Effective Multi-level Monte Carlo Methods for Stochastic Biochemical Kinetics

by

Priyabrata Senapati

Bachelor of Science, Memorial University of Newfoundland, 2014

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Applied Mathematics

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

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

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

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

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

Effective Multi-level Monte Carlo Methods for Stochastic Biochemical Kinetics

Master of Science 2018

Priyabrata Senapati

Applied Mathematics

Ryerson University

Abstract

Stochastic mathematical models are essential for an accurate description of biochemical processes at the cellular level. The effect of random fluctuations may be significant when some species have low molecular counts. While exact stochastic simulation methods exist, they are typically expensive on systems arising in applications. Thus more effective strategies are required for simulating complex stochastic models of biochemical system. Often, the expected value of some function of the final time solution of the stochastic model is of interest. Then, the approach employing multi-level Monte Carlo methods is more efficient than the traditional techniques. In this thesis, we study multi-level Monte Carlo (MLMC) schemes for a reliable and effective simulation of stochastic models of biochemical kinetics. The advantages of these MLMC strategies are illustrated on several biochemical models arising in applications.

Acknowledgements

I would like to wholeheartedly thank my parents, my wife, my brother and my grandmother for their immense patience, instilling their faith in me and their constant inspiration to work hard. I would like to express my sincerest thanks and gratitude to the individuals who assisted with this thesis. First of all, I would like to thank my supervisor Prof. Silvana Ilie; her ongoing support and guidance were integral to the completion of both the research and this document. Finally, to my examining committee members, Prof. Jean-Paul Pascal, Prof. Dejan Delic and Prof. Dzung Ha, I want to express my great appreciation for the time set aside to read the thesis and attend the thesis defense.

Dedication

I dedicate this thesis to Sanatana Hindu Dharma, for its deepest understanding on cosmic nature that inspired me to pursue higher levels of education.

Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

Stochastic modelling is an important research topic in Systems Biology. Mathematical modelling and simulation of biochemical systems are more cost effective and faster than experiments in lab. In the last decade, many modelling approaches for biochemical networks have been considered.The experimental data is enormous in Cellular Biology and these data require to be analyzed, therefore, the need for accurate models of these biological processes and efficient tools for simulating and studying them. The traditional mathematical modeling of biochemical systems uses the continuous deterministic model of the reaction rate equations In deterministic models, concentrations of chemical species are continuous variables and the standard theory of chemical kinetics uses the reaction rate equations, which is a set of ordinary differential equations, to model the dynamics of the system. The reaction-rate equations are based on the *Law of Mass Action* which gives a relationship between the reaction rates and the molecular concentration. Deterministic modelling approaches have been sucessfully used for chemically reaction systems, where there are large molecular counts of the reacting species. However, key biological processes may involve some species with low population numbers. In an environment when there are small molecular numbers of certain species, the deterministic models are genearally inaccurate and stochastic models are required. For instance, when there are few regulatory molecules available in a single cell, a continuous deterministic model fails to describe the system dynamics and the intrinsic random fluctuations [20, 39]. In stochastic models, species may have an integer or a real number of molecules and the reactions are treated as discrete and random events. Stochastic models are essential for studying the behaviour of biochemical systems with low amounts of certain species. McAdams & Arkin [4] showed that stochasticity does play an important role in the lysis/lysogeny decision of the bacteria λ-phage [28]. Samoilov et al.[35] demonstrated that noise can induce bi-stability in a monostable system.

The *Chemical Master Equation* (CME) [12] is one of the most refined models of well-stirred biochemical systems. The most accurate model of non-homogenous biochemical sytems in Molecular Dynamics. Molecular Dynamics is a very complex model, as it keeps a record of all positions and velocities of the molecules in the system. Daniel Gillespie proposed an exact algorithm for simulating the solution of the Chemical Master Equation [11, 12]. This exact algorithm simulates every reaction one at a time. When biochemical systems have some fast reactions, it become computationally intensive to simulate every reaction in the system. For improving the computational time, Gillespie [14] proposed the tau-leaping method, where the system is advanced with a predetermined step size τ, which leaps over many iterations. Gillespie [14] also proposed the *Chemical Langevin Equation*, a continuous stochastic model and a bridge between the *Chemical Master Equation* and the *Reaction Rate Equation* models. The continuous stochastic model of biochemical kinetics, the *Chemical Langevin Equation* (CLE) is a reduction of the Chemical Master Equation in the regime of large molecular counts. When a system has very large population numbers for all reactants, the system dynamics may be modeled using the *RRE* instead of the *CLE*.

Other approximate algorithms have been developed in the litereature for solving the Chemical Master Equation and this remains an active area of research [19]. Such approximate algorithms for the CME were proposed by Rathinam et al. [31]. Cao, Gillespie & Petzold [6], Tian & Burrage [36] and Chatterjee et al. [7]. Stability and consistency studies of the tau-leaping method were conducted by Rathinam et al. [32], higher order tau-leaping methods were developed by Li [26] and adaptive time-stepping tau-leaping strategies were introduced by by Anderson [2]. Also, the tau-leaping method has been a connection between a microscopic, stochastic and discrete model of well-stirred biochemical kinetics (the Chemical Master Equation [14]) and a macroscopic, stochastic and continuous model (the Chemical Langevin Equation [14]). Langevin equations which are charachteristically stochastic differential equations (SDE) have received considerable attention for their practical applications in physics, chemistry, biology [37, 9] besides their uses in Systems Biology [24]. An introduction to the numerical solutions of stochastic differential equations can be found in Higham [18].

The exact algorithms and the approximate algorithms may fare well on certain models depending on the stiffness of the biochemical system. The exact algorithm, for instance, behaves well on non-stiff systems. But, when it comes to simulating stiffer systems, i.e, systems with fast and slow reactions, approximate methods may be more efficient. Another strategy to reduce the computational time and maintaining the level of accuracy while simulating more challenging biochemical systems is to use hybrid methods. Hybrid methods use a combination of approximate and exact models and/or methods. Among them are the methods of Alfonsi et al. [1], Cao et al. [5], Haseltine & Rawlings [16], Hellander & Lotstedt [17], Kiehl et al. [22], MacNamara et al. [27], Mattheyses & Simmons [23], Puchalka & Kierzek [29], Rao & Arkin [30] , Salis & Kaznessis [33], Samant & Vlachos [34] and Weinam et al. [38].

In this thesis, we study multi-level Monte Carlo methods (MLMC) for stochastic discrete models of biochemical systems. We investigate the efficiency of these methods, when approximating the mean values of some function of the state of the biochemical system. Giles [10] proposed the MLMC methods in order to approximate $E(f(X(t)))$, where $E(\cdot)$ is the expected value, f is a polynomial and $X(t)$ is a stochatic process of interest. The MLMC strategies aim to estimate the expected value $E(f(X(t)))$ at a reduced computational time compared to existing methods, while maintaining a good accuracy of the estimation. The method uses different levels of accuracy. A base estimator is computed using the tau-leaping method using with a large step size. In the subsequent levels, we calculate correction estimators which are then added to the base estimator in order to reduce bias and to improve accuracy. The key concept behind this method is to reduce the variance, by simulating sets of coupled trajectories in calculating correction estimators. Each trajectory simulated on a coarser grid is coupled with a trajectory simulated on a finer grid. The coupling of trajectories has to be carefully implemented so that the variance of the difference of the coupled trajectories is reduced, leading to a reduction of the computational time. The *MLMC* methods give good estimation of the mean value at a fraction of the computational cost of the exact algorithm. Anderson and Higham [3] developed multi-level Monte Carlo methods for models of biochemical systems. However, their approach may not be the most effective on generic biochemical systems. In this work, we study the MLMC strategies proposed by Lester et al [25]

This thesis is organized as follows. In Chapter 2, we introduce stochastic models of well-stirred biochemical systems. The models we discuss are the Chemical Master Equation (CME). In Chapter 3, we present the multi-level Monte Carlo methods for the CME, and investigate improvements to their computational cost. In Chapter 4, we illustrate the advantages of the MLMC methods over the stochastic simulation algorithm by testing them on three biochemical models of practical interest. The models that we used are Potassium Channel model, Michaelis Menten model and Cyclical Reaction model. Finally, in Chapter 5, we present our conclusions and discuss our future work.

Chapter 2

Background

2.1 Stochastic Models

In this section we consider a process that involves N different types of molecules or biochemical species $S_1, ..., S_n$. We are also assuming that the system is a well-stirred (or homogeneous) system so that the molecules are spread uniformly across the spatial domain. These N species will take part in M different types of chemical reactions $R_1, .., R_n$. For instance, a molecule of type A can combine with a molecule of type B to create a molecule of type C. We also assume that the system is in thermal equilibrium and that the volume of the spatial domain is constant.

Suppose that initially at time t = 0, we know that the molecular amounts and our objective is to describe how these number of molecules evolve over time. Thus the state vector, may be represented as an N X 1 matrix. The i-th row corresponds to the number of S_i molecules at time t.

$$\mathbf{X(t)} := \begin{pmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_N(t) \end{pmatrix} \tag{2.1}$$

where X_i is a non negative integer that shows how many molecules of species i are present at time t. We also introduce a state change vector $\mathbf{v}_j$,

$$\mathbf{v_j} := \begin{pmatrix} v_{1j} \\ v_{2j} \\ \vdots \\ v_{Nj} \end{pmatrix}, \tag{2.2}$$

The stoichiometric matrix V has v_j as its j-th column where each of the v_{ij} is the change in the number of S_i molecules caused by the reactions. A chemical reaction system consists of N chemical species X_1,..,X_N and M reactions R_1,..,R_M. The state vector is denoted by $X(t) = [X_1(t), ..., X(t)]$, where $X_i(t)$ is represents the amount of S_i molecules at time t. At time $t = 0$, initial population numbers are given.

Associated with the jth reaction is the propensity function [19], $a_jX(t)$: the probability of this reaction R_j taking place in the infinitesimal time interval $[t, t + dt)$ is $a_jX(t)dt$.

$Unimolecular : S_m \xrightarrow{c_j} something$ has propensity function $a_jX(t) = c_jX_m(t)$

$Dimerization : S_m + S_m \xrightarrow{c_j} something$ has $a_j(X(t)) = c_j\frac{1}{2}X_m(t)(X_m(t) - 1)$

$Bimolecular : S_m + S_n \xrightarrow{c_j} something$ has $a_j(X(t)) = c_jX_m(t)X_n(t), where, m \neq n.$ For the *dimerization*, we have $\frac{1}{2}X_m(t)(X_m(t) - 1)$, which represents the number of combinations possible when 2 molecules are chosen from X_m number of total molecules, i.e $\frac{(X_m)!}{(2!)(X_m-2)!}$.

2.2 Chemical Master Equation (CME)

In this section we derive stochastic discrete model of well-stirred biochemical kinetics, the Chemical Master Equation. Now, we study the quantity $P(\mathbf{x}, t)$ defined by:

$$P(\mathbf{x}, t|x_0, t_0) \triangleq Prob\{X(t) = \mathbf{x}, given\ X(t_0) = x_0\} for\ t \geq t_0 \tag{2.3}$$

If we know the probability of being in any state, at time t, then we can find the probability of being in state $\mathbf{x}$ at time $t+dt$, assuming that no more than one reaction can take place over the small interval $[t, t+dt)$. The two situations that can occur is, either the system was already in state $\mathbf{x}$ at time t and no reaction took place over the interval $[t, t+dt)$, or for some $1 \leq j \leq M$ the system was in state $\mathbf{x} - v_j$ at time t and the jth reaction took place over the interval $[t, t+dt)$ which later brought the system to the state $\mathbf{x}$. Using the *law of total probability*, we have that

$$P(A) = \sum_{j=0}^{M+1} P(A|H_j)P(H_j) \tag{2.4}$$

where $H_0, H_1, ..., H_M, H_{M+1}$ represent disjoint (not more than one can happen) and exhaustive (one of them must happen) events and A is the event of interest. If H_0 is the event that the system is in $\mathbf{x}$ at t, let H_j for $1 \leq j \leq M$ be the event for the system is in $\mathbf{x}$-v_j at time t and H_{M+1} be the event that the system is in any other state at time t. Then we have

$$P(A|H_j) = a_j(\mathbf{x} - v_j)dt,\ 1 \leq j \leq M \tag{2.5}$$

If $P(A|H_0)$ is the probability of no reactions taking place in the interval $[t, t+dt)$, then we have

$$P(A|H_0) = 1 - \sum_{j=1}^{M} a_j(\mathbf{x})dt \tag{2.6}$$

This means 1 minus the probability of any reaction happening. We also have that $P(A|H_{M+1}) = 0$ because H_{M+1} contains all the states that are more than one reaction away from $\mathbf{x}$. Using the above four equations, we derive

$$\begin{aligned} P(\mathbf{x}, t+dt) = &\left((1 - \sum_{j=1}^{M} a_j(\mathbf{x})dt\right) P(\mathbf{x}, t) \\ &+ \sum_{j=1}^{M} a_j(\mathbf{x} - v_j)dtP(\mathbf{x} - v_j, t). \end{aligned} \tag{2.7}$$

Arranging the above equation and setting a limit $dt \to 0$, we get

$$\lim_{dt \to\ 0} \frac{P(\mathbf{x}, t + dt) - P(\mathbf{x}, t)}{dt} = \sum_{j=1}^{M} (a_j(\mathbf{x} - v_j)P(\mathbf{x} - v_j, t) - a_j(\mathbf{x})P(\mathbf{x}, t). \tag{2.8}$$

The above equation leads to Chemical Master Equation below:

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{j=1}^{M} (a_j(\mathbf{x} - v_j)P(\mathbf{x} - v_j, t) - a_j(\mathbf{x})P(\mathbf{x}, t)). \tag{2.9}$$

The CME (2.9) is a linear ODE (ordinary differential equation) system with one ODE for each possible state, CME is usually hard to solve because it is a linear ODE system with one ODE for each possible state. If the number of states is large then the CME is a very large system of ODEs, which is very challenging to solve numerically.

2.3 Derivations of Stochastic Simulation Algorithm (Gillespie's Algorithm)

The CME is very high-dimensional and it is computationally intensive to deal with. The Stochastic Simulation Algorithm (SSA) [11, 12] gets around this issue by computing single realizations of the state vector rather than computing an entire probability distribution. The SSA is in exact agreement with the CME. We introduce the probability quantity $P_0(\tau|\mathbf{x}, t)$ as follows: given that $\mathbf{X}(t) = \mathbf{x}$, $P_0(\tau|\mathbf{x}, t)$ is the probability that no reaction takes place in the interval $[t,\ t + \tau)$. Now we consider the time interval $[t,\ t + \tau$+d$\tau)$. We also assume that the event(s) happening over the interval $[\ t,\ t + \tau)$ are independent of the events happening over the interval $[\ t$+τ , $t + \tau$+d$\tau)$. We have that probability that no reaction takes place in $[\ t$,t+τ+d$\tau)$ is probability that no reaction takes place in $[\ t$,t + $\tau)$ and no reaction takes place in $[\ t$+τ, t+τ+d$\tau)$. Mathematically, this translates to independent events, i.e probability that no reaction takes place in $[\ t,\ t + \tau)$ x probability that no reaction takes place in $[\ t$+τ, $t + \tau +$ d$\tau)$ = probability that no reaction takes place in $[\ t,\ t + \tau)$ x (1 - sum of probability that each reactions taking place over the interval $[\ t{+}\tau,\ t+\tau+\mathrm{d}\tau))$. This is written as:

$$P_0(\tau + \delta\tau|\mathbf{x}, t) = P_0(\tau|\mathbf{x}, t)\left(1 - \sum_{k=1}^{M} a_k(\mathbf{x})d\tau\right) \tag{2.10}$$

the above equation is re-arranged and taking the limit $d\tau \to 0$

$$\lim_{dt\to\ 0} \frac{P_0(\tau + \delta\tau|\mathbf{x}, t) - P_0(\tau|\mathbf{x}, t)}{d\tau} = -P_0(\tau|\mathbf{x}, t)\sum_{k=1}^{M} a_k(\mathbf{x})d\tau) \tag{2.11}$$

we get

$$\frac{dP_0(\tau|\mathbf{x}, t)}{d\tau} = -P_0(\tau|\mathbf{x}, t)\sum_{k=1}^{M} a_k(\mathbf{x})d\tau) \tag{2.12}$$

Solving this ODE with $P_0(\tau|x, t) = 1$, gives

$$P_0(\tau|\mathbf{x}, t) = e^{-\sum_{k=1}^{M} a_k(\mathbf{x})\tau} \tag{2.13}$$

Now we need to determine $p(\tau|\mathbf{x}, t)$ defined by $p(\tau, j|\mathbf{x}, t)d\tau$ is the probability that the next reaction (a) will be the *j*th reaction and (b) it will occur in the time interval $[t{+}\tau{,}t{+}\tau{+}\mathrm{d}\tau)$ if $X(t) = X$. The following propositions from Wilkinson [39] will be used in justifying the stochastic simulation algorithm *SSA*.

Proposition 2.3.1. *If $X_i{\sim}Exp(\lambda_i)$ with parameter λ_i, i = 1,2,..,n, are independent exponential random variables, then*

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

Proof. First note that for $X{\sim}Exp(\lambda)$, we have $P(X > x) = Exp(-\lambda x)$. Then

$$\begin{aligned} P(X_0 > x) &= P(\min_i\{X_i\} > x) \\ &= P([X_1 > x]\cap[X_2 > x]\cap...\cap[X_n > x]) \\ &= \prod_{i=1}^{n} P(X_i > x) \\ &= \prod_{i=1}^{n} e^{-\lambda_i x} \\ &= e^{-x\sum_{i=1}^{n}\lambda_i} \\ &= e^{-\lambda_0 x} \end{aligned}$$

So $P(X_0{\leq}x)$ = 1-$e^{-\lambda_0 x}$ and hence $X_0{\sim}Exp(\lambda_0)$ □

This lemma is for the following proposition.

Lemma 2.3.1. *Suppose that $X{\sim}Exp(\lambda)$ and $Y{\sim}Exp(\mu)$ are independent random variables. Then*

$$P(X < Y) = \tfrac{\lambda}{\lambda+\mu}.$$

Proof.

$$\begin{aligned} P(X < Y) &= \int_{o}^{\infty} P(X < Y|Y = y)f(y)dy \\ &= \int_{o}^{\infty} P(X < y)f(y)dy \\ &= \int_{o}^{\infty} (1 - e^{-\lambda y})\mu e^{-\mu y}dy \\ &= \tfrac{\lambda}{\lambda+\mu} \end{aligned}$$

□

This next result gives the likelihood of a particular exponential random quantity of an independent collection being the smallest.

Proposition 2.3.2. *If $X_i{\sim}Exp(\lambda_i)$, i = 1,2,..,n, are independent exponential random variables with parameter λ_i, let j be the index of the smallest number X_i. Then j is a discrete random variable with probability mass function (pmf)*

$$\pi_i = \tfrac{\lambda_i}{\lambda_0},\ i = 1,2,,,n,\ where\ \lambda_0 = \sum_{i=1}^{n}\lambda_i$$

Proof.

$$\begin{aligned}\pi_j &= P(X_j < \min_{i\neq j}\{X_i\}) \\ &= P(X_j < Y)\end{aligned}$$

where Y = $\min_{i\neq j}\{X_i\}$, so that Y$\sim Exp(\lambda_{-j})$, where $\lambda_{-j} = \sum_{i\neq j}\lambda_i$

$$\begin{aligned}&= \tfrac{\lambda_j}{\lambda_j+\lambda_{-j}} \text{ (by the lemma)} \\ &= \tfrac{\lambda_j}{\lambda_0}\end{aligned}$$

□

Using the independence of occurrence of events, we have that probability (a) and (b) = probability that no reaction took place over $[t,t+\tau)$ * probability that the jth reaction took place over $[t+\tau,t+\tau+\mathrm{d}\tau)$ assuming that dτ is so small that at most one reaction took place over that time interval, we have

$$p(\tau, j|\mathbf{x}, t)d\tau = P_0(\tau|\mathbf{x}, t)a_j(\mathbf{x})d\tau \tag{2.14}$$

and from (2.13) we have

$$p(\tau, j|\mathbf{x}, t) \ = \ a_j(\mathbf{x})e^{-\sum_{k=1}^{M}a_k(\mathbf{x})\tau} \tag{2.15}$$

If we let $a_0(\mathbf{x}) = \sum_{k=1}^{M}a_k(\mathbf{x})$, then the equation (2.15) can be re-written as

$$p(\tau, j|\mathbf{x}, t) = \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})}\left(a_0(\mathbf{x})e^{-a_0(\mathbf{x})\tau}\right) \tag{2.16}$$

We express this joint probability density function as a product of two density functions:

- **Next Reaction Index** j is a discrete random variable representing the index of the next reaction.
- **Time Until Next Reaction** the density function of τ is the time to the next reaction.

As the **next reaction index** j and **time until next reaction** τ are independent random variables, we can compute then separately. The Stochastic Simulation Algorithm by Daniel Gillespie is based on [12] a Monte Carlo approach and is as follows:

Stochastic Simulation Algorithm

1. Initiate the system at t_0=0 and $X(t_0) = x_0$
2. At time t evaluate $a_1(x), ..., a_M(x)$, and $a_0(x) \equiv \sum_{j=1}^{M} a_j(x)$
3. Let r_1 and r_2 be two uniform random numbers in (0,1) and compute τ and j as follows:

   a. $\tau = \frac{1}{a_0(x)} \ln \frac{1}{r_1}$

   b. j = the smallest integer satisfying $\sum_{k=1}^{j} a_k(x) > r_2 a_0(x)$
4. Replace $t \leftarrow t + \tau$ and $x \leftarrow x + v_j$.
5. Record (x, t). Return to step 2, else end simulation.

2.4 The Tau-Leaping Method

The SSA is an exact method and therefore it is computationally expensive on systems with fast reaction. In this method, a reaction time and reaction index is computed for each reaction that occurs in the system before adjusting state change vectors and propensity functions. If the system is a stiff system, i.e, if there are many molecules of certain species which participate in fast reaction then that propensity function of such reactions are large aleading to a very small step size τ to the next reaction it is computationally expensive. The exact state of the system at time $t+\tau$ (see Kurtz [8]) is given by the following equation:

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j(\int_t^{t+\tau} a_j(\mathbf{X}(s))ds) \tag{2.17}$$

In the above equation $P_j(\cdot)$ is a Poisson random variable, and $\int_t^{t+\tau} a_j(\mathbf{X}(s))ds$ is its parameter.

- **Leap Condition** We assume that $\tau > 0$ is small enough such that the propensity functions $a_j(\mathbf{X}(t))$, for $j = 1$,...,M remain relatively constant over the interval $t \le s \le t+\tau$, so that few reactions occur, thus $a_j(\mathbf{X}(s)) \approx a_j(\mathbf{X}(t))$.

Using the leap condition and substituting it into equation(2.17), we have

$$\begin{aligned}\mathbf{X}(t+\tau) &\approx \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j(\int_t^{t+\tau} a_j(\mathbf{X}(t))ds)\\ &\approx \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j(a_j(\mathbf{X}(t))\int_t^{t+\tau} ds)\\ &\approx \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j(a_j(\mathbf{X}(t))\tau).\end{aligned}$$

Thus the tau-leaping method is obtained (see Gillespie [14]):

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j(a_j(\mathbf{X}(t))\tau). \tag{2.18}$$

In this equation $\{P_j(a_j(\mathbf{X}(t))\tau)\}_{j=1}^{M}$ represent the random variables which are to be computed in this method.

The probability of the jth reaction taking place over the small time interval of length dτ is given by $a_j(X(t))d\tau$ where $a_j(X(t))$ is almost constant over the time interval [t,t+τ).

Then the number of type j reactions within $[t, t{+}\tau)$ may be computed using $P_j(a_j(X(t))\tau)$ with parameter $a_j(x)\tau$ which is a Poisson distribution with mean and variance $a_j(X(t))\tau$

The algorithm for tau-leaping is computed as SSA:

Tau-Leaping Algorithm

1. Calculate $\{P_j\}_{j=1}^{M}$ from the distribution of Poisson random variables.
2. Initialize the system t_0=0, $x(t_0) = x_0$
3. Then update $\mathbf{X}(t + \tau)$ to $\mathbf{X}(\mathbf{t}) + \sum_{j=1}^{M} \nu_j p_j$
4. Update t to $t + \tau$.
5. Go to 2 or stop.

2.5 Chemical Langevin Equation (CLE)

Suppose [14] that the leap time τ is chosen in such a way that the mean $a_j(\mathbf{X}(t))\tau$ of $P_j(a_j(\mathbf{X}(t))\tau)$ is large for $j = 1, .., M$, i.e, every reaction occurs many times over the interval [t,t+τ). Then we approximate each Poisson random variable by a normal normal random variable with the same mean and variance. According to Stirling's approximation [13], $P_j(a_j(\mathbf{X}(t))\tau){\approx}Z_j(a_j(\mathbf{X}(t))\tau, a_j(\mathbf{X}(t))\tau)$, if $a_j(\mathbf{X}(t)\tau >> 1$. By linear combination theorem for normal random variables, we derive

$$Z(a_j(\mathbf{X}(t)\tau, a_j(\mathbf{X}(t), \tau) = a_j(\mathbf{X}(t)\tau + \sqrt{a_j(\mathbf{X}(t)\tau}Z_j(0, 1). \tag{2.19}$$

Replacing $P_j(a_j(\mathbf{X}(t)\tau))$ in (2.18) with $a_j(\mathbf{X}(t))\tau + \sqrt{a_j(\mathbf{X}(t))\tau}Z_j$, according to (2.19), we get Z_j where Z_j are independent normal (0,1) random variables.

$$X(t+\tau) = X(t) + \tau \sum_{j=1}^{M} v_j a_j(X(t)) + \sqrt{\tau} \sum_{j=1}^{M} v_j \sqrt{a_j(X(t))} Z_j(0,1). \tag{2.20}$$

Equation (2.20) is known as the Chemical Langevin Equation (as CLE). We also notice that the tau-leaping method uses an integer-valued Poisson random variables, while (2.20) employs real-valued normal random variables. In the limit $\tau \rightarrow dt$ (2.19) becomes a stochastic differential equation of the form:

$$dX(t) = \sum_{j=1}^{M} v_j [a_j(X(t))dt + \sqrt{a_j(X(t))} dW_j(t)] \tag{2.21}$$

where W_j, $1 \leq j \leq M$, are independent Wiener processes.

Definition 2.5.1. *A scalar standard [18] Wiener process over [0,T] is a random variable W(t) that depends continuously on $t \in$ [0,T] and satisfies the following three conditions:*

1. *W(0) = 0 (with probability 1)*

2. *For any 0≤s < t≤T the random variable given by the increment W(t)-W(s) is normally distributed with mean zero and variance (t-s); equivalently, W(t)-W(s) ∼ $\sqrt{t-s}N(0,1)$ where $N(0,1)$ denotes a normally distributed random variable with zero mean and variance 1.*

3. *For any 0≤s < t < u < v≤T the increments W(t)-W(s) and W(v)-W(u) are independent.*

Note that equation (2.20) is the Euler-Maruyama method for applied to the stochastic differential equations (2.20). The assumptions used to derive the CLE are:

i the propensity functions do not change significantly over the time interval [t,t+τ),

ii the value of each propensity function times the step size, i.e. $a_j(X(t))\tau$, is large.

The condition (i) and (ii) applies simultaneously when the molecular populations of each species is large.

2.6 Derivation of Reaction Rate Equation

We investigate the assumptions under which the Chemical Master Equation (*CME*) can be reduced to the continuous deterministic model of the reaction rate equation (*RRE*). If K is the countable state space of the process X_t satisfying the CME, then (see [39] for more detail).

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

where K is the countable state space of the process. By applying equation (2.9) we get

$$\begin{aligned}\frac{\partial}{\partial t}E(X_t) &= \sum_{\mathbf{x}\in K}\mathbf{x}\frac{\partial}{\partial t}P(\mathbf{x},t|\mathbf{x}_0,t_0)\\ &= \sum_{\mathbf{x}\in K}\mathbf{x}\left[\sum_{j=1}^{M}(a_j(\mathbf{x}-v_j)P(\mathbf{x}-v_j,t|\mathbf{x}_0,t_0)-a_j(\mathbf{x})P(x,t|\mathbf{x}_0,t_0)\right]\\ &= \sum_{j=1}^{M}\left[\sum_{\mathbf{x}\in K}\mathbf{x}(a_j(\mathbf{x}-v_j)P(\mathbf{x}-v_j,t|\mathbf{x}_0,t_0)-\sum_{\mathbf{x}\in K}\mathbf{x}a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right]\end{aligned}$$

We can substitute $\mathbf{x}+v_j$ for $\mathbf{x}$ because $\mathbf{x}+v_j\in K$ and derive

$$\frac{\partial}{\partial t}E(X_t) = \sum_{j=1}^{M}\left[\sum_{\mathbf{x}\in K}(\mathbf{x}+v_j)(a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)-\sum_{\mathbf{x}\in K}\mathbf{x}a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right]$$

Simplifying this equation we obtain

$$\frac{\partial}{\partial t}E(X_t) = \sum_{j=1}^{M}v_ja_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)$$

By the definition of expected values, the above equation can be written as

$$\frac{\partial}{\partial t}E(X_t) = \sum_{j=1}^{M}[v_jE(a_j(X_t))].$$

Substituting $Y(t) = E(X_t)$ in the above, we get

$$\frac{d}{dt}Y(t) = \sum_{j=1}^{M} v_j E(a_j(X_t)$$

We now investigate the types of reactions is the reaction rate equation is valid,

$$E(a_j(X_t)) = a_j(E(X_t)) = a_j(Y(t))$$

Consider the following reactions of various orders:

1. If $a_j(X(t)) = m_j$, then $E(a_jX(t)) = E(m_j) = m_j = a_j(Y(t))$.
2. If $a_j(X(t)) = m_jx_i$, then $E(a_j(X(t))) = E(m_jx_i) = m_jE(x_i) = a_j(E(X(t))) = a_j(Y(t))$,
3. If $a_j(X(t)) = m_jx_ix_k$, where $i \neq k$, the population species S_i and S_k may not be independent, then $E(a_j(X(t))) = E(m_jx_ix_k) = m_jE(x_ix_k) \neq m_jE(x_i)E(x_j) = a_j(Y(t))$.

Thus the average behaviour of the solution of the CME satisfies the RRE if the biochemcal system has only reactions of order 0 and 1. The system reaches the *thermodynamic limit* (Gillespie [15]) when its volume V and each species population $X_i(t)$ approach infinity, while the species concentration (i.e. species population/volume) remains constant, i.e., if $X_i \rightarrow \infty$ and $V \rightarrow \infty$ then $\frac{X_i}{V} = constant$. When there are large number of molecules, the deterministic part of the CLE grows as the system size but the stochastic part grows as its square root. Therefore the stochastic part is negligible compared to the deterministic part.

So, from equation (2.21), if we ignore the stochastic part of the CLE then we get):

$$\frac{d}{dt}Y(t) = \sum_{j=1}^{M} v_j a_j(Y(t)). \tag{2.22}$$

This equation is called the reaction rate equation (*RRE*). Therefore when the system has very large number of molecules, it can be approximated by using the RRE model.

In biochemical stochastic models where the molecular number of each species is greater than 1000, we are able to reduce the stochastic model of the CME (or CLE) to that of the deterministic reaction-rate equations.

Chapter 3

The Multi-Level Monte Carlo Method

Monte Carlo methods have been extensively used in computational finance to approximate the expected value of a quantity of interest, which is a function of a stochastic process. This stochastic process may be the solution of a stochastic differential equation (SDE). Consider a SDE of the form:

$$dS(t) = a(S,t)dt + b(S,t)dW(t),\ 0 < t < T, \tag{3.1}$$

where $a(\cdot)$ and $b(\cdot)$ are the drift and diffusion terms, respectively. If S_0 is the initial data (in our case, the initial number of molecules of a species), the goal is to compute the expected value of $f(S(T))$. The function f satisfies the uniform Lipschitz condition, i.e., there exists a constant $c > 0$ and a domain D such that for any U and V in D

$$|f(U) - f(V)| \leq c|U - V|. \tag{3.2}$$

Then discretization of this SDE using the Euler-Maruyama method with time step h yields

$$S_{n+1} = S_n + a(S_n, t_n)h + b(S_n, t_n)\Delta W_n \tag{3.3}$$

and the estimate for $E[f(S_T)]$ is the mean of $f(S_{T/h})$ from N independent path simulations [10]

$$Y = \frac{1}{N}\left[\sum_{i=1}^{N} f(S^{(i)}{}_{T/h})\right] \tag{3.4}$$

The expected mean square error (MSE) is of the form [10]

$$MSE \approx c_1 N^{-1} + c_2 h^2 \tag{3.5}$$

where c_1 and c_2 are constants. In this equation, the first term corresponds to the variance in Y and the second term comes from the biasness of Euler Maruyama discretizations.

In the multi-level Monte Carlo method developed by Giles [10] the time steps for the discretizations are given by $h_l = M^{-l}T$, $l = 0$, 1,..., L, for integer $M \geq 2$. This is a geometric sequence of time steps with h_L being the smallest time step. In a multi grid method, the stepsize is half the previous stepsize in the previous grid. On a fine grid, the accuracy is much better but the computational cost is high. On a coarse grid the accuracy is lower but the computational cost is low. In this Chapter, we describe the multi-level Monte Carlo (MLMC) method applied to the stochastic discrete model of well-stirred biochemical systems, the CME and how does it work in order to compute the expected values efficiently.

3.1 MLMC General Idea

The stochastic process $X(t)$ governed by the CME satisfies the following equation (see Kurtz [8])

$$\mathbf{X}(t+\tau) \;=\; \mathbf{X}(t) + \sum_{j=1}^{M} \nu_j P_j\left(\int_t^{t+\tau} a_j(\mathbf{X}(s))ds\right).$$

If ${p_j}^1, {p_j}^2, .., {p_j}^m$ are independent Poisson processes then

$${P_j}^1\left(\int_0^{\tau} a_j(\mathbf{X}(s))ds\right) + {P_j}^2\left(\int_{\tau}^{2\tau} a_j(\mathbf{X}(s))ds\right) + ... + {P_j}^m\left(\int_{(m-1)\tau}^{t} a_j(\mathbf{X}(s))ds\right)$$

$$\begin{aligned} &= Y_j\left(\int_0^{\tau} a_j(\mathbf{X}(s))ds + \int_{\tau}^{2\tau} a_j(\mathbf{X}(s))ds + \int_{(m-1)\tau}^{t} a_j(\mathbf{X}(s))ds\right) \\ &= Y_j(\int_0^t a_j(\mathbf{X}(s))ds) \end{aligned}$$

where Y_j are unit rate Poisson processes and $(m-1)\tau \leq t \leq m\tau$. For the state vector $X(t)$ which obeys the CME satisfies the the random time change representation [8])

$$X(t) \;=\; X(0) + \sum_{j=1}^{R} Y_j\left(\int_0^t a_j(X(s)ds)\right) v_j, \tag{3.6}$$

where v_j and a_j are the state change vector and the propensity corresponding to reaction R_j, respectively and $\{Y_j\}_{j=1}^{M}$ are independent unit-rate Poisson processes. We demonstrate the use of the equation (3.6) in the biochemical reaction system $S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2$. For example in the reaction, $S_1 \xrightarrow{c_1} S_2$, S_1 molecules are converted to S_2 molecules at a rate of $c_1 X_1$, X_1 being the number of molecules of type S_1. If the system satisfies the mass action kinetics, the stochastic process $X(t) = \left(X_1(t), X_2(t)\right)^T$ can be represented as

$$X(t) = X(0) + Y_1\left(\int_0^t c_1 X_1(s)ds\right)(-1,\ 1)^T + Y_2\left(\int_0^t c_2 X_2(s)ds\right)(1,\ -1)^T \tag{3.7}$$

where $(-1,\ 1)^T$ and $(1,\ -1)^T$ are the state change vectors form the chemical reactions above. To solidify our notation, consider a network $S_1 \xrightarrow{c_1}$, $S_2 \xrightarrow{c_2} S_1$, $2S_2 \xrightarrow{c_3} S_3$ and $2S_2 \xrightarrow{c_3} S_3$. Here c_1, c_2 and c_3 are rate constants. Then, the state vector of this biochemical system obeys;

$$\begin{aligned} X(t) = X(0) + Y_1\left(\int_0^t c_1 X_1(s)ds\right)(-1,\ 1,\ 0)^T + Y_2\left(\int_0^t c_2 X_2(s)ds\right)(1,\ -1,\ 0)^T \\ +Y_3\left(\int_0^t \frac{1}{2} c_3 X_2(s)(X_2(s)-1)ds\right)(0,\ -2,\ 1)^T \end{aligned}$$

where $v_1 = [-1, 1, 0]^T$, $v_2 = [1, -1, 0]^T$, $v_3 = [0, -2, 1]^T$ are the state change vectors.

3.2 The MLMC Method for Stochastic Biochemical Kinetics

In this section, we will present the multi-level Monte Carlo method to estimate the mean of $f(X(t))$ where f is a function of interest, e.g., $f(x) = x$ and $X(t)$ is the stochastic process representing the state vector of a well-stirred biochemical system. In order to reduce the computational costs, the multi-level Monte Carlo strategy couples the trajectories to accurately approximate the expected value of a random variable for rather than the entire distribution. By coupling trajectories on consecutive levels, the method reduces the variance of the mean estimate. By levels, we are referring to computing estimates using finer step sizes in higher consecutive levels. From our sample distribution, the MLMC method finds the moments of the distribution such as mean if $f(x) = x$ and variance, if $f(x) = x^2$ in case the mean was computed in advance.

The multi-level Monte Carlo method generates many levels and each level corresponds to an estimate. The summation of the estimates from all the levels is employed to our approximate the desired mean, expected value of $f(X(t))$. For instance, on level 0, the tau-leaping method is used to generate a larger number of sample paths (n_0). Then the point estimate for X_i is

$$\mathbf{Q_0} := E[Z_{\tau_0}] \approx \frac{1}{n_0} \sum_{r=1}^{n_0} Z_{\tau_0}^{(r)}(T), \tag{3.8}$$

where $Z_{\tau}^{(r)}$ is the number of molecules of the species of interest at time T in path r generated using the tau-leaping method with time step τ. If τ is large, the estimates calculated are cheap, but inaccurate.

In the next level (level 1), we introduce a correction term to the estimator which will reduce the bias. In order to improve the accuracy of the estimator, we calculate two sets of n_1 sample paths. The first set of sample paths is computed using the tau (τ_0) from the previous level. The second set of trajectories is calculated by using the step size $\tau_1 = \tau_0$/K, where $K \geq 2$ integer. Then the correction term is the difference between the estimates calculated using two sample n_1 paths

$$\mathbf{Q_1} := E[Z_{\tau_1} - Z_{\tau_0}] \approx \frac{1}{n_1}\sum_{r=1}^{n_1}[Z_{\tau_1}^{(r)}(T) - Z_{\tau_0}^{(r)}(T)]. \tag{3.9}$$

Adding this correction term to the estimator calculated on the base level (level 0) reduces the bias of the resulting estimator. We can note that

$$Q_0 + Q_1 \;\;=\;\; E[Z_{\tau_0}] + E[Z_{\tau_1} - Z_{\tau_0}] = E[Z_{\tau_1}].$$

So adding the two estimators gives a bias equivalent to that of the tau-leaping method with $\tau = \tau_1$. This process is repeated for the subsequent levels. Summing up the estimates obtained from all the levels leads to a more accurate estimator, equivalent in accuracy to an approximation by the tau-leaping method with the stepsize of the finest mesh. The way to develop efficiency of the multi-level method is to generate two sets of sample paths

$$\left\{Z_{\tau_1}(T)^{(r)}, Z_{\tau_0}(T)^{(r)} : r = 1, ..., n_1\right\} \tag{3.10}$$

so that the variance in their difference is minimized. If the variance in their difference is denoted by V_l, then the estimator variance is given as $\dfrac{V_l}{n_l}$. A lower variance implies that fewer sample paths are needed to achieve the same accuracy of the estimation. On the second level, this process is repeated to generate a second correction term. In this level, two sets of n_2 sample paths are generated. One set has $\tau = \tau_1$ and the second has $\tau = \frac{\tau_1}{K} = \tau_2$, such that $\tau_2 < \tau_1$. The correction term is the estimator of their difference

$$\mathbf{Q_2} := E[Z_{\tau_2} - Z_{\tau_1}] \approx \frac{1}{n_2}\sum_{r=1}^{n_2}[Z_{\tau_2}^{(r)}(T) - Z_{\tau_1}^{(r)}(T)]. \tag{3.11}$$

Adding the correction term generated from level 2 to level 1 and level 0, we derive

$$Q_0 + Q_1 + Q_2 \;\;=\;\; E[Z_{\tau_0}] + E[Z_{\tau_1} - Z_{\tau_0}] + E[Z_{\tau_2} - Z_{\tau_0}] = E[Z_{\tau_2}]$$

by the linearity property of expected values. Thus the summation of the three estimators gave a bias equivalent to the tau leaping method with $\tau = \tau_2$. Continuing this procedure gives the following telescopic sum

$$E[Z_{\tau_l}] = E[Z_{\tau_0}] + \sum_{l=1}^{L} E[Z_{\tau_l} - Z_{\tau_{l-1}}] = \sum_{l=0}^{L} Q_l \tag{3.12}$$

When the correction terms are added to subsequent levels, the bias of the estimator is reduced until a desired accuracy is achieved. Finally and optionally, we can generate two sample of n_{L+1} sample paths, one set using the expected values obtained by *SSA* and the other using tau-leaping with $\tau = \tau_L$, we can compute the final correction term

$$Q_{L+1} = E[X_i - Z_{\tau_l}] \approx \frac{1}{n_{L+1}} \sum_{r=1}^{n_{L+1}} [X_i{}^{(r)}(T) - Z_{\tau_L}^{(r)}(T)] \tag{3.13}$$

and the above estimation can be added to the telescopic sum in order to get an unbiased estimator below

$$Q_b = E[X_i] = E[Z_{\tau_0}] + \sum_{l=1}^{L} E[Z_{\tau_l} - Z_{\tau_{l-1}}] + E[X_i - Z_{\tau_L}] = \sum_{l=0}^{L} Q_l + Q_{L+1}^{*} \tag{3.14}$$

The above equation will give us an unbiased estimation of the quantity of interest, in our case the mean $E(X(T))$. It is important to note that the total CPU time taken to generate the estimates from all the levels is less than the total CPU time required to estimate X_i using the exact stochastic simulation algorithm for an insignificant loss in accuracy of the estimation.

To use the multi-level Monte Carlo method, we need to consider the following:

- the choice of the number of levels in the algorithm as it will affect the total computational time,
- the value of the estimator variance of each level, $\dfrac{V_l}{n_l}$ will ensure accuracy and also affect the total computational time.

The key idea in this method is to reduce the overall estimator variance V_l by making the variance from each levels sufficiently small so that fewer trajectories are needed on each level to obtain an estimation of the desired accuracy reducing the computational time.

An important point to make is, that the error of a Monte Carlo is

$$error_l \sim \frac{\sigma_l}{\sqrt{N_l}}. \tag{3.15}$$

where ${\sigma_l}^2$ is the variance and N_ℓ is the number of Monte Carlo trajectories. We can see from the relation between error and variance that, if the variance is reduced, then fewer trajectories are required for the corresponding level to obtain a similar error in the estimation

Coupling in MLMC

The algorithm below explains how the coupling is done in order to reduce the variance [3].

Fix an integer $M \geq$ *2* . *Fix* $h_l > 0$ and set $h_{l-1} = M * h_l$. Set $Z_l(0) = Z_{l-1}(0) = x_0$, $t_0 = 0$, $n = 0$. *repeat the following steps until* $t_n \geq T$:

1. *For j = 1, ..., M,*

   (a) *Set*

   - $m_{k,1} = \min(a_k(Z_l), a_k(Z_{l-1}))$
   - $m_{k,2} = a_k(Z_l) - m_{k,1}$
   - $m_{k,3} = a_k(Z_{l-1}) - m_{k,1}$

   (b) *For each k, let*

   - $p_{k,1} = Poisson(m_{k,1} * h_l)$
   - $p_{k,2} = Poisson(m_{k,2} * h_l)$
   - $p_{k,3} = Poisson(m_{k,3} * h_l)$

- $P_k = p_{k,1} + p_{k,2}$
- $Q_k = p_{k,1} + p_{k,3}$
- $G_k = G_k + P_k * v_k$
- $F_k = F_k + Q_k * v_k$

(c) it Set

- $Z_l = Z_l + G_k$
- $Z_{l-1} = Z_{l-1} + F_k$

2. *Set* $t_{n+1} = t_n + h_{l-1}$.

3. *Set* $n = n + 1$

Some of the observations we make are:

1. We do not need to update $a_k(Z_{l-1})$ during the workings of the inner loop of j = *1,...,M.*

2. At most one of m_2, m_3 will be nonzero during each step with both being zero whenever $a_k(Z_l) = a_k(Z_{l-1})$. Therefore at most two Poisson random variables are required per reaction channel at each step and not three.

3. While two paths are being generated, $\max\{m_2, m_3\}$ should be small for each step. Hence the work in computing Poisson random numbers will fall on $p_{k,1}$ [1], which is the same amount of work as would be needed for the generation of a single path of tau-leaping. Note that $p_{k,1}$ is a common Poisson random number for the coupled trajectories. Since $p_{k,1}$ is larger than $p_{k,2}$ and $p_{k,3}$, the coupling is strong, and therefore the variance of the estimation is reduced.

If each sample path on level ℓ takes c_l time to generate the estimator using n_l sample paths, then we minimise the total computational time according to the method proposed

[1] The cost of generating a Poisson random variable generally decreases with the size of the mean.

in [25]

$$\min_{n_l}\sum_{l=0}^{L} n_l c_l \text{ such that } \sum_{l=0}^{L}\frac{V_l}{n_l} < \epsilon^2 \tag{3.16}$$

Here ϵ^2 controls the estimator variance. Consider a $\lambda \in R$ and the Lagrange function,

$$L(c_l, n_l, V_l, \lambda) = \left\{\sum_{l=0}^{L} n_l c_l + \lambda \sum_{l=0}^{L}\frac{V_l}{n_l}\right\}.$$

We wish to find λ such that $\frac{\partial L}{\partial n_l} = 0$. Calculating the derivative with respect to n_l gives

$$\frac{\partial}{\partial n_l} L(c_m, n_m, V_m, \lambda) = \frac{\partial}{\partial n_l}\left\{\sum_{m=0}^{L} n_m c_m + \lambda \sum_{m=0}^{L}\frac{V_m}{n_m}\right\} = 0,$$

for ℓ = 0, 1,...,L We know that

$$\frac{\partial n_m}{\partial n_l} = \delta_{lm} = \begin{cases} 1 & \text{if } m = l, \\ 0 & \text{if } m \neq l \end{cases}$$

Thus we get

$$\frac{\partial}{\partial n_l} L(c_m, n_m, V_m, \lambda) = \frac{\partial}{\partial n_l}\left\{\sum_{m=0}^{L} n_m c_m + \lambda \sum_{m=0}^{L}\frac{V_m}{n_m}\right\} = \sum_{m=0}^{L} c_m \delta_{lm} + \lambda \sum_{m=0}^{L}\frac{V_m \delta_{lm}}{(-{n_m}^2)} = 0$$

leading to

$$c_l - \lambda \frac{V_l}{{n_l}^2} = 0.$$

Therefore the number of trajectories on level ℓ, n_ℓ minimizing the quantity in (3.16) is

$$n_l = \sqrt{\lambda \frac{V_l}{c_l}}. \tag{3.17}$$

Recall that

$$\sum_{m=0}^{L}\frac{V_m}{n_m} \leq \epsilon^2, \tag{3.18}$$

Substituting n_l from (3.17) into (3.18), we get

$$\sum_{m=0}^{L} \frac{V_m}{n_m} = \sum_{m=0}^{L} \sqrt{V_m c_m} \frac{1}{\sqrt{\lambda}} \leq \epsilon^2$$

Thus

$$\sqrt{\lambda} \geq \frac{1}{\epsilon^2} \sum_{m=0}^{L} \sqrt{V_m c_m} \tag{3.19}$$

To get the optimal n_l from (3.17) with λ satisfying (3.19). We choose

$$n_l = \frac{1}{\epsilon^2} \left\{ \sum_{m=0}^{L} \sqrt{V_m c_m} \right\} \sqrt{\frac{V_l}{c_l}}. \tag{3.20}$$

We can use the values n_l for the number of trajectories at level l. When the values of V_l and c_l are available it is also possible to estimate c_l ans $c_l \approx \frac{K^l}{\tau_0}$, whereas the variances V_l may not be known but we can estimate them from simulating few sample paths.

In order to calculate the total computational cost, we let c_l and V_l be the cost and variance of a sample path Z_l - Z_{l-1}, then the total computational cost of the multi-level estimator is $\sum_{l=0}^{L} n_l c_l$. We can choose according to (3.20), then the total computational cost is

$$\mathbf{c} = \epsilon^{-2} \left(\sum_{l=0}^{L} \sqrt{V_l c_l} \right)^2 . \tag{3.21}$$

3.2.1 An MLMC Method Using Scaling

Another MLMC method for stochastic discrete biochemical system was developed by Anderson and Higham [3]. This method uses a different strategy for calculating the number of trajectories needed for each levels, based on the largest molecular amount. Consider V, to be the largest initial molecular amount of a species. The species population will vary over the period of time during simulation, the best guess we have is the largest initial species number. This approach scales the molecular number of species S_i by parameter $\alpha_i \geq 0$ by setting ${X_i}^v(t) = V^{-\alpha_i}X_i(t)$ so that ${X_i}^v \ = \ O(1) \ = V^{-\alpha_i}X_i$ such that $O(1) = V^{-\alpha_i}X_i(0)$ for each species S_i. Here ${X_i}^v(t)$ is the abundance of for each species S_i. We have to carefully pick α_i such that ${X_i}^v = O(1)$. We want α_i to be large enough but not so large that ${X_i}^v$ converges to zero as $V{\rightarrow}\infty$. If we take logarithms on both sides, then we can rewrite the above equation as

$$\alpha_i = \log_V(X_i(t))$$

The general form of such a scaled model is

$$X^V(t) \ = \ X^V(0) + \sum_k Y_k \left(V^\gamma \int_0^t V^{c_k} a_k(X^V(s))ds \right) \zeta^V,$$

where Y_k is a Poisson process, γ and c_k are scalars, $|\zeta^V| = O(V^{-c_k})$, and both X^v and $a_k(X^V)$ are of order one since ${X_i}^V = O(1)$. Here ${a_k}^V$ is a propensity function and depends on V. It is also natural to have $\bar{V} = V^\gamma \sum_k V^{c_k}$ [3] as the order of magnitude for the number of computations required to generate a single realisation using exact algorithm. The parameter γ is interpreted as being related to the time scale model. If $\gamma > 0$, then the shortest time scale (time taken by chemical reactions) in the problem is much smaller than 1 and if $\gamma < 0$ it is much larger. The optimum is to have $\gamma{\leq}0$ in order to avoid the error bounds grow rapidly.

Similarly for each reaction rates c_i, each of these reaction rates are scaled according to $c_j = O(1)V^{\beta_j}$ and if we take logarithm on both sides then we can rewrite as

$$\beta_j = \log_V(c_j)$$

The scaling factor for the reaction as a whole is γ, calculated as

$$\gamma = \max_{i,j:\mathbf{v}_{ij}\neq 0} \{\beta_j + v_j\alpha - \alpha_i\}$$

Here v_{ij} is the (i,j) entry of the stoichiometric matrix and v_j is reaction R_j and α is a vector containing all the α_i [21]. We also need a parameter ρ which is computed as follows:

$$\rho = \min_{i,j:\mathbf{v}_{ij}\neq 0}\{\alpha_i\}$$

so that $|{\zeta_k}^V| \approx V^{-\rho_k}$. we have that $\rho{\geq}0$ and by the choice of γ we have $c_k - \rho_k{\leq}0$ for all k. Aso, we point that γ is chosen such that $c_k{=}0$ for at least one k. It is explicitly noted that the classical scaling holds if and only if $c_k{\equiv}\rho_k{\equiv}1$ and $\gamma = 0$ [3]. Then, the required number of trajectories for the base estimator, n_0, is calculated as

$$n_0 = V^{-\rho}V^{\gamma}\epsilon^{-2},$$

where ϵ is the desired accuracy of the estimator for $E(f(X(t)))$. The number of trajectories required to calculate correction estimators at level l, n_l is calculated as

$$n_l = V^{-\rho}V^{\gamma}(L - l_0)h_l\epsilon^{-2}$$

where $\ell \in \{$0,1,...,L$\}$, the step size $\tau_l = TM^{-\ell}$ and L is chosen by taking $L = O(|\ln(\epsilon^{-1})|)$. Recall that we can estimate the expected value at the finest level may be computed as

$$E[Z_{\tau_l}] = E[Z_{\tau_0}] + \sum_{l=1}^{L} E[Z_{\tau_l} - Z_{\tau_{l-1}}] = \sum_{l=0}^{L} Q_l.$$

We can estimate the above quantities using n_0 and $\{n_l\}_{1\leq\ell\leq L}$ within an error of $O(\epsilon)$ by evaluating the right side of the above equation. As discussed earlier the implementation of the MLMC can be done with any desired accuracy but with more levels comes higher computational cost for obtaining a higher accuracy. Our experiment with this approach showed that it is not the most efficient when applied to a wide class of models. However, it could be said that there could be other models that we have not tested but might be compatible with this approach.

Chapter 4

Numerical Results

In this section, we test the MLMC method described in Chapter 3 on three models of practical interest. We do so by simulating expected values using 10,000 trajectories with *SSA* and the *MLMC* method with a certain tolerance, ϵ.

Our test show that a significant amount of computational time is saved using the *MLMC* strategy with a lower tolerance compared to the *SSA*. which is an exact method, but comes at a higher computational cost.

We plotted the expected values for the molecular amounts of various species, obtained using the exact method of Gillespie [12] (SSA) and with the MLMC method as functions of time and found good agreement in the results of the two methods. For each model we also got the probability distribution of the molecular amount of each species, obtained using 10,000 SSA trajectories.

4.1 Potassium Channel (Model 1)

Consider the system the Potassium Channel system studied in [28],

$$S_1 \xrightarrow{c_1} S_2$$
$$S_2 \xrightarrow{c_2} S_1$$

$$\begin{aligned}
S_2 &\xrightarrow{c_3} S_3\\
S_3 &\xrightarrow{c_4} S_2\\
S_3 &\xrightarrow{c_5} S_4\\
S_4 &\xrightarrow{c_6} S_3\\
S_4 &\xrightarrow{c_7} S_5\\
S_5 &\xrightarrow{c_8} S_4\\
S_5 &\xrightarrow{c_9} S_3\\
S_3 &\xrightarrow{c_{10}} S_5
\end{aligned}$$

The propensity functions corresponding to the reactions above are:

$$\begin{aligned}
a_1(x) &= c_1x_1;\\
a_2(x) &= c_2x_2;\\
a_3(x) &= c_3x_2;\\
a_4(x) &= c_4x_3;\\
a_5(x) &= c_5x_3;\\
a_6(x) &= c_6x_4;\\
a_7(x) &= c_7x_4;\\
a_8(x) &= c_8x_5;\\
a_9(x) &= c_9x_5;\\
a_{10}(x) &= c_{10}x_3;
\end{aligned}$$

Here x_i's correspond to the molecular population number of species S_i. The stoichiometric matrix for this biochemical network is:

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

This model has the following reaction rate, parameters and initial conditions:
X(1) = 100; X(2) = 50; X(3) = 100; X(4) = 50; X(5) = 100, c(1) = c(2) = c(3) = c(4) = c(5) = 0.1. The interval of integration is [0,20]

	SSA	MLMC(ϵ_1=0.5)	MLMC(ϵ_2=1)	MLMC(ϵ_3=1.5)	MLMC(ϵ_4=2)
X1	81.1214	81.1122	80.8897	81.36	81.0312
X2	80.2564	80.0417	79.1176	78.5467	81.1562
X3	79.7566	79.7396	80.3015	80.44	80.2812
X4	79.3184	79.3871	79.6838	78.0024	78.9688
X5	79.5472	79.7194	80.0074	78.4133	78.5624
CPU_{time}	412s	490s	73s	36s	12s

Table 4.1: Comparison of expected values obtained through SSA and MLMC (using different ϵ-values) for the Potassium Channel Model and the computational times of these algorithms

Levels n_ℓ	MLMC(ϵ_1=0.5)	MLMC(ϵ_2=1)	MLMC(ϵ_3=1.5)	MLMC(ϵ_4=2)
n_0	695	136	75	32
n_1	570	108	60	23

Table 4.2: Comparison of trajectories obtained through simulation of the MLMC (using different ϵ-values) for the Potassium Channel Model

We present in table 4.1 the estimates of the expected values of the X_i molecules obtained from simulating the Potassium Channel model using both the SSA and the MLMC methods. The SSA is an exact method whereas MLMC is an approximate method. We have discussed earlier that while being an exact thus accurate method, SSA simulation comes at a high cost for models with some fast reactions. However, the MLMC strategy provides significant computational time savings without significant loss in accuracy of estimating the mean values. We see that the the expected values obtained from both methods are very close to each other, with a difference of approximately 1, 2 or 3 at most number molecules for some species. For example, consider species 5, the SSA simulation gives us 79.5472 whereas MLMC simulation gives 78.5624 (by using tolerance ϵ_4=2). The relative error of MLMC compared to SSA ones is $\frac{|79.5472-78.5624|}{|79.8148|} * 100\% \approx 1.2\%$. This shows a very good accuracy of our simulation. We also see from Table 4.1 that, the MLMC scheme is approximately 20 (412/20) times faster than the SSA. Figures 4.6-4.10 show the expected values for the molecular amounts of the species $X_1, X_2, ..., X_5$ respectively as functions of times estimated from 10000 trajectories utilizing the *SSA* and the *MLMC* strategy with a given tolerance. Figure 4.11 gives a loglog plot of the norm-2 of the absolute error of the MLMC compared to the 'exact' solution of the SSA, as a function of the tolerance ϵ set for the estimator. The slope in the loglog plot is almost 1.

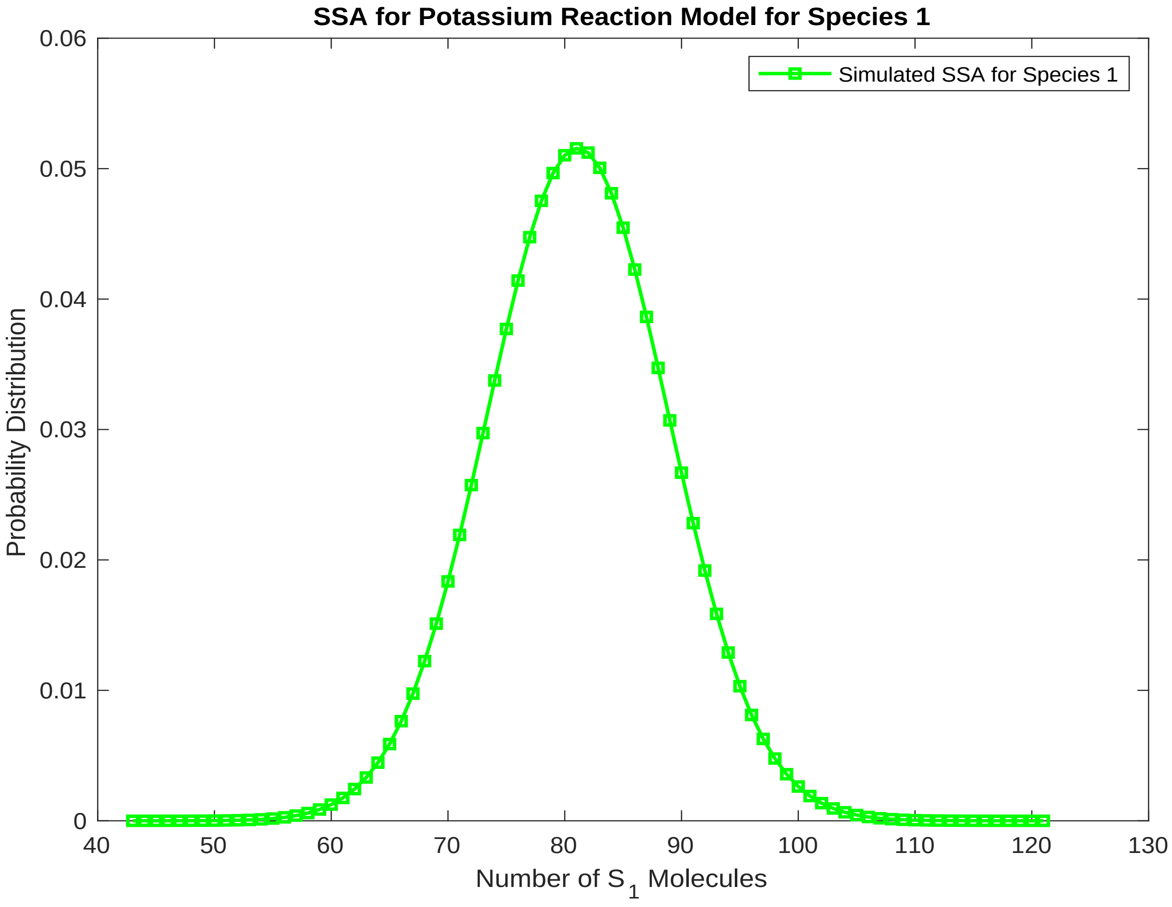

Figure 4.1: Potassium Reaction Model; probability distribution of species S_1 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 1 is 412 seconds.

,

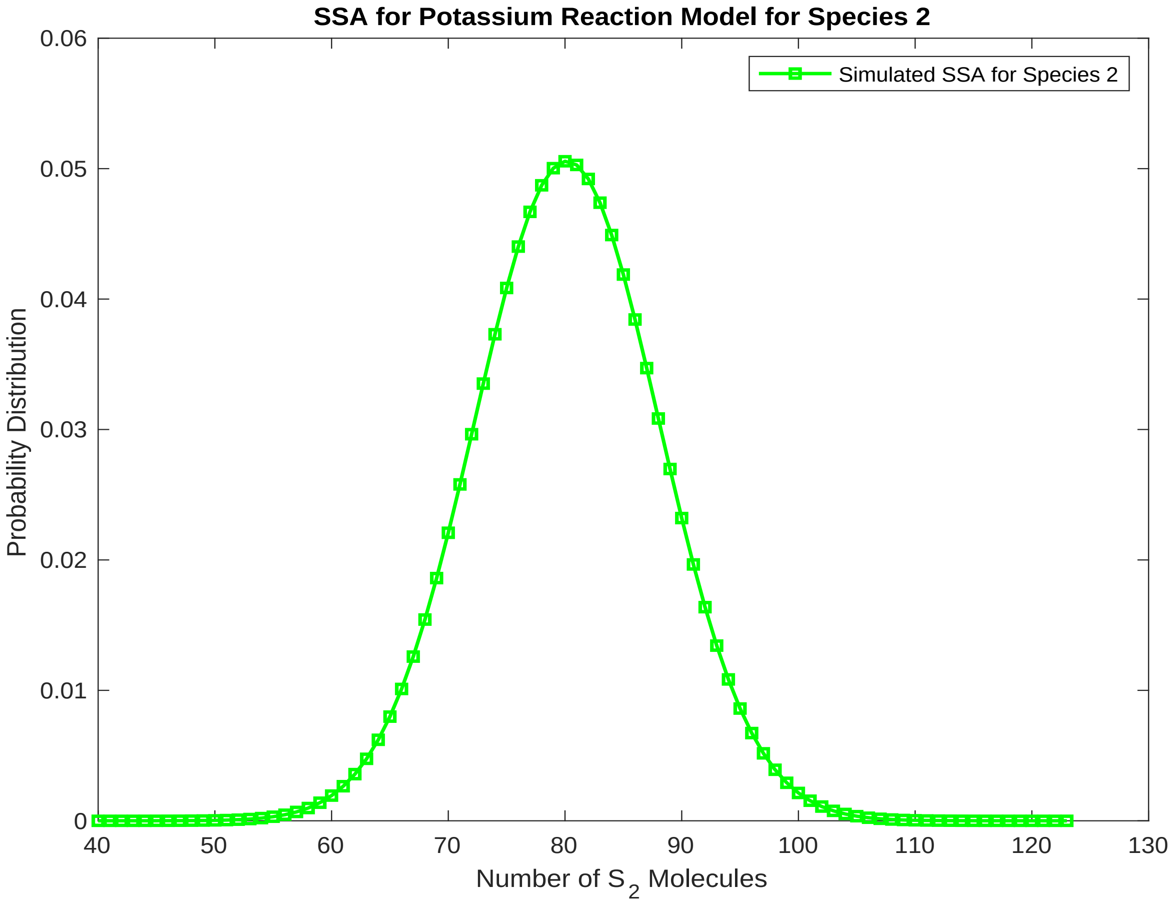

Figure 4.2: Potassium Reaction Model; probability distribution of species S_2 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 2 is 412 seconds.

,

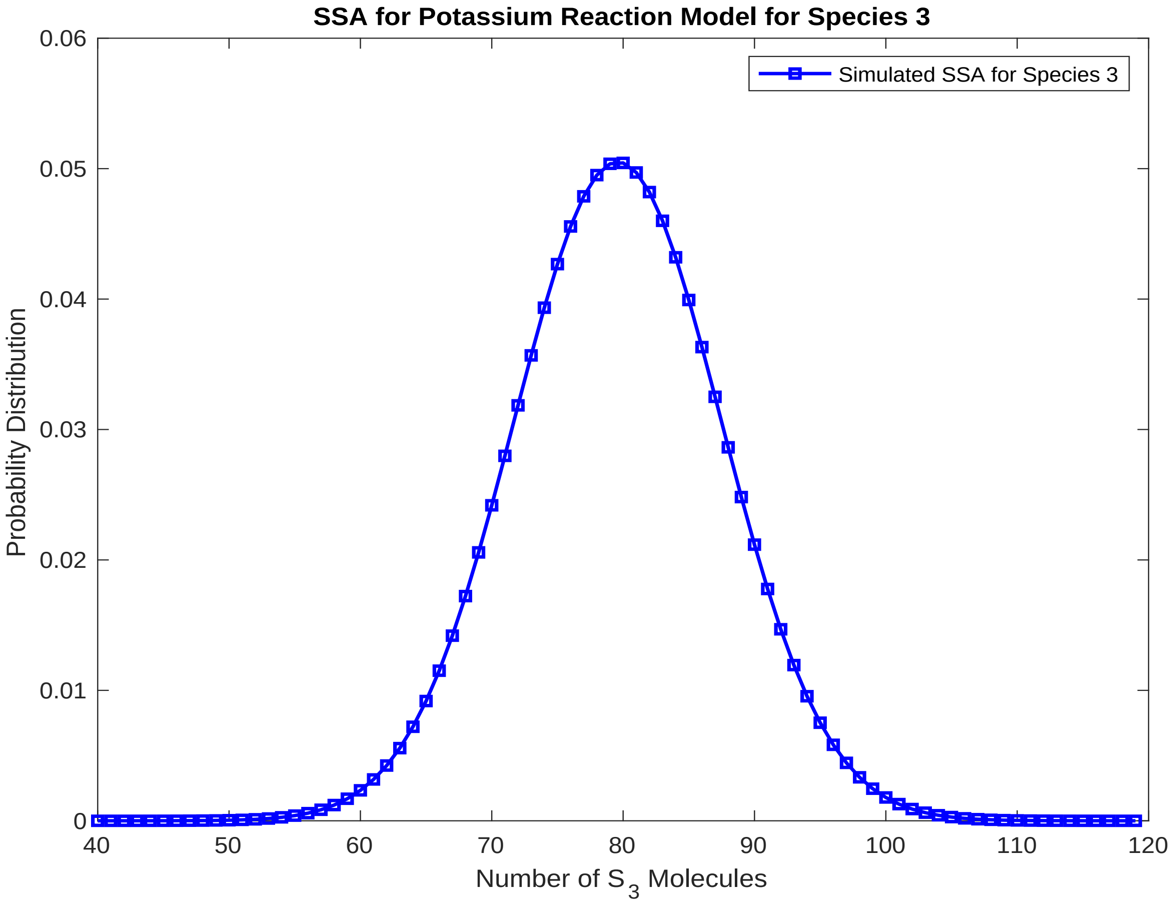

Figure 4.3: Potassium Reaction Model; probability distribution of species S_3 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 3 is 412 seconds.

,

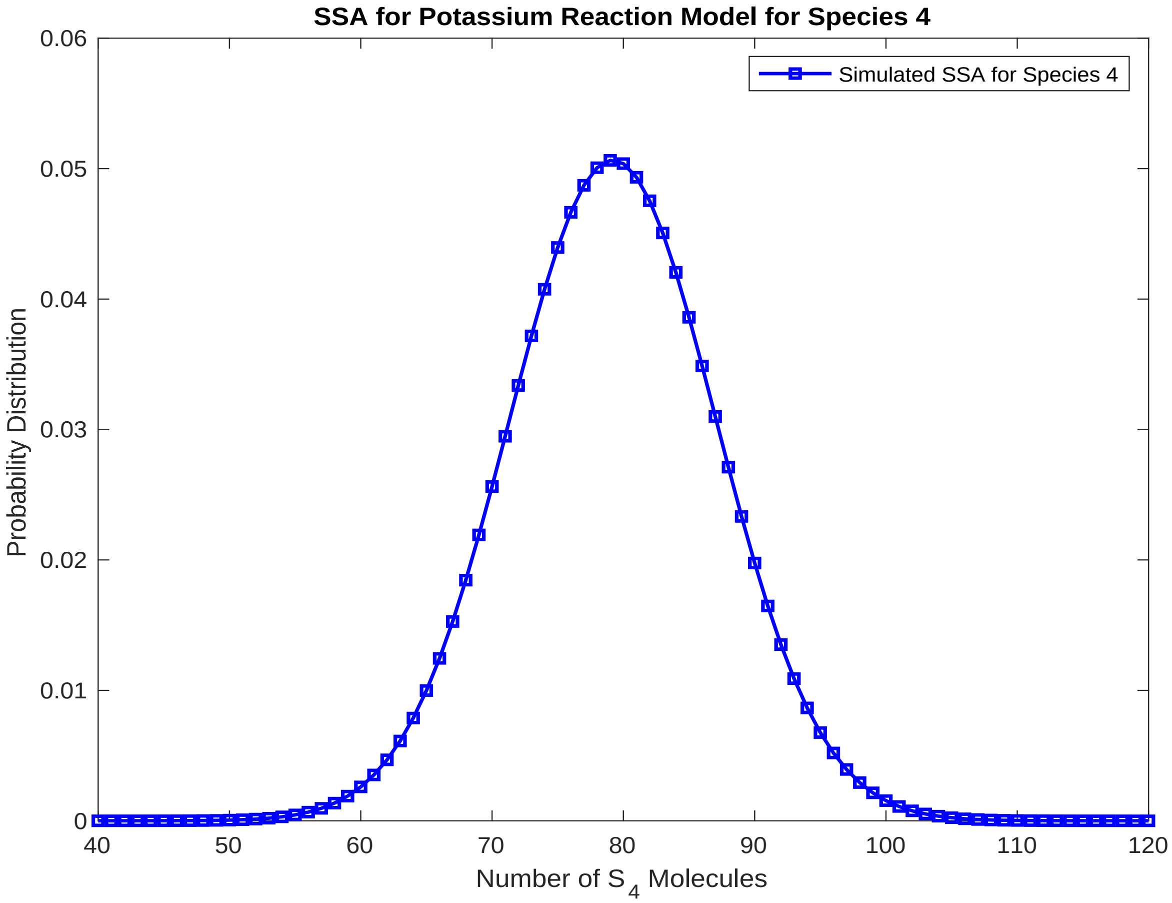

Figure 4.4: Potassium Reaction Model; probability distribution of species S_4 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 4 is 412 seconds.

,

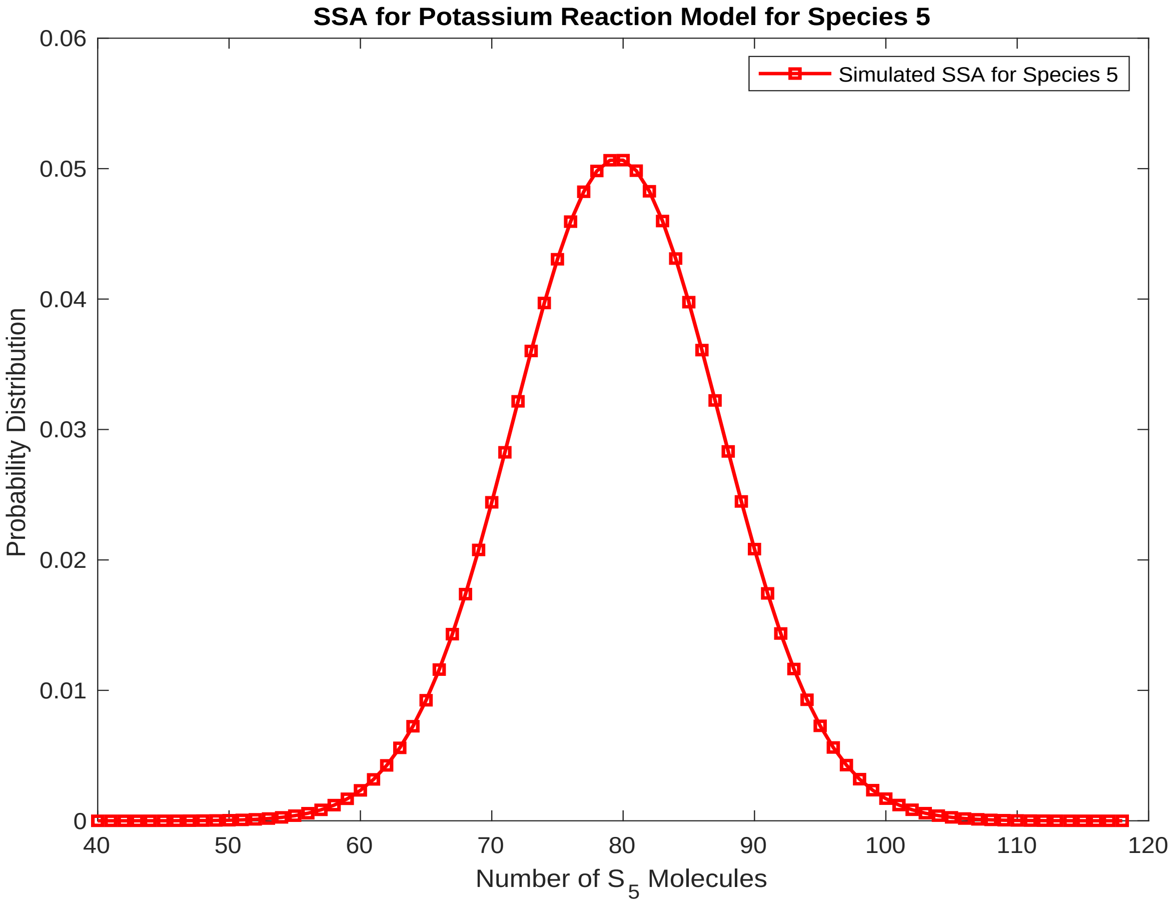

Figure 4.5: Potassium Reaction Model; probability distribution of species S_5 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 5 is 412 seconds.

,

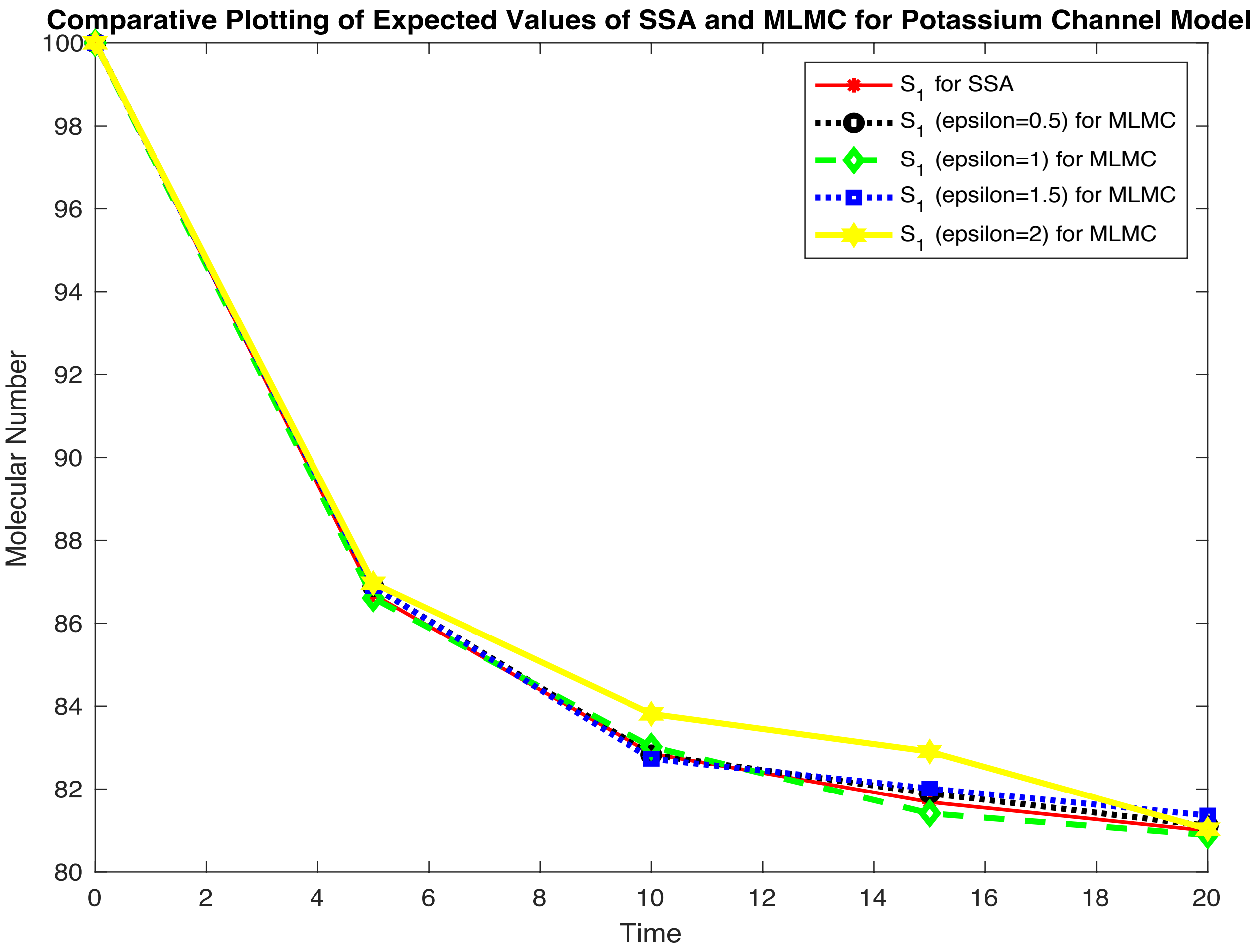

Figure 4.6: Potassium Channel Model: Means of number of molecules of species S_1, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for ϵ=1.5 and below. At the final time, i.e, t_{final} = 20, we see that there is no difference of molecules between the results of the SSA and the MLMC with ϵ=2 and resulting in a relative error of 0.1%.

,

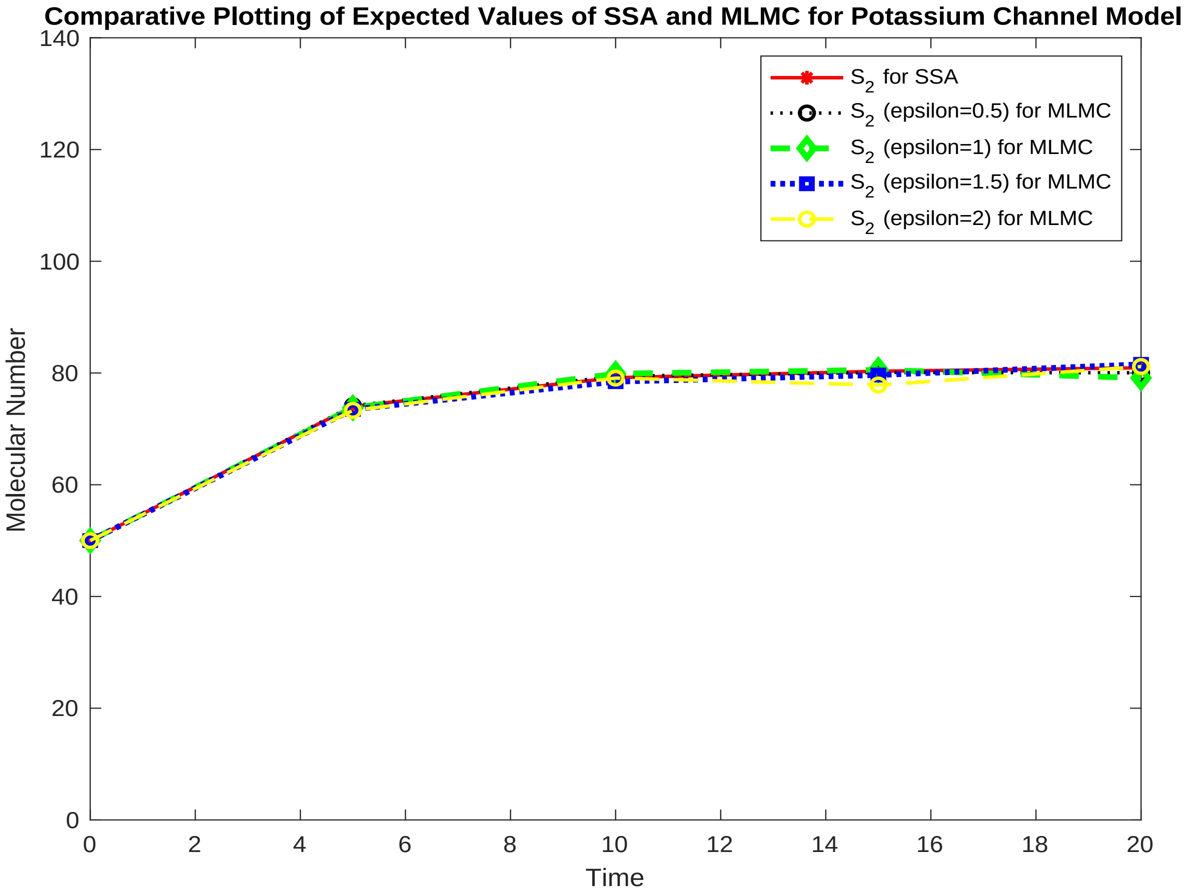

Figure 4.7: Potassium Channel Model: Means of number of molecules of species S_2, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, t_{final} = 20, we see that there is a difference of 1 molecule between the results of the SSA and the MLMC with ϵ=2 and resulting in relative error of approximately 1.1%.

,

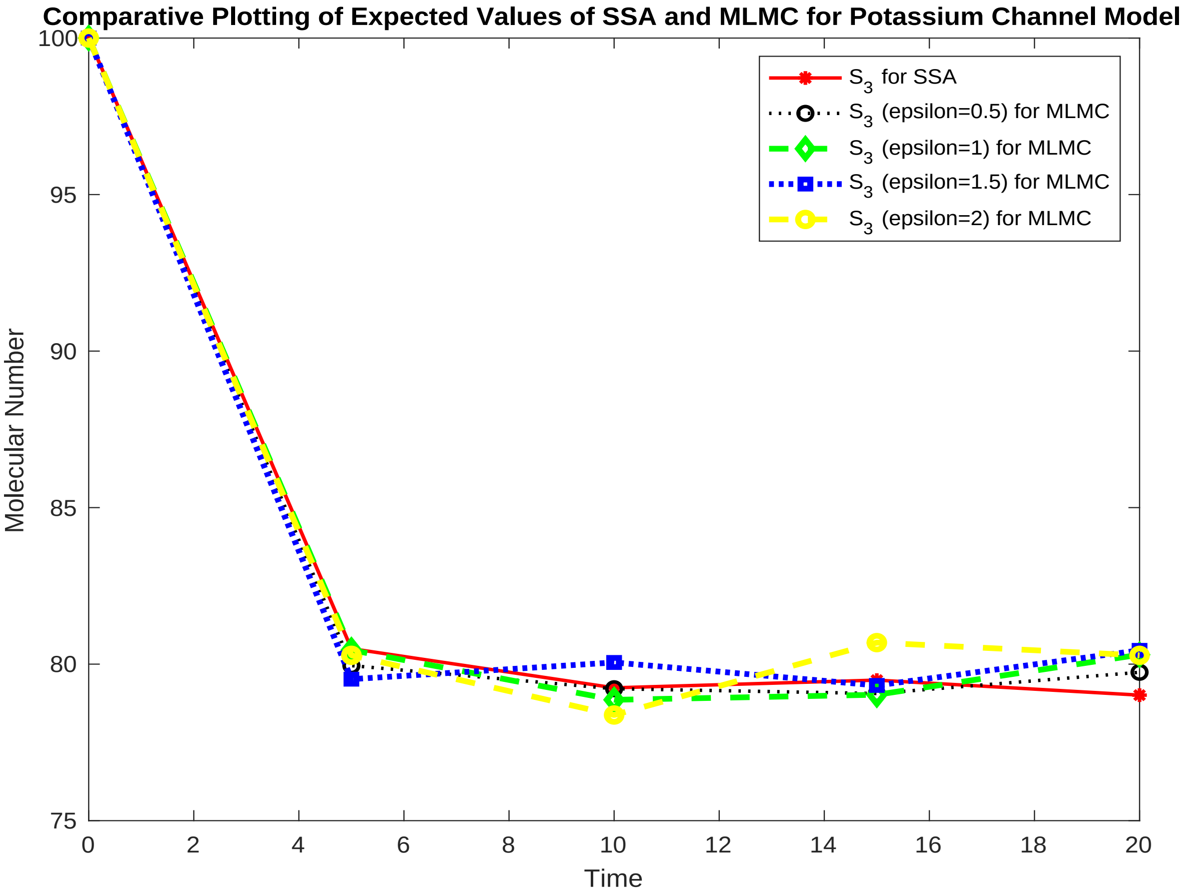

Figure 4.8: Potassium Channel Model: Means of number of molecules of species S_3, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of approximately 1 molecule between the results of the SSA and the MLMC with all ϵ values and resulting in relative error of approximately 0.6%.

,

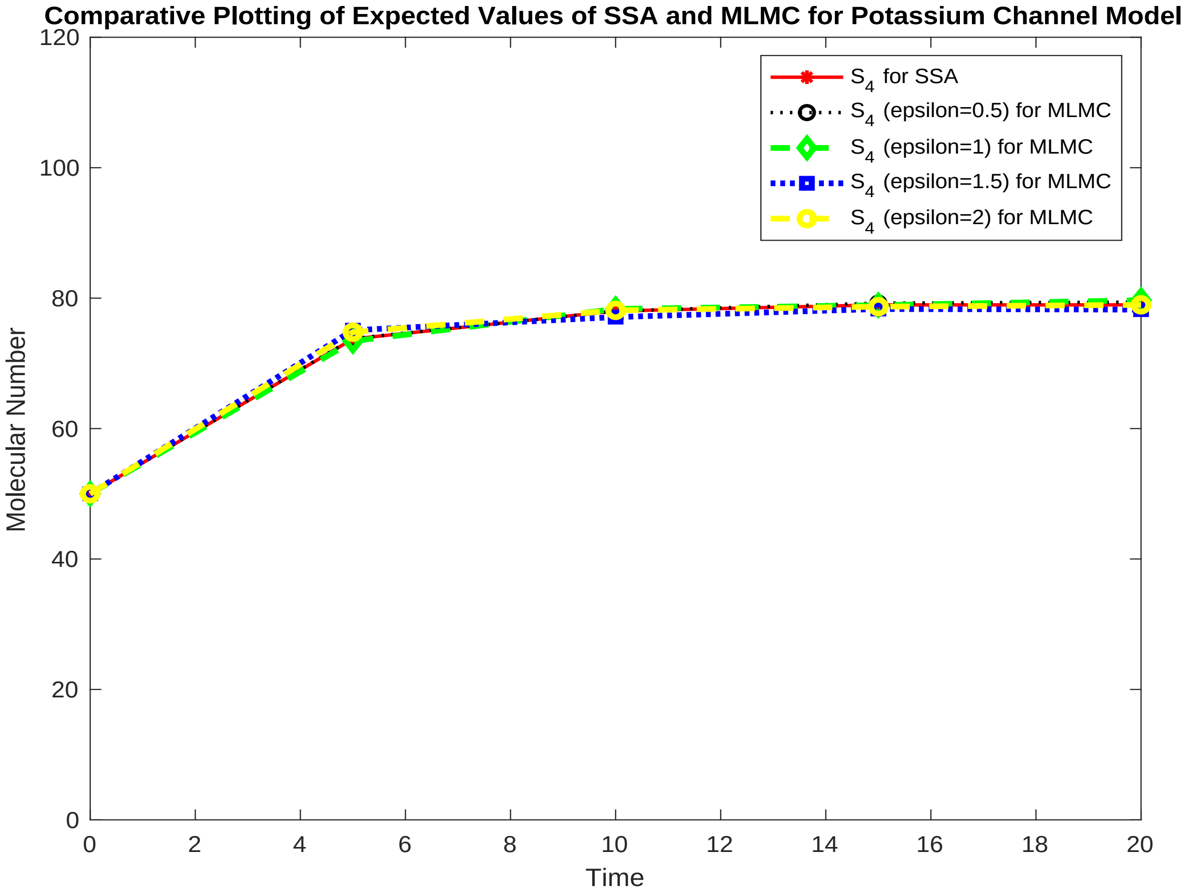

Figure 4.9: Potassium Channel Model: Means of number of molecules of species S_4, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of at most 1 molecule between the results of the SSA and the MLMC with ϵ=1.5 and resulting in relative error of approximately 1.4%.

,

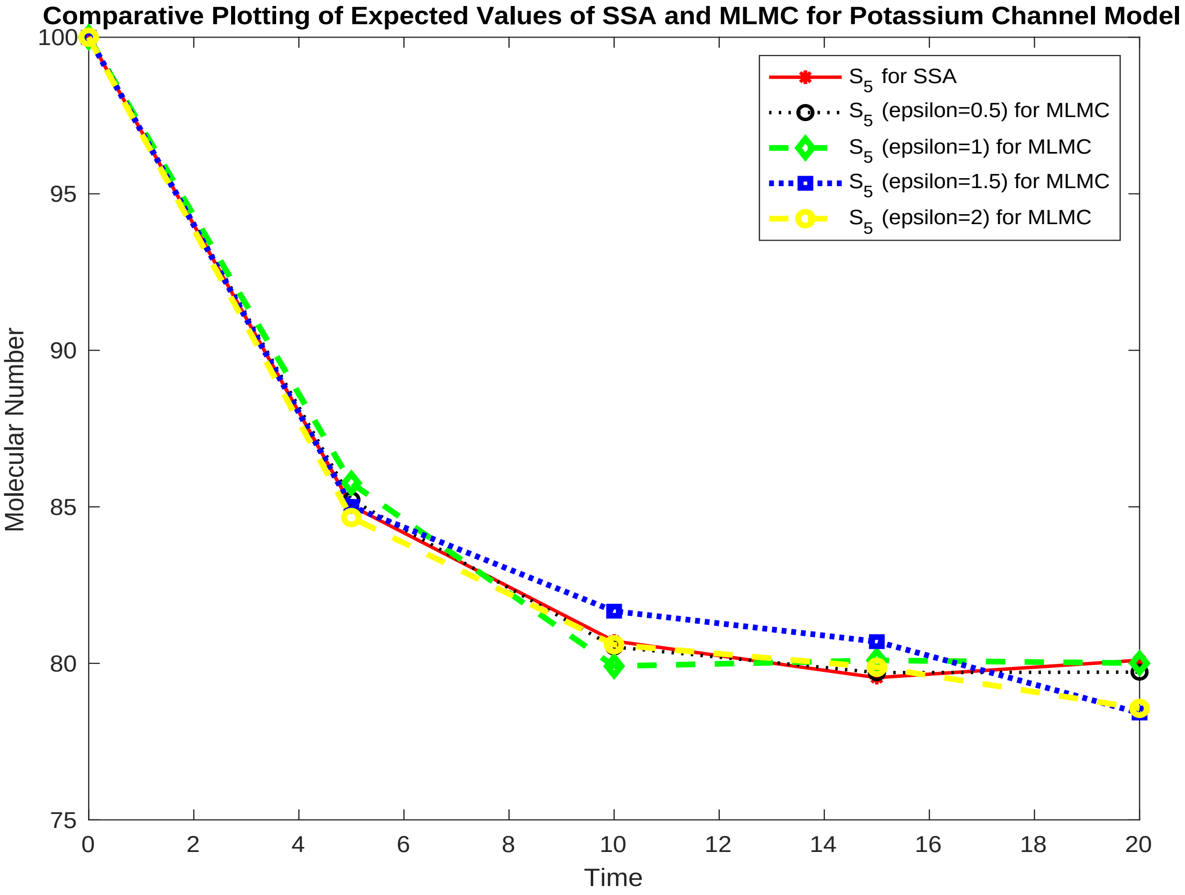

Figure 4.10: Potassium Channel Model: Means of number of molecules of species S_5, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of at most 1 molecule between the results of the SSA and the MLMC with ϵ=1.5 and resulting in relative error of approximately 1.4%.

,

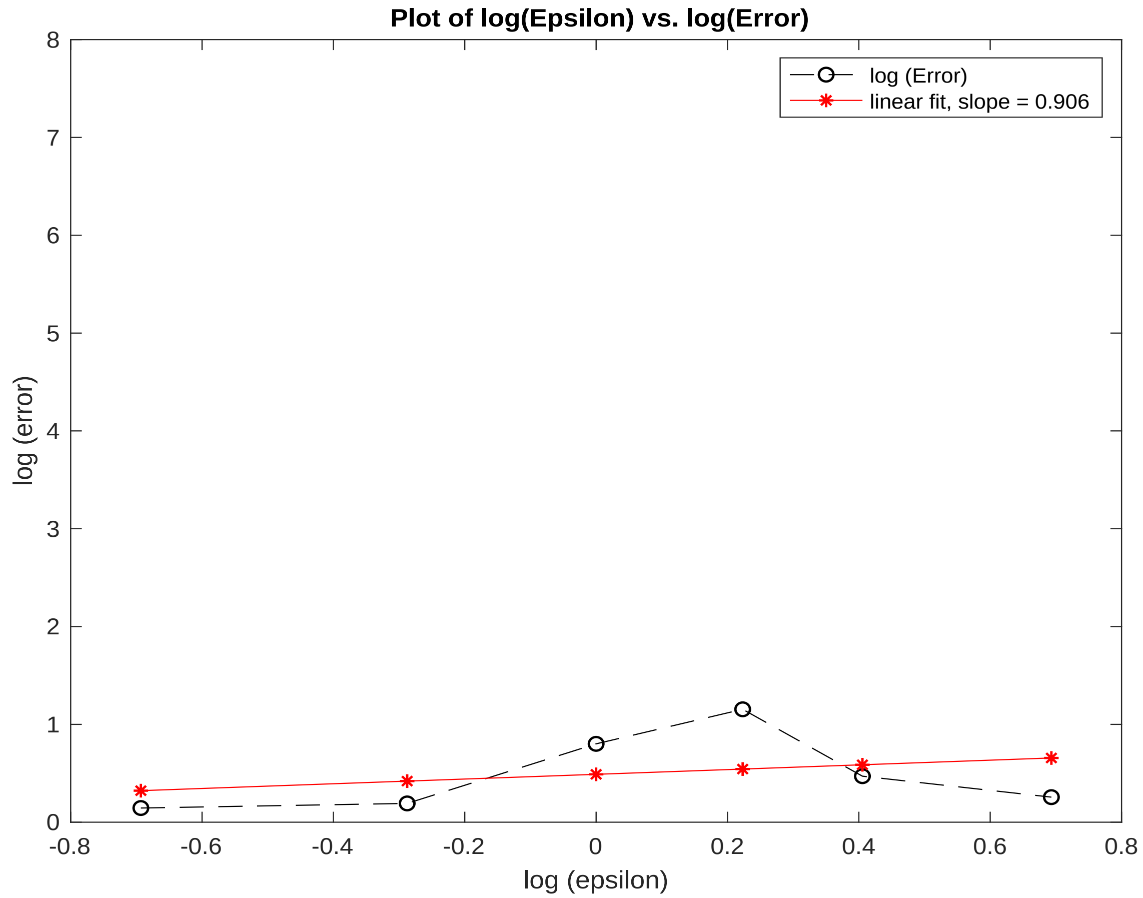

Figure 4.11: Potassium Channel Model: the loglog plot of the absolute error of the MLMC with various values of ϵ compared to the SSA, as a function of the tolerance ϵ. The values of tolerance ϵ considered are ϵ_1=0.5,ϵ_2=0.75, ϵ_3=1, ϵ_4=1.25, ϵ_5=1.5 and ϵ_6=2

,

	SSA	MLMC(ϵ_1=0.5)	MLMC(ϵ_2=1)	MLMC(ϵ_3=1.5)	MLMC(ϵ_4=2)
X1	72.5692	73.79	73.64	72.6984	72.25
X2	46.8809	46.09	46.622	45.4049	44.9583
X3	73.1191	73.9	73.3780	74.5951	75.0417
CPU_{time}	94.4 seconds	13 seconds	4.3 seconds	3.1 seconds	2.3 seconds

Table 4.3: Comparison of expected values obtained through SSA and MLMC (using different ϵ-values) for the Michaelis Menten Model and the computational times of these algorithms

4.2 Michaelis Menten System (Model 2)

The second model used for testing is the Michaelis-Menten system [19]. The focus species are subject to three reactions:

$$S_1 + S_2 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_3$$
$$S_3 \xrightarrow{c_3} S_2 + S_4.$$

The propensity functions corresponding to these reactions are:

$$\begin{aligned} a_1(x) &= c_1 x_1 x_2; \\ a_2(x) &= c_2 x_3; \\ a_3(x) &= c_3 x_3; \end{aligned}$$

while the stoichiometric matrix is:

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

The reaction rate parameters and the initial conditions are listed below:
n_A (Avogadro's number) = 6.023e23; Vol = 1e-15 liters;
X(1) = round(5e-7); X(2) = 2e-7; X(3) = 0; X(4)=0;
c(1) = $\frac{1e6}{n_A vol.}$; c(2) = 1e-4; c(3) = 1e-1;
The simulation is performed on the time interval [0,20]. We simulated 10,000 trajectories using the SSA and the MLMC method with tolerances ϵ = 0.5, 1, 1.5 and 2.

Table 4.3 shows the estimates of the expected values of the molecular amounts of various species obtained from simulating Michaelis Menten model using both the SSA and the MLMC methods with tolerance values ϵ_1=0.5, ϵ_2=1, ϵ_3=1.5 and ϵ_4=2. The values X_i correspond to estimated expected values of the population number for species S_i. We see that approximations of the expected values obtained from the MLMC methods with the tolerances tried are very close to those obtained with the SSA (an exact algorithm), with a difference of approximately 1 or 2 at most number molecules for some species. For example, consider species S_1, the SSA simulation gives an estimation of the $E(X(20))$ of 72.5692 whereas MLMC (ϵ_4=2) simulation gives 72.25. The relative error of MLMC compared to SSA is $\frac{|72.5692-72.25|}{|72.5692|} * 100\% \approx 0.4\%$. This shows a very good accuracy of our simulation. We also see from Table 4.3 that the speed up of MLMC method (ϵ_1=0.5) is approximately 7.3 (94.4/13) times faster than SSA. The speed up of MLMC method (ϵ_2=1) is approximately 22 times faster than SSA. Similarly, the speed up of MLMC method (ϵ_3=1.5) is approximately 30.4 (94.4/3.1) times faster than SSA and finally the speed up of MLMC method (ϵ_3=2) is approximately 41 (94.4/2.3). In Figures 4.12 - 4.14, we present the probability distribution of the molecular amounts of the species S_1, S_2 and S_3, respectively, computed using the SSA, at T=20. We compare the SSA and the the MLMC method with the ϵ=0.5, 1 and 1.5 in Figures 4.15 - 4.17 for species S_1, S_2 and S_3 respectively. Each plot shows the estimation of the mean number of the S_i molecules as a function of time, computed the SSA and by the MLMC method with the sequence of tolerances. Figure 4.18 gives a loglog plot of the norm 2 error i.e.,

$$error(T) = \|X_{SSA}(T) - X_{MLMC}(T)\|_2,$$

for the MLMC compared to the SSA for the Michaelis Menten system. Here error is a function of the tolerance ϵ. The slope of the linear fit is 0.2.

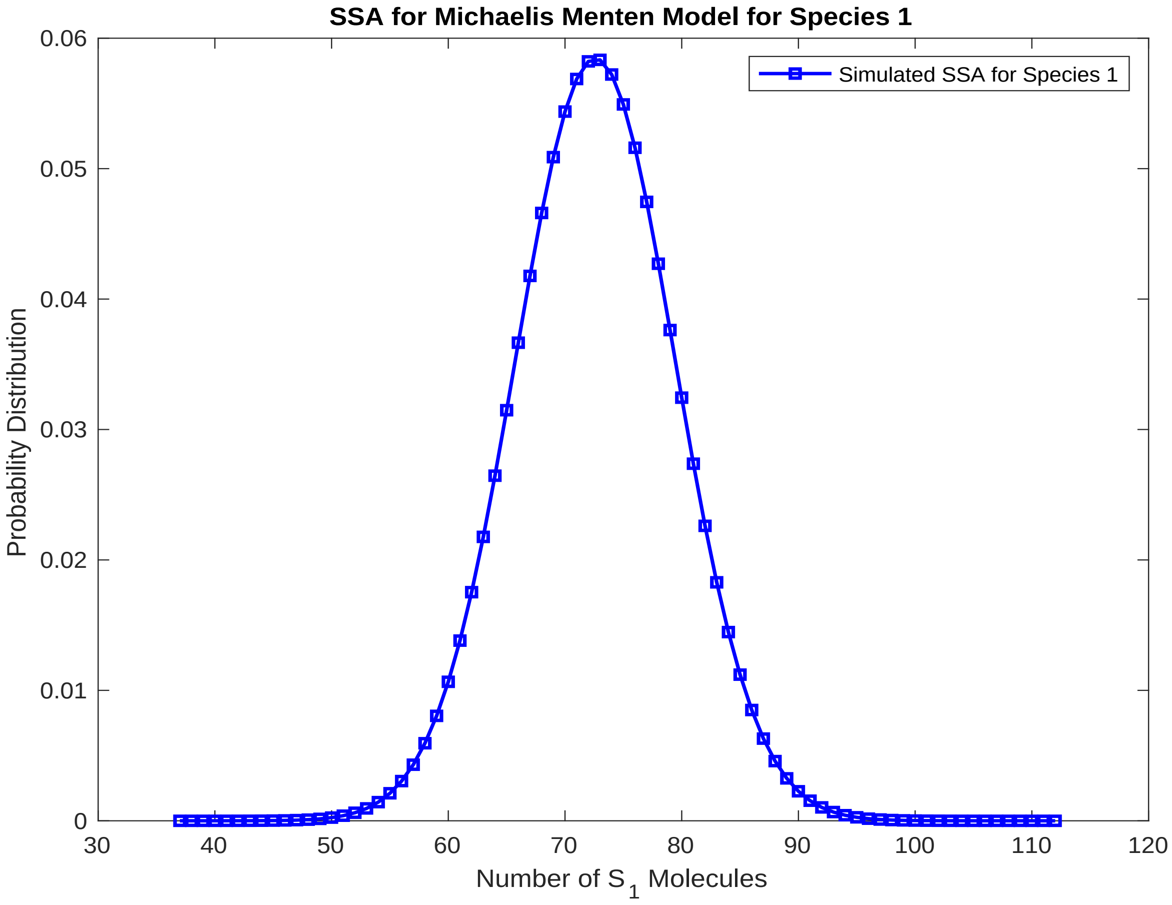

Figure 4.12: Michaelis Menten Model; probability distribution of species S_1 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 1 is 94.4 seconds.

,

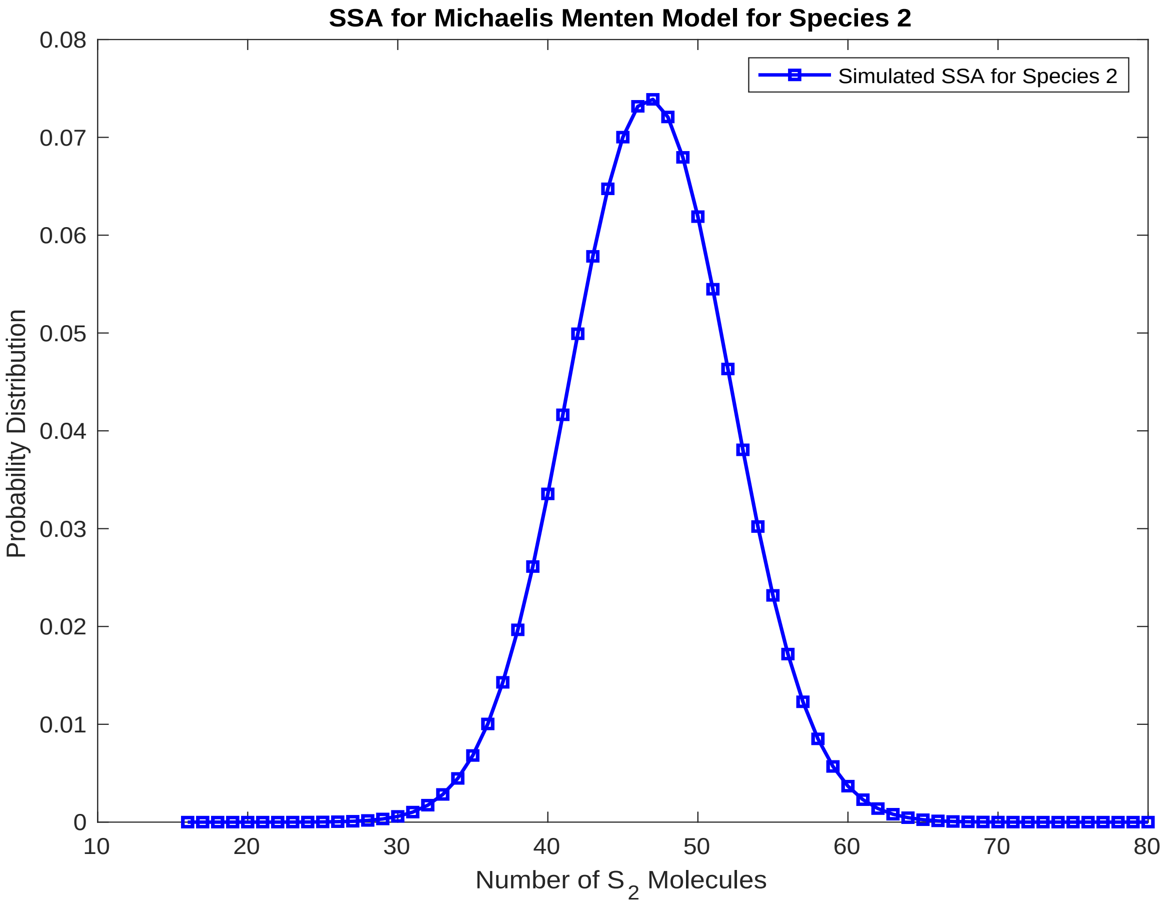

Figure 4.13: Michaelis Menten Model; probability distribution of species S_2 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 2 is 94.4 seconds.

,

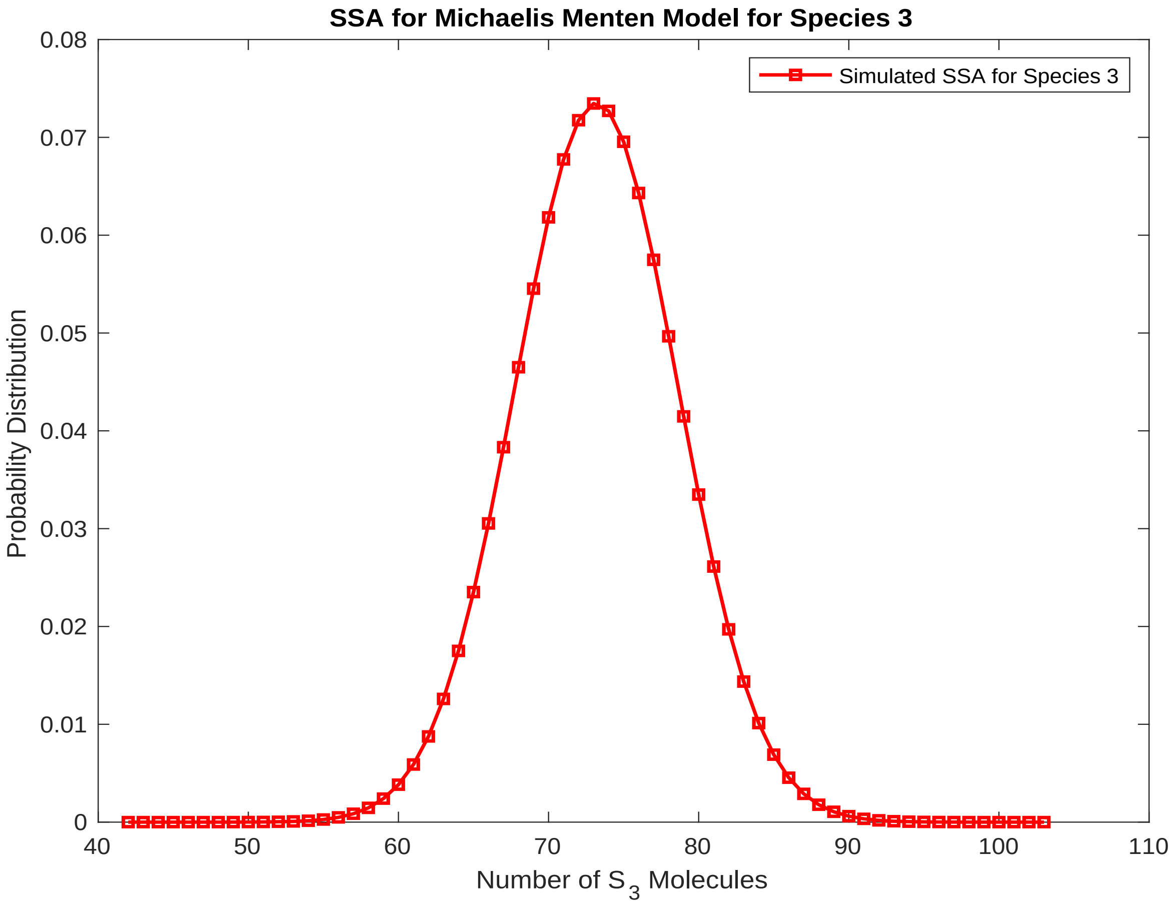

Figure 4.14: Michaelis Menten Model; probability distribution of species S_3 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 3 is 94.4 seconds.

,

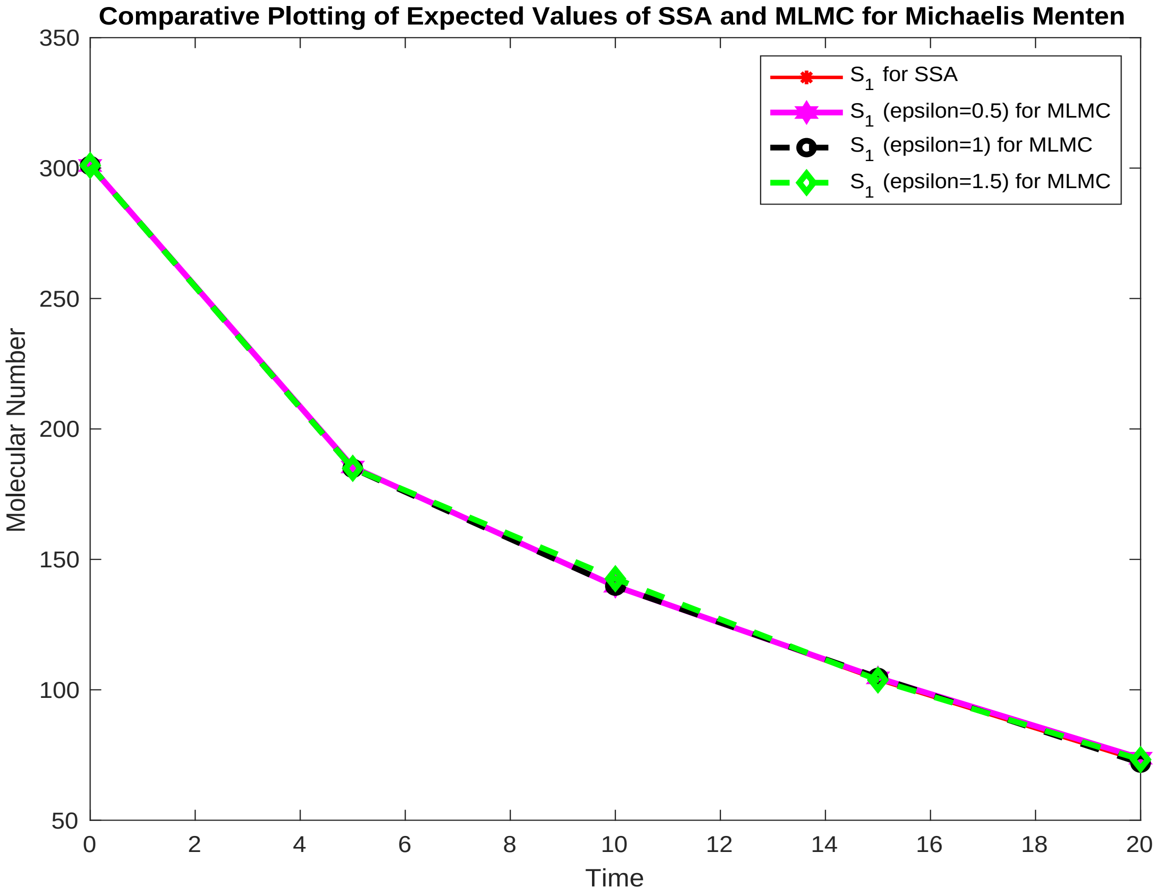

Figure 4.15: Michaelis Menten Model: Means of number of molecules of species S_1, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1 and 1.5. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, t_{final} = 20, we see that there is a difference of at most 1 molecule between the results of the SSA and the MLMC with ϵ=1 and resulting in relative error of approximately 1.3%.

,

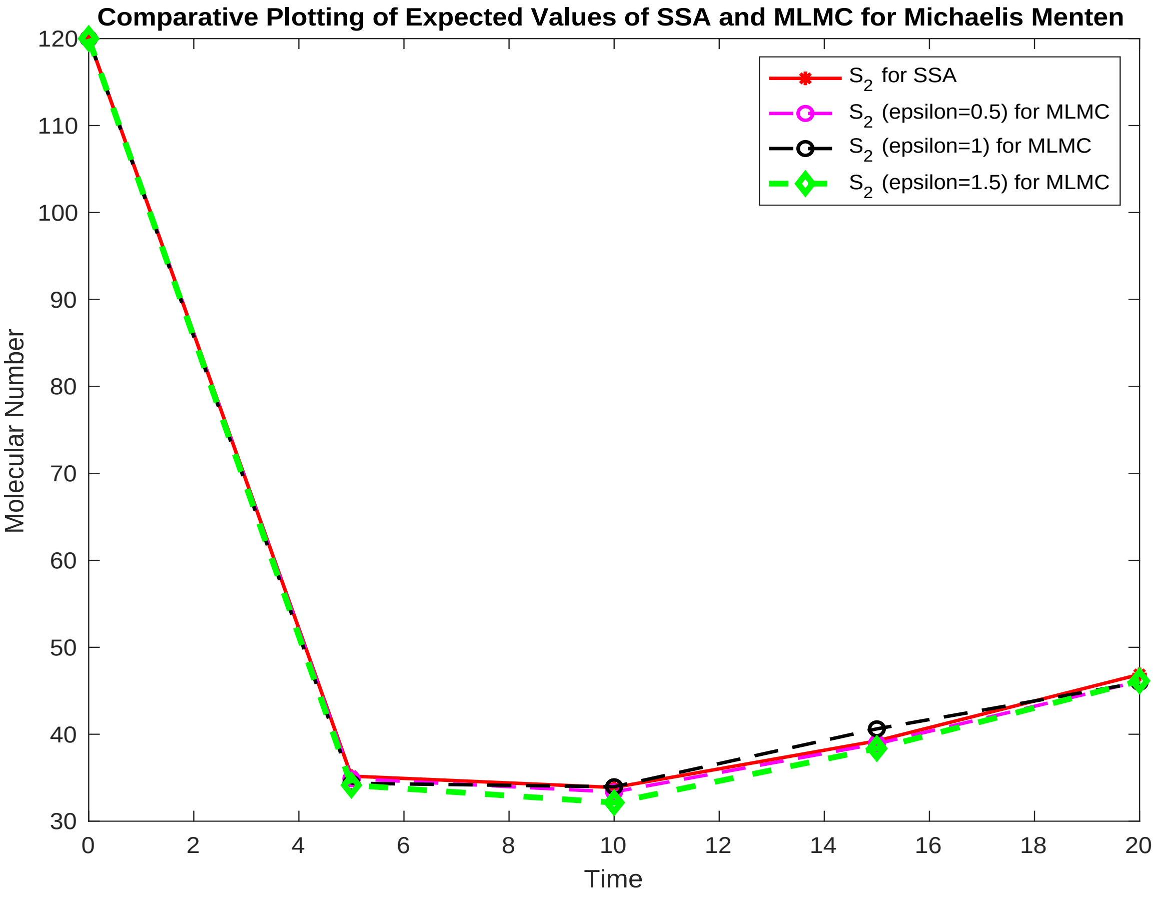

Figure 4.16: Michaelis Menten Model: Means of number of molecules of species S_2, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1 and 1.5. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of at most 1 molecule between the results of the SSA and the MLMC with ϵ=1.5 and resulting in relative error of approximately 3.1%.

,

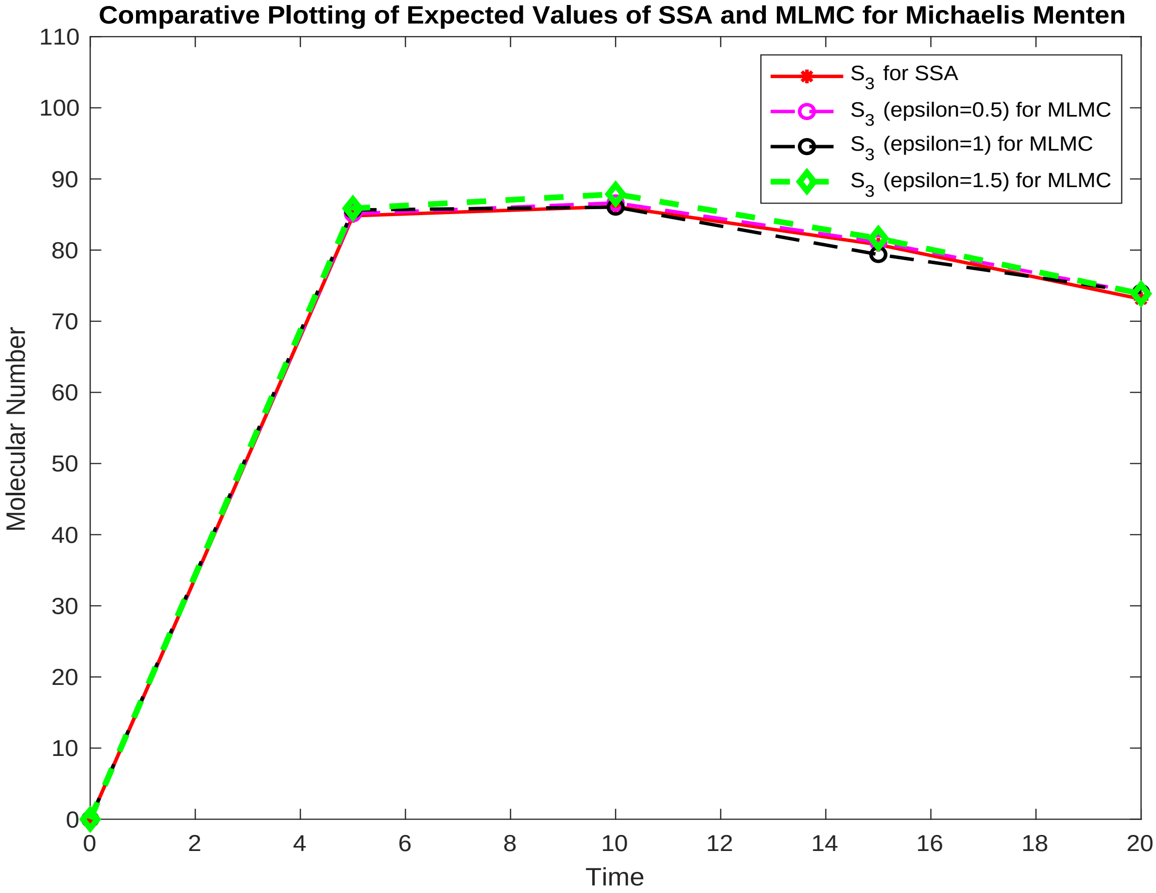

Figure 4.17: Michaelis Menten Model: Means of number of molecules of species S_3, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1 and 1.5. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, t_{final} = 20, we see that there is a difference of at most 2 molecule between the results of the SSA and the MLMC with ϵ=1 and resulting in relative error of approximately 0.3%.

,

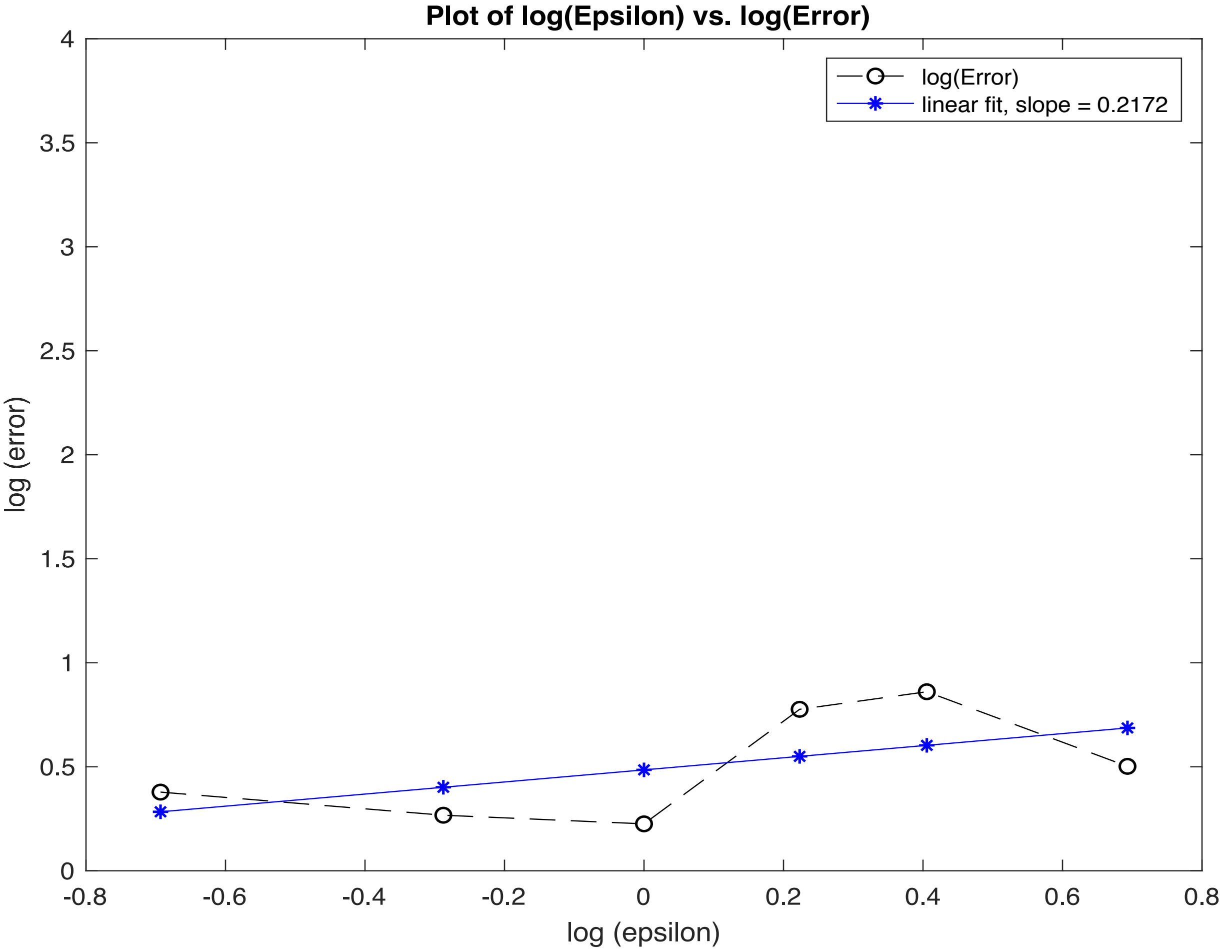

Figure 4.18: Michaelis Menten Model: the loglog plot of the absolute error of the MLMC with various values of ϵ compared to the SSA, as a function of the tolerance ϵ. The values of tolerance ϵ_1=0.5,ϵ_2=0.75, ϵ_3=1, ϵ_4=1.25, ϵ_4=1.5 and ϵ_4=2

.

,

4.3 Cyclical Reaction System (Model 3)

Finally, we study the Cyclical Reaction System from [28]. This system consists of three reactions involving three reacting species. The reaction channels are:

$$S_1 \xrightarrow{c_1} S_2$$
$$S_2 \xrightarrow{c_2} S_3$$
$$S_3 \xrightarrow{c_3} S_1$$

and have propensities:

$$a_1(x) = c_1 x_1;$$
$$a_2(x) = c_2 x_2;$$
$$a_3(x) = c_3 x_3.$$

This reaction system has the following stochiometric matrix:

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

1. We simulate the Cyclical Reaction system with the following initial conditions and reaction rate parameters X(1) = 100; X(2) = 80; X(3) = 100, c(1) = 0.1; c(2) = 0.1; c(3) = 0.1, on the time interval [0,20].

2. As with the previous model, we simulate the SSA using 10,000 trajectories and the MLMC strategy with a sequence of tolerances, which in this case is ϵ=0.5,1, 1.5 and 2. The probability distributions of the molecular amounts computed over 10,000 trajectories of the SSA, at T=20,included in figures 4.20-4.22.

3. The estimates of the average number of molecules of each species S_i as functions of time, computed using the SSA and the MLMC with different tolerance levels, are shown in figures 4.23-4.25.

Table 4.4 presents the average number of molecules of each species at final time T=20, computed over 10,000 trajectories of the stochastic simulation algorithm and by the

	SSA	MLMC(ϵ_1=0.5)	MLMC(ϵ_2=1)	MLMC(ϵ_3=1.5)	MLMC(ϵ_4=2)
X1	93.8182	90.3133	92.5638	93.9167	94.5832
X2	93.4114	94.4423	94.0787	93.5877	96.3388
X3	92.7707	95.2444	93.3845	92.4956	89.0780
CPU_{time}	69 seconds	19 seconds	4 seconds	2.8 seconds	2 seconds

Table 4.4: Comparison of expected values obtained through SSA and MLMC (using different ϵ-values) for the Cyclical Reaction Model and the computational times of these algorithms

MLMC method with the following sequence of tolerances ϵ_1=0.5, ϵ_2=1, ϵ_3=1.5 and ϵ_4=2. Also, Table 4.4 gives the computational time required by each method. We see that the the expected values obtained from all methods are very close to each other with a difference of approximately 1, 2 or 3 at most number molecules for some species. For instance, for species S_2, the SSA simulation estimates an average of 93.4112 whereas MLMC with $\epsilon = 0.5$ simulation predicts an average of 94.4423 S_2 molecules. The relative error of the MLMC compared to SSA is $\frac{|93.4114-94.4423|}{|93.4114|} * 100\% \approx 1.1\%$. This shows a very good accuracy of our simulation. We also see from Table 4.4 above, the computing time of the MLMC scheme is approximately 35 (69/2) times faster than SSA. Figures 4.19-4.21 show the expected values for the molecular amounts of the species $X_1, X_2, ..., X_5$ respectively as functions of times estimated from 10,000 trajectories utilizing the *SSA* and the *MLMC* strategy with a given tolerance. Figure 4.25 gives a loglog plot of the norm-2 of the total error of the MLMC method compared to the SSA, as a function of the tolerance ϵ. The slope in the loglog plot is almost 2.

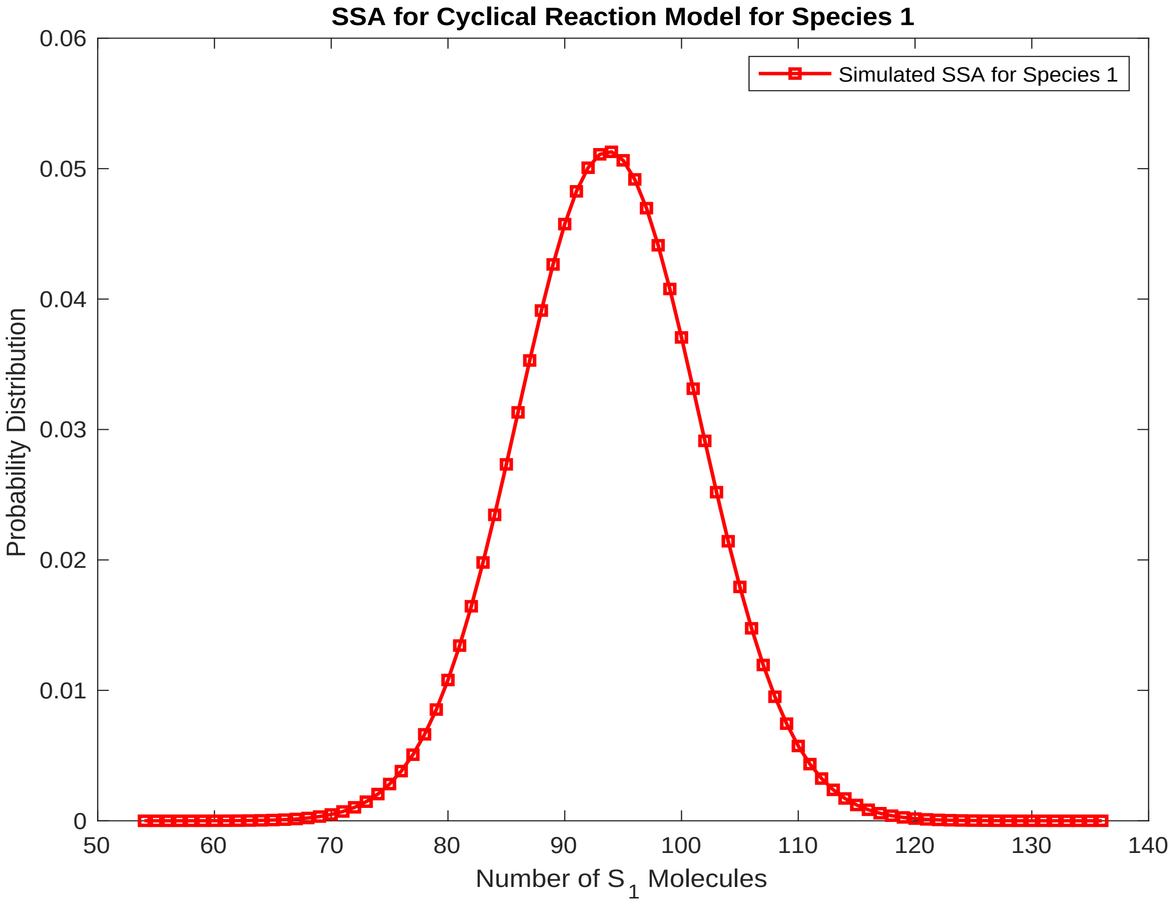

Figure 4.19: Cyclical Reaction Model; probability distribution of species S_1 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 1 is 69 seconds.

,

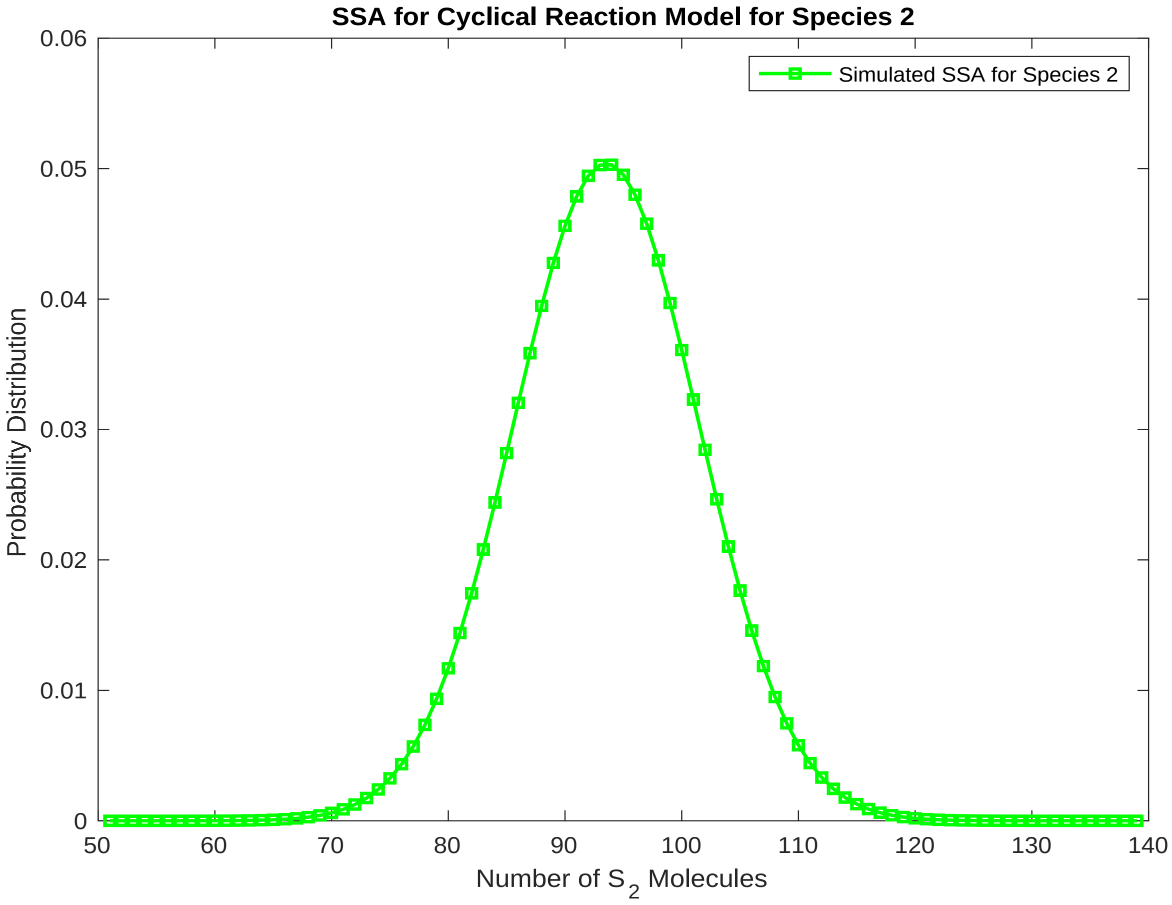

Figure 4.20: Cyclical Reaction Model; probability distribution of species S_2 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 2 is 69 seconds.

,

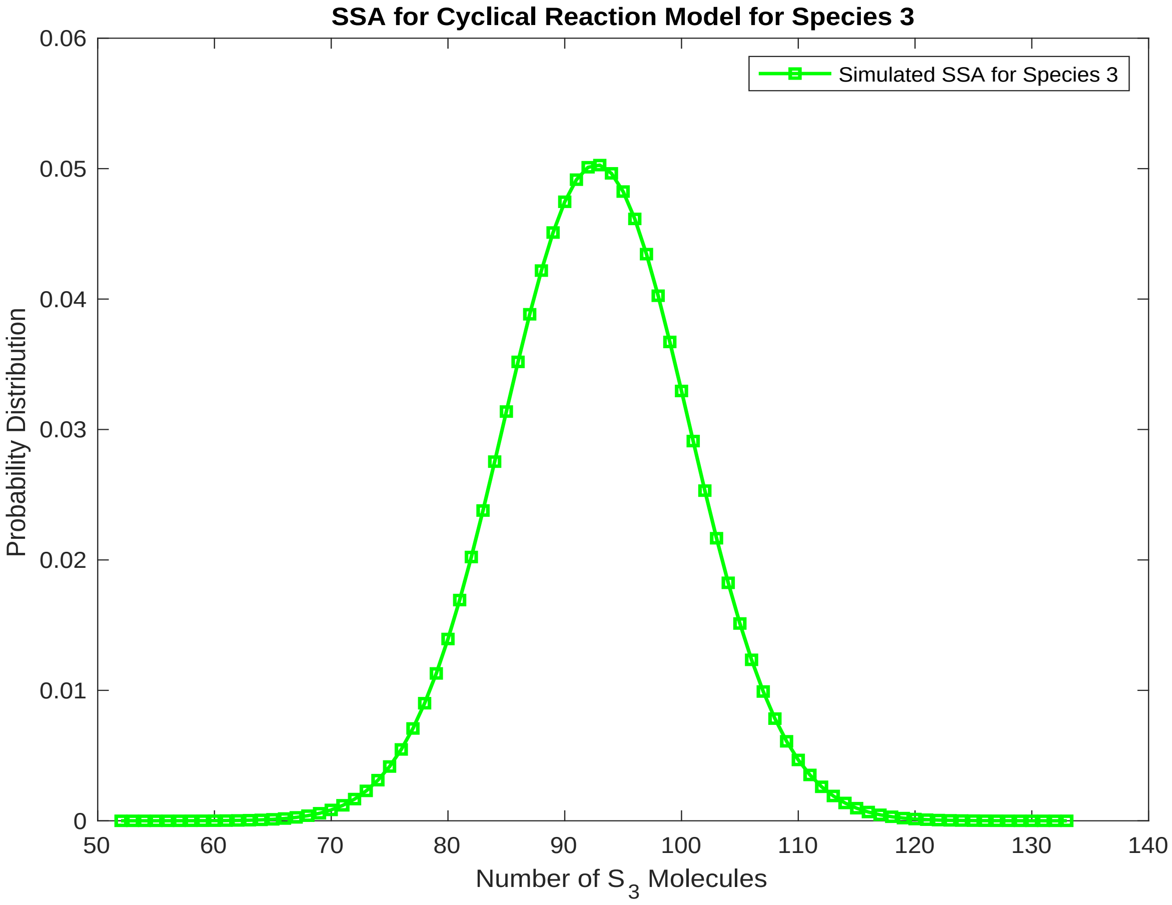

Figure 4.21: Cyclical Reaction Model; probability distribution of species S_3 at time T=20, using 10,000 trajectories of the SSA. The total time taken to simulate for Species 3 is 69 seconds.

,

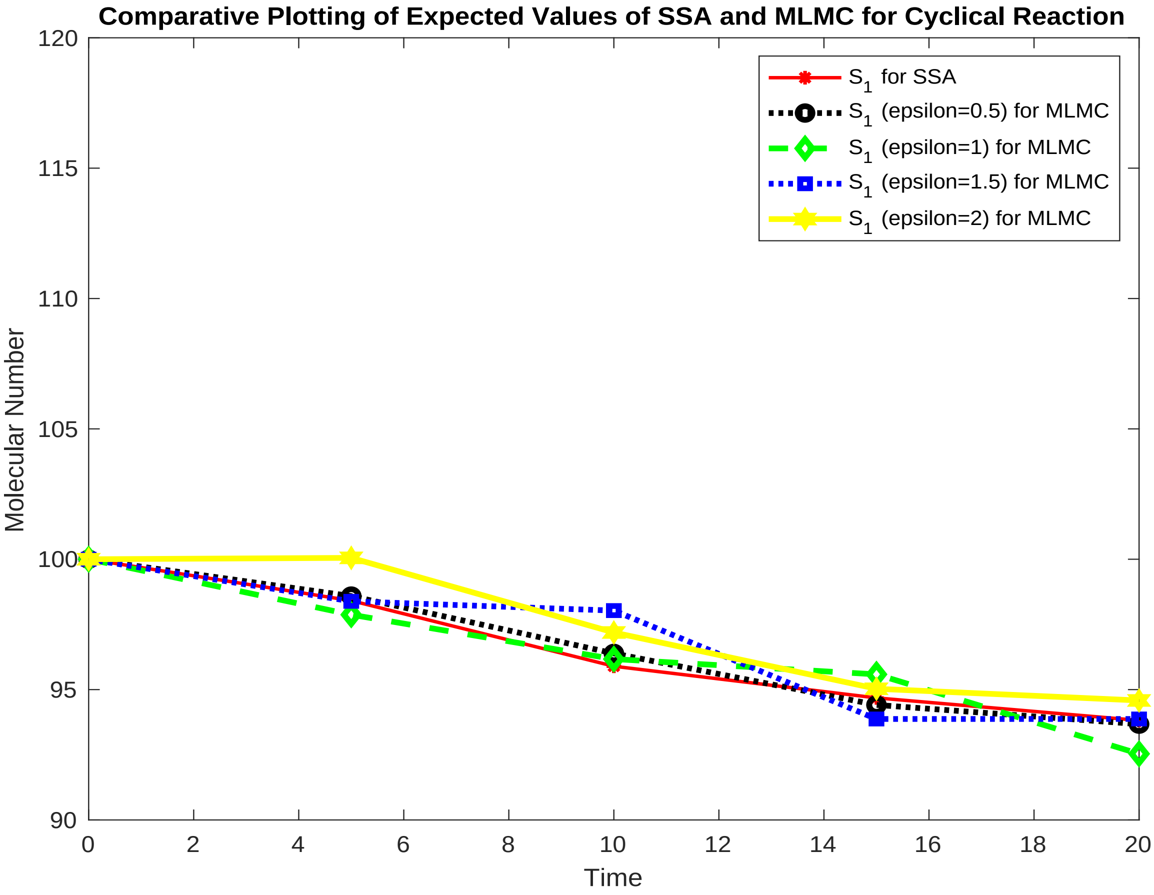

Figure 4.22: Cyclical Reaction Model: Means of number of molecules of species S_1, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, t_{final} = 20, we see that there is a difference of at most 2 molecule between the results of the SSA and the MLMC with ϵ=2 and resulting in relative error of approximately 0.7%.

,

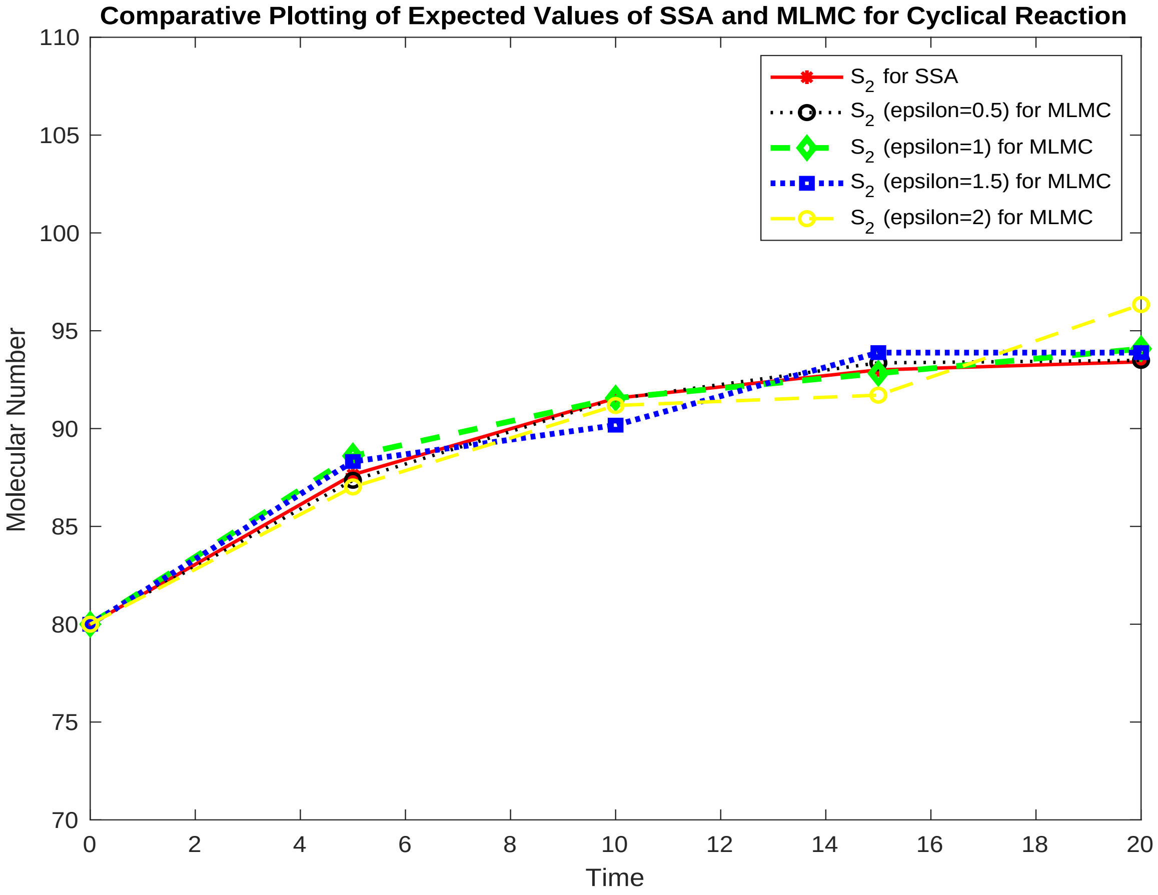

Figure 4.23: Cyclical Reaction Model: Means of number of molecules of species S_2, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of at most 1 molecule between the results of the SSA and the MLMC with ϵ=0.5 and resulting in relative error of approximately 1.1%.

,

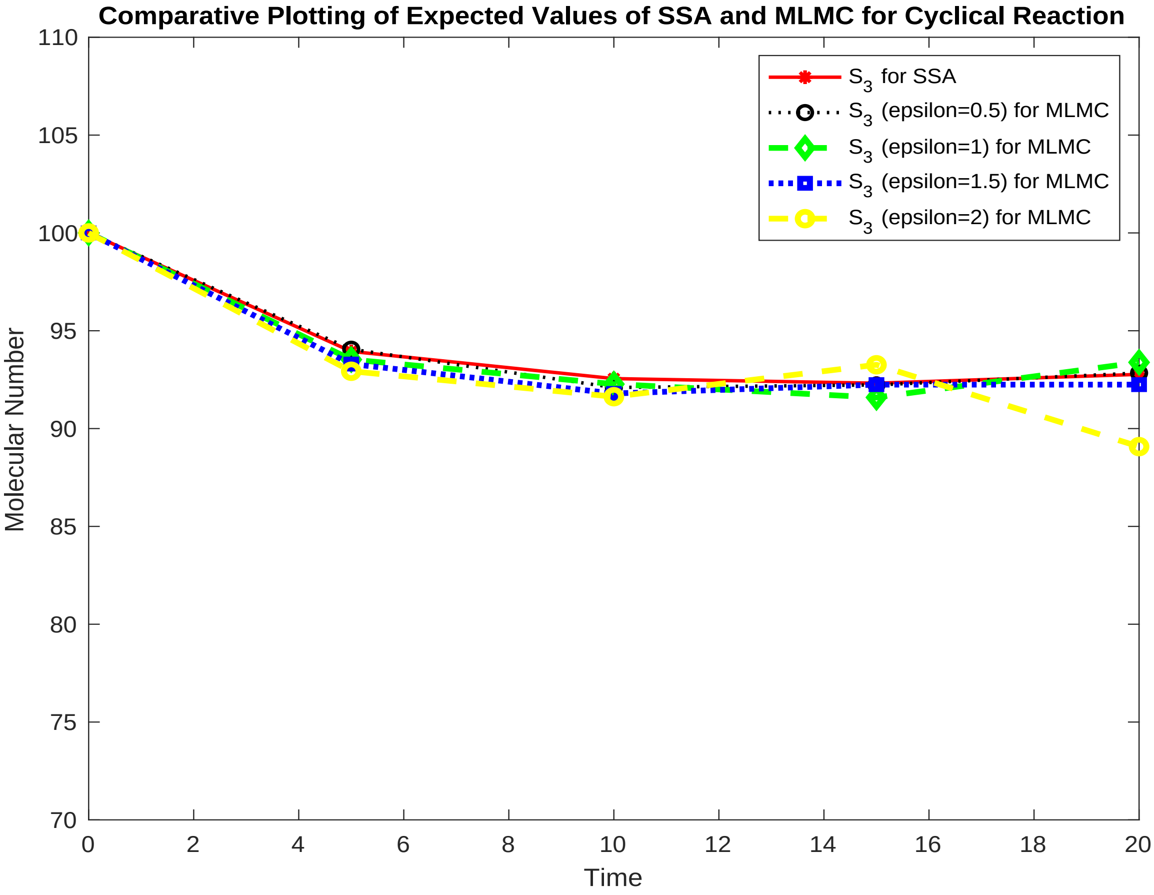

Figure 4.24: Cyclical Reaction Model: Means of number of molecules of species S_3, as a function of time, estimated using 10,000 trajectories of the SSA and the MLMC method with values of the tolerance ϵ being 0.5, 1, 1.5 and 2. The time of integration used in T = 20. The accuracy of the MLMC method is excellent for all ϵ values. At the final time, i.e, $t_{final} = 20$, we see that there is a difference of at most 2 molecule between the results of the SSA and the MLMC with $\epsilon = 0.5$ and resulting in relative error of approximately 2.6%.

,

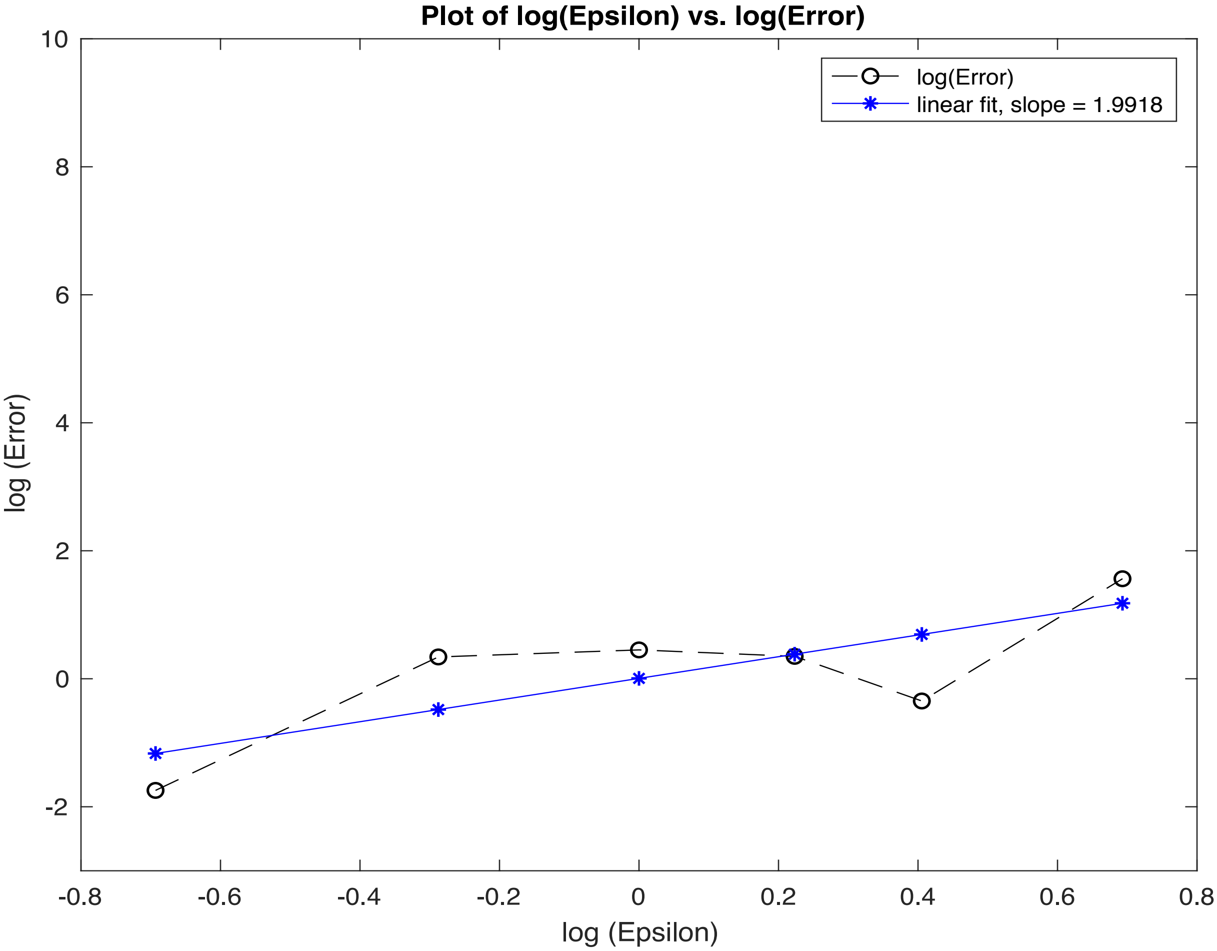

Figure 4.25: Cyclical Reaction Model: the loglog plot of the absolute error of the MLMC with various values of ϵ compared to the SSA, as a function of the tolerance ϵ. The values of tolerance ϵ_1=0.5,ϵ_2=0.75, ϵ_3=1, ϵ_4=1.25, ϵ_4=1.5 and ϵ_4=2

.

,

Chapter 5

Conclusion

This thesis discusses numerical strategies for simulating stochastic discrete models of well-stirred biochemical kinetics, namely the Chemical Master Equation. The focus is on efficient and accurate methods for approximating the expected value of a function of interest (f) of the stochastic process modeling the evolution of the biochemical network, $X(t)$. The recently developed multi-level Monte Carlo (MLMC) of Giles is more efficient that the existing stochastic simulation algorithm of Gillespie, for evaluating the expected values $E(f(X(t)))$.

Although, the *SSA* is an exact algorithm, it often requires a high computational cost, especially when some reactions in the system are fast therefore taking smaller time-steps because the step size in SSA in inversely proportional to the sum of propensity functions. The multi-level Monte Carlo method provides a more cost-effective way to estimate $E(f(X_i(t)))$ than the Stochastic Simulation Algorithm, by choosing a sequence of levels from coarser grids to finer grids. Trajectories on coarser grids are cheaper to simulate but inaccurate, while trajectories on finer grid are expensive to simulate, but accurate. In the *MLMC* tau-leaping method, we observed that we can pick a fix step size and calculate the expected value of our functional using different levels with the first level being the base estimator and subsequent levels being the bias correction estimators. These correction estimators are added to the base estimator to reduce bias resulting in higher order of accuracy. The accuracy of the estimator obtained from this addition is that of the most accurate levels added.

The MLMC method reduces the variance of the estimator by coupling the sets of trajectories simulated at each level. If the variance of the estimator is reduced then fewer trajectories are needed to obtain a similar accuracy of the estimator, thus reducing its computational cost. Our approximation may be as close as possible to the exact algorithm by choosing more levels but higher number of levels also comes at a higher computational cost. So, there is a trade-off between the efficiency of the estimator and the achieved accuracy. We investigated improvements for the number of trajectories on each level taken by the MLMC strategy for a certain accuracy. This improved MLMC strategy has accuracy comparable to that of the SSA, but at a small fraction of the SSA's computational time. In addition, we investigated the effect of choosing a certain tolerance, on the accuracy achieved by the MLMC technique compared to the accuracy of the exact SSA.

We investigated the behaviour of the MLMC strategies compared to the SSA on several models of biochemical networks arising in applications and we discussed the advantages of the MLMC approach.

Future Work

In our future work, we shall design adaptive time-stepping strategies method for MLMC techniques for stochastic biochemical networks. Such adaptive schemes are particularly important for the cost of moderately stiff to stiff biochemical systems. For stiff problems, a fixed-stepsize scheme significantly increased the computational time of the algorithm, for a given level of accuracy compared to a variable stepsize scheme.

.

Appendix 1

Probability Distributions

- The random variable X has a *uniform distribution* over the range [a,b], $X \sim$ U(a,b), if it has a probability density function of the form

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

- The random variable X has an *exponential distribution* with parameter λ, $X \sim$ *Exp*(λ), if it has a probability density function of the form

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

- The random variable X has a *Poisson distribution* with parameter λ, $X \sim P(\lambda)$, if it has a probability mass function of the form

$$P(X = k) \quad = \quad \tfrac{\lambda^k}{k!} e^{-k}$$

  if $k = 0, 1, 2, ...$

- The random variable X has a *binomial distribution* with parameters n and p, $X \sim$

$B(n,p)$, if it has a probability mass function of the form

$$P(X = k) = \left(\begin{array}{c} n \\ p \end{array} \right) p^k (1-p)^{n-k}$$

if $k = 0, 1, 2, ..., n$

Appendix 2

Poisson Process

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

- The counting process $\{N(t), t \geq 0\}$ is said to be a Poisson process having rate $\lambda > 0$, if:

  1. $N(0) = 0$,

  2. the process has independent increments,

  3. the number of events in any interval of length t has a Poisson distribution with mean λt, i.e., for all $s, t \geq 0$ the following is true

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

     for $n =$ 0,1,...

2.1 Linear combination of independent Poisson random variables

If X_1 and X_2 are independent Poisson random variables such that $X_1 \sim P_1(\lambda_1)$ and $X_2 \sim P_2(\lambda_2)$ then $X_1 + X_2 \sim P(\lambda_1 + \lambda_2)$, that is $X_1 + X_2$ is a Poisson distribution of parameters $\lambda_1 + \lambda_2$.

Appendix 3

Convergence in Probability

Let Ω be a set, then a $\sigma - algebra\ F$ on ω has the following properties:

1. $\phi \in F$
2. F $\in F$
3. If $A_1, A_2, .. \in F$ then $A := \cup_{i=1}^{\infty} A_i \in F$

Here (Ω, F) is a measurable space. A probability measure P is a mapping $P : F \rightarrow$[0,1] with the following properties:

1. $P(\phi) = 0$ and $P(\Omega) = 1$
2. If the sequence of sets $\{A_i\}_{i=1}^{\infty} \in F$ and are disjoint, then $P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$.

The triplet (Ω, F, P) is a probability space.

Let (Ω, F, P) be a probability space. The sequence of random variables (Y_n) converges in limit (convergence in probability) to Y, i.e, $Y_n \rightarrow Y$ if,

$$P(\omega : |Y_n(\omega) - Y(\omega)| > \epsilon) \rightarrow\ 0$$

as $n \rightarrow \infty$ and $\forall\ \epsilon > 0$

References

[1] Aurélien Alfonsi, Eric Cances, Gabriel Turinici, Barbara Di Ventura, and Wilhelm Huisinga. Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. In *ESAIM: proceedings*, volume 14, pages 1–13. EDP Sciences, 2005.

[2] David F Anderson. Incorporating postleap checks in tau-leaping. *The Journal of chemical physics*, 128(5):054103, 2008.

[3] David F Anderson and Desmond J Higham. Multilevel monte carlo for continuous time markov chains, with applications in biochemical kinetics. *Multiscale Modeling & Simulation*, 10(1):146–179, 2012.

[4] Adam Arkin, John Ross, and Harley H McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149(4):1633–1648, 1998.

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

[6] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Efficient step size selection for the tau-leaping simulation method. *The Journal of chemical physics*, 124(4):044109, 2006.

[7] Abhijit Chatterjee, Dionisios G Vlachos, and Markos A Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *The Journal of chemical physics*, 122(2):024112, 2005.

[8] Stewart N Ethier and Thomas G Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.

[9] C Gardiner. Stochastic methods: a handbook for the natural and social sciences 4th ed.(2009).

[10] Michael B Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[11] 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.

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

[13] Daniel T Gillespie. The chemical langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

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

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

[16] 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.

[17] Andreas Hellander and Per Lötstedt. Hybrid method for the chemical master equation. *Journal of Computational Physics*, 227(1):100–122, 2007.

[18] Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

[19] Desmond J Higham. Modeling and simulating chemical reactions. *SIAM review*, 50(2):347–368, 2008.

[20] Silvana Ilie, Wayne H Enright, and Kenneth R Jackson. Numerical solution of stochastic models of biochemical kinetics. *Canadian Applied Mathematics Quarterly*, 17(3):523–554, 2009.

[21] Hye-Won Kang, Thomas G Kurtz, et al. Separation of time-scales and model reduction for stochastic reaction networks. *The Annals of Applied Probability*, 23(2):529–583, 2013.

[22] Thomas R Kiehl, Robert M Mattheyses, and Melvin K Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20(3):316–322, 2004.

[23] Thomas R Kiehl, Robert M Mattheyses, and Melvin K Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20(3):316–322, 2004.

[24] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206, 2002.

[25] Christopher Lester, Ruth E Baker, Michael B Giles, and Christian A Yates. Extending the multi-level method for the simulation of stochastic biological systems. *Bulletin of mathematical biology*, 78(8):1640–1677, 2016.

[26] Tiejun Li. Analysis of explicit tau-leaping schemes for simulating chemically reacting systems. *Multiscale Modeling & Simulation*, 6(2):417–436, 2007.

[27] Shev MacNamara, Alberto M Bersani, Kevin Burrage, and Roger B Sidje. Stochastic chemical kinetics and the total quasi-steady-state assumption: application to the stochastic simulation algorithm and chemical master equation. *The Journal of chemical physics*, 129(9):09B605, 2008.

[28] Bence Mélykúti, Kevin Burrage, and Konstantinos C Zygalakis. Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical langevin equation. *The Journal of chemical physics*, 132(16):164109, 2010.

[29] Jacek Puchałka and Andrzej M Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86(3):1357–1372, 2004.

[30] 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.

[31] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003.

[32] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Consistency and stability of tau-leaping schemes for chemical reaction systems. *Multiscale Modeling & Simulation*, 4(3):867–895, 2005.

[33] 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.

[34] Asawari Samant and Dionisios G Vlachos. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: a multiscale monte carlo algorithm. *The Journal of chemical physics*, 123(14):144114, 2005.

[35] Michael Samoilov, Sergey Plyasunov, and Adam P Arkin. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2310–2315, 2005.

[36] Tianhai Tian and Kevin Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of chemical physics*, 121(21):10356–10364, 2004.

[37] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[38] E Weinan, Di Liu, and Eric Vanden-Eijnden. Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales. *Journal of computational physics*, 221(1):158–180, 2007.

[39] Darren J Wilkinson. *Stochastic modelling for systems biology.* Chapman and Hall/CRC, 2006.