# Slow Scale Tau-leaping Method *

Yang Cao [†]     Linda Petzold [‡]

**Abstract**

For chemical systems involving both fast and slow scales, stiffness presents challenges for efficient stochastic simulation. Two different avenues have been explored to solve this problem. One is the slow-scale stochastic simulation (ssSSA) based on the stochastic partial equilibrium assumption. The other is the tau-leaping method. In this paper we propose a new algorithm, the slow-scale tau-leaping method, which combines some of the best features of these two methods. Numerical experiments are presented which illustrate the effectiveness of this approach.

# 1 Introduction

In recent years, concerns over stochastic effects resulting from the small numbers of certain reactant molecules in microscopic systems[1–4] have called for accurate and efficient stochastic simulation methods. The fundamental simulation method is Gillespie's Stochastic Simulation Algorithm (SSA).[5,6] Although progress[7,8] has been made to improve the efficiency of implementations of the SSA, as an exact procedure that simulates every reaction, it is necessarily inefficient for most realistic problems. The main reason for the low efficiency of the SSA is the multiscale nature of the underlying problem.

The multiscale problem in biochemical systems usually has two aspects. The first is the timescale. Some reactions occur much faster than others. Often the fast reactions quickly reach a stable state and the dynamics of the system is driven by the slow reactions. The SSA simulates every reaction and thus puts a great deal of effort into the more frequently occurring fast reactions, even though they do not contribute much to the dynamics and stochasticity of the system. This multiscale problem in time is known in the deterministic regime as *stiffness*.[9] Second, the populations of different species are of widely different magnitude. Some species are present with a large population while other species have only very few copies in a cell. Species with a small population should be modeled by a discrete

[*] This work was supported by the California Institute of Technology under DARPA Award No. F30602-01-2-0558, by the U. S. Department of Energy under DOE award No. DE-FG02-04ER25621, by the National Institute of Health under awards GM078893 and GM075297, by the National Science Foundation under award CCF-0726763, and by the Institute for Collaborative Biotechnologies through grant DAAD19-03-D-0004 from the U. S. Army Research Office.

[†] Department of Computer Science, Virginia Tech, VA 24061, ycao@cs.vt.edu.

[‡] Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106. petzold@engineering.ucsb.edu.

stochastic process, whereas species with a large population can be efficiently modeled by a deterministic ordinary differential equation (ODE). SSA treats all of the species as discrete stochastic processes.

A number of algorithms have been proposed to solve the multiscale problem in stochastic simulation of chemical systems. One of the most promising approaches is the tau-leaping method.[10] By using a Poisson approximation, the tau-leaping method can "leap over" many reactions and approximate the stochastic behavior of the system very well. The tau-leaping method makes a natural connection between the SSA in the discrete stochastic regime and the explicit Euler method applied to the chemical Langevin equation in the continuous stochastic regime, and to the reaction rate equations (RREs) in the continuous deterministic regime. In this sense, the tau-leaping method is ideal for multiscale stochastic simulation. However, the original explicit tau-leaping method is not efficient when stiffness is present. Implicit tau-leaping methods have been proposed to solve this problem, in particular the implicit tau-leaping method[11] and the trapezoidal tau-leaping method.[12] Convergence and stability for fixed stepsizes of the explicit and implicit tau-leaping methods have been studied.[13] For efficient implementation, algorithms for automatic selection of the stepsize $\tau$ has been proposed.[14,15] Moreover, recently an adaptive method[15] has been introduced that automatically switches between implicit and explicit tau-leaping methods, based on the comparison of stepsizes selected for both methods. Nonetheless, tau-leaping methods still have difficulty in effectively handling the situation when both multiscale features present together, particularly when a species with a small population is involved in a fast reaction channel. The latter situation also results in low efficiency for hybrid methods.[16,17]

A typical system of this type can be illustrate by the following example. Consider the system composed of fast reversible reaction channels

$$S_1 + S_2 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_3, \tag{1}$$

and slow reaction channel

$$S_1 + S_4 \xrightarrow{c_3} S_5 + S_1, \tag{2}$$

where $S_1$ is of a small population but $S_2$ and $S_3$ are of large populations. Both reaction channels in (1) are much faster than the reaction channel (2). The most extreme case is when the state of $S_1$ switches between zero and one. This situation actually occurs in the stochastic simulation of the heat shock response (HSR) model,[18,19] in which the sigma factor $\sigma_{32}$, which plays a crucial role in the HSR of E. Coli, has a small population but is involved in fast reactions. Due to the low population, currently all tau-leaping methods must switch to SSA to simulate the corresponding reaction channel, which makes the simulation very slow.

An efficient way to deal with stiffness in the accelerated simulation of discrete stochastic systems is to make use of a stochastic version of the quasi steady state or partial equilibrium assumptions.[20–22] In the deterministic case, the quasi steady state approximation assumes that on the time scale of interest, the instantaneous rates of change for some intermediate species are approximately equal to zero. The partial equilibrium approximation assumes that some fast reaction channels are always in equilibrium. In many cases these two assumptions are equivalent. The quasi-steady state approximation focuses on the state variables, while the partial equilibrium approximation concentrates on the reaction channels. The quasi steady state approximation was extended to the stochastic quasi steady state approximation (SQSSA),[23] while the partial equilibrium approximation was extended to the slow-scale SSA method.[24] Both were shown to be very effective in accelerating stochastic simulation. Since they are very similar, we will focus on the partial equilibrium approximation.

In the case of a system with (1) and (2), tau-leaping methods are not efficient but slow-scale SSA can be very efficient.[24] On the other hand, tau-leaping methods are advantageous over SSA in some other situations: they can more easily handle problems involving a multiplicity of different time scales. It would be ideal to combine the best features of both methods to form a slow-scale tau-leaping method. An important step has to be taken before we can do that. For either the slow-scale SSA or slow-scale tau-leaping method, we must design an algorithm to automatically detect the "fast but stable" reaction channels. Here we present a novel idea based on partitioning of the reaction channels into reversible reaction groups. Note that this strategy is designed for automatic partitioning. If one can pre-select a fast subsystem based on preknowledge of the system, the slow-scale tau-leaping method can be applied even more efficiently.

The outline of this paper is as follows. In Section 2 we briefly review the SSA, tau-leaping methods and the slow-scale SSA method. In Section 3 we introduce a general partitioning strategy. In Section 4 the methods for detecting equilibrium and estimating relaxation time are discussed for reversible reaction channels. The slow-scale tau-leaping method is described in Section 5, and numerical experiments are presented in Section 6.

# 2 Background: Stochastic Simulation Algorithms for Chemical Kinetics

## 2.1 SSA and Tau-leaping Methods

Suppose the system involves $N$ molecular species $\{S_1, \ldots, S_N\}$. The state vector is denoted by $X(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the number of molecules of species $S_i$ at time $t$. $M$ reaction channels $\{R_1, \ldots, R_M\}$ are involved in the system. Assume that the system is well-stirred and in thermal equilibrium. The dynamics of reaction channel $R_j$ is characterized by the *propensity function* $a_j$ and by the *state change vector* $\nu_j = (\nu_{1j}, \ldots, \nu_{Nj})$: $a_j(x)dt$ ia the probability that one $R_j$ reaction will occur in the next infinitesimal time interval $[t, t+dt)$, and $\nu_{ij}$ denotes the change in the $S_i$ molecular population induced by one $R_j$ reaction.

The dynamics of the system can be simulated by the SSA method.[5,6] With $X(t) = x$, let $a_0(x) = \sum_{j=1}^{M} a_j(x)$. On each step, SSA generates two random numbers $r_1$ and $r_2$ in $U(0,1)$, the uniform distribution on the interval $(0,1)$. The time for the next reaction to occur is given by $t + \tau$, where $\tau$ is given by

$$\tau = \frac{1}{a_0(x)} \log(\frac{1}{r_1}). \tag{3}$$

The index $j$ for the next reaction is given by the smallest integer satisfying

$$\sum_{l=1}^{j} a_l(x) > r_2 a_0(x). \tag{4}$$

The system states are updated by $X(t+\tau) = x + \nu_j$. The simulation proceeds to the next occurring time, until it reaches the final time.

Because it must proceed one reaction at a time, the SSA can be very slow for many practical problems. Gillespie[10] proposed a scheme called *tau-leaping* to accelerate the SSA. The basic idea of the tau-leaping method is to ask the question: How many times does each reaction channel fire in each subinterval? In each step, the tau-leaping method can proceed with many reactions. This is achieved at the cost of some accuracy. Define

$$\begin{aligned} K_j(\tau; x, t) \quad = \quad & \text{the number of times, given } X(t) = x, \text{ that reaction channel } R_j \\ & \text{will fire in the time interval } [t, t+\tau) \quad (j = 1, \ldots, M). \end{aligned} \tag{5}$$

Tau-leaping assumes the **Leap Condition**: *For the current state $x$, require $\tau$ to be small enough that the change in the state during $[t, t+\tau)$ will be so small that no propensity function will suffer an appreciable change in its value.* $K_j(\tau; x, t)$ is then well approximated by the Poisson random variable with mean and variance $a_j(x)\tau$,

$$K_j(\tau; x, t) = P(a_j(x)\tau) \quad (j = 1, \ldots, M). \tag{6}$$

The basic tau-leaping method proceeds as follows: Choose a value for $\tau$ that satisfies the Leap Condition. Generate for each $j = 1, \ldots, M$ a sample value $k_j$ of the Poisson random variable $P(a_j(x)\tau)$, and update the state by

$$X(t+\tau) = x + \sum_{j=1}^{M} k_j \nu_j. \tag{7}$$

If the populations of all reactant species are sufficiently large, the Poisson random variable $P(a_j(x)\tau)$ can be approximated by its mean $a_j(x)\tau$, and the basic (explicit) tau-leaping formula limits to the explicit Euler formula for the deterministic reaction rate equation.

## 2.2 Nonnegative Poisson Tau-Leaping and Tau Selection Formulas

It has been found that when some consumed reactant species are present in small numbers, the original explicit tau-leaping method may drive some reactant populations negative. Several strategies have been proposed to circumvent this problem. Tian and Burrage,[25] and independently Chatterjee et al.,[26] proposed to replace the unbounded Poisson random numbers $K_j$ with bounded binomial random numbers. It turns out that it is usually not the unboundedness of the Poisson $k_j$'s that produces negative populations, but rather the lack of coordination in tau-leaping between different reaction channels that separately decrease the population of a common species. An improvement[27] to the binomial tau-leaping has been proposed to use multinomial random numbers to generate $K_j$ values. Using the original Poisson tau-leaping, Cao et al.[28] have also proposed a different approach that resolves this difficulty and establishes a smooth connection with the SSA.

The Nonnegative Poisson tau-leaping algorithm[28] is based on the fact that negative populations typically arise from multiple firings of reactions that are only a few firings away from consuming all the molecules of one of their reactants. To focus on those reaction channels, the modified tau-leaping algorithm introduces a second control parameter $n_c$, a positive integer that is usually set somewhere between 5 and 20. Any reaction channel with a positive propensity function that is currently within $n_c$ firings of exhausting one of its reactants is then classified as a *critical* reaction. The modified algorithm chooses $\tau$ in such a way that no more than one firing of *all* the critical reactions can occur during the leap. Essentially, the algorithm simulates the *critical* reactions using an adapted (and thus not quite exact) version of the SSA, and the remaining *non-critical* reactions using the previously described Poisson tau-leaping method. Since no more than one firing of a critical reaction can occur during a leap, the probability of producing a negative population is reduced to nearly zero. On those rare occasions when a negative population does arise (from firings of some non-critical reaction), the leap can simply be rejected and repeated with $\tau$ reduced by half, or else the simulation can be started over using a larger value for $n_c$. The details can be found in Cao et al.[28]

In order to implement the tau-leaping method efficiently, we need to have a procedure to quickly determine the largest value of $\tau$ that is compatible with the Leap Condition. Gillespie[10] first formulated the Leap Condition. Later Gillespie and Petzold[29] improved the formula. In a more recent work, Cao et al.[14] proposed an improvement of the tau selection formula. The new tau-selection formula is given by

$$\tau = \min_{i \in I_{rs}} \left\{ \frac{\max\{\epsilon x_i / g_i, 1\}}{|\mu_i(x)|}, \frac{\max\{\epsilon x_i / g_i, 1\}^2}{\sigma_i^2(x)} \right\}, \tag{8}$$

where $I_{rs}$ is the set of indices of all *reactant species*, $g_i$ is given by a formula which guarantees that bounding the relative change of states is sufficient for bounding the relative change of propensity functions, and $\mu_i$, $\sigma_i$ are given by

$$\mu_i(x) \triangleq \sum_{j \in J_{ncr}} \nu_{ij} a_j(x), \quad \forall i \in I_{rs}, \tag{9a}$$

$$\sigma_i^2(x) \triangleq \sum_{j \in J_{ncr}} \nu_{ij}^2 a_j(x), \quad \forall i \in I_{rs}, \tag{9b}$$

where $J_{ncr}$ is the set of indices of all *non-critical reactions.*

## 2.3 Slow-Scale SSA

An efficient approach to stochastic simulation of stiff chemical systems is the slow-scale SSA[24] (ssSSA). The ssSSA was inspired by the well-known Michaelis-Menten approximation in deterministic chemical kinetics. The algorithm proceeds in a series of steps, the first of which is to make a provisional partitioning of the reaction channels $R = \{R_1, \ldots, R_M\}$ into fast and slow subsets, $R^f$ and $R^s$. Assigned to $R^f$ are those reactions whose propensity functions tend to have the largest values. All the other reactions are assigned to $R^s$. The second step is to partition the species $S = \{S_1, \ldots, S_N\}$ into fast and slow subsets, $S^f$ and $S^s$, according to the following rule: Any species whose population gets changed by a fast reaction is classified as a fast species; all other species (if there are any) are classified as slow. Note the subtle but important asymmetry that a fast species can get changed by a slow reaction, but a slow species cannot get changed by a fast reaction. The third step defines the virtual fast process $\hat{X}^f(t)$ as the fast species populations evolving under only the fast reactions $R^f$; i.e., $\hat{X}^f(t)$ is $X^f(t)$ with all the slow reactions switched off. The virtual fast process $\hat{X}^f(t)$ is a Markov process, whereas the real fast process $X^f(t)$ is generally non-Markovian, and hence practically intractable. Next we require that two stochastic stiffness conditions be satisfied: First, $\hat{X}^f(t)$ must be stable, in that it approaches as $t \longrightarrow \infty$ a well-defined time-independent random variable $\hat{X}^f(\infty)$. Second, the limit $\hat{X}^f(t) \longrightarrow \hat{X}^f(\infty)$ must be effectively accomplished in a time that is small compared with the expected time to the next slow reaction. This is a more precise specification of the degree of separation that must exist between the timescales of the fast and slow reactions. If these two stiffness conditions are satisfied, then our original classification of the fast reactions is deemed acceptable;

With the stochastic stiffness conditions satisfied, we now invoke the slow-scale approximation. The slow-scale approximation states, in essence, that we can ignore the fast reactions and simulate the system one slow reaction at a time, provided we replace the propensity function of each slow reaction by its average with respect to the asymptotic virtual fast process $\hat{X}^f(\infty)$. More precisely, if $\hat{P}(y^f, \infty | x^f, x^s)$ is the probability that $\hat{X}^f(\infty) = y^f$ given that $X(t) = (x^f, x^s)$, then the propensity function $a_j^s(x^f, x^s)$ of each slow reaction $R_j^s$ at time $t$ can be approximated on the timescale of the slow reactions by

$$\bar{a}_j^s(x^f, x^s) = \sum_{y^f} \hat{P}(y^f, \infty | x^f, x^s) a_j^s(y^f, x^s). \tag{10}$$

The ssSSA thus proceeds by simulating, in the manner of the SSA, the slow reactions using the propensity functions (10) and ignoring the fast reactions. A number of different approaches have been proposed to implement ssSSA (how the averages (10) are computed and how the fast-species populations are generated). The ssSSA has been successfully applied to a number of simple stiff systems, as well to the prototypical Michaelis-Menten system that is so ubiquitous in enzymatic reactions. These applications showed increases in simulation speed over the exact SSA of two to three orders of magnitude, with no perceptible loss of simulation accuracy.

## 3 Partitioning of Reaction Channels

Important questions still remain open for the slow-scale SSA. According to the analysis, the success of the slow-scale SSA depends on the partial equilibrium assumption and well separated reaction scales, which can be characterized by the relaxation time. A natural strategy to find the "fast and stable" reaction channel is to take two steps: First, check if some reaction channels are in equilibrium; second, check if the relaxation time for these reaction channels are much smaller than the reaction time for other reaction channels. Thus two fundamental questions arise

1. How should we determine whether or not a reaction group is in equilibrium?

2. How should we calculate the relaxation time if the reaction group is in equilibrium?

These two questions could be very complicated in general. But note that in many biochemical systems, partial equilibrium is reached in reversible reaction pairs based on the *detailed* *balance* assumption. In this paper we will focus our analysis on reversible reaction channels and propose a simulation strategy based on that. Note that partial equilibrium could be reached in realistic problems even if the detailed balance assumption is not satisfied. In that case, the method introduced here is not directly applicable.

Before working on the details for the reversible reaction channels, we partition all the reaction channels into groups. Each pair of reversible reaction channels is an individual reaction group. Irreversible reaction channels are put into another big reaction group. Thus in most cases each group consists of only a pair of reversible reaction channels. Note that this partitioning strategy allows automatic detection of the equilibrium state. In some special cases, one may put more than two reaction channels into a reaction group. In this paper we will discuss only the group with only a pair of reversible reaction channels. In the following we will use the notations below.

Let $G_j$ denote the reaction group $j$ and $M_j$ denote the number of reaction channels in group $G_j$. Let $a_{jl}$ denote the corresponding propensity function of the reaction channel $l$ in $G_j$ and $a_j$ be the sum of all the propensities in group $G_j$. Thus

$$a_j = \sum_{l=1}^{M_j} a_{jl}. \tag{11}$$

For a reaction group $G_j$ composed of only a pair of reversible reaction channels, $M_j = 2$. Since there are only two reaction channels in this group, we can simplify the notation. We will denote the forward propensity function as $a_{j+}(x)$ and the backward propensity function as $a_{j-}(x)$. Thus

$$a_j(x) = a_{j+}(x) + a_{j-}(x). \tag{12}$$

For the two reaction directions, the state change vectors are $\nu_j$ and $-\nu_j$, where $\nu_j$ is the state change vector for the forward reaction channel.

Why group reversible reaction channels together? Consider the reversible reaction channels (1). Suppose that $S_1$ presents with a very small population between 0 and 1, but $S_2$ and $S_3$ present with large populations. It is natural to use the propensities of the two reaction channels to measure the fast or slow scale. However, it will be very confusing if we consider the reversible reaction channels separately. Suppose we use a threshold of the propensities to decide whether or not they are fast. When $X_1 = 0$, $a_+ = 0$, the forward reaction will be taken as a slow reaction channel. But when $X_1 = 1$, $a_+$ is large. It will then be taken as a fast reaction channel. As this pair of fast reversible reactions fire frequently, the corresponding partitioning will keep changing. This problem can be easily solved if we consider the reversible reaction channels together. This group is considered fast if its propensity $a_j(x)$ is large.

The scales of a system are then determined by the magnitude of the $a_j$'s. Focusing on the fast reaction groups, the following situations are possible:

1. The reaction group is not in equilibrium and all the involved species present with large population. In this case, reactions in this group will be simulated by the tau-leaping ¡method, which allows to leap with a relatively large $\tau$.

2. The reaction group is not in equilibrium and at least one of the involved species presents with a small population. Then reactions in this group should be simulated by SSA. But since this reaction channel is fast, this subgroup will quickly reach equilibrium.

3. The reaction group is in the equilibrium state. Then we should estimate the relaxation time and compare it with the possible $\tau$ value allowed by the non-equilibrium reaction groups to decide if we can apply the partial equilibrium assumption for this group. If that is the case, we can apply the slow scale tau-leaping/SSA to simulate the other reaction groups.

In the last case, we need to derive the algorithms to detect the equilibrium state and measure the relaxation time.

# 4 Reversible Reaction Channels

## 4.1 Partial Equilibrium

To detect equilibrium for a general reaction group can be very complicated. However for a group consisting of only a pair of reversible reaction channels, it is much easier.

A straightforward method follows the equilibrium condition in the deterministic regime. If $a_{j+}(x) \approx a_{j-}(x)$, then this pair can be considered in equilibrium. This can be formulated as

$$|a_{j+}(x) - a_{j-}(x)| < \epsilon a_j(x), \tag{13}$$

where $\epsilon$ is a small positive real number. However, the situation in the stochastic regime can be more complicated. Since the state variables take only integer values, in some equilibrium states, (13) will never be satisfied.

Here we propose to add a simple mechanism upon the failure of (13) to detect equilibrium by keeping track of the *expected reaction direction* (ERD). There are two possible reaction directions for a pair of reversible reactions: the forward direction (+) and the backward direction (-). We can determine the ERD by comparing the propensities of the forward and backward reactions. For example, for a reaction group $G_j$ with a pair of reversible reaction channels, let the corresponding propensity functions be given by $a_{j+}(x)$ and $a_{j-}(x)$. If $a_{j+}(x(t)) > a_{j-}(x(t))$, the propensities are in favor of the forward direction. The ERD is then forward (+). Otherwise it is backward (-). Now consider only the firing of this pair of reaction channels. Suppose the reactions fired in the expected direction. If after one step (in the SSA this would correspond to firing one reaction; in tau-leaping, many reactions can occur during one tau-leaping step) the ERD changes, then this pair of reversible reactions should also be considered in dynamic equilibrium. If $a_{j+}(x(t)) > a_{j-}(x(t))$, the condition can be formulated as

$$a_{j+}(x(t+\tau)) < a_{j-}(x(t+\tau)). \tag{14}$$

Note that condition (14) is similar to the condition (6-4-22) in Ref.[30] (page 438). If a species with small population is involved and thus SSA is required for this group, the condition (14) can be simply written as

$$a_{j+}(x+\nu_j) < a_{j-}(x+\nu_j). \tag{15}$$

Equation (15) presents a practical condition for dynamic equilibrium in the discrete regime. However, this condition can be too strict when all involved species are present with a relatively large population. As long as the population of any species does not suffer a substantial change, multiple SSA steps (one tau-leaping step) should be allowed. If the ERD changes after one tau-leaping step, the subgroup should still be considered in dynamic equilibrium. To formulate this condition, the first step is to select the length $s_j$ of one tau-leaping step for this pair $G_j$ of reversible reactions. Note that $s_j$ is not a $\tau$ value we use to run the simulation, but an estimate of the time interval that, if the reversible reaction channels in group $G_j$ fire by the *expected values*, the populations of all species in group $G_j$ will not suffer a substantial change. For a pair of reversible reactions, if one of the reactions is critical, we allow only one SSA step in the expected reaction direction and use (15) to determine if it is in dynamic equilibrium. Otherwise, both reactions are not critical reactions. Following a similar analysis in Cao et al.,[14] $s_j$ can be given by

$$s_j = \min_{i \in I_{rs,j}} \frac{\max\{\epsilon x_i, 1\}}{|(a_{j+}(x) - a_{j-}(x))\nu_{j,i}|}, \tag{16}$$

where $I_{rs,j}$ is the set of indices of all reactant species and $\nu_{j,i}$ is the $i$-th component of the state change vector $\nu_j$ *in group* $G_j$.

After $s_j$ is determined by (16), $x(t + s_j)$ can be estimated by taking the *expected values* for the numbers of firings for both reaction directions. Then we have

$$\hat{n}_j = \max\left(1, [(a_{j+}(x) - a_{j-}(x))s_j]\right), \tag{17}$$

where $[y]$ represents the largest integer that is smaller than $y$. $\hat{n}_j$ thus estimates the average effect of the reversible reaction channels during the next time interval of $s_j$. The condition (14) becomes

$$a_{j+}(x + \hat{n}_j \nu_j) < a_{j-}(x + \hat{n}_j \nu_j). \tag{18}$$

A similar formula can be given if $a_{j+}(x(t)) < a_{j-}(x(t))$. This is a straightforward way to detect the equilibrium in the discrete stochastic regime, but more expensive than (13) in the continuous deterministic regime. For efficiency considerations, we check condition (13) first. If it is satisfied, the group is deemed to be in equilibrium. If it fails, we then check condition (14).

## 4.2 Relaxation Time

If some reaction channels are in equilibrium, according to the analysis of slow-scale SSA[24] an important criteria to decide whether or not we can neglect the stochasticity of these reaction channels is to see if they are fast enough so that, after a small fluctuation, they are able to resume equilibrium before any slow reaction fires. To verify that condition, we should compare the relaxation time of these fast reaction channels with the $\tau$ value given by the other non-equilibrium reaction channels using the tau-leaping method or the SSA method. In order to do that, we need to be able to efficiently calculate the relaxation time. In general this is a hard task. For example, studies about mean relaxation time[31] for a general stochastic system can lead to equations whose solution could be even more expensive (we have to solve it for every slow-scale step) than directly simulating the system by tau-leaping/SSA. A simple algorithm is needed for practical purposes. When the equilibrium is between a pair of reversible reaction channels, we propose the following.

We can measure the relaxation time by estimating how fast the mean values change back to the equilibrium state after a small perturbation. This can be done through deterministic analysis. Suppose a group with a pair of reversible reaction channels is close to equilibrium. We formulate the reaction rate equations only for this pair of reaction channels and get

$$\dot{x}(t) = (a_{j+}(x) - a_{j-}(x))\nu_j. \tag{19}$$

Suppose $x_j^*(t)$ is the local equilibrium state for the group $G_j$. Linearizing (19), we obtain

$$\dot{e}_j(t) = \nu_j[\frac{\partial a_{j+}(x)}{\partial x} - \frac{\partial a_{j-}(x)}{\partial x}]^T e(t), \tag{20}$$

where $e_j(t) = x(t) - x_j^*(t)$, The relaxation rate is determined by the eigenvalues of the matrix $J = \nu_j[\frac{\partial a_{j+}(x)}{\partial x} - \frac{\partial a_{j-}(x)}{\partial x}]^T$. It is easy to show that the eigenvalues of $J$ are either 0 or $\lambda_j(x) = [\frac{\partial a_{j+}(x)}{\partial x} - \frac{\partial a_{j-}(x)}{\partial x}]^T \nu_j$. Note that $\lambda_j(x)$ is always less than 0. (The state change vector $\nu_j$ takes 0 or a negative value for reactants and a positive value for products for any reaction.) Thus $e(t)$ will damp to 0 with a relaxation rate given by $|\lambda_j(x)|$. The relaxation time is then characterized by $\tau_{j,relax} = \frac{1}{|\lambda_j(x)|}$.

**Example 1: The Fast Reversible Isomerization**

One of the simplest stable fast processes arises from the reversible isomerization,

$$S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2. \tag{21}$$

We have

$$a_+(x) = c_1 x_1, \quad a_-(x) = c_2 x_2,$$

and $\nu = (-1 \;\; 1)^T$. Thus we obtain the eigenvalue of $J$ as

$$\lambda = [\frac{\partial a_+(x)}{\partial x} - \frac{\partial a_-(x)}{\partial x}]^T \nu = -(c_1 + c_2). \tag{22}$$

The relaxation rate is then given by $|\lambda| = (c_1 + c_2)$, and the relaxation time can be characterized by $\frac{1}{c_1+c_2}$. This result matches with the analysis in Cao et al.[24]

**Example 2: The Enzyme-Substrate System**

The reversible reaction pair in (1) often appears in the enzyme-substrate system. We have

$$a_+(x) = c_1 x_1 x_2, \quad a_-(x) = c_2 x_3,$$

and $\nu = (-1 \;\; -1 \; 1)^T$. We can calculate the eigenvalue

$$\lambda = -(c_1(x_1 + x_2) + c_2), \tag{23}$$

and obtain the relaxation rate as $|\lambda| = c_1(x_1 + x_2) + c_2$. The relaxation time is then characterized by $\frac{1}{c_1(x_1+x_2)+c_2}$. This result is very close to the previous analysis[32] as well.

If we have a hybrid system with many reversible reaction channels, how should we calculate the relaxation time? We can prove that the relaxation time of such a hybrid system can be estimated by the longest relaxation time of all subsystems.

**Theorem 4.1** *The relaxation time of a hybrid system is bounded by the largest relaxation time of all subsystems.*

The proof is given in the Appendix. Thus if we have a virtual fast process that consists of many subgroups, we can calculate the relaxation time for each subgroup and then take the maximum as the relaxation time for the virtual fast process.

# 5 The Slow-Scale Tau-Leaping Algorithm

With the mechanisms to partition the reaction groups, detect the equilibrium and measure the relaxation time, we have an automatic method to implement the slow-scale SSA. Now we are ready to implement the adaptive slow-scale tau-leaping algorithm. Suppose we have partitioned the reaction channels into subgroups, $G_1, \ldots, G_m$, where each $G_j$ is composed of only a pair of reversible reaction channels, except possibly for a few user-defined subgroups. Preknowledge of the system is certainly helpful for the partitioning process. Note that too many subgroups can reduce the efficiency. We always would like to construct subgroups for fast reversible reactions. For slow reactions, we can simply put them together with the irreversible reactions into a big subgroup whose reactions are not checked for equilibrium. The algorithm proceeds as follows.

**The slow-scale tau-leaping algorithm:**

1. Calculate $a_{jl}$ for each reaction channel $l$ in each subgroup $G_j$.

2. For each subgroup, determine whether it is in the partial equilibrium state using conditions (13) and (14).

3. For all subgroups $G_j$ that are not in partial equilibrium, calculate the corresponding tau-leaping stepsize $\tau$ using the tau-selection procedure introduced in Cao et al.[14]

4. For each subgroup $G_j$ that is in partial equilibrium, calculate $\tau_{j,relax}$. Then compare $\tau_{j,relax}$ with $\tau$. If $\tau_{j,relax} << \tau$, this subgroup will be sorted into the set of fast processes. Otherwise we sort this subgroup into the set of slow processes. Recalculate the $\tau$ value for the newly updated slow processes. When this process is done, we have $\tau_{relax} = \max_{\{j \text{ in virtual fast process}\}}\{\tau_{j,relax}\}$ and still $\tau_{relax} << \tau$.

5. Simulate this step. First, we relax the system so that all the fast and equilibrium reaction channels quickly reach the partial equilibrium state. The procedure to force the relaxation is to use a stiff formula such as the implicit Euler method to simulate the RREs given by only the virtual fast process. The time for the implicit Euler method is $\tau_{relax}$. Then we recalculate the propensities of the slow-scale reaction channels and run the adaptive tau-leaping/SSA method[14, 28] for them.

Note that in mass action kinetics, a pair of reversible reactions can occur in a small number of different structures. The most common structure is the subsystem (1). Then there are other subsystems like (21) or (26) that we will see in the numerical example. We can formulate the algebra of the equilibrium and relaxation time for each type following the analysis in Section 4. That makes the implementation much faster.

# 6 Numerical Experiments

We tested the slow-scale tau-leaping method on two example problems. All of the numerical experiments were implemented with a modified version of the StochKit[33] software package on a .

**Example 1: Simple Reactions**

This is the example system with reaction channels (1) and (2). We rewrite them as

$$\begin{array}{rcl} S_1 + S_2 & \overset{c_1}{\underset{c_2}{\rightleftharpoons}} & S_3, \\ S_1 + S_4 & \xrightarrow{c_3} & S_5 + S_1, \end{array} \tag{24}$$

where parameters $c_1 = 10^4$, $c_2 = 10^3$, $c_3 = 1$.

We simulated the system using the original SSA, slow-scale SSA and slow-scale tau-leaping method with $\epsilon = 0.03$ from $t_0 = 0$ to $T_f = 1$ under the initial condition $x_1 = 0$, $x_2 = x_3 = 10^3$, $x_4 = 10^3$ and $x_5 = 0$. The ensembles generated by both the slow-scale SSA and slow-scale tau-leaping yield very accurate distributions, when compared to the ensemble given by the full SSA (not shown as this is a very simple example.) For an ensemble of $10,000$ simulations, the computation time was $9,750$ CPU seconds for the the original SSA, 0.628 CPU seconds for the slow-scale SSA, and 0.715 CPU seconds for the slow-scale tau-leaping method. Both the slow-scale SSA and slow-scale tau-leaping method show great advantage over the original SSA. For this simple system it seems that slow-scale tau-leaping is comparable with the slow-scale SSA. However, if the population increases further, the slow-scale tau-leaping method will show an efficiency advantage.

To verify that, we simulated the above system using the three methods again from $t_0 = 0$ to $T_f = 1$ under a different initial condition. The only change we made was to increase the initial population of $S_4$ from $10^3$ to $10^5$. The results of all three methods still match very well. An ensemble of $10,000$ simulations took $10,715$ seconds CPU time for the original SSA, 54.94 seconds CPU time for the slow-scale SSA, and 7.16 seconds CPU time for the slow-scale tau-leaping method. We can see that the CPU time for the slow-scale SSA increased nearly 100-fold, while the CPU time for the slow-scale tau-leaping method increased only about 10-fold. This example demonstrates the efficiency of the slow-scale tau-leaping method over the slow-scale SSA.

**Example 2: Stiff Decaying Dimerization Process**

The stiff decaying dimerization model was originally proposed in Gillepsie[10] and modified in Rathinam et al.[11] The model consists of three species $S_1, S_2$ and $S_3$ and four reaction channels:

$$\begin{array}{c} S_1 \xrightarrow{c_1} 0 \\ S_1 + S_1 \overset{c_3}{\underset{c_2}{\rightleftharpoons}} S_2 \\ S_2 \xrightarrow{c_4} S_3. \end{array} \tag{25}$$

Following Rathinam et al.,[11] the reaction rates were chosen as $c_1 = 1,\ c_2 = 10,\ c_3 = 1000,\ c_4 = 0.1$.

This model consists of a pair of reversible reactions

$$S_1 + S_1 \underset{c_2}{\overset{c_3}{\rightleftharpoons}} S_2. \tag{26}$$

First we set the initial condition as $x_1(0) = 400$, $x_2(0) = 798$ and $x_3(0) = 0$ so that initially the reversible reaction pair (26) is in equilibrium. We simulated the system using the original SSA, slow-scale SSA, slow-scale tau-leaping method and adaptive explicit-implicit tau-leaping method with $\epsilon = 0.03$ from $t_0 = 0$ to $T_f = 0.2$. For an ensemble of $10,000$ simulations, it took $1,527$ seconds CPU time for the the original SSA, 2.72 seconds CPU time for the the slow-scale SSA, 4.41 (without the down-shifting[11,15] technique) and 17.88 (with the down-shifting) seconds CPU time for the adaptive explicit-implicit tau-leaping method, and 2.23 seconds CPU time for the slow-scale tau-leaping method. Note that for this example, in every slow-scale tau-leaping step, a normal random number was generated, as described in Cao et al.,[24] to give the population of the fast variables $X_1$ and $X_2$. Thus no down-shifting is necessary for the slow-scale SSA or slow-scale tau-leaping implementation. The mean and standard deviation of the ensembles by different methods are given in Table 1.

| Method | Mean | | | Standard Deviation | | |
|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_1$ | $x_2$ | $x_3$ |
| SSA | 386.95 | 749.79 | 15.42 | 18.43 | 10.44 | 3.88 |
| ssSSA | 387.15 | 749.76 | 15.50 | 18.74 | 10.71 | 3.93 |
| ss tau-leaping | 387.00 | 749.76 | 15.46 | 18.69 | 10.61 | 3.90 |
| adaptive explicit-implicit tau without downshifting | 386.82 | 749.77 | 15.45 | 15.98 | 9.50 | 3.90 |
| Adaptive explicit-implicit tau with down-shifting | 386.91 | 749.67 | 15.42 | 19.13 | 10.88 | 3.89 |

Table 1: Accuracy comparison for different methods under initial condition 1 for the decaying dimerization problem. The data are based on ensembles of $10,000$ simulations.

Second we set the initial condition as $x_1(0) = 10,000$, $x_2(0) = 0$ and $x_3(0) = 0$ so that initially the reversible reaction pair (26) is not in equilibrium. We simulated the system using the original SSA, slow-scale SSA and slow-scale tau-leaping method and adaptive explicit-implicit tau-leaping method with $\epsilon = 0.03$ from $t_0 = 0$ to $T_f = 10$. For an ensemble of $10,000$ simulations, it took 209 seconds CPU time for the slow-scale SSA, 139 (without the down-shifting technique) and 166 (with the down-shifting) seconds CPU time for the adaptive explicit-implicit tau-leaping method, and 35.9 seconds CPU time for the slow-scale tau-leaping method, while a single simulation with the original SSA took 17.49 seconds CPU time. The mean and standard deviation of the ensembles by different methods are given in Table 2.

From this example we can see that the three methods designed for stiff stochastic simulation all perform very well compared with the original SSA method. The slow-scale tau-leaping method showed an advantage in this model for the second set of initial conditions.

| Method | Mean | | | Standard Deviation | | |
|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_1$ | $x_2$ | $x_3$ |
| SSA | 238.3 | 284.2 | 1789 | 15.67 | 17.88 | 30.47 |
| ssSSA | 240.2 | 288.5 | 1789 | 15.57 | 18.21 | 30.53 |
| ss tau-leaping | 238.1 | 284.1 | 1789 | 15.63 | 18.15 | 30.59 |
| adaptive explicit-implicit tau without downshifting | 238.4 | 285.1 | 1789 | 14.52 | 18.06 | 30.70 |
| Adaptive explicit-implicit tau with down-shifting | 238.2 | 284.5 | 1794 | 15.57 | 18.28 | 30.88 |

Table 2: Accuracy comparison for different methods under initial condition two for the decaying dimerization problem. All the data are based on ensembles of $10,000$ simulations.

# 7 Conclusion and Discussion

In this paper we have proposed an algorithm to handle multiscale stochastic simulation using a combination of the slow-scale SSA and the adaptive tau-leaping method, based on the analysis of reversible reactions. The main contributions rest on the tests for partial equilibrium and the efficient algorithm for computing the relaxation time for reversible reaction channels. The new method combines the best features of the tau-leaping method and of ssSSA: like tau-leaping it can efficiently handle species with moderate populations, and the combination with ssSSA enables in practice the efficient treatment of stiff systems where a species with a small population is involved in fast reversible reactions.

It is possible to improve the mechanism introduced in this paper. An interesting question is: Do we really need to check the equilibrium condition? If the relaxation time is short enough, the fast processes will reach the partial equilibrium state in almost an instant time. If we can just use the implicit Euler method to relax the virtual fast process, even when the initial condition of the virtual fast process is not in equilibrium, the slow-scale SSA or tau-leaping may still be valid. In that case, we would not have to use the full-scale SSA or tau-leaping to simulate when the system is in the transient state, which would make the algorithm more efficient. Theory and implementation details are still under investigation.

# References

[1] H. McAdams, A. Arkin. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, 94:814–819, 1997.

[2] A. Arkin, J. Ross and H. McAdams. Stochasitc kinetic analysis of developmental pathway bifurcation in phage $\lambda$-infected escherichia coli cells. *Genetics*, 149:1633–1648, 1998.

[3] H.H. McAdams and A. Arkin. It's a noisy business! *Trends in Genetics*, 15(2):65–69, 1999.

[4] N. Fedoroff and W. Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 2002.

[5] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.

[6] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–61, 1977.

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

[8] Y. Cao, H. Li and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121:4059–67, 2004.

[9] Brenan, K.E., Campbell, S.L., and Petzold, L.R. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, Philadelphia, Pa, 1996 (second edition).

[10] D. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115:1716, 2001.

[11] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. *J. Chem. Phys.*, 119:12784–94, 2003.

[12] Y. Cao and L. Petzold. Trapezoidal tau-leaping formula for the stochastic simulation of chemically reacting systems. *Proceedings of Foundations of Systems Biology in Engineering (FOSBE 2005)*, pages 149–152, 2005.

[13] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Consistency and stability of tau leaping schemes for chemical reaction systems. *SIAM Multiscale Modeling*, 4:867–895, 2005.

[14] Y. Cao, D. Gillespie and L. Petzold. Efficient Stepsize Selection for the Tau-Leaping Method. *J. Chem. Phys.*, 124:044109, 2006.

[15] Y. Cao, D. Gillespie and L. Petzold. The adaptive explicit-implicit tau-leaping method with automatic tau selection. To appear in J. Chem. Phys., 2007.

[16] E. Haseltine and J. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.*, 117:6959–6969, 2002.

[17] T. Mattheyses and M. Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20:316–22, 2004.

[18] H. Kurata, H. El-Samad, T. Yi, M. Khammash and J. Doyle. Feedback regulation of the heat shock response in E. Coli. *Proceedings of the 40th IEEE conference on Decision and Control*, pages 837–842, 2001.

[19] H. Kurata, M. Khammash and J. Doyle. Stochastic analysis of the heat shock response in E. Coli. *3rd International Conference on Systems Biology*, 2002.

[20] J. D. Ramshaw. Partial chemical equilibrium in fluid dynamics. *Phys. Fluid*, 23, 1980.

[21] M. Rein. The partial-equilibrium approximation in reacting flows. *Phys. Fluids A*, 4, 1992.

[22] L. A. Segel and M. Slemrod. The quasi-steady-state assumption: a case study in perturbation. *SIAM Review*, 31, 1989.

[23] C. Rao and A. Arkin. Stochastic chemical Kinetics and the quasi steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.*, 118:4999–5010, 2003.

[24] Y. Cao, D. Gillespie and L. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122:014116, 2005.

[25] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121:10356–64, 2004.

[26] A. Chatterjee, D. G. Vlachos and M. A. Katsoulakis. Binomial distribution based tau-leap accelerated stochastic simulation. *J. Chem. Phys.*, 122:024112, 2005.

[27] M. Pettigrew and H. Resat. Multinomial tau-leaping method for stochastic kinetic simulation. *J. Chem. Phys.*, 126:084101, 2007.

[28] Y. Cao, D. Gillespie and L. Petzold. Avoiding negative populations in explicit tau leaping. *J. Chem. Phys.*, 123:054104, 2005.

[29] D. Gillespie and L. Petzold. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.*, 119:8229–34, 2003.

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

[31] A. Drozdov and J. Brey. Decay of metastable states: mean relaxation time formulation. *J. Chem. Phys.*, pages 7133–7141, 1999.

[32] Y. Cao, D. Gillespie and L. Petzold. Accelerated Stochastic Simulation of the Stiff Enzyme-Substrate Reaction. *J. Chem. Phys.*, 123:144917, 2005.

[33] H. Li, Y. Cao, L. Petzold and D. Gillespie. Algorithms and software for stochastic simulation of biochemical reacting systems. submitted, 2007.

[34] Y. Cao, D. Gillespie and L. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *J. Comput. Phys.*, 206:395–411, 2005.

[35] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1991.

## Appendix

**Proof of Theorem 4.1** The mean values of the state changes of the hybrid system can be written as

$$\dot{x} = \sum_{k=1}^{K} f_k(x), \tag{27}$$

where $f_k$ represents the state change for each subsystem. For example, for a pair of reversible reactions, $f_j(x) = (a_{j+}(x) - a_{j-}(x))\nu_j$. Suppose $x^*(t)$ is the local equilibrium state of the hybrid system. Linearizing (27) we obtain

$$\dot{e}(t) \approx \sum_{k=1}^{K} A_k(x^*)e(t), \tag{28}$$

where $e(t) = x(t) - x^*(t)$, $A_k(x^*)$ is the corresponding Jacobian of $f_k$ at $x^*$. In the following by default $A_k$ represents $A_k(x^*)$. The relaxation rate is determined by the real part of the eigenvalues of the matrix $\sum_{k=1}^{K} A_k$, Define

$$\lambda(A) = \max\{\mathrm{Re}(\lambda)|\det(\lambda I - A) = 0\}. \tag{29}$$

The relaxation time for each subsystem is given by $\frac{1}{-\lambda(A_k)}$. Here we assume for each $k$ that $A_k$ is a real matrix and $\lambda(A_k) < 0$, [1] As we can see in Section 4, this assumption is true for reversible reaction channels. Now we need only show that

$$\lambda(\sum_{k=1}^{K} A_k) \leq \max_{1\leq k\leq K}\{\lambda(A_k)\}. \tag{30}$$

This can be stated as

**Lemma .1** *Suppose $A$ and $B$ are real matrices that satisfy $\lambda(A) < 0$ and $\lambda(B) < 0$. Then $\lambda(A + B) \leq \max\{\lambda(A), \lambda(B)\}$.*

**Proof of Lemma .1** If both $A$ and $B$ are symmetric, this result follows immediately from Corollary 8.1.3 in Ref.[35] Thus we need only show that for a real matrix $A$, $\lambda(A) = \lambda(\frac{A+A^T}{2})$.

For a real matrix $A$, there exists a matrix $U$ such that $U^*U = I$ and $U^*AU = \mathrm{diag}\{\lambda_1, \ldots, \lambda_r, A_1, \ldots, A_s, 0, \ldots, 0\}$. where $\lambda_i$ are real eigenvalues of $A$, and $A_i$ has the form $\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$, where $a \pm bi$ are eigenvalues of $A$. Thus

$$U^* \left(\frac{A^T + A}{2}\right) U = \mathrm{diag}\{\lambda_1, \ldots, \lambda_r, a_1, \ldots, a_s, 0, \ldots, 0\}. \tag{31}$$

The result $\lambda(A) = \lambda(\frac{A+A^T}{2})$ follows directly from (31) .

[1] Otherwise there is a subsystem that will not reach its equilibrium. The hybrid system will not reach its equilibrium either.