A Modified Uniformization Method for the Solution of the Chemical Master Equation

Jingwei Zhang[*], Layne T. Watson[*+], Yang Cao[+]

Departments of Mathematics[*], and Computer Science[+]

Virginia Polytechnic Institute and State University

Blacksburg, Virginia 24061-0106

e-mail: jwzhang@vt.edu, ltw@cs.vt.edu, ycao@cs.vt.edu

telephone: (540) 231-7540

Abstract

The chemical master equation is considered an accurate description of general chemical systems, and especially so for modeling cell cycle and gene regulatory networks. This paper proposes an efficient way of solving the chemical master equation for some prototypical problems in systems biology. A comparison between this new approach and some traditional approaches is also given.

Keywords: cell cycle model, chemical master equation, external uniformization method, gene regulatory network

1. Introduction

There are two ways to describe the dynamics of chemical reactions. One is deterministic description, which is accurate when the number of reacting molecules is large enough to allow a continuum point of view. In the modeling of cell cycle and gene regulatory networks, the number of molecules of a given chemical species is typically on the order of hundreds. In such a situation, the randomness in the system usually cannot be ignored. Thus, one is forced to adopt a stochastic description.

The chemical master equation ([1], [2]) is such a stochastic description, derived from the Markov property of the underlying stochastic process. The master equation is a formulation of the Markov property for discrete random variables in continuous time. If the chemical system is determined by specifying the number of molecules of each species, then the master equation governs the dynamics of the probability distribution for the system. It is well known that such a description suffers from the "curse of dimensionality", i.e., each species adds one dimension to the problem, and the computational complexity grows exponentially.

Currently, Monte Carlo algorithms are used to analyze the chemical master equation. These include the stochastic simulation algorithm (SSA) ([3], [4]), τ-leaping [5], and various other methods ([6]–[8]) that

make quasi-steady state assumptions in order to accelerate simulation time. These methods are exact in a statistical sense, and they simulate one trajectory at a time. Even though simulating one trajectory might be performed relatively cheaply, many trajectories need to be simulated in order to estimate statistical parameters accurately. Moreover, because of the explicit flavor of the method, simulating one trajectory itself may not be easy for stiff systems that require very small time steps.

Recently, there has been considerable work on solving the master equation directly ([9]–[12]). Basically, these approaches fit in two categories: an ordinary differential equation (ODE) point of view or a partial differential equation (PDE) point of view. From an ODE point of view, one simply integrates the master equation in time. The problem is that the ODE system dimension is often huge, so in order to make this approach feasible, many state approximation techniques have been proposed, like the sparse grids technique [9] and the finite state projection algorithm [10].

From a PDE point of view, the master equation is just a special kind of parabolic partial differential equation. The difficulty lies in that the state space is discrete. In a recent paper, Engblom [11] proposed a discrete version of the spectral method for the chemical master equation. Another alternative is solving the Fokker-Planck partial differential equation ([12], [13]); the Fokker-Planck equation can be regarded as a continuous approximation of the master equation. However, it is often difficult to determine a priori how good the approximation would be.

Taking the ODE point of view, success in solving the master equation depends on evaluating the matrix exponential series. In principle, the exponential of a matrix can be computed in many ways, but in practice, taking computational stability and efficiency into account, none of them are completely satisfactory [14]. Hence, the most appropriate method should be based upon particular properties of the matrix.

Uniformization or Jensen's Method ([15], [16]) is a special technique devised to compute the exponential of the infinitesimal generator of almost any continuous time Markov chain. It has a natural stochastic/probabilistic interpretation. It is easy to implement, only involves matrix-vector multiplication and is numerically stable. There is also a simple error bound for the matrix exponential approximation [17]. Conclusively, for most non-stiff problems, uniformization provides an accurate and economical numerical solution. Unfortunately, for stiff problems, often the case in biological systems, it is computationally inefficient.

To deal with stiff problems, Ross [18] proposed a new approach based on uniformization in 1987. Stiffness is managed by assuming the observation time intervals to be random variables with Erlangian distribution. Such a technique has also been called external uniformization [19]. The major difficulty with this approach is that it requires solving a large linear system obtained from the state transition rate matrix [20]. Typically though, in biological systems, this state transition rate matrix is ultra sparse. Therefore well-developed

sparse linear system solvers are applicable. Numerical experiments presented here show that this approach is quite successful for several biological system models.

The remainder of this paper is organized as follows: in Section 2, a biological toggle switch model is introduced, and a brief comparison between different methods is given for this typical problem. The third section illustrates the uniformization method and Ross' modified algorithm. Numerical results for several molecular biology models are given in Section 4. The final section concludes with a discussion of the capabilities of Ross' algorithm in the field of systems biology and suggests future research directions.

2. Toggle Switch

2.1 Model

It has been proposed that gene regulatory networks with virtually any desired property can be constructed from simple regulatory elements. Examples of such properties include multistability and oscillations [21]. A genetic toggle switch, which has been constructed in *Escherichia coli* already, is such a simple regulatory element. It is a synthetic bistable gene regulatory network that could be constructed from any two repressible promoters. The bistability of the toggle switch is obtained from the mutually inhibitory arrangement of these two [21]. The reaction equations are

$$\left.\begin{array}{lll} \emptyset & \overset{\alpha/(\beta+x_2^2)}{\longrightarrow} & x_1 \\ x_1 & \overset{\mu x_1}{\longrightarrow} & \emptyset \\ \emptyset & \overset{\gamma/(\delta+x_1^2)}{\longrightarrow} & x_2 \\ x_2 & \overset{\mu x_2}{\longrightarrow} & \emptyset \end{array}\right\}$$

with parameters $\alpha = \gamma = 1000, \beta = \delta = 6000$ and $\mu = 10^{-3}$ [11].

Suppose the two promoters are x_1 and x_2, and initially, the number of x_1 molecules is larger than that of x_2. Then one can observe that the production of x_2 molecules is inhibited by the large number of x_1 molecules, so that the system will reach a stable state with $x_1 > x_2$. However, with a certain small probability, the stochastic noise eventually makes the number of x_2 molecules grow. If this is the case, then the production of x_1 molecules will be inhibited by the growing number of x_2 molecules and the roles of x_1 and x_2 may switch [11]. One interesting feature of the toggle switch model is that any deterministic simulation of the model will only predict one stable state, which makes the stochastic simulation crucial in this situation.

2.2 Master equation

Define $p(x,t)$ to be the probability that the chemical system has a molecular population vector x at time t. Suppose that the state of the system can change through M different reaction channels and $a_i(x)$ and $w^{(i)}$ are the nonnegative propensity function and the stoichiometric transition vector, respectively, for reaction channel i [10], i.e.,

$$x - w^{(i)} \overset{a_i(x-w^{(i)})}{\longrightarrow} x.$$

The master equation [1] is then given by

$$\frac{\partial p(x,t)}{\partial t} = \sum_{i=1}^{M} a_i(x - w^{(i)}) p(x - w^{(i)}, t) - a_i(x) p(x,t).$$

Now, one can write down the master equation for this special toggle switch model. Define operators A_1 and A_2 by

$$A_1 p(x,t) = \mu(x_1+1)p(x_1+1, x_2, t) + \frac{\alpha}{\beta + x_2^2} p(x_1 - 1, x_2, t) - \left(\mu x_1 + \frac{\alpha}{\beta + x_2^2}\right) p(x,t)$$

and

$$A_2 p(x,t) = \mu(x_2+1)p(x_1, x_2+1, t) + \frac{\gamma}{\delta + x_1^2} p(x_1, x_2 - 1, t) - \left(\mu x_2 + \frac{\gamma}{\delta + x_1^2}\right) p(x,t).$$

Then the master equation is given by

$$\frac{\partial p(x,t)}{\partial t} = A_1 p + A_2 p.$$

2.3 Results

In references [11] and [9] the toggle switch problem is solved with different formulations. In [11] the same problem is solved on a state space $\mathbf{Z}^2_{201}$, the integer lattice points in $[0,200] \times [0,200]$. The master equation is approximated by an ODE system of dimension 400. The ODE system is then solved in Matlab (ode15s). Only solutions and errors in different norms are reported in [11]. In [9] the problem setting is a little different with a state space $\mathbf{Z}^2_{51}$. A sparse grid technique is used to make the state space even smaller. The exponentials of the smaller matrix are then computed by some Krylov space projection methods [22]. As reported in [9], it takes less than 15 minutes to solve another problem with $16^{10} \approx 10^{12}$ states (1001 grid with 16 grid points each after approximation) on a 2 GHz AMD64 based PC with 1 GByte of memory. Sparse grid can also be used as an approximation technique in the proposed algorithm below.

The master equation for the toggle switch model can also be solved by Monte Carlo algorithms. For future comparison, Table 1 gives the computational efficiency (CPU time) of some Monte Carlo algorithms, implemented in StochKit [23].

Table 1: Computation cost (sec) of two different Monte Carlo algorithms. Each simulation starts at $(x_1, x_2) = (60, 10)$ and ends at time $t_f = 2.0 \times 10^5 s$. In the adaptive τ-leaping method, the error control epsilon is set at 0.03.

	10^3 runs	10^4 runs
SSA	52.7	526.7
Adaptive τ-leaping	49.3	492.9

3. Methods and Algorithms

3.1 Mathematical background

Let $\mathcal{X} = \{X(t), t \geq 0\}$ be a continuous time Markov chain (CTMC) with a state space $\mathcal{S}$. The number of possible states is finite and is equal to N. For any $i, j \in \mathcal{S}$, let

$$p_{ij}(t) = P[X(t) = j \mid X(0) = i].$$

Given initial state probability vector $\pi(0)$, one is interested in computing $\pi(t)$, the state probability vector at time t. Obviously,

$$\pi(t) = \pi(0)P(t),$$

where $P(t) = (p_{ij}(t))$.

Suppose that when in state i, the CTMC makes a transition into state j at an instantaneous rate q_{ij} and let $q_i = \sum_{j \neq i} q_{ij}$ denote the rate at which $\mathcal{X}$ leaves state i. Then the matrix

$$Q = \begin{bmatrix} -q_1 & q_{12} & \cdots & q_{1N} \\ q_{21} & -q_2 & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & -q_N \end{bmatrix}$$

is called the infinitesimal generator of $\mathcal{X}$. Since the state space $\mathcal{S}$ is finite, $P(t)$ satisfies both Kolmogorov's backward equations

$$P'(t) = QP(t),$$

and Kolmogorov's forward equations

$$P'(t) = P(t)Q.$$

The solution to these two equations is

$$\pi(t) = \pi(0)P(t) = \pi(0)e^{Qt} = \pi(0) \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!}.$$

This gives a stochastic background for the master equation. However, using a truncation of the above infinite summation to approximate $\pi(t)$ is subject to severe roundoff error [14].

3.2 Uniformization

Let $\eta = \max_{1 \le i \le N} q_i$. Then through uniformization, it can be shown that, for any $\lambda \ge \eta$

$$P(t) = e^{Qt} = e^{-\lambda t} e^{\lambda t (I + \frac{Q}{\lambda})} = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \tilde{P}^n,$$

where $\tilde{P} = I + \frac{1}{\lambda} Q$ is a probability matrix. Hence,

$$\pi(t) = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} v(n),$$

where $v(n) = \pi(0)\tilde{P}^n$ is the state probability vector of a discrete time Markov chain (DTMC) after n transitions. In practice, a piece of this infinite summation is used.

There are several interesting features about the uniformization method. It associates a DTMC and a Poisson process with the CTMC. The Poisson process is "internal", because its frequency parameter λ must be larger than η, a parameter characterizing how quickly the continuous time Markov process changes its state.

For a given error tolerance, uniformization without left side truncation (meaning that terms for small n in the infinite series are not dropped) requires $O(\eta t)$ terms. Even if the distribution is truncated from both sides (meaning that terms for both small and large n are dropped), it still requires $O(\sqrt{\eta t})$ terms. Additionally, successively squaring $\tilde{P}$ to get the first significant DTMC state probability vector requires time $O(N^3 \log l)$, where l is the number of terms in the left tail $\sum_{n=0}^{l-1}$ [24]. Hence, for large values of ηt and N, the computation can be cumbersome.

For example, consider the bistable toggle switch problem described above. If $\mathcal{S} = \mathbf{Z}_{201}^2$ and $t = 2.0 \times 10^5 s$, then the value of $\lambda t = \eta t$ would be 4×10^4. This means that at least 40000 matrix vector multiplications are needed, while the dimension of the square matrix and vector is also about 40000.

However, the uniformization method has its own advantages. It only involves matrix vector multiplication, so it is easy to implement and is numerically stable. There is also a simple error bound calculated from the Poisson process.

3.3 External uniformization method

In 1989, Ross introduced an external uniformization technique so that one could overcome the restriction on choosing λ larger than η. In the usual uniformization procedure, the CTMC is allowed to make transitions only at arrival epochs of a Poisson process, so λ has to be greater than η. In the external uniformization, instead, the CTMC is observed at arrival epochs of an independent Poisson process with any rate λ that has nothing to do with η.

Consider another random event $\mathcal{E}$, which occurs at times $\tau_1, \tau_2, ...$, where the intervals $\tau_i - \tau_{i-1}$, for $i > 0$, are i.i.d. exponential random variables with rate λ independent of the Markov process $\mathcal{X}$. Assume the initial state $X(0) = i$. Then either a transition of $\mathcal{X}$ occurs before event $\mathcal{E}$, or $\mathcal{E}$ occurs first, and the first event occurs with probability $q_i/(q_i + \lambda)$ and the second with probability $\lambda/(q_i + \lambda)$. From the memoryless property of the exponential distribution and conditioning on which event occurs first,

$$p_{ij}(\tau_1) = \frac{q_i}{q_i + \lambda} \sum_{k \neq i} p_{kj}(\tau_1) \frac{q_{ik}}{q_i} + \frac{\lambda}{q_i + \lambda} \delta_{ij},$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. Rewrite this into matrix form, giving

$$P(\tau_1) = \left(I - \frac{Q}{\lambda} \right)^{-1}.$$

From the Chapman-Kolmogorov equations

$$P(\tau_r) = \left(I - \frac{Q}{\lambda} \right)^{-r}.$$

Now, use the fact that $E[\tau_r] = r/\lambda$. If $\lambda = r/t$, then, as $r \to \infty$, the random variable τ_r approaches t. As a consequence, $P(\tau_r)$ should be a good approximation to $P(t)$ [17].

In his paper Ross also provides some reasonable evidence that this approach is accurate, even if a small number of steps r is chosen for the recursion. This is verified again in the numerical examples here.

3.4 Algorithm and error estimation

Algorithm:

(i) Choose appropriate r and set $\lambda = r/t$.

(ii) Compute the matrix $(I - \frac{1}{\lambda} Q)$.

(iii) Set $\pi_r^{(0)} = \pi(0)$. Solve the linear system $\pi_r^{(i)}(I - \frac{1}{\lambda} Q) = \pi_r^{(i-1)}$ recursively, for $i = 1, ..., r$. Then $\pi_r^{(i)}$ is an approximation for $\pi(it/r)$, and $\pi_r^{(r)}$ is the desired approximation for $\pi(t)$.

In order to prove the convergence, one just has to notice that

$$\lim_{r \to \infty} P(\tau_r) = \lim_{r \to \infty} \left(I - \frac{Qt}{r} \right)^{-r} = e^{Qt} = P(t).$$

There is also a rough error bound for this method, based on the variance of exponential distributions [25]:

(i)

$$|E[p_{ij}(\tau_r)] - p_{ij}(t)| \leq \frac{1}{2}(t^2/r) \max_{m \in \mathcal{S}} \sum_k |q_{mk} q_{kj}|,$$

(ii)

$$\max_{i,j \in \mathcal{S}} |E[p_{ij}(\tau_r)] - p_{ij}(t)| \leq \frac{1}{2}(t^2/r) \max_{m,l \in \mathcal{S}} \sum_k |q_{mk} q_{kl}|.$$

The proof of these bounds just follows from a Taylor expansion of $p_{ij}(\tau_r)$ and the fact that τ_r has mean t and variance t^2/r. In the numerical examples, however, the actual error is far smaller than this error bound.

4. Numerical Experiments

In this section the master equation of several models from molecular biology will be solved using the proposed method, including the toggle switch model. Comparisons between the proposed method and some other methods are also given when appropriate. The performance of the proposed method largely depends on the effectiveness of the sparse linear system solver. Hence, the algorithm has been implemented based on two different kinds of sparse linear system solvers: iterative [26] (such as GMRES) and direct [28] (such as Gaussian elimination). The iterative solver software package chosen here is SPARSKIT [27], and the direct solver package used is SuiteSparse [29]. There is also a comparison between these two.

4.1 A simple birth-death process

In this model x molecules are produced at a constant rate k and decayed at a rate proportional to the total number of molecules simultaneously. The reaction equations are

$$\left.\begin{array}{lll} \emptyset & \stackrel{k}{\rightarrow} & x \\ x & \stackrel{\mu x}{\rightarrow} & \emptyset \end{array}\right\}.$$

The master equation for this system is

$$\begin{aligned} \frac{\partial p(x,t)}{\partial t} &= kp(x-1,t) + \mu(x+1)p(x+1,t) \\ &\quad - (k+\mu x)p(x,t). \end{aligned}$$

This problem can be solved analytically if the initial data is given in the form of a Poisson distribution [30],

$$p(x,0) = \frac{a_0^x}{x!}e^{-a_0}.$$

In this case the solution is given by

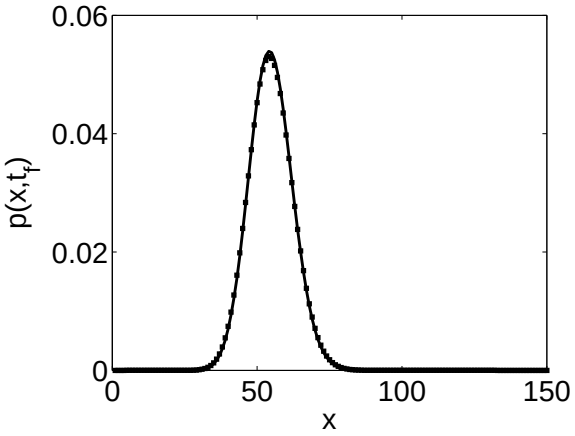
$$p(x,t) = \frac{a(t)^x}{x!}e^{-a(t)},$$

where $a(t) = a_0 e^{-\mu t} + \frac{k}{\mu}(1 - e^{-\mu t})$.

Fig. 1 displays the computational result using the algorithm proposed in Section 3.3, with parameters $k = 1$, $\mu = 0.01$, and $a_0 = 50$. It can be seen that as the final time t_f grows, larger r has to be chosen to get accurate solutions.

The solid line in Fig. 2 shows the L^∞ error ($t_f = 100s$) for different r values. It appears that the error decays rather slowly as r increases. Such a nearly linear log-log plot bodes well for acceleration, such as Aitken's δ^2 method [31] and extrapolation methods. The dotted line is the L^∞ error after applying Aitken's δ^2 method on $\pi_r^{(r)}$ and the dashed line is the L^∞ error after applying the same method on $\pi_{2^i}^{(2^i)}$. Both of them show improvements in error with no more expensive evaluations. However, for such a small problem, it takes less than a second to run the algorithm even if $r = 10000$, which gives an L^∞ error less than 10^{-5} even without acceleration.

(A)



(B)

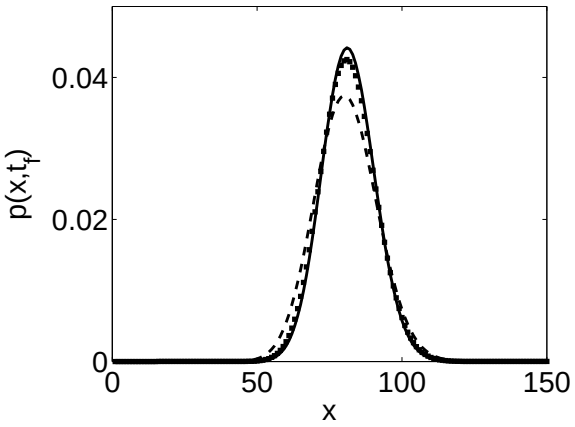


Fig. 1 *Solutions of the master equation pertaining to the simple birth-death process. (A)* $t_f = 10s$*, the solid line is the exact solution, the dotted line is the numerical solution* $\pi_{10}^{(10)}$*. (B)* $t_f = 100s$*, the solid line is the exact solution, dashed and dotted lines are the numerical solutions* $\pi_{10}^{(10)}$ *and* $\pi_{50}^{(50)}$*, respectively.*

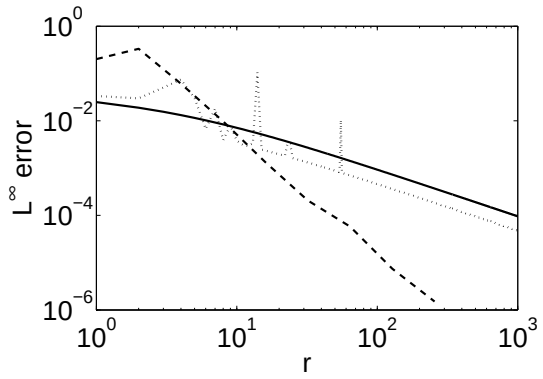


Fig. 2 *The* L^∞ *error versus* r *(*$t_f = 100s$*) before and after Aitken's* δ^2 *acceleration is applied.*

4.2 Schlögl reaction

The Schlögl reaction [32] is famous for its bistable distribution. The reaction equations are given by

$$\left.\begin{array}{ccc} b_1 + 2x & \underset{c_2}{\overset{c_1}{\rightleftharpoons}} & 3x \\ b_2 & \underset{c_4}{\overset{c_3}{\rightleftharpoons}} & x \end{array}\right\},$$

where b_1 and b_2 denote buffered species whose respective molecular populations are assumed to be constant. Now, the propensity functions are

$$\begin{aligned} a_1(x) &= \frac{c_1}{2} b_1 x(x-1), \\ a_2(x) &= \frac{c_2}{6} x(x-1)(x-2), \\ a_3(x) &= c_3 b_2, \\ a_4(x) &= c_4 x. \end{aligned}$$

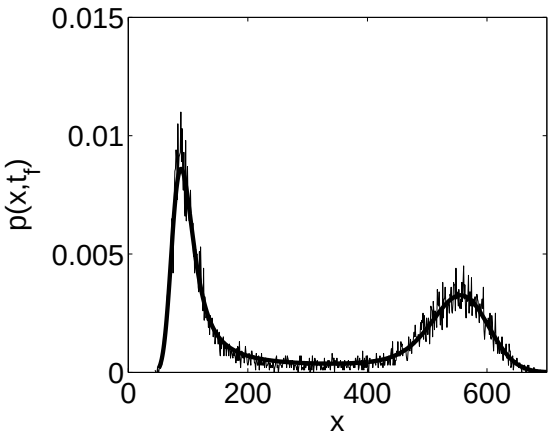
The master equation for this system is

$$\begin{aligned} \frac{\partial p(x,t)}{\partial t} = &\big(a_1(x-1) + a_3(x-1)\big)p(x-1,t) \\ &+ \big(a_2(x+1) + a_4(x+1)\big)p(x+1,t) \\ &- \sum_{i=1}^{4} a_i(x)p(x,t). \end{aligned}$$

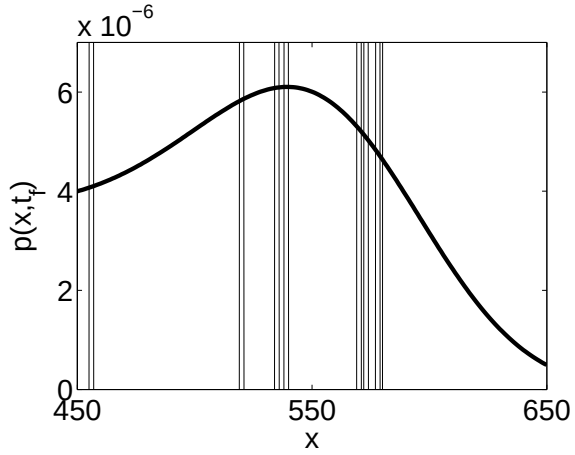
Fig. 3 compares the numerical results obtained from SSA and the external uniformization method. The parameters here are $c_1 = 3 \times 10^{-7}$, $c_2 = 10^{-4}$, $c_3 = 10^{-3}$, $c_4 = 3.5$, $b_1 = 1 \times 10^5$, $b_2 = 2 \times 10^5$, and the final time t_f is $4.0s$. This Schlögl model displays the bistable distribution only if the initial state is wisely chosen. In [32], the author has shown that this system has two stable states $x_1 = 82$ and $x_2 = 563$ and one barrier state $x_b = 248$. The bistable property of the distribution is apparent only when the initial state is close enough to the barrier state, as shown in Fig. 3 (A). In Fig. 3 (B), the initial state is set to the left of the barrier state. Trajectories starting from this initial state are more likely to end up around stable state x_1, which makes it more difficult to capture the other stable state x_2. Similarly, Fig. 3 (C) illustrates what happens when the initial state is set to the right of the barrier — most trajectories end up around stable state x_2, which makes stable state x_1 more difficult to capture. Figs. 3 (B) and (C) show that the proposed method captures the behavior of the distribution around the stable point with small probability density far better than histograms based on 10,000 SSA simulations.

Moreover, for such a small problem, the computational cost for the external uniformization method is negligible. For example, for $r = 30$, it takes approximately 0.018 sec if one uses a direct linear system solver, or 0.37 sec if an iterative one is used instead. On the other hand, the histogram estimation of the probability density function based on 10,000 simulation runs is quite rough, and costs nearly two minutes on the same machine.

(A)



(B)



(C)

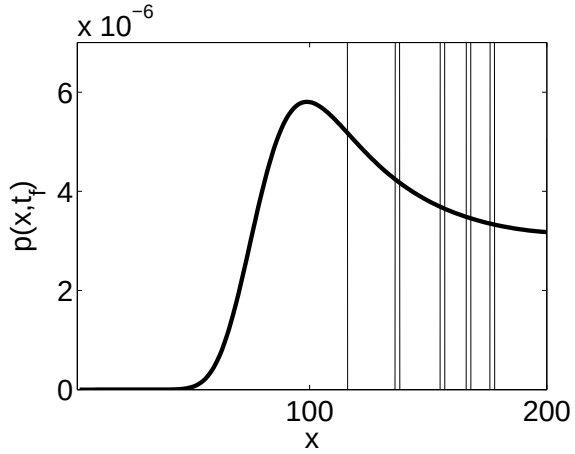


Fig. 3 *Comparison of the numerical results for the Schlögl reaction, with different initial states: (A)* $x = 250$*, (B)* $x = 150$*, (C)* $x = 400$*. The histogram (thin solid line) is based on 10,000 SSA simulations. The thick solid line is the numerical solution* $\pi_{30}^{(30)}$ *from the proposed method.*

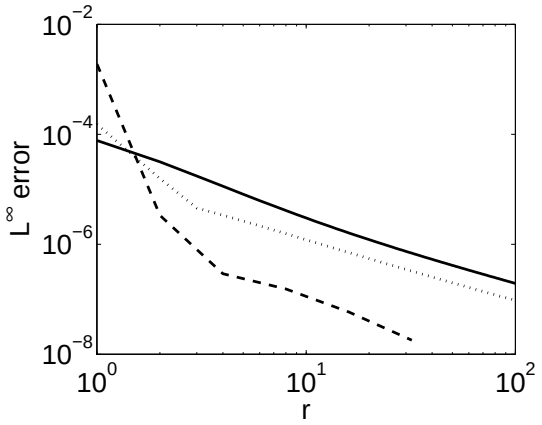


Fig. 4 *The L^∞ error versus r before and after Aitken's δ^2 acceleration is applied. The initial condition is $(x_1, x_2) = (60, 10)$ and the final time $t_f = 2.0 \times 10^5 s$.*

4.3 Toggle switch model

This model problem (also considered briefly in [33]) is solved with different r in order to estimate the convergence rate of the method. The error is estimated using a reference solution ($r = 10000$) and measured in the L^∞ norm. Here, Fig. 4 shows results similar to those in Fig. 2.

Fig. 5 contains the contour plots for numerical results. The bistable property is clearly apparent in each contour plot. Again, the contour plots obtained from SSA have much more noise than those two from the proposed method. Running 1,000,000 SSA simulations takes more than 10 hours, while the proposed method just takes a few seconds on the same machine. Fig. 5 also contains a contour plot of $\pi_{20}^{(10)}$, which approximates the probability density function at time $1.0 \times 10^5 s$. Actually, the vectors $\pi_{20}^{(i)}$ $(i = 1, ..., 20)$ approximate the probability density function at any epoch $\frac{i}{20} \times 2.0 \times 10^5 s$. These probability density functions altogether provide information on how the system reaches the equilibrium state.

Table 2 lists the CPU time for different r values with a comparison between iterative and direct linear system solvers. It shows that for this model problem the direct solver (UMFPACK, the unsymmetric multifrontal method for sparse LU factorization) performs better than the iterative solver (BiCGSTAB, the biconjugate gradient stabilized method).

Further analysis shows that for direct linear system solvers the computation mainly involves two parts: LU factorization and triangular system solve. Since the same sparse matrix is used in each step, only one LU factorization is needed. For example, for $r = 5$, the CPU time is $1.3s \approx 0.7s + 5 \times 0.06s + 0.3s$. The LU factorization takes 0.7, triangular system solve takes $5 \times 0.06s$, and the remaining $0.3s$ is for initialization and I/O. The run time of LU factorization and triangular system solve mainly depends on the structure of the sparse matrix, which implies that one LU factorization or one triangular system solve takes almost the same amount of time for different r and different steps. Therefore, the total run time increases nearly

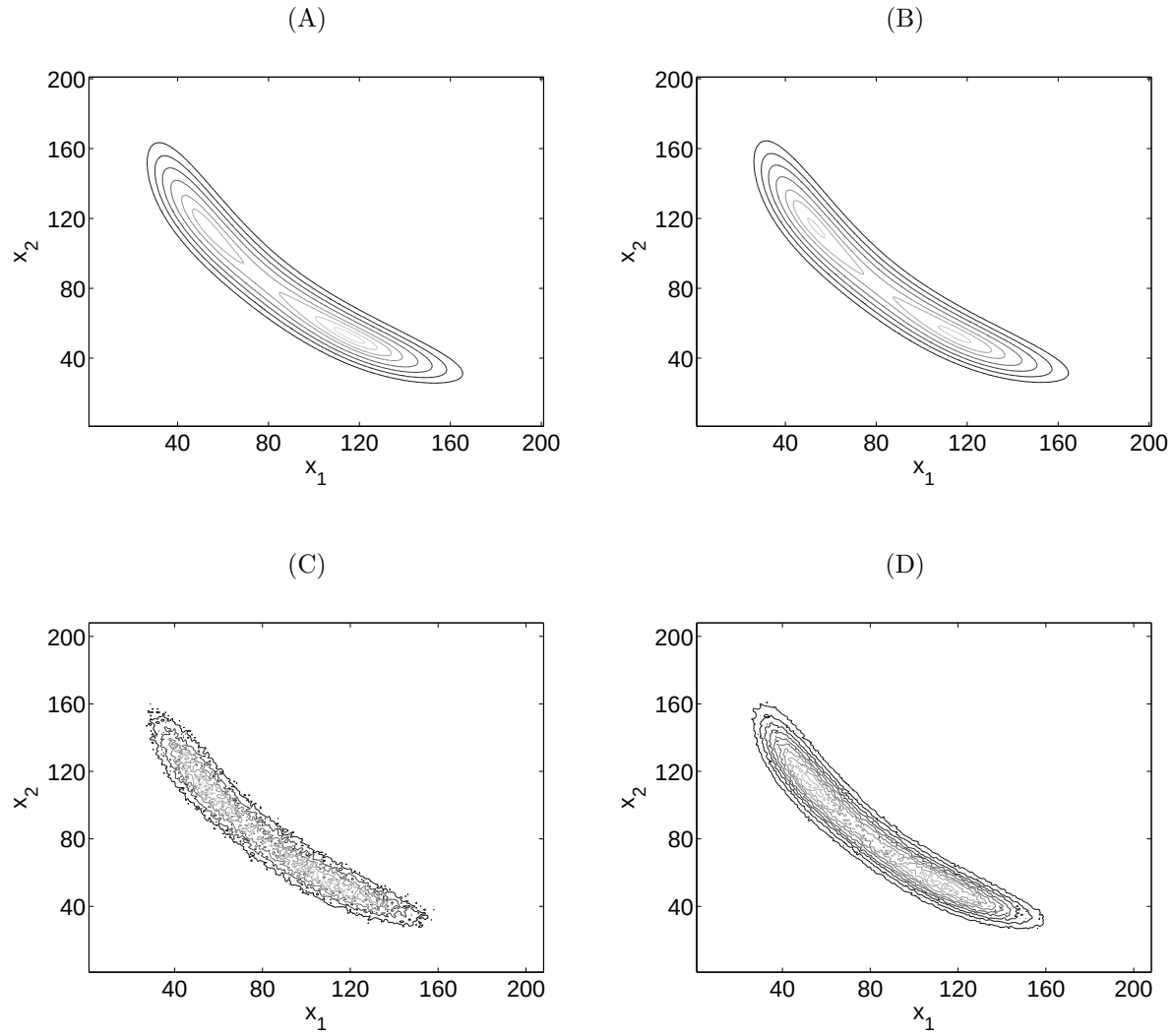


Fig. 5 *Toggle switch computational results. The initial condition is* $(x_1, x_2) = (60, 10)$ *and the final time* t_f *is* $2.0 \times 10^5 s$. *(A) Contour plot for* $\pi_{20}^{(10)}$. *(B) Contour plot for* $\pi_{20}^{(20)}$. *(C) Contour plot based on 100,000 SSA simulations. (D) Contour plot based on 1,000,000 SSA simulations.*

Table 2: CPU time (sec) for different r and different linear system solvers.

r	iterative	direct
5	21.3	1.3
10	31.8	1.6
20	40.5	2.3
40	50.4	3.6
80	59.1	6.1

linearly as r increases. Fig. 6 displays the run time for different r at each step if the iterative solver is used. It shows that the run time decreases almost at the same magnitude step by step for different r; this decrease is expected since the iterates $\pi_r^{(i)}$ are converging to $\pi_r^{(r)}$, so the starting points get better and better.

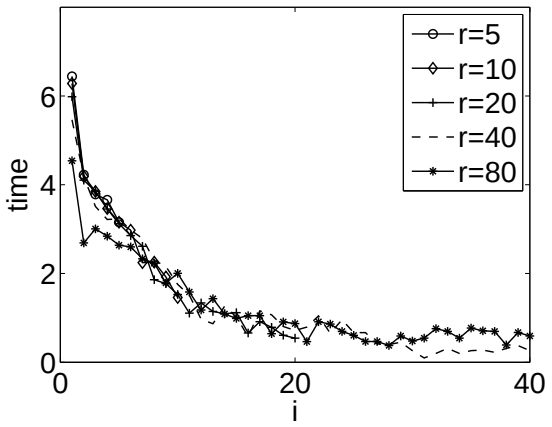


Fig. 6 *CPU time (sec) for different r values at each step of sparse linear system solving.*

4.4 A prototypical cell cycle model

A simple deterministic cell cycle model can be described by the normalized phenomenological rate equations ([33], [34]):

$$\begin{aligned}
\frac{d}{dt}Y_1 &= \kappa_1 m - (\kappa_2' + \kappa_2'' Y_2) Y_1, \\
\frac{d}{dt}Y_2 &= \frac{\kappa_3'' Y_3 (1 - Y_2)}{\Gamma_3 + (1 - Y_2)} - \frac{\kappa_4 Y_1 Y_2}{\Gamma_4 + Y_2}, \\
\frac{d}{dt}Y_3 &= \kappa_5' + \kappa_5'' \frac{Y_1^2}{\Gamma_5^2 + Y_1^2} - \kappa_6 Y_3.
\end{aligned}$$

In the above equations, $Y = (X_{CycB}, X_{Cdh1}, X_{Cdc20})$ and $X_S = [S]/c_S$ is the normalized concentration of species S and c_S is the characteristic concentration of the species. The equations also assume that the normalized concentration of total Cdh1 is 1, so that the concentration of the phosphorylated form can be written as $X_{Cdh1P} = 1 - X_{Cdh1}$. The variable m reflects the fact that $CycB$ is assumed synthesized at a supralinear rate and thus its concentration increases with cell mass. In terms of numbers of molecules $y = (x_{CycB}, x_{Cdh1}, x_{Cdc20})$ of a given species, the equations are:

$$\begin{aligned}
\frac{d}{dt}y_1 &= k_1 m V_s - (k_2' + \frac{k_2''}{V_s} y_2) y_1, \\
\frac{d}{dt}y_2 &= \frac{k_3'' y_3 (c_{Cdh1} V_s - y_2)}{J_3 V_s + (c_{Cdh1} V_s - y_2)} - \frac{k_4 y_1 y_2}{J_4 V_s + y_2}, \\
\frac{d}{dt}y_3 &= k_5' V_s + k_5'' V_s \frac{y_1^2}{(J_5 V_s)^2 + y_1^2} - k_6 y_3,
\end{aligned}$$

where the parameter V_s is equal to the nominal volume of the cell times Avogadro's number and here equals 18 molecules/nMolar. The relationships between the normalized and unnormalized parameters, as well as the values of the normalized parameters, are given in Table 3.

Table 3: Relationships between the normalized and unnormalized parameters and the values of the normalized parameters.

κ_1	k_1/c_{CycB}	$0.01\,\mathrm{min}^{-1}$
κ_2'	k_2'	$0.04\,\mathrm{min}^{-1}$
κ_2''	$k_2''c_{Cdh1}$	$1.0\,\mathrm{min}^{-1}$
κ_3''	$k_3''c_{Cdc20}/c_{Cdh1}$	$10.0\,\mathrm{min}^{-1}$
κ_4	k_4c_{CycB}/c_{Cdh1}	$35\,\mathrm{min}^{-1}$
κ_5'	k_5'/c_{Cdc20}	$0.005\,\mathrm{min}^{-1}$
κ_5''	k_5''/c_{Cdc20}	$0.2\,\mathrm{min}^{-1}$
κ_6	k_6	$0.1\,\mathrm{min}^{-1}$
Γ_3	J_3/c_{Cdh1}	0.04
Γ_4	J_4/c_{Cdh1}	0.04
Γ_5	J_5/c_{CycB}	0.3

To produce an accurate stochastic model of this system, unpack it into elementary chemical reactions without intermediates and with variable propensities.

$$\left.\begin{array}{rcl}
\emptyset & \overset{k_1 m V_s}{\underset{k_2'}{\rightleftharpoons}} & x_{CycB} \\
x_{CycB}+x_{Cdh1} & \overset{k_2''/V_s}{\longrightarrow} & x_{Cdh1} \\
x_{CycB}+x_{Cdh1} & \overset{\frac{k_4}{J_4V_s+x_{Cdh1}}}{\longrightarrow} & x_{CycB}+x_{Cdh1P} \\
x_{Cdc20}+x_{Cdh1P} & \overset{\frac{k_3''}{J_3V_s+x_{Cdh1P}}}{\longrightarrow} & x_{Cdc20}+x_{Cdh1} \\
\emptyset & \overset{k_5'V_s+k_5''V_s\frac{x_{CycB}^2}{(J_5V_s)^2+x_{CycB}^2}}{\underset{k_6}{\rightleftharpoons}} & x_{Cdc20}
\end{array}\right\}.$$

The propensity functions are

$$\begin{aligned}
a_1(x) &= k_1 m V_s, \\
a_2(x) &= k_2' x_{CycB}, \\
a_3(x) &= \frac{k_2''}{V_s} x_{CycB} x_{Cdh1}, \\
a_4(x) &= \frac{k_4 x_{CycB} x_{Cdh1}}{J_4 V_s + x_{Cdh1}}, \\
a_5(x) &= \frac{k_3'' x_{Cdc20}(c_{Cdh1}V_s - x_{Cdh1})}{J_3 V_s + (c_{Cdh1}V_s - x_{Cdh1})}, \\
a_6(x) &= k_5' V_s + k_5'' V_s \frac{x_{CycB}^2}{(J_5 V_s)^2 + x_{CycB}^2}, \\
a_7(x) &= k_6 x_{Cdc20}.
\end{aligned}$$

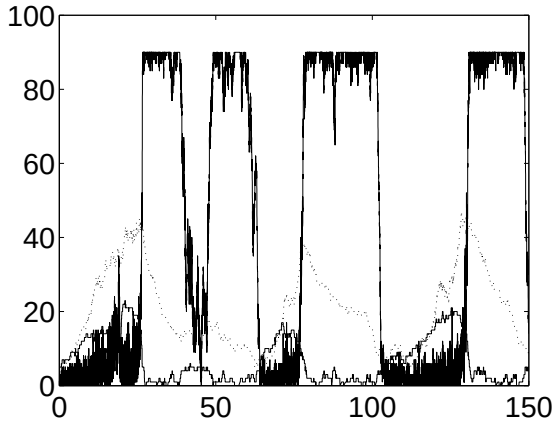


Fig. 7 *One trajectory for the cell cycle model ($t_f = 150\,\mathrm{min}$). CycB, grey line. Cdh1, solid line. Cdc20, dotted line.*

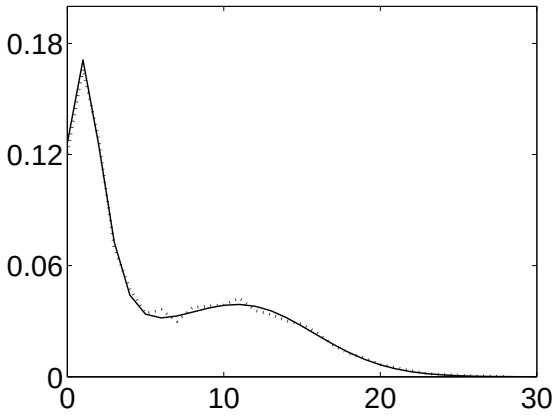
In the numerical experiments the characteristic concentrations $c_{CycB} = 5.0nM$, $c_{Cdh1} = 5.0nM$, $c_{Cdc20} = 5.0nM$. The system is initialized with $x = (x_{CycB}, x_{Cdh1}, x_{Cdc20}) = (5, 5, 5)$ and $m = 1.5$. Fig. 7 and Fig. 8 display the numerical results, while Table 4 and Table 5 list the computational costs. The numerical results from the proposed method match the histograms obtained from SSA simulations and are smoother. The result $\pi_{10}^{(10)}$ has accuracy comparable to the Monte Carlo methods but at about half the cost.

5. Conclusions and Future Work

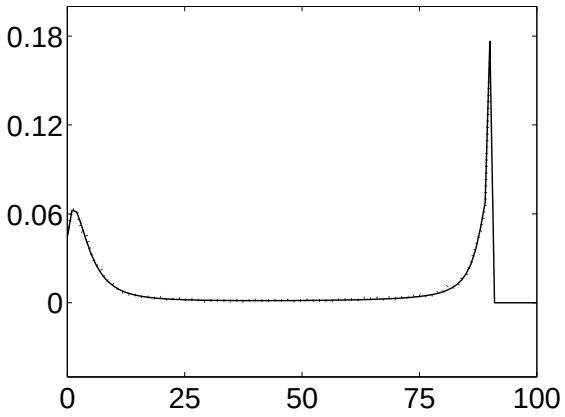
In the theory of Markov processes, determining the value of the state probability vector at any time before the system reaches the stable state is called transient analysis. In contrast to steady state analysis, transient analysis requires solving linear differential equations instead of linear algebraic equations, which makes transient analysis much more difficult. Nevertheless, many methods have been proposed for transient analysis, based on traditional ODE solvers, the exponential of a matrix, Laplace transforms, Krylov subspaces, and uniformization ([17], [35]). The uniformization method was proposed by Jensen and has become very popular in the last twenty years. This paper has investigated a variant of the standard uniformization method, called the external uniformization method. Numerical results here show that for a number of problems, especially problems with two or three species, the external uniformization method is numerically efficient and accurate.

One important feature of (external) uniformization method is its simplicity. Only an efficient sparse linear system solver is needed. In the examples here, direct linear system solvers outperform iterative solvers, but this may not be the case for higher dimension problems or problems with more irregular structures. Generally, the performance of iterative linear solvers depends crucially on preconditioning, but that has not been explored here.

(A)



(B)



(C)

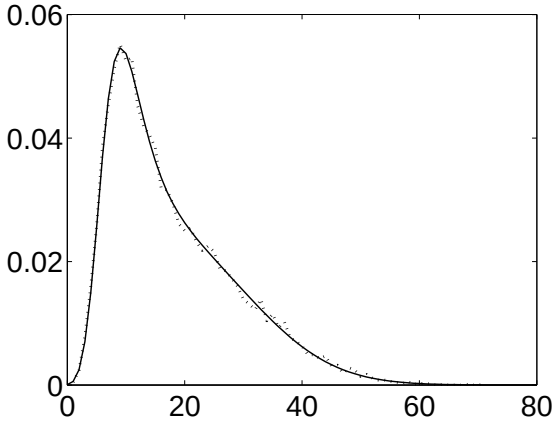


Fig. 8 *Cell cycle computational results. Comparisons of estimations for marginal probability distributions. The histograms (dotted lines) are based on 10,000 SSA simulations. The solid lines are based on* $\pi_{10}^{(10)}$ *computed from the proposed method. The final time is* $1000\,\mathrm{min}$. *(A) CycB. (B) Cdh1. (C) Cdc20.*

Table 4: CPU time (sec) for different Monte Carlo algorithms (StochKit). The error control epsilon in the adaptive τ-leaping method equals 0.03.

	10^3 runs	10^4 runs
SSA	240.9	2400.7
Adaptive τ-leaping	246.8	2464.0

Table 5: CPU time (sec) for different r using the direct linear system solver (UMFPACK). The run time is decomposed as factorization + triangular system solve + other.

$r = 10 : 1203.0 = 1078.8 + (7.2 + 9 \times 11.4) + 14.4$

$r = 20 : 1271.9 = 1009.0 + (7.9 + 19 \times 12.7) + 13.7$

$r = 40 : 1526.2 = 1008.7 + (7.9 + 39 \times 12.7) + 14.3$

Note that $\pi_r^{(r)}$ is approximately a left eigenvector of $I - \frac{1}{\lambda}Q$ corresponding to the dominant eigenvalue one, hence an Arnoldi method [36] may be a viable way to approximate $\pi_r^{(r)}$ for large r. Eigentheory for large sparse matrices has also not been considered here.

Like other direct methods for the chemical master equations, the major computational challenge for the external uniformization method comes from the curse of dimensionality. Combining the method with model reduction techniques, like sparse grid approximation for the state space, and novel techniques for efficiently computing $\pi_r^{(r)}$, holds promise. The next step is to attempt to solve the master equation for a state of the art cell cycle model such as the 46 ODE budding yeast model [37].

REFERENCES

[1] van Kampen, N. G.: 'Stochastic Processes in Physics and Chemistry, 5th edition' (Elsevier, Amsterdam, 2004)

[2] Gillespie, D. T.: 'A rigorous derivation of the chemical master equation', *Physica A*, 1992, **188**, pp. 404–425

[3] Gillespie, D. T.: 'Exact stochastic simulation of coupled chemical reactions', *J. Phys. Chem.*, 1977, **81**, pp. 2340–2360

[4] Gibson, M. A., Bruck, J.: 'Efficient exact stochastic simulation of chemical systems with many species and many channels', *J. Phys. Chem.*, 2000, **104**, pp. 1876–1889

[5] Gillespie, D. T.: 'Approximate accelerated stochastic simulation of chemically reacting systems', *J. Chem. Phys.*, 2001, **115**, pp. 1716–1733

[6] Cao, Y., Gillespie, D. T., and Petzold, L. R.: 'The slow-scale stochastic simulation algorithm', *J. Chem. Phys.*, 2005, **122**, 014116

[7] Cao, Y., Gillespie, D. T., and Petzold, L. R.: 'Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems', *J. Comput. Phys.*, 2005, **206**, pp. 395–411

[8] Rao, C. V., Arkin, A. P.: 'Stochastic chemical kinetics and quasi-steady-state assumption: Application to the Gillespie algorithm', *J. Chem. Phys.*, 2003, **118**, pp. 4999–5010

[9] Hegland, M., Burden, C., Santoso, L., MacNamara, S., and Booth H.: 'A solver for the stochastic master equation applied to gene regulatory networks', *J. Comput. Appl. Math.*, 2005, **205**, pp. 708–724

[10] Munsky, B., Khammash, M.: 'The finite state projection algorithm for the solution of the chemical master equation', *J. Chem. Phys.*, 2006, **124**, 044104

[11] Engblom, S., 'A discrete spectral method for the chemical master equation', Technical Report 2006-036, Dept. of Information Technology, Uppsala University

[12] Sjöberg, P., Lötstedt, P., and Elf, J.: 'Fokker Planck approximation of the master equation in molecular biology', *Computing and Visualization in Science*, 2007

[13] Elf, J., Lötstedt, P., and Sjöberg, P.: 'Problems of high dimension in molecular biology'. Proc. 19th GAMM-SeminarLeipzig, 2003

[14] Moler, C., Van Loan, C. F.: 'Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later', *SIAM Rev.*, 2003, **45**, pp. 3–49

[15] Jensen, A.: 'Markoff chains as an aid in the study of Markoff processes', *Skandinavisk Aktuarietidskrift*, 1953, **36**, pp. 87–91

[16] Stewart, W. J.: 'Numerical Solution of Markov Chains' (Marcel Dekker, New York, 1991)

[17] Grassmann, W. K.: 'Computational Probability, 1st edition' (Springer, 1999)

[18] Ross, S. M.: 'Approximating transition probabilities and mean occupation times in continuous-time Markov chains', *Prob. Eng. Inform. Sci.*, 1987, **1**, pp. 251–264

[19] Yoon, B. S., Shanthikumar, J. G.: 'Bounds and approximations for the transient behavior of continuous-time Markov chains', *Prob. Eng. Inform. Sci.*, 1989, **3**, pp. 175–198

[20] Carmo, R., de Souza e Silva, E., and Marie, R., 'Efficient solutions for an approximation technique for the transient analysis of Markovian models', Technical Report, IRISA, no. 3055, Campus universitaire de Beaulieu

[21] Gardner, T. S., Cantor, C. R., and Collins, J. J.: 'Construction of a genetic toggle switch in Escherichia coli', *Nature*, 2000, **403**, pp. 339–342

[22] Sidje, R. B.: 'Expokit: A software package for computing matrix exponentials', *ACM Trans. Math. Soft.*, 1998, **24**, pp. 130–156

[23] Li, H., Cao, Y., Petzold, L., and Gillespie, D.: 'Algorithms and software for stochastic simulation of biochemical reacting systems', *Biotech. Prog.*, to appear

[24] Reibman, A., Trivedi, K.: 'Numerical transient analysis of Markov models', *Computers and Operations Research*, 1988, **15**, pp. 19–36

[25] Angus, J. E.: 'Some bounds on the error in approximating transition probabilities in continuous-time Markov process', *SIAM R.*, 1992, **34**, pp. 110–113

[26] Saad, Y.: 'Iterative methods for sparse linear systems, 2nd edition' (SIAM, Philadelphia, PA, 2003)

[27] Saad, Y., 'Sparskit: a basic tool kit for sparse matrix computation, version 2', Technical Report, Computer Science Department, Univ. of Minnesota, Minneapolis, MN, 1994

[28] Davis, T. A.: 'Direct Methods for Sparse Linear Systems' (SIAM, Philadelphia, 2006)

[29] Davis, T. A., 'UMFPACK version 5.0 user guide', Technical Report, TR-04-003, Dept. of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 2004

[30] Jahnke, T., Huisinga, W.: 'Solving the chemical master equation for monomolecular reaction systems analytically', *J. Math. Biol.*, 2007, **54**, pp. 1–26

[31] Stoer, J., Bulirsch, R.: 'Introduction to Numerical Analysis, 2nd edition' (Springer-Verlag, New York, 1993)

[32] Gillespie, D. T.: 'Markov Processes: An Introduction for Physical Scientists' (Academic Press, Boston, 1992)

[33] Zhang, J., Watson, L.: 'A modified uniformization method for the chemical master equation'. to appear in Proc. IEEE 7th International Symp. on Bioinformatics and Bioengineering, Boston, USA, October 2007

[34] Tyson, J. J., Novak, B.: 'Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions', *J. Theo. Biol.*, 2001, **210**, pp. 249–263

[35] Sidje, R. B., Stewart, W. J., 'A survey of methods for computing large sparse matrix exponentials arising in Markov chains', Technical Report, TR-96-06, Computer Science Department, North Carolina State University

[36] Lehoucq, R. B., Sorensen D. C., and Yang, C.: 'ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods' (SIAM, Philadelphia, 1998)

[37] Panning, T.D., Watson, L.T., Allen, N.A., Chen, K.C., Shaffer, C.A., and Tyson, J.J.: 'Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle', *J. Global Opt.*, to appear