Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates

Weinan E[a)]
Department of Mathematics and Program in Applied and Computational Mathematics (PACM), Princeton University, Princeton, New Jersey 08544

Di Liu[b)] and Eric Vanden-Eijnden[c)]
Courant Institute of Mathematical Sciences, New York University, New York, New York 10012

(Received 2 May 2005; accepted 12 September 2005; published online 14 November 2005)

An efficient simulation algorithm for chemical kinetic systems with disparate rates is proposed. This new algorithm is quite general, and it amounts to a simple and seamless modification of the classical stochastic simulation algorithm (SSA), also known as the Gillespie [J. Comput. Phys. **22**, 403 (1976); J. Phys. Chem. **81**, 2340 (1977)] algorithm. The basic idea is to use an outer SSA to simulate the slow processes with rates computed from an inner SSA which simulates the fast reactions. Averaging theorems for Markov processes can be used to identify the fast and slow variables in the system as well as the effective dynamics over the slow time scale, even though the algorithm itself does not rely on such information. This nested SSA can be easily generalized to systems with more than two separated time scales. Convergence and efficiency of the algorithm are discussed using the established error estimates and illustrated through examples. © *2005 American Institute of Physics*. [DOI: 10.1063/1.2109987]

I. INTRODUCTION

This paper addresses two important issues on stochastic models of chemical kinetic systems with disparate rates, namely, effective models over the slow time scale and efficient simulation algorithms. These issues have received a great deal of attention in recent years, and much progress has been made.[1–5] One main idea, pursued by many authors,[2,4,5] is to make a quasiequilibrium approximation for the fast processes and solve the empirically averaged slow processes. Even though this has been a common theme in much of the recent works on this subject, it seems that the key issue of identifying the effective process and effective variables over different time scales has not been fully understood. As a result, the validity and generality of these approaches have not been satisfactorily demonstrated.

The present paper has two purposes. The first is to provide effective models and effective variables for stochastic chemical kinetic systems with disparate rates. Compared with previous works, such as Refs. 4 and 5 which rely on the identification of slow and fast species, our work gives a general prescription for identifying effective slow variables. As we will see later, in general the slow variables are not associated with specific species, but are rather combinations of the molecular numbers of several species.

The second purpose of this paper is to suggest an efficient stochastic simulation algorithm for systems with disparate rates in the style of the well-known Gillespie algorithm.[6,7] What should be noted is the simplicity of this algorithm, it is completely seamless. Even though understanding the effective dynamics on the slow time scale requires knowing the slow variables, the algorithm itself does *not* use this information. It only requires knowing the slow and fast reactions. We also discuss the convergence and efficiency of the algorithm based on the error estimates that we have established.

As mentioned above, some recent efforts have been made for the simulation of chemical kinetic systems with disparate rates. In Ref. 1 a multiscale simulation method was proposed in which the slow and fast reactions are simulated separately at each event of the slow reactions. The slow reactions are simulated with Gillespie's algorithm and the fast reactions are simulated with a Langevin dynamics, assuming the number of the molecules involved is so large that the kinetic dynamics converges to a diffusion process. In Ref. 3 a similar multiscale scheme is proposed in which the fast dynamics is simulated with deterministic ordinary differential equations. The approaches in Refs. 1 and 3 require that the volume of the system be sufficiently large in addition to having well-separated rates. In Ref. 2 a scheme based on the quasiequilibrium assumption is proposed, assuming that the probability density of the fast processes conditioned on the slow processes is known exactly or can be approximated, e.g., by a Gaussian. The same quasiequilibrium assumption is used in Refs. 4 and 5 except that the probability density of the fast processes conditioned on the slow processes is computed via a modified process called the virtual fast process. The method proposed in Refs. 4 and 5 is more general than previous methods, but it still has the following limitations. It applies only to cases when the slow and fast processes are associated with specific species; the rate functions of the slow processes are assumed to be simple and are approxi-

[a)]Electronic mail: weinan@princeton.edu
[b)]Electronic mail: diliu@cims.nyu.edu
[c)]Electronic mail: eve2@cims.nyu.edu

mated empirically by solving a system of algebraic equations. In contrast, our work relies only on the disparity of the rates, and makes no *a priori* assumption on what the slow and fast variables are, or the analytic form of the rate functions.

In the following we first recall Gillespie's algorithm, the stochastic simulation algorithm (SSA) for modeling the general chemical kinetic systems, and introduce the nested SSA for systems with fast and slow rates. Afterwards we discuss the effective process on the slow time scale and the identification of the slow variables. This discussion also helps us to understand why and how the nested SSA works for systems with disparate rates. We illustrate the efficiency of the new algorithm with the example of the stochastic Petri net model for the heat shock response of *E. Coli* proposed in Ref. 8. Finally, we discuss straightforward generalizations of the method to systems involving more than two time scales, e.g., with ultrafast, fast, and slow rates, and to systems in which the grouping into fast and slow reactions changes dynamically. These generalizations are also tested against numerical examples to show their efficiency.

We note that the nested SSA is in the same spirit as the algorithms proposed in Ref. 9 and further developed in Refs. 10 and 11 in the context of stochastic differential equations that involve different time scales. In the context of deterministic differential equations, similar ideas are developed in Refs. 12 and 13. More generally, ideas of this kind are discussed for a variety of other applications in Ref. 14.

II. A NESTED STOCHASTIC SIMULATION ALGORITHM

A. The stochastic simulation algorithm

We begin with the general setup. Assume that a well-mixed, isothermal system has N_S species of molecules S_i, $i=1,\ldots,N_S$, and there are M_R reaction channels R_j, $j=1,\ldots,M_R$. Let x_i be the number of molecules of species S_i. Then the state of the system is given by

$$x=(x_1,\cdots,x_{N_R})\in X, \tag{1}$$

where X denotes the state space. Each reaction R_j is characterized by a rate function $a_j(x)$ and a vector ν_j that describes the change of the state due to the reaction. We write

$$R_j=(a_j,\nu_j). \tag{2}$$

The dynamics of the model is completely specified by the following rules.

(1) Given the state x, the reactions are independent of each other on an infinitesimal time interval of duration dt and the probability for the reaction R_j to happen is given by $a_j(x)dt$.
(2) The state of the system after R_j is given by $x+\nu_j$.

These rules specify a Markov process on the state space X with generator

$$L(x,y)=\sum_{j=1}^{M_R} a_j(x)\,\delta_{y=x+\nu_j}, \tag{3}$$

for $x\neq y$ and $L(x,x)=-\Sigma_{y\neq x}L(x,y)$.

The standard computer implementation of such a model is given by the well-known SSA proposed in Refs. 6 and 7 (see also Ref. 15). Let

$$a_0(x)=\sum_{j=1}^{M_R} a_j(x). \tag{4}$$

Assume that the current time is $t=t_n$, and the state of the system is $x=x_n$. The essential steps of the SSA are the following.

(1) Generate independent random numbers r_1 and r_2 with uniform distribution on the unit interval. Let

$$\delta t_{n+1}=-\frac{\ln r_1}{a_0(x)}, \tag{5}$$

and k_{n+1} to be the natural number, such that

$$\frac{1}{a_0(x)}\sum_{j=1}^{k_{n+1}-1} a_j(x)<r_2\leqslant\frac{1}{a_0(x)}\sum_{j=1}^{k_{n+1}} a_j(x). \tag{6}$$

(2) Update the time and the state of the system

$$t_{n+1}=t_n+\delta t_{n+1},\quad x_{n+1}=x_n+\nu_{k_{n+1}}. \tag{7}$$

B. A seamless algorithm for systems with disparate rates

Next we consider the case when the rates have very different magnitudes. For simplicity of discussion, we will first assume that the rates $a_j(x)$ are divided into two groups (the general case with more than two groups is treated in Sec. V A): One group corresponding to the fast processes with rates of order $1/\varepsilon$ and the other group corresponding to the slow processes with rates of order 1. Here $\varepsilon\ll 1$,

$$a(x)=(a^s(x),a^f(x)), \tag{8}$$

where

$$\begin{aligned} a^s(x)&=(a_1^s(x),\ldots,a_{M_s}^s(x))=O(1),\\ a^f(x)&=(a_1^f(x),\ldots,a_{M_f}^f(x))=O(1/\varepsilon), \end{aligned} \tag{9}$$

in dimensionless units. The corresponding reactions and the associated state change vectors can be grouped accordingly as

$$R^s=(a^s,\nu^s),\quad R^f=(a^f,\nu^f). \tag{10}$$

As a simple example, consider the system

$$S_1\underset{a_2^f}{\overset{a_1^f}{\rightleftarrows}}S_2,\quad S_2\underset{a_2^s}{\overset{a_1^s}{\rightleftarrows}}S_3,\quad S_3\underset{a_4^f}{\overset{a_3^f}{\rightleftarrows}}S_4, \tag{11}$$

with reaction channels given by

$$
\begin{aligned}
&a_1^f = 10^5 x_1, \quad \nu_1^f = (-1,1,0,0);\\
&a_2^f = 10^5 x_2, \quad \nu_2^f = (1,-1,0,0);\\
&a_3^f = 10^5 x_3, \quad \nu_3^f = (0,0,-1,1);\\
&a_4^f = 10^5 x_4, \quad \nu_4^f = (0,0,1,-1);\\
&a_1^s = x_2, \quad \nu_1^s = (0,-1,1,0);\\
&a_2^s = x_3, \quad \nu_2^s = (0,1,-1,0).
\end{aligned}
\tag{12}
$$

Let us take ε to be 10^{-5}, for this example. There are a total of four species and six reaction channels, with four fast reactions and two slow ones.

For problems of this type, direct application of the SSA will result in time steps of size ε [the total rate $a_0(x)$ in (4) is of order $1/\varepsilon$] with a total cost of order $1/\varepsilon$ if we want to advance the whole system through a time interval of order unity. Most of the cost will be spent on the fast reactions, which are often of little interest. Indeed for such systems, we are usually interested in the slow processes since they are the rate-limiting steps. Here we propose a modified SSA that captures the slow processes at a cost that is independent of ε, and therefore much less than that of the direct SSA when $\varepsilon \ll 1$. The modified SSA consists of two nested SSA: The outer SSA is on the slow processes only, but uses modified slow rates. The inner SSA is on the fast processes only: it uses the original fast rates and serves to give the modified slow rates. Let $t=t_n$, $x=x_n$ be the current time and state of the system, respectively. Given (t_n,x_n), do the following.

(1) Inner SSA. Run N-independent replicas of SSA with the fast reactions $R^f=(a^f,\nu^f)$ only, for a time interval of T_f. During this calculation, compute the modified slow rates. For $j=1,\ldots,M_s$, these are

$$\tilde{a}_j^s = \frac{1}{N}\sum_{k=1}^{N} \frac{1}{T_f}\int_{T_0}^{T_f+T_0} a_j^s(x_k(\tau))d\tau, \tag{13}$$

where $x_k(\tau)$ is the kth replica of the auxiliary fast process at virtual time τ whose initial value is $x_k(0)=x_n$, and T_0 is a parameter we choose in order to minimize the effect of the transients in the auxiliary fast process. The auxiliary fast process is the same as the virtual fast process defined in Ref. 5 and we will refer to it as such.

(2) Outer SSA. Run one step of SSA for the modified slow reactions $\tilde{R}^s=(\tilde{a}^s,\nu^s)$ to generate (t_{n+1},x_{n+1}) from (t_n,x_n).

Then repeat.

Note that we can take $N=1$ in the above algorithm, but it is advantageous to take a larger N since the inner SSA can be trivially parallelized.

A key issue is how long one should run the inner SSA (how big T_f should be) and how big the error is. Another important issue is the total cost of this algorithm. We will discuss these issues later. But before doing so, let us note that the algorithm as presented is completely seamless and general. We do not need to know what the slow and fast variables are and certainly we do not need to make analytical approximations to get the effective slow rates.

III. SLOW VARIABLES AND EFFECTIVE DYNAMICS

Even though the formulation of the nested SSA does not require explicit identification of the slow variables and the effective rates of the slow process, to understand why and how the algorithm works, we do need to understand these issues.

A. Identification of the slow variables

First, we discuss the observables, which are functions of the state variable x. By definition, slow observables are conserved quantities during the fast reactions, i.e., $\upsilon(x)$ is a slow observable if for any $x\in X$ and any state change vector ν^f associated with the fast reactions one has

$$\upsilon(x+\nu^f)=\upsilon(x). \tag{14}$$

This means that the value of the slow observable $\upsilon(x)$ is unaffected by the fast reactions. To find a general representation of such observables, we consider special slow observables which are linear functions that satisfy (14). We call such slow observables slow variables. It is easy to see that $\upsilon(x)=b\cdot x$ is a slow variable if

$$b\cdot\nu^f=0, \tag{15}$$

for all ν^f. Let $b_1,b_2,\ldots,b_J$ be a set of basis vectors that satisfy (15). Define

$$y_j=b_j\cdot x \quad \text{for} \quad j=1,\ldots,J. \tag{16}$$

Then $y_1,y_2,\ldots,y_J$ defines a complete set of slow variables, i.e., all slow observables can be expressed as functions of $y_1,y_2,\ldots,y_J$. For the example considered in Sec. II B, it is easy to see that both x_1+x_2 and x_3+x_4 are conserved during the fast reactions, i.e., $y_1=x_1+x_2$ and $y_2=x_3+x_4$ are the slow variables of that system. Note that for this particular example, every species is involved in at least one fast reaction. Therefore there is no slow species in this example. This means that the methodology proposed in Refs. 4 and 5 would not result in any changes over the straightforward SSA.

B. Effective dynamics on slow time scales

We can now put the quasiequilibrium assumption in precise terms. Fix the slow variables y and consider the virtual fast process defined by retaining only the fast reaction channels. Two important conclusions can be drawn for this system. The first is that the slow variables do not change. The second is that this system approaches a unique equilibrium state (which depends on the value of y) on a time scale of order ε. This equilibrium is the desired quasiequilibrium, which we denote by $\mu_y(x)$. The rates for the effective slow process are obtained by averaging the slow rates with respect to this quasiequilibrium

$$\bar{a}_j(y)=\langle a_j(x)\rangle_y \equiv \sum_{x\in X} a_j(x)\mu_y(x). \tag{17}$$

It is obvious that the effective rates are only functions of y. The effective dynamics is completely specified by

$$\bar{R} = (R^s, \bar{a}(y)). \tag{18}$$

It is shown in Ref. 16 (see also Ref. 17) by singular perturbation analysis that the original dynamics converges to the above effective dynamics with an error of order $O(\varepsilon)$. A formal derivation is also provided in Ref. 2, assuming the slow variables coincide with some of the reacting species.

C. Convergence and efficiency of the nested SSA

If we knew $\bar{a}(y)=(\bar{a}_1(y),\ldots,\bar{a}_{M_s}(y))$ explicitly, we could have carried out SSA using these rates. This would capture the process on the slow time scale, which is what we are interested in. For convenience, we will call such a procedure "averaged SSA." Unfortunately we usually do not have an explicit expression for the effective rates (17). The nested SSA proposed above is a way of getting approximate values of these rates "on the fly."

To see why this algorithm should work, it is clear that the only difference between the nested SSA and the averaged SSA is that the averaged SSA uses the rates in (17), whereas the nested SSA uses the rates in (13). However, by ergodicity we have that $\tilde{a}$ converges to $\bar{a}$ when T_f and N go to infinity. Therefore the results of the two algorithms also become closer and closer for large T_f and N. Quantitative error estimates can be obtained. The details are given in Ref. 16. Among other things, it is proved in Ref. 16 that

$$\mathbb{E}|\bar{a}_j - \tilde{a}_j| \leqslant C\left(\frac{e^{-\alpha T_0/\varepsilon}}{1+T_f/\varepsilon} + \frac{1}{\sqrt{N(1+T_f/\varepsilon)}}\right), \tag{19}$$

for some constants C and α which are independent of ε. Here $\mathbb{E}$ denotes expectation with respect to the statistics of the virtual fast process in the inner SSA. The first term on the right-hand side of (19) measures the deviation from the quasiequilibrium if the inner SSA is run only for a time duration of T_f, starting at T_0. α measures the rate of convergence for the virtual fast process to equilibrium. The second term measures the sampling error resulted from using time and ensemble averaging on a time interval of duration T_f with an ensemble of N replicas.

Let us now estimate the cost of the nested SSA. For simplicity we will take $T_0=0$ here and also in our numerical examples. One feature of (19) is that this estimate depends on the ratio T_f/ε rather than T_f alone. This means that, when $\varepsilon \ll 1$, we can achieve a small error on $\tilde{a}_j$ by choosing $T_f/\varepsilon \gg 1$ and yet have $T_f \ll 1$ (remember that we have assumed that the time scale for the slow process is of order 1). This is the very reason why the nested SSA is more efficient than a direct SSA. To quantify this, suppose that we want to compute the results within an error tolerance λ. To control each term in (19) by λ, the optimal choice of parameters is

$$N = T_f/\varepsilon = 1/\lambda^2. \tag{20}$$

Then the cost for evolving the nested SSA for a unit time is estimated to be

$$\text{cost} = \tilde{a}_0^s \times N a_0^f T_f/\epsilon = O(1/\lambda^2), \tag{21}$$

which is independent of ε

TABLE I. List of species and their initial value (in number of molecules) in the Petri net model of heat shock response of *E. Coli* proposed in. Ref. 8.

Species	Initial value
DNA.σ^{32}	1
mRNA.σ^{32}	17
σ^{32}	15
RNAPσ^{32}	76
DNA.DnaJ	1
DNA.FtsH	0
DNA.GroEL	1
DnaJ	4640
FtsH	200
GroEL	4314
DnaJ.Unfoldedprotein	5×10^6
Protein	5×10^6
σ^{32}.DnaJ	2959
Unfoldedprotein	2×10^5

IV. NUMERICAL EXAMPLE: HEAT SHOCK RESPONSE OF *E. Coli*

As an example, we will study a stochastic Petri net model proposed in Ref. 8 to quantify the response of the bacteria *E. Coli* to a sudden temperature increase. This response is a complicated mechanism of protection that the bacteria uses to fight against denaturation (unfolding) of its constituent proteins induced by the increase of temperature. This response mechanism is referred to as the heat shock response. The stochastic Petri net model of the heat shock response proposed in Ref. 8 involves 14 species of molecules, as given in Table I together with their initial conditions. These 14 species are involved in 17 reactions as specified in Table II.

The reaction rate for each reaction is given by the product of a rate constant and the molecular numbers of the reactants. The rate constants are listed in the middle column of Table II. For the second-order reactions, the rate constants are already normalized with the Avogadro constant and cell volume $V=1.5\times10^{-15}$ liter. The typical magnitudes of the corresponding reaction rates $a_i(x)$ are also listed in the right column. The rates and initial values are chosen as in Ref. 3 except for the initial value of DnaJ which is chosen to be further away from equilibrium to make our test more stringent.

We can see that the last three reactions are much faster than the others. The time scales of the fast and slow reactions differ by about five orders of magnitude ($\varepsilon=10^{-5}$). For this particular example, the slow variables are the reactants in the slow reactions (the first 14 reactions in Table II) except DnaJ. The fast variable DnaJ shows up in the coefficients of the slow reactions marked with ($\star\star$). According to (18), the effective dynamics on the slow time scale is given by averaging the coefficients of the corresponding slow reactions with respect to the equilibrium of the fast reactions, i.e., one must use the averaged value $\langle \text{DnaJ} \rangle_y$ instead of DnaJ in the reactions marked with ($\star\star$). This average is computed using an inner SSA which involves only the last three reactions listed in Table II, and it is used in an outer SSA which involves the first 14 reactions in the table.

TABLE II. Reaction list for the heat shock model of *E. Coli* proposed in. Ref. 8. The rate constant is the number c_i in $a_i(x)=c_i x_j$ for the reactions involving one species, or in $a_i(x)=c_i x_j x_k$ for the reactions involving two species. The rate magnitude is the value of $a_i(x)$ evaluated at initial time or equilibrium. The last three reactions marked with a ($\star$) in the table are fast: they are used in the inner SSA. All the other reactions are used in the outer SSA, and the rates of the reactions marked with a ($\star\star$) are averaged according to (17).

Reaction	Rate constant	Rates magnitude
$\mathrm{DNA}.\sigma^{32}\rightarrow \mathrm{mRNA}.\sigma^{32}$	1.4×10^{-3}	1.4×10^{-3}
$\mathrm{mRNA}.\sigma^{32}\rightarrow \sigma^{32}+\mathrm{mRNA}.\sigma^{32}$	0.07	1.19
$\mathrm{mRNA}.\sigma_{32}\rightarrow$ degradation	1.4×10^{-6}	2.38×10^{-5}
$\sigma_{32}\rightarrow \mathrm{RNAP}\sigma^{32}$	0.7	10.5
$\mathrm{RNAP}\sigma^{32}\rightarrow \sigma^{32}$	0.13	9.88
$\sigma^{32}+\mathrm{DnaJ}\rightarrow \sigma^{32}.\mathrm{DnaJ}$ ($\star\star$)	3.62×10^{-3}	25.2
$\mathrm{DnaJ}\rightarrow$ degradation ($\star\star$)	6.4×10^{-10}	2.97×10^{-6}
$\sigma^{32}.\mathrm{DnaJ}\rightarrow \sigma^{32}+\mathrm{DnaJ}$	4.4×10^{-4}	1.30
$\mathrm{DNA}.\mathrm{DnaJ}+\mathrm{RNAP}\sigma^{32}\rightarrow \mathrm{DnaJ}+\mathrm{DNA}.\mathrm{DnaJ}+\sigma^{32}$	8	3.71
$\mathrm{DNA}.\mathrm{FtsH}+\mathrm{RNAP}.\sigma^{32}\rightarrow \mathrm{FtsH}+\mathrm{DNA}.\mathrm{FtsH}+\sigma^{32}$	4.88×10^{-2}	0
$\mathrm{FtsH}\rightarrow$ degradation	7.4×10^{-11}	1.48×10^{-8}
$\mathrm{GroEL}\rightarrow$ degradation	1.8×10^{-8}	7.76×10^{-5}
$\sigma^{32}.\mathrm{DnaJ}+\mathrm{FtsH}\rightarrow \mathrm{DnaJ}+\mathrm{FtsH}$	1.42×10^{-5}	8.4
$\mathrm{DNA}.\mathrm{GroEL}+\mathrm{RNAP}\sigma^{32}\rightarrow \mathrm{GroEL}+\mathrm{DNA}.\mathrm{GroEL}+\sigma^{32}$	0.063	4.78
$\mathrm{Protein}\rightarrow \mathrm{Unfoldedprotein}$ ($\star$)	0.2	10^6
$\mathrm{DnaJ}+\mathrm{Unfoldedprotein}\rightarrow \mathrm{DnaJ}.\mathrm{Unfoldedprotein}$ ($\star$)	0.108	10^7
$\mathrm{DnaJ}.\mathrm{Unfoldedprotein}\rightarrow \mathrm{DnaJ}+\mathrm{Unfoldedprotein}$ ($\star$)	0.2	10^6

To test the efficiency of the nested SSA and compare it to a direct SSA, we use the mean value and the variance at time $T=10$ of the heat shock sigma factor, σ^{32}, as a benchmark. A computation of this average by a direct SSA using $N_0=1000$ realizations led to

$$\overline{\sigma^{32}} = 14.8 \pm 0.1, \quad \mathrm{var}(\sigma^{32}) = 14.2 \pm 0.1. \tag{22}$$

This calculation took 19 719.2 s of CPU time on our machine. Notice that these expectations must be computed by ensemble average (not time average) since the system is out of equilibrium.

To test the nested SSA, first, we make a series of simulations in which we choose the size of the ensemble and the simulation time of the inner SSA in the nested SSA scheme according to

$$(N, T_f/\varepsilon) = (1, 2^{2k}/10), \tag{23}$$

for different values of $k=0,1,2,\ldots$. The error estimate in (19) then implies that the error λ of the nested SSA should decay with rate

$$\lambda = O(2^{-k}). \tag{24}$$

Table III gives the total CPU time and the values of $\overline{\sigma^{32}}$ and $\mathrm{var}(\sigma^{32})$ obtained by the nested SSA with the parameters in (23) and using $N_0=1000$ realizations of the outer SSA (same as in the direct SSA). The relative errors on $\overline{\sigma^{32}}$ are shown in Fig. 1.

TABLE III. Efficiency of nested SSA when $N=1$. Since we used $N_0=1000$ realizations of the outer SSA to compute $\overline{\sigma^{32}}$ and $\mathrm{var}(\sigma^{32})$, the statistical errors on these quantities is about 0.1.

$(N, T_f/10^{-6})$	(1,1)	(1,4)	(1,16)	(1,64)	(1,256)	(1,1024)
CPU	0.62	1.32	2.98	9.56	35.81	142.08
$\overline{\sigma^{32}}$	4.60	8.66	13.60	14.52	14.98	15.00
$\mathrm{var}(\sigma^{32})$	4.41	8.11	12.22	13.13	13.73	14.66

Obviously the results produced by the first couple of tests shown in Table III are very far off from the correct value. But we can see that the last several values are close to the correct value, with an error that is comparable to the error bar in (22), and the cost is a small fraction of the cost of direct SSA. The nested SSA is also able to track the time evolution of the variables in a given realization, as shown in Fig. 2 where the growth of the number of molecules of GroEL computed with the nested SSA is compared with the growth predicted by the direct SSA.

Next, we make a second series of simulations in which we choose the size of the ensemble and the simulation time of the inner SSA in the nested SSA scheme as

$$(N, T_f/\varepsilon) = (2^k, 2^k/10), \quad k = 0,1,2,\ldots. \tag{25}$$

The error estimate (19) then implies that the error λ of the nested SSA should also decay with a rate of the form of (24). Table IV gives the total CPU time for the nested SSA with the parameters in (25), and the value of $\overline{\sigma^{32}}$ and $\mathrm{var}(\sigma^{32})$ obtained. The relative errors on $\overline{\sigma_{32}}$ is shown in Fig. 1.

V. GENERALIZATIONS

The nested SSA proposed in Sec. II B can be straightforwardly generalized to systems involving more than two separate time scales and to systems where the grouping into fast and slow variables evolves dynamically. We consider these generalizations next.

A. Multilevel SSA

Consider a system where the rates $a_j(x)$ can be grouped into three groups: One group corresponding to ultrafast processes with rates of order $1/\delta$, one group corresponding to

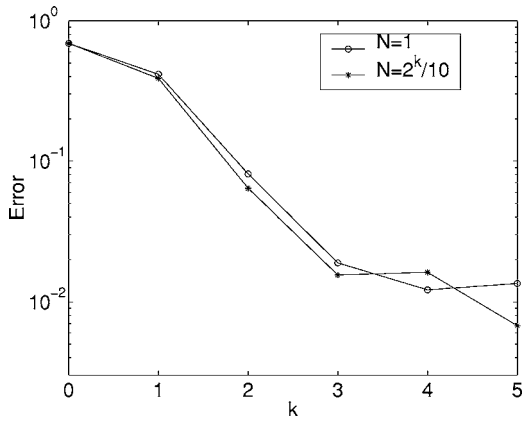

FIG. 1. Relative errors on $\overline{\sigma_{32}}$ computed by the nested SSA with different N and T_f. Notice that the error saturate, not because the error estimate (19) fails, but because we reach statistical accuracy on the computation of $\overline{\sigma^{32}}$ and $\mathrm{var}(\sigma^{32})$ with $N_0=1000$ realizations.

fast processes with rates of order $1/\varepsilon$, and one group corresponding to slow processes with rates of order 1. Here $\delta \ll \varepsilon \ll 1$,

$$a(x) = (a^s(x), a^f(x), a^{uf}(x)), \tag{26}$$

where

$$\begin{aligned} a^s(x) &= (a_1^s(x), \ldots, \quad a_{M_s}^s(x)) = O(1), \\ a^f(x) &= (a_1^f(x), \ldots, \quad a_{M_f}^f(x)) = O(1/\varepsilon), \\ a^{uf}(x) &= (a_1^{uf}(x), \ldots, \quad a_{M_{uf}}^{uf}(x)) = O(1/\delta). \end{aligned} \tag{27}$$

The corresponding reactions and the associated state change vectors can then be grouped accordingly as

$$R^s = (a^s, \nu^s), \quad R^f = (a^f, \nu^f), \quad R^{uf} = (a^{uf}, \nu^{uf}), \tag{28}$$

and it is easy to see that this case can be handled by using a nested SSA with three levels. The innermost SSA uses only the ultrafast rates and serves to compute averaged fast and slow rates using a formula similar to (13); the inner SSA uses only the averaged fast rates and the results are used again to compute the average slow rates (which are already averaged with respect to the ultrafast reactions) as in (13); finally, the outer SSA uses only the averaged slow rates. The cost of such a nested SSA is independent of δ and ε, and as before, precise error estimates can be obtained in terms of T_{uf} (the time interval over which the innermost SSA is run), N_{uf} (the number of replicas in the innermost SSA), T_f (the time interval over which the inner SSA is run), and N_f (the number of replicas in the inner SSA). The generalization to systems involving more groups of separated rates is straightforward and simply amounts to using more levels in the nested SSA.

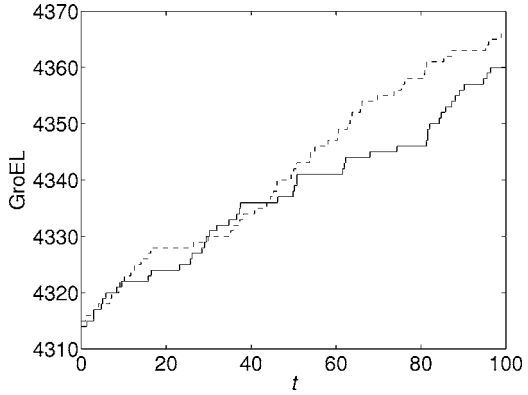

FIG. 2. Time evolution in one realization of the number of molecules of GroEL computed by the direct SSA algorithm (solid line) and the nested SSA (dashed line).

TABLE IV. Efficiency of nested SSA with multiple replicas in the inner SSA. Again the statistical errors on $\overline{\sigma^{32}}$ and $\mathrm{var}(\sigma^{32})$ is about 0.1.

$(N, T_f/10^{-6})$	(1,1)	(2,2)	(4,4)	(8,8)	(16,16)	(32,32)
CPU	0.64	1.38	3.17	10.13	36.94	142.65
$\overline{\sigma_{32}}$	4.60	9.06	13.85	14.57	15.04	14.90
$\mathrm{var}(\sigma_{32})$	4.41	8.68	13.07	13.63	14.01	14.38

As an example, consider the system

$$\begin{aligned} &S_1 \underset{a_2^{uf}}{\overset{a_1^{uf}}{\rightleftarrows}} S_2, \quad S_3 \underset{a_4^{uf}}{\overset{a_3^{uf}}{\rightleftarrows}} S_4, \quad S_5 \underset{a_6^{uf}}{\overset{a_5^{uf}}{\rightleftarrows}} S_6, \quad S_7 \underset{a_8^{uf}}{\overset{a_7^{uf}}{\rightleftarrows}} S_8, \\ &S_2 \underset{a_2^{f}}{\overset{a_1^{f}}{\rightleftarrows}} S_3, \quad S_6 \underset{a_4^{f}}{\overset{a_3^{f}}{\rightleftarrows}} S_7, \quad S_3 \underset{a_2^{s}}{\overset{a_1^{s}}{\rightleftarrows}} S_6, \end{aligned} \tag{29}$$

with the ultrafast reaction channels given by

$$\begin{aligned} a_1^{uf} &= 2\times 10^{12} x_1, \quad \nu_1^{uf} = (-1,1,0,0,0,0,0,0), \\ a_2^{uf} &= 3\times 10^{12} x_2, \quad \nu_2^{uf} = (1,-1,0,0,0,0,0,0), \\ a_3^{uf} &= 2\times 10^{12} x_3, \quad \nu_3^{uf} = (0,0,-1,1,0,0,0,0), \\ a_4^{uf} &= 3\times 10^{12} x_4, \quad \nu_4^{uf} = (0,0,1,-1,0,0,0,0), \\ a_5^{uf} &= 2\times 10^{12} x_5, \quad \nu_5^{uf} = (0,0,0,0,-1,1,0,0), \\ a_6^{uf} &= 3\times 10^{12} x_6, \quad \nu_6^{uf} = (0,0,0,0,1,-1,0,0), \\ a_7^{uf} &= 2\times 10^{12} x_7, \quad \nu_7^{uf} = (0,0,0,0,0,0,-1,1), \\ a_8^{uf} &= 3\times 10^{12} x_8, \quad \nu_8^{uf} = (0,0,0,0,0,0,1,-1), \end{aligned} \tag{30}$$

the fast reaction channels given by

TABLE V. Efficiency and accuracy of the adaptive nested SSA.

$T_f/10^{-5}$	1	4	16	64	256
CPU	298.2	304.4	310.3	347.9	426.7
$\overline{x_1}$	27.07	27.10	27.28	27.20	27.32
$\mathrm{var}(x_1)$	20.03	20.25	20.04	20.22	20.57

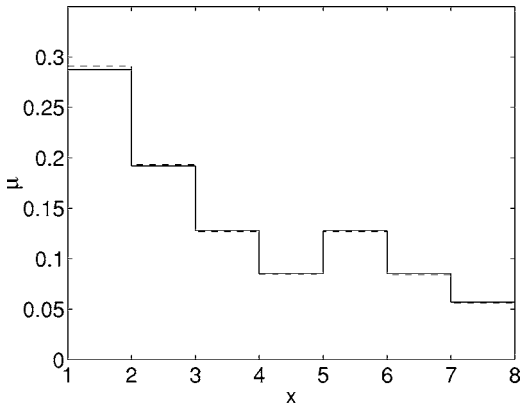

FIG. 3. The exact equilibrium distribution of example (29) (solid line) is compared to the one estimated by the nested SSA (dashed line). The two histograms can barely be distinguished.

$$
\begin{aligned}
&a_1^f = 2\times 10^6 x_2, \quad \nu_1^f = (0,-1,1,0,0,0,0,0),\\
&a_2^f = 3\times 10^6 x_3, \quad \nu_2^f = (0,1,-1,0,0,0,0,0),\\
&a_3^f = 2\times 10^6 x_6, \quad \nu_3^f = (0,0,0,0,0,-1,1,0),\\
&a_4^f = 3\times 10^6 x_7, \quad \nu_4^f = (0,0,0,0,0,1,-1,0),
\end{aligned}
\tag{31}
$$

and the slow reaction channels given by

$$
\begin{aligned}
&a_1^s = 2\times x_3, \quad \nu_1^s = (0,0,-1,0,0,1,0,0),\\
&a_2^s = 3\times x_6, \quad \nu_2^s = (0,0,1,0,0,-1,0,0).
\end{aligned}
\tag{32}
$$

Thus, with respect to the ultrafast reactions, the slow variables are x_1+x_2, x_3+x_4, x_5+x_6, and x_7+x_8, whereas with respect to the fast reactions, the slow variables are $x_1+x_2+x_3+x_4$ and $x_5+x_6+x_7+x_8$. Note that all variables are involved in at least one ultrafast reaction, and there is a 12 order of magnitude difference between the ultrafast rates and the slow ones ($\delta=10^{-12}$ and $\varepsilon=10^{-6}$ in this example). As a test for the nested SSA, we took as initial condition

$$
(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8) = (1,0,0,0,0,0,0,0), \tag{33}
$$

and computed the equilibrium distribution for this system. The equilibrium distribution can easily be computed exactly for this system and it is compared in Fig. 3 with the result of the nested SSA up to time $T_0=10^4$ using

$$
(N_f,N_{uf},T_f,T_{uf}) = (1,1,10^{-4},10^{-10}). \tag{34}
$$

The result in Fig. 3 shows that the time interval $[0,T_0]=[0,10^4]$ was long enough to estimate very accurately the equilibrium distribution of the system. In contrast, since the average time step in a direct SSA is of the order of $\delta=10^{-12}$, it is impossible to run the direct SSA up to time $T_0=10^4$. To compare the efficiency of the nested SSA and the direct SSA, we fixed the total number of iterations in the calculation. The calculation with the nested SSA using the parameters in (34) requires $O(10^8)$ iterations. With the same number of iterations, the direct SSA only advanced to time $T_0=O(10^{-4})$, which is way too small to produce an accurate estimate for the equilibrium distribution. In fact, the probability that the system has visited any state other than S_1, S_2, S_3, and S_4 by time T_0' is so small that this event was not observed in our simulation.

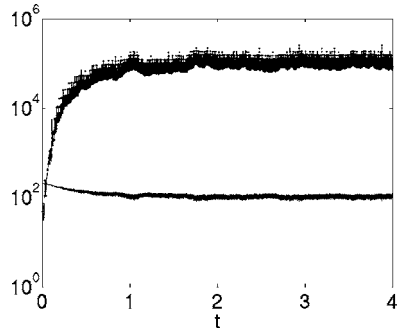

FIG. 4. The time evolution of the magnitudes of reaction rates $a_1+a_2+a_3+a_4$ (lower curve) and a_5+a_6 (upper curve). It can be seen that rates $a_1+a_2+a_3+a_4$ keeps low all the time while a_5+a_6 grows to a very high level.

B. Adaptive SSA

Consider the following reaction:

$$
S_1 \underset{a_2}{\overset{a_1}{\rightleftarrows}} S_2, \quad S_1 \underset{a_4}{\overset{a_3}{\rightleftarrows}} S_3, \quad 2S_2 + S_3 \underset{a_6}{\overset{a_5}{\rightleftarrows}} 3S_4. \tag{35}
$$

The reaction rates and the state change vectors are

$$
\begin{aligned}
&a_1 = x_1, \quad \nu_1 = (-1,+1,0,0),\\
&a_2 = x_2, \quad \nu_2 = (+1,-1,0,0),\\
&a_3 = x_1, \quad \nu_3 = (-1,0,+1,0),\\
&a_4 = x_3, \quad \nu_4 = (+1,0,-1,0),\\
&a_5 = 2x_2(x_2-1)x_3, \quad \nu_5 = (0,-1,-2,+3),\\
&a_6 = 2x_4(x_4-1)(x_4-2), \quad \nu_6 = (0,+1,+2,-3).
\end{aligned}
\tag{36}
$$

Let us choose the initial condition to be

$$
(x_1,x_2,x_3,x_3) = (100,3,3,3). \tag{37}
$$

At the beginning, when the concentrations of S_2 and S_3 are low, all the reactions are slow. As the number of S_2 and S_3 grows, the last two reactions become faster and faster. Figure 4 shows the evolution of the reactions rates. The ratio of the time scales reaches $\epsilon=10^{-4}$ during this simulation.

To account for this effect, we dynamically monitor the set of fast reactions over time. From Fig. 4, we can see that the time scale separation stabilizes when time $t>1.5$. We use direct SSA to simulate the whole system when $t\leqslant 1.5$ and use a two-level nested SSA with the last two reactions as fast reaction when $t>1.5$. For more complex examples, more sophisticated ways of adaptively choosing the fast reaction set is needed and is discussed in. Ref. 16

To test this adaptive SSA and compare it with the direct SSA, we again use the mean and the variance at time $T=4$ of x_1, the number of species S_1, as a benchmark. A computation of the direct SSA with $N_0=10\,000$ gives

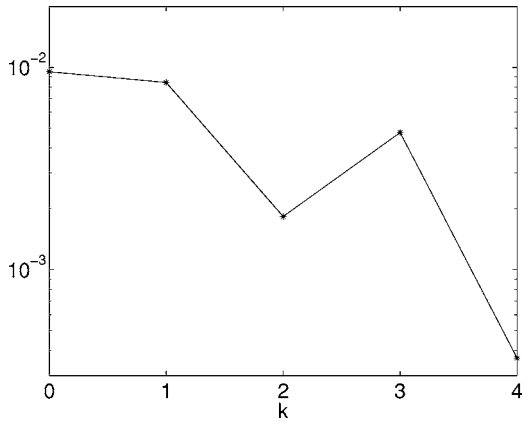

FIG. 5. The relative error on $\overline{x_1}$ in the adpative nested SSA with the parameters as in (39). The scaling of the error in consistent with (40).

$$\overline{x_1} = 27.33 \pm 0.2, \quad \mathrm{var}(x_1) = 20.47 \pm 0.2. \tag{38}$$

This calculation took 1144.9 s of CPU time on our machine. The results are shown in Table V.

To test the nested SSA, we make a series of simulations in which the size of the ensemble and simulation time of the inner SSA in the nested SSA scheme are chosen to be

$$(N, T_f) = (1, 2^{2k} \times 10^{-5}), \tag{39}$$

for different values of $k=0,1,2,\ldots$. The error is expected to be

$$\lambda = O(2^{-k}). \tag{40}$$

The following table gives the CPU times and values of the mean and variance of x_1 using $N_0=10\,000$. The relative errors of the mean are shown in the Fig. 5. The results are consistent with the examples discussed earlier.

VI. CONCLUDING REMARKS

In summary, we have presented a simple strategy for identifying the system of slow reaction dynamics in models of stochastic chemical kinetics with disparate rates. This leads to a general and seamless multiscale method that captures the dynamics of the system over the slow time scale with a much smaller cost than the direct SSA. This multiscale method is a small modification of the direct SSA, in the form of a nested SSA, with inner loops for the fast reactions, and outer loop for the slow reactions. The number of groups can be more than two, and the grouping into fast and slow reactions can be done adaptively. The efficiency of the nested SSA has been illustrated through a series of examples. The algorithm itself does not rely on undestanding the slow and fast variables, even though such an understanding is important for the analysis of the algorithm.

ACKNOWLEDGMENTS

This work was motivated by a presentation of Linda Petzold at IMA in November of 2004. One of the authors (W.E.) is partially supported by NSF via Grant No. DMS01-30107. Another author (D.L.) is partially supported by NSF via Grant No. DMS97-29992. One of the authors (E.V.-E.) is partially supported by NSF via Grant Nos. DMS02-09959 and DMS02-39625. We thank Princeton Institute for Computational Science and Engineering (PicSCie) for providing their computing resources.

[1] E. L. Haseltine and J. B. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[2] C. V. Rao and A. P. Akin, J. Chem. Phys. **118**, 4999 (2003).
[3] K. Takahashi, K. Kaizu, B. Hu, and M. Tomita, Bioinformatics **20**, 538 (2004).
[4] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. **122**, 014116 (2005).
[5] Y. Cao, D. Gillespie, and L. Petzold, J. Comput. Phys. **206**, 395 (2005).
[6] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[7] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[8] R. Srivastava, M. S. Peterson, and W. E. Bently, Biotechnol. Bioeng. **75**, 120 (2001).
[9] E. Vanden-Eijnden, Commun. Math. Sci. **1**, 385 (2003).
[10] I. Fatkullin and E. Vanden-Eijnden, J. Comput. Phys. **200**, 605 (2004).
[11] W. E. D. Liu and E. Vanden-Eijnden, Commun. Pure Appl. Math. **58**, 1544 (2005).
[12] W. E, Commun. Math. Sci. **1**, 423, (2003).
[13] B. Engquist and Y.-H. Tsai, Math. Comput. **74**, 1707 (2005).
[14] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, Lect. Notes Math. (in press).
[15] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, J. Comput. Phys. **17**, 10 (1975).
[16] W. E. D. Liu and E. Vanden-Eijnden, J. Comput. Phys. (submitted).
[17] R. Z. Khasminskii, G. Yin, and Q. Zhang, Q. Appl. Math. **55**, 177 (1997).