Reduction and solution of the chemical master equation using time scale separation and finite state projection

Slaven Peleš,[a] Brian Munsky,[b] and Mustafa Khammash[c]
Department of Mechanical Engineering, University of California, Santa Barbara, California 93106-5070

(Received 14 August 2006; accepted 23 October 2006; published online 27 November 2006)

The dynamics of chemical reaction networks often takes place on widely differing time scales—from the order of nanoseconds to the order of several days. This is particularly true for gene regulatory networks, which are modeled by chemical kinetics. Multiple time scales in mathematical models often lead to serious computational difficulties, such as numerical stiffness in the case of differential equations or excessively redundant Monte Carlo simulations in the case of stochastic processes. We present a model reduction method for study of stochastic chemical kinetic systems that takes advantage of multiple time scales. The method applies to finite projections of the chemical master equation and allows for effective time scale separation of the system dynamics. We implement this method in a novel numerical algorithm that exploits the time scale separation to achieve model order reductions while enabling error checking and control. We illustrate the efficiency of our method in several examples motivated by recent developments in gene regulatory networks. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2397685]

I. INTRODUCTION

Living organisms have evolved complex robust control mechanisms with which they can regulate intracellular processes and adapt to changing environments. Experiments have shown that significant stochastic fluctuations are present in these processes. The investigation of stochastic properties in genetic systems involves the formulation of a mathematical representation of molecular noise and devising efficient computational algorithms for computing the relevant statistics of the modeled processes. When devising computational models for describing these cellular systems, one must take into consideration that many of the cellular processes take place far from equilibrium and on time scales longer than the cell replication cycle. As a result, these processes never reach the asymptotic state. Furthermore, characteristic time scales in intracellular processes often differ by several orders of magnitude. These considerations pose considerable challenges to any computational approach for modeling cellular networks.

The most significant progress has been made when modeling intracellular processes as a series of stochastic chemical reactions involving proteins, RNA and DNA molecules. Mathematical formulation for such models is generally provided by the chemical master equation.[1] However, the complexity of gene regulatory networks poses serious computational difficulties and makes any quantitative prediction a difficult task. Monte Carlo based approaches are typically used in getting realizations of the stochastic processes whose distributions evolve according to the chemical master equation. One Monte Carlo simulation technique that has gained wide use is stochastic simulation algorithm.[2] Here random numbers are generated for every individual reaction event in order to determine (i) when the next reaction will occur and (ii) which reaction it will be. However, for most systems, huge numbers of individual reactions may occur, and the stochastic simulation algorithm can be too computationally expensive and does not provide guaranteed error bounds. To address the speed issue, approximations have been developed that exploit time scale separation or that leap through several reactions at a time. These are discussed in more detail in the text. However, a different approach is presented by the recently proposed finite state projection algorithm[3] which gets approximate solutions of the chemical master equation direction with guaranteed error bounds and often improved speed.

The finite state projection approach provides an analytical alternative that avoids many of the shortcomings of Monte Carlo methods. Thus far the advantages of the finite state projection have been demonstrated for a number of problems.[3–6] In this paper we show that the applicability of the finite projection approach can be dramatically enhanced by taking advantage of tools from the fields of modern control theory and dynamical systems. In particular, we present a new approach that utilizes singular perturbation theory in conjunction with the finite state projection approach to improve the computation time and facilitate model reduction by taking advantage of multiple time scales. Model reduction approaches based on singular perturbation theory have been used in various areas of engineering and science.[7–10] When coupled with the finite state projection method, many of the advantages of singular perturbation approach find an application in the field of stochastic chemical kinetics. The finite state projection method retains an important subset of the state space and projects the remaining part (which can be infinite) onto a single state, while keeping the approximation error strictly within prespecified accuracy. The resulting fi-

[a]URL: http://www.engineering.ucsb.edu/~peles/
[b]Electronic mail: brianem@engineering.ucsb.edu
[c]Electronic mail: khammash@engineering.ucsb.edu

nite model is given in an analytical form and allows us to implement reduction techniques based on singular perturbations. When multiple time scales are present, our proposed singular perturbation approach attains dramatic speedups without compromising the accuracy of the computation, which is known *a priori* and which can be adjusted before the bulk of calculations is carried out.

We illustrate our method using two examples arising from recent experiments with *Escherichia coli* bacteria: we analyze the PAP gene regulatory network and cellular heat shock response. Our method is not limited to biological systems, and can be applied to any chemical kinetics problem that is described by a master equation.

This paper is organized as follows: in Sec. II we give a brief overview of some computational methods that have been used to study stochastic gene network models. In Sec. III we describe the mathematical details of our method. In Sec. IV A we demonstrate how to use time scale separation together with the finite state projection method, and in Sec. IV B we provide an example of how our method can be applied to a realistic gene network problem. In Sec. V we discuss the advantages of our approach over presently used methods and, finally, in Sec. VI, we summarize our results and outline prospects for further research.

II. BACKGROUND

For a system of n chemical species, the state of the system inside the cell is specified by copy numbers of each relevant molecule $\mathbf{X}=(X_1,X_2,\ldots,X_n)$. Often, these numbers are relatively small and reactions take place far from the thermodynamic limit, so that mesoscopic effects, most notably fluctuations, have to be taken into account. The state space of the system is not continuous, but a discrete lattice, where each node corresponds to a different $\mathbf{X}$. The size of the lattice is limited by the maximum possible populations of the n chemical species in the finite volume cell.

At the mesoscopic scale one describes the dynamics of the system in terms of the probability of finding the system in a given state $\mathbf{X}$, rather than in terms of trajectories in the state space. The dynamics of the system can be modeled by the master equation for a Markov process on a lattice[1] or jump Markov process. Although respectable attempts have been made to introduce deterministic mesoscopic models for chemical reactions,[11] presently stochastic methods are used almost exclusively in the study of intracellular processes at the mesoscopic level.

The master equation describes the time evolution of the probability of finding the system in a particular state $\mathbf{X}$. With an enumeration $\mathbf{X}\rightarrow i$, which maps each possible state to a single index, the master equation can be written in a familiar gain-loss form[1]

$$\frac{dp_i(t)}{dt}=\sum_{j\neq i}\left[w_{ij}p_j(t)-w_{ji}p_i(t)\right], \tag{1}$$

where p_i is the probability of finding the system in the ith state, while w_{ij} are propensities. The latter define probabilities $w_{ij}dt$ that the system will transition from the jth to the ith state during an infinitesimal time interval dt. The propensities may be obtained from the chemical reaction rates, which often can be measured experimentally. Propensities w_{ij} are either constant or may depend on time if the system is in an external time-dependent field. For simplicity in our presentation we consider only constant propensities, nevertheless the same formalism applies in the time-dependent case. The first term on the right hand side of the master equation describes an increase in the probability p_i due to transitions to the ith state from all other states j, while the second term describes a decrease in p_i due to transitions form the ith state to other states j. If the system is initially found in a state k, the initial condition for the chemical master equation can be written as $p_i(0)=\delta_{ik}$, where δ_{ik} is the Kronecker delta.

The solution for this problem is the probability $p_i(t)$ that the system initially found in state k will be in state i at the later time t. If we define $A_{ij}=w_{ij}-\delta_{ij}\Sigma_k w_{ki}$, the chemical master equation can be written in a more compact form

$$\dot{p}_i(t)=\sum_j A_{ij}p_j(t). \tag{2}$$

Therefore, the chemical master equation on a discrete state space can be written as a system of countably many ordinary differential equations. Note that such system is linear even when the chemical kinetics is governed by nonlinear processes.[1,12]

The solution to the chemical master equation generally can be expressed in terms of evolution operator $\mathbf{p}(t)=\mathcal{A}(t,0)\mathbf{p}(0)$, which in case of a finite A can be written as

$$\mathbf{p}(t)=\exp(At)\mathbf{p}(0). \tag{3}$$

Solving the master equation at first seems to be a rather simple problem, as there are many efficient methods for solving systems of linear ordinary differential equations. However, if we consider, for example, a process involving three proteins, where each protein comes in, say, 1000 copies per cell, that gives us up to a billion of different states and a myriad of possible transitions between them. Carrying out calculations for a such system without any insight about its biological structure would be impractical at least.

This problem may be ameliorated by using a Monte Carlo type of computation.[13] The idea behind this approach is to start from some initial probability distribution $p_j(0)=\delta_{jk}$, then using some probabilistic rule we choose which reaction will take place next, and compute the new state j where the system will be found at some later time t. The probabilities of picking a particular reaction are given by propensities w_{ij}. The hope is that after sufficiently many calculations like this the histogram containing all outcomes will approximate well the solution of the chemical master equation $\mathbf{p}(t)$. The advantage of this approach is that we do not need to calculate the whole matrix A. Instead, we calculate on the fly only those matrix elements that are required for the computation at hand. Furthermore, this method is broadly applicable as it requires little knowledge about the details of the system under consideration. It has been demonstrated[2] that in the limit case of infinitely many runs the Monte Carlo solution approaches the exact solution to the chemical master

equation. Therefore the accuracy of the computation can be increased by simply generating more Monte Carlo simulations.

On the downside Monte Carlo methods are notorious for their slow convergence,[13] and the amount of computation necessary to get an accurate result may be too large to be completed in a reasonable amount of time. Also, computers cannot produce truly random numbers, so one has to generate something that is as close as possible. Programs called random number generators[14] create periodic sequences of numbers with a large period, which imitate series of random numbers. If the period is too short, periodic patterns will create numerical artifacts in the calculation. On the other hand, high quality random number generators, such as RANLUX,[15] take significantly more computer processing time to execute.

Despite their shortcomings Monte Carlo methods remain an important tool for the study of intracellular processes. Over the years a variety of specialized Monte Carlo implementations[16–22] that address the above mentioned issues has been developed.

An alternative approach known as the finite state projection[3,4,6] has been proposed recently by Munsky and Khammash. The method is based on a simple observation that some states are more likely to be reached in a finite time than are others. One can then aggregate all improbable states in (2) into a single sink, and consider all transitions to those states as an irreversible loss. This method automatically provides a guaranteed error bound that may be made as small as desired.[3] With some intuition about the system's dynamics, such as knowing the macroscopic steady state, one can develop an efficient system reduction scheme. It has been demonstrated for a number of biological problems[3,4] that in this way the system (2) can be reduced to a surprisingly small number of linear ordinary differential equations, thereby dramatically reducing the computation time. The reduced system can be treated *analytically*, and the method does not require computationally expensive random number generation.

By discarding unlikely states in a systematic way, the finite state projection method provides for a bulk system reduction, but the original finite state projection stops far short from what can be achieved. For example, the method does not consider how transitions between the remaining states take place. Transition rates between different states typically vary over several orders of magnitude, and by treating them equally one may waste considerable time performing computations to obtain a precision that far outstrips the models accuracy.

Low probability transitions occur infrequently, so the processes involving them take place over long time scales, while high probability transitions correspond to fast intracellular processes. Different time scales can pose computational problems, as the system of ordinary differential equations (2) becomes stiff. On the other hand, depending on the length of the observation time, the system can be further simplified. For short times, slow processes may be neglected; for long times, the effects of fast processes can be averaged.

In what follows we introduce a computational method that addresses these shortcomings by taking advantage of multiple time scales in the master equation to simplify the system of equations and reduce the computation time. This method is in a sense complementary to the finite state projection. It can be used independently, but significant benefits may be achieved when the two methods are combined.

III. TIME SCALE SEPARATION

In order to define a proper chemical master equation, matrix A has to satisfy some general properties. Since by definition propensity functions $w_{ij} \geqslant 0$, all off-diagonal elements of A are non-negative. For the same reason, all diagonal elements of A are nonpositive.

In a closed system the probability has to be conserved, so that $\Sigma_i p_i(t) = \text{const}$ for all times. That means

$$\frac{d}{dt}\sum_i p_i(t) = 0 \Rightarrow \sum_i \sum_j A_{ij} p_j(t) = 0, \tag{4}$$

and hence

$$\sum_j \left(\sum_i A_{ij} \right) p_j(t) = 0, \tag{5}$$

for any probability distribution $\mathbf{p}(t) = (p_1(t), \ldots, p_N(t))$. Here with N we denote the number of all possible states where the system can be found.[36]

Therefore it must hold that $\Sigma_i A_{ij} = 0$, i.e., the sum of the elements in each column of A must be zero. In other words vector $\mathbf{1} = (1, 1, \ldots, 1)$ is a left eigenvector of A with associated eigenvalue zero,

$$\mathbf{1}^T A = 0. \tag{6}$$

This further means that for the matrix A there exists at least one right eigenvector $\mathbf{v}$ with the zero eigenvalue,

$$A\mathbf{v} = 0. \tag{7}$$

The eigenvector $\mathbf{v}$ represents the steady state probability distribution for the system, and is the nontrivial solution to (2). Furthermore it can be shown[1] that other eigenvalues of A have negative real parts if the matrix A is irreducible, i.e., it cannot be written in a block diagonal form.

Note that we are interested here in the *nontrivial* solution to (2), which exists since $\det A = 0$. There also exists a trivial solution $\mathbf{p} = \mathbf{0}$, but we can discard it as nonphysical since it does not satisfy the normalization condition $|\mathbf{p}| = 1$.

In gene networks we can often identify clusters of states within which transitions occur quite frequently, while transitions between the clusters are relatively rare. The chemical master equation that corresponds to such a situation has a nearly block diagonal structure, so the matrix A in (2) can be written in the form

$$A = H + \epsilon V, \tag{8}$$

where H is a block diagonal matrix describing transitions within the clusters, matrix V describes transitions from one cluster to another, and $\epsilon > 0$ is a small parameter.

In the limit case, $\epsilon \to 0$, the system remains in one cluster of states for an infinitely long time, and the probability for the system to be found in one of the states within the original

cluster is one. Therefore, same as the matrix A, each block of H should define a proper master equation. Each block of H has one zero eigenvalue with associated eigenvector $\mathbf{v}_i$, which describes the steady state probability distribution in the ith cluster, while all other eigenvalues of the block have negative real parts.

It is relatively inexpensive to compute the full eigensystems for the smaller blocks of H. From the eigenvectors for each block, one can then easily construct a matrix S that diagonalizes H,

$$S^{-1}HS = \Lambda, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N). \tag{9}$$

Matrix S has the same block diagonal structure as H. This procedure is further simplified if some blocks of H are identical, as is often the case. We label eigenvectors and eigenvalues of H so that $\mathrm{Re}\{\lambda_1\} \geqslant \mathrm{Re}\{\lambda_2\} \cdots \geqslant \mathrm{Re}\{\lambda_N\}$. The first m eigenvalues are then equal to zero ($\lambda_{i \leqslant m}=0$) and the rest have negative real parts.

In order to keep our presentation streamlined, we shall assume that for all negative eigenvalues $|\mathrm{Re}\{\lambda_{i>m}\}| \gg \epsilon$. This is always satisfied if it is possible to make a clear distinction between fast and slow reactions. This assumption can be relaxed and similar results obtained, as we shall demonstrate later.

By applying now S^{-1} to both sides of (2) we obtain

$$\dot{\mathbf{x}} = (\Lambda + \epsilon \widetilde{V})\mathbf{x}, \tag{10}$$

where $\mathbf{x}=S^{-1}\mathbf{p}$, and $\widetilde{V}=S^{-1}VS$. The equation above can be written in the component form as

$$\dot{x}_i = \lambda_i x_i + \epsilon \sum_{j=1}^{N} \widetilde{V}_{ij} x_j. \tag{11}$$

From singular perturbation theory (see Appendix) there exists a near identity transformation

$$\mathbf{x} = (I + \epsilon G)\mathbf{y}, \tag{12}$$

which removes all $O(\epsilon)$ terms, which depend on $x_{i>m}$, from the first m equations ($i \leqslant m$). In other words, Eq. (11) where $\lambda_i=0$ can be decoupled from the rest of the system by a coordinate transformation (12) through the order $O(\epsilon)$. In the new coordinates the first m equations become

$$\dot{y}_i = \epsilon \sum_{j=1}^{m} \widetilde{V}_{ij} y_j + O(\epsilon^2). \tag{13}$$

By truncating $O(\epsilon^2)$ terms in (13) we reduce our system of equations to an m-dimensional problem. The reduced system still approximates well the dynamics of the full system, but it is computationally less expensive to solve. Because (11) has a stable fixed point solution, if initially $|\mathbf{x}(0)-\mathbf{y}(0)|=O(\epsilon)$, then for all times $t>0$ it holds $|\mathbf{x}(t)-\mathbf{y}(t)|=O(\epsilon)$.

Note that if λ_i is smaller or of the same order as ϵ, the near-identity transformation (12) and its inverse introduce corrections to the ith equation that is only of order $O(\epsilon^2)$. Therefore we do not need to find the exact form of the near-identity transformation, we can simply truncate all terms containing $x_{i>m}$ from the system (11).

The first m equations can be solved now independently of the rest of the system, and their solution can be written as

$$y_i(t) = \sum_{j=1}^{m} [\exp(\epsilon \widetilde{V}' t)]_{ij} y_j(0), \tag{14}$$

where $\widetilde{V}'$ is $m \times m$ principal submatrix of $\widetilde{V}$, with elements $\widetilde{V}_{i,j \leqslant m}$. In many cases of interest, solving (13) is a manageable problem, unlike getting general solution for the chemical master equation (2). Since in the long time limit

$$\lim_{t \to \infty} x_{i>m}(t) = O(\epsilon), \tag{15}$$

as we show in the Appendix, we claim that from the solution to the truncated system (14), we can easily construct an approximation to the full solution of the chemical master equation (3). To do so, we first define an evolution operator $\widetilde{\mathcal{V}}(t)$ such that $\widetilde{\mathcal{V}}_{ij}(t)=[\exp(\epsilon \widetilde{V}' t)]_{ij}$ for $i,j \leqslant m$, and $\widetilde{\mathcal{V}}_{ij}(t)=0$ otherwise. In a block matrix form this is

$$\widetilde{\mathcal{V}}(t) = \begin{pmatrix} \exp(\epsilon \widetilde{V}' t) & 0 \\ 0 & 0 \end{pmatrix}. \tag{16}$$

The price we pay for simplicity here is that $\widetilde{\mathcal{V}}(0)$ is not an identity matrix, so the initial condition $y_{i>m}(0)$ also gets truncated. That results in an additional transient error that is generally larger than $O(\epsilon)$. Also, note that neither $\widetilde{V}$ nor $\widetilde{V}'$ are generators for the evolution operator $\widetilde{\mathcal{V}}$, so their eigenvectors cannot be used directly to compute the steady state probability distribution for $\mathbf{p}(t)$. Finally, by performing the inverse S transformation on $\widetilde{\mathcal{V}}(t)$, we obtain

$$\mathcal{V}(t) = S\widetilde{\mathcal{V}}(t)S^{-1}, \tag{17}$$

which leads to the $O(\epsilon)$ approximation to the asymptotic solution of the chemical master equation (2), that is

$$\lim_{t \to \infty} |\mathbf{p}(t) - \mathcal{V}(t)\mathbf{p}(0)| = O(\epsilon). \tag{18}$$

We can extend this result to finite times (see Appendix) since we are guaranteed that there exists a finite time $T(\epsilon)$ after which the transient truncation error becomes smaller than $O(\epsilon)$. That time can be estimated from the leading nonzero eigenvalue as

$$T(\epsilon) \sim \ln \epsilon / \mathrm{Re}\{\lambda_{m+1}\}. \tag{19}$$

If the time scale separation in (8) is done accurately, this transient is negligible for all practical purposes. However, time scale separation in a large system is not always obvious, and may be error prone. We discuss this further as we formulate our algorithm.

A. Computational algorithm

Due to the truncation of $\widetilde{V}$, only contributions of the first m columns of S and m rows of S^{-1} affect the approximate solution. As a result the computation can be greatly simplified—instead of calculating full eigensystems for each block H_i, it suffices to find only the eigenvectors $\mathbf{v}_i$ associated with the zero eigenvalues. Instead of S we use the N

$\times m$ matrix S^R, whose columns are made up of the right eigenvectors of H, while instead of S^{-1} we use the $m \times N$ matrix S^L, whose rows are made up of the left eigenvectors of H. Note that the left eigenvectors are always $\mathbf{1}_i^T$, provided all $|\mathbf{v}_i|=1$, so the matrix S^L is obtained at no computational cost. The accuracy of the calculation is known *a priori* to be $O(\epsilon)$ for all $t > T(\epsilon)$.

To improve the reliability and robustness of our calculation, we can optionally add a transient time check to our algorithm. To do so we first need to find eigenvalues for all blocks H_i. This comes at a relatively small computational cost, and can be performed before all the other calculations. The transient time needed to obtain the desired accuracy is estimated from the leading negative eigenvalue λ_{m+1} according to (19). If the transient is too long, that can be remedied by expanding matrices S^R and S^L to include the right and left eigenvectors corresponding to λ_{m+1}, respectively. The transient time is then governed by next negative eigenvalue λ_{m+2}. This procedure can be repeated until the desired accuracy is achieved, thereby sacrificing computational time for precision. Note that in this case the right eigenvectors corresponding to nonzero eigenvalues cannot be obtained trivially.

By performing this test, we also ensure that condition $|\lambda_{i>m}| \gg \epsilon$ is satisfied. Eigenvalues of H that are $O(\epsilon)$ or smaller will result in long transient times. By expanding transformation S to include eigenvectors corresponding to these eigenvalues, we essentially treat them as if they were part of $\tilde{V}$. This procedure adds robustness to the method with respect to separating fast and slow reactions in (8). We can summarize the proposed algorithm in following six steps.

(1) Specify problem parameters. If necessary apply a finite projection to the full state space. Use propensity functions and/or physical intuition to separate H and V.
(2) Find the eigenvalues of the uncoupled system, and identify "slow" ones with respect to a preset transient time $T(\epsilon)$.
(3) Find the right and left eigenvectors corresponding to the slow eigenvalues and construct rectangular matrices S^R and S^L.
(4) Calculate $k \times k$ matrix $\tilde{V}' = S^L V S^R$, where k is the number of slow eigenvalues.
(5) Compute $k \times k$ matrix $\exp(\epsilon \tilde{V}' t)$.
(6) Perform the inverse transformation $S^R \exp(\epsilon \tilde{V}' t) S^L = \mathcal{V}(t)$ in order to obtain the approximation to $\exp(At)$ for all times $t > T(\epsilon)$.

Solving the chemical master equation, written in a form of a system of linear ordinary differential equations (2), is essentially a matrix eigenvalue problem. Therefore, the computational cost for our method will be almost entirely determined by the efficiency of the eigenvalue algorithm we use. Typically, for these algorithms the computational cost scales as a cube of the dimension of the matrix (see, e.g., Ref. 23 and references therein). We now estimate how the efficiency is improved by performing time scale separation. For simplicity we assume that all blocks of matrix H have the same size N/m. The computational cost of finding the eigensystem of each block is then $(N/m)^3$. There are m such blocks, so a conservative estimate of reducing the system using our algorithm would be N^3/m^2. The total cost is the sum of the cost of the model reduction and the cost of solving the reduced system, that is, $N^3/m^2 + m^3$. It is easy to show that this is always smaller than N^3, the cost of solving the full system, as long as $1 < m < N$. Therefore the computational cost may be reduced by a factor of

$$\frac{N^3}{N^3/m^2 + m^3}$$

when using the time scale separation. Of course, this is only a rough estimate of how the computational cost scales, but it gives a good idea of what improvements can be expected.

B. Example

Let us illustrate this technique with a simple example. We assume two weakly interacting systems that can be found in three different states each. We choose matrices H and V in an arbitrary way, with the only constraint that they define a master equation. In our example

$$H = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix} \tag{20}$$

is a block diagonal matrix with blocks

$$H_1 = \begin{pmatrix} -4 & 2 & 4 \\ 1 & -2 & 0 \\ 3 & 0 & -4 \end{pmatrix} \quad \text{and} \quad H_2 = \begin{pmatrix} -6 & 3 & 2 \\ 2 & -3 & 0 \\ 4 & 0 & -2 \end{pmatrix}. \tag{21}$$

We find that blocks H_1 and H_2 have one zero eigenvalue each, with corresponding right eigenvectors $\mathbf{v}_1 = (4,2,3)$ and $\mathbf{v}_2 = (3,2,6)$. From these eigenvectors, we assemble the matrix S^R,

$$S^R = \begin{pmatrix} 4/9 & 0 \\ 2/9 & 0 \\ 3/9 & 0 \\ 0 & 3/11 \\ 0 & 2/11 \\ 0 & 6/11 \end{pmatrix}. \tag{22}$$

The matrix composed of left eigenvectors of H_1 and H_2 is similarly used to form S^L,

$$S^L = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{23}$$

In our example the coupling matrix is

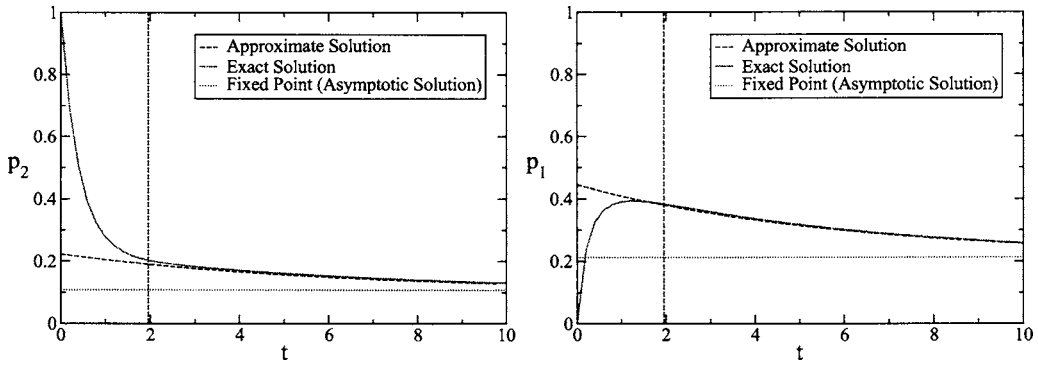

FIG. 1. Comparison of the approximate and the exact solution to the master equation in Sec. III B. The initial probability distribution is $p_i(0)=\delta_{2i}$. The transient time is estimated to be $T(\epsilon)=\ln \epsilon/\lambda_3=1.96$ for $\epsilon=0.01$, and is denoted by the vertical line on the graph.

$$V=\begin{pmatrix} -8 & 0 & 0 & 5 & 3 & 2 \\ 0 & -5 & 0 & 2 & 3 & 1 \\ 0 & 0 & -12 & 4 & 6 & 2 \\ 4 & 2 & 3 & -11 & 0 & 0 \\ 1 & 2 & 5 & 0 & -12 & 0 \\ 3 & 1 & 4 & 0 & 0 & -5 \end{pmatrix}. \tag{24}$$

To get the equations for the slowly changing variables (13), we calculate

$$\tilde{V}'=S^L V S^R=\begin{pmatrix} -87/11 & 78/11 \\ 29/3 & -26/3 \end{pmatrix}. \tag{25}$$

Next, we calculate the evolution operator for the truncated system, $\exp(\epsilon\tilde{V}'t)$, and perform the inverse S transformation to obtain

$$\mathcal{V}(t)=S^R \exp(\epsilon\tilde{V}'t)S^L. \tag{26}$$

Finally, we obtain the approximate solution as

$$\mathbf{p}(t)=\mathcal{V}(t)\mathbf{p}(0). \tag{27}$$

As an illustration, in Fig. 1 we show components $p_1(t)$ and $p_2(t)$ of the solution above for the initial condition $p_i(0)=\delta_{2i}$, and $\epsilon=0.01$. We can see that after the transient time (19) has elapsed we obtain a good agreement between the exact and the approximate solution to this example problem.

To further support our results, we randomly generate a large number of master equations with similar near block diagonal structure and compare their exact solutions to the approximate solutions obtained using our approach. The numerical results presented in Fig. 2 show that the approximation error is controlled by the small parameter ϵ.

IV. APPLICATIONS

A. Three-species fast-slow reaction

In the previous section we demonstrated the efficiency of the singular perturbation theory when applied to chemical master equation (2). One can immediately see, however, that this method becomes less feasible to implement as the size of the system under consideration increases. The finite state projection method[3] provides for a preliminary reduction of the system (2) that allows for much broader implementation of our method.

Consider a three-species reaction system described by

$$S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2 \xrightarrow{c_3} S_3. \tag{28}$$

Such reactions are common in gene regulatory networks. For example, they arise in modeling of cellular heat shock response,[24] where s_1, s_2, and s_3 correspond to the σ_{32}-DnaK complex, the σ_{32} heat shock regulator, and the σ_{32}-RNAP complex, respectively. At normal physiological temperatures σ_{32} protein is found almost exclusively in a complex σ_{32}-DnaK. As the temperature increases this complex becomes less stable and there is a non-negligible probability of

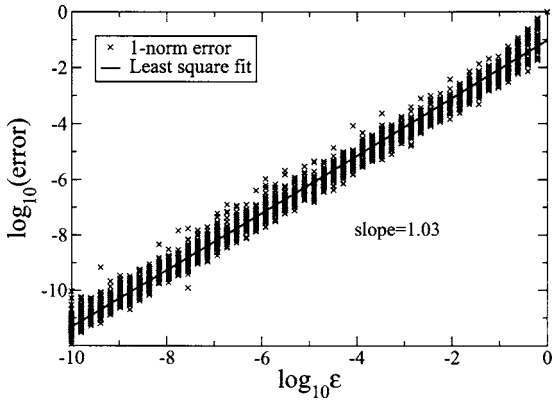

FIG. 2. 1-norm error in probability distribution for the truncated solution vs ϵ. For each value of ϵ we have randomly generated 50 matrices H and V, so that every $H+\epsilon V$ defines a proper master equation. Each matrix H has between 2 and 6 blocks and each block has size between 2 and 21. The elements of H and V are randomly generated from a uniform distribution between 0 and 1. The probability distributions were calculated after time $t=2T(\epsilon)=2\ln \epsilon/\lambda_{m+1}$.

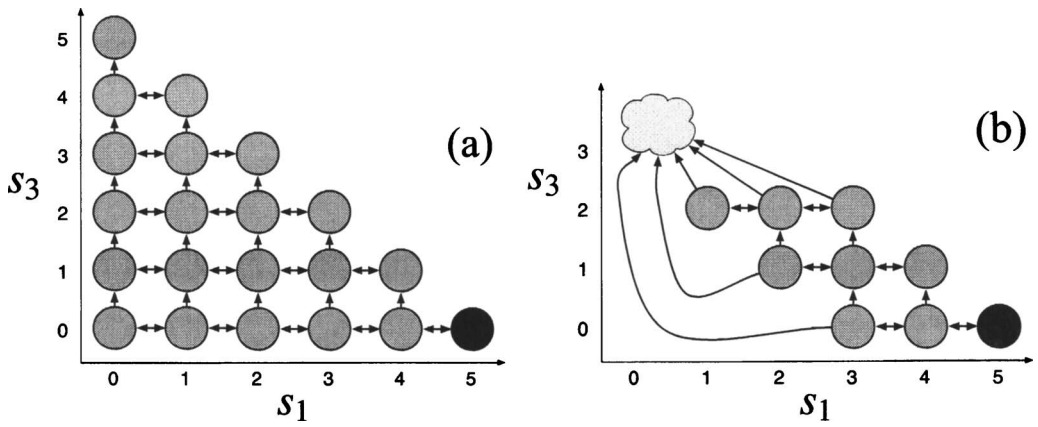

FIG. 3. (a) Two-dimensional lattice representing possible states and transitions in the heat shock model. Here s_1 and s_3 are populations of σ_{32}-DnaK and σ_{32}-RNAP compounds, respectively, while s_2 is the population of free σ_{32} molecules. Reactions $s_1 \rightleftharpoons s_2$ are represented by bidirectional horizontal arrows and reactions $s_2 \rightarrow s_3$ are represented with vertical arrows. The total number of σ_{32} is constant (in this example $s_1+s_2+s_3=5$), so the chemical state of the system is uniquely defined by s_1 and s_3 alone. (b) The same lattice after applying the finite state projection. Unlikely states have been aggregated into a single sink state.

finding free σ_{32} inside the cell. The free σ_{32} then can combine with RNA polymerase through what can be considered an irreversible reaction to form a σ_{32}-RNAP complex. In turn, σ_{32}-RNAP initiates transcription of genes encoding heat shock proteins. This reaction has been analyzed before using various computational methods including Monte Carlo implementations.[17,25]

In the biological system, the relative rates of the reactions are such that the reaction from s_2 to s_1 is by far the fastest, and σ_{32} molecules infrequently escape from DnaK long enough to form the σ_{32}-RNAP complex. The purpose of this mechanism is to strike a balance between fixing the damage produced by heat and saving the cell's resources, as a significant portion of cell energy is consumed when producing heat shock proteins. The optimal response to the heat shock is not massive, but measured production of heat shock proteins, which leaves sufficient resources for other cellular functions. We use the following set of parameters values for the reaction rates.[17,25]

$$c_1 = 10, \quad c_2 = 4 \times 10^4, \quad c_3 = 2. \tag{29}$$

For simplicity, in our model we assume that the total number of σ_{32} protein—free or in compounds—is constant, so that $s_1+s_2+s_3=$const. With this constraint the reachable states of this three-species problem can be represented on a two-dimensional lattice.

For illustrative purposes, Fig. 3(a) shows one such lattice for an initial condition of $s_1=5$ and $s_2=s_3=0$. Here, the total population is fixed at five, and there is a total of 21 reachable states.

We first apply the finite state projection. We estimate that all states where $s_2>2$ or $s_3>2$ are unlikely to be reached in a short time, so we aggregate them into a sink node as shown in Fig. 3(b) thereby reducing this to a ten state problem. From the transitions to the aggregated state, we find a strict upper bound on the error introduced by such an approximation. For our set of parameters the 1-norm approximation error is guaranteed to be below 0.08 for any time $t \leqslant 500$.

Next, we further reduce this system by applying time scale separation. Elements of the matrix A_{FSP}, which defines the master equation for the system obtained after the finite state projection, can be read off of Fig. 3(b). A smart bookkeeping practice would be to write $A_{\mathrm{FSP}}=H+\epsilon V$, and record all reversible reactions $s_1 \rightleftharpoons s_2$ in the matrix H and all other reactions, including $s_2 \rightarrow s_3$ and transitions to the aggregated state, in the matrix V. By doing so we ensure that all fast reactions are contained in H. Note that there is no unique way to separate fast and slow reactions and we chose this one for its simplicity. Matrix H has a block diagonal structure

$$H = \begin{pmatrix} H_3 & & & \\ & H_2 & & \\ & & H_1 & \\ & & & 0 \end{pmatrix}, \tag{30}$$

where each block

$$H_k = \begin{pmatrix} -(k+2)c_1 & c_2 & 0 \\ (k+2)c_1 & -(k+1)c_1 - c_2 & 2c_2 \\ 0 & (k+1)c_1 & -2c_2 \end{pmatrix} \tag{31}$$

corresponds to a row of states in Fig. 3(b). The zero in the last row of H is just a scalar, and it corresponds to the aggregated state. Note that in this case it was the finite state projection that generated this characteristic near block diagonal structure.

The matrix ϵV is made up of irreversible reactions [vertical transitions in Fig. 3(b)] and therefore has a lower triangular form,

$$\epsilon V = \begin{pmatrix} 0 & & & & & & & & \\ 0 & -c_3 & & & & & & & \\ 0 & 0 & -3c_1-2c_3 & & & & & & \\ 0 & c_3 & 0 & 0 & & & & & \\ 0 & 0 & 2c_3 & 0 & -c_3 & & & & \\ 0 & 0 & 0 & 0 & 0 & -2c_1-2c_3 & & & \\ 0 & 0 & 0 & 0 & c_3 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 2c_3 & 0 & -c_3 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c_1-2c_3 \\ 0 & 0 & 3c_1 & 0 & 0 & 2c_1 & 0 & c_3 & c_1+2c_3 \quad 0 \end{pmatrix}.$$

For the reaction rates above, the first four eigenvalues of H are zero, and the rest have negative real parts, each with magnitude of order 10^4 or larger, suggesting that the truncation in (16) is indeed valid for this problem. Therefore the dynamics of nine-dimensional system obtained by the finite state projection can be well approximated by a system of only four linear ordinary differential equations. By applying algorithm from Sec. III A we find that the time scale separation introduces error of order 10^{-3}, with respect to the solution obtained by finite state projection alone. The transient time (19) is estimated to be 2×10^{-4}, and is negligible considering the time interval of interest.

In Fig. 4 we compare the results obtained by solving the full system directly, using finite state projection alone and using finite state projection and time scale separation combined. The figure shows how probability of having no σ_{32}-RNAP complex in the cell decreases with time. All three results are in a good agreement as our calculations predicted.

The advantage of combining the finite state projection and time scale separation becomes obvious if we consider a more realistic and much larger problem with the same reactions but with initial conditions $s_1=2000$ and $s_2=s_3=0$. In this case there are 2 001 000 reachable states, and the full chemical master equation is too large to be tackled directly. However, by applying the finite state projection we find that truncating every state where $s_3\leqslant 350$ and $s_2\leqslant 11$ introduces 1-norm error that is less than 10^{-3} for times $t\leqslant 300$ s. The resulting matrix A is of size 3851×3851, and has near block diagonal form (8) similar to the example in Fig. 3. Its block diagonal part H contains 350 irreducible blocks each with 11 rows and columns. Same as in the previous example the leading nonzero eigenvalue of H has a negative real part of magnitude 10^4, so the system can be reduced to a 351 state model using the time scale separation algorithm. Should we apply time scale separation directly to the full system, the amount of the computation would be significantly larger. The solution to this problem shows how the number of compounds σ_{32}-RNAP grows in time if the temperature is constant and slightly above normal physiological level. This number is proportional to the number of heat shock proteins produced in the cell. With the finite state projection solution, we have computed the probability distribution for s_3 at three times $t=100$, 200, and 300 (Fig. 5, solid lines). We computed the same distributions using time scale separation applied atop of the finite state projection (Fig. 5, dots), and we found that the difference between the two results is indistinguishable. Following the discussion from Sec. III A we estimate that the computational cost is reduced by more than 1000-fold when using time scale separation. Indeed, computational times for the two sets of results in Fig. 5 differ by that order, as shown in Tables I and II.

We further use this example to compare the efficiency of our approach to Monte Carlo based methods. We find that

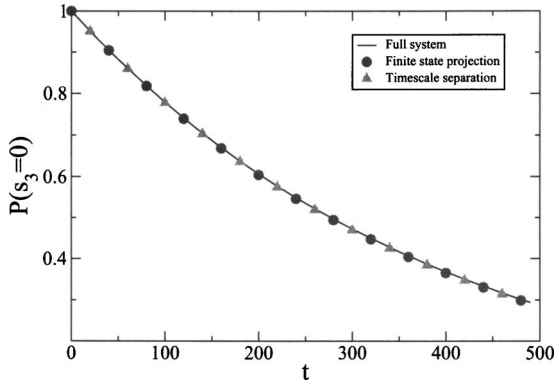

FIG. 4. Probability that no σ_{32}-RNAP molecule has been synthesized in the heat shock toy model.

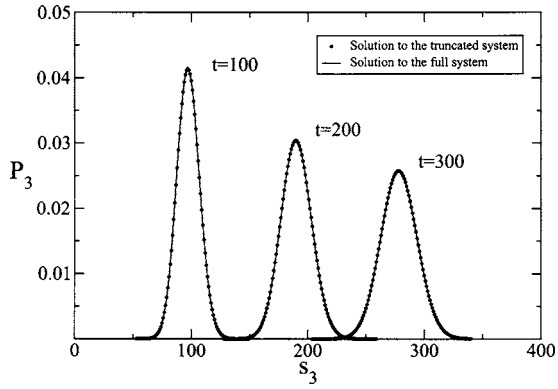

FIG. 5. Probability distribution for s_3 calculated at three different times. The truncated solution (dots) approximates well the solution to the full system (solid lines).

TABLE I. A comparison of the computational cost and accuracy of the finite state projection (FSP) and stochastic simulation algorithm (SSA) for the solution of the chemical master equation, arising in the toy heat shock model, at $t=300$.

Method	No. samples	Time (s)	Error (1-norm)
FSP	N/A[a]	1 472	$\leqslant 2\times 10^{-5}$
SSA	10^3	>20 000	≈ 0.25

[a]The finite state projection runs only once with prespecified error of 2×10^{-5}.

finite state projection outperforms by a wide margin the stochastic simulation algorithm,[2] both in terms of computational time and accuracy (Table I). Neither method attempts to distinguish between fast and slow processes.

When comparing the finite state projection method combined with singular perturbation against Monte Carlo methods designed to deal with systems with multiple time scales, we again find significant advantages when using our approach. In Table II we provide head to head comparison between our method and recently proposed slow scale stochastic simulation algorithm.[17]

All our simulations are coded in MATLAB version 7.2 and run on 2.0 MHz PowerPC Dual G5. Whenever possible we used built in MATLAB functions and we made no attempt to optimize original algorithms. The results shown in Tables I and II should not be interpreted as strict benchmarks, but rather as an indicative examples from our experience in using these methods.

B. PAP switch

Pili are small hairlike structures that enable bacteria to bind to epithelial cells and thereby significantly increase the bacteria's ability to infect host organisms. However, pili expression comes at cost to the bacteria, as the production of pili requires a large portion of the cellular energy. Whether or not *E. coli* are piliated depends upon the regulation of genes such as the pyelonephritis-associated pili (PAP) genes. Here we study a simplified version of the PAP switch model,[4] which analyzes the regulatory network responsible for controlling one type of pili.

Recent experiments[26,27] have identified two transcription factors that affect the expression of the PAP gene, and six binding sites for the two. The transcription factors are DNA adenine methylase (Dam) and leucine responsive regulatory protein (Lrp). Dam binds and applies methyl groups to GATC sites at 2 and 5, as shown in Fig. 6. This Dam methylation is an irreversible and relatively slow process. On the other hand, Lrp binds cooperatively to three adjacent sites at a time, either 1-2-3 or 4-5-6 (Fig. 6). These reactions are fast and reversible. Lrp binding also inhibits Dam methylation. Altogether this makes 16 possible states in which the PAP switch can be found. We describe these chemical reactions by the network model shown in Fig. 7. In our model we assume that PAP transcription occurs only in configuration 11 (Fig. 7) when Lrp is bound to sites 1-2-3 and site 5 is methylated. We further assume that cell replication always resets the system to configuration 1. A solution to the chemical master equation for this problem gives the time evolution of the probability of finding the system in each configuration including configuration 11, which is proportional to the probability of PAP gene expression.

TABLE II. A comparison of the total computational effort and accuracy of the finite state projection with singular perturbation (FSP+SP) and slow scale stochastic simulation algorithm (ssSSA) for the solution of the chemical master equation, arising in the toy heat shock model, at $t=300$.

Method	No. of samples	Time (s)	Error (1-norm)
FSP+SP	N/A	1.88	$\approx 6.6\times 10^{-4}$
ssSSA	10^3	82	≈ 0.24
ssSSA	10^4	826	≈ 0.066
ssSSA	10^5	8130	≈ 0.027

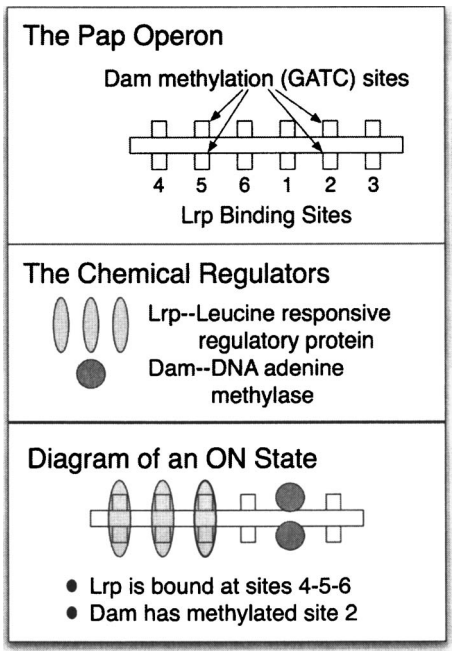

FIG. 6. Schematic of the PAP operon (top), key regulatory components of the PAP switch (middle), and diagram of the operon in its on state (bottom).

Since the two transcription factors bind at significantly different rates, following our bookkeeping practice we record Lrp binding propensities in the matrix H and methylation propensities in V as defined in (8). With a convenient labeling scheme, as shown in Fig. 7, we can express H in a simple block diagonal form,

$$H=\begin{pmatrix} H_1 & 0 & 0 & 0 \\ 0 & H_2 & 0 & 0 \\ 0 & 0 & H_3 & 0 \\ 0 & 0 & 0 & H_4 \end{pmatrix}. \tag{32}$$

Recent experimental data[27] reveal that the propensities of Lrp binding at sites 4-5-6 depend strongly on the methylation pattern of site 5, while propensities of Lrp at sites 1-2-3 do not significantly depend upon the methylation pattern of site 2. Thus we find that there are only two distinct blocks as

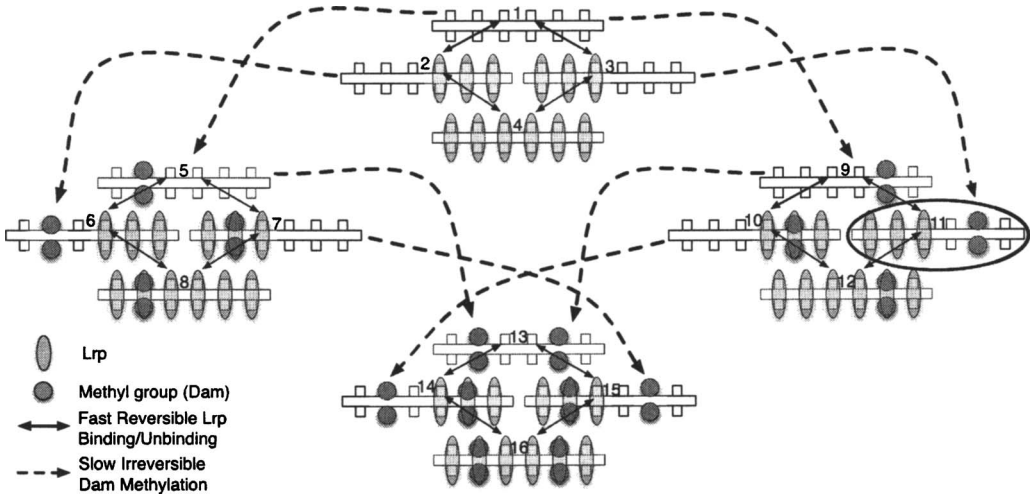

FIG. 7. PAP switch schematic diagram.

$$H_1 = H_3 = \begin{pmatrix} -9500 & 6.8 & 0.09 & 0 \\ 9270 & -18.4 & 0 & 0.09 \\ 230 & 0 & -463.29 & 6.79 \\ 0 & 11.6 & 463.2 & -6.88 \end{pmatrix} \quad (33)$$

and

$$H_2 = H_4 = \begin{pmatrix} -9500 & 62 & 0.09 & 0 \\ 9270 & -73 & 0 & 0.09 \\ 230 & 0 & -463.29 & 61.76 \\ 0 & 11 & 463.2 & -61.85 \end{pmatrix}. \quad (34)$$

Leading eigenvalues for both H_1 and H_2 are zero, while the next largest eigenvalue is of order $\lambda_5 \sim -10$. On the other hand we estimate that all methylation propensities have the same value $\epsilon = 0.17$. Following our labeling scheme (Fig. 7) the nonzero entries of matrix V are then $V_{1,1} = -2$, $V_{2,2} = V_{3,3} = V_{5,5} = V_{7,7} = V_{9,9} = V_{10,10} = -1$, and $V_{5,1} = V_{9,1} = V_{6,2} = V_{11,3} = V_{13,5} = V_{13,9} = V_{14,10} = V_{15,7} = 1$. Therefore, all we need to construct the matrix S^R are the right eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ that correspond to the zero eigenvalues of H_1 and H_2, respectively. Following the footsteps outlined in Sec. III we reduce the PAP switch model to a four-dimensional system and carry out calculation for the probability p_{11}, which is proportional to the PAP transcription probability.

The PAP switch model we presented here is simple enough to be integrated directly so we can compare results for the full system and the reduced system. As we show in Fig. 8, all the important information about the system's behavior is preserved in the reduced model.

This model predicts a short time lag between replication and PAP production, since methylation of site 2 must occur before PAP expression. Further, since Dam methylation at 5 prohibits expression, if the cell waits too long to decide to switch "on," it will most probably miss its chance and remain "off." Thus, a newly created *E. coli* cell will most likely express the *pap* gene at some point shortly after replication. Probabilities of expressing pili drops significantly at later times and cell resources are used for other functions, such as initiating the next replication cycle.

V. DISCUSSION

Monte Carlo methods have been the primary tools for solving the chemical master equation in the mesoscopic study of chemical reactions. The recently proposed finite state projection method showed that solving the chemical master equation can be approached from an entirely different perspective with significant benefits. The original finite state projection method appears to be particularly effective in the study of chemical reactions in biological systems, and it has been demonstrated that in many cases of interest it outperforms standard Monte Carlo implementations.

Over the last five years a number of accelerated or leaping Monte Carlo methods that significantly improve the speed of the calculation have been proposed.[19–22] Leaping algorithms are designed for problems where propensities change slowly after consecutive chemical reactions. That allows for contribution from several reaction channels with similar propensities to be calculated in one step. Instead of sampling to find each individual reaction event, the contribution of each reaction channel during a given time step is obtained from some statistical distribution (e.g., Poisson and binomial). The algorithm effectively "leaps" over a number of reactions in one time step. However, in many biological systems with high reaction rates and low molecule copy

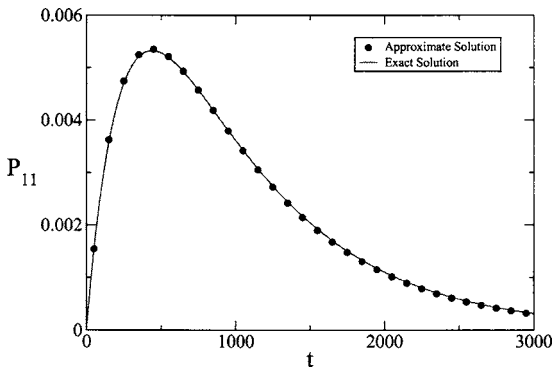

FIG. 8. Time evolution of PAP gene expression probability. Initially no transcription factors are bound the PAP operon, so the initial condition is $p_1(0) = 1$ and $p_{i \neq 1} = 0$. The transient time (19) is less than 1 in our time units.

numbers propensities change sharply with virtually every chemical reaction. One such example is given in Sec. IV A, where free heat shock protein σ_{32} appears in a small number of copies, while its binding rate to DnaK is very high. As the number of free σ_{32} changes with every reaction, the propensities change by a large amount as well, and no two consecutive reactions can be bundled together. In that situation leaping algorithms fall back to a standard Monte Carlo computation.[2] On the other hand the finite state projection method applies equally well to systems where propensities change slowly as to systems where propensities change rapidly. The finite state projection is particularly suitable for the reactions where chemical species come in low copy numbers. This is discussed in more detail in Ref. 1.

Another area where the efficiency of Monte Carlo simulations has been substantially improved is in systems with multiple time scales. If chemical reactions in a system take place at different rates so that fast reactions equilibrate in a time within which slow reactions are unlikely to take place, then the efficiency of a Monte Carlo algorithm can be substantially improved by using quasi-steady-state approximation.[16,17,28–30] In this approximation only slow reactions are simulated using Monte Carlo computations. Fast reactions enter the simulation as averages that are computed from the equilibrium condition at every Monte Carlo step, under the assumption that no slow reaction takes place.[17] Fast reactions occur much more frequently than slow ones, so this approach not only reduces the number of reactions to simulate, but also removes the most computationally demanding part of the system.

The same kind of improvement can be achieved when applying singular perturbation to the chemical master equation (2) or its finite state projection. Moreover, we have demonstrated that our method can significantly outperform Monte Carlo methods using a quasi-steady-state approximation such as the slow scale stochastic simulation algorithm, even when applied to very fundamental problems (Sec. IV A).

Hybrid methods[18,31–33] use a similar approach to tackle stochastic problems with multiple time scales—slow reactions are calculated by Monte Carlo simulations, while the contribution from the fast reactions is averaged over the time between consecutive slow reaction incidents. Hybrid methods approximate fast reactions as a continuous Markov process and calculate their average distribution by solving the appropriate chemical Langevin equation. Solving a stochastic differential equation is a challenging computational task by itself, so the potential benefits of hybrid methods come at some extra cost. Furthermore, the continuous Markov process assumption implies that individual fast reactions cause small relative changes in numbers of reactant and product species.[32] This is clearly not the case in the examples discussed in Sec. IV, and hybrid methods cannot be applied effectively to these problems.

We believe that finite state projection based methods will be no less important in mesoscopic study of chemical kinetics than Monte Carlo methods. They are particularly suitable for systems where chemical species appear in only a few copies, the kind of problems where many Monte Carlo based methods often cannot be efficiently applied.

Also, finite state projection and singular perturbation methods involve only linear algebra operations and do not require use of random number generators. Unlike Monte Carlo methods our approach allows for *a priori* estimates of error and computational cost, an important consideration when one is interested in low probability events in biology.

VI. CONCLUSION

Until recently, it was thought that the chemical master equation could not be analytically solved except for the most trivial systems. Previous work on the finite state projection demonstrated that for many biological systems, bulk system reductions could bring models closer into the fold of solvable problems. Here we have shown that the finite state projection method can be further enhanced when solving the chemical master equation for systems involving multiple time scales. In combination with finite state projection method, we have shown that our algorithm, based upon singular perturbation theory, provides a powerful computational tool for studying intracellular processes and gene regulatory networks.

Similar problems were studied earlier with specially designed Monte Carlo implementations[16,17] or hybrid methods.[18] In contrast to these, our method does not require random number generation, and its accuracy is given *a priori*. A further advantage of our method is its ease of implementation and the speed of computations. The proposed algorithm is particularly fast when implemented on systems for which there are strict means of separating slow and fast reactions. While Monte Carlo based methods are indispensable in the mesoscopic study of chemical reactions, we believe that the finite state projection and related methods present new valuable tools. Indeed, there is a number of cases where they are more efficient and provide better accuracy than their Monte Carlo counterparts.

The finite state projection and our time scale separation approach also provide valuable insight as to how one may further deal with the bewildering complexity that intracellular processes exhibit. First, cellular processes are limited by cell size and available energy. It is then plausible that the main features of intracellular dynamics can be captured in a relatively small subset of the state space, as the results obtained by finite state projection suggest. Another typical feature of intracellular processes is that they are composed of reactions that take place on different time scales. Depending on the observation time of interest, some of these reactions can be neglected, while some will contribute only through their averages. Preliminary success with our approach gives us a hope that relatively simple models for intracellular processes can be tailored when a region in the state space and observation time of interest are known.

Of course, one can easily envision that additional model reductions may be possible to even further enhance the power of both the finite state projection and the time scale separation approach. Indeed some reductions based upon control theory[6] are already becoming apparent. Also, in our

computations we have used off the shelf numerical routines for eigensystem calculations and matrix exponentiation. Further improvements in computational speed can be achieved if these routines are optimized for matrices which define master equations and their special properties. We intend to investigate these possibilities in the future.

ACKNOWLEDGMENTS

This material is based upon work supported by National Science Foundation Grant No. CCF-0326576, Institute of Collaborative Biotechnologies Grant No. DAAD19-03-D-0004 from the US Army Research Office. One of the authors (S.P.) thanks Krešimir Josić of University of Houston for a productive discussion and useful comments.

APPENDIX: SINGULAR PERTURBATION

Singular perturbation theory has been extensively studied in various literatures. However, most of the literature in this area is of wide scope and often very technical. In order to spare the reader some time, here we present a heuristic argument, which provides a mathematical justification for our method, while keeping technicalities at minimum. For rigorous proofs interested reader may want to consult, for example, Refs. 34 and 35.

Consider a weakly perturbed linear N-dimensional system described by (11)

$$\dot{x}_i = \lambda_i x_i + \epsilon \sum_{j=1}^{N} V_{ij} x_j, \tag{A1}$$

where $\lambda_i = 0$ for $i \leqslant m$, and λ_i has negative real part for $i > m$. We want to find a near identity coordinate transformation (12) that would remove as many $O(\epsilon)$ terms as possible from (A1) and "push" them to higher orders in ϵ. After substituting $\mathbf{x} = (I + \epsilon G)\mathbf{y}$ in (A1) we get

$$\dot{y}_i = \lambda_i y_i + \epsilon \sum_{j=1}^{N} V_{ij} y_j - \epsilon \sum_{j=1}^{N} G_{ij} \lambda_j y_j + \epsilon \lambda_i \sum_{j=1}^{N} G_{ij} y_j + O(\epsilon^2). \tag{A2}$$

By equating all $O(\epsilon)$ terms to zero we find

$$\sum_{j=1}^{N} (V_{ij} - G_{ij}\lambda_j + \lambda_i G_{ij}) y_j = 0, \tag{A3}$$

and by solving for G_{ij} we obtain

$$G_{ij} = \frac{V_{ij}}{\lambda_j - \lambda_i}. \tag{A4}$$

Therefore, we can always find G_{ij} except when $\lambda_i = \lambda_j$. In other words, all nonresonant terms can be removed through $O(\epsilon)$ from (A1) by a near identity transformation (12). In our method we are interested in separating slow and fast processes in the system, so we shall define matrix G in (12) as follows:

$$G_{ij} = \begin{cases} \dfrac{V_{ij}}{\lambda_j - \lambda_i}, & i \leqslant m < j \\ 0, & \text{otherwise.} \end{cases} \tag{A5}$$

By substituting this expression for G in (A2) we find that

$$\dot{y}_i = \epsilon \sum_{j=1}^{m} V_{ij} y_j + O(\epsilon^2), \quad i \leqslant m.$$

$$\dot{y}_i = \lambda_i y_i + \epsilon \sum_{j=1}^{N} V_{ij} y_j + O(\epsilon^2), \quad i > m.$$

We observe that first m equations decouple from the rest of the system through $O(\epsilon)$, and can be solved independently after truncating higher order terms. Furthermore, the near identity transformation (12) does not introduce any new $O(\epsilon)$ terms to the first m equations, so it is essentially just a truncation of all $x_{i>m}$ terms from (A1). We do not need to calculate G and perform transformation (12) as such transformation is guaranteed to exist.

It remains to show that the solution to truncated system (13) will be $O(\epsilon)$ close to solution to (A1) on a time interval of interest. These equations are linear and hence can be solved analytically, but let us take an extra step here and expand the solution to (A1) in powers of ϵ,

$$x_i(t) = x_i^{(0)}(t) + \epsilon x_i^{(1)}(t) + \epsilon^2 x_i^{(2)}(t) + \cdots. \tag{A6}$$

By substituting this expression into (A1) and grouping same orders in ϵ we get series of equations

$$\epsilon^0 : \dot{x}_i^{(0)}(t) = \lambda_i x_i^{(0)}(t),$$

$$\epsilon^1 : \dot{x}_i^{(1)}(t) = \lambda_i x_i^{(1)}(t) + \sum_{j=1}^{N} V_{ij} x_i^{(0)}(t),$$

$$\epsilon^2 : \dot{x}_i^{(2)}(t) = \lambda_i x_i^{(2)}(t) + \sum_{j=1}^{N} V_{ij} x_i^{(1)}(t),$$

$$\cdots \cdots,$$

which we can solve in a straightforward way to obtain

$$x_i^{(0)}(t) = e^{\lambda_i t} x_i^{(0)}(0),$$

$$x_i^{(1)}(t) = e^{\lambda_i t} x_i^{(1)}(0) + e^{\lambda_i t} \sum_{j=1}^{N} V_{ij} x_i^{(0)}(0) \int_0^t e^{(\lambda_j - \lambda_i)s} ds.$$

Let us first consider equations $i \leqslant m$. The solution to (A1) through $O(\epsilon)$ then can be written as

$$x_i(t) = x_i(0) + \epsilon \sum_{j=1}^{m} V_{ij} x_j(0) t + \epsilon \sum_{j=m+1}^{N} \frac{e^{\lambda_j t} - 1}{\lambda_j} V_{ij} x_j(0) + O(\epsilon^2),$$

where we substituted $x_i(0) = x_i^{(0)}(0) + \epsilon x_i^{(1)}(0) + O(\epsilon^2)$ and $\lambda_i = 0$. Since the system has one stable steady state solution the series above must converge for all times. The first two terms

in the expansion above are equal to the first two terms in the expansion of the solution (14) for the truncated system. Therefore, for $y_i(0)=x_i(0)$ it is

$$|x_i(t)-y_i(t)|=\epsilon\left|\sum_{j=m+1}^{N}\frac{e^{\lambda_j t}-1}{\lambda_j}V_{ij}x_j^{(0)}(0)\right|+O(\epsilon^2).$$

In the expression above all $\lambda_j<0$, therefore $|x_i(t)-y_i(t)|=O(\epsilon)$ holds for all $t>0$. Since the expression for $x_i(t)$ is convergent series and $x_i(0)$ are linearly independent, we conclude that $y_i(t)$ must also have a fixed point solution, which is $O(\epsilon)$ close to the solution of the full system.

Next we consider equations in (A1) where $i>m$. The solution to these can be expanded in terms of ϵ as

$$x_i(t)=e^{\lambda_i t}x_i(0)+\epsilon\frac{e^{\lambda_i t}-1}{\lambda_i}\sum_{j=1}^{m}V_{ij}x_j(0)+\epsilon te^{\lambda_i t}\sum_{j=m+1}^{N}V_{ij}x_j(0)+O(\epsilon^2).$$

Our truncation algorithm (Sec. III A) sets all $y_i(t)\equiv 0$, so initially the difference between full and truncated solution is whatever the initial condition $x_i(0)$ is, and it can be larger than $O(\epsilon)$. However, in the limit case

$$\lim_{t\to\infty}|x_i(t)|=\epsilon\left|\frac{1}{\lambda_i}\sum_{j=1}^{m}V_{ij}x_j^{(0)}(0)\right|+O(\epsilon^2). \tag{A7}$$

That means the truncation introduces $O(\epsilon)$ error to the asymptotic solution. Larger errors may occur only during the finite transient time $0<t<T(\epsilon)$, where $T(\epsilon)$ is given in (19). One can verify this by substituting the right hand side of (19) for time in the solution $x_{i>m}(t)$ above.

[1] N. G. V. Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier, New York, 2001).
[2] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[3] B. Munsky and M. Khammash, J. Chem. Phys. **124**, 044104 (2006).
[4] B. Munsky, A. Hernday, D. Low, and M. Khammash, in Proceedings of FOSBE (University of California, Santa Barbara, CA, 2005), pp. 145–148.
[5] B. Munsky and M. Khammash, in Proceedings of the 14th Mediterranean Conference on Control and Automation MED06 (World Scientific Publishing, 2006).
[6] B. Munsky and M. Khammash, in Proceedings of the 45th IEEE Conference on Decision and Control (2006), to be published.
[7] H. A. Simon and A. Ando, Econometrica **29**, 111138 (1961).
[8] R. G. Phillips and P. V. Kokotovic, IEEE Trans. Autom. Control **26**, 1087 (1981).
[9] S. Dey and I. Mareels, IEEE Trans. Signal Process. **52**, 1242 (2004).
[10] M. A. Gallivan and R. M. Murray, Int. J. Robust Nonlinear Control **14**, 113 (2004).
[11] L. A. Bunimovich and M. F. Demers, J. Stat. Phys. **120** (2005).
[12] J. Keizer, *Statistical Thermodynamics of Nonequilibrium Processes* (Springer-Verlag, New York, 1987).
[13] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, 2nd ed. (Cambridge University Press, Cambridge, 2005).
[14] P. L'Ecuyer, in *Handbook on Simulation*, edited by J. Banks (Wiley, New York, 1998), Chap. 4, pp. 93–137.
[15] M. Lüscher, Comput. Phys. Commun. **79**, 100 (1994).
[16] C. V. Rao and A. P. Arkin, J. Chem. Phys. **118**, 4999 (2003).
[17] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **122**, 014116 (2005).
[18] Y. N. Kaznessis, Chem. Eng. Sci. **61**, 940 (2006).
[19] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[20] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[21] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, J. Chem. Phys. **119**, 12784 (2003).
[22] A. Chatterjee, K. Mayawala, J. S. Edwards, and D. G. Vlachos, Bioinformatics **21**, 21362137 (2005).
[23] C. Moler and C. V. Loan, SIAM Rev. **45**, 3 (2003).
[24] H. E. Samad, H. Kurata, J. C. Doyle, C. A. Gross, and M. Khammash, Proc. Natl. Acad. Sci. U.S.A. **102**, 2736 (2005).
[25] H. E. Samad, M. Khammash, L. Petzold, and D. T. Gillespie, Int. J. Robust Nonlinear Control **15**, 691 (2005).
[26] A. D. Hernday, M. Krabbe, B. A. Braaten, and D. A. Low, Proc. Natl. Acad. Sci. U.S.A. **99**, 16470 (2002).
[27] A. D. Hernday, B. A. Braaten, and D. A. Low, Mol. Cell **12**, 947 (2003).
[28] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Comput. Phys. **206**, 395 (2005).
[29] J. Goutsias,J. Chem. Phys. **122**, 184102 (2005).
[30] A. Samant and D. G. Vlachos, J. Chem. Phys. **123**, 144114 (2005).
[31] E. L. Haseltine and J. B. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[32] H. Salis and Y. Kaznessis, J. Chem. Phys. **122**, 054103 (2005).
[33] H. Salis and Y. Kaznessis, J. Chem. Phys. **123**, 214106 (2005).
[34] T. J. Kaper, Proc. Symp. Appl. Math. **56**, 85 (1999).
[35] T. Kato, *Perturbation Theory for Linear Operators* (Springer-Verlag, Berlin, 1980).

[36] We shall assume through the rest of the paper that N is finite. For systems in which N is infinite, the finite state projection could be applied to find a finite systems, which approximates the original to an arbitrary degree of accuracy.