Journal of Applied Analysis and Computation
Volume 6, Number 2, May 2016, 322–335
Website:http://jaac-online.com/
DOI:10.11948/2016025

SOLVING STOCHASTIC CHEMICAL KINETICS BY METROPOLIS-HASTINGS SAMPLING

Azam Mooasvi[1,†], Paul Tranquilli[1] and Adrian Sandu[1]

Abstract This study considers using Metropolis-Hastings algorithm for stochastic simulation of chemical reactions. The proposed method uses SSA (Stochastic Simulation Algorithm) distribution which is a standard method for solving well-stirred chemically reacting systems as a desired distribution. A new numerical solvers based on exponential form of exact and approximate solutions of CME (Chemical Master Equation) is employed for obtaining target and proposal distributions in Metropolis-Hastings algorithm to accelerate the accuracy of the tau-leap method. Samples generated by this technique have the same distribution as SSA and the histogram of samples show it's convergence to SSA.

Keywords Metropolis-Hastings, SSA, CME, tau-leap.

MSC(2010) 92Bxx, 49Mxx, 62xx.

1. Introduction

In biological systems, chemical reactions are modeled stochastically. The system's state (the number of of molecules of each individual species) is described by probability densities describing the quantity of molecules of different species at a given time. The evolution of probabilities through time is described by the chemical master equation (CME) [6].

The Stochastic Simulation Algorithm (SSA) first introduced by Gillespie [6], is a Monte Carlo approach to sample from the CME. The accuracy of different approaches in simulating stochastic chemical reactions is compared to histogram of samples obtained by SSA. However, SSA has a number of drawbacks such as it simulates one reaction at a time and therefore it is inefficient for most realistic problems. Alternative approaches have been developed trying to enhance the efficiency of SSA but most of them suffer from accuracy issues. The explicit tau-leaping method [7] is able to simulate multiple chemical reactions in a pre-selected time step of length τ by using Poisson random variables [13]. However, explicit tau-leaping method is numerically unstable for stiff systems [3]. Different implicit tau-leap approaches have been proposed to alleviate the stability issue [1, 7, 8, 18]. Sandu [20] considers an exact exponential solution to the CME, leading to a solution vector that coincides with the probability of SSA. Several approximation methods to the exact exponential solution as well as approximation to the explicit tau-leap are given in [16].

[†]the corresponding author. Email address: azmosavi@vt.edu(A. Moosavi)
[1]Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060. Phone: (540) 231-2193, Fax: (540) 231-6075

The availability of exact and approximate probability solutions motivates the use of Markov chain Metropolis algorithm to enhance the accuracy of explicit tau-leap method when using large time steps. The proposed method relies on explicit tau-leaping to generate candidate samples. The proposed probability density corresponds to that of tau-leaping [20], and the target probability density is provided by the CME. During the Markov process the candidate samples are evaluated based on approximations of target and proposal probability and are either accepted or rejected. The proposed technique requires the computation of a matrix exponential during the Markov process. The dimension of matrix grows with increasing number of species in a reaction system. In order to manage the computational expense of matrix exponentiation efficient approaches based on Krylov [21] and rational approximations [15,17] are employed. Further computational savings are obtained by exponentiating only a sub-matrix that encapsulates the essential information about the transition of the system from the current to the proposed state.

The paper is organized as follows. Section 2 reviews Monte Carlo approaches, and Section 3 discusses the application of Metropolis-Hastings algorithm to sample from the probability distribution generated by CME. Computationally efficient methods to accelerate exponentiating the matrix are discussed in Section 4. Numerical experiments carried out in Section 5 illustrate the accuracy of the proposed schemes. Conclusions are drawn in Section 6.

2. Markov Chain Monte Carlo methods

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms to generate samples from desired probability distributions. A Markov chain is a discrete time stochastic process, i.e., a sequence of random variables (states) $x_0, x_1, \cdots$ where the probability of the next state of system depends only on the current state of the system and not on previous ones [2].

2.1. Metropolis methods

The Metropolis method is an MCMC process to obtain a sequence of random samples from a desired probability distribution $\pi(x)$, $x \subset X \subset \mathbb{R}^n$, which is usually complex. A Markov chain with state space X and equilibrium distribution $\pi(x)$ is constructed and long runs of the chain are performed [19]. The original MCMC algorithm was given by Metropolis et al. [14] and was later modified by Hastings [9], with a focus on statistical problems.

A random walk is performed around the current state of the system x_{t-1}. A proposal distribution $g\left(x^*|x_{t-1}\right)$ is used to suggest a candidate x^* for the next sample given the previous sample value x_{t-1}. The proposal distribution should be symmetric $g\left(x_{t-1}|x^*\right) = g\left(x^*|x_{t-1}\right)$. The algorithm works best if the proposal density matches the shape of the target distribution, i.e. $g\left(x_{t-1}|x^*\right) \approx \pi(x)$. Proposals x^* are accepted or rejected in a manner that leads system toward the region of higher target probability $\pi(x)$ [11]. Specifically, one computes the target density ratio

$$\alpha = \frac{\pi\left(x^*\right)}{\pi\left(x_{t-1}\right)}, \tag{2.1}$$

and draws a random variable $\zeta \sim uniform(0,1)$. The proposal is accepted or

rejected as follows:

$$x_t := \begin{cases} x^* & \text{if} \quad \zeta < \min(1, \alpha) \text{ (proposal accepted)}, \\ x_{t-1} & \text{otherwise} \qquad\qquad \text{(proposal rejected)}. \end{cases}$$

2.2. Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm allows more freedom in the choice of the proposal distribution by relaxing the symmetry constraint [5]. The acceptance ratio (2.1) is changed to

$$\alpha = \alpha_1 \cdot \alpha_2. \tag{2.2}$$

Here α_1 is the ratio between the target probabilities of the proposal sample x^* and of the previous sample x_{t-1}. This can be evaluated by a function f which is an approximation of π

$$\alpha_1 = \frac{\pi(x^*)}{\pi(x_{t-1})} \approx \frac{f(x^*)}{f(x_{t-1})}, \tag{2.3}$$

α_2 is the ratio of the proposal densities of x^* conditioned by x_{t-1}, and of x_{t-1} conditioned by x^*. This ratio is equal to one if the proposal distribution is symmetric.

$$\alpha_2 = \frac{g(x_{t-1}|x^*)}{g(x^*|x_{t-1})}. \tag{2.4}$$

Convergence of the Markov chain is guaranteed if the properties of detailed balance and ergodicity conditions are fulfilled [12]. Detailed balance requires that the probability of moving from x_{t-1} is the same as moving from x^*.

$$\pi(x_{t-1})\, g(x_{t-1}|x^*) = \pi(x^*)\, g(x^*|x_{t-1}).$$

Ergodicity requires that a chain starting from any state x_1 will return to x_1 if it runs long enough. In practice, it is not possible to establish with full certainty that a chain has converged [12].

3. Metropolis-Hastings for stochastic simulation

Here we discuss the application of the Metropolis-Hastings algorithm to generate samples from the CME distribution. SSA is currently the standard model for solving well-stirred chemically reacting systems; however, SSA does one reaction at a time that making it slow for real problems. On the other hand, alternative techniques such as explicit and implicit tau-leap methods are faster than SSA but suffer from low accuracy at larger time steps.

In the proposed approach, explicit tau-leap is employed to generate candidate samples. The samples are evaluated based on the acceptance ratio of Metropolis-Hastings algorithm. At the end of algorithm, the samples generated by this technique have the same distribution as given by CME, and the histogram of samples converges to the histogram of SSA solutions.

3.1. Target distribution

The target (exact) distribution $\mathcal{P}(x,t)$ of the state of the chemical system is given by the solution of the CME [6]

$$\frac{\partial \mathcal{P}(x,t)}{\partial t} = \sum_{r=1}^{M} a_r(x - v_r)\,\mathcal{P}(x - v_r, t) - a_0(x)\,\mathcal{P}(x,t)\,. \tag{3.1}$$

Let Q^i is the total possible number of molecules of species S^i, $i = 1,\ldots,N$. The total number of all possible states of the system is

$$Q = \prod_{i=1}^{N} \left(Q^i + 1\right). \tag{3.2}$$

CME is a linear ODE on the discrete state space of states $\mathbb{R}^Q$

$$\mathcal{P}' = A \cdot \mathcal{P}\,, \quad \mathcal{P}(\bar{t}) = \delta_{\mathcal{I}(\bar{x})}\,, \quad t \geq \bar{t}\,. \tag{3.3}$$

and has an exact solution:

$$\mathcal{P}(\bar{t} + \tau) = \exp(T\,A) \cdot \mathcal{P}(\bar{t}) = \exp\left(\tau \sum_{r=0}^{M} A_r\right) \cdot \mathcal{P}(\bar{t})\,. \tag{3.4}$$

As explained in [16,20], the diagonal matrix $A_0 \in \mathbb{R}^{Q\times Q}$ and the Toeplitz matrices $A_1, \cdots, A_M \in \mathbb{R}^{Q\times Q}$ are:

$$(A_0)_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}\,, \quad (A_r)_{i,j} = \begin{cases} a_r(x_j) & \text{if } i - j = d_r, \\ 0 & \text{if } i - j \neq d_r, \end{cases} \tag{3.5}$$

and their sum $A \in \mathbb{R}^{Q\times Q}$ is

$$A = A_0 + \cdots + A_M\,, \quad A_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j\,, \\ a_r(x_j) & \text{if } i - j = d_r,\ r = 1,\cdots,M\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.6}$$

Here x_j denotes the unique state with state space index $j = \mathcal{I}(x_j)$, where $\mathcal{I}(x)$ is the state-space index of state $x = [X^1, \ldots, X^N]$:

$$\begin{aligned} \mathcal{I}(x) = &\left(Q^{N-1} + 1\right) \cdots \left(Q^1 + 1\right) \cdot X^N + \cdots \\ &+ \left(Q^2 + 1\right)\left(Q^1 + 1\right) \cdot X^3 + \left(Q^1 + 1\right) \cdot X^2 + X^1 + 1. \end{aligned} \tag{3.7}$$

One firing of reaction R_r changes the state from x to $\bar{x} = x - v_r$. The corresponding change in state space index is:

$$\begin{aligned} &\mathcal{I}(x) - \mathcal{I}(x - v_r) = d_r, \\ &d_r = \left(Q^{N-1} + 1\right) \cdots \left(Q^1 + 1\right).v_r^N + \cdots \\ &\quad + \left(Q^2 + 1\right)\left(Q^1 + 1\right).v_r^3 + \left(Q^1 + 1\right).v_r^2 + v_r^1. \end{aligned}$$

At the current time $\bar{t}$ the system is in the known state $x(\bar{t}) = \bar{x}$ and consequently the current distribution $\mathcal{P}(\bar{t}) = \delta_{\mathcal{I}(\bar{x})}$ is equal to one at $\mathcal{I}(\bar{x})$ and is zero everywhere else. The target distribution in our method is the exact solution (3.4)

$$\pi = \exp\left(\tau \sum_{r=0}^{M} A_r\right) \cdot \delta_{\mathcal{I}(\bar{x})}\,. \tag{3.8}$$

3.2. Proposal distribution

In our algorithm the explicit tau-leap method is employed to generate the candidate samples. Sandu [20] shows that the probability distribution generated by the tau-leap method is the solution of a linear approximation of the CME

$$\mathcal{P}\left(\bar{t}+\tau\right)=\exp\left(\tau\,\bar{A}\right)\cdot\mathcal{P}\left(\bar{t}\right)=\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\mathcal{P}\left(\bar{t}\right), \tag{3.9}$$

where the diagonal matrix $\bar{A}_0 \in \mathbb{R}^{Q\times Q}$ and the Toeplitz matrices $\bar{A}_1,...,\bar{A}_M \in \mathbb{R}^{Q\times Q}$ are:

$$(\bar{A}_0)_{i,j}=\begin{cases}-a_0\left(\bar{x}\right) & \text{if } i=j,\\ 0 & \text{if } i\neq j,\end{cases},\quad (\bar{A}_r)_{i,j}=\begin{cases}a_r(\bar{x}) & \text{if } i-j=d_r,\\ 0 & \text{if } i-j\neq d_r,\end{cases} \tag{3.10}$$

where the arguments of all propensity functions are the current state $\bar{x}$ [16]. Therefore the proposal distribution used in our method is:

$$g=\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}\,. \tag{3.11}$$

3.3. Markov process

The Markov process starts with the values of species at the current time. The candidate sample is generated by the tau-leap method. Both the candidate sample and the current sample are evaluated based on the acceptance ratio (2.2). The target density ratio (2.3) is

$$\alpha_1=\frac{\pi\left(x^*\right)}{\pi\left(x_{t-1}\right)}=\frac{\delta^T_{\mathcal{I}(x^*)}\cdot\exp\left(\tau\sum_{r=0}^{M}A_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}{\delta^T_{\mathcal{I}(x_{t-1})}\cdot\exp\left(\tau\sum_{r=0}^{M}A_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}. \tag{3.12}$$

For the tau-leap method x^* is generated independent of x_{t-1} and vice versa. Hence the proposal density ratio (2.4) is

$$\alpha_2=\frac{g\left(x_{t-1}\right)}{g\left(x^*\right)}=\frac{\delta^T_{\mathcal{I}(x_{t-1})}\cdot\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}{\delta^T_{\mathcal{I}(x^*)}\cdot\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}. \tag{3.13}$$

From (3.12) and (3.13) the acceptance ratio α is:

$$\alpha=\frac{\delta^T_{\mathcal{I}(x^*)}\cdot\exp\left(\tau\sum_{r=0}^{M}A_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}{\delta^T_{\mathcal{I}(x_{t-1})}\cdot\exp\left(\tau\sum_{r=0}^{M}A_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}\cdot\frac{\delta^T_{\mathcal{I}(x_{t-1})}\cdot\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}{\delta^T_{\mathcal{I}(x^*)}\cdot\exp\left(\tau\sum_{r=0}^{M}\bar{A}_r\right)\cdot\delta_{\mathcal{I}(\bar{x})}}. \tag{3.14}$$

In the acceptance/rejection test, samples which have a higher density ratio will be selected as the next state and samples which have a lower density ratio will be rejected. The samples generated by this approach have approximately the same density as CME (SSA) even when using a large time step in the proposal (explicit tau-leap). The only drawback of this method is the cost of performing matrix exponential. In the following section we discuss several ways to reduce this computational cost.

4. Matrix exponential

The computation of a large matrix exponential is a problem of general interest, and a multitude of approaches suited to different situations are available. The most straightforward, and naive approach is a direct application of the definition of the matrix exponential

$$\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \tag{4.1}$$

While this approach is guaranteed to converge if sufficiently, possibly very many, terms are used, there are substantial numerical stability problems in the case where either the norm or the dimension of A is very large [15, 21].

4.1. Rational approximation methods for full matrix exponential

Several rational approximation methods have been developed to overcome the stability and speed of convergence problems posed by the direct method. These are based on standard function approximation methods, in the case of the Pade approximation [15], or on the approximation of complex contour integrals, in the case of CRAM [17]. These methods are usually paired with the "scaling and squaring" process of Higham [10] to further increase the stability of the computation.

4.1.1. Pade approximation

The Pade approximation for $\exp(A)$ is computed using the (p, q)-degree rational function:

$$\begin{aligned} P_{pq}(A) &= [D_{pq}(A)]^{-1} N_{pq}(A), \\ N_{pq}(A) &= \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} A^j, \\ D_{pq}(A) &= \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-A)^j \end{aligned}$$

which is obtained by solving the algebraic equation:

$$\sum_{k=0}^{\infty} \frac{A^k}{k!} - \frac{N_{pq}(A)}{D_{pq}(A)} = O\left(A^{p+q+1}\right),$$

in which $P_{pq}(x)$ must match the Taylor series expansion up to order $p + q$ [21]. MATLAB's `expm` function makes use of thirteenth order Pade approximation with scaling and squaring [15].

4.1.2. Rational approximations of integral contours

The integral contour approach constructs parabolic and hyperbolic contour integrals on left complex plane and uses quadrature points θ_k from the contour and

quadrature weights α_k [17] to approximate full matrix exponentiation:

$$r(z) = \sum_{k=1}^{N} \frac{\alpha_k}{z - \theta_k}.$$

In the case where the spectrum of A is confined to a region near the negative real axis of the complex plane methods based on the rational approximation of integral contours are likely to convergence faster than the Pade approximation. In this work we use the Matlab scripts provided by [17] for both rational approximation methods and the coefficients θ and α.

4.1.3. Chebyshev Rational Approximation Method (CRAM)

Let $\pi_{k,k}$ denote the set of rational functions

$$r_{k,k}(x) = \frac{p_k(x)}{q_k(x)},$$

where p_k and q_k are the polynomials of order k computed such as to optimize the following error

$$(p_k, q_k) = \arg \inf_{r_{k,k} \in \pi_{k,k}} \left\{ \sup_{x \in \mathbb{R}_-} \mid r_{k,k}(x) - e^x \mid \right\}.$$

The primary difficulty in making use of the CRAM method is the procurement of suitable coefficients of the polynomials p_k and q_k. A method for obtaining these coefficients is given in [4], and the values for $k = 14$ and $k = 16$ are provided in [17].

4.2. Krylov based approximation for matrix vector products

For our purposes we do not seek the entire solution of $\exp(A)$, in fact we would like only a single element of the result. Krylov based approximations get us one step closer to this ideal. Where the rational approximation methods seek to approximate the entirety of equation (4.1), Krylov based methods seek only an approximation to the matrix-vector product $\exp(A)b$.

This is done by first computing the m-dimensional Krylov subspace

$$\mathcal{K}_m = \text{span}\left\{b,\ Ab, \ldots, A^{m-1}b\right\},$$

using the Arnoldi iteration to compute the $n \times m$ orthonormal basis matrix V_m and the $m \times m$ upper Hessenberg matrix Hb_m with $m \ll n$ such that

$$\text{span}(V_m) = K_m, \qquad Hb_m = V^T A V.$$

The approximation is constructed as

$$\exp(A)b = V_m V_m^T \exp(A) V_m V_m^T b = \|b\| V_m \exp(Hb_m) e_1, \tag{4.2}$$

where e_1 is the first canonical basis vector. The small matrix exponential term in (4.2) can be computed using one of the rational approximation methods with scaling and squaring extremely cheaply. The EXPOKIT software of Sidje [21] makes use of these techniques, with some extra consideration for Markovian cases, where the approximation of $w(t) = \exp(tA)v$ is subject to the constraint that the resulting vector is a probability vector with components in the range of [0, 1] and the sum of these components is approximately one.

4.3. Faster approximation techniques for a single element

Since we seek only a single element of the matrix exponential $(\exp(A))_{i,j}$ we propose two techniques to speed up this computation.

4.3.1. A single element Krylov approach

Using equation (4.2) with $b = e_j$ leads to

$$(\exp(A))_{i,j} = e_i^T \exp(A)\, e_j = (e_i^T\, V_m)\, (\exp(Hb_m)\, e_1). \tag{4.3}$$

The exponential matrix entry is computed for the cost of an m-dimensional Pade approximation and an m-dimensional dot product since $(e_i^T V_m)$ can be computed for "free" by simply reading off the ith row of V_m, and similarly $(\exp(Hb_m)e_1)$ is just the first column of $\exp(Hb_m)$. This approach avoids the construction of any additional n-dimensional vectors or their products.

4.3.2. Exponentiation of a selected sub-matrix

Computing the exponential of a large matrix is expensive. When the number of species in a reaction system is high, the dimensions of the matrix (3.5) for target probability as well as dimensions of matrix (3.10) for proposal probability grow quickly. For the case of n species where each has a maximum Q molecules the dimension of matrix will be $(Q+1)^n \times (Q+1)^n$.

In order to reduce costs we propose to exponentiate a sub-matrix of the full matrix. The selected rows and columns contain indices of both the current state of system at t_n and candidate state at $t_n + \tau$. The motivation comes from the fact that states which are far from the current and the proposed ones do not impact significantly the acceptance/rejection test of Metropolis-Hastings algorithm and can be disregarded. Numerical experiments indicate that the error in an element $(\exp(A))_{i,j}$ computed using a properly sized sub-matrix instead of full matrix is small.

In order to obtain the proper size of a sub-matrix for each reaction system, we use specific information from the reaction system such as propensity functions, time step and maximum number of molecules in the system. Recall the general tau-leap formula [7].

$$x\,(\bar{t}+\tau) = x\,(\bar{t}) + \sum_{j=1}^{M} V_j\, K\,(a_j\,(x\,(\bar{t}))\,\tau)\,,$$

where $K\,(a_j\,(x\,(\bar{t}))\,\tau)$ is a random number drawn from a Poisson distribution with parameter $a_j\,(x\,(\bar{t}))\,\tau$ and V_j is the j-th column of stoichiometry matrix. The expected value of the jump in the number of molecules is

$$\mathtt{E}[x\,(\bar{t}+\tau) - x\,(\bar{t})] = \sum_{j=1}^{M} V_j\, a_j\,(x\,(t_0))\;\tau. \tag{4.4}$$

Motivated by (4.4) we consider the following initial estimate of the size of the sub-matrix:

$$S \propto \frac{\|V\|}{N} \sum_{j=1}^{M} a_j\,(x\,(t_0))\;\tau \propto \bar{a}\,(x\,(t_0))\;\tau, \tag{4.5}$$

where $\bar{a}\left(x\left(t_0\right)\right)$ is the average of propensity functions for all reactions calculated at the initial values of number of molecules.

We seek to select a range of state indices that covers the current and proposed states. The sub-matrices are built by selecting only the rows and columns in this range from (3.5) and (3.10). If the range of indices is small then the exponential computations are fast. However, if this range does not cover the representative states (both the current sample and the proposed sample), the probability ratio of the proposed sample can be far from the target probability, and the proposed sample is likely to be rejected. Choosing the size of the sub-matrix for maximum efficiency has to balance the cost of obtaining a sample (smaller is better for the cost of exponentiation) with the likelihood of accepting samples (larger is better for accuracy of approximation).

5. Numerical experiments

This section discusses the application of the Metropolis-Hastings algorithm to generate samples from the SSA distribution for three test systems: Schlogl [3], reversible isomer [3], and Lotka Volterra reactions [6].

5.1. Schlogl reaction

We first consider the Schlogl reaction system from [3]

$$\begin{aligned} &\mathrm{B}_1 + 2\,\mathrm{x} \underset{\mathrm{c}_2}{\overset{\mathrm{c}_1}{\rightleftharpoons}} 3\,\mathrm{x}, \\ &\mathrm{B}_2 \underset{\mathrm{c}_4}{\overset{\mathrm{c}_3}{\rightleftharpoons}} \mathrm{x}, \end{aligned} \tag{5.1}$$

whose solution has a bi-stable distribution. Let N_1, N_2 be the numbers of molecules of species B_1 and B_2, respectively. The reaction stoichiometry matrix and the propensity functions are:

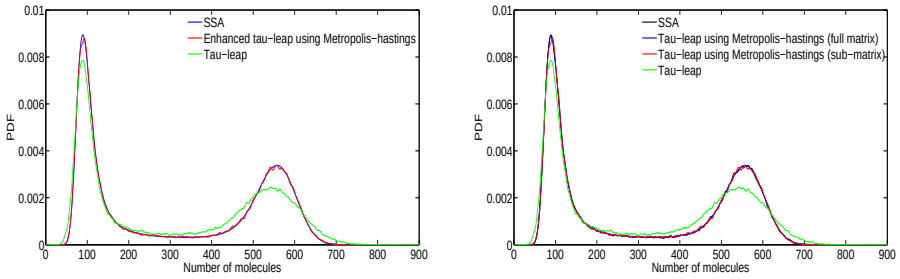
$$V = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}, \qquad \begin{aligned} a_1(x) &= \tfrac{c_1}{2} N_1 x(x-1), \\ a_2(x) &= \tfrac{c_2}{6} N_1 x(x-1)(x-2), \\ a_3(x) &= c_3 N_2, \\ a_4(x) &= c_4 x. \end{aligned}$$

The following parameter values (each in appropriate units) are used:

$$\begin{aligned} &c_1 = 3 \times 10^{-7},\ c_2 = 10^{-4}, \qquad c_3 = 10^{-3}, \\ &c_4 = 3.5, \qquad\quad N_1 = 1 \times 10^5,\ N_2 = 2 \times 10^5, \end{aligned}$$

with final time $T = 4$, initial conditions $x(0) = 250$ molecules, and maximum values of species $Q^1 = 900$ molecules. We consider a time step $\tau = 0.4$ for which the explicit tau-leap solution has a relatively large error compared to SSA.

The initial guess for the size of sub-matrix given by (4.5) is 250×250 and works well for the model. To accept $1,000$ samples the MCMC process rejects about $\sim 1,200$ samples when using full matrix (whose size is 901×901). While



(a) (b)

Figure 1. Histograms of Schlogl system (5.1) solutions with $\tau = 0.4$ (units), final time T=4 (units), and 10,000 samples.

the number of rejected using sub-matrix is approximately $1,300$. Decreasing the size of sub-matrix leads to a larger number of rejected samples. For example using a sub-matrix of size 100×100 results in approximately $2,500$ rejected samples, so this matrix size is too small. Another metric to assess whether the sub-matrix size is appropriate is the size of the residual obtained by exponentiating the full matrix and the sub-matrix. In this simulation the residual is below 10^{-8} for a sub-matrix size of 250×250. We have observed empirically that when the residual is larger than 10^{-2} the sample is likely to be rejected. The moderate number of rejected samples using the sub-matrix and the small residual indicate that the 250×250 size yields a good approximation for large matrix exponentiation.

Figure 1(a) illustrates the histogram of Schlogl reaction results obtained by SSA, explicit tau-leap and Metropolis-Hastings using full matrix size. Figure 1(b) shows that the results obtained with a sub-matrix of size 250×250 have no visible reduction in accuracy. Since all the eigenvalues of the matrix lie very closely to each other we employ the order ten of rational approximation of integral contours discussed in Section 4.1.2 which is faster than other techniques for exponentiating both the full matrix and the sub-matrix. The CPU time for obtaining one sample using the sub-matrix is 0.32 sec., about half the CPU time required when using the full matrix (0.76 sec). For comparison the CPU times for obtaining one sample are 0.15 sec. using SSA and 0.02 sec. using tau-leaping.

5.2. Isomer reaction

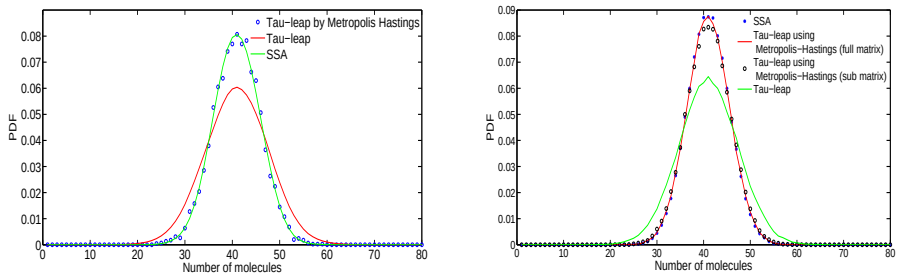
The reversible isomer reaction system from [3] is given by:

$$\mathrm{x}_1 \underset{\mathrm{c}_2}{\overset{\mathrm{c}_1}{\rightleftharpoons}} \mathrm{x}_2 \tag{5.2}$$

and the stoichiometry matrix and the propensity functions are:

$$V = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \begin{aligned} a_1(x) &= c_1 x_1\,, \\ a_2(x) &= c_2 x_2\,. \end{aligned}$$

The reaction rate values are $c_1 = 10$, $c_2 = 10$ (units), the time interval is $[0, T]$ with $T = 10$ (time units), the initial conditions are $x_1(0) = 40$, $x_2(0) = 40$ molecules, and the maximum values of the species are $Q^1 = 80$ and $Q^2 = 80$ molecules.



(a) (b)

Figure 2. Histograms of isomer system (5.2) solutions with $\tau = 0.05$ (units), final time T=1 (units), and 10,000 samples.

The estimate give by equation (4.5) is 20 and since this reaction system has two species the initial guess for the size of the sub-matrix is $20^2 \times 20^2$. In order to be more conservative a sub-matrix of size 500×500 is selected. In order to accept $1,000$ samples the Markov process rejects approximately $5,000$ samples when using the full matrix (of size $6,561 \times 6,561$) , and about $8,000$ samples when using the sub-matrix. Decreasing the size of sub-matrix leads to many more rejected samples. Our empirical observations show again that when the residual is larger than 10^{-2} the sample is likely to be rejected. We conclude that the current sub-matrix provides a good approximation for large matrix exponentiation.

Figure 2(a) shows the histogram of the isomer reaction solutions obtained by SSA, explicit tau-leap and by Metropolis-Hastings using the full size matrix (3.5) and (3.10). Figure 2(b) shows the results using the sub-matrix of size 500×500. There is no visible reduction in accuracy. Since all the eigenvalues of the matrix lie very closely to each other we employ the order ten of rational approximation of integral contours discussed in Section 4.1.2 which is faster than other techniques for exponentiating both the full matrix and the sub-matrix. The CPU times for obtaining one sample are 20.37 sec. using the sub-matrix and 38.70 sec. using the full matrix. For comparison obtaining one sample using SSA takes 0.15 sec. and using tau-leaping takes 0.05 sec.

5.3. Lotka Volterra reaction

The last test case is Lotka Volterra reaction system [6]:

$$\begin{aligned} Y + x_1 &\xrightarrow{c_1} 2x_1, \\ x_1 + x_2 &\xrightarrow{c_2} 2x_2, \\ x_2 &\xrightarrow{c_3} Y, \\ x_1 &\xrightarrow{c_4} Y. \end{aligned} \tag{5.3}$$

The reaction stoichiometry matrix and the propensity functions are:

$$V = \begin{bmatrix} 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}, \quad \begin{aligned} a_1(x) &= c_1 x_1 Y\,, \quad a_2(x) = c_2 x_2 x_1, \\ a_3(x) &= c_3 x_2\,, \quad a_4(x) = c_4 x_1. \end{aligned}$$

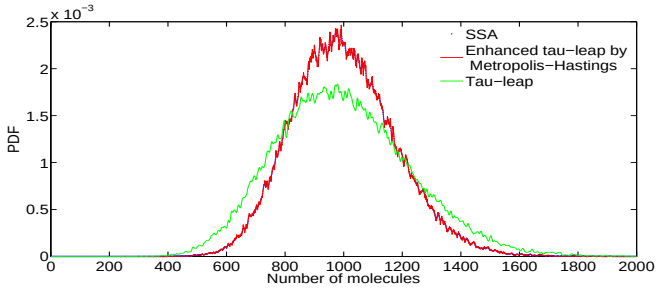


Figure 3. Histograms of Lotka-Volterra system (5.3) solutions with $\tau = 0.01$ (units), final time T=1 (units), and 10,000 samples.

The following parameter values are used (in appropriate units):

$$c_1 = 0.0002, \quad c_2 = 0.01, \quad c_3 = 10, \quad c_4 = 10, \quad Y = 10^{-5},$$

the final time is $T = 1$, the initial conditions are $x_1(0) = 1000$, $x_2(0) = 1000$ molecules, and the maximum values of species are $Q^1 = 2000$ and $Q^2 = 2000$ molecules. The resulting full matrix has dimension $2001^2 \times 2001^2$ and exponentiation is not feasible without the sub-matrix approximation.

The value predicted by equation (4.5) is 125, and since this reaction system has two species the initial guess for the size of sub-matrix is $15,625 \times 15,625$. This size does not work well for this system with very large number of molecules and almost never covers both current and candidate states. We increase the size of sub-matrix to $500,000 \times 500,000$, a value obtained by trial and error. Figure 3 illustrates the histogram of Lotka-Volterra solutions obtained by SSA, tau-leap method, and Metropolis-Hastings using a sub-matrix of size discussed above. The Metropolis-Hastings sampling is very accurate.

The CPU time of matrix exponentiation using the contour integral method discussed in Section 4.1.2 is four times faster than using the Krylov method discussed in Section 4.2. However, the Krylov method gives more accurate and stable results for large matrices [21]. Using 30 vectors in Krylov method, gives smaller number of rejected samples during the Markov process than the number of rejected samples using contour integral method. The CPU time for obtaining one sample using Metropolis-Hastings is a few hours, in comparison to 1.51 sec. using SSA and 0.21 sec. using tau-leap.

6. Conclusions

This study applies the Metropolis-Hastings algorithm to stochastic simulation of chemical kinetics. The proposed approach makes use of the CME and the exponential form of its exact solution as the target probability in the Markov process. The approximation of the explicit tau-leap method is then employed for the proposal probability. The samples generated by constructing the Markov process have the same distribution as SSA even the proposals are obtained using explicit tau-leap with a large time step. Computing matrix exponentials can become a computational bottleneck for large systems. Efficient approximation methods are needed. A practical approach consists of selecting a sub-matrix and exponentiating it using

fast approaches like Expokit and rational approximation to significantly reduce the cost of the algorithm. Current work of the authors focuses on developing faster approximation techniques that compute a single element of the matrix exponential. While the practical performance of the Metropolis-Hastings stochastic simulation algorithm is not yet competitive with existing methods, the approach is novel and opens the door for many future developments.

Acknowledgements

This work was partially supported by awards NSF DMS–1419003, NSF CCF–1218454, AFOSR FA9550–12–1–0293–DEF, AFOSR 12–2640–06, and by the Computational Science Laboratory at Virginia Tech.

References

[1] T. Ahn and A. Sandu, *Implicit simulation methods for stochastic chemical kinetics*, (2013). Http://arxiv.org/abs/1303.3614.

[2] A. Altaleb and D. Chauveau, *Bayesian analysis of the logit model and comparison of two metropolis-hastings strategies.*, Comput. Stat. Data Anal, 39(2002)(2), 137–152.

[3] Y. Cao, R. Petzold, M.Rathinam and D. Gillespie, *The numerical stability of leaping methods for stochastic simulation of chemically reacting systems.*, Journal of Chemical Physics, 121(2004)(24), 12169–12178.

[4] A. Carpenter, A. Ruttan and R. Varga, *Extended numerical computations on the 1/9 conjecture in rational approximation theory.*, Rational approximation and interpolation in lecture notes in mathematics, Springer., (1984)(1105), 383–411.

[5] S. Chib and E. Greenberg, *Understanding the metropolis-hastings algorithm.*, The American Statistician., 49(1995)(4), 327–335.

[6] D. Gillespie, *Exact stochastic simulation of coupled chemical reactions*, Journal of Chemical Physics, 81(1977)(25), 2340–2361.

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

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

[9] W. Hastings, *Monte carlo sampling methods using markov chains and their applications*, Biometrika, 57(1970).

[10] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited.*, SIAM review., 51(2009)(4), 747–764.

[11] D. Hitchcock, *A history of the metropolis-hastings algorithm*, The American Statistician, 75(2003)(4), 254–257.

[12] C. Kastner, A. Braumann, L. Man et al., *Bayesian parameter estimation for a jet-milling model using metropolishastings and wanglandau sampling*, Chemical Engineering Science., 99(2013)(4), 244–257.

[13] T. Kurtz, *The relationship between stochastic and deterministic models for chemical reactions.*, Journal of Chemical Physics, 57(1972)(7), 2976–2978.

[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth et al., *Equations of state calculations by fast computing machine*, Journal of Chemical Physics, (1953)(21), 1087–1091.

[15] C. Moler and C. Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45(2003)(1).

[16] A. Mooasvi and A. Sandu, *Approximate exponential algorithms to solve the chemical master equation*, Mathematical Modelling and Analysis, 20(2015)(3), 382–395.

[17] M. Pusa, *Rational approximations to the matrix exponential in burn up calculations.*, Nuclear science and engineering., 169(2010)(2), 155–167.

[18] M. Rathinam, L. Petzold, Y.Cao and D.Gillespie, *Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method*, Journal of Chemical Physics, 119(2003), 784–12.

[19] G. O. Roberts and A. F. M. Smith, *Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms.*, Stochastic Processes and their Applications, 49(1994)(2), 207–216.

[20] A. Sandu, *A new look at chemical master equation*, Numerical Algorithms, 65(2013)(3), 485–498.

[21] R. Sidje, *Expokit: A software package for computing matrix exponentials.*, ACM Trans. Software., 24(1998)(1), 130–156.