Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Journal of Computational Physics xxx (2006) xxx–xxx

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales

Weinan E [a], Di Liu [b,*], Eric Vanden-Eijnden [c]

[a] *Department of Mathematics and PACM, Princeton University, Princeton, NJ 08544, USA*
[b] *Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA*
[c] *Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA*

Received 24 October 2005; received in revised form 4 April 2006; accepted 7 June 2006

Abstract

We present an efficient numerical algorithm for simulating chemical kinetic systems with multiple time scales. This algorithm is an improvement of the traditional stochastic simulation algorithm (SSA), also known as Gillespie's algorithm. It is in the form of a nested SSA and uses an outer SSA to simulate the slow reactions with rates computed from realizations of inner SSAs that simulate the fast reactions. The algorithm itself is quite general and seamless, and it amounts to a small modification of the original SSA. Our analysis of such multi-scale chemical kinetic systems allows us to identify the slow variables in the system, derive effective dynamics on the slow time scale, and provide error estimates for the nested SSA. Efficiency of the nested SSA is discussed using these error estimates, and illustrated through several numerical examples.

Keywords: Kinetic Monte-Carlo; Continuous-time Markov chains; Chemical master equations; Stochastic Petri nets; Gillespie algorithm; Averaging theorems; Multiscale numerical methods; Homogenization; Stochastic modeling; Multi-scale computation; Chemical kinetic systems

1. Introduction

The stochastic simulation algorithm (SSA in short), also known as the Gillespie algorithm and originally introduced in the context of chemical kinetic systems, has found a wide range of applications in many different fields, including computational biology, chemistry, combustion, and communication networks [20,10,11]. Besides being an effective numerical algorithm, SSA is also a model for chemical kinetic systems that takes into account the discreteness and finiteness of the molecular numbers as well as stochastic effects. This feature makes it an attractive alternative to the approach of using systems of deterministic ODEs, particularly in situations when the stochastic effects are important [9]. In addition, since SSA uses less modeling assumptions and is therefore closer to the first principle models, it is often easier to determine the parameters in the model. In fact, the main modeling parameters are the rate functions which can in principle be computed using the rate theories [21].

[*] Corresponding author. Tel.: +1 517 353 8143; fax: +1 517 432 1562.
E-mail address: diliu@math.msu.edu (D. Liu).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.06.019

The disadvantage of SSA is that it is computationally more expensive to handle than the systems of ODEs. Besides being stochastic in nature, the system often involves many disparate time scales. This is easy to appreciate, since chemical reaction rates often depend exponentially on the activation energy. For a deterministic system of ODEs, this results in the stiffness of the ODEs, for which many efficient numerical methods have been developed [12]. However, the situation for SSA is much less satisfactory.

In recent years, this issue has received a great deal of attention and some important progress has been made. The main idea, pursued in different forms by many people, is to model the effective dynamics on the slow time scale, by assuming that the fast processes are in quasi-equilibrium [13,25,22,3,4]. In [13], a multi-scale simulation method was proposed in which the slow and fast reactions are simulated differently. The slow reactions are simulated using Gillespie algorithm and the fast reactions are simulated using Langevin dynamics. In [25], a similar multi-scale scheme is proposed in which the fast dynamics is simulated using deterministic ODEs. Both the approaches in [13,25] require that the volume of the system be sufficiently large in addition to having well-separated rates. [22] proposes a scheme based on the quasi-equilibrium assumption by assuming that the probability densities of the fast species conditioned on the slow species is known exactly or can be approximated, e.g. by normal distributions. The same quasi-equilibrium assumption is used in [3,4], except that the probability density of the fast species conditioned on the slow species is computed via a modified process called the virtual fast process.

The method proposed in [3,4] is more general than previous methods, but it still has limitations. It assumes the equilibrium distributions of the fast processes can be approximated by simple functions and the fast species are independent of each other at equilibrium. Moreover, the rate functions of the slow processes are also assumed to be of special forms and are approximated empirically by solving a system of algebraic equations. These limitations are removed in the recent work, [7], in which a nested SSA is proposed to deal with the time scale issue. This 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. The recent work in [23] is much closer to our work in spirit. It also adopted a nested structure with inner loop on the fast reactions and the outer loop on the slow reactions. However, the outer loop algorithm is significantly different from ours, without faithfully capturing the effective dynamics on the slow time scale. In particular, they also resort to a partition into slow and fast species, a partition that is avoided in our work.

It is worthwhile to emphasize that, as we will see in Section 3, the algorithm proposed in [7] is quite general and seamless. In particular, it makes no explicit mentioning of the fast and slow variables. At a first sight, this might seem surprising, since there are counterexamples showing that algorithms of the same spirit do not work for deterministic ODEs with separated time scales [8] if the slow variables are not explicitly identified and made use of. But in the present context, the slow variables are linear functions of the original variables, as a consequence of the fact that the state change vectors $\{\nu_j\}$s are constant vectors, and this is the reason why the seamless algorithm works.

However, unlike the original SSA which is exact, the nested SSA is approximate and to understand the errors in the nested SSA, it is important to understand what the slow and fast variables are and what the effective process is on the slow time scale. These issues were dealt with briefly in [7], and one main purpose of the present paper is to study them in more detail. This will allow us to estimate the optimal numerical parameters and the overall cost of the algorithm. In addition, we will discuss various extensions of the nested SSA, as well as important implementation issues such as adaptively determining slow and fast processes.

The paper is organized as follows. In Section 2, we define the slow variables and derive the effective dynamics on the slow time scale for chemical kinetic systems with two disparate time scales. Section 3 introduces the nested SSA for the special case when the system has two disparate time scales. Error estimates for the nested SSA are proved and illustrated through numerical examples. We also elaborate on why the nested SSA algorithm is seamless, and when a similar seamless algorithm can be developed in the context of ordinary differential equation such as, for instance, the ones that arises from the chemical kinetic system in the large volume limit. Then in Section 4, we show how to adaptively determine the partition of the system into slow and fast reactions during the simulation. Finally, in Section 5, we discuss the effective dynamics and nested SSA for system with multiple (more than two) well-separated time scales. In this case, both the averaging principle and the nested SSA can be applied iteratively, similar to the case in iterated homogenization [1]. We also study the system over the diffusive time scale.

2. Chemical kinetic systems with two disparate time scales

We will discuss first the case when the chemical kinetic system has two well-separated time scales. Systems with multiple (more than two) disparate time scales will be discussed in later sections.

2.1. The general setting

Let us first fix some notations. We will consider the time evolution of an isothermal, spatially homogeneous mixture of chemically reacting molecules contained in a fixed volume V. Suppose there are N_S species of molecules $S_{i=1,\ldots,N_S}$ involved, with M_R reactions $R_{j=1,\ldots,M_R}$. Let x_i be the number of molecules of species S_i. Then the state-space of the system is given by

$$\mathcal{X} \subset \mathbb{N}^{N_S} \tag{1}$$

and we will denote the elements in this state-space by $x = (x_1, \ldots, x_{N_S}) \in \mathcal{X}$. Each reaction R_j can be characterized by a rate function $a_j(x)$ and a state change (or stoichiometric) vector v_j which satisfies $x + v_j \in \mathcal{X}$ for all $x \in \mathcal{X}$ such that $a_j(x) \neq 0$. We write

$$R_j = (a_j, v_j), \quad R = \{R_1, \ldots, R_{M_R}\}. \tag{2}$$

Given state x, the occurrences of the reactions on an infinitesimal time interval dt are independent of each other and the probability for reaction R_j to happen during this time interval is given by $a_j(x)\,\mathrm{d}t$. The state of the system after reaction R_j is $x + v_j$. We assume that the state space is finite, as is the case for all chemical reactions in real life. The rate functions usually take the form of polynomials of x.

Consider the observable $u(x,t) = \mathbb{E}_x f(X_t)$, where X_t is the state variable at time t, and $\mathbb{E}_x$ denotes expectation conditional on $X_{t=0} = x$. $u(x,t)$ satisfies the following backward Kolmogorov equation:

$$\frac{\partial u(x,t)}{\partial t} = \sum_j a_j(x)(u(x + v_j, t) - u(x,t)) =: (Lu)(x,t). \tag{3}$$

The operator L is the infinitesimal generator of the Markov process associated with the chemical kinetic system.

Now we turn to chemical kinetic systems with two disparate time scales. Assume that the rate function of a chemical kinetic system $R = \{(a, v)\}$ has the following form:

$$a(x) = (a^{\mathrm{s}}(x), \epsilon^{-1} a^{\mathrm{f}}(x)), \tag{4}$$

where $\epsilon \ll 1$ represents the ratio of time scales of the system. The corresponding reactions and the associated state change vectors can be grouped accordingly:

$$R^{\mathrm{s}} = \{(a^{\mathrm{s}}, v^{\mathrm{s}})\}, \quad R^{\mathrm{f}} = \left\{\left(\frac{1}{\epsilon} a^{\mathrm{f}}, v^{\mathrm{f}}\right)\right\}. \tag{5}$$

We call R^{s} the slow reactions and R^{f} the fast reactions.

To illustrate these definitions, consider the following simple example system that we will investigate in more detail later:

$$S_1 \underset{a_2}{\overset{a_1}{\rightleftarrows}} S_2, \quad S_2 \underset{a_4}{\overset{a_3}{\rightleftarrows}} S_3, \quad S_3 \underset{a_6}{\overset{a_5}{\rightleftarrows}} S_4. \tag{6}$$

The reaction rates and the state change vectors are

$$\begin{aligned}
a_1 &= 10^5 x_1, \quad v_1 = (-1, +1, 0, 0),\\
a_2 &= 10^5 x_2, \quad v_2 = (+1, -1, 0, 0),\\
a_3 &= x_2, \quad v_3 = (0, -1, +1, 0),\\
a_4 &= x_3, \quad v_4 = (0, +1, -1, 0),\\
a_5 &= 10^5 x_3, \quad v_5 = (0, 0, -1, +1),\\
a_6 &= 10^5 x_4, \quad v_6 = (0, 0, +1, -1).
\end{aligned} \tag{7}$$

For this system, there are four species of molecules ($N_S = 4$) and six reactions channels ($M_R = 6$). From the reaction rates, it can be seen that the first and the third isomerization reactions are faster than the second isomerization reaction. We can partition the reactions into fast and slow groups:

$$R^{\mathrm{s}} = \{(a_3, v_3), (a_4, v_4)\}, \quad R^{\mathrm{f}} = \{(a_1, v_1), (a_2, v_2), (a_5, v_5), (a_6, v_6)\}. \tag{8}$$

If the initial values of the x_is are of O(1), the ratio of the time scales is of the order $\epsilon = 10^{-5}$. Notice that every variable x_i, $i = 1, 2, 3, 4$, is involved in at least one fast reaction so there is no slow species. On the other hand, the variables $y_1 = x_1 + x_2$ and $y_2 = x_3 + x_4$ are conserved during the fast reactions. In other words, each x_i, $i = 1, 2, 3, 4$, evolves over the fast time scale of $O(\epsilon)$ whereas y_i, $i = 1, 2$ evolves over the slow time scale of O(1). Fig. 1 gives the time evolution of $y_1 = x_1 + x_2$ and x_3 on an intermediate time scale of $O(10^{-3})$ starting from the initial value $(x_1, x_2, x_3, x_4) = (13, 3, 3, 3)$. It can been seen that x_3 changes its value many times while y_1 keeps unchanged on this intermediate time scale.

2.2. Effective dynamics on the slow time scale

For the kind of systems discussed above, very often we are interested mostly in the effective dynamics over the slow time scale. In this section we will derive the model for this effective dynamics.

The analysis is built upon the perturbation theory developed in [17,19,14–16]. First we need to understand what the slow variables are in the system. Let v be a function of the state variable x, which we call an observable. We say $v(x)$ is a slow observable if it does not change during the fast reactions, i.e. if for any x and any state change vector v_j^{f} associated with the fast reactions one has

$$v\left(x + v_j^{\mathrm{f}}\right) = v(x). \tag{9}$$

This is equivalent to saying that the slow observables are conserved quantities for the fast process R^{f} defined in (5). A general representation of such observables is given by special slow observables which are linear functions satisfying (9). We call such slow observables *slow variables*. It is easy to see that $v(x) = b \cdot x$ is a slow variable if

$$b \cdot v_j^{\mathrm{f}} = 0, \tag{10}$$

for all $\{v_j^{\mathrm{f}}\}$s. The set of such vectors form a linear subspace in R_{N_S}. Let $b_1, b_2, \ldots, b_J$ be a set of basis vectors of this subspace, and let

$$y_j = b_j \cdot x \quad \text{for } j = 1, \ldots, J, \tag{11}$$

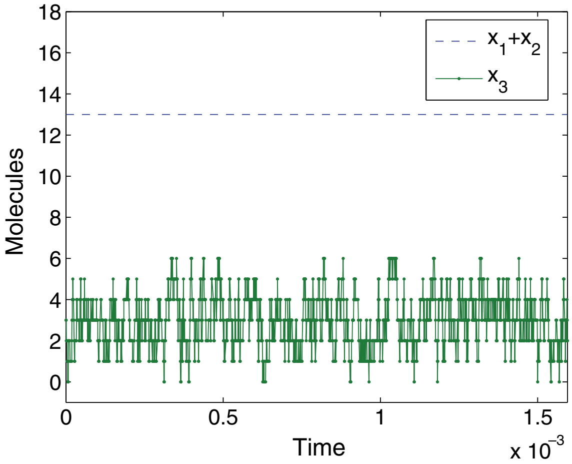

Fig. 1. Evolution of slow variable $y_1 = x_1 + x_2$ and fast variable x_3 on the intermediate timescale.

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.

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 $y_1, y_2, \ldots, y_J$. We define the slow variable y as

$$y = (y_1, \cdots, y_J) \tag{12}$$

and we will denote by $\mathscr{Y}$ the space over which y is defined. The state exchange vectors associated with the slow variables are naturally defined as:

$$\bar{v}_i^{\mathrm{s}} = (b_1 \cdot v_i^{\mathrm{s}}, \ldots, b_J \cdot v_i^{\mathrm{s}}), \quad i = 1, \ldots, N_s. \tag{13}$$

We will also adopt the notion of virtual fast process, defined in [3]. This is an auxiliary process that contains the fast reactions only, assuming, as we do now, that the set of fast and slow reactions do not change over time. This turns out to be a quite restrictive assumption and in later sections we will discuss the modifications needed when this assumption is no longer satisfied. Now we derive the effective dynamics on the slow time scale using singular perturbation theory. We assume that, for each fixed value of the slow variable y, the virtual fast process admits a unique equilibrium distribution $\mu_y(x)$ in the state space. We define a projection operator P by

$$(Pv)(y) = \sum_{x \in \mathscr{X}} \mu_y(x) v(x). \tag{14}$$

By this definition, for any $v : \mathscr{X} \to \mathbb{R}$, Pv depends only on the slow variable y, i.e. $Pv : \mathscr{Y} \to \mathbb{R}$.

The backward Kolmogorov equation for the multi-scale chemical kinetic system with reaction channels as in (5) reads:

$$\frac{\partial u}{\partial t} = L_0 u + \frac{1}{\epsilon} L_1 u. \tag{15}$$

where L_0 and L_1/ϵ are the infinitesimal generators associated with the slow and fast reactions, respectively: for any $f : \mathscr{X} \to \mathbb{R}$,

$$\begin{aligned} (L_0 f)(x) &= \sum_{j=1}^{M_{\mathrm{s}}} a_j^{\mathrm{s}}(x)(f(x + v_j^{\mathrm{s}}) - f(x)), \\ (L_1 f)(x) &= \sum_{j=1}^{M_{\mathrm{f}}} a_j^{\mathrm{f}}(x)(f(x + v_j^{\mathrm{f}}) - f(x)), \end{aligned} \tag{16}$$

where M_{s} is the number of slow reactions in R^{s} and M_{f} is the number of fast reactions in R^{f}. Look for a solution of (15) in the form of

$$u = u_0 + \epsilon u_1 + \epsilon^2 u_2 + \cdots \tag{17}$$

Inserting this into (15) and equating the coefficients, we arrive at the hierarchy of equations:

$$\begin{cases} L_1 u_0 = 0, \\ L_1 u_1 = \frac{\partial u_0}{\partial t} - L_0 u_0, \\ L_1 u_2 = \cdots \end{cases} \tag{18}$$

The first equation implies that u_0 belongs to the null-space of L_1, which by the ergodicity assumption of the fast process, is equivalent to

$$u_0(x, t) = \overline{U}(b \cdot x, t) \equiv \overline{U}(y, t), \tag{19}$$

for some $\overline{U}$ yet to be determined. Inserting (19) into the second equation in (18), gives (using the explicit expression for L_0)

$$\begin{aligned} L_1 u_1(x, t) &= \frac{\partial \overline{U}(b \cdot x, t)}{\partial t} - \sum_{i=1}^{M_{\mathrm{s}}} a_i^{\mathrm{s}}(x)(\overline{U}(b \cdot (x + v_i^{\mathrm{s}}), t) - \overline{U}(b \cdot x, t)) \\ &= \frac{\partial \overline{U}(y, t)}{\partial t} - \sum_{i=1}^{M_{\mathrm{s}}} a_i^{\mathrm{s}}(x)(\overline{U}(y + \bar{v}_i^{\mathrm{s}}, t) - \overline{U}(y, t)). \end{aligned} \tag{20}$$

This equation requires a solvability condition, namely that the right-hand side be perpendicular to the left null-space of L_1. By the ergodicity assumption of the fast process, this amounts to requiring that

$$\frac{\partial \overline{U}}{\partial t} = \sum_{i=1}^{M_s} \bar{a}_i^{\mathrm{s}}(y)(\overline{U}(y+\bar{v}_i^{\mathrm{s}}) - \overline{U}(y)), \tag{21}$$

where

$$\bar{a}_i^{\mathrm{s}}(y) = (Pa_i^{\mathrm{s}})(y) = \sum_x a_i^{\mathrm{s}}(x)\mu_y(x). \tag{22}$$

(21) is the effective dynamics on the slow time scale, and the effective reaction kinetics on this time scale are given in terms of the slow variable by

$$\overline{R} = (\bar{a}^{\mathrm{s}}(y), \bar{v}^{\mathrm{s}}). \tag{23}$$

The accuracy of the approximation of u by u_0 can be estimated as follows. From (18) and (21), it follows that

$$\left(\frac{\partial}{\partial t} - L_0 - \frac{1}{\epsilon}L_1\right)(u - u_0 - \epsilon u_1) = \left(L_0 + \frac{1}{\epsilon}L_1 - \frac{\partial}{\partial t}\right)(u_0 + \epsilon u_1) = \epsilon\left(L_0 - \frac{\partial}{\partial t}\right)u_1, \tag{24}$$

where u_1 is to be obtained by solving (20). Assuming that $u(x,0) = u_0(x,0) = f(b \cdot x)$, the above equality means that on fixed time-intervals

$$u - u_0 = \mathrm{O}(\epsilon). \tag{25}$$

2.3. Seamless form of the limiting theorem

The limiting dynamics in (21) can be reformulated on the original state space $\mathscr{X}$. Indeed, it is easy to check that (21) is equivalent to

$$\frac{\partial u_0}{\partial t} = \sum_{i=1}^{M_s} \bar{a}_i^{\mathrm{s}}(b \cdot x)(u_0(x + v_i^{\mathrm{s}}) - u_0(x)), \tag{26}$$

in the sense that if $u_0(x, t = 0) = f(b \cdot x)$, then

$$u_0(x,t) = \overline{U}(b \cdot x, t), \tag{27}$$

where $\overline{U}(y,t)$ solves (21) with the initial condition $\overline{U}(y,0) = f(y)$. The fact that (21) can be reformulated as (26) is the key reason why the nested SSA presented in Section 3 is a seamless algorithm that does not require to explicitly determine what the slow variables $y = b \cdot x$ are.

2.4. The example revisited

We now go back to the example (6) to illustrate these constructions. The fast reactions $R^{\mathrm{f}} = \{(\frac{1}{\epsilon}a^{\mathrm{f}}, v^{\mathrm{f}})\}$ have the following form:

$$\begin{aligned}
a_1^{\mathrm{f}} &= 10^5 x_1, \quad v_1^{\mathrm{f}} = (-1, +1, 0, 0),\\
a_2^{\mathrm{f}} &= 10^5 x_2, \quad v_2^{\mathrm{f}} = (+1, -1, 0, 0),\\
a_5^{\mathrm{f}} &= 10^5 x_3, \quad v_5^{\mathrm{f}} = (0, 0, -1, +1),\\
a_6^{\mathrm{f}} &= 10^5 x_4, \quad v_6^{\mathrm{f}} = (0, 0, +1, -1).
\end{aligned} \tag{28}$$

The slow reactions $R^{\mathrm{s}} = \{(a^{\mathrm{s}}, v^{\mathrm{s}})\}$ are

$$\begin{aligned}
a_3^{\mathrm{s}} &= x_2, \quad v_3^{\mathrm{s}} = (0, -1, +1, 0),\\
a_4^{\mathrm{s}} &= x_3, \quad v_4^{\mathrm{s}} = (0, +1, -1, 0).
\end{aligned} \tag{29}$$

The slow variables are

$$y_1 = x_1 + x_2, \quad y_2 = x_3 + x_4. \tag{30}$$

Notice that for each fixed set of values (y_1, y_2), the virtual fast process has a unique equilibrium distribution μ_y given by:

$$\mu_{y_1,y_2}(x_1,x_2,x_3,x_4) = \frac{y_1!y_2!}{x_1!x_2!x_3!x_4!}(1/2)^{y_1}(1/2)^{y_2}\delta_{x_1+x_2=y_1}\delta_{x_3+x_4=y_2}. \tag{31}$$

The effective dynamics on the slow time scale is given by the slow reactions with rate functions averaged with respect to these distributions

$$\begin{aligned} \bar{a}_3^{\mathrm{s}} &= Px_2 = \frac{x_1+x_2}{2} = \frac{y_1}{2}, \quad \bar{v}_3^{\mathrm{s}} = (-1,+1), \\ \bar{a}_4^{\mathrm{s}} &= Px_3 = \frac{x_3+x_4}{2} = \frac{y_2}{2}, \quad \bar{v}_4^{\mathrm{s}} = (+1,-1). \end{aligned} \tag{32}$$

3. The nested stochastic simulation algorithm

In this section, we introduce a nested stochastic simulation algorithm for chemical kinetic systems with two disparate rates. We discuss the convergence and efficiency of the scheme and illustrate them through an example of a virus infection model.

3.1. The stochastic simulation algorithm (SSA)

First, let us review briefly the standard stochastic simulation algorithm for chemical kinetic systems, proposed in [10,11] (see also [2]), also known as the Gillespie algorithm. Suppose we are given a chemical kinetic system with reaction channels $R_j = (a_j, v_j)$, $j = 1, 2, \ldots, M_R$. Let

$$a(x) = \sum_{j=1}^{M_R} a_j(x). \tag{33}$$

Assume that the current time is t_n, and the system is at state X_n. We perform the following steps:

(1) Generate independent random numbers r_1 and r_2 with uniform distribution on the unit interval (0,1]. Let

$$\delta t_{n+1} = -\frac{\ln r_1}{a(X_n)}, \tag{34}$$

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

$$\frac{1}{a(X_n)}\sum_{j=0}^{k_{n+1}-1} a_j(X_n) < r_2 \leqslant \frac{1}{a(X_n)}\sum_{j=0}^{k_{n+1}} a_j(X_n), \tag{35}$$

where $a(0) = 0$ by convention.

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

$$t_{n+1} = t_n + \delta t_{n+1}, \quad X_{n+1} = X_n + v_{k_{n+1}}. \tag{36}$$

Then repeat.

3.2. Nested SSA for system with two separated time scales

In [7], a modified SSA with a nested structure is proposed to simulate the chemical kinetic systems with multiple time scales. The process at each level of the time scale is simulated with an SSA with some possibly modified rates. Results from simulations on fast time scales are used to compute the rates for the SSA at slower time scale. For simple systems with only two time scales, the nested SSA consists of two SSAs

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.

organized with one nested in the other: An outer SSA for the slow reactions only, but with modified slow rates which are computed in an inner SSA modeling fast reactions only. Let t_n, X_n be the current time and state of the system, respectively. The nested SSA for systems with two time scales does the following:

(1) *Inner SSA:* Run N independent replicas of SSA with the fast reactions $R^{\mathrm{f}} = \{(\epsilon^{-1}\, a^{\mathrm{f}}, v^{\mathrm{f}}\})\}$ only, for a time interval of $T_0 + T_{\mathrm{f}}$. During this calculation, compute the modified slow rates for $j = 1,\ldots,M_{\mathrm{s}}$

$$\tilde{a}^{\mathrm{s}}_j = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{T_{\mathrm{f}}}\int_{T_0}^{T_{\mathrm{f}}+T_0} a^{\mathrm{s}}_j(X^k_\tau)\,\mathrm{d}\tau, \tag{37}$$

where X^k_τ is the result of the kth replica of this auxiliary virtual fast process at virtual time τ whose initial value is $X^k_{t=0} = X_n$, and T_0 is a parameter we choose in order to minimize the effect of the transients to the equilibrium in the virtual fast process.

(2) *Outer SSA:* Run one step of SSA for the modified slow reactions $\tilde{R}^{\mathrm{s}} = (\tilde{a}^{\mathrm{s}}, v^{\mathrm{s}})$ to generate (t_{n+1}, X_{n+1}) from (t_n, X_n).
Then repeat.

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 empirical approximations to get the effective slow rates. The reason why the algorithm works stems from (26), which shows that the effective equation for the slow variables living in $\mathscr{Y}$ can in fact be reformulated on the original state space $\mathscr{X}$. However, it is worth noting that this conclusion is specific to SSA and, it would in general not be true for systems described by (ordinary or stochastic) differential equation rather than a Markov chain. We elaborate on this point in Section 3.6.

Without fully realizing the effective dynamics (21), in the nested SSA proposed in [23], the outer SSA is advanced by picking up the next slow reaction using rates without being averaged with respect to the fast reactions, which will definitely induce a significant error in the scheme.

In the HMM (heterogeneous multi-scale method) [5] language, the macro-scale solver is the outer SSA, the data that need to be estimated is the effective rates for the slow reactions. These data are obtained by simulating the virtual fast process which plays the role of micro-scale solvers here.

3.3. Convergence of the nested SSA

The original SSA is an exact realization of the chemical kinetic system. The nested SSA, on the other hand, is an approximation. The errors in the nested SSA can be analyzed using the same strategy as in [6].

To begin with, since the state space is finite, it is easy to show that the virtual fast process is φ-irreducible and satisfies the stability condition in [18]. Hence for any test function $g : \mathscr{X} \to \mathbb{R}$, there exist positive constants R and α such that

$$\sup_{x\in\mathscr{X}} \left|(\mathrm{e}^{L_1 t} g)(x) - (Pg)(b\cdot x)\right| \leqslant R\mathrm{e}^{-\alpha t}. \tag{38}$$

Denote by $\tilde{X}_t$ the solution of the nested SSA. Consider the observable $v(x,t) = \mathbb{E}_x f(b\cdot\tilde{X}_t)$ where the expectation for f is with respect to the randomness in the outer SSA only. $v(x,t)$ satisfies a backward Kolmogorov equation similar to (26), in which the averaged rate $\bar{a}^{\mathrm{s}}_j$ in (22) are replaced by the random rates $\tilde{a}^{\mathrm{s}}_j$ obtained from (37) in the current realization of the inner SSA:

$$\frac{\partial v(x,t)}{\partial t} = \tilde{L}v(x,t), \tag{39}$$

where

$$\tilde{L}v(x,t) = \sum_{j=1}^{M_{\mathrm{s}}} \tilde{a}^{\mathrm{s}}_j(b\cdot x)(v(x+v^{\mathrm{s}}_j) - v(x)). \tag{40}$$

Let $u(x,t)$ be the solution of the system (15) with $u(x,0) = f(b\cdot x)$. We have the following theorem:

Theorem 3.1. *For any* $T > 0$, *there exist constants* C *and* α *independent of* (N, T_0, T_f) *such that*,

$$\sup_{0\leqslant t\leqslant T, x\in\mathscr{X}} \mathbb{E}|v(x,t)-u(x,t)| \leqslant C\left(\epsilon + \frac{\mathrm{e}^{-\alpha T_0/\epsilon}}{1+T_{\mathrm{f}}/\epsilon} + \frac{1}{\sqrt{N(1+T_{\mathrm{f}}/\epsilon)}}\right). \tag{41}$$

Proof. Let X_t^k be the kth realization at the virtual time t in the inner SSA. We have

$$\begin{aligned}\mathbb{E}|\tilde{a}^{\mathrm{s}}(x)-\bar{a}^{\mathrm{s}}(x)|^2 &= \frac{1}{N^2}\mathbb{E}_x\left|\frac{1}{T_{\mathrm{f}}}\sum_k\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t\right|^2 \\ &= \frac{1}{T_{\mathrm{f}}^2N^2}\sum_k\mathbb{E}_x\left|\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t\right|^2 + \frac{1}{T_{\mathrm{f}}^2N^2}\sum_{k\neq l}\mathbb{E}_x\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k) \\ &\quad -\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_{t'}^l)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t' \\ &=: A_1 + A_2.\end{aligned} \tag{42}$$

where $\mathbb{E}_x$ denotes expectation conditional on $X_{t=0}^k = x$. Using (38), we get the following estimate

$$\begin{aligned}A_1 &= \frac{2}{T_{\mathrm{f}}^2N^2}\sum_k\left(\mathbb{E}_x\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x))\times\mathbb{E}_{X_t^k}\int_t^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_\tau^k)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}\tau\,\mathrm{d}t\right) \\ &\leqslant \frac{2}{T_{\mathrm{f}}^2N^2}\sum_k\mathbb{E}_x\int_{T_0}^{T_0+T_{\mathrm{f}}}|a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x)|\int_t^{T_0+T_{\mathrm{f}}}R|a^{\mathrm{s}}|\mathrm{e}^{-\alpha(\tau-t)/\epsilon}\,\mathrm{d}\tau\,\mathrm{d}t \\ &\leqslant \frac{4R|a^{\mathrm{s}}|^2(\mathrm{e}^{-\alpha T_{\mathrm{f}}/\epsilon}-1+\alpha T_{\mathrm{f}}/\epsilon)}{N(\alpha T_{\mathrm{f}}/\epsilon)^2} \leqslant \frac{C}{N(1+T_{\mathrm{f}}/\epsilon)}.\end{aligned} \tag{43}$$

At the same time, we have from (38)

$$\begin{aligned}A_2 &\leqslant \frac{1}{T_{\mathrm{f}}^2N^2}\sum_{k\neq l}\left|\mathbb{E}_x\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_{t'}^l)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t'\right| \\ &\leqslant \frac{1}{T_{\mathrm{f}}^2}\left|\mathbb{E}\int_{T_0}^{T_0+T_{\mathrm{f}}}(a^{\mathrm{s}}(X_t^k)-\bar{a}^{\mathrm{s}}(x))\,\mathrm{d}t\right|^2 \leqslant \frac{R^2|a^{\mathrm{s}}|^2\mathrm{e}^{-2\alpha T_0/\epsilon}(1-\mathrm{e}^{-\alpha T_{\mathrm{f}}/\epsilon})^2}{(\alpha T_{\mathrm{f}}/\epsilon)^2} \leqslant \frac{C\mathrm{e}^{-2\alpha T_0/\epsilon}}{(1+T_{\mathrm{f}}/\epsilon)^2}.\end{aligned} \tag{44}$$

Hence we have

$$\mathbb{E}|\tilde{a}^{\mathrm{s}}(x)-\bar{a}^{\mathrm{s}}(x)|^2 \leqslant C\left(\frac{\mathrm{e}^{-2\alpha T_0/\epsilon}}{(1+T_{\mathrm{f}}/\epsilon)^2}+\frac{1}{N(1+T_{\mathrm{f}}/\epsilon)}\right), \tag{45}$$

which implies that

$$\mathbb{E}\|\tilde{L}-PL_0P\| \leqslant C'\left(\frac{\mathrm{e}^{-\alpha T_0/\epsilon}}{1+T_{\mathrm{f}}/\epsilon}+\frac{1}{\sqrt{N(1+T_{\mathrm{f}}/\epsilon)}}\right). \tag{46}$$

The finiteness of the state space implies the boundedness of v. Let $w(x,t)$ be the solution of the effective Eq. (21) with $w(x,0)=f(x)$. We have

$$\frac{\mathrm{d}\mathbb{E}|v-w|}{\mathrm{d}t} = \mathbb{E}|(\tilde{L}-PL_0P)v+PL_0P(v-w)| \leqslant C\mathbb{E}|v-w| + C\left(\frac{\mathrm{e}^{-\alpha T_0/\epsilon}}{1+T_{\mathrm{f}}/\epsilon}+\frac{1}{\sqrt{N(1+T_{\mathrm{f}}/\epsilon)}}\right), \tag{47}$$

which, together with (25), gives (41). □

3.4. Efficiency of the nested SSA

Now we discuss the efficiency of the nested SSA based on the error estimate (41). Given a chemical kinetic system with $R = \{(a_j, v_j)\}$, we assume that the total rate $a(x) = \sum a_j(x)$ does not fluctuate a lot in time. Given an error tolerance λ, we choose the parameters in the nested SSA such that each term in (41) is less than $\mathrm{O}(\lambda)$. One possible choice of the parameters is

$$T_0 = 0, \quad N = 1 + \epsilon^{-1} T_\mathrm{f} = \frac{1}{\lambda}. \tag{48}$$

The total cost for the nested SSA over a time interval of O(1) is

$$\text{cost} = \mathrm{O}(N(1 + T_0/\epsilon + T_\mathrm{f}/\epsilon)) = \mathrm{O}\left(\frac{1}{\lambda^2}\right). \quad \text{(nested SSA)} \tag{49}$$

The cost of the direct SSA is

$$\text{cost} = \mathrm{O}\left(\frac{1}{\epsilon}\right), \quad \text{(direct SSA)} \tag{50}$$

since the time step size is of order ϵ. When $\epsilon \ll \lambda^2$, the nested SSA is much more efficient than the direct SSA.

Next we discuss the influence of the other numerical parameters on the efficiency. The parameter T_0 which plays the role of numerical relaxation time does not influence much the efficiency. Given the same error tolerance λ, for the last term in the error estimate (41) to be less than $\mathrm{O}(\lambda)$, we need to have

$$N(1 + \epsilon^{-1} T_\mathrm{f}) \geqslant \mathrm{O}\left(\frac{1}{\lambda^2}\right). \tag{51}$$

This implies that the cost satisfies

$$\text{cost} \geqslant \mathrm{O}(N(1 + \epsilon^{-1} T_\mathrm{f})) = \mathrm{O}\left(\frac{1}{\lambda^2}\right), \quad \text{(nested SSA)} \tag{52}$$

!which is the same as (50) regardless the value of T_0. The above argument also implies that the cost of $\mathrm{O}(1/\lambda^2)$ is optimal for the nested SSA to achieve an error tolerance of λ.

We then move to the effect of parameter N, the number of realizations for inner SSA. Suppose we take $N = 1$, i.e. only one realization of the fast process in the inner SSA. For the error estimate (41) to satisfy the same error tolerance λ, we have to choose

$$1 + \epsilon^{-1} T_\mathrm{f} = \frac{1}{\lambda^2}. \tag{53}$$

The cost of the nested SSA is given by

$$\text{cost} = \mathrm{O}(N(1 + \epsilon^{-1} T_\mathrm{f})) = \mathrm{O}\left(\frac{1}{\lambda^2}\right), \quad \text{(nested SSA)} \tag{54}$$

which is the minimum cost of SSA for error tolerance λ as discussed above. This implies that using multiple realizations in the inner SSA does not increase the efficiency of the overall scheme either. But using multiple realizations allows us to speed up the computation on parallel computers. Suppose we use M processors to simulate independent copies of the fast processes, then the computing time on each processor reduces to

$$\text{cost} = \mathrm{O}\left(\frac{1}{M\lambda^2}\right), \quad \text{(nested SSA per processor)} \tag{55}$$

for simulating the inner SSA. Another technique for speeding up the computation of the nested scheme is to establish an on-the-fly chart for $\tilde{a}^\mathrm{s}(y)$ and re-use the same data of $\tilde{a}^\mathrm{s}(y)$ whenever the slow process revisit the same slow state y. This is especially effective when the state space is small.

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.

3.5. A numerical example: A virus infection model

A virus infection model was proposed in [24] as an example of the failure of modeling genetic reacting networks with deterministic dynamics. The model is studied in [13] as an example of reactions with disparate rates. The reactions considered in this model ($M_R = 6$) are listed in Table 1. The reacting species that need to be modeled are *genome*, *struct*, *template* and *virus* ($N_s = 4$). *Genome* is the vehicle of the viral genetic information which can take the form of DNA, positive-strand RNA, negative-strand RNA, or some other variants. *Struct* represents the structural proteins making up the virus. *Template* refers to the form of the nucleic acid that is transcribed and involved in catalytically synthesizing every viral component. The nucleotides and amino acids are assumed to be available at constant concentrations.

When $template > 0$, the production and degradation of *struct*, which are the third and fifth reactions marked with * in Table 1, are faster than the others. From the reaction rates, we can see that the ratio of time scales is about $\epsilon = 10^{-3}$. In the system that consists of only the fast reactions, *struct* has an equilibrium measure of a Poisson distribution with the parameter $\lambda = 500 \times \text{template}$ such that

$$\mathbb{P}_{template}(struct = n) = \frac{(500 \times template)^n}{n!} \exp(-500 \times template). \tag{56}$$

Notice that *struct* only shows up in the last slow reaction. The reduced dynamics in the form of the slow reactions ($a_{1,2,4,6}$) with the rates averaged with respect to the quasi-equilibrium of the fast reactions ($a_{3,5}$) can be given as a system with four reactions given in Table 2.

To test the convergence and efficiency of the nested SSA and compare it with the direct SSA, we use the mean value and the variance of *template* at time $T = 20$ as a benchmark. The initial condition is chosen to be:

$$(struct, genome, template, virus) = (0, 0, 10, 0). \tag{57}$$

A computation of this average by a direct SSA using $N_0 = 10^6$ realizations led to

$$\overline{template} = 3.7170 \pm 0.005, \quad \text{var}(template) = 4.9777 \pm 0.005. \tag{58}$$

This calculation took 34806.39 s of CPU time on our machine. For the nested SSA, we make a series of simulations in which we choose the size of the ensemble and the simulation time of the inner SSA according to

$$(N, T_0, T/\epsilon) = (1, 0, 2^{2k}), \tag{59}$$

for different values of $k = 0, 1, 2, 3, \ldots$. The error estimate in (41) then implies that the error δ should decay with rate:

$$\delta = \mathrm{O}(2^{-k}). \tag{60}$$

Table 3 gives the total CPU time and the obtained values of $\overline{template}$ and var(*template*) with the parameters of inner SSA chosen according to (59) and using $N_0 = 10^6$ realizations of the outer SSA (same as in the direct SSA). The relative errors on $\overline{template}$ is shown in Fig. 2.

Table 1
Reaction channels of the virus infection model

$Nucleotides \overset{a_1 = 1.\times template}{\longrightarrow} genome$
$Nucleotides\ +\ genome \overset{a_2 = .025\times genome}{\longrightarrow} template$
$Nucleotides\ +\ aminoacids \overset{a_3 = 1000\times template}{\longrightarrow} struct^{*}$
$Template \overset{a_4 = .25\times template}{\longrightarrow} degraded$
$Struct \overset{a_5 = 1.9985\times struct}{\longrightarrow} degraded/secreted^{*}$
$Genome\ +\ struct \overset{a_6 = 7.5d-6-genome\times struct}{\longrightarrow} virus$

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.

Table 2
The reduced virus infection model

$$Nucleotides \overset{a_1=1.\times template}{\longrightarrow} genome$$

$$Nucleotides + genome \overset{a_2=.025\times genome}{\longrightarrow} template$$

$$Template \overset{a_4=.25\times template}{\longrightarrow} degraded$$

$$Genome + struct \overset{a_6=3.75d-3\times genome^2\times struct}{\longrightarrow} virus$$

3.6. Some remarks on the large volume limit

As explained in Section 2.3, because the limiting equation on the slow time scale can be written as (26) on the original state space $\mathscr{X}$, the nested SSA presented in Section 2.3 always works in the context of chemical kinetic systems. This is somewhat surprising since similar statements do not hold in the case of ordinary or stochastic differential equations with multiple time scales. Here we make some remarks concerning this.

Consider the ordinary differential equation

$$\dot{X}_t = \frac{1}{\varepsilon} f(X_t) + g(X_t), \tag{61}$$

for some variable $X_t \in \mathbb{R}^n$. Assume that there exists a vector valued function $\varphi : \mathbb{R}^n \to \mathbb{R}^m (m < n)$ such that:

(1) We have

$$f(x) \cdot \nabla\varphi(x) = 0; \tag{62}$$

(2) For each fixed $y \in \mathbb{R}^m$, the dynamics

$$\dot{X}^{\mathrm{f}}_t = f(X^{\mathrm{f}}_t), \tag{63}$$

is ergodic on the level set $\varphi(x) = y$ with respect to the equilibrium distribution $\mathrm{d}\mu_y(x)$ (which might not be atomic).

Then $Y_t = \varphi(X_t)$ are slow variables satisfying the following equation:

$$\dot{Y}_t = H(Y_t), \tag{64}$$

where

$$H(y) = \int_{\mathbb{R}^n} g(x) \cdot \nabla\varphi(x)\, \mathrm{d}\mu_y(x). \tag{65}$$

(64) holds provided that conditions (1) and (2) are satisfied and the expectation in (65) is finite. The existence of a limiting dynamics has been exploited in [26] to construct efficient algorithms for the simulation of (61) when $\epsilon \ll 1$. However, these algorithms cannot, in general, be put in a seamless form because the mapping φ defining the slow variable is usually nonlinear, in contrast to what happens in the context of chemical kinetic systems. In particular, it is easy to see that the equation

Table 3
Efficiency of the nested SSA for the virus infection model

T_{f}/ϵ	1	4	16	64
CPU	154.8	461.3	2068.2	9190.9
$\overline{template}$	4.027	3.947	3.796	3.757
var ($template$)	5.401	5.254	5.007	4.882

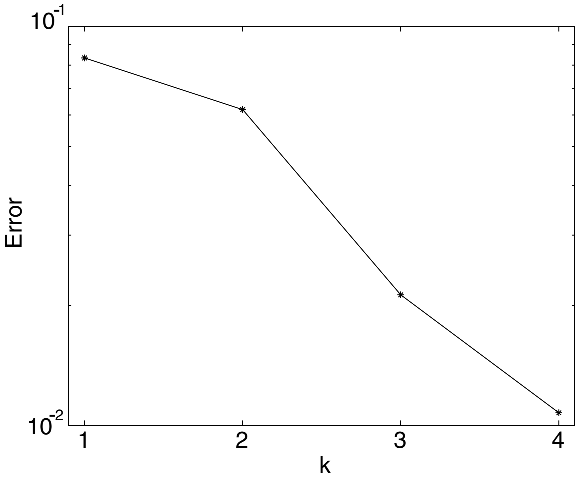

Fig. 2. Relative errors of $\overline{template}$ using the nested SSA for the virus infection model.

$$\dot{\overline{X}}_t = G(\varphi(\overline{X}_t)), \tag{66}$$

where

$$G(y) = \int_{\mathbb{R}^n} g(x)\,\mathrm{d}\mu_y(x), \tag{67}$$

will in general not be equivalent to (64) (in the sense that $\varphi(\overline{X}_t) \neq Y_t$), unless $\nabla\varphi(x)$ is a function of y only, i.e.

$$\nabla\varphi(x) = J(\varphi(x)), \tag{68}$$

for some $J : \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$. Only if (68) is satisfied do we have $H(y) = G(y)J(y)$ and $\varphi(\overline{X}_t) = Y_t$.

Condition (68) is rather restrictive. Quite remarkably, however, it is satisfied for the system of ordinary differential equations that arise from (15) in the large volume limit. Indeed, assuming that the number of molecules of each species is large, and after appropriate rescaling of the variables, it is well known that (15) leads to

$$\dot{X}_t = \frac{1}{\epsilon}\sum_{j=1}^{M_\mathrm{f}} a_j^\mathrm{f}(X_t)v_j^\mathrm{f} + \sum_{j=1}^{M_\mathrm{s}} a_j^\mathrm{s}(X_t)v_j^\mathrm{s}. \tag{69}$$

These equations are in the form of (61), and it is easy to see that if $(b_1,\ldots,b_m)$ is a basis of vector satisfying $b_i \cdot v_j^\mathrm{f}$, then

$$\varphi_j(x) = b_j \cdot x, \tag{70}$$

satisfy (62) and (68) (with J being the constant matrix with rows consisting of the vector b_j). This suggests that, in the limit as $\epsilon \to 0$, the solution of (69) converges to the solution of

$$\dot{\overline{X}}_t = \sum_{j=1}^{M^\mathrm{s}} \bar{a}_j^\mathrm{s}(b \cdot \overline{X}_t)v_j^\mathrm{s}. \tag{71}$$

Here

$$\bar{a}_j^\mathrm{s}(y) = \int_{\mathbb{R}^n} a_j^\mathrm{s}(x)\,\mathrm{d}\mu_y(x), \tag{72}$$

where $\mathrm{d}\mu_y(x)$ is the equilibrium of the fast process

$$\dot{X}_t^\mathrm{f} = \frac{1}{\epsilon}\sum_{j=1}^{M^\mathrm{f}} a_j^\mathrm{f}(X_t^\mathrm{f})v_j^\mathrm{f}. \tag{73}$$

Since (72) can be approximated by

$$\tilde{a}_{\mathrm{s}} = \frac{1}{T_{\mathrm{f}}} \int_0^{T_{\mathrm{f}}} a_j^{\mathrm{s}}(X_t^{\mathrm{f}})\,\mathrm{d}t, \tag{74}$$

for some appropriate T_{f}, this naturally leads to a very simple and seamless nested algorithm for simulating (69) for small ϵ: The inner loop solves the virtual fast system (63) for some time T_{f}; the outer loop solves (71), with $\bar{a}_J^{\mathrm{s}}$ approximated by $\bar{a}_j^{\mathrm{s}}(y) = a_j^{\mathrm{s}}(X_{T_{\mathrm{f}}}^{\mathrm{f}})$.

For instance, in the large volume limit, the ODEs corresponding to the simple example treated in Section 2.4 are

$$\begin{cases} \dot{X}_1 = 10^5(-X_1 + X_2), \\ \dot{X}_2 = 10^5(X_1 - X_2) - X_2 + X_3, \\ \dot{X}_3 = 10^5(-X_3 + X_4) - X_3 + X_2, \\ \dot{X}_4 = 10^5(X_3 - X_4). \end{cases} \tag{75}$$

In this case, the slow variables are $Y_1 = X_1 + X_2$ and $Y_2 = X_3 + X_4$, and it is easy to see that the fast process drives the variables toward the fixed point

$$X_1^{\mathrm{f}} = X_2^{\mathrm{f}} = Y_1/2, \quad X_3^{\mathrm{f}} = X_4^{\mathrm{f}} = Y_2/2, \tag{76}$$

meaning that $\mathrm{d}\mu_y(x)$ is atomic in this case

$$\mathrm{d}\mu_y(x) = \delta(x_1 - y_1/2)\delta(x_2 - y_1/2)\delta(x_3 - y_2/2)\delta(x_4 - y_2/2)\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3\,\mathrm{d}x_4. \tag{77}$$

Hence from (64), the limiting dynamics is

$$\begin{cases} \dot{Y}_1 = -X_2 + X_3 = (-Y_1 + Y_2)/2, \\ \dot{Y}_2 = -X_3 + X_2 = (Y_1 - Y_2)/2, \end{cases} \tag{78}$$

which from (71), can also be written in terms of the original variables as

$$\begin{cases} \dot{X}_1 = 0, \\ \dot{X}_2 = (-X_2 + X_3), \\ \dot{X}_3 = (-X_2 + X_3), \\ \dot{X}_4 = 0. \end{cases} \tag{79}$$

However, in general for the ODE systems, there might be additional independent slow variables beyond those identified by the linear conserved variables $y = b \cdot x$. Even though by assumption the original kinetic chemical system is ergodic on the components indexed by $y = b \cdot x$, this property may be lost in the infinite volume limit leading to (69). In other words, the ergodicity condition (2) above may not be satisfied unless additional, hidden slow variables are introduced. However, one can also show in this case the seamless algorithm in the style discussed earlier is still valid provided that the virtual fast system is dissipative on each of its ergodic component in the sense that the dynamics converges to unique steady states. This can also be seen from (68), since the equilibrium distribution $\mu_y(\mathrm{d}x)$ is a delta distribution centered on the steady state.

4. Adaptively partitioning the set of slow and fast reactions

In this section, we discuss the generalization of the nested SSA to systems for which the set of fast reactions changes over time. We would like the nested SSA to pick up the set of fast reactions dynamically. Consider the following system:

$$S_1 \underset{a_2}{\overset{a_1}{\rightleftarrows}} S_2, \quad S_2 \underset{a_4}{\overset{a_3}{\rightleftarrows}} S_3, \quad 2S_2 + S_3 \underset{a_6}{\overset{a_5}{\rightleftarrows}} 3S_4. \tag{80}$$

The reaction rates and the state change vectors are

$$\begin{aligned}
&a_1 = x_1, \quad \nu_1 = (-1, +1, 0, 0),\\
&a_2 = x_2, \quad \nu_2 = (+1, -1, 0, 0),\\
&a_3 = 10^4 x_2, \quad \nu_3 = (0, -1, +1, 0),\\
&a_4 = 10^4 x_3, \quad \nu_4 = (0, +1, -1, 0),\\
&a_5 = 2x_2(x_2 - 1)x_3, \quad \nu_5 = (0, -2, -1, +3),\\
&a_6 = 2x_4(x_4 - 1)(x_4 - 2), \quad \nu_6 = (0, +2, +1, -3).
\end{aligned} \tag{81}$$

Suppose that we start with the following initial condition:

$$(x_1, x_2, x_3, x_3) = (100, 3, 3, 3). \tag{82}$$

At the beginning, when the concentration of S_2 is low, only the transition between S_2 and S_3 is fast. As the number of S_2 grows, the last two reactions become faster and faster. Fig. 3 shows the evolution of the sum of the reaction rates $a_1 + a_2$ and $a_5 + a_6$. It can be seen that the reaction rate $a_5 + a_6$ grows from O(1) to O(10^5) on a time scale of O(1).

To test the nested SSA and compare it with the direct SSA, we 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

$$\overline{x_1} = 27.62 \pm 0.2, \quad \mathrm{var}(x_1) = 20.97 \pm 0.2. \tag{83}$$

The calculation took 8781.83 s of CPU time on our machine.

In the nested SSA, we dynamically change the set of fast reactions by constantly monitoring the following quantity

$$\kappa(t) = \int_0^t \left(\frac{a_5(s) + a_6(s)}{a_1(s) + a_2(s)}\right) \exp(s - t)\,\mathrm{d}s. \tag{84}$$

The kernel $\exp(s - t)$ serves to smooth out the oscillation in the ratio $\frac{a_5+a_6}{a_1+a_2}$. If $\kappa > 10^4$, the last two reactions are included in the set of fast reactions. If $\kappa < 10^3$, the last two reactions are treated as slow reactions. Otherwise, the direct SSA is adopted to simulate the whole system. Fig. 4 shows the above adaptive strategy. The indicator is set to be 0 when $\kappa < 10^3$, 2 when $\kappa > 10^4$ and 1 when κ is between 10^3 and 10^4. It can be seen that the adaptive scheme first treats the last two reactions as being slow and then as being fast when the time scale separation increases to the predetermined value of 10^4. The scheme oscillates between the direct and the nested SSA during some brief period of time when the last two reactions evolve from being slow to being fast. This is not a serious problem since the direct simulation is exact and there is still an efficiency gain due to the fact that the nested SSA is used most of the time.

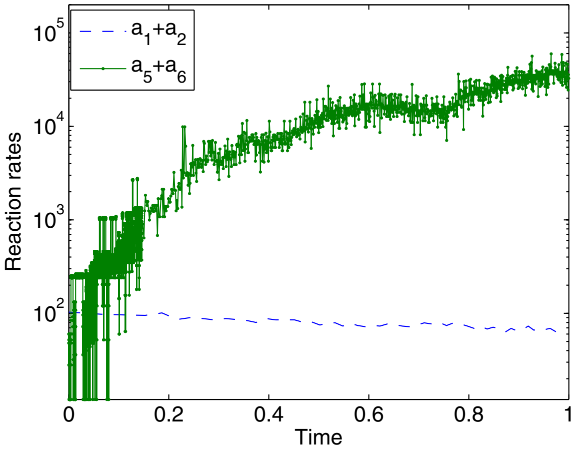

Fig. 3. Time evolution of the reaction rates $a_1 + a_2$ and $a_5 + a_6$.

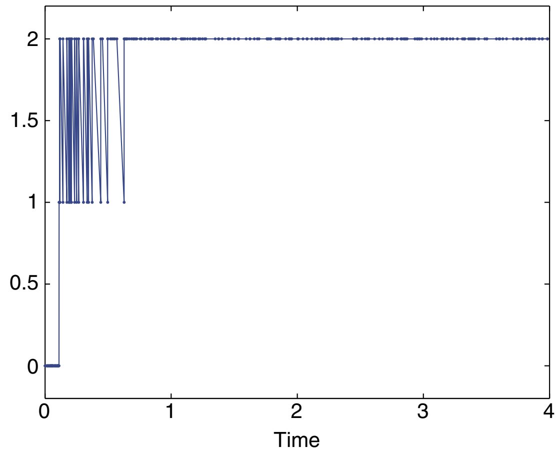

Fig. 4. Adaptive mechanism of the nested SSA.

Table 4
Efficiency and accuracy of the adaptive nested SSA

$T_f/10^{-5}$	1	2	4	8	16	32	64
CPU	13.6	18.6	28.0	47.2	86.2	163.0	316.2
$\overline{x_1}$	27.50	27.55	27.44	27.51	27.55	27.55	27.61
var(x_1)	20.58	20.65	20.82	20.57	20.84	20.58	21.01

To test the nested SSA, we conduct 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

$$(N, T_\mathrm{f}) = (1, 2^k \times 10^{-5}), \tag{85}$$

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

$$\lambda = \mathrm{O}(2^{k/2}). \tag{86}$$

Table 4 gives the CPU time and the values of the mean and variance of x_1 using $N_0 = 10{,}000$. The sum of the relative errors of the mean and the variance is shown in Fig. 5.

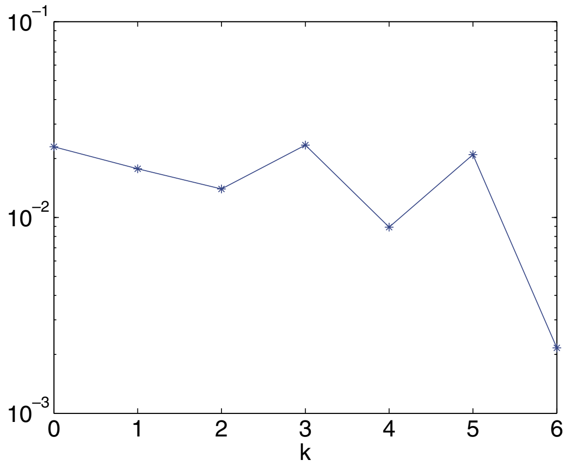

Fig. 5. Accuracy of the adaptive nested SSA.

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.

The strategy for adaptively determining the set of fast and slow reactions may not be the best one. Further work in this direction is clearly needed.

5. Nested SSA for systems with multiple time scales

Now we discuss chemical kinetic systems with multiple (more that two) well separated time scales. For simplicity, we focus on the instance when there are only three different time scales. The general case can be studied similarly. We will provide the asymptotic analysis for the effective dynamics on slow time scales and present the modified nested SSA for systems of this type.

5.1. Effective dynamics for systems with multiple time scales

Suppose we are given a chemical kinetic system with $R = \{(a_j, v_j)\}$ in which the rates $a_j(x)$ fall into three groups: One group corresponding to the ultra-fast processes with rates of order $1/\epsilon^2$; one group corresponding to fast processes with rates of order $1/\epsilon$; and one group corresponding to slow processes with rates of order 1:

$$a(x) = \left(a^{\mathrm{s}}(x), \frac{1}{\epsilon}a^{\mathrm{f}}(x), \frac{1}{\epsilon^2}a^{\mathrm{uf}}(x)\right). \tag{87}$$

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

$$R^{\mathrm{s}} = \{(a^{\mathrm{s}}, v^{\mathrm{s}})\}, \quad R^{\mathrm{f}} = \left(\frac{1}{\epsilon}a^{\mathrm{f}}, v^{\mathrm{f}}\right), \quad R^{\mathrm{uf}} = \left(\frac{1}{\epsilon^2}a^{\mathrm{uf}}, v^{\mathrm{uf}}\right). \tag{88}$$

The backward Kolmogorov equation for the observable $u(x,t) = \mathbb{E}_x f(X_t)$ is then of the form:

$$\frac{\partial u}{\partial t} = L_0 u + \frac{1}{\epsilon}L_1 u + \frac{1}{\epsilon^2}L_2 u, \tag{89}$$

where L_0, L_1/ϵ and L_2/ϵ^2 are the generators of the Markov processes associated with R^{s}, R^{f} and R^{uf}. As before, we can define a set of variables y which are slow compared with ultra-fast processes and are independent linear functions conserved during the ultra-fast reaction R^{uf}. As in Section 2.2, we shall denote these variables by

$$y_j = b_j \cdot x \tag{90}$$

where $(b_1,\ldots,b_J)$ are a basis of the subspace of vectors such that $b \cdot v_j^{\mathrm{uf}} = 0$ for all v_j^{uf}. Similarly, we can now defined slow variables z compared to both the fast and ultra-fast reaction as being independent linear functions conserved during the fast and ultra-fast reactions $(R^{\mathrm{f}}, R^{\mathrm{uf}})$. It is convenient to define the z variables as linear combinations of the y:

$$z_j = c_j \cdot y \tag{91}$$

where $(c_1,\ldots,c_K)$ are a basis of the subspace of vectors in $\mathbb{R}^J$ such that $c \cdot \bar{v}_j^{\mathrm{f}} = 0$ for all $\bar{v}_j^{\mathrm{f}} = (b_1 \cdot v_j^{\mathrm{f}}, \ldots, b_J \cdot v_j^{\mathrm{f}})$.

To derive the effective dynamics, let us expand u as

$$u = u_0 + \epsilon u_1 + \epsilon^2 u_2 + \cdots, \tag{92}$$

and insert this expansion in (89). This leads to

$$\begin{cases} L_2 u_0 = 0, \\ L_2 u_1 = -L_1 u_0, \\ L_2 u_2 = \frac{\partial u_0}{\partial t} - L_1 u_1 - L_0 u_0. \end{cases} \tag{93}$$

The first equation means that u_0 belongs to the null-space of L_2, i.e. $u_0(x,t) = \overline{U}(b \cdot x, t)$ for some $\overline{U}(y,t)$ yet to be determined. Inserting this expression into the second equation in (93) and looking for the solvability condition for the resulting equation, we arrive at

$$0 = PL_1\overline{U} = \sum_{i=1}^{M_{\mathrm{f}}} \bar{a}_i^{\mathrm{f}}(y)\left(\overline{U}(y + \bar{v}_i^{\mathrm{f}}) - \overline{U}(y)\right). \tag{94}$$

Here $\bar{v}_i^{\mathrm{f}} = b \cdot v_i^{\mathrm{f}}$ and

$$\bar{a}_i^{\mathrm{f}}(y) = \sum_{x \in \mathscr{X}} a_i^{\mathrm{f}}(x)\mu_y(x). \tag{95}$$

where $\mu_y(x)$ is the equilibrium distribution of the fast process on the ergodic component indexed by y. (94) implies that $\bar{U}$ belongs to the null-space of the following generator defined on function $f : \mathscr{Y} \to \mathbb{R}$:

$$(\bar{L}_1 f)(y) = \sum_{i=1}^{M_{\mathrm{f}}} \bar{a}_i^{\mathrm{f}}(y)(f(y + \bar{v}_i) - f(y)). \tag{96}$$

Assuming that the corresponding process generated by $\bar{L}_1$ is ergodic, with ergodic component indexed by z, i.e. it means that $\bar{U}(y)$ is in fact a function of $z = c \cdot y$, i.e.

$$\bar{\bar{U}}(y,t) = \bar{\bar{U}}(c \cdot y, t), \tag{97}$$

for some $\bar{\bar{U}}(z,t)$ to be determined. Let us denote by $\bar{\mu}_z(y)$ the equilibrium distribution of the process generated by (96) on the ergodic component indexed by z. Associated with $\mu_z(y)$ there is a projection operator Q defined as follows: for any $v : \mathscr{Y} \to \mathbb{R}$, it gives $Qv : \mathscr{X} \to \mathbb{R}$ as

$$(Qv)(z) = \sum_{y \in \mathscr{Y}} \bar{\mu}_z(y) v(y). \tag{98}$$

The equation for $\bar{\bar{U}}$ is obtained from the solvability condition for the third equation in (93), which is obtained by projection this equation first by P, then by Q. It reads

$$\frac{\partial \bar{\bar{U}}}{\partial t} = QPL_1\bar{\bar{U}} = \sum_{i=1}^{M_{\mathrm{s}}} \bar{\bar{a}}_i^{\mathrm{s}}(z)\left(\bar{\bar{U}}(z + \bar{\bar{v}}_i^{\mathrm{s}}) - \bar{\bar{U}}(z)\right). \tag{99}$$

Here $\bar{\bar{v}}_i^{\mathrm{s}} = c \cdot (b \cdot v_i^{\mathrm{s}})$ and

$$\bar{\bar{a}}_i^{\mathrm{s}}(z) = \sum_{y \in \mathscr{X}} \sum_{x \in \mathscr{X}} a_i^{\mathrm{s}}(x)\mu_y(x)\bar{\mu}_z(y). \tag{100}$$

Notice that (99) is equivalent to following equation for $\bar{\bar{u}}(x,t)$ on the original state-space $\mathscr{X}$,

$$\frac{\partial \bar{\bar{u}}}{\partial t} = \sum_{i=1}^{M_{\mathrm{s}}} \bar{\bar{a}}_i^{\mathrm{s}}(c \cdot (b \cdot x))\left(\bar{\bar{u}}(x + v_i^{\mathrm{s}}) - \bar{\bar{u}}(x)\right) \tag{101}$$

in the sense that if we solve (101) with the initial condition $\bar{\bar{u}}(x, t = 0) = f(c \cdot (b \cdot x))$, then $\bar{\bar{u}}(x,t) = \bar{\bar{U}}(c \cdot (b \cdot x), t)$ where $\bar{\bar{U}}$ solves (99) with the initial condition $\bar{\bar{U}}(z, t = 0) = f(z)$. We will make use of (101) in the following section. Notice also what we did above is an iterated averaging, a technique that has been developed in the context of homogenization [1].

5.2. Multi-level nested SSA

When the assumptions for iteratively averaged dynamics (101) hold, we can generalize the nested SSA with two levels proposed in Section 3 straightforwardly to handle multi-scale system (88) by using a nested SSA with more than two levels. Here we consider three levels. The innermost SSA uses only the ultra-fast rates and serves to compute the averaged fast and slow rates using formulas similar to (37). This will give us the dynamics on the ultra-fast time scale and the following quantities

$$\tilde{a}^{\mathrm{s}} \approx Pa^{\mathrm{s}}, \quad \tilde{a}^{\mathrm{f}} \approx Pa^{\mathrm{f}}. \tag{102}$$

The inner SSA uses only the above averaged fast rates $\tilde{a}^{\mathrm{f}}$ and the results are used again to compute the averaged slow rates (which are already averaged with respect to the ultra-fast reactions) as in (37):

$$\hat{a}^{\mathrm{s}} \approx QPa^{\mathrm{s}}. \tag{103}$$

Finally, the outer SSA uses only the above averaged slow rates. The cost of such a nested SSA is independent of ϵ, and as before, precise error estimates can be given in the same form of (41) in terms of T_{uf} – (the time interval over which the Innermost SSA is run and $(\tilde{a}^{\mathrm{s}}, \tilde{a}^{\mathrm{f}})$ is averaged), N_{uf} (the number of replicas in the Innermost SSA), T_{f} (the time interval over which the Inner SSA is run and $\hat{a}^{\mathrm{s}}$ is averaged), and N_{f} (the number of replicas in the Inner SSA):

$$\text{error} \leqslant C\left(\epsilon + \frac{1}{1+T_{\mathrm{f}}/\epsilon} + \frac{1}{1+T_{\mathrm{uf}}/\epsilon^2} + \frac{1}{\sqrt{N_{\mathrm{f}}(1+T_{\mathrm{f}}/\epsilon)}} + \frac{1}{\sqrt{N_{\mathrm{uf}}(1+T_{\mathrm{uf}}/\epsilon^2)}}\right). \tag{104}$$

Let us take a look at an example. Consider the following system

$$S_1 \underset{a_2}{\overset{a_1}{\rightleftarrows}} S_2, \quad S_2 \underset{a_4}{\overset{a_3}{\rightleftarrows}} S_3, \quad S_3 \underset{a_6}{\overset{a_5}{\rightleftarrows}} S_4. \tag{105}$$

with the reaction rates and state change vectors

$$\begin{aligned} &a_1 = 2\times 10^{10}x_1, \quad v_1 = (-1,+1,0,0),\\ &a_2 = 10^{10}x_2, \quad v_2 = (+1,-1,0,0),\\ &a_3 = 10^5x_2, \quad v_3 = (0,-1,+1,0),\\ &a_4 = 2\times 10^5x_3, \quad v_4 = (0,+1,-1,0),\\ &a_5 = x_3, \quad v_5 = (0,0,-1,+1),\\ &a_6 = x_4, \quad v_6 = (0,0,+1,-1). \end{aligned} \tag{106}$$

In this system, the first isomerization is ultra-fast, the second one is fast, the third one is slow, and $\epsilon = 10^{-5}$. The fast variables conserved during ultra-fast reactions are

$$(y_1, y_2, y_3) = (x_1 + x_2, x_3, x_4). \tag{107}$$

The slow variables conserved during fast and ultra-fast reactions are

$$(z_1, z_2) = (y_1 + y_2, y_3) = (x_1 + x_2 + x_3, x_4). \tag{108}$$

For each fast variable y, the ultra-fast reaction has an equilibrium distribution:

$$\mu_y(x_1, x_2) = \frac{y_1!}{x_1!x_2!}(1/3)^{x_1}(2/3)^{x_2}\delta_{x_1+x_2=y_1}\delta_{x_3=y_2}\delta_{x_4=y_3}. \tag{109}$$

So the effective rates on the fast time scale are

$$\begin{aligned} &\bar{a}_3^{\mathrm{f}} = P(10^5x_2) = \frac{2\times 10^5}{3}y_1, \quad \bar{v}_3^{\mathrm{f}} = (-1,+1,0),\\ &\bar{a}_4^{\mathrm{f}} = P(10^5x_3) = 2\times 10^5 y_2, \quad \bar{v}_4^{\mathrm{f}} = (+1,-1,0). \end{aligned} \tag{110}$$

For each slow variable z_1, the above reaction admits a unique equilibrium in the space of fast variable y:

$$\bar{\mu}_z(y_1, y_2) = \frac{z_1!}{y_1!y_2!}(3/4)^{y_1}(1/4)^{y_2}\delta_{y_1+y_2=z_1}\delta_{y_3=z_2}. \tag{111}$$

The effective rates on the slow time scale are:

$$\begin{aligned} &\bar{\bar{a}}_5^{\mathrm{s}} = QP(x_3) = Q(y_2) = \frac{z_1}{4}, \quad \bar{\bar{v}}_5^{\mathrm{s}} = (-1,+1),\\ &\bar{\bar{a}}_6^{\mathrm{s}} = QP(x_4) = Q(y_3) = z_2, \quad \bar{\bar{v}}_6^{\mathrm{s}} = (+1,-1). \end{aligned} \tag{112}$$

Suppose that we take the total number of molecules in the system to be $N = 1$. Then when the whole system is at equilibrium on the slow timescale, the overall equilibrium distribution is

$$\mu = (.2, .4, .2, .2). \tag{113}$$

To test the convergence and efficiency of the nested SSA, we take the following initial values

$$(x_1, x_2, x_3, x_4) = (1, 0, 0, 0), \tag{114}$$

and run the nested SSA to some long time on the slow time scale, say $T = 10^4$ using the parameters:

$$(N_{\mathrm{f}}, N_{\mathrm{uf}}, T_{\mathrm{f}}, T_{\mathrm{uf}}) = (1, 1, 10^{-2}, 10^{-7}). \tag{115}$$

We estimate the average equilibrium values of x_i by recording the visiting frequency of the states. With the above parameters, we obtain the following results:

$$\mu = (.1999, .3994, .1991, .2017). \quad \text{(three-level nested SSA)} \tag{116}$$

The maximum error is about 0.0085 compared with the exact values given by (113). In contrast, it is almost impossible to run the direct SSA to $T = 10^4$. To compare the efficiency of the nested SSA with the direct SSA, we fix the total number of iterations in the calculations. The calculation with the nested SSA with the parameters in (115) requires $\mathrm{O}(10^{10})$ iterations. With the same number of iterations, the direct SSA only advanced up to time $T_0' = \mathrm{O}(1)$, which is way too small to produce an accurate estimate for the equilibrium distribution. Fig. 6 shows this result. It can be seen that result from direct SSA is far from being accurate.

5.3. Nested SSA for the diffusive limit

In this section, we discuss the situation where the following centering condition holds

$$PL_1\overline{U} = 0, \tag{117}$$

which means that (94) is trivially satisfied. In this case, there is no need to introduce z variable as the slow dynamics on the O(1) time scale involves the y variables themselves. If (117) is satisfied, the second equation in (93) can be formally solved as

$$u_1 = -L_2^{-1}L_1\overline{U}. \tag{118}$$

Inserting this expression into the third equation in (93), and looking for the solvability condition for the resulting equation, we arrive at the following equation for $\overline{U}$:

$$\frac{\partial \overline{U}}{\partial t} = PL_0\overline{U} - PL_1L_2^{-1}L_1\overline{U}. \tag{119}$$

The generators at the left hand side of this equation can be expressed more explicitly. The first one is simply

$$PL_0\overline{U} = \sum_{j=1}^{M_s} \bar{a}_j^{\mathrm{s}}(y)\Big(\overline{U}(y + \bar{v}_j^{\mathrm{s}}, t) - \overline{U}(y, t)\Big), \tag{120}$$

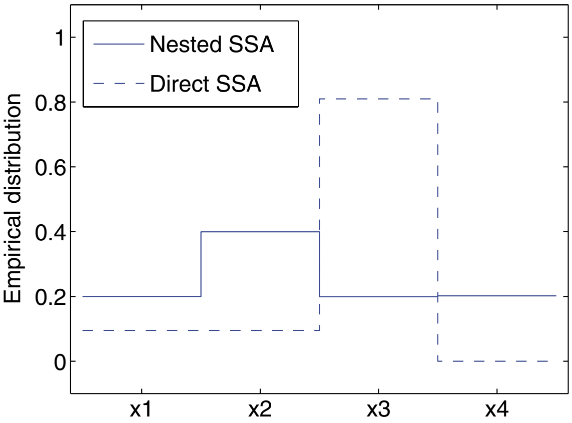

Fig. 6. Empirical distributions obtained by nested SSA and direct SSA at the same cost. The distribution produced by nested SSA is nearly exact, whereas the one produced by direct SSA is totally inaccurate.

where $\bar{v}_j^{\mathrm{s}} = b \cdot v_j^{\mathrm{s}}$ and

$$\bar{a}_j^{\mathrm{s}}(y) = \sum_{x \in \mathscr{X}} a_j^{\mathrm{s}}(x)\mu_y(x). \tag{121}$$

As for the second term in (119), let us denote by $X_{x,t}^{\mathrm{uf}}$ a sampling path of the process involving only the ultra-fast reactions associated with L_2 starting from x. Then for any $v : \mathscr{X} \to \mathbb{R}$ such that $\sum_{x\in\mathscr{X}} v(x)\mu_y(x) = 0$, we have

$$(-L_2^{-1}v)(x) = \int_0^\infty (\mathrm{e}^{L_2 t}v)(x)\,\mathrm{d}t = \int_0^\infty \mathbb{E}\; v(X_{x,t}^{\mathrm{uf}})\,\mathrm{d}t. \tag{122}$$

Using this expression together with condition (117), after some algebra one arrives at

$$-PL_1L_2^{-1}L_1\overline{U} = \sum_{j=1}^{M_{\mathrm{f}}} A_j(y)\Big(\overline{U}(y+\bar{v}_j^{\mathrm{f}},t) - \overline{U}(y,t)\Big), \tag{123}$$

where $\bar{v}_j^{\mathrm{f}} = b \cdot v_j^{\mathrm{f}}$ and

$$A_j(y) = \sum_{x\in\mathscr{X}} \mu_y(x) \sum_{i=1}^{M_{\mathrm{f}}} a_i^{\mathrm{f}}(x) \int_0^\infty \mathbb{E}\Big(a_j^{\mathrm{f}}(X_{x+v_i^{\mathrm{f}},t}^{\mathrm{uf}}) - a_j^{\mathrm{f}}(X_{x,t}^{\mathrm{uf}})\Big)\,\mathrm{d}t. \tag{124}$$

(119) is equivalent to the following equation for $\bar{u}(x,t)$ on the original state-space $\mathscr{X}$

$$\frac{\partial \bar{u}}{\partial t} = \sum_{j=1}^{M_{\mathrm{s}}} \bar{a}_j^{\mathrm{s}}(b\cdot x)\Big(\bar{u}(x+v_j^{\mathrm{s}},t) - \bar{u}(x,t)\Big) + \sum_{j=1}^{M_{\mathrm{f}}} A_j(b\cdot x)(\bar{u}(x+v_j^{\mathrm{f}},t) - \bar{u}(x,t)), \tag{125}$$

in the sense that the solution of (125) with the initial condition $\bar{u}(x,0) = f(b\cdot x)$ is $\overline{U}(y,t)$, the solution of (119) with the initial condition $\overline{U}(y,0) = f(y)$. (125) is a chemical kinetic system with the following reaction channels

$$\overline{R} = ((\bar{a}^{\mathrm{s}}, v^{\mathrm{s}}), (A, v^{\mathrm{f}})). \tag{126}$$

Based on the effective dynamics (125), a nested SSA for diffusive limit can be formulated. The nested SSA still consists of two levels of SSA. The inner level runs the ultra-fast reactions and the outer level runs the fast and slow reactions with modified rates estimated from the ultra-fast simulations using the following estimator for $\bar{a}_j^{\mathrm{s}}$ and A_j:

$$\begin{aligned} \tilde{a}_j^{\mathrm{s}} &= \frac{1}{N}\sum_{k=1}^N \frac{1}{T_{\mathrm{uf}}}\int_0^{T_{\mathrm{uf}}} a_j^{\mathrm{s}}(X_\tau^k)\,\mathrm{d}\tau, \\ \tilde{A}_j &= \frac{1}{N}\sum_{k=1}^N \frac{1}{T_{\mathrm{uf}}}\int_0^{T_{\mathrm{uf}}} \sum_{i=1}^{M_{\mathrm{f}}} a_i^{\mathrm{f}}(X_\tau^k)\int_0^{T'_{\mathrm{uf}}} \Big(a_j^{\mathrm{f}}(Y_{\tau+\omega}^k) - a_j^{\mathrm{f}}(X_{\tau+\omega}^k)\Big\}\,\mathrm{d}\omega\,\mathrm{d}\tau, \end{aligned} \tag{127}$$

where $(X_\tau^k, Y_{\tau+\omega}^{k,i})$ is the kth replica of the virtual ultra-fast process R^{uf} with initial values $X_{\tau=0}^k = X_n$ (the current state in the effective dynamics) and $Y_0^{k,i} = X_0^k + v_i^{\mathrm{f}}$. T'_{uf} is the virtual time we use to truncate the integral in (122). We can also obtain an error estimate for this scheme:

$$\text{error} \leqslant C\left(\epsilon + \frac{1}{1+T_{\mathrm{uf}}/\epsilon^2} + \frac{1}{\sqrt{N(1+T_{\mathrm{f}}/\epsilon^2)}} + \mathrm{e}^{-\alpha T'_{\mathrm{uf}}/\epsilon^2}\right). \tag{128}$$

For an explicit example, consider the following system

$$S_1 \underset{a_2}{\overset{a_1}{\rightleftarrows}} S_2, \quad S_2 \xrightarrow{a_3} S_3. \tag{129}$$

with the reaction rates and state change vectors

$$\begin{aligned} a_1 &= 10^5 x_1, \quad v_1 = (-1,+1,0), \\ a_2 &= 10^{10} x_2, \quad v_2 = (+1,-1,0), \\ a_3 &= 10^5 x_2, \quad v_3 = (0,-1,+1). \end{aligned} \tag{130}$$

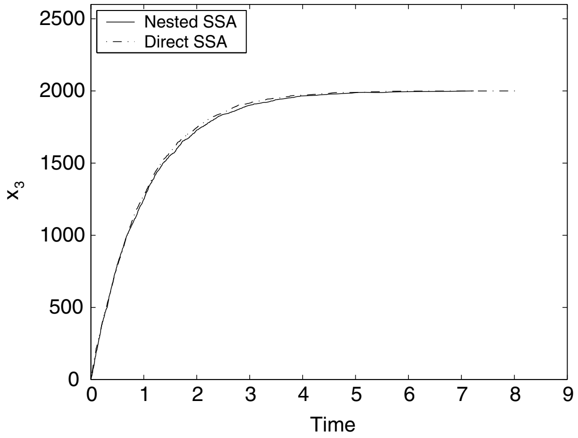

Fig. 7. Time evolution of the slow variable $y_1 = x_3$ on the diffusive timescale.

For the above system, $L_0 = 0$ and $\epsilon = 10^{-5}$. We can eliminate variable x_1 by the conservation of the total number of molecules $x_1 + x_2 + x_3 = M_0$ hence the ultra-fast variable is (x_2, x_3). The fast variable that keeps constant in the ultra-fast reaction is $y_1 = x_3$. The equilibrium distribution of x_2 for the virtual ultra-fast process is that of a one direction birth-death process such that

$$\text{Prob}(x_2 = 0) = 1. \tag{131}$$

The action of P is then simply taking $x_2 = 0$. The solvability condition (117) is satisfied since

$$PL_1U = P(x_1(U(y_1,t) - U(y_1,t)) + x_2(U(y_1+1,t) - U(y_1,t))) = 0. \tag{132}$$

The solution of $-L_2 v = x_2$ is $v = -L_2^{-1} x_2 = x_2$, which leads to

$$A_3(y) = P(M_0 - x_2 - x_3) = M_0 - y_1. \tag{133}$$

Thus the only reaction channel in the diffusive limit is

$$\overline{R} = (M_0 - x_3, v_3), \tag{134}$$

which is also a one direction birth-death process. For the nested SSA, we choose the parameters:

$$(N, T_{\text{uf}}, T'_{\text{uf}}) = (1, 2 \times 10^{-9}, T_{\text{extinct}}), \tag{135}$$

where $T'_{\text{uf}} = T_{\text{extinct}}$ means x_2 is run till extinction in the ultra-fast simulation. Fig. 7 shows the time evolution of the slow variable x_3 on the slow time scale obtained by the nested and direct SSA simulations. The mean relative error for the nth jumping time of x_3 is less than 0.067. The computation of the direct SSA takes 131.2 s of CPU time while the nested SSA takes only 0.37 s.

6. Conclusion

We analyzed a nested stochastic simulation algorithm proposed in [7] for multi-scale chemical kinetic systems. Convergence and efficiency are proved and illustrated through examples. Generalizations to systems with dynamic partition and multiple time scales of the fast and slow reactions are discussed.

Acknowledgments

The work of E is partially supported by NSF via Grant DMS01-30107. Liu is partially supported by NSF via Grant DMS97-29992. Vanden-Eijnden is partially supported by NSF via Grants DMS02-09959 and DMS02-39625. We thank Princeton Institute for Computational Science and Engineering (PicSCie) for providing the computing resources.

References

[1] A. Bensoussan, J.-L. Lions, G.C. Papanicolaou, Asymptotic analysis for periodic structuresStudies in Mathematics and Its Applications, vol. 5, Elsevier, North-Holland, New York, 1978.

[2] A.B. Bortz, M.H. Kalos, J.L. Lebowitz, A new algorithm for Monte Carlo simulation of Ising spin systems, J. Comp. Phys. 17 (1975) 10–18.

[3] Y. Cao, D. Gillespie, L. Petzold, The slow scale stochastic simulation algorithm, J. Chem. Phys. 122 (2005) 014116.

[4] Y. Cao, D. Gillespie, L. Petzold, Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems, J. Comp. Phys. 206 (2005) 395–411.

[5] W. E, B. Engquist, X. Li, W. Ren, E. Vanden-Eijnden, The heterogeneous multiscale method: A review, preprint, 2005.

[6] W. E, D. Liu, E. Vanden-Eijnden, Analysis of multiscale methods for stochastic differential equations, Comm. Pure Appl. Math. 58 (2005) 1544–1585.

[7] W. E, D. Liu, E. Vanden-Eijnden, Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates, J. Chem. Phys. 123 (2005) 194107.

[8] B. Engquist, Y.-H. Tsai, The heterogeneous multiscale methods for a class of stiff ODEs, Math. Comp. 74 (2005) 1707–1742.

[9] N. Fedoroff, W. Fontana, Small numbers of big molecules, Science 297 (2002) 1129–1131.

[10] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comp. Phys. 22 (1976) 403–434.

[11] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361.

[12] E. Hairer, G. Wanner, Solving ordinary differential equations II: stiff and differential-algebraic problems, second ed. Springer Series in Computational Mathematics, Springer, 2004.

[13] E.L. Haseltine, J.B. Rawlings, Approximate simulation of coupled fast and slow reactions for stochastic kinetics, J. Chem. Phys. 117 (2002) 6959–6969.

[14] R.Z. Khasminskii, On stochastic processes defined by differential equations with a small parameter, Theory Prob. Appl. 11 (1966) 211–228.

[15] R.Z. Khasminskii, A limit theorem for the solutions of differential equations with random right-hand sides, Theory Prob. Appl. 11 (1966) 390–406.

[16] R.Z. Khasminskii, G. Yin, Q. Zhang, Constructing asymptotic series for probability distributions of Markov chains with weak and strong interactions, Quart. Appl. Math. 55 (1997) 177–200.

[17] T.G. Kurtz, A limit theorem for perturbed operator semigroups with applications for random evolutions, J. Funct. Anal. 12 (1973) 55–67.

[18] S.P. Meyn, R.L. Tweedie, Stability of markov processes, I, II, and III, Adv. Appl. Prob. 24 (1992) 542–574, and 25, 518–548, 1993.

[19] G.C. Papanicolaou, Introduction to the asymptotic analysis of stochastic differential equations, in: R.C. DiPrima (Ed.), Lectures in Applied Mathematics, vol. 16, American Mathematical Society, Providence, R.I., 1977.

[20] C.A. Petri, Kommunikation mit Automaten, Institut fur Instru- mentelle Mathematik, Bonn, Schriften des IIM Nr. 2, 1962.

[21] R.D. Present, Kinetic Theory of Gases, McGraw-Hill, 1958 (Chapter 8).

[22] C.V. Rao, A.P. Arkin, Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm, J. Chem. Phys. 118 (2003) 4999–5010.

[23] A. Samant, D.G. Vlachos, Overcoming stiffness in stochastic simulation stemming from partial equilibrium: a multiscale Monte Carlo algorithm, J. Chem. Phys. 123 (2005) 144114.

[24] R. Srivastava, L. You, J. Summers, J. Yin, Stochastic vs. deterministic modeling of intracellular viral kinetics, J. Theor. Biol. 218 (2002) 309–321.

[25] K. Takahashi, K. Kaizu, B. Hu, M. Tomita, A multi-algorithm, multi-timescale method for cell simulation, Bioinformatics 20 (2004) 538–546.

[26] E. Vanden-Eijnden, Numerical techniques for multiscale dynamical systems with stochastic effects, Comm. Math. Sci. 1 (2003) 385–391.

Please cite this article as: Weinan E et al., Nested stochastic simulation algorithms for chemical ..., Journal of Computational Physics (2006), doi:10.1016/j.jcp.2006.06.019.