Journal of Computational Physics 235 (2013) 446–457

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Time dependent solution for acceleration of tau-leaping

Jin Fu *, Sheng Wu, Linda R. Petzold

Department of Computer Science, University of California, Santa Barbara, United States

ARTICLE INFO

Article history:
Received 7 October 2011
Received in revised form 21 March 2012
Accepted 19 October 2012
Available online 16 November 2012

Keywords:
Stochastic simulation
Tau-leaping
Time dependent solution
Enzyme substrate system

ABSTRACT

The tau-leaping method is often effective for speeding up discrete stochastic simulation of chemically reacting systems. However, when fast reactions are involved, the speed-up for this method can be quite limited. One way to address this is to apply a stochastic quasi-steady state assumption. However we must be careful when using this assumption. If the fast subsystem cannot reach a steady distribution fast enough, the quasi-steady-state assumption will propagate error into the simulation. To avoid these errors, we propose to use the *time dependent solution* rather than the quasi-steady-state. Generally speaking, the time dependent solution is not easy to derive for an arbitrary network. However, for some common motifs we do have time dependent solutions. We derive the time dependent solutions for these motifs, and then show how they can be used with tau-leaping to achieve substantial speed-ups, including for a realistic model of blood coagulation. Although the method is complicated, we have automated it.

1. Introduction

Ordinary differential equation (ODE) models are widely used in the simulation of chemical systems where all chemical species are present with large population. For the simulation of biochemical systems inside a living cell, however, the population of some chemical species may be so small that stochastic fluctuations become important [1–3]. For these systems, a discrete stochastic model is more appropriate. The stochastic simulation algorithm (SSA) [4,5] is commonly used to simulate such a system. The SSA is exact, in the sense that each simulation is a realization of the Chemical Master Equation [5]. As the number of stochastic realizations goes to infinity, their statistics approach the probability density vectors (PDVs) which are the solutions to the Chemical Master Equation.

Typically, a great many (hundreds of thousands to millions) of simulations are required to get a good approximation to the PDVs. At the same time, each realization can be quite expensive because SSA, as an exact algorithm, requires the simulation of every reaction event in the system, which may include some very fast reactions. Tau-leaping [6] was developed to speed up the simulations. Tau-leaping is an approximate algorithm that can for many systems take time steps that are considerably larger than the time to the next reaction (i.e. the SSA timestep). It accomplishes this by allowing multiple reaction events to fire during a timestep as long as these reactions do not change the system dramatically, i.e. the change of each species during a step is small compared with its population. The stepsize for tau-leaping can become constrained, however, for systems with fast reactions that involve at least one species that is present in very small population [7].

One way to accelerate both SSA and tau-leaping for such stiff systems is to make use of a quasi-steady-state assumption. The quasi-steady-state assumption is a widely used strategy to handle systems that have different time scales, for both ODE [8] and SSA models [9–11]. The essence of this strategy is to divide the system into fast and slow subsystems. If the fast

* Corresponding author.
E-mail addresses: iamfujin@hotmail.com (J. Fu), sheng@cs.ucsb.edu (S. Wu), petzold@cs.ucsb.edu (L.R. Petzold).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.10.036

subsystem can reach a quasi-steady-state in a very short time, then we can use the quasi-steady-state as an approximation of the fast variables during a step of the slow subsystem. One can also apply the quasi-steady-state assumption in tau-leaping [7]. However, we must be careful when using this assumption. If the fast subsystem cannot reach a steady distribution rapidly enough, the quasi-steady-state assumption will propagate error into the simulation.

To avoid these errors, we can use the *time dependent solution* rather than the quasi-steady-state. The idea of using the time dependent solution to speed up a discrete stochastic simulation has been applied via a splitting method in [12]. That method first partitions the reactions into subgroups such that some of them have analytical solutions, which can be used to directly sample the state of the subsystem at any given time if reactions outside the subsystem keep silent. Then the method advances the system by advancing each subsystem separately in a given order with some stepsize. Since it can directly sample the state without sampling individual reaction events for those subsystems that have analytical solutions, it is more efficient than SSA if these subsystems contain many reaction events. However, it does not handle non-catalytic bimolecular reactions with the time dependent solution, or provide a stepsize selection strategy. The adaptive tau-leaping method addresses these two issues. It approximates the number of firings for bimolecular reactions for each step [6] and it also has an adaptive stepsize selection algorithm [13]. Here we will apply the time dependent solution in a tau-leaping framework. Thus the analytical solution can be used to approximate bimolecular reactions such as $S_1 + S_2 \rightarrow \textit{something}$ within a tolerance. It will inherit the adaptive stepsize selection method naturally as well.

Generally speaking, the time dependent solution is not easy to derive for an arbitrary network motif. However, for some common motifs we do have time dependent solutions. These solutions can be used to improve the performance of tau-leaping for some widely used models like the enzyme-substrate model.

The remainder of this paper is organized as follows. In Section 2, we provide a brief introduction to tau-leaping with adaptive timestep selection. In Section 3 we derive the time dependent solution for some common network motifs. We begin with a simple example to demonstrate the tau-leaping algorithm using the time dependent solution. Then we extend the algorithm to more general cases. Numerical experiments are provided in Section 4, including application of the method to a realistic model of blood coagulation, and the algorithm is briefly summarized in Section 5. Detailed mathematical derivations are provided in the supplementary material.

2. Tau-leaping

Consider a system of N species $\{S_1, \ldots, S_N\}$ and M reactions $\{R_1, \ldots, R_M\}$. The state vector of the system is $\boldsymbol{X} = \{x_1, \ldots, x_N\}$ which is the population of each of the species. The probability that reaction R_i fires in an infinitesimal interval dt is given by $a_i(\boldsymbol{X})dt$, where $a_i(\boldsymbol{X})$ is the propensity function of R_i. Tau-leaping advances the system in small steps; it assumes that the state vector $\boldsymbol{X}$ changes so little in each step that the propensity functions $\{a_1, \ldots, a_M\}$ can be treated as constants. Thus the number of firings in each reaction channel R_i is a Poisson random number with parameter $a_i(\boldsymbol{X})\tau$, where τ is the stepsize. To advance the system, we need only to sample these Poisson random numbers and update the state vector $\mathbf{X}$.

Yang et al. [13] suggest a strategy to determine the stepsize. The idea is that it should be chosen so that the mean and standard deviation of the change of each species is small compared to its population. Denoting the population change of species S_i as Δx_i, the stepsize as τ, and the number of firings of each reaction during a step as $r_1(\tau), \ldots, r_M(\tau)$, tau leaping computes

$$\Delta x_i = \sum_{j=1}^{M} v_{ij} r_j(\tau),$$

where v_{ij} is the stoichiometry of species S_i in reaction R_j. Assuming that the reaction firings are independent during a step, the mean and variance of Δx_i are given by

$$\mathbb{E}\Delta x_i = \sum_{j=1}^{M} v_{ij}\mathbb{E}\big(r_j(\tau)\big), \quad \mathrm{Var}(\Delta x_i) = \sum_{j=1}^{M} v_{ij}^2 \mathrm{Var}\big(r_j(\tau)\big).$$

Keeping $\mathbb{E}\Delta x_i$ and $\sqrt{\mathrm{Var}\Delta x_i}$ small (relative to the tolerance ϵ) compared with x_i requires [13]

$$\mathbb{E}\Delta x_i \leqslant \max\left(\frac{\epsilon}{g_i}x_i, 1\right), \quad \sqrt{\mathrm{Var}(\Delta x_i)} \leqslant \max\left(\frac{\epsilon}{g_i}x_i, 1\right), \tag{1}$$

where g_i is a constant that depends on the highest order of the reactions which involve S_i as a reactant. Solving the above inequalities yields the upper bound on τ, which we will denote by τ_i, for which species S_i can be expected to change by less than the prescribed tolerance. The adaptive tau-leaping algorithm chooses the smallest τ_i as its stepsize.

$$\tau = \min_{1 \leqslant i \leqslant N} \tau_i. \tag{2}$$

Over a step of size τ, tau-leaping approximates the population of every species as a constant. Thus $r_i(\tau)$ is a Poisson random variable

$$r_i(\tau) \sim \mathcal{P}(a_i\tau).$$

Solving (1) for τ_i gives

$$\tau_i \leqslant \frac{\max\left(\frac{\epsilon}{g_i}x_i, 1\right)}{\sum_{j=1}^{M} v_{ij}a_j}, \tau_i \leqslant \frac{\max\left(\frac{\epsilon^2}{g_i^2}x_i^2, 1\right)}{\sum_{j=1}^{M} v_{ij}^2 a_j} \Rightarrow \tau_i = \min\left(\frac{\max\left(\frac{\epsilon}{g_i}x_i, 1\right)}{\sum_{j=1}^{M} v_{ij}a_j}, \frac{\max\left(\frac{\epsilon^2}{g_i^2}x_i^2, 1\right)}{\sum_{j=1}^{M} v_{ij}^2 a_j}\right) \tag{3}$$

and substituting this into (2) yields the tau-leaping stepsize.

It is easy to see that tau-leaping can be substantially more efficient than SSA. However, this is only the case when it can use a stepsize over which many reaction firings would have taken place. However, if some species S_i is changing rapidly, then the change in that species may be constraining the stepsize. On each timestep, the species that is constraining the stepsize is the one for which τ_i is smallest. Thus we propose to use the time dependent solution described in the next section to solve for that species in place of standard tau-leaping (provided that it occurs in one of the common network motifs for which we have a time dependent solution).

Using the time dependent solution is a natural way to remove the stepsize constraint from the limiting species. This idea can also be extended to cases where several species require a very small stepsize. Though a general solution for arbitrary motifs may not be easy to find, we do have the solution for some common motifs. The results will be shown in the next section.

3. Tau-leaping using the time dependent solution

The time dependent solution makes use of the exact analytical solution of common reaction motifs to increase the speed of tau-leaping. The splitting method [12] also uses the analytical solution of monomolecular, catalytic bimolecular, and autocatalytic reactions. It separates these reactions from the system to form subsystems that can be simulated using their analytical solutions. The time dependent solution improves on the splitting method in the following two ways.

- Applicability to non-catalytic bimolecular reactions.
  In order to use the analytical solution for a bimolecular reaction, the splitting method requires that one of its reactants has zero stoichiometry (i.e. catalytic bimolecular reaction). The time dependent solution removes this requirement by observing that if one of the reactants of a non-catalytic bimolecular reaction has a slow relative rate of change, we should be able to allow it to use the analytical solution to within some tolerance.
  This change brings new requirements to the system partition strategy. In the splitting method the subsystems are determined by the stoichiometry. Thus it can partition the system at the very beginning and use it throughout the simulation. However, if we allow the subsystems to include non-catalytic bimolecular reactions, the stoichiometry matrix will not be sufficient to determine the partitioning of the system. We also need the information of the dynamically changing reaction rates. Thus the time dependent solution includes a scheme for dynamic partitioning.
- *Adaptive stepsize selection.*
  An operator bounding analysis for the splitting method was given in [12]. For simulation purposes, it would be ideal if the analysis can generate an algorithm to adaptively select the stepsize. Here, since our partition will be more complex and our implementation of the time dependent solution is in the tau-leaping framework, making use of the adaptive stepsize selection strategy from tau-leaping [13] is a more natural and easy option for our method.

In this section we will demonstrate the use of the time dependent solution using the tau-leaping method. We begin with a simple example.

3.1. Using the time dependent solution of one species

Let us take a look at one species in particular, say S_1. There are reactions which either generate or consume S_1, as shown in Fig. 1. We will refer to the motif illustrated in Fig. 1 as motif I in the following sections.

If for any reaction in the system, its reactants involve at most one S_1 molecule and its products also involve at most one S_1 molecule, then we can find the analytical solution for the population of S_1, under the assumption that the populations of

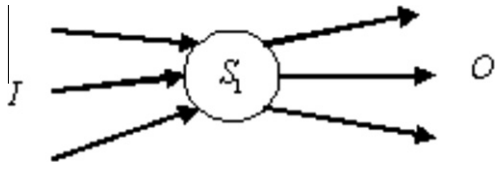

Fig. 1. Motif I, I denotes the set of reactions that generate S_1, and O denotes the set of reactions that consume S_1.

other species can be considered as constants. This assumption is reasonable as long as we use a stepsize that can be accepted by those other species.

Let I be the set of reactions that generate S_1, and O be the set of reactions that consume S_1. Denote the total propensity that an S_1 will be generated as

$$a_I \triangleq \sum_{R_i \in I} a_i$$

and the total rate that S_1 will be consumed as

$$c_O \triangleq \sum_{R_i \in O} \tilde{c}_i,$$

where $\tilde{c}_i = a_i/x_1$.

The time dependent population of S_1 can be written as (see Appendix A in the supplementary material)

$$x_1(t) \sim \mathcal{B}\left(x_1(0), e^{-c_O t}\right) + \mathcal{P}\left(\frac{a_I}{c_O}\left(1 - e^{-c_O t}\right)\right) \tag{4}$$

$$\sim \mathcal{B}\left(x_1(0), e^{-c_O t}\right) + \mathcal{B}\left(r_I, \frac{1}{c_O t}\left(1 - e^{-c_O t}\right)\right), \tag{5}$$

where $x_1(0)$ is the initial value of x_1 at the beginning of the step, and r_I is the input to S_1, i.e. the total number of firings for reactions in I. $\mathcal{B}(n,p)$ is a binomial random number with parameters n, p. $\mathcal{P}(\lambda)$ is a Poisson random number with parameter λ. The two random variables in (4) and (5) are independent.

The corresponding output from S_1, i.e. the total number of firings in O, is given by

$$r_O(t) \triangleq \sum_{R_i \in O} r_i(t) = x_1(0) + r_I - x_1(t)$$

$$\sim \mathcal{B}\left(x_1(0), 1 - e^{-c_O t}\right) + \mathcal{B}\left(r_I, 1 - \frac{1}{c_O t}\left(1 - e^{-c_O t}\right)\right). \tag{6}$$

To simulate the number of firings in each reaction channel $R_i \in O$, we distribute r_O using the multinomial distribution according to the rate $\tilde{c}_i$ of each reaction R_i

$$\{r_i : R_i \in O\} \sim \mathcal{M}\left(r_O, \frac{\tilde{c}_i}{c_O} : R_i \in O\right), \tag{7}$$

or equivalently (see Appendix C in the supplementary material),

$$r_i(t) \sim \mathcal{B}\left(x_1(0), \frac{\tilde{c}_i}{c_O}\left(1 - e^{-c_O t}\right)\right) + \mathcal{P}\left(\frac{\tilde{c}_i}{c_O}\left(a_I t - \frac{a_I}{c_O}\left(1 - e^{-c_O t}\right)\right)\right). \tag{8}$$

Here $\mathcal{M}(n, p_1, \ldots, p_n)$ is a multinomial random variable with parameters n and $p_1, \ldots, p_n$.

Now we apply this time dependent solution to accelerate tau-leaping for the simple example.

$$S_0 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_1 \overset{c_3}{\rightarrow} S_2.$$

When the population of S_0 is much greater than the population of S_1, S_1 will be the species that limits the tau-leaping stepsize. Using the time dependent solution of S_1 we arrive at the following algorithm.

1. Use (3) to compute the acceptable stepsizes τ_i for every species (in this case S_0 and S_1. There is no need to compute S_2 because it is a pure product and it never changes any propensity function).
2. Find the smallest τ_i (Here we assume $\tau_1 < \tau_0$ for demonstration purposes, so $I = \{R_1\},\ O = \{R_2, R_3\}$).
3. Recompute the stepsize. In this example we need to recompute τ_0 for S_0. We do this because the original τ_0 was based on the assumption that x_1 is a constant during the step. Since this is no longer the case, we need to reevaluate τ_0. To do this, we still try to bound the mean and variance of Δx_0 using (1). The only change is that the number of firings of R_2 is no longer a Poisson random variable. Instead, we have formula (8) for r_2, so both $\mathbb{E}(r_2)$ and $\mathrm{Var}(r_2)$ can be obtained explicitly and used to compute the new value for τ_0. (Here we need to solve a nonlinear algebraic equation since $\mathbb{E}(r_2)$ and $\mathrm{Var}(r_2)$ contain $e^{-c_O t}$ terms. Newton iteration is a good option because the explicit formulas of the equations are known).
4. Sample the number of firings in all reaction channels except those belonging to O (Sample $r_1(\tau)$ in the example). These reactions do not depend on the species for which we use the time dependent solution (S_1 in the example), so the original strategy in tau-leaping still works. Reactions in I are sampled in this step so that we know the value of r_I.
5. Sample r_O using (6) and distribute it into each channel in O using (7). (Now r_2 and r_3 have been sampled).
6. Update the system and start the next step, or terminate if the end time of the simulation has been reached.

In some reacting systems, there can be reactions that use S_1 as a catalyst. For example, suppose that we add the following reaction R_4 to the above system

$$R_4 : S_1 \xrightarrow{c_4} S_1 + S_3.$$

This reaction cannot be sampled using a Poisson random number $\mathcal{P}(c_4 x_1(0)\tau)$ in the previous framework, since S_1 may undergo a big change during the step. This reaction does not belong to O, since it does not consume S_1. It needs to be treated as a different case.

The value of r_4 during a step is given by

$$r_4 \sim \mathcal{P}\left(\int_0^\tau c_4 x_1(t)dt\right).$$

Since we cannot compute the integral exactly, we will need to make an approximation. A natural choice is to use the mean value $\mathbb{E}(x_1(t))$ instead of the exact random number $x_i(t)$, which yields

$$r_4 \approx \mathcal{P}\left(c_4 \int_0^\tau \mathbb{E}(x_1(t))dt\right). \tag{9}$$

This value is capable of being sampled, since we can derive the formula for $\mathbb{E}(x_1)$ from (4). Thus we have a formula for the integral expression. This approximation can capture the mean value of r_4 accurately but its variance is smaller than the exact value of $\mathrm{Var}(r_4)$ (see Appendix B in the supplementary material). This is because $\mathbb{E}(x_1(t))$ averages $x_1(t)$, thus it loses the specific information of the trajectory. To recover the variance, we need to include this information in the approximation. Since in Step 5 of the algorithm $x_1(\tau)$ is sampled (more precisely, we sample r_O, however we can get $x_1(\tau)$ by $x_1(\tau) = x_1(0) + r_I - r_O(\tau)$), it would be advantageous if we could include this information in the approximation. This yields another approximation formula:

$$r_4 \approx \mathcal{P}\left(c_4 \int_0^\tau \left(\mathbb{E}(x_1(t)) + \frac{t}{\tau}(x_1(\tau) - \mathbb{E}(x_1(\tau)))\right)dt\right) \sim \mathcal{P}\left(c_4\left(\int_0^\tau \mathbb{E}(x_1(t))dt + \frac{\tau}{2}(x_1(\tau) - \mathbb{E}(x_1(\tau)))\right)\right). \tag{10}$$

The interpolation of the difference between $x_1(t)$ and $\mathbb{E}(x_1(t))$ at the end time of the step has been added into the integrand. Numerical experiments (Section 4) demonstrate that (10) gives a much better approximation of the variance $\mathrm{Var}(r_4)$.

Armed with the strategy of using the time dependent solution for one species, we can move onto the more general case where we use the time dependent solution of several species.

3.2. Using the time dependent solution of several species

In many cases there are several species that are limiting the stepsize. They may be linked with each other via the reactions in which they participate. Consider, for example, the motif shown in Fig. 2. We will refer to this motif as motif II in the following sections.

A popular model that uses this motif is the enzyme substrate system,

$$E + S \underset{c_2}{\overset{c_1}{\rightleftharpoons}} ES \xrightarrow{c_3} E + P,$$

where S has a huge population while E and ES are present in small populations. Let τ_E, τ_S and τ_{ES} denote the stepsizes for E, S and ES given by (3). It is obvious that $\tau_E,\ \tau_{ES} \ll \tau_S$. Thus if we want to accelerate the simulation, we need to use the time dependent solution for both E and ES.

In general, the population of the enzyme is dynamic rather than constant. It can be produced and consumed by other reactions. For example, consider adding the following set of reactions into the enzyme substrate system:

$$R_4 : \phi \xrightarrow{a_4} E, \quad R_5 : E \xrightarrow{c_5} \phi, \quad R_6 : ES \xrightarrow{c_6} \phi.$$

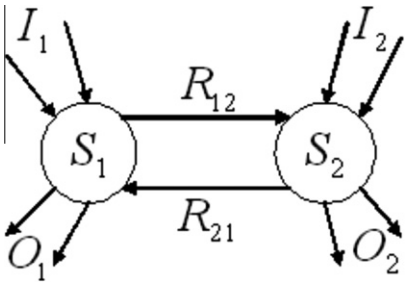

Fig. 2. Motif II, I_i denotes the set of reactions that generate S_i without consuming S_j; O_i denotes the set of reactions that consume S_i without generating S_j; R_{ij} denotes the set of reactions that consume S_i and generate S_j at the same time, $i, j = 1, 2,\ i \neq j$.

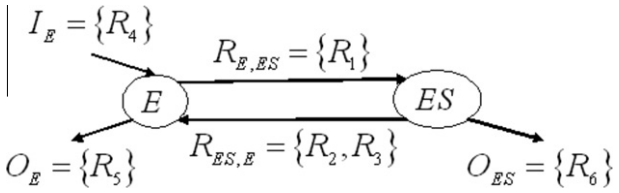

Fig. 3. E and ES are within the scope of motif II, R_4 is the input reaction for E, and R_5 and R_6 are the output reactions for E and ES respectively. R_1 converts E to ES, R_2 and R_3 convert ES to E.

This model is still within the scope of motif II (see Fig. 3). The good news is that we have the analytical solution for the time dependent solution of E and ES for the previous system during a stepsize of τ_S (which implies that S can be treated as constant).

Before giving the formula, we define some notation. Let $I_E = \{R_4\}$ be the set of reactions that generate E while not consuming ES, $O_E = \{R_5\}$ be the set of reactions that consume E while not producing ES, $O_{ES} = \{R_6\}$ be the set of reactions that consume ES while not producing E, $R_{E,ES} = \{R_1\}$ be the set of reactions that consume E and generate ES, and $R_{ES,E} = \{R_2, R_3\}$ be the set of reactions that consume ES and generate E.

Similar to the previous example, we have

$$\begin{aligned} a_I^E &= \sum_{R_i \in I_E} a_i = a_4, \quad r_I^E = \sum_{R_i \in I_E} r_i = r_4 \\ c_{E,ES} &= \sum_{R_i \in R_{E,ES}} \tilde{c}_i = c_1 x_S \\ c_{ES,E} &= \sum_{R_i \in R_{ES,E}} \tilde{c}_i = c_2 + c_3 \\ c_O^E &= \sum_{R_i \in O_E} \tilde{c}_i = c_5, \quad c_O^{ES} = \sum_{R_i \in O_{ES}} \tilde{c}_i = c_6 \end{aligned} \tag{11}$$

and

$$r_O^E = \sum_{R_i \in O_E} r_i = r_5, \quad r_O^{ES} = \sum_{R_i \in O_{ES}} r_i = r_6. \tag{12}$$

Here r_O^E and r_O^{ES} are the total number of firings for reactions in O_E and O_{ES}.

Using the notation above, the time dependent solution of this system can be written as

$$\begin{aligned} &\left(x_E(t), x_{ES}(t), r_O^E(t), r_O^{ES}(t)\right) \sim \mathcal{M}\left(x_E(0), p_1^E(t), p_2^E(t), p_{O1}^E(t), p_{O2}^E(t)\right) \\ &\quad + \mathcal{M}\left(x_{ES}(0), p_1^{ES}(t), p_2^{ES}(t), p_{O1}^{ES}(t), p_{O2}^{ES}(t)\right) + \mathcal{M}\left(r_I^E, \frac{\lambda_1(t)}{a_I^E t}, \frac{\lambda_2(t)}{a_I^E t}, \frac{\lambda_{O1}(t)}{a_I^E t}, \frac{\lambda_{O2}(t)}{a_I^E t}\right), \end{aligned} \tag{13}$$

where the formulas for each parameter are given in Appendix A in the supplementary material (see (A28) in Appendix A).

This result can be extended from two species to n species $\widehat{S} = \{S_1, \ldots, S_n\}$ when the following condition holds:

Condition $(*)$: *For any reaction R that can change the population of a species in $\widehat{S}$, one firing of R consumes at most one molecule in $\widehat{S}$, and produces at most one molecule in $\widehat{S}$.*

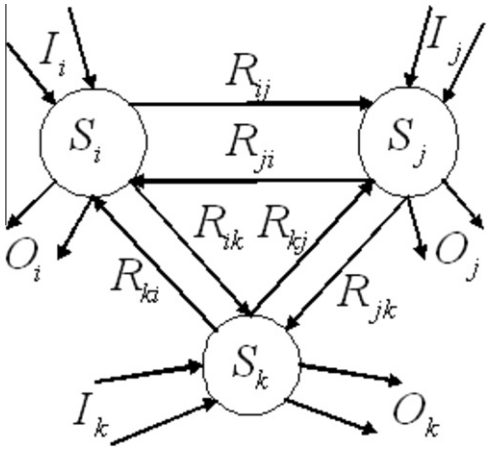

Fig. 4. General motif.

A diagram of this general motif is given in Fig. 4.

Now the definitions in (11) and (12) can be extended for any $1 \leqslant i \neq j \leqslant n$ as follows:

$$a_I^i \triangleq \sum_{R_k \in I_i} a_k, \quad r_I^i \triangleq \sum_{R_k \in I_i} r_k, \quad c_{ij} \triangleq \sum_{R_k \in R_{ij}} \tilde{c}_k, \quad c_O^i \triangleq \sum_{R_k \in O_i} \tilde{c}_k, \quad r_O^i \triangleq \sum_{R_k \in O_i} r_k.$$

The time dependent solution for this general motif is given by

$$(\mathbf{x}(t), \mathbf{r}_O(t)) \sim \sum_{i=1}^{n} \mathcal{M}\left(x_i(0), \mathbf{p}^i(t), \mathbf{p}_O^i(t)\right) + \sum_{i=1}^{n} \mathcal{M}\left(r_I^i, \frac{1}{a_I^i t}\boldsymbol{\lambda}^i, \frac{1}{a_I^i t}\boldsymbol{\lambda}_O^i\right), \tag{14}$$

where the formulas for each parameter are given in Appendix A in the supplementary material.

Now that we have the time dependent solution for our motifs, it is time to outline the steps of employing the time dependent solution in tau-leaping, using the enzyme substrate (E-S) system as an example.

1. Use (3) to compute the acceptable stepsizes τ_i for every species (in the E-S example we compute the stepsizes for $E,\ S$ and ES). For demonstration purposes, we assume $\tau_1 \leqslant \tau_2 \leqslant \ldots \leqslant \tau_N$ (and in the E-S example we have $\tau_E,\ \tau_{ES} < \tau_S$).
2. Construct the set of species U for which we will use the time dependent solution. Start from the species with the smallest stepsize, i.e. S_1. If S_1 satisfies condition $(*)$, add it into U to obtain $U = \{\{S_1\}\}$. Now go onto the species which has the second smallest stepsize, i.e. S_2. If $\{S_1, S_2\}$ does not satisfy condition $(*)$, end step 2 with $U = \{\{S_1\}\}$. Otherwise, add S_2 into U. If S_2 is linked to S_1, i.e. $c_{12} \neq 0$ or $c_{21} \neq 0$, add S_2 into U to obtain $U = \{\{S_1, S_2\}\}$. Otherwise add it into U to obtain $U = \{\{S_1\}, \{S_2\}\}$. Continue adding species into U in a similar way until you cannot add any more species that satisfy the condition $(*)$. Now each element in U is a set of species for which we can use the time dependent solution. (In the E-S example we end up with $U = \{\{E, ES\}\}$. We cannot add S into U since $\widehat{S} = \{E, ES, S\}$ does not satisfy condition $(*)$, as R_1 consumes two molecules in $\widehat{S}$).
3. Recompute the stepsize. For species not in U, we need to recompute their stepsizes with the new value of each r_i which may no longer be the original Poisson random variable (see Appendix C in the supplementary material for a more detailed computation. In the E-S example, we need to recompute the stepsize τ_S).
4. Sample the number of firings for all reactions that do not involve the species in U as reactants. For these reactions tau-leaping is appropriate, so sample Poisson random numbers for them (in the E-S example, r_4 is sampled).
5. Sample each element in U using its time dependent solution (14). (In the E-S example, $x_E(t),\ x_{ES}(t),\ r_O^E(t),\ r_O^{ES}(t)$ are sampled)
6. For each species S_i in U, sample reactions in O_i using the multinomial distribution

$$\{r_j : R_j \in O_i\} \sim \mathcal{M}\left(r_O^i, \frac{\tilde{c}_j}{c_O^i} : R_j \in O_i\right).$$

   (In the E-S example, r_5 and r_6 are sampled, and the multinomial distribution yields $r_5 = r_O^E,\ r_6 = r_O^{ES}$).

7. Sample the reactions in R_{ij}. This is not trivial since we have to maintain the flow conservation of the network, so what we actually sample is an instance of a feasible flow. An algorithm to sample the flow is presented in Appendix D in the supplementary material. For the E-S example, this step is very simple. First sample r_1 using formula (10). Here $\mathbb{E}(x_E(t))$ in the formula has the form (see Appendix A in the supplementary material for detailed derivation)

   $$\mathbb{E}(x_E(t)) = x_E(0)p_1^E(t) + x_{ES}(0)p_1^{ES}(t) + \lambda_1(t),$$

   where $p_1^E(t),\ p_1^{ES}(t)$ and $\lambda_1(t)$ are the parameters that appeared in (13).
   The conservation equation

   $$r_4 + x_E(0) + (r_2 + r_3) = x_E(t) + r_1 + r_5,$$

   gives

   $$(r_2 + r_3) = x_E(t) + r_1 + r_5 - r_4 - x_E(0).$$

   Then sample r_2 and r_3 from their sum using the binomial distribution

   $$r_2 = \mathcal{B}\left(x_E(t) + r_1 + r_5 - r_4 - x_E(0), \frac{c_2}{c_2 + c_3}\right),$$
   $$r_3 = x_E(t) + r_1 + r_5 - r_4 - x_E(0) - r_2.$$

8. If there are reactions involving species in U that are acting as a catalyst, for example

   $$S_i \to S_i + S_j,$$

where S_j is not in U (this is guaranteed by the algorithm, because species in U satisfies condition $(*)$), use formula (10) to approximate the number of their firings. In the E-S example there is no such reaction.

9. Update the system and begin the next step or terminate if the end time of the simulation has been reached.

This algorithm is adaptive in the sense that it always applies the time dependent solution to the motifs which limit the tau-leaping stepsize, even though the limiting motifs change during the simulation. We achieve this goal by constructing the limiting motifs U on the fly in step 2, rather than partitioning the system at the beginning of the simulation.

In the enzyme substrate example, allowing non-catalytic bimolecular reactions to be grouped into the motif plays an important role. If such an operation is not allowed, reaction $R_1 : E + S \to ES$ will be taken away from the motif and we will have a partition of the system as $I_1 = \{R_1\},\ I_2 = \{R_2, \ldots, R_6\}$. This partition will significantly decrease the stepsize because I_1 takes into account only the reaction that converts E to ES, while I_2 includes the reactions in the opposite direction. Thus if we use a big stepsize, E will be depleted in subsystem I_1 in a short time, as will ES in R_2. During the remaining time of the step, the system will do nothing. This is obviously not the correct physics of the model. Our method can avoid this partition because we allow R_1 to be included in the motif as shown in Fig. 3. Thus the motif contains all the reactions in both directions and it can take a much longer stepsize than the previous partition.

4. Numerical simulation

In this section we present the results for the numerical simulations of the examples in Section 3. We also demonstrate the time dependent solution for a more complex real world model of blood coagulation.

4.1. Example 1

The first example is the one mentioned in Section 3.1:

$$S_0 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_1 \xrightarrow{c_3} S_2,$$
$$S_1 \xrightarrow{c_4} S_1 + S_3.$$

The parameters are taken to be $c_1 = 0.1,\ c_2 = 1,\ c_3 = 1,\ c_4 = 1$. The initial population of each species is given by $x_0 = 1e+6,\ x_1 = x_2 = x_3 = 0$. The result of a one second simulation is shown in Table 1.

In this example, the stepsize for S_1 is smaller than the stepsize for S_0, thus the stepsize of tau-leaping is constrained by the stepsize for S_1. Using the time dependent solution of S_1, we can remove the stepsize requirement of S_1 (which tries to keep x_1 almost constant during the step) and use the stepsize of S_0 for the simulation, which yields a huge speedup. If we use the time dependent solution of both S_1 and S_0, we have no stepsize requirement at all! The last method in Table 1 simply samples the population of each species at time $t = 1$ directly. This explains why it is so fast.

Speed is important, however we do not want to trade speed at the cost of losing too much accuracy. The population distributions given by SSA and the last method in Table 1 are compared in Fig. 5. The result shows that accuracy is not sacrificed. The distribution of every species is maintained.

Formula (10) plays an important role for sampling the population of S_3. If we use only the mean value of x_1 to do the sampling, i.e. using (9), the distribution will have a noticeable error. Fig. 6 shows the distribution of S_3 if (9) is used. The distribution has the correct mean but the variance is too small.

4.2. Example 2

The second example is the one we used in Section 3.2:

$$E + S \underset{c_2}{\overset{c_1}{\rightleftharpoons}} ES \xrightarrow{c_3} E + P$$
$$\phi \xrightarrow{a_4} E, \quad E \xrightarrow{c_5} \phi, \quad ES \xrightarrow{c_6} \phi.$$

Table 1
The time used for 100000 realizations of the one second simulation for Example 1, $\epsilon = 0.003$.

Method	Time used (s)
SSA	5943.97
Tau leaping	1006.84
Tau leaping/TDS[a]	8.18854
Tau leaping/TDS[b]	1.30296

[a] Tau leaping using time dependent solution of motif I.
[b] Tau leaping using time dependent solution of motif II.

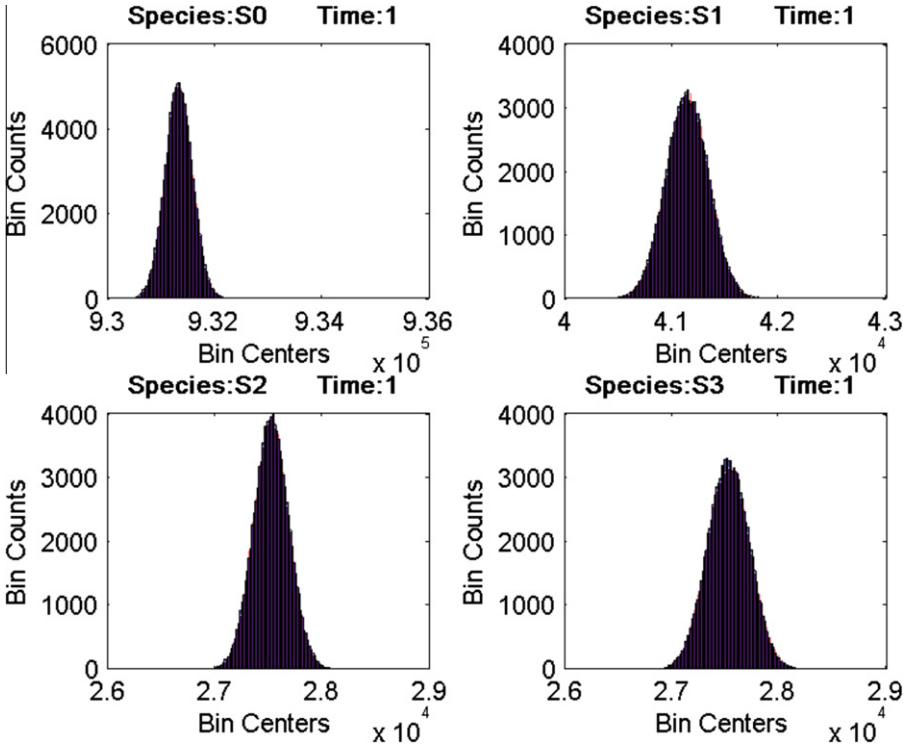

Fig. 5. Histograms of each species in Example 2. Comparison of result given by SSA and tau-leaping using time dependent solution of motif II. Red is SSA, blue is tau-leaping using time dependent solution, and purple is the overlap of the two histograms.

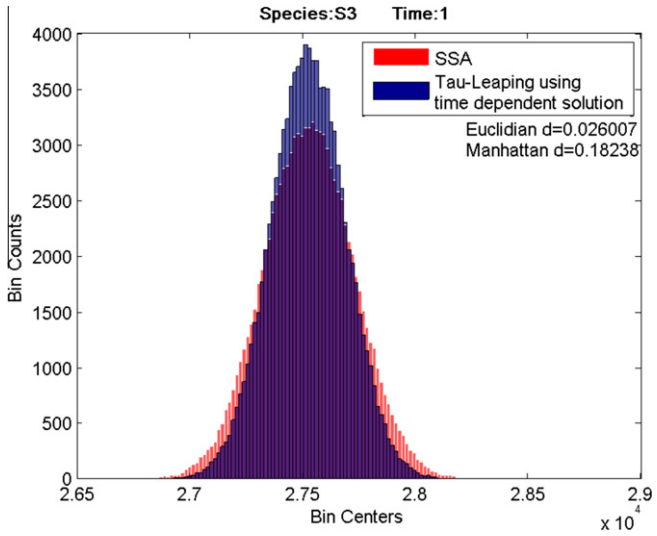

Fig. 6. The distribution of S_3 if (9) is used. It has the correct mean value but the variance is too small.

The parameters were taken to be $c_1 = 0.0001,\ c_2 = 10,\ c_3 = c_5 = c_6 = 1,\ a_4 = 100$. The initial population was taken as $x_S = 1e+6,\ x_E = 1000,\ x_{ES} = x_P = 0$. We do a one second simulation. The results are shown in Table 2 and Fig. 7.

In this example it will not help much if we use the time dependent solution of only one species (the third method in Table 2). This is because both *E* and *ES* require a small stepsize, thus relaxing the stepsize requirement for one of them will not completely solve our problem. The last method in Table 2 uses the time dependent solution of both *E* and *ES*, thus the

Table 2
The time used for 100000 realizations of a one second simulation of Example 2 with $\epsilon = 0.003$.

Method	Time used (s)
SSA	519.708
Tau leaping	787.655
Tau leaping/TDS[c]	475.314
Tau leaping/TDS[d]	2.57195

[c] Tau leaping using time dependent solution of motif I.
[d] Tau leaping using time dependent solution of motif II.

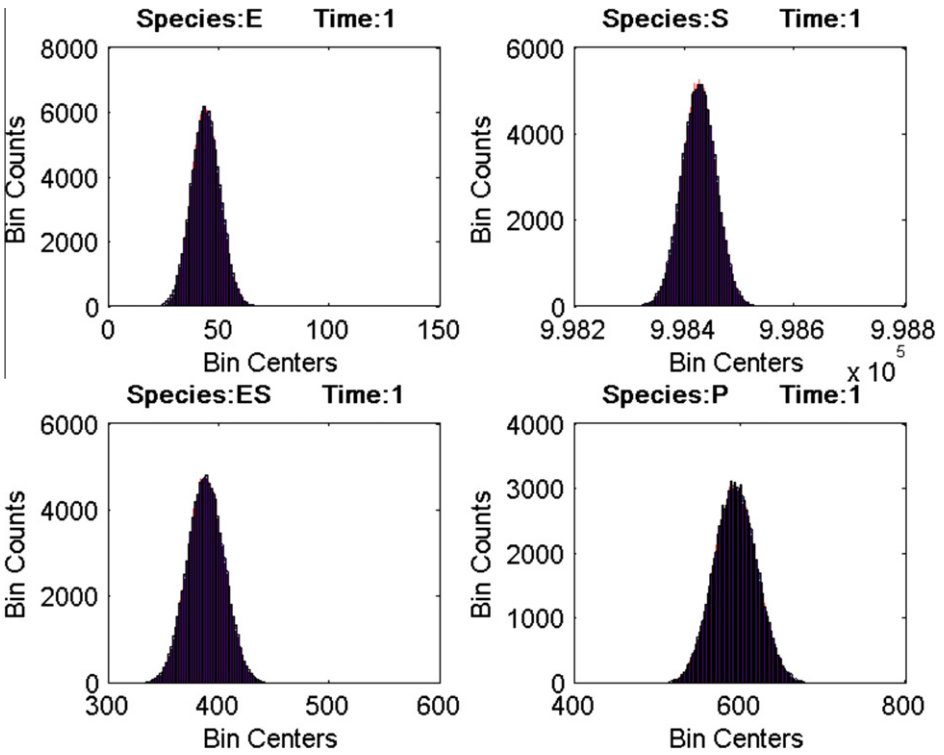

Fig. 7. Histograms of each species in Example 2. Comparison of result given by SSA and tau-leaping using time dependent solution of motif II. Red is SSA, blue is tau-leaping using time dependent solution, and purple is the overlap of the two histograms.

Table 3
The time used for one realization of a 700 s simulation of the coagulation model, with $\epsilon = 0.02$. The results are averaged over ten realizations.

Method	Time used (s)
SSA	273.498
Tau leaping	39.2127
Tau leaping/TDS[e]	7.61337

[e] Tau leaping using time dependent solution of motif I + II.

stepsize of the method is actually the stepsize of S, which is much larger than those of E and ES. In the simulation, the stepsize of S is greater than one second therefore the last method basically samples the population of each species at $t = 1$ directly.

4.3. Coagulation model

For the final example, we apply our method to a model of blood coagulation [14] with 43 reactions and 33 species. The coagulation model contains reaction pathways that form several levels of cascades. Different factors are activated at different time intervals, which finally leads to the activation of thrombin. Meanwhile, the negative regulation factor antithrombin III

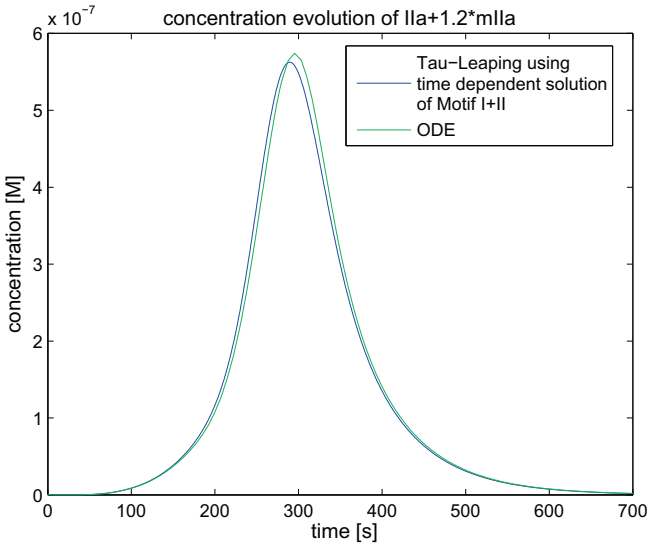

Fig. 8. Concentration of thrombin (IIa + 1.2 × mIIa). Blue curve: Tau-leaping using time dependent solution of motif I + II. Green curve: ODE. (For interpretation of the references to colours in this figure legend, the reader is referred to the web version of this article.)

binds to thrombin as well as to some other factors in order to control the coagulation process. In this model the species which constrain the stepsize vary as time goes on. However, we do not need to worry about this in the simulation. Our algorithm does not require any prior knowledge about the system. It automatically detects the motifs that limit the stepsize and applies the time dependent solution to them if applicable.

The original model uses concentration for each species rather than population. We convert the concentration to population by selecting a 1 mm long cylinder with diameter 0.01 mm as the control volume. The time used for one realization of a 700 s simulation is shown in Table 3.

The last method in Table 3 applies the time dependent solution of motifs I and II. We can see that it already is significantly faster compared to standard tau-leaping. We can expect that if we fully implement the algorithm and use the time dependent solution of motifs containing more than two species, it will further accelerate the speed of the simulation.

According to Table 3, if we do a 10000-realization simulation, it takes about 31.7 days for SSA, 4.5 days for tau-leaping, and about 21.1 h for the time dependent solution implemented as described above. We have code that can run the simulation in parallel. Thus the 10000-realization simulation using the third method required only 5.2 h running on a 4 core workstation. Since it takes too much time to obtain a complete SSA result of 10000 runs, we do not compare the species distributions for this model. Instead, we compare the evolution of thrombin's mean value with the result given by the ODE model. Here we plot the mean values of IIa + 1.2 × mIIa given by 10000 tau-leaping runs using the time dependent solution (blue) and the ODE model (green) in Fig. 8. The error tolerance of the adaptive tau leaping simulation is 0.02, which is larger than the previous examples, so the result will not be as accurate. However Fig. 8 shows that this result is already able to catch the trend of thrombin.

5. Conclusion

Tau-leaping using the time dependent solution provides a means to accelerate the simulation of systems that have rapidly changing species. The key point of the method is that it uses the time dependent solution for the fast changing species. Thus, it can use a much larger stepsize than standard tau-leaping, without noticeable loss of accuracy. The auto detection feature grants the algorithm the ability to handle systems whose fast changing species vary over time. However, the method still has some limitations.

1. It can handle only networks that satisfy condition (∗). If (∗) is violated, we may not have the formula for the time dependent solution. Actually, it is still possible to derive PDEs for the generating function, as we do in Appendix A in the supplementary material. However the PDEs will be second order and the analytical solution may not be easy to obtain. Even if we find the solution for the PDEs, we still need to convert them into proper random variables that are easy to sample, which is also nontrivial.
2. For systems that do not have fast-changing species, the method will not benefit the simulation.

The time-dependent solution for acceleration of tau-leaping is already applicable to many real-world systems. The formulas and hence the implementation are complicated, but we have automated the method so that this is not a limitation.

Acknowledgments

The authors acknowledge the following financial support: National Institute of Biomedical Imaging and Bioengineering (Grant No. 1R01EB014877-01); US Army Research Office (Grant No. W911NF-10-2-0114); Institute for Collaborative Biotechnologies from the US Army Research Office (Grant No. W911NF-09-D-0001) and DOE Contract No. DE-SC0008975.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jcp.2012.10.036.

References

[1] H.H. McAdams, A. Arkin, Stochastic mechanisms in gene expression, Proc. Natl. Acad. Sci. USA 94 (1997) 814–819.
[2] A. Arkin, J. Ross, H.H. McAdams, Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells, Genetics 149 (1998) 1633–1648.
[3] N. Fedoroff, W. Fontana, Small numbers of big molecules, Science 297 (2002) 1129–1131.
[4] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (1976) 403–434.
[5] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361.
[6] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J. Chem. Phys. 115 (2001) 1716–1733.
[7] Y. Cao, L.R. Petzold, Slow-scale tau-leaping method, Comput. Methods Appl. Mech. Eng. 197 (2008) 3472–3479.
[8] L.A. Segel, M. Slemrod, The quasi-steady-state assumption: a case study in perturbation, SIAM Rev. 31 (1989) 446–477.
[9] 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.
[10] Y. Cao, D.T. Gillespie, L.R. Petzold, The slow-scale stochastic simulation algorithm, J. Chem. Phys. 122 (2005) 014116.
[11] E.A. Mastny, E.L. Haseltine, J.B. Rawlings, Two classes of quasi-steady-state model reductions for stochastic kinetics, J. Chem. Phys. 127 (2007) 094106.
[12] T. Jahnke, D. Altintan, Efficient simulation of discrete stochastic reaction systems with a splitting method, BIT Numer. Math. 50 (2010) 797–822.
[13] Y. Cao, D.T. Gillespie, L.R. Petzold, Efficient step size selection for the tau-leaping simulation method, J. Chem. Phys. 124 (2006) 044109.
[14] M.F. Hockin, K.C. Jones, S.J. Everse, K.G. Mann, A model for the stoichiometric regulation of blood coagulation, J. Biol. Chem. 277 (2002) 18322–18333.