved using a variety of numerical methods, from the Euler method to higher-order Runge-Kutta methods, many of which are

readily available in software packages that can easily be incorporated into existing simulation code.

2.3 Discrete Reactions

Having considered the simulation of the continuous reactions, we now describe how the discrete reaction events are simulated. Since the propensities of the discrete reactions depend on the state changes due to the continuous reactions, one can think of the discrete reactions as being represented by a time-dependent Markov process [11]. Gillespie [22] has given a general method for exact stochastic simulation of such a process, and derived the joint probability density function $P(\tau, j)$ for the reaction that occurs next and the time when it occurs. Conditioning $P(\tau, j)$ gives the time-dependent probability density $P(\tau)$ of the Direct variant of the stochastic simulation algorithm as

$$P(\tau) = \lambda_{tot}(X(t+\tau)) \exp\left(- \int_t^{t+\tau} \lambda_{tot}(X(t'))dt' \right) \tag{2.4}$$

where

$$\lambda_{tot}(X(t)) = \sum_{j:R_j \in D} \lambda_j(X(t)) \tag{2.5}$$

Note that if λ_{tot} is constant in time, Equation (2.4) becomes the density function of an exponential distribution with rate λ_{tot}. Given that a reaction occurs in time τ, the probability that reaction R_j occurs is then

$$P(j|\tau) = \frac{\lambda_j(X(t+\tau))}{\lambda_{tot}(X(t+\tau))} \tag{2.6}$$

In order to sample from the time-dependent probability densities defined in Equations (2.4) and (2.6), we use a Monte Carlo technique that equates the integration of a time-dependent density function to a random number. Given two random numbers, x_1 from the unity mean

exponential distribution and u uniformly distributed between 0 and $\lambda_{tot}(X(t+\tau))$, τ and j are chosen as follows:

$$\int_{t}^{t+\tau} \lambda_{tot}(t')dt' = x_1 \tag{2.7}$$

$$\sum_{k=1}^{j-1} \lambda_k(X(t+\tau))I_k < u \leq \sum_{k=1}^{j} \lambda_k(X(t+\tau))I_k \tag{2.8}$$

where I_k is the indicator function

$$I_k = \begin{cases} 1 & \text{if } R_k \in D \\ 0 & \text{if } R_k \in C \end{cases} \tag{2.9}$$

Thus, in our implementation we determine the next reaction time by integrating the ODE system and λ_{tot} forward in time until Equation (2.7) is satisfied. We then choose which reaction occurred according to Equation (2.8).

2.4 Comparison with Related Work

In comparing our approach to related work, we consider the three aspects described in the previous sections: the partitioning techniques, the approximation of continuous reactions, and the simulation of discrete reaction events.

2.4.1 Partitioning

A partitioning scheme can be characterized by two components: (a) when partitioning is done and (b) how it is done. For (a), the partitioning can be done statically, in an off-line manner [16], or dynamically, i.e., the system partitioning may change over the course of the simulation [19]. As for (b), the system can be partitioned based on the populations of species involved in the reactions, the reaction propensities themselves [18], or some combination of the two [16, 19]. In theory, if one had some a priori knowledge of the system, one could classify

as continuous those reactions that occur most frequently. However, that may not be plausible for some systems, and certainly does not consider the problem that different reactions could be bottlenecks at different times during the simulation or that some of the reactions that occur most frequently involve small populations. For these reasons, we employ an on-line and dynamic partitioning scheme that uses both population- and rate-based criteria.

2.4.2 Continuous reactions

As an alternative to ODEs, which are fundamentally deterministic, others have proposed stochastic differential equations (SDEs) to model the continuous reactions. The Langevin equation, which extends the ODE formulation to include a stochastic noise term, has been adapted for application to chemical systems by Gillespie [12]. The chemical Langevin equation, written as

$$dX_i = \sum_{j:R_j \in C} \nu_{j,i} \lambda_j(X) dt + \sum_{j:R_j \in C} \nu_{j,i} \sqrt{\lambda_j(X)} dW_j \tag{2.10}$$

where W is a $|C|$-dimensional Wiener process, can be used to simulate the stochastic dynamics of the system as a result of the continuous reactions [18, 19].

We choose to model the continuous reactions using ODEs instead of SDEs for simplicity. While ODEs are a theoretically less accurate approximation, we justify this decision with the wide availability of rigorous ODE solvers and the acceptable levels of accuracy observed. Note that in the thermodynamic limit, the system evolves deterministically as the chemical Langevin equation reduces to the ODE system in (2.3). Obviously the limit is unattainable in physical systems, but if one is primarily interested in the stochasticity of the small populations as opposed to the large populations, the approximation is acceptable.

2.4.3 Discrete reactions

One may also describe the transitions in the time-dependent Markov process, which represents the discrete reactions, with other probability distributions. Salis and Kaznessis [19]

use the time-dependent probability density of the Next Reaction variant of the stochastic simulation algorithm, and construct a system of "jump equations," one for each discrete reaction, whose solution produces the time at which the next reaction occurs. Regardless of the distribution, the approaches are mathematically equivalent. While the Direct approach is more computationally efficient, the Next Reaction approach offers the flexibility of allowing different distributions of reaction times among the individual reactions.

CHAPTER 3

METHODS

We now describe our methods for solving the partitioned reaction system of Chapter 2, beginning with an overview of the hybrid simulation algorithm followed by a discussion of the implementation of the algorithm.

3.1 Algorithm

At the beginning, X is set to the initial state vector X_0, the simulation time t is initialized to zero, and the stoichiometric matrix ν is computed. The set of reactions is partitioned according to the criteria established in Section 2.1. If none of the reactions are classified as continuous, the stochastic simulation algorithm based on Gillespie's Direct method is performed, and the ODE integration step is skipped. Otherwise, a random number is selected from the unity mean exponential distribution to prepare for the numerical integration of the ODE system. The integration is then done until one of the following stopping conditions is met: a discrete event has been generated at time τ as described by Equation (2.7), the assumptions for the continuous approximation are not valid, or the end of the simulation is reached. If a discrete event has been generated, the next reaction to occur is randomly chosen, weighted by the relative reaction propensities, and the state vector is updated accordingly. This procedure continues until the entire trajectory is computed, i.e., $t = t_{end}$.

1. Initialize the system:

(a) $X = X_0$, $t = 0$.

(b) Compute stoichiometric matrix ν.

2. Loop while $t < t_{end}$

(a) Partition set of reactions using the algorithm outlined in Section 2.1.

(b) If continuous reactions exist:

i. Select a random number from the unity mean exponential distribution.

ii. Numerically integrate the ODE using Equation (2.3). Stop when:

i. Equation (2.7) states a discrete reaction should occur,

ii. Equation (2.1) determines a continuous reaction should be repartitioned, or

iii. t_{end} is reached.

(c) If a discrete reaction is to occur:

i. For each $R_k \in D$, compute discrete reaction propensity λ_k and set $\lambda_{tot} = \sum \lambda_k$.

ii. If no continuous reactions exist, choose τ from the exponential distribution with mean $1/\lambda_{tot}$.

iii. Generate u from the uniform distribution $(0, \lambda_{tot})$.

iv. Choose j such that $\sum_{k=1}^{j-1} \lambda_k I_k < u \leq \sum_{k=1}^{j} \lambda_k I_k$.

v. Let $X = X + \nu_j$, where ν_j is the jth row of ν, and set $t = t + \tau$.

3. End simulation and report results.

3.2 Implementation

Our hybrid simulation algorithm was implemented as part of an extensible reaction simulation framework written in ISO C++. For comparison, the framework also consists of a

stochastic simulator, which implements the Direct variant of Gillespie's method [9], and a deterministic differential equation solver. We used these simulators to compare accuracy and computation time among the various algorithms for several published models, and to compare our results with published results.

The simulation framework makes extensive use of object-oriented design principles and inheritance. Each solver is implemented as a derived class of the base simulator class, and employs its own simulation algorithm and data structures. In particular, the modular design of the hybrid simulator allows us to quickly and easily plug in different ODE solution methods and experiment with different partitioning techniques.

The framework includes two different high-order approximate solution methods for ODE systems, both of which use adaptive step size control based on the estimate of error at each internal solver step. The first is a standard fifth-order Cash-Karp Runge-Kutta algorithm [23]. The second is CVODE [24], a solver of initial value problems for stiff and nonstiff ODE systems. CVODE is part of the SUNDIALS suite of solvers for differential and nonlinear algebraic systems, developed by the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory.

All model and solution parameters, such as Λ from (2.2), are specified in a model description file. The implementation parses the model description file, instantiates the appropriate simulator object, and initializes its data structures.

3.3 Validity of Confidence Intervals

The confidence intervals as computed by the hybrid simulator can be used to specify the confidence level in the estimate of the mean (or higher moments) of the simulated hybrid process. It is important to note that the confidence intervals constructed do not provide a confidence level for the estimate with respect to the original stochastic process, but with respect to the hybrid process as defined by the partitioning scheme.

Thus, to prove that the confidence intervals are valid, one must prove that the results for different sample paths of the hybrid simulator are independent and identically distributed, assumptions needed by the central limit theorem. To see this, consider the CTMC simulated by the stochastic simulator. Let $P : R \rightarrow \{C, D\}$. That is, a partitioning P, which is a function of the system state X (as defined by the partitioning scheme), is a function mapping each reaction in R to either the continuous or discrete subset. Consider two states X_1 and X_2 that have corresponding partition functions P_1 and P_2, and partition the (possibly infinite) state space of the CTMC according to the following equivalence relation:

$$X_1 \equiv X_2 \iff P_1 = P_2 \tag{3.1}$$

Given a γ and Λ, for each state there is a unique classification of reactions P. Since there is a finite number of species in the system M, there is a finite number of such equivalence classes, bounded above by 2^M.

For each partition (equivalence class), there are d species that do not participate in continuous reactions and $c = M - d$ species that do participate in continuous reactions. Replace each partition with a c-dimensional positive Euclidean space $\mathbb{R}^c$. Now consider an infinitessimal time interval of length dt. During this time interval, there are two types of finite transitions that could have occurred to each point in the c-dimensional space representing each partition.

1. a discrete reaction transition that fires with probability λdt, where $\lambda = \sum \lambda_j$ and λ_j is the instantaneous exponential rate of discrete reaction R_j given by the kinetic rate law

2. a continuous reaction transition that occurs with probability $1 - \lambda dt$ and transitions to the new state $X' = X + \frac{dX}{dt} * dt + o(dt)$

Either of these transitions may result in a new state in a different equivalence class, corresponding to a repartitioning of reactions. For a fixed dt, let $\{Y_t | t \geq 0\}$ represent the process described above. Since Y_t is a discrete-time Markov chain, the confidence intervals constructed are valid indicators of the confidence in the estimate. Consider that

$$Z_t = \lim_{dt \to 0} Y_t \tag{3.2}$$

is identical to the hybrid process simulated by the hybrid simulator, and the result follows.

CHAPTER 4

RESULTS AND DISCUSSION

We first present in detail an example model of biological significance to demonstrate the performance of the hybrid simulation algorithm. In order to show that the algorithm has been tested, and performs well, on a wide variety of models, we will briefly present some results for other models in the sections that follow.

The chosen models provide good test cases because they involve species present at varying orders of magnitude, and, as a result, some reactions occur much more frequently than others. Thus, exact stochastic simulation is quite expensive computationally. In addition, some of the models exhibit bimodal behavior, thus rendering an ODE solution ineffective. We therefore investigate hybrid simulation as a reasonable alternative, and consider the accuracy and speedup of the algorithm when applied to these models.

All experiments in this study were performed on a Linux Fedora Core 3 system with an Athlon XP 2400 processor and 512 MB of RAM. The ODEs in the hybrid algorithm are integrated using the CVODE higher-order methods with adaptive step sizing. Similar results were obtained using Runge-Kutta methods.

4.1 HIV-1 Tat Transactivation Example

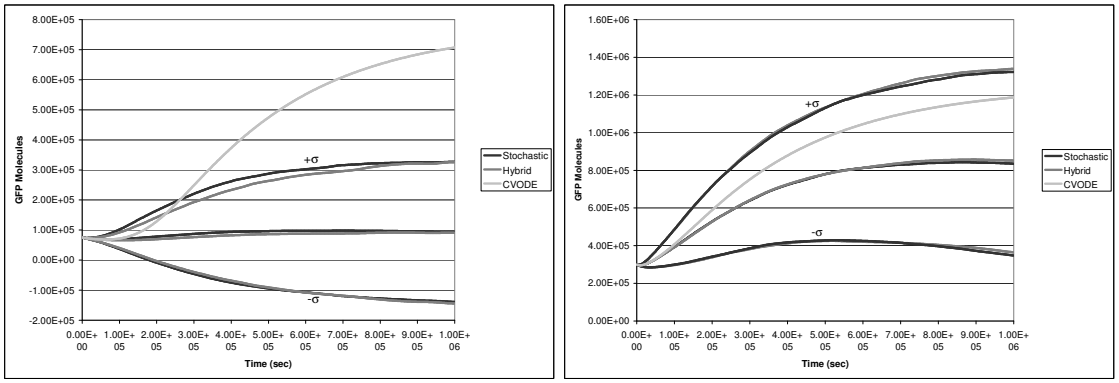
We consider here a previously introduced [6] model of the stochastic fluctuations in the HIV-1 Tat protein within the Tat transactivation positive feedback loop. These fluctuations

have been experimentally and computationally shown to influence the progression of the virus to either latency or productive infection, resulting in two distinct phenotypes. This example was chosen to test our algorithm because it is part of a large and growing body of work investigating the influence of stochastic gene expression on phenotypic variability. In addition, because of the bimodal nature of the system, a deterministic model does not accurately capture the dynamics, and a stochastic description is needed. However, CPU time becomes an obstacle to a discrete simulation, and thus we believe this model fits nicely into our hybrid system framework.

We simulated two variations of the model, corresponding to different initial GFP and Tat concentrations (*Dim* bulk sort and *Mid* bulk sort), as well as different kinetic coefficients for GFP translation. In this model, the GFP degradation and translation reactions are by far the ones occurring most frequently, and thus present the greatest burden to a discrete simulation. They are the reactions that the hybrid algorithm targets for elimination by approximating them as continuous events. A list of reactions and parameters of the Tat transactivation model is available in Appendix A. Simulations of each model were performed using the hybrid algorithm, exact stochastic simulation, and a deterministic ODE solution.

Figure 4.1 compares the time evolution of the first two moments of the GFP marker, as computed by the three algorithms for 10^6 s ($\sim$1.65 weeks) of simulation time. Qualitatively, this figure shows that the hybrid algorithm manages to capture these moments very well. Furthermore, the deterministic ODE solution is unable to reconstruct even the first moment. The reason is the bimodal nature of the probability density of GFP, which can be seen in Figure 4.2 for the *Mid* sort variation of the model at time 10^6 s.

Note that the hybrid algorithm is quite accurate in reconstructing the entire distribution as computed by the exact solution method. We parameterize the hybrid algorithm with $\Lambda = 2$ and $\gamma = 5$ to construct this figure, but similar results were observed for other values of the solution parameters. In fact, the insensitivity of the result to the value of γ can be explained by the following. Regardless of the fraction of time during which the reactions



(a) *Dim* bulk sort (b) *Mid* bulk sort

Figure 4.1 A comparison of the time evolution of the mean (center line) and standard deviation ($\pm\sigma$) of GFP for discrete stochastic, hybrid, and deterministic ODE simulation. Results are based on 1000 trials.

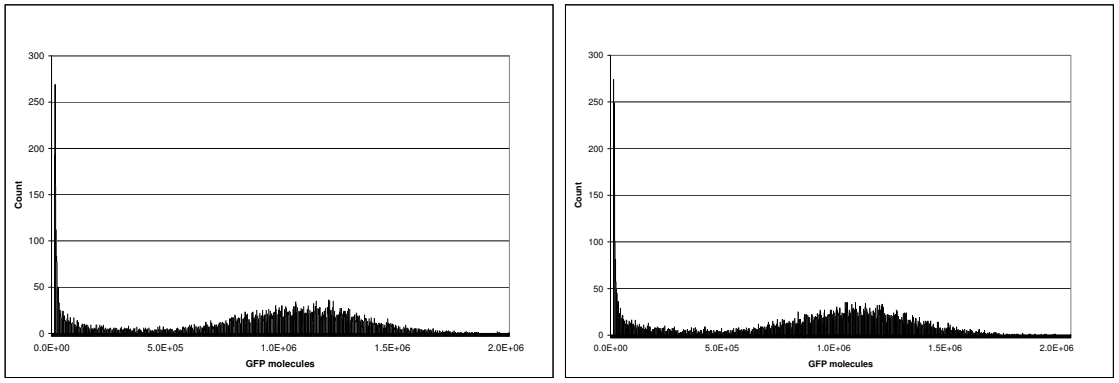


Figure 4.2 The distribution of GFP molecules at time $t = 10^6$ s as computed by exact stochastic simulation (left) and hybrid simulation (right) for the *Mid* sort variation of the Tat transactivation model. Results are based on 10 000 simulations.

involving GFP are classified as continuous or discrete (i.e., regardless of the parameter γ), the continuous approximation will be very good because GFP molecules are present in such high numbers, on the order of thousands to a few million molecules. Thus, for this particular model, the ability of the hybrid algorithm to reconstruct the distribution of GFP molecules is independent of the choice of solution parameters. Furthermore, Figure 4.3 shows that increasing γ, beyond a certain point, has a dramatic effect on the computational expense of the hybrid algorithm. The reason is that with increasing γ, more of the system is being modeled discretely, and the cost of simulation grows with the number of discrete events. Note that for $\gamma \leq 5000$, the hybrid algorithm significantly outperforms the stochastic algorithm, as the CPU time of the former is not affected by the changing solution parameter. For $\gamma \geq 500\ 000$, however, the elimination of reaction events in the hybrid algorithm does not outweigh the overhead of partitioning and switching between continuous and discrete solutions, and thus no speedups are observed with the hybrid algorithm. Nonetheless, as varying γ has no effect on the quality of the hybrid approximation for this example, one does not have to sacrifice accuracy to obtain performance gains. This is not always the case, however, as can be seen in the results from the intracellular viral infection example [18, 25] included in Section 4.4. In that example, the choice of γ does affect the accuracy of the approximation because the template species is present in smaller numbers, on the order of tens of molecules.

Tables B.1 and B.2 summarize the performance of the hybrid algorithm, when compared to the stochastic simulation algorithm, on the Tat transactivation example, as well as a few other benchmark models collected from the literature. The tables compare the performance of the algorithms on this set of examples, including a model of intracellular viral infection [18, 25], the cycle test [19], and a simple crystallization system [18, 19], with respect to the average number of discrete events and CPU time per run. Additional results for these benchmarks can be found in the following sections.

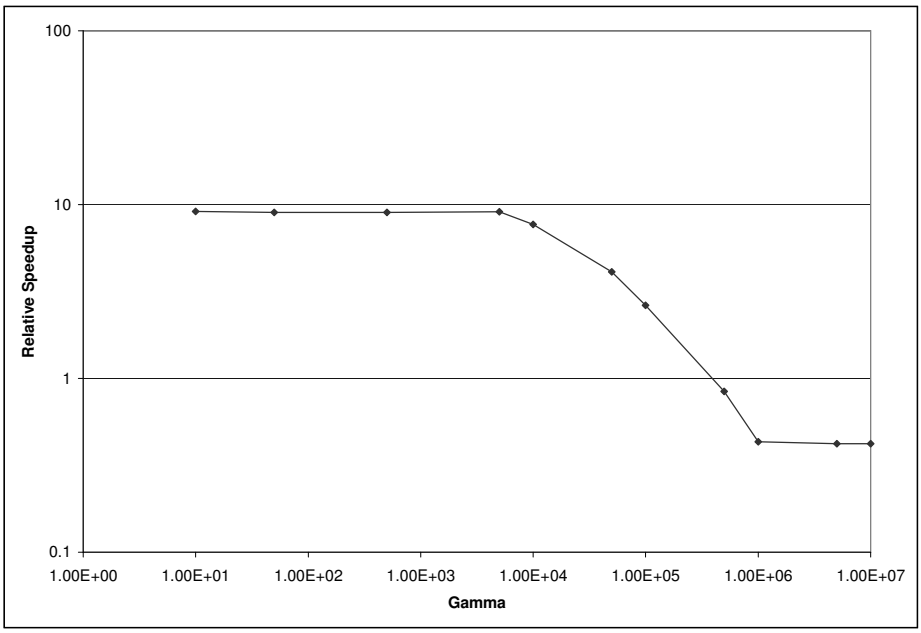


Figure 4.3 A comparison of the CPU time per run of the hybrid algorithm versus the baseline stochastic simulation, as the solution parameter γ is varied, for the *Dim* sort variation of the Tat transactivation model. Relative speedup is measured as the ratio of the computational run time of stochastic simulation to that of hybrid simulation.

4.2 Cycle Test

The system of reactions given in Table 4.1 is called the *cycle test*. The model is run at multiple "system sizes," where the initial conditions of the species and the kinetic coefficients of reactions (4) and (5) are varied in order to increase the separation between the fast and slow reaction rates. Reactions (1)-(3) occur with much more frequency than the other two, and therefore are the primary target for the continuous approximation, while reactions (4) and (5) are more often classified as discrete. Note that the kinetic coefficients of (4) and (5) decrease with system size, giving them an average rate of 0.75 and 1 molecules/s, respectively, while the rates of reactions (1)-(3) increase as the system size is increased. Figure 4.4 demonstrates how the cost of exact stochastic simulation scales with Θ, while that of hybrid simulation remains constant. Similar trends can be seen in the results for the example in the next section.

Table 4.1 Model parameters for the cycle test model.

Reactions	Kinetic coefficient
(1) A $\rightarrow$ B	$k_1 = 0.20\ s^{-1}$
(2) B $\rightarrow$ C	$k_2 = 0.30\ s^{-1}$
(3) C $\rightarrow$ A	$k_3 = 0.40\ s^{-1}$
(4) A+C $\rightarrow$ D	$k_4 = 117\ 810\ 500.0/\Theta^2\ [Ms]^{-1}$
(5) B+C $\rightarrow$ E	$k_5 = 235\ 620\ 900.0/\Theta^2\ [Ms]^{-1}$

Volume = 10^{-15} L
$A_0 = \Theta$, $B_0 = 2\Theta$, $C_0 = 3\Theta$, $D_0 = E_0 = 0$
System sizes: $\Theta = \{100, 1000, 10\ 000, 100\ 000\}$

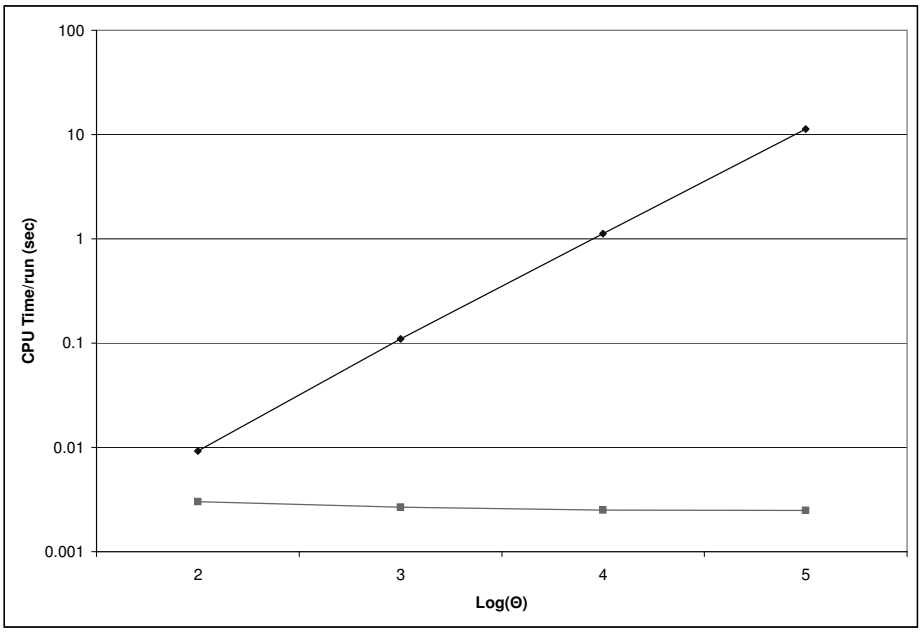


Figure 4.4 A comparison of the running times for different values of Θ in the cycle test model using discrete stochastic simulation (diamonds) and hybrid simulation (squares). Results are based on 10 000 trials.

4.3 Simple Crystallization Example

The following is a previously treated [18, 19] simplified model for the crystallization of species A, consisting of the two reactions:

$$2\mathrm{A} \xrightarrow{k_1} \mathrm{B} \tag{4.1}$$

$$\mathrm{A} + \mathrm{C} \xrightarrow{k_2} \mathrm{D} \tag{4.2}$$

Table 4.2 Model parameters for the simple crystallization system.

Parameter	Value
k_1	30.11 $[\text{M s}]^{-1}$
k_2	60.22 $[\text{M s}]^{-1}$
A_0	Θ
B_0	0
C_0	10
D_0	0

Volume = 10^{-15} L
$\Theta = \{10^6, 10^7, 10^8, 10^9\}$

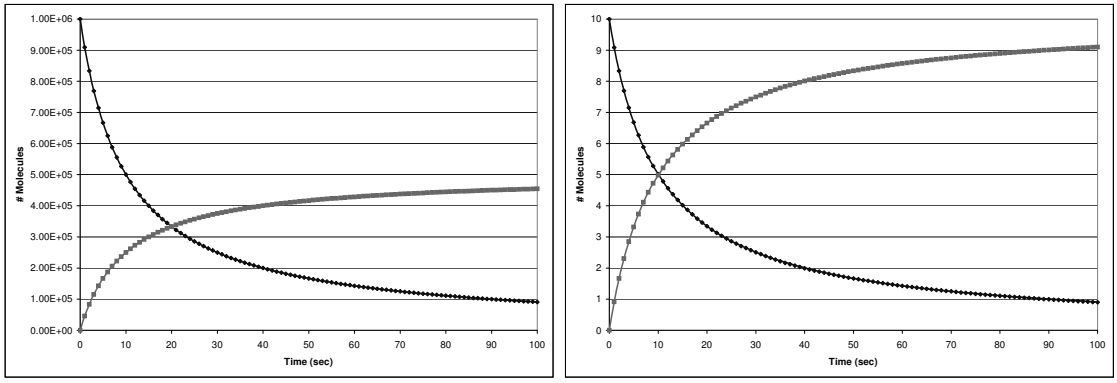
Given the kinetic coefficients and the initial number of molecular species in Table 4.2, the first reaction occurs many more times than the second. For this reason, it is optimal to classify the first reaction as continuous and the second as discrete. This is what our partitioning scheme does, after an initial transient period in which the number of molecules of B is less than γ, forcing the first reaction to be modeled discretely. The model itself is parameterized by Θ, the initial number of molecules of species A. As Θ increases, the difference in propensity between the reactions becomes larger. In particular, the initial rate of the first reaction is proportional to Θ^2, while the initial rate of the second reaction grows linearly with Θ.

We first perform 10 000 trials of the crystallization model using the hybrid algorithm ($\Lambda = 25$, $\gamma = 100$) and exact stochastic simulation. Figure 4.5 compares the solutions computed by the two algorithms for 100 s of simulation time with $\Theta = 10^6$, and shows that the hybrid approximation accurately reconstructs the mean for all species.

To quantify the error in the first and second moments, we compute the absolute difference in the mean and variance between the hybrid (H) and stochastic (S) simulation results:

$$|E[X^S(t)] - E[X^H(t)]| \tag{4.3}$$

$$|Var[X^S(t)] - Var[X^H(t)]| \tag{4.4}$$



(a) Compares the means of species A (diamonds) and B (squares).

(b) Compares the means of species C (diamonds) and D (squares).

Figure 4.5 Comparison of the results of hybrid simulation (points) to those of exact stochastic simulation (lines) based on 10 000 trials of the crystallization model with $\Theta = 10^6$.

The mean and variance errors for species A and C are shown in Figure 4.6.

Based on 100 runs of the hybrid and stochastic simulation algorithms, we then plot the CPU time per run for each parameterization of the model in Figure 4.7. As with the cycle test, we observe that while the computational expense of exact stochastic simulation grows with increasing Θ, the cost of hybrid simulation remains fairly constant. Thus, order of magnitude speedups are possible when hybrid methods are applied to this model.

4.4 Viral Infection Example

We now consider a general model of the infection of a cell by a virus [18, 25].

$$\text{template} \xrightarrow{k_1} \text{template} + \text{genome} \tag{4.5}$$

$$\text{genome} \xrightarrow{k_2} \text{template} \tag{4.6}$$

$$\text{template} \xrightarrow{k_3} \text{template} + \text{struct} \tag{4.7}$$

$$\text{template} \xrightarrow{k_4} \text{degraded} \tag{4.8}$$

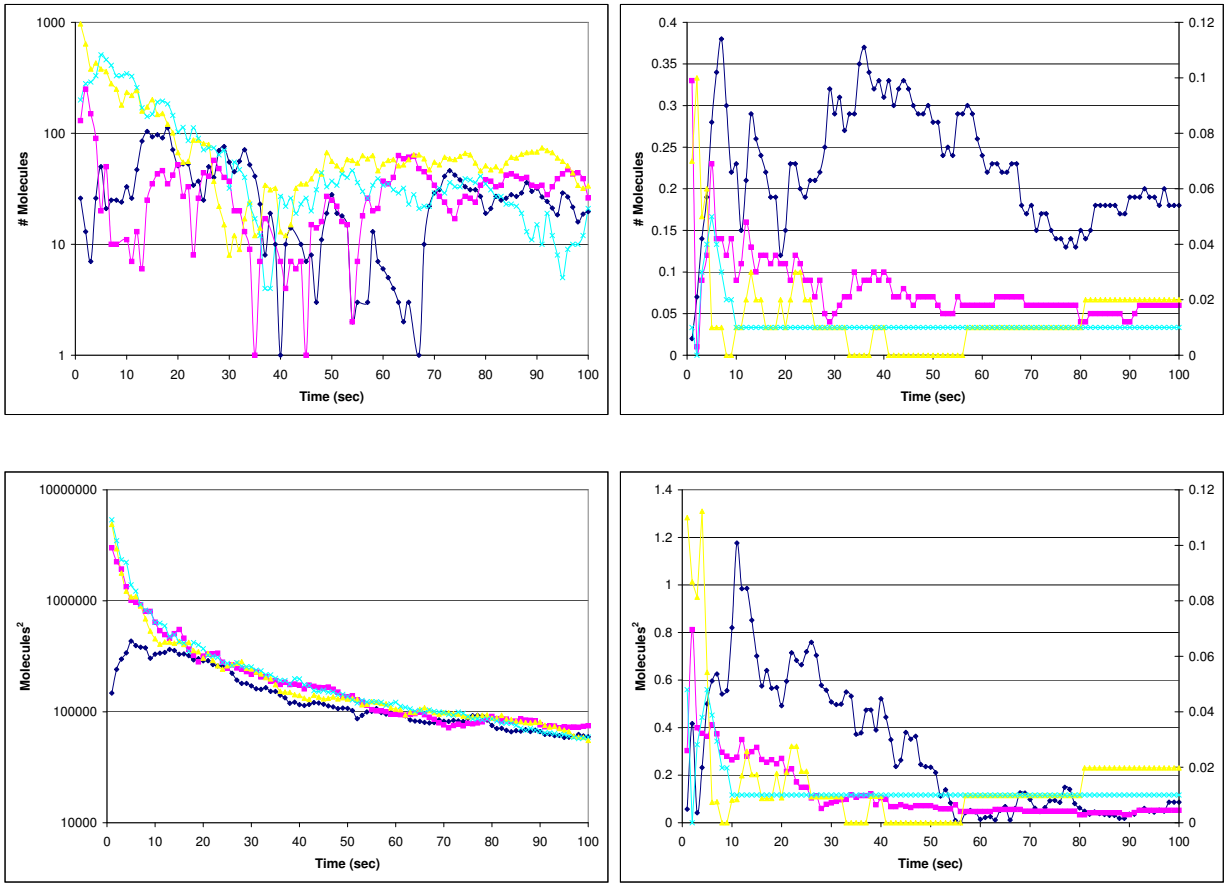


Figure 4.6 Errors in mean (top) and variance (bottom) of species A (left) and C (right) for 100 runs of the crystallization model with $\Theta = 10^6$ (diamonds), 10^7 (squares), 10^8 (triangles), and 10^9 ($\times$'s). Errors for species C are plotted with $\Theta = 10^8$ and 10^9 on the secondary axis.

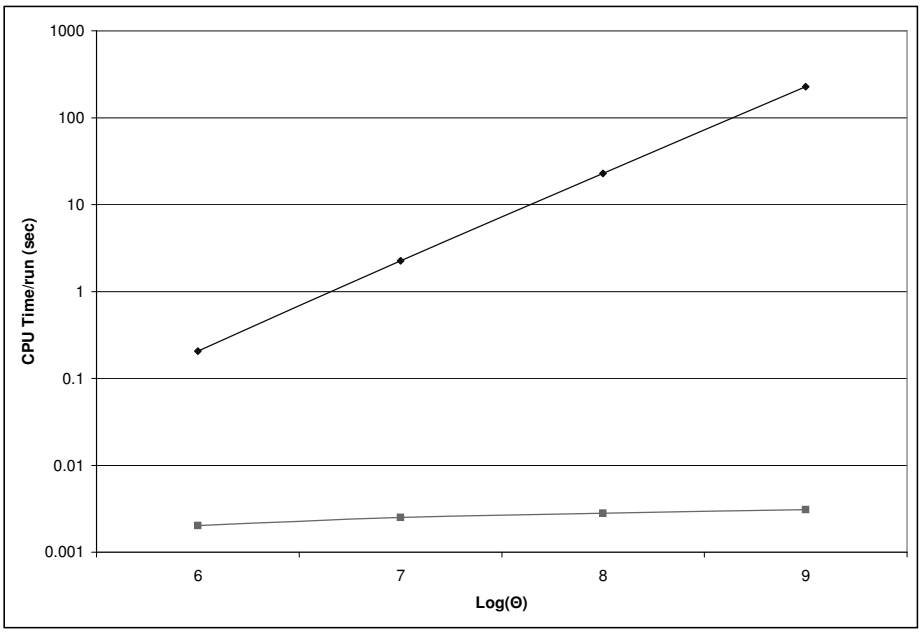


Figure 4.7 A comparison of the running times for different values of $\Theta = A_0$ in the crystallization model using discrete stochastic simulation (diamonds) and hybrid simulation (squares).

$$\text{struct} \xrightarrow{k_5} \text{secreted/degraded} \tag{4.9}$$

$$\text{genome} + \text{struct} \xrightarrow{k_6} \text{secreted virus} \tag{4.10}$$

where *genome* and *template* are, respectively, the genomic and template viral nucleic acids and *struct* is the viral structural protein. We assume that nucleotides and amino acids are available at constant concentrations, and thus are not included in the model. Also, the template species is a catalyst for reactions (4.5) and (4.7), and thus is neither produced nor consumed in these reactions. Table 4.3 lists the kinetic coefficients and the initial conditions of the model.

We perform 1000 simulations of the model using the hybrid algorithm, exact stochastic simulation, and a deterministic ODE solution. We considered multiple parameterizations of the hybrid algorithm by varying γ while keeping constant $\Lambda = 25$. Smaller values of γ lead to aggressive partitioning (more of the system is modeled continuously), while larger values lead to more conservative partitioning (more of the system is modeled discretely).

Figures 4.8 and 4.9 compare the bimodal distribution of template molecules at 200 days

Table 4.3 Model parameters for the viral infection example.

Parameter	Value
k_1	1 day^{-1}
k_2	0.025 day^{-1}
k_3	1000 day^{-1}
k_4	0.25 day^{-1}
k_5	1.9985 day^{-1}
k_6	7.5E-6 [molecules·day]$^{-1}$
template_0	1
genome_0	0
struct_0	0

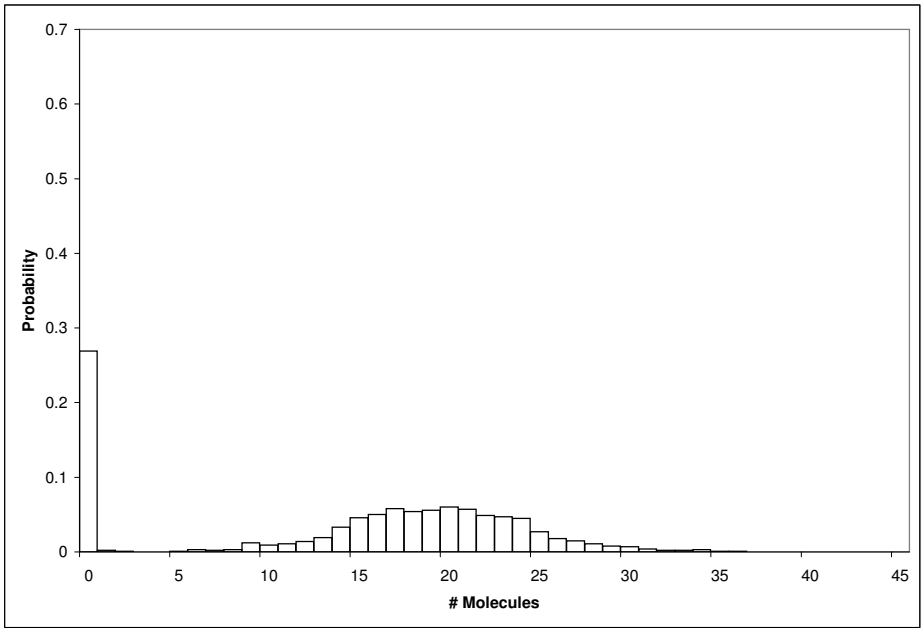
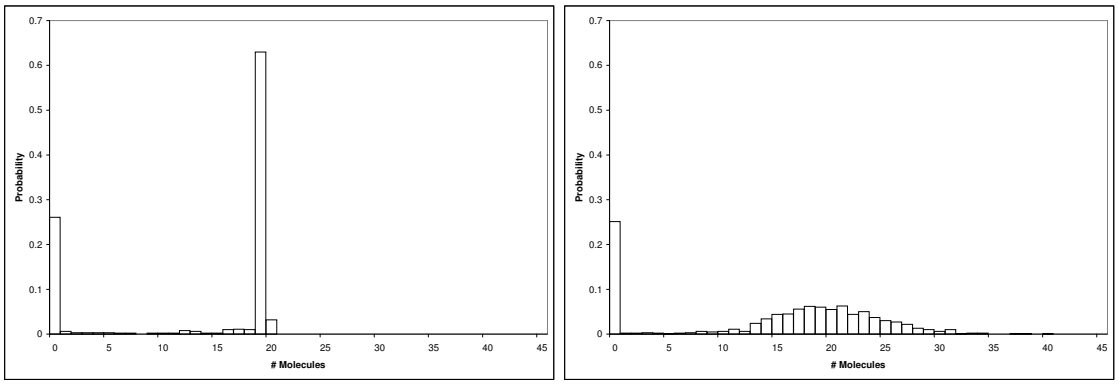


Figure 4.8 The distribution of the template species at time $t = 200$ days as computed by exact stochastic simulation.

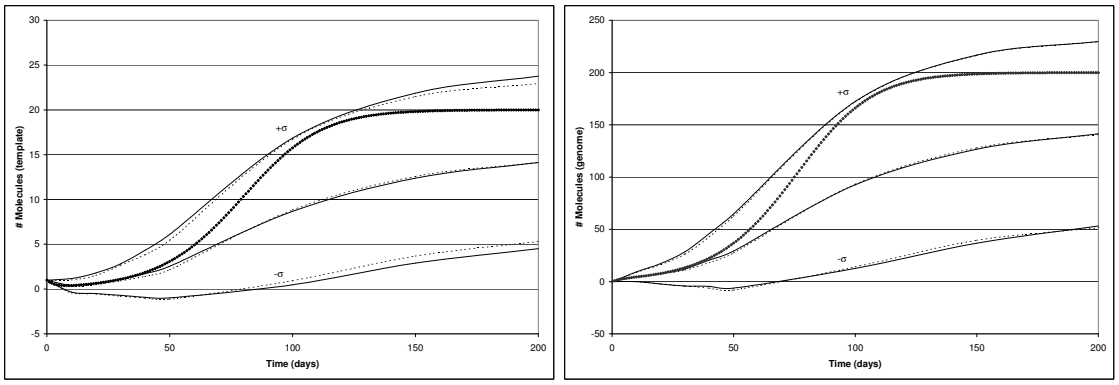
for exact stochastic simulation, hybrid simulation with $\gamma = 10$, and hybrid simulation with $\gamma = 100$. Note that while the hybrid algorithm with $\gamma = 10$ (aggressive partitioning) is unable to reconstruct the entire distribution, the hybrid algorithm with $\gamma = 100$ (conservative partitioning) is quite accurate.

Figure 4.10 compares the time evolution of the first two moments of the template and genome species, as computed by the three algorithms for 200 days of simulation time. We



(a) Hybrid simulation with $\gamma = 10$ (b) Hybrid simulation with $\gamma = 100$

Figure 4.9 The distribution of the template species at time $t = 200$ days as computed by hybrid simulation.



(a) Template (b) Genome

Figure 4.10 A comparison of the time evolution of the mean (center line) and standard deviation ($\pm\sigma$) of the template and genome species for discrete stochastic (solid lines), hybrid with $\gamma = 10$ (dashed lines), and deterministic ODE (points) simulation.

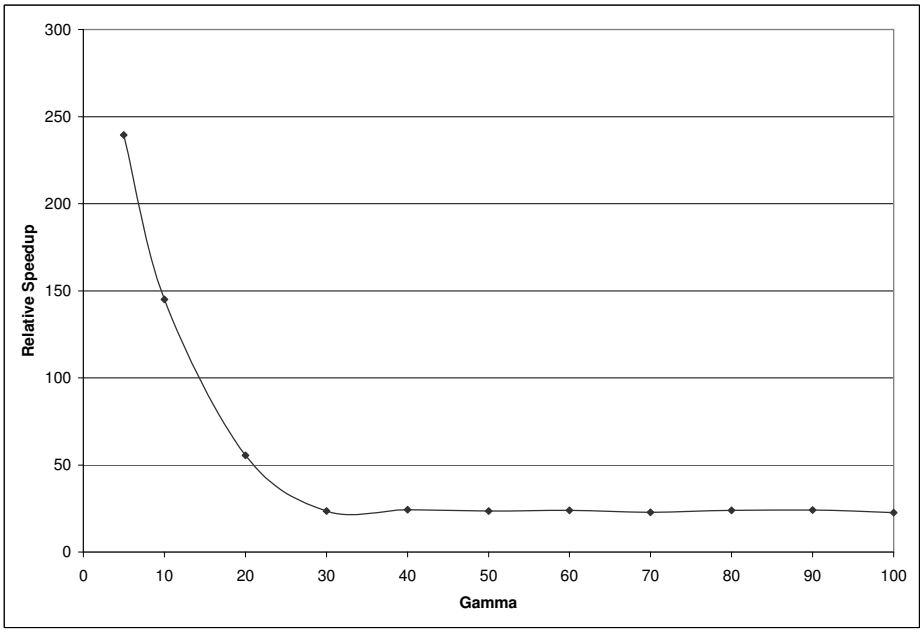


Figure 4.11 The reduction in computational expense of the hybrid algorithm over stochastic simulation for the viral infection model, as the solution parameter γ is varied. Relative speedup is measured as the ratio of the computational run time of stochastic simulation to that of hybrid simulation.

use the hybrid algorithm with $\gamma = 10$ to construct these figures, but similar results were observed for other values of γ. As noted, the approximation of the distribution of template molecules is poor with $\gamma = 10$. However, this figure shows that even in such conditions, the hybrid algorithm is quite accurate in reconstructing the time evolution of the first two moments. As with the Tat transactivation example, the deterministic ODE solution is unable to reconstruct even the mean of the distribution.

Figure 4.11 shows the relationship between the computational savings provided by the hybrid algorithm over exact stochastic simulation and the value of γ. This figure indicates that smaller values of γ, or more aggressive partitioning strategies, result in greater computational gains. For example, a 239-fold reduction in computational expense over a discrete stochastic solution is possible with $\gamma = 5$. However, we reported that the hybrid algorithm, when parameterized in such a way, is unable to reconstruct higher moments. This observation suggests there is a tradeoff between the speedup and the accuracy of the algorithm.

In particular, if the measure of interest is a mean, or the average behavior, more aggressive partitioning will suffice and result in larger computational savings. However, if the measure of interest is a probability or a distribution, more conservative partitioning may be required, resulting in smaller computational gains. Nonetheless, for this example, even the most conservative partitioning results in a 24-fold speedup over exact stochastic simulation.

While we acknowledge that this is a fairly simple model, it does serve to illustrate the correctness of the algorithm, and offer some support to a measure-driven notion of partitioning. We realize that in real, complex biological models, the partitioning may not be so intuitive. However, we believe that this is an important step towards gaining insight into the types of models that may benefit from a hybrid approach, and how different approaches to partitioning may affect the accuracy and computational expense of the algorithm.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

While Gillespie has developed exact methods for simulating the stochastic kinetics of a well-mixed chemical system, these methods become quite computationally intensive for larger systems with higher numbers of reaction events. Continuous models, such as those described by differential equations, can approximate the exact stochastic solution and are much simpler computationally. However, they are not as useful when modeling systems with low-frequency reaction events and small populations, as they fail to preserve the stochastic nature of such discrete-event systems.

In this study, we presented a hybrid simulation algorithm that partitions the reaction system into continuous and discrete subsets, approximates the continuous reactions deterministically as an ODE system, and generates discrete reaction events by integrating a time-dependent probability density function. We described a unique approach to partitioning in which the extents of reaction are not partitioned based on an absolute threshold, but instead on a threshold relative to the distribution of discrete reaction propensities. The partitioning, which uses a combination of population- and rate-based criteria, is done online and updated dynamically.

We implemented the hybrid algorithm in an extensible simulation framework using principles of object-oriented design and inheritance. Our implementation also utilizes two different

rigorous ODE solvers to simulate the continuous reactions. To demonstrate the accuracy and performance of the hybrid algorithm when compared to exact stochastic simulation, an example model was given and both algorithms were used to compute various measures of interest on the model. In this example and the others that followed, we were able to show that the hybrid algorithm is reasonably accurate, and that computational savings are possible if the distributions of partitioned reaction extents vary by some orders of magnitude. In the worst case, our hybrid algorithm reduces to Gillespie's Direct method for discrete stochastic simulation with some additional overhead. Furthermore, for one of the examples, the algorithm was parameterized in such a way as to vary the fraction of the system that is modeled continuously. Results from that model suggested there may be a tradeoff between the accuracy and speedup of the algorithm, controlled by the aggressiveness of the partitioning technique employed. The tradeoff is likely dependent on the measure of interest to be computed from the model, particularly if the measure is on small-numbered species. Currently the partitioning strategy is tuned manually, by adjustment of the solution parameters Λ and γ, but it should be possible to determine these parameters automatically given the type of measure to be computed and topological information about the reaction network.

Directions of future efforts could include incorporating stochastic differential equations into the solution engine and investigating alternative partitioning schemes. We also have plans to develop an XML parser that will allow us to quickly simulate models specified in other standards, such as SBML [26] and CellML [27], within our framework.

APPENDIX A

TAT TRANSACTIVATION REACTIONS AND PARAMETERS

A list of reactions of the Tat transactivation model described in Section 4.1 is available here. The parameters for the two variations of the model, *Dim* bulk sort and *Mid* bulk sort, can be found in Tables A.1 and A.2, respectively.

$$LTR \xrightarrow{k_{basal}} LTR + mRNA_{nuc}$$

$$mRNA_{nuc} \xrightarrow{k_{export}} mRNA_{cyt}$$

$$mRNA_{cyt} \xrightarrow{k1_{translate}} mRNA_{cyt} + GFP$$

$$mRNA_{cyt} \xrightarrow{k2_{translate}} mRNA_{cyt} + Tat_{deac}$$

$$Tat_{deac} + LTR \underset{k_{unbind}}{\overset{k_{bind}}{\longleftrightarrow}} pTEFb_{deac}$$

$$pTEFb_{deac} \underset{k_{deacetyl}}{\overset{k_{acetyl}}{\longleftrightarrow}} pTEFb_{acet}$$

$$pTEFb_{acet} \xrightarrow{k_{transact}} LTR + mRNA_{nuc} + Tat_{deac}$$

$$GFP \xrightarrow{d_{GFP}} 0, \qquad Tat_{deac} \xrightarrow{d_{Tat}} 0$$

$$mRNA_{cyt} \xrightarrow{d_{cyt}} 0, \qquad mRNA_{nuc} \xrightarrow{d_{nuc}} 0$$

Table A.1 Model parameters for the Tat transactivation example with initial GFP concentrations corresponding to a *Dim* sort.

Parameter	Value
k_{basal}	1e-9 s^{-1}
k_{export}	7.2e-4 s^{-1}
$k1_{translate}$	0.5 s^{-1}
$k2_{translate}$	0.00132 s^{-1}
k_{bind}	1.5e-4 $[molecules \cdot s]^{-1}$
k_{unbind}	0.017 s^{-1}
k_{acetyl}	0.001 s^{-1}
$k_{deacetyl}$	0.13 s^{-1}
$k_{transact}$	0.1 s^{-1}
d_{GFP}	3.01e-6 s^{-1}
d_{Tat}	4.3e-5 s^{-1}
d_{cyt}	4.8e-5 s^{-1}
d_{nuc}	4.8e-5 s^{-1}
LTR_0	1
Tat_0	5
GFP_0	75 000

Table A.2 Model parameters for the Tat transactivation example with initial GFP concentrations corresponding to a *Mid* sort.

Parameter	Value
k_{basal}	1e-9 s^{-1}
k_{export}	7.2e-4 s^{-1}
$k1_{translate}$	0.8 s^{-1}
$k2_{translate}$	0.00132 s^{-1}
k_{bind}	1.5e-4 $[molecules \cdot s]^{-1}$
k_{unbind}	0.017 s^{-1}
k_{acetyl}	0.001 s^{-1}
$k_{deacetyl}$	0.13 s^{-1}
$k_{transact}$	0.1 s^{-1}
d_{GFP}	3.01e-6 s^{-1}
d_{Tat}	4.3e-5 s^{-1}
d_{cyt}	4.8e-5 s^{-1}
d_{nuc}	4.8e-5 s^{-1}
LTR_0	1
Tat_0	100
GFP_0	300 000

APPENDIX B

SUMMARY OF RESULTS

Tables B.1 and B.2 summarize the performance of the hybrid algorithm, when compared to the stochastic simulation algorithm, on the Tat transactivation example, as well as the other models presented in this thesis. The tables compare the performance of the algorithms on this set of examples with respect to the average number of discrete events and CPU time per run. Additional results for these benchmarks can be found in the corresponding sections of Chapter 4.

Table B.1 A summary of performance results for several benchmark models comparing the stochastic and hybrid simulation algorithms. Average number of discrete events and CPU time, in seconds, are reported per run.

	Stochastic	
Model	**Avg. # Events**	**CPU Time**
HIV-1 Tat *Dim* sort	561415	0.576328
HIV-1 Tat *Mid* sort	4.83371e+06	4.97246
Intracellular viral infection	3.13796e+06	2.11551
Cycle test ($\Theta = 1000$)	162068	0.109848
Crystallization ($\Theta = 10^6$)	454551	0.206052

	Hybrid	
Model	**Avg. # Events**	**CPU Time**
HIV-1 Tat *Dim* sort	4140.42	0.0630483
HIV-1 Tat *Mid* sort	23869.7	0.359591
Intracellular viral infection	1249.66	0.0398304
Cycle test ($\Theta = 1000$)	27.613	0.00267859
Crystallization ($\Theta = 10^6$)	110.105	0.00202239

Table B.2 Stochastic/hybrid relative reduction in simulation cost for the benchmarks above.

	S/H	
Model	**Avg. # Events**	**CPU Time**
HIV-1 Tat *Dim* sort	135.59	9.14
HIV-1 Tat *Mid* sort	202.5	13.83
Intracellular viral infection	2511.05	53.11
Cycle test ($\Theta = 1000$)	5869.26	41.01
Crystallization ($\Theta = 10^6$)	4128.34	101.89

REFERENCES

[1] E. Korobkova, T. Emonet, J. Vilar, T. Shimizu, and P. Cluzel, "From molecular noise to behavioural variability in a single bacterium," *Nature*, vol. 428, pp. 574–578, April 2004.

[2] J. Raser and E. O'Shea, "Control of stochasticity in eukaryotic gene expression," *Science*, vol. 304, no. 5678, pp. 1811–1814, June 2004.

[3] M. Elowitz, A. Levine, E. Siggia, and P. Swain, "Stochastic gene expression in a single cell," *Science*, vol. 297, no. 5584, pp. 1183–1186, August 2002.

[4] N. Rosenfeld, J. Young, U. Alon, P. Swain, and M. Elowitz, "Gene regulation at the single-cell level," *Science*, vol. 307, no. 5717, pp. 1962–1965, March 2005.

[5] J. Pedraza and A. van Oudenaarden, "Noise propagation in gene networks," *Science*, vol. 307, no. 5717, pp. 1965–1969, March 2005.

[6] L. Weinberger, J. Burnett, J. Toettcher, A. Arkin, and D. Schaffer, "Stochastic gene expression in a lentiviral positive-feedback loop: Hiv-1 tat fluctuations drive phenotypic diversity," *Cell*, vol. 122, no. 2, pp. 169–182, July 2005.

[7] J. Peccoud, K. Vander Velden, D. Podlich, C. Winkler, L. Arthur, and M. Cooper, "The selective values of alleles in a molecular network model are context dependent," *Genetics*, vol. 166, no. 4, pp. 1715–1725, April 2004.

[8] A. Arkin, J. Ross, and H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, August 1998.

[9] D. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[10] T. Gardner, C. Cantor, and J. Collins, "Construction of a genetic toggle switch in *Escherichia coli*," *Nature*, vol. 403, no. 6767, pp. 339–342, January 2000.

[11] M. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *Journal of Physical Chemistry*, vol. 104, pp. 1876–1889, 2000.

[12] D. Gillespie, "The chemical Langevin equation," *Journal of Chemical Physics*, vol. 113, no. 1, pp. 297–306, July 2000.

[13] D. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, July 2001.

[14] Y. Cao, D. Gillespie, and L. Petzold, "Efficient stepsize selection for the tau-leaping simulation method," *J. Chem. Phys.*, 2006, to appear.

[15] C. Rao and A. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm," *Journal of Chemical Physics*, vol. 118, no. 11, pp. 4999–5010, March 2003.

[16] T. Kiehl, R. Mattheyses, and M. Simmons, "Hybrid simulation of cellular behavior," *Bioinformatics*, vol. 20, no. 3, pp. 316–322, February 2004.

[17] K. Takahashi, K. Kaizu, B. Hu, and M. Tomita, "A multi-algorithm, multi-timescale method for cell simulation," *Bioinformatics*, vol. 20, no. 4, pp. 538–546, March 2004.

[18] E. Haseltine and J. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics," *Journal of Chemical Physics*, vol. 117, no. 15, pp. 6959–6969, October 2002.

[19] H. Salis and Y. Kaznessis, "Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions," *Journal of Chemical Physics*, vol. 122, no. 5, p. 054103, January 2005.

[20] K. R. Joshi, N. Neogi, and W. Sanders, "Dynamic partitioning of large discrete event biological systems for hybrid simulation and analysis," in *Proceedings of the 7th International Workshop on Hybrid Systems: Computation and Control (HSCC 2004)*, March 2004, pp. 463–476.

[21] C. Rao, D. Wolf, and A. Arkin, "Control, exploitation and tolerance of intracellular noise," *Nature*, vol. 420, no. 6912, pp. 231–237, November 2002.

[22] D. Gillespie, *Markov Processes: An Introduction for Physical Scientists*. New York: Academic Press, 1992.

[23] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. New York: Cambridge University Press, 1992.

[24] S. Cohen and A. Hindmarsh, "CVODE, a stiff/nonstiff ODE solver in C," *Computers in Physics*, vol. 10, no. 2, pp. 138–143, March-April 1996.

[25] R. Srivastava, L. You, J. Summers, and J. Yin, "Stochastic vs. deterministic modeling of intracellular viral kinetics," *Journal of Theoretical Biology*, vol. 218, no. 3, pp. 309–321, October 2002.

[26] M. Hucka et al., "The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, March 2003.

[27] C. Lloyd, M. Halstead, and P. Nielsen, "CellML: Its future, present and past," *Progress in Biophysics and Molecular Biology*, vol. 85, no. 2-3, pp. 433–450, June-July 2004.