Adaptive Aggregation Method for the Chemical Master Equation

Jingwei Zhang, Layne T. Watson, and Yang Cao

***Abstract*—The chemical master equation, which is often considered as an accurate stochastic description of general chemical systems, usually imposes intensive computational requirements when used to characterize molecular biological systems. The major challenge comes from the curse of dimensionally, which has been tackled by a few research papers. The essential goal is to aggregate the system efficiently with limited approximation error. This paper presents an adaptive way to implement the aggregation process using information collected from Monte Carlo methods. Numerical results show the effectiveness of the proposed algorithm despite the lack of explicit estimation of approximation error.**

I. Introduction

Efficient analysis tools for chemical systems have always been of great interest to the systems biology community. Traditional deterministic modeling like reaction rate equations often fails to capture the inherent (molecular level) randomness of the biological system. On the other hand, more accurate stochastic models often impose intensive computational requirements. Algorithms for solving stochastic models for chemical reaction systems can be roughly divided into two categories. One approach is to simulate the system using Monte Carlo methods, the most famous one of which is the stochastic state simulation algorithm (SSA) [1]. These methods simulate one possible trajectory at a time at a relatively low computational cost. However, many trajectories need to be simulated to get an accurate estimation of statistical parameters. The second approach is to solve the chemical master equation ([2],[3]), which is basically an ordinary differential equation (ODE) system for the probability distribution function describing the evolution of the chemical system. Every possible state is taken into account in the distribution function, thus the size of the ODE system is often huge.

Suppose a chemical system consists of D different species and R reaction channels. If the size of the state space of each species S_i is N_i, then the total state space size $N = N_1 \times N_2 \times \cdots \times N_D$. As the number of species increases, the total state space size increases exponentially, known as the curse of dimensionality. Let $p(x,t)$ denote the probability mass of the state x at time t, where $x = (x_1, x_2, \ldots, x_D) \in \mathbf{Z}_{N_1} \times \mathbf{Z}_{N_2} \times \cdots \times \mathbf{Z}_{N_D}$ is a vector of integers representing one possible state of the system. It is worth noticing that there are cases where the number of molecules of a certain species S_i does not lie in $[0, N_i - 1]$ (i.e., the integers 0, 1, ..., $N_i - 1$ represent an encoding of the actual number of molecules) and/or the whole state space is not rectangular or even finite. However, most of the discussion here is still valid under those problem settings.

The chemical master equation (CME), which is derived from the Markov property of the underlying stochastic process ([4]–[7]), is the ODE system describing the time evolution of the function $p(x,t)$ for every possible state x,

$$\frac{dP(X,t)}{dt} = P(X,t)Q,$$

where $P(X,t) = \big(p(x^{(1)},t), p(x^{(2)},t), \ldots\big)$ is the complete probability vector at time t and the vector $X = \big(x^{(1)}, x^{(2)}, \ldots\big)$ is a particular enumeration of the state space. Here Q is a constant sparse square matrix (called the infinitesimal generator of the system) with each of its components q_{ij} $(i \neq j)$ denoting the instantaneous rate at which the system makes one transition from one state $x^{(i)}$ to another state $x^{(j)}$ through one of the R prescribed reaction channels. Each row of the matrix Q sums up to zero, i.e., $q_{ii} = -\sum_{j \neq i} q_{ij}$, so that the whole system satisfies the conservation law of the overall probability mass. Now, each row of Q has at most $R+1$ nonzero entries, hence the matrix Q is extremely sparse.

For example, the CME for a simple birth-death process

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

can be written as

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

Various approximation methods have been proposed to reduce the size of the matrix Q. Consider aggregation/disaggregation operators E and F. The aggregation operator E maps any complete probability vector $P(X,t)$ to an aggregated probability vector $\tilde{P}(Y,t)$ and the disaggregation operator F does the opposite. In the discrete case, the operators E and F are just matrices. Now, the original ODE system can be condensed into a much

Manuscript received June 14, 2008. This work was supported in part by the U.S. Department of Energy under Grant DE-FG02-06ER25720

Jingwei Zhang is with the Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA jwzhang@vt.edu

Layne T. Watson is with the Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA ltw@cs.vt.edu

Yang Cao is with the Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA ycao@cs.vt.edu

This work was supported by Department of Energy Grant DE-FG02-06ER25720.

smaller ODE system corresponding to the aggregated state space Y,

$$\frac{d\tilde{P}(Y,t)}{dt} = \tilde{P}(Y,t)FQE.$$

Note that this is essentially the model order reduction problem of control theory or mechanics. There are many different ways to choose appropriate E and F, determined by the measure of the distance from $P(X,t)$ to $\tilde{P}(Y,t)$. One idea, called the finite state projection algorithm ([8]–[10]), is to choose E and F such that FQE is a submatrix of the original matrix Q. Let $J \subset \{1,2,\ldots,N\}$, and denote by X_J the subvector of X formed from the indices in J, and by Q_{JJ} the submatrix of Q with rows and columns indexed by J. Let $Y = (y^{(1)}, y^{(2)}, \ldots) = X_J$ denote the finite vector of states of specific interest, and the matrix Q_{JJ} be the submatrix of the matrix Q corresponding to the vector Y. Then with $\tilde{Q} = Q_{JJ}$, $\tilde{P} = P_J$, and $Y = X_J$, the condensed system to be solved is

$$\frac{d\tilde{P}(Y,t)}{dt} = \tilde{P}(Y,t)\tilde{Q}.$$

Since $\tilde{Q}$ is only part of the matrix Q, the overall probability mass for the new system no longer satisfies the conservation law. From a simulation point of view, in this new system any trajectory that reaches outside of the states Y before time t is lost forever. That is the major drawback of the finite state projection method. However, if the size of the original state space is infinite, this is the only way to reduce the original problem to a finite state problem.

Another idea is to first divide the state space into bins using grids, then let the aggregation operator E map states in the same bin into a single state in the reduced system [11]. The probability mass of each single (reduced) state thus equals the sum of the probability masses of all those states mapped into it. One easy way to choose the corresponding disaggregation matrix F is to divide the probability mass of each single (reduced) state evenly into parts and assign this value as the probability mass of every state in the same bin that maps to that single (reduced) state by operator E.

In most chemical systems, it often happens that the probability masses of nearby states are very close to each other, therefore it is reasonable to combine these nearby states together to reduce the size of the problem. A more plausible way would be to apply the above reduction only to the part of the state space where probability mass is low and remains almost constant over time, not to the part of the state space where probability mass is high and changes significantly. In practice, collections of simulation results may be used to determine how to choose aggregation operators this way, leading to the adaptive aggregation method proposed here.

II. Adaptive aggregation method

In aggregation methods, aggregated grids and their aggregation/disaggregation operators E and F may be determined statically or dynamically. In the static case, the grids are determined at the beginning of the computation process and never changed after that, while in the dynamic case the grids adjust to the dynamics of the computation. Dynamic gridding makes more sense when the computation domain that matters most is much smaller than the whole state space and changes over time.

When solving the CME, dynamic gridding for the aggregation technique means coarse grids for the states with low probabilities and fine grids for the states with high probabilities. One simple way to distinguish these two groups of states is by Monte Carlo methods such as SSA. The whole state space is first divided into a suitable number of bins. Let $t_0 = 0$ be the initial time, $t_n = t_f$ the final time, and check times $t_0, t_1, t_2, \ldots, t_n$ equally spaced in the time interval $[t_0, t_f]$. Simulate the dynamics of the chemical system from t_0 through t_n several times using Monte Carlo methods like SSA, and on each time interval $[t_i, t_{i+1}]$ $(0 \le i \le n-1)$ record all the bins that have been touched by at least one trajectory between time t_i and t_{i+1}. Now, all the states within these recorded bins are categorized as states with likely high probability masses, whereas all other states are categorized as states with likely low probability masses. When integrating the original ODE system from time t_i through time t_{i+1}, choose the aggregation operator E such that all the states that have been marked as states with likely high probability masses remain the same after the mapping, and all other states, namely states with likely low probability masses, are aggregated into reduced states according to the aggregation grids.

Here are two numerical examples on the simple birth-death process with $k = 1.2s^{-1}$, $\mu = 0.01s^{-1}$. Figure 1 shows the numerical result when the reduced model is constructed using information from 10 SSA simulations, all started from the same initial state, $x_1(0) = 10$. The aggregation matrix is then formed based on these 20 trajectories shown in Figure 1 (A). For example, on the time (seconds) interval $[60, 80]$, every state in the closed interval $[50, 79]$ is not aggregated, while all other states are aggregated according to the grids on the state space. Hence, the size of the state space after aggregation is $3 \times 10 + (15 - 3) = 42$, much less than the size of the original state space, 15×10. Figure 2 shows the fact that increasing the number of SSA runs can improve the approximation accuracy of the reduced model at the expense of additional computational cost. For instance, in this case with 20 SSA simulations every state in the closed interval $[40, 99]$ is not aggregated on the same time interval $[60, 80]$. Thus, the number of states after aggregation is now $6 \times 10 + (15 - 6) = 69$, larger than the previous case as expected, however, the numerical

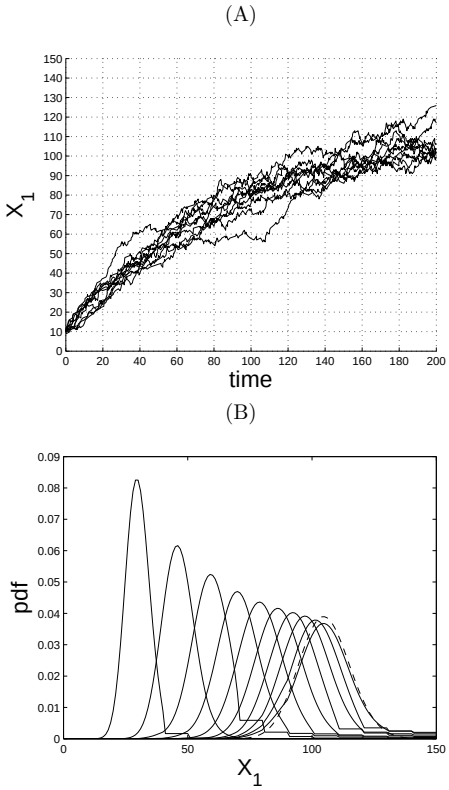

Figure 1. (A) Ten SSA simulations of the simple birth-death process (of species X_1) with $t_f = 200$. The grids on the graph divide the state space into bins and the time space into subintervals. (B) Computational results from the full model and the reduced model. The sequence of solid lines shows the solutions to the reduced model at each time $t_i = 20i$, $1 \le i \le 10$. The dashed line, which is very close to the solid line at the time t_{10}, is the solution to the full model at the final time $t_f = t_{10}$.

solutions in Figure 2 (B) are smoother and closer to the actual solutions than in Figure 1 (B).

The only constraint for the corresponding disaggregation matrices F is that the aggregation/disaggregation matrix pair E and F should satisfy the constraint that $FE = I$, where I is a square identity matrix, like

$$F = \begin{pmatrix} \frac{1}{k} & \cdots & \frac{1}{k} & & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & & & & & \frac{1}{k} & \cdots & \frac{1}{k} \end{pmatrix}$$

and

$$E^t = \begin{pmatrix} 1 & \cdots & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & \cdots & 1 \end{pmatrix},$$

where the integer k is the bin size. Take $P = (p_1, p_2, \ldots, p_{3k})$ as any complete probability vector. Then $PE = (\sum_{i=1}^{k} p_i, p_{k+1}, p_{k+2}, \ldots, p_{2k}, \sum_{i=2k+1}^{3k} p_i) = \tilde{P} = (\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_{k+2})$ produces the reduced probability vector $\tilde{P}$ of vector P, and $\tilde{P}F = (\frac{1}{k}\tilde{p}_1, \ldots, \frac{1}{k}\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_{k+1}, \frac{1}{k}\tilde{p}_{k+2}, \ldots, \frac{1}{k}\tilde{p}_{k+2})$ maps the reduced probability vector $\tilde{P}$ back to the complete probability vector space by dividing $\tilde{p}_1$ and $\tilde{p}_{k+2}$ into k parts.

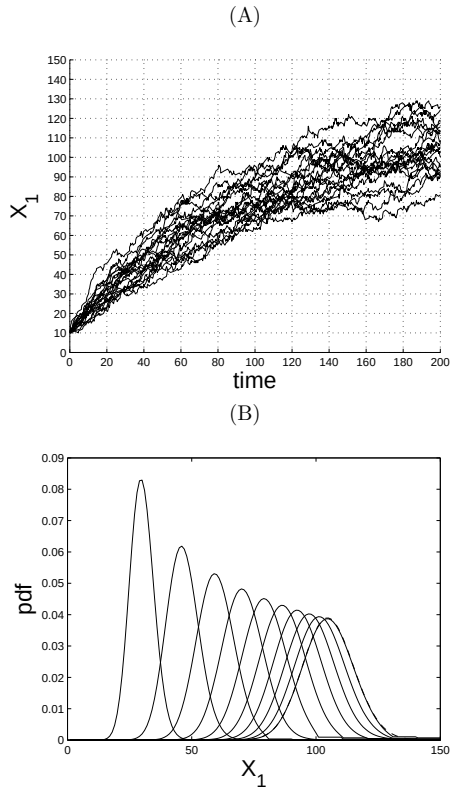

Figure 2. (A) Twenty SSA simulations of the simple birth-death process with $t_f = 200$. (B) Computational results from the full model and the reduced model. The sequence of solid lines shows the solutions to the reduced model at each time $t_i = 20i$, $1 \le i \le 10$. The dashed line, which almost coincides with the solid line at the time t_{10}, is the solution to the full model at the final time $t_f = t_{10}$.

Assume that the state space is divided evenly into M bins and the size of each bin is k, that is, $N = Mk$. Furthermore, assume the number of bins that has been reached by all the trajectories on each time interval is relatively small compared to M. Then the size of the reduced model is $O(k) + O(M)$, which reaches its minimum when M and k are approximately balanced, $M = \Theta(k)$, and so the minimum obtained is $O(\sqrt{N})$. This implies that the size of the state space would be reduced to its square root at best, and thus unequal size bins are required for any further reduction.

To generalize the adaptive aggregation method to higher dimensions, appropriate aggregation grids are needed. The easiest way to construct such grids is to use Cartesian products as used in the numerical examples in Section 3. However, simple Cartesian product grids may still suffer from the curse of dimensionality, which often limits the computation to three or two dimensional

problems. An alternative is to use the sparse grid approximation [11], which reduces the number of bins substantially while the error of the approximation is closely bounded.

III. Numerical results

3.1. Toggle reaction

One problem with the finite state projection algorithm is its inefficiency when dealing with bistable distributions. When the probability distribution is bistable, the best reduced model should only focus on the two stable states. However, if all the states between these two stable states are wiped out completely as could happen with the finite state projection algorithm, then the innate bistable nature of the model is lost. In the proposed adaptive aggregation method, however, this will not happen, since all the states with small probabilities are not simply thrown away but aggregated according to the coarse grids so that the bistable property of the original system is preserved. This speculation can be verified from numerical experiments on the well-known bistable toggle switch reaction [14],

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

with parameters $\alpha = \gamma = 1000\,\mathrm{min}^{-1}, \beta = \delta = 6000$ and $\mu = 10^{-3}\,\mathrm{min}^{-1}$.

Figure 3 shows how the proposed algorithm works on this specific problem starting from $x_1(0) = 10,\ x_2(0) = 60$, on the time (minutes) interval $[t_0, t_f] = [0, 10000]$. The whole time interval is divided evenly into twenty subintervals by grid points $t_i = 500i,\ 0 \le i \le 20$. Figure 3 shows the computation result (contour plots of the aggregated probability vector $\tilde{P}(Y,t)$) together with the projected high probability domain, i.e., the part of the state space categorized as states with likely high probability so that no grid aggregation is applied. Altogether, the three graphs demonstrate how the aggregation algorithm is applied adaptively and efficiently for the toggle switch model.

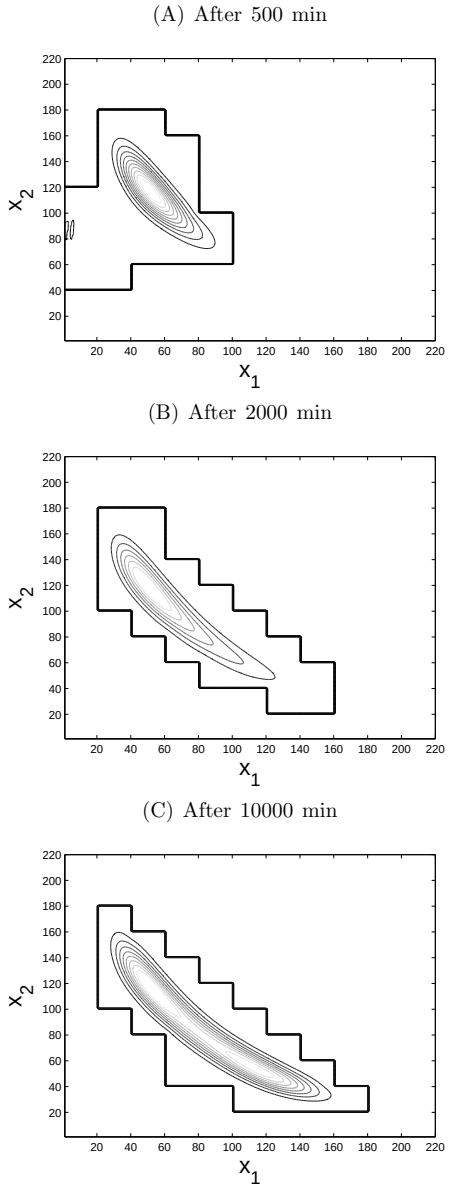

Figure 3. The evolution of the projected high probability domain and the numerical solution $\tilde{P}(Y,t)$ when using the adaptive aggregation method on the toggle switch problem.

Figure 4 compares the estimated marginal probability distributions of species X_1 and X_2 from three different methods. The full model CME can be considered as an exact description, while the accuracy of the SSA algorithm and the reduced model CME are both proportional to the cost. The comparison of numerical results are based on comparably accurate SSA and reduced model CME as drawn in Figure 4. The CPU runtime for 10^5 SSA simulations is $2688.3s$ (using the SSA algorithm implemented in Stochkit [15]), and the runtimes for the full model and reduced model CME are $73.3s$ and $39.7s$ respectively (the ODE systems are solved by the Krylov space projection method [16], implemented in SPARSKIT [17]). The runtime reduction for the reduced model is not as great as might have been expected since the size of the reduced model is around one fourth that of the full model and the proposed algorithm also introduces overhead computing the matrices E and F and matrix vector multiplications using E and F. However, the reduction in computational effort (with a Cartesian product grid) will become significant if the problem size is large enough as shown in the next example.

3.2. A simple cell cycle model

This section focuses on a simple cell cycle model derived from some normalized phenomenological rate equations [18]. The model can be described by the following elementary reactions without intermediates and with variable propensity rates and parameters $k_1/c_{CycB} = 0.01\,\mathrm{min}^{-1}$, $k_2' = 0.04\,\mathrm{min}^{-1}$, $k_2'' c_{Cdh1} = 1.0\,\mathrm{min}^{-1}$, $k_3'' c_{Cdc20}/c_{Cdh1} = 10.0\,\mathrm{min}^{-1}$, $k_4 c_{CycB}/c_{Cdh1} = 35\,\mathrm{min}^{-1}$, $k_5'/c_{Cdc20} = 0.005\,\mathrm{min}^{-1}$,

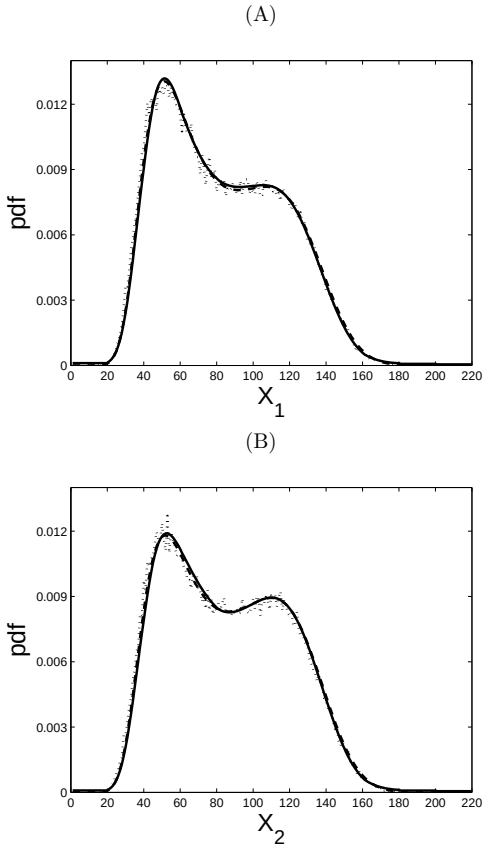

Figure 4. Numerical estimations for marginal probability distributions using the full model (dashed lines), the reduced model (solid lines), histograms from 10^5 SSA simulation runs (dotted lines).

$k_5''/c_{Cdc20} = 0.2\,\mathrm{min}^{-1}$, $k_6 = 0.1\,\mathrm{min}^{-1}$, $J_3/c_{Cdh1} = 0.04$, $J_4/c_{Cdh1} = 0.04$, $J_5/c_{CycB} = 0.3$:

$$\left.\begin{array}{rcl}
\emptyset & \underset{k_2'}{\overset{k_1 m V_s}{\rightleftharpoons}} & x_{CycB} \\
x_{CycB} + x_{Cdh1} & \overset{k_2''/V_s}{\longrightarrow} & x_{Cdh1} \\
x_{CycB} + x_{Cdh1} & \overset{\frac{k_4}{J_4 V_s + x_{Cdh1}}}{\longrightarrow} & x_{CycB} + x_{Cdh1P} \\
x_{Cdc20} + x_{Cdh1P} & \overset{\frac{k_3''}{J_3 V_s + x_{Cdh1P}}}{\longrightarrow} & x_{Cdc20} + x_{Cdh1} \\
\emptyset & \underset{k_6}{\overset{k_5' V_s + k_5'' V_s \frac{x_{CycB}^2}{(J_5 V_s)^2 + x_{CycB}^2}}{\rightleftharpoons}} & x_{Cdc20}
\end{array}\right\},$$

where c_S is the characteristic concentration of the species S. The cell mass variable m is introduced to reflect the assumption that the species $CycB$ is synthesized at a supralinear rate such that its molecular number increases with cell mess and V_s, the nominal volume of the cell times Avogadro's number, is chosen to be 18 molecules/nMolar. The species $Cdh1P$ is the phosphorylated form of $Cdh1$ so that $x_{Cdh1} + x_{Cdh1P} = c_{Cdh1} V_s$.

Numerical experiments are conducted here under three different settings of characteristic concentrations, $c_{CycB} = c_{Cdh1} = c_{Cdc20} = 5.0nM$, $c_{CycB} = c_{Cdh1} = c_{Cdc20} = 10.0nM$, $c_{CycB} = c_{Cdh1} = c_{Cdc20} = 20.0nM$, and the final time (minutes) is $t_f = 70.0$. Commonly, the larger the characteristic concentration, the larger is the size of the state space, the bin size in Cartesian product gridding, and the number of SSA runs to construct an accurate reduced model. Accordingly, the bin sizes for all three cases are $6 \times 13 \times 8$, $8 \times 14 \times 12$, and $10 \times 19 \times 15$, and the numbers of SSA runs are 20, 50, and 100. Figure 5 compares the numerical results from different numerical schemes under the first setting and Table 1 compares the CPU times (in seconds) under all three settings.

Since the problem size is relatively large, two types of ODE system integrators for the CME are tested here [19]. The first one is the Krylov space method, which only requires procedures to compute matrix vector products and no storage of the whole matrix. The second one is a numerically stable ODE solver (the backward Euler method), considering the prevalence of stiff problems in chemical kinetics. Numerical results show that when the problem size is moderate, stable ODE solvers usually require less CPU time than the Krylov space methods. However, most stable ODE solvers also need efficient sparse matrix linear system solvers, which often demand extra computer memory depending on the implementation.

TABLE 1. CPU time (in seconds) for different methods.

c_S	5.0	10.0	20.0
SSA*	179.3	3.5×10^3	7.3×10^4
CME(full)	1646.0	6.0×10^4	7.1×10^5
CME(reduced)	342.5	4.3×10^3	2.7×10^4
CME(reduced)**	194.9	2.5×10^3	2.8×10^4

* 10^4 SSA runs when characteristic concentrations c_S are all $5.0nM$, 10^5 runs for $10.0nM$, and 10^6 runs for $20.0nM$.
** The CME is solved by the backward Euler method with a fixed step size of 0.07 instead of the Krylov space projection method, and the matrix linear systems are solved by the software package UMFPACK (the unsymmetric multifrontal method for sparse LU factorization) in SuiteSparse ([20], [21]).

Table 1 shows that the CPU times of the algorithm proposed are comparable to the Monte Carlo method SSA and much less than solving the CME on the full state space. Actually, the reduction in computational effort becomes more substantial as the problem size grows.

IV. Conclusion

One major challenge when using the chemical master equation to model gene regulatory networks and some other biological systems is that it often requires integrating some very large ODE systems to get solutions to the CME. This paper proposes a possible way of using information from approaches like Monte Carlo methods to diminish the size of the ODE systems imposed by the CME, as shown in the numerical examples. One potential problem with the proposed algorithm is the difficulty estimating the model reduction error. Empirically, this error depends on the number of simulation runs used to construct the reduced model, as shown in the simple birth-death example.

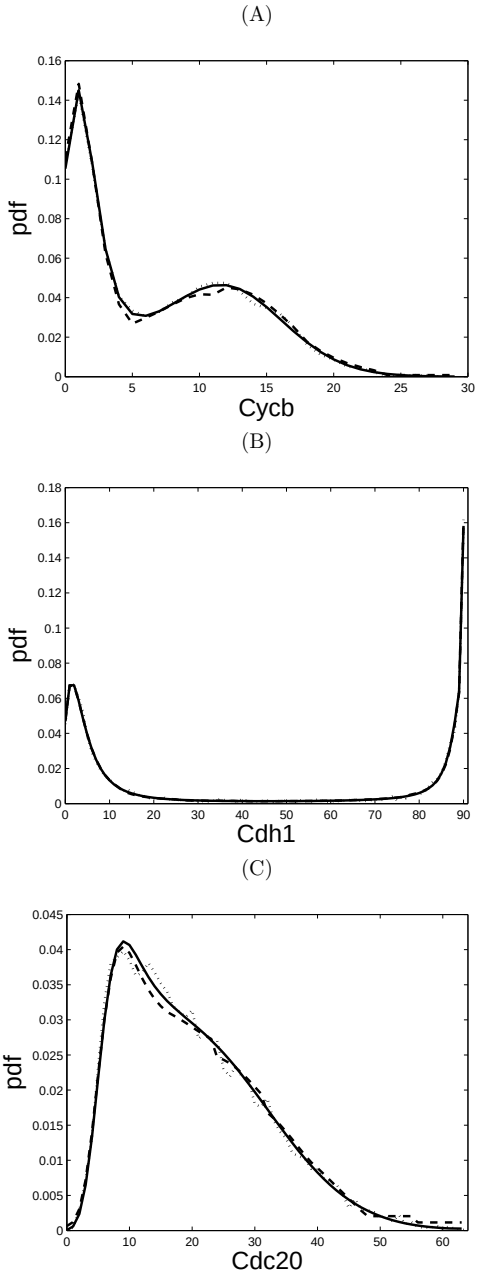

Figure 5. Numerical estimations for marginal probability distributions at time (minutes) $t_f = 70.0$, $c_{CycB} = 5.0nM$, $c_{Cdh1} = 5.0nM$, $c_{Cdc20} = 5.0nM$, the full CME model (solid lines), the Cartesian grid reduced CME model (dashed lines), and the histogram from 10^4 SSA simulations (dotted lines).

The proposed algorithm works best when the statistical variance of the system is relatively small and not so well if the variance is high, which often occurs when the chemical system is bistable. However, for most of the numerical examples presented here, where the variance of the system is relatively high, the performance of the proposed algorithm is satisfactory.

Information collected from Monte Carlo methods is used in a quite simple way in the proposed algorithm. A more complicated and effective way may be to first construct a surrogate model [22] out of data from Monte Carlo methods and reduce the size of the ODE system based on the surrogate model, instead of using data from Monte Carlo simulations directly. In other words, the grids are generated with different mesh sizes according to the estimated probability mass function using its value and its spatial derivatives.

References

[1] Gillespie, D. T., "Exact stochastic simulation of coupled chemical reactions," *J. Phys. Chem.*, 1977, **81**, pp. 2340–2360.

[2] van Kampen, N. G., *Stochastic Processes in Physics and Chemistry*, Elsevier, Amsterdam, 5th edition, 2004.

[3] Gillespie, D. T., "A rigorous derivation of the chemical master equation," *Physica A*, 1992, **188**, pp. 404–425.

[4] Stewart, W. J., *Numerical Solution of Markov Chains*, Marcel Dekker, New York, 1991.

[5] Grassmann, W. K., *Computational Probability*, Springer, 1st edition, 1999.

[6] Stewart, W. J., *Introduction to the Numerical Solution of Markov Chains*, Princeton, New Jersey, 1994.

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

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

[9] Burrage, K., Hegland, M., Macnamara, S. and Sidje, R., "A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems," *Proc. of The A.A.Markov 150th Anniversary Meeting*, 2006, no. 21–37.

[10] Peles, S., Munsky, B., and Khammash, M., "Reduction and solution of the chemical master equation using time-scale separation and finite state projection," *J. Chem. Phys.*, 2006, **125**, 204104.

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

[12] Gillespie, D. T., *Markov Processes: An Introduction for Physical Scientists*, Academic Press, Boston, 1992.

[13] 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.

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

[15] Li, H., Cao, Y., Petzold, L., and Gillespie, D., "Algorithms and software for stochastic simulation of biochemical reacting systems," *Biotech. Prog.*, 2008, **24**, pp. 56–61.

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

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

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

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

[20] Davis, T. A., *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2006.

[21] 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.

[22] Cheney, E. W., Light, W. A., *A Course in Approximation Theory*, Brooks/Cole, Pacific Grove, 2000.