Time Dependent Solution for Acceleration of Tau-Leaping

Jin Fu[a,*], Sheng Wu[a], Linda R. Petzold[a]

[a]*Department of Computer Science, University of California, Santa Barbara*

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 to 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.

*Corresponding author

Email addresses: iamfujin@hotmail.com (Jin Fu), sheng@cs.ucsb.edu (Sheng Wu), petzold@cs.ucsb.edu (Linda R. Petzold)

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, 2, 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, 10, 11]. The essence of this strategy is to divide the system into fast and slow subsystems. If the fast 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 to 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 approximatie bimolecular reactions such as $S_1 + S_2 \rightarrow \mathit{something}$ 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 $\boldsymbol{X}$.

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

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

where ν_{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

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

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]

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

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.

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

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

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

Solving (1) for τ_i gives

$$\begin{aligned} \tau_i \leq \frac{\max\left(\frac{\epsilon}{g_i}x_i, 1\right)}{\sum_{j=1}^{M}\nu_{ij}a_j}, \quad \tau_i \leq \frac{\max\left(\frac{\epsilon^2}{g_i^2}x_i^2, 1\right)}{\sum_{j=1}^{M}\nu_{ij}^2 a_j} \\ \Rightarrow \tau_i = \min\left(\frac{\max\left(\frac{\epsilon}{g_i}x_i, 1\right)}{\sum_{j=1}^{M}\nu_{ij}a_j}, \frac{\max\left(\frac{\epsilon^2}{g_i^2}x_i^2, 1\right)}{\sum_{j=1}^{M}\nu_{ij}^2 a_j}\right), \end{aligned} \tag{3}$$

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.

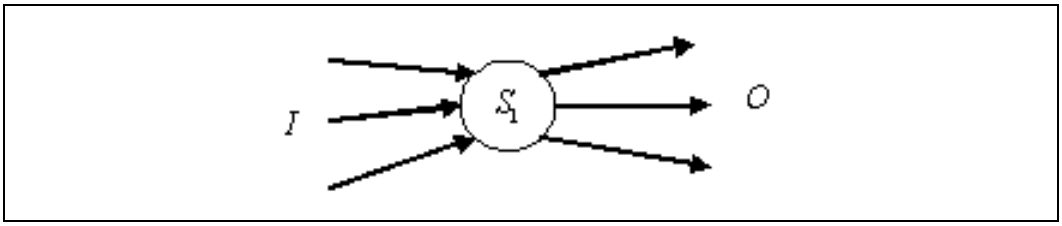

Figure 1: Motif I, I denotes the set of reactions that generate S_1, and O denotes the set of reactions that consume S_1.

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 Figure 1. We will refer to the motif illustrated in Figure 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 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

$$a_I \stackrel{\triangle}{=} \sum_{R_i \in I} a_i,$$

and the total rate that S_1 will be consumed as

$$c_O \stackrel{\triangle}{=} \sum_{R_i \in O} \tilde{c}_i,$$

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 $\mathcal{M}\left(n,\ p_1,\ldots,p_n\right)$ 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.

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

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:

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

The interpolation of the difference between $x_1(t)$ and $\mathbb{E}(x_1(t))$ at the end

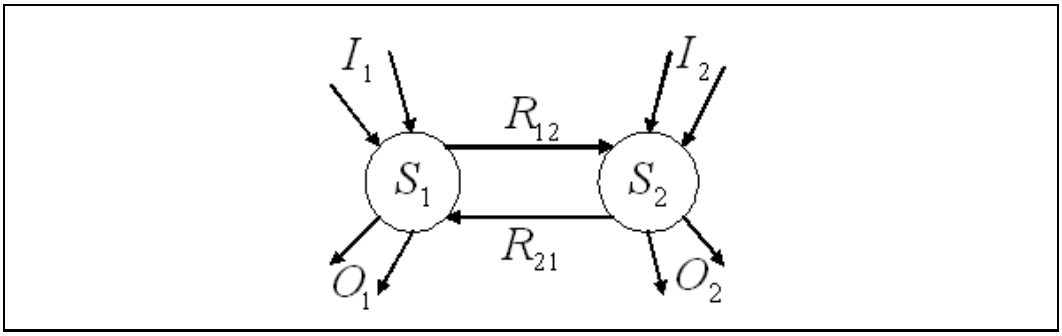

Figure 2: Motif II, I_i denotes the set of reactions that generate S_i without consuming S_j; O_i denotes the set of reactions that consume S_i without generating S_j; R_{ij} denotes the set of reactions that consume S_i and generate S_j at the same time, $i, j = 1, 2,\ i \neq j$.

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 on to 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 Figure 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,

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

where S has a huge population while E and ES are present in small pop-

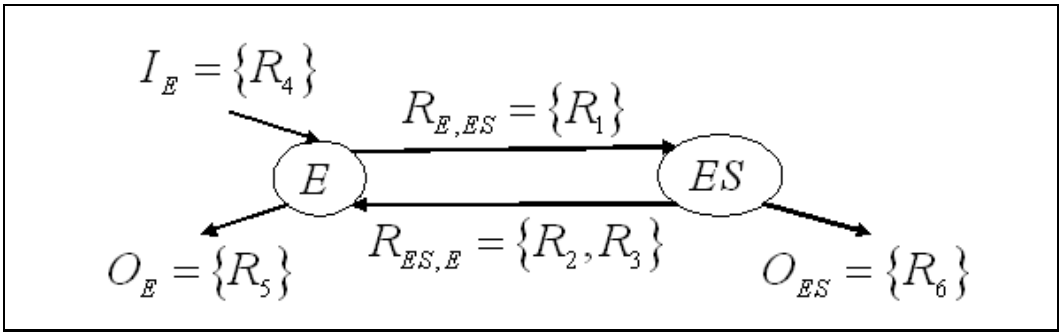

Figure 3: E and ES are within the scope of Motif II, R_4 is the input reaction for E, and R_5 and R_6 are the output reactions for E and ES respectively. R_1 converts E to ES, R_2 and R_3 convert ES to E.

ulations. 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:

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

This model is still within the scope of Motif II (see Figure 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

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

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

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

where the formulas for each parameter are given in Appendix A in the sup-

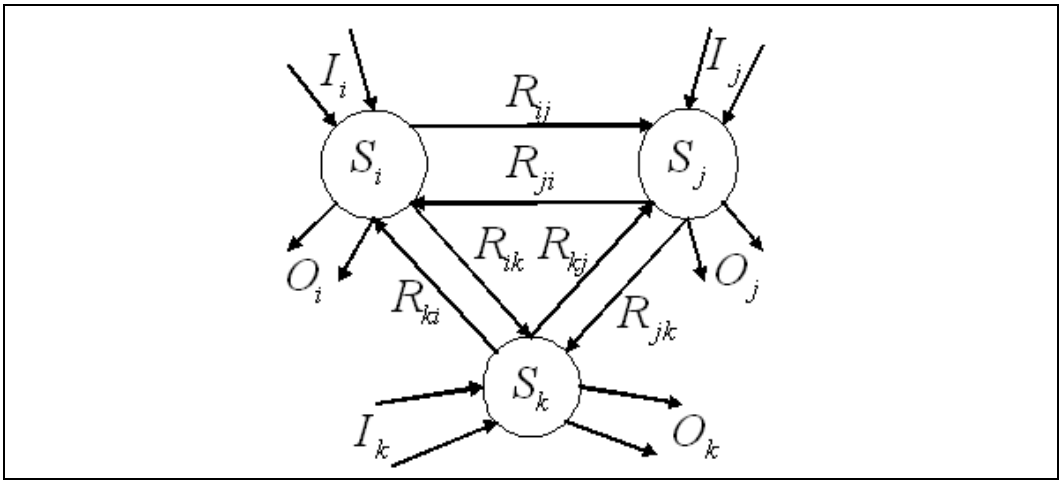

Figure 4: General motif

plementary material (see (A28) in Appendix A).

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

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

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

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

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

The time dependent solution for this general motif is given by

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

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 $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$ (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 on to 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 $\hat{S} = \{E,\ ES,\ S\}$ does not satisfy condition $(*)$, as R_1 consumes two molecules in $\hat{S}$).

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 $\mathbb{E}\left(x_E(t)\right)$ in the formula has the form (see Appendix A in the supplementary material for detailed derivation)

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

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 $R_1 : E + S \rightarrow ES$ 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 Figure 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.

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

Method	Time used
SSA	5943.97s
Tau Leaping	1006.84s
Tau Leaping/TDS[1]	8.18854s
Tau Leaping/TDS[2]	1.30296s

[1]Tau Leaping using time dependent solution of Motif I
[2]Tau Leaping using time dependent solution of Motif II

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 don't want to trade speed at the cost of

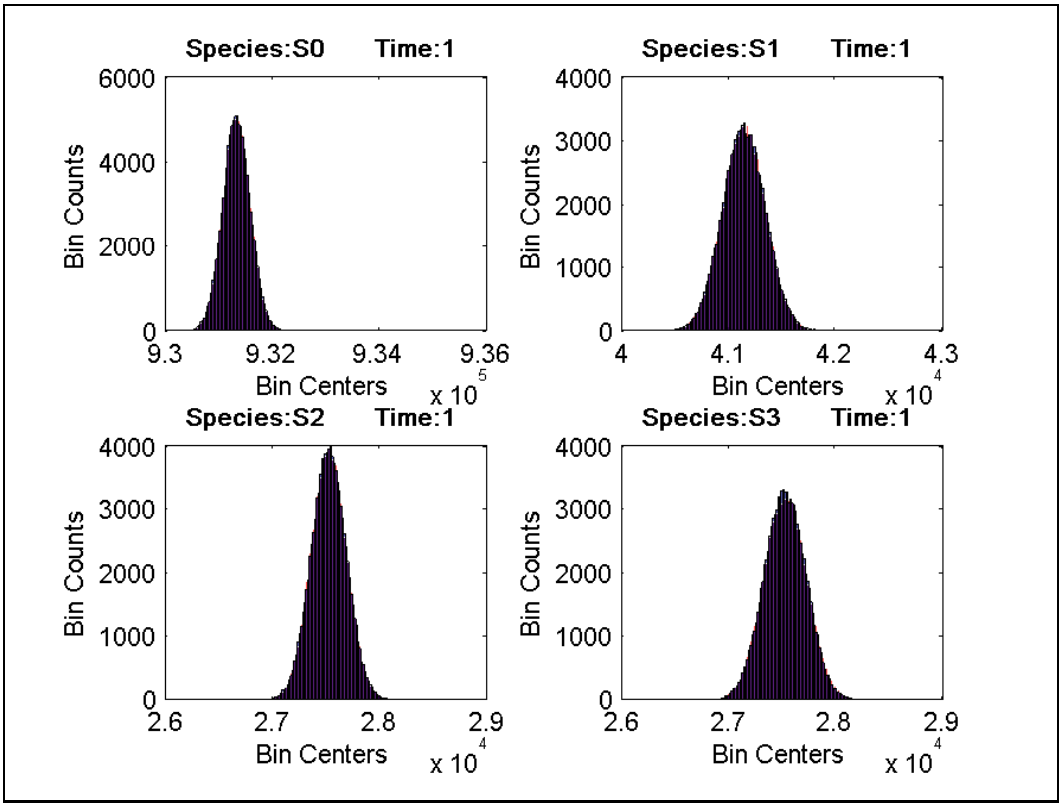

Figure 5: Histograms of each species in Example 1. Comparison of result given by SSA and tau-leaping using time dependent solution of Motif II. Red is SSA, blue is tau-leaping using time dependent solution, and purple is the overlap of the two histograms.

losing too much accuracy. The population distributions given by SSA and the last method in Table 1 are compared in Figure 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. Figure 6 shows the distribution of S_3 if (9) is used. The distribution has the correct mean but the variance is too small.

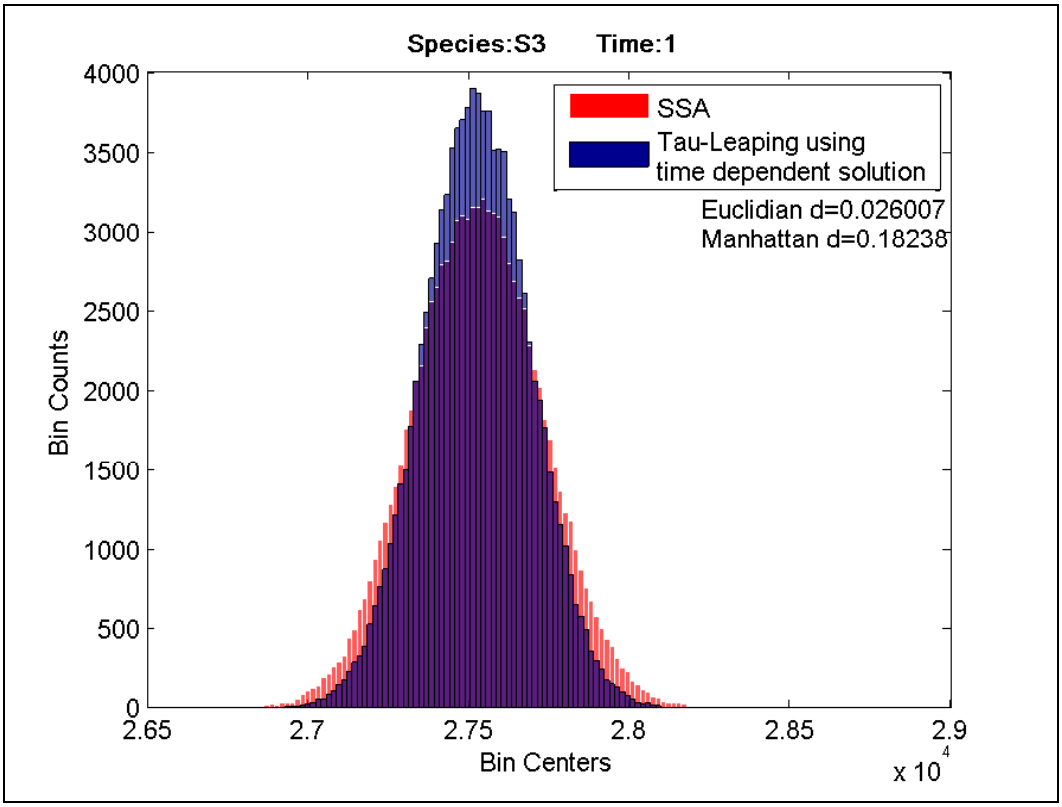

Figure 6: The distribution of S_3 if (9) is used. It has the correct mean value but the variance is too small.

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

Method	Time used
SSA	519.708s
Tau Leaping	787.655s
Tau Leaping/TDS[1]	475.314s
Tau Leaping/TDS[2]	2.57195s

[1]Tau Leaping using time dependent solution of Motif I
[2]Tau Leaping using time dependent solution of Motif II

4.2. Example 2

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

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

$$\phi \xrightarrow{a_4} E, \quad E \xrightarrow{c_5} \phi, \quad ES \xrightarrow{c_6} \phi.$$

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 Figure 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 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

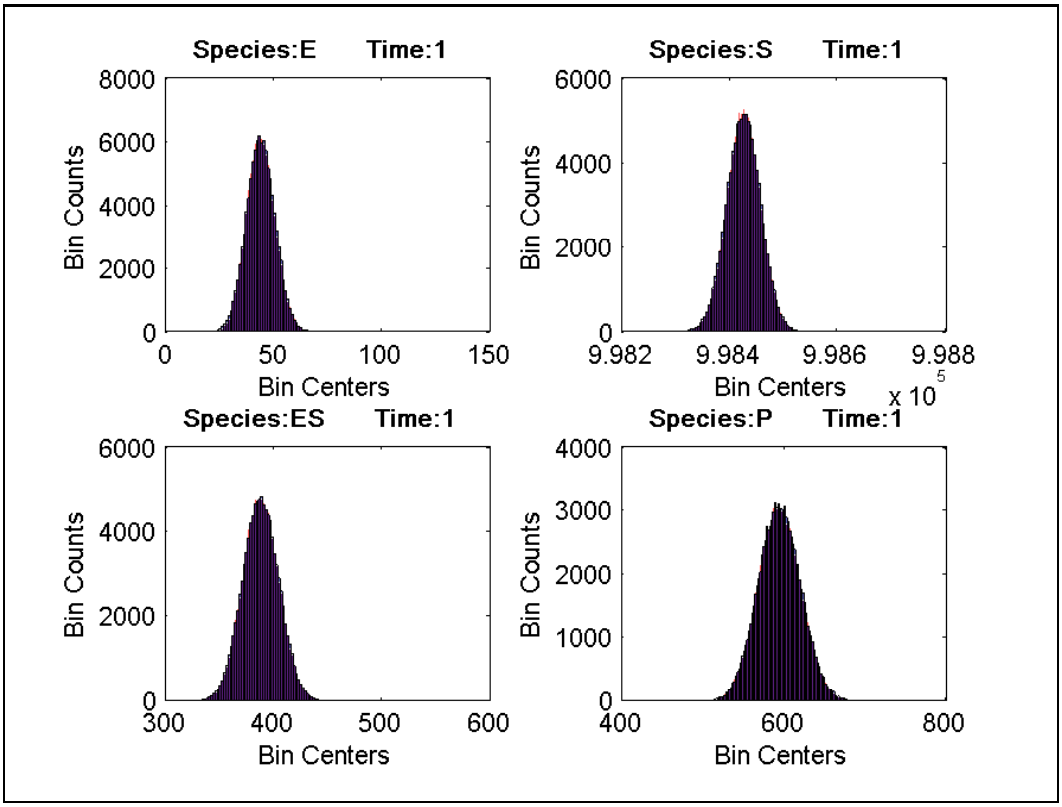

Figure 7: Histograms of each species in Example 2. Comparison of result given by SSA and tau-leaping using time dependent solution of Motif II. Red is SSA, blue is tau-leaping using time dependent solution, and purple is the overlap of the two histograms.

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

Method	Time used
SSA	273.498s
Tau Leaping	39.2127s
Tau Leaping/TDS[1]	7.61337s

[1]Tau leaping using time dependent solution of Motif I+II.

$t = 1$ 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 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 1mm long cylinder with diameter 0.01mm as the control volume. The time used for one realization of a 700 second simulation is shown in Table 3.

The last method in Table 3 applies the time dependent solution of Motif I and Motif 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 hours 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 hours 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 Figure 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 Figure 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

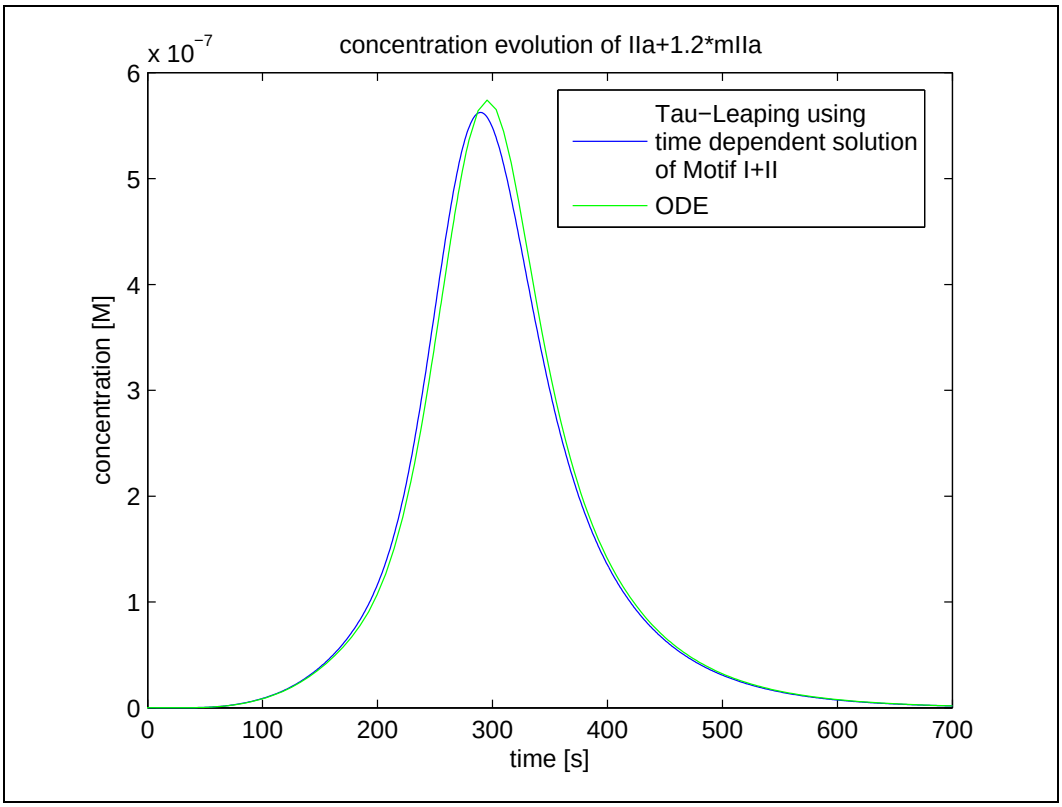

Figure 8: Concentration of thrombin (IIa+1.2×mIIa). Blue curve: Tau-leaping using time dependent solution of Motif I+II. Green curve: ODE.

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.

Acknowledgment

The authors acknowledge the following financial support: National Institute of Biomedical Imaging and Bioengineering (Grant No. 5R01EB007511-03); 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-FG02-04ER25621.

Reference

[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. Engrg. 197 (2008) 3472–3479.

[8] L. A. Segel, M. Slemrod, The quasi-steady-state assumption: a case study in perturbation, SIAM Review 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.

Supplementary Material

Jin Fu, Sheng Wu, Linda R. Petzold

Appendix A. Derivation of the time dependent solution

We use the probability generating function to derive the formula. For a nonnegative discrete random variable X, its probability generating function is defined as

$$G_X(s) = \sum_{i=0}^{\infty} s^i p(X = i),$$

where $p(X = i)$ is the probability that X takes the value of i. The generating function of a Poisson random variable $X \sim \mathcal{P}(\lambda)$ is given by

$$\begin{aligned} G_X(s) &= \sum_{i=0}^{\infty} s^i p(X = i) = \sum_{i=0}^{\infty} s^i \frac{\lambda^i}{i!} e^{-\lambda} = e^{-\lambda} \sum_{i=0}^{\infty} \frac{(s\lambda)^i}{i!} \\ &= e^{-\lambda} e^{s\lambda} = e^{(s-1)\lambda}. \end{aligned} \tag{A.1}$$

The joint generating function of multiple random variables $(X_1, \ldots, X_n)$ is defined as

$$\begin{aligned} &G_{X_1,\ldots,X_n}(s_1, \ldots, s_n) \\ &= \sum_{i_1,\ldots,i_n} s_1^{i_1} \ldots s_n^{i_n} p(X_1 = i_1, \ldots, X_n = i_n). \end{aligned} \tag{A.2}$$

It is convenient to compute the generating function of every variable from their joint generating function. For example, if we want the generating func-

tion of X_j, we can simply plug $s_i = 1,\ i \neq j$ into (A.2). This is because

$$\begin{aligned} & G_{X_1,\ldots,X_n}(1,\ldots,s_j,\ldots,1) \\ &= \sum_{i_1,\ldots,i_n} s_j^{i_j} p(X_1 = i_1,\ldots,X_n = i_n) \\ &= \sum_{i_j} s_j^{i_j} \sum_{i_k,k\neq j} p(X_1 = i_1,\ldots,X_n = i_n) \\ &= \sum_{i_j} s_j^{i_j} p(X_j = i_j) \\ &= G_{X_j}(s_j). \end{aligned} \tag{A.3}$$

A useful property of the joint generating function is given by
Theorem 1: Random variables $(X_1,\ldots,X_n)$ are independent if and only if

$$G_{X_1,\ldots,X_n}(s_1,\ldots,s_n) = G_{X_1}(s_1)\ldots G_{X_n}(s_n).$$

The proof can be found in any probability textbook (see Theorem (29) for two variable case in [1]).

Now let us look at the time dependent population of multiple species. Suppose that we have n species $\hat{S} = \{S_1,\ldots,S_n\}$. As shown in Figure A.1, for each S_i there is an input from outside the system that increases the population of S_i with propensity a_I^i, i.e. a reaction $R_I^i:\ \phi \to S_i$. There is also an output from S_i with rate constant c_O^i, i.e. a reaction $R_O^i:\ S_i \to \phi$. In addition, one S_i molecule can become a S_j molecule due to a reaction $R_{ij}:\ S_i \to S_j$ with rate constant c_{ij}.

Denote by x_i the population of species S_i, r_O^i the number of firings of reaction R_O^i, r_I^i the number of firings of reaction R_I^i, and $p_{i_1,\ldots,i_n,j_1,\ldots,j_n}(t)$ the

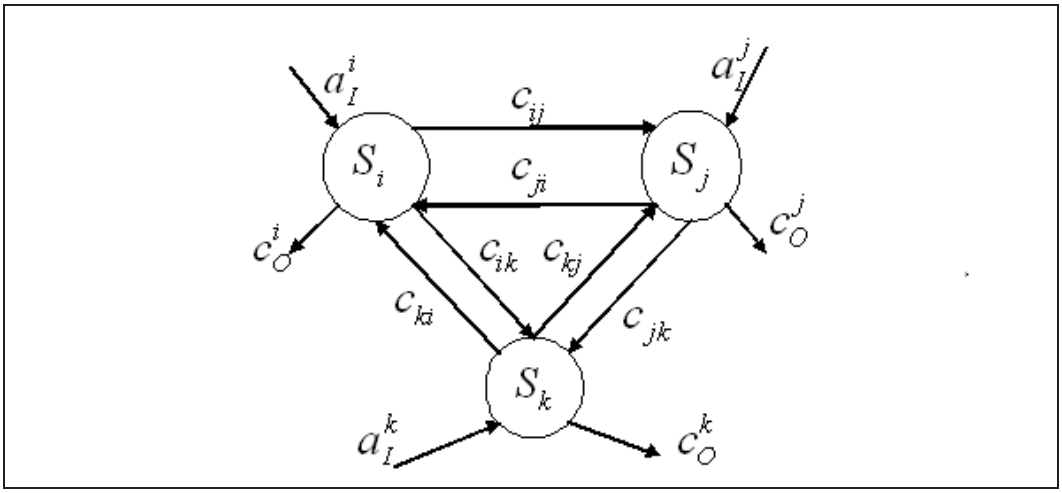

Figure A.1: Example system

probability that $x_1 = i_1, \ldots, x_n = i_n,\ r_O^1 = j_1, \ldots, r_O^n = j_n$. Then the master equation can be written as

$$\begin{aligned}
\frac{dp_{i_1,\ldots,i_n,j_1,\ldots,j_n}(t)}{dt} &= \sum_{k=1}^{n} p_{i_1,\ldots,i_k-1,\ldots,i_n,j_1,\ldots,j_n}(t) a_I^k \\
&+ \sum_{k\neq l} p_{i_1,\ldots,i_k+1,\ldots,i_l-1,\ldots,i_n,j_1,\ldots,j_n}(t) c_{kl}(i_k+1) \\
&+ \sum_{k=1}^{n} p_{i_1,\ldots,i_k+1,\ldots,i_n,j_1,\ldots,j_k-1,\ldots,j_n}(t) c_O^k (i_k+1) \\
&- p_{i_1,\ldots,i_n,j_1,\ldots,j_n}(t) \left(\sum_{k=1}^{n} a_I^k + \sum_{k\neq l} c_{kl} i_k + \sum_{k=1}^{n} c_O^k i_k \right). \qquad \text{(A.4)}
\end{aligned}$$

To simplify the notation we will use p_{i_k+1,j_l-1} to refer to $p_{i_1,\ldots,i_k+1,\ldots,i_n,j_1,\ldots,j_l-1,\ldots,j_n}$.

Multiplying $s_1^{i_1} \ldots s_n^{i_n} u_1^{j_1} \ldots u_n^{j_n}$ on both sides of the master equation (A.4)

gives

$$
\begin{aligned}
&\frac{\partial s_1^{i_1} \dots s_n^{i_n} u_1^{j_1} \dots u_n^{j_n} p_{i_1,\dots,i_n,j_1,\dots,j_n}(t)}{\partial t} \\
&= \sum_{k=1}^{n} s_1^{i_1} \dots s_k^{i_k-1} \dots s_n^{i_n} u_1^{j_1} \dots u_n^{j_n} p_{i_k-1}(t) s_k a_I^k \\
&+ \sum_{k\neq l} c_{kl} s_l \frac{\partial}{\partial s_k} \left(\dots s_k^{i_k+1} \dots s_l^{i_l-1} \dots p_{i_k+1,i_l-1}(t) \right) \\
&+ \sum_{k=1}^{n} c_O^k u_k \frac{\partial}{\partial s_k} \left(\dots s_k^{i_k+1} \dots u_k^{j_k-1} \dots p_{i_k+1,j_k-1}(t) \right) \\
&- s_1^{i_1} \dots s_n^{i_n} u_1^{j_1} \dots u_n^{j_n} p_{i_1,\dots,i_n,j_1,\dots,j_n}(t) \left(\sum_{k=1}^{n} a_I^k \right) \\
&- \sum_{k\neq l} c_{kl} s_k \frac{\partial}{\partial s_k} \left(s_1^{i_1} \dots s_n^{i_n} u_1^{j_1} \dots u_n^{j_n} p_{i_1,\dots,i_n,j_1,\dots,j_n}(t) \right) \\
&- \sum_{k=1}^{n} c_O^k s_k \frac{\partial}{\partial s_k} \left(s_1^{i_1} \dots s_n^{i_n} u_1^{j_1} \dots u_n^{j_n} p_{i_1,\dots,i_n,j_1,\dots,j_n}(t) \right) .
\end{aligned}
$$

Summing both sides over $i_1, \dots, i_n, j_1, \dots, j_n$ and using the definition of gen-

erating function (A.2), we have

$$
\begin{aligned}
&\frac{\partial G\left(s_1,\ldots,s_n,u_1,\ldots,u_n,t\right)}{\partial t}\\
&=\sum_{k=1}^{n} G s_k a_I^k+\sum_{k\neq l} c_{kl}s_l\frac{\partial G}{\partial s_k}+\sum_{k=1}^{n} c_O^k u_k\frac{\partial G}{\partial s_k}\\
&-G\left(\sum_{k=1}^{n} a_I^k\right)-\sum_{k\neq l} c_{kl}s_k\frac{\partial G}{\partial s_k}-\sum_{k=1}^{n} c_O^k s_k\frac{\partial G}{\partial s_k}\\
&=\sum_{k=1}^{n}\left(\sum_{l\neq k} c_{kl}\left(s_l-s_k\right)+c_O^k\left(u_k-s_k\right)\right)\frac{\partial G}{\partial s_k}+G\sum_{k=1}^{n} a_I^k\left(s_k-1\right)\\
&=\left(\frac{\partial G}{\partial \boldsymbol{s}}\right)^T\left(-\boldsymbol{A}\left(\boldsymbol{s}-\mathbf{1}\right)+\operatorname{diag}\left(\boldsymbol{c}_O\right)\left(\boldsymbol{u}-\mathbf{1}\right)\right)+G\boldsymbol{a}_I^T\left(\boldsymbol{s}-\mathbf{1}\right), \qquad \text{(A.5)}
\end{aligned}
$$

where

$$
\boldsymbol{A}=\begin{pmatrix}
\sum\limits_{j\neq 1} c_{1j}+c_O^1 & -c_{12} & \cdots & -c_{1n}\\
-c_{21} & \sum\limits_{j\neq 2} c_{2j}+c_O^2 & \cdots & -c_{2n}\\
 & \vdots & & \\
-c_{n1} & -c_{n2} & \cdots & \sum\limits_{j\neq n} c_{nj}+c_O^n
\end{pmatrix}
$$

and

$$
\begin{aligned}
&(\boldsymbol{s}-\mathbf{1})^T = (s_1-1,\ldots,s_n-1)\\
&(\boldsymbol{u}-\mathbf{1})^T = (u_1-1,\ldots,u_n-1)\\
&\left(\frac{\partial G}{\partial \boldsymbol{s}}\right)^T = \left(\frac{\partial G}{\partial s_1},\ldots,\frac{\partial G}{\partial s_n}\right)\\
&\boldsymbol{a}_I^T = \left(a_I^1,\ldots,a_I^n\right),\ \boldsymbol{c}_O^T = \left(c_O^1,\ldots,c_O^n\right).
\end{aligned}
$$

Here, $\mathrm{diag}\,(\boldsymbol{c}_O)$ is the diagonal matrix with diagonal elements $(c_O^1,\ldots,c_O^n)$.

This is a PDE for $G\,(s_1,\ldots,s_n,u_1,\ldots,u_n,t)$. To determine the solution, we also need an initial condition. Let us begin with the simple case that the system is initially empty, i.e. all of the molecules come from the input channels $R_I^1,\ldots,R_I^n$. Thus at $t=0$ we have $x_1=\cdots=x_n=r_O^1=\cdots=r_O^n=0$. The initial condition is given by

$$
\begin{aligned}
&G\,(s_1,\ldots,s_n,u_1,\ldots,u_n,0)\\
&=\sum_{i_1,\ldots,i_n,j_1,\ldots,j_n} s_1^{i_1}\ldots s_n^{i_n}u_1^{j_1}\ldots u_n^{j_n}p_{i_1,\ldots,i_n,j_1,\ldots,j_n}\,(0)\\
&=s_1^0\ldots s_n^0u_1^0\ldots u_n^0\times 1=1. \qquad\qquad (A.6)
\end{aligned}
$$

The solution for (A.5), (A.6) can be written as

$$
G=e^{\boldsymbol{\lambda}^T(\boldsymbol{s}-\mathbf{1})+\boldsymbol{\lambda}_O^T(\boldsymbol{u}-\mathbf{1})}=\prod_{k=1}^{n}e^{\lambda_k(s_k-1)}\prod_{k=1}^{n}e^{\lambda_{Ok}(u_k-1)}, \qquad (A.7)
$$

where

$$\boldsymbol{\lambda}^T \triangleq (\lambda_1, \ldots, \lambda_n) = \boldsymbol{a}_I^T \left(\int_0^t e^{\boldsymbol{A}x} dx \right) e^{-\boldsymbol{A}t} \tag{A.8}$$

$$\boldsymbol{\lambda}_O^T \triangleq (\lambda_{O1}, \ldots, \lambda_{On}) = \boldsymbol{a}_I^T \left(\int_0^t e^{\boldsymbol{A}x} \int_x^t e^{-\boldsymbol{A}y} dy dx \right) \mathrm{diag}\left(\boldsymbol{c}_O\right). \tag{A.9}$$

In particular, if $\boldsymbol{A}$ is invertible and has n linearly independent eigenvectors $\boldsymbol{v}_1^A, \ldots, \boldsymbol{v}_n^A$, with the corresponding eigenvalues $\lambda_1^A, \ldots, \lambda_n^A$, then (A.8) and (A.9) can be replaced by

$$\boldsymbol{\lambda}^T = \boldsymbol{a}_I^T \boldsymbol{V}_A \mathrm{diag}\left(\frac{1 - e^{-\lambda_i^A t}}{\lambda_i^A} \right) \boldsymbol{V}_A^{-1} \tag{A.10}$$

$$\boldsymbol{\lambda}_O^T = \left(\boldsymbol{a}_I^T t - \boldsymbol{\lambda}^T \right) \boldsymbol{A}^{-1} \mathrm{diag}\left(\boldsymbol{c}_O\right), \tag{A.11}$$

where $\boldsymbol{V}_A = \left(\boldsymbol{v}_1^A, \ldots, \boldsymbol{v}_n^A\right)$ is the matrix composed of the eigenvectors of A. $\mathrm{diag}\left(x_i\right) \triangleq \mathrm{diag}(\boldsymbol{x})$ where $\boldsymbol{x}^T = (x_1, \ldots, x_n)$.

We can easily obtain the generating function of $x_i, i = 1, \ldots, n$ and $r_O^i, i = 1, \ldots, n$ from their joint generating function (A.7) using (A.3):

$$G_{x_i} = G\left(1, \ldots, s_i, \ldots, 1\right) = e^{\lambda_i (s_i - 1)}$$

$$G_{r_O^i} = G\left(1, \ldots, u_i, \ldots, 1\right) = e^{\lambda_{Oi} (u_i - 1)}.$$

Comparing with (A.1), we can see that x_i is a Poisson random variable with parameter λ_i, and r_O^i is a Poisson random variable with parameter λ_{Oi}. According to Theorem 1, (A.7) implies that $x_1, \ldots, x_n$, $r_O^1, \ldots, r_O^n$ are

independent Poisson random variables

$$x_i \sim \mathcal{P}(\lambda_i),\ r_O^i \sim \mathcal{P}(\lambda_{Oi}),\ i = 1, \ldots, n. \tag{A.12}$$

The next problem is to find a way to sample those random variables in the simulation. The inputs $r_I^1, \ldots, r_I^n$ are just independent Poisson random variables with parameters $a_I^1 t, \ldots, a_I^n t$, so they are easy to sample. However when the inputs are sampled, we should not sample x^i and r_O^i directly from $\mathcal{P}(\lambda_i)$ and $\mathcal{P}(\lambda_{Oi})$. For example if we accidentally sampled a very large value for x_i that it is even greater than the sum of all the inputs we sampled, then the result does not make sense. Instead we need to sample $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{r}_O = (r_O^1, \ldots, r_O^n)$ conditioned on $\boldsymbol{r}_I = (r_I^1, \ldots, r_I^n)$. In other words, we need to sample $\boldsymbol{x}$ and $\boldsymbol{r}_O$ using their conditional distribution when $\boldsymbol{r}_I$ is given.

Since the molecules coming from an input channel R_I^i behave independently from molecules coming from other input channels, we can first focus on the molecules from R_I^i and switch off $R_I^j,\ j \neq i$. Now we have only one input channel, and (A.8), (A.9) become (we have added the index i to the notation to indicate that the values are contributed by input channel R_I^i)

$$\left(\boldsymbol{\lambda}^i\right)^T = \left(\lambda_1^i, \ldots, \lambda_n^i\right) = a_I^i \boldsymbol{e}_i^T \left(\int_0^t e^{\boldsymbol{A}x} dx\right) e^{-\boldsymbol{A}t} \tag{A.13}$$

$$\left(\boldsymbol{\lambda}_O^i\right)^T = \left(\lambda_{O1}^i, \ldots, \lambda_{On}^i\right) = a_I^i \boldsymbol{e}_i^T \left(\int_0^t e^{\boldsymbol{A}x} \int_x^t e^{-\boldsymbol{A}y} dy dx\right) \mathrm{diag}\left(\boldsymbol{c}_O\right), \tag{A.14}$$

where $\boldsymbol{e}_i^T$ is the unit vector with the ith element being 1.

Now our purpose is to find the distributions of $\boldsymbol{x}$ and $\boldsymbol{r}_O$ when r_I^i is given.

The following theorem answers this question directly.

Theorem 2: If $X_i \sim \mathcal{P}(\lambda_i)\,(i = 1, \ldots, n)$ are independent Poisson random variables, then

$$X_i \left| \sum_{j=1}^{n} X_j \sim \mathcal{B}\left(\sum_{j=1}^{n} X_j,\ \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \right). \right.$$

Proof. We show the proof for $n = 2$. For $n > 2$, the problem can be converted to the $n = 2$ case using the fact that the sum of independent Poisson random variables is still a Poisson random variable.

As X_1 and X_2 are independent Poisson random variables

$$\begin{aligned} & X_1 + X_2 \sim \mathcal{P}(\lambda_1 + \lambda_2) \\ & \Rightarrow P(X_1 + X_2 = n) = \frac{(\lambda_1 + \lambda_2)^n}{n!} e^{-(\lambda_1 + \lambda_2)} \end{aligned}$$

$$\begin{aligned} & P(X_1 = i \,|X_1 + X_2 = n\,) = \frac{P(X_1 = i)\, P(X_2 = n - i)}{P(X_1 + X_2 = n)} \\ & = \frac{\lambda_1^i}{i!} e^{-\lambda_1} \frac{\lambda_2^{n-i}}{(n-i)!} e^{-\lambda_2} \Bigg/ \left(\frac{(\lambda_1 + \lambda_2)^n}{n!} e^{-(\lambda_1 + \lambda_2)} \right) \\ & = \frac{n!}{i!\,(n-i)!} \left(\frac{\lambda_1}{\lambda_1 + \lambda_2} \right)^i \left(\frac{\lambda_2}{\lambda_1 + \lambda_2} \right)^{n-i} \\ & = P(Y = i)\,, \end{aligned}$$

where

$$Y \sim \mathcal{B}\left(n,\ \frac{\lambda_1}{\lambda_1 + \lambda_2} \right). \qquad \square$$

According to this theorem, the conditional distribution of $\boldsymbol{x}\,|r_I^i$ and $\boldsymbol{r}_O\,|r_I^i$

is actually a multinomial distribution:

$$\begin{aligned}&\left(x_1,\ldots,x_n,r_O^1,\ldots,r_O^n\right)|r_I^i\\&\sim\mathcal{M}\left(r_I^i,\frac{\lambda_1^i}{a_I^it},\ldots,\frac{\lambda_n^i}{a_I^it},\frac{\lambda_{O1}^i}{a_I^it},\ldots,\frac{\lambda_{On}^i}{a_I^it}\right).\end{aligned}$$

Here

$$a_I^it=\sum_{i=1}^n\lambda_i^i+\sum_{i=1}^n\lambda_{Oi}^i$$

because the sum of all of the Poisson random variables should be equal to the total input. This can also be verified in the following way. From (A.8) and (A.9), we have

$$\begin{aligned}&\frac{d}{dt}\left(\boldsymbol{\lambda}^T\mathbf{1}+\boldsymbol{\lambda}_O^T\mathbf{1}\right)\\&=\boldsymbol{a}_I^T\left(\boldsymbol{I}-\left(\int_0^t e^{\boldsymbol{A}x}dx\right)e^{-\boldsymbol{A}t}\boldsymbol{A}\right)\mathbf{1}+\boldsymbol{a}_I^T\left(\int_0^t e^{\boldsymbol{A}x}e^{-\boldsymbol{A}t}dx\right)\mathrm{diag}\left(\boldsymbol{c}_O\right)\mathbf{1}\\&=\boldsymbol{a}_I^T\left(\mathbf{1}-\left(\int_0^t e^{\boldsymbol{A}x}dx\right)e^{-\boldsymbol{A}t}\boldsymbol{c}_O\right)+\boldsymbol{a}_I^T\left(\int_0^t e^{\boldsymbol{A}x}dx\right)e^{-\boldsymbol{A}t}\boldsymbol{c}_O=\boldsymbol{a}_I^T\mathbf{1}.\end{aligned}$$

Together with the initial condition $\boldsymbol{\lambda}(0)=\mathbf{0},\ \boldsymbol{\lambda}_O(0)=\mathbf{0}$, this yields

$$\boldsymbol{\lambda}^T\mathbf{1}+\boldsymbol{\lambda}_O^T\mathbf{1}=\left(\boldsymbol{a}_I^T\mathbf{1}\right)t$$

Now we can extend the result by switching on the other input channels. Since the molecules from different input channels do not interrupt each other, the result in this situation should be the sum all the multinomial random

variables produced by each input channel,

$$\begin{aligned}&\left(x_1,\ldots,x_n,r_O^1,\ldots,r_O^n\right)|\boldsymbol{r}_I\\&\sim\sum_{i=1}^{n}\mathcal{M}\left(r_I^i,\frac{\lambda_1^i}{a_I^it},\ldots,\frac{\lambda_n^i}{a_I^it},\frac{\lambda_{O1}^i}{a_I^it},\ldots,\frac{\lambda_{On}^i}{a_I^it}\right).\end{aligned}\tag{A.15}$$

Now let us remove the assumption that the system is initially empty. We also start from a simple case, assuming at time $t=0$ that we have $x_i(0)\neq 0$, $x_j(0)=0\ \ (j\neq i)$. Since these molecules have nothing to do with those coming from the input channels, we can switch off all the input channels and just look at the behavior of these molecules. Consider one such molecule. At any time $t>0$, there is a probability $p_j^i(t)$ that the molecule stays at the state S_j. There is also a probability $p_{Oj}^i(t)$ that the molecule has already left the system through channel R_O^j. More importantly, these probabilities should be the same for every molecule that initially stays in S_i. Thus $(x_1(t),\ldots,x_n(t),r_1(t),\ldots,r_n(t))$ should have a multinomial distribution. To determine the parameters for this distribution, we need to compute $\boldsymbol{p}^i(t)\triangleq(p_1^i(t),\ldots,p_n^i(t))$ and $\boldsymbol{p}_O^i(t)\triangleq(p_{O1}^i(t),\ldots,p_{On}^i(t))$.

The master equation for a single molecule is given by

$$\frac{dp_j^i(t)}{dt}=\sum_{k\neq j}p_k^i(t)\,c_{kj}-p_j^i(t)\left(\sum_{k\neq j}c_{jk}+c_{Oj}\right)\tag{A.16}$$

$$\frac{dp_{Oj}^i(t)}{dt}=p_j^i(t)\,c_{Oj}\tag{A.17}$$

$$j=1,\ldots,n,$$

with initial condition

$$p_i^i(0) = 1,\ p_j^i(0) = 0, j \neq i \tag{A.18}$$

$$p_{Oj}^i(0) = 0, j = 1 \ldots, n. \tag{A.19}$$

The solution to (A.16) and (A.18) is given by

$$\boldsymbol{p}^i(t) = e^{\boldsymbol{B}t}\boldsymbol{e}_i, \tag{A.20}$$

and from (A.17) and (A.19) we have

$$\boldsymbol{p}_O^i(t) = \int_0^t \operatorname{diag}(\boldsymbol{c}_O)\, e^{\boldsymbol{B}x}\boldsymbol{e}_i dx \tag{A.21}$$

where

$$\boldsymbol{B} = -\boldsymbol{A}^T.$$

If $\boldsymbol{B}$ has n linearly independent eigenvectors $\boldsymbol{v}_1^B, \ldots, \boldsymbol{v}_n^B$, with the corresponding eigenvalues $\lambda_1^B, \ldots, \lambda_n^B$, then (A.20) and (A.21) can be replaced by

$$\boldsymbol{p}^i(t) = \boldsymbol{V}_B \operatorname{diag}\left(e^{\lambda_j^B t}\right) \boldsymbol{V}_B^{-1}\boldsymbol{e}_i \tag{A.22}$$

$$\boldsymbol{p}_O^i(t) = \operatorname{diag}(\boldsymbol{c}_O)\, \boldsymbol{V}_B \operatorname{diag}\left(\frac{e^{\lambda_j^B t} - 1}{\lambda_j^B}\right) \boldsymbol{V}_B^{-1}\boldsymbol{e}_i, \tag{A.23}$$

where $\boldsymbol{V}_B = \left(\boldsymbol{v}_1^B, \ldots, \boldsymbol{v}_n^B\right)$ is the matrix composed of the independent eigenvectors of B.

Putting all the molecules together, the distribution of $\boldsymbol{x}(t)$ and $\boldsymbol{r}_O(t)$

should be a multinomial distribution

$$(\boldsymbol{x}(t),\ \boldsymbol{r}_O(t)) \sim \mathcal{M}\left(x_i(0), \boldsymbol{p}^i(t), \boldsymbol{p}_O^i(t)\right).$$

Now we can let every species have a nonzero initial population. Since they do not influence each other, the result in this case should be the sum of all the multinomial random variables

$$(\boldsymbol{x}(t), \boldsymbol{r}_O(t)) \sim \sum_{i=1}^{n} \mathcal{M}\left(x_i(0), \boldsymbol{p}^i(t), \boldsymbol{p}_O^i(t)\right). \tag{A.24}$$

Having obtained the solution for the initial molecules, it is time to put everything together by switching on the input channels. The result in this case is the sum of (A.15) and (A.24)

$$\begin{aligned}(\boldsymbol{x}(t), \boldsymbol{r}_O(t)) \sim & \sum_{i=1}^{n} \mathcal{M}\left(x_i(0),\ \boldsymbol{p}^i(t),\ \boldsymbol{p}_O^i(t)\right) \\ & + \sum_{i=1}^{n} \mathcal{M}\left(r_I^i,\ \frac{1}{a_I^i t}\boldsymbol{\lambda}^i,\ \frac{1}{a_I^i t}\boldsymbol{\lambda}_O^i\right).\end{aligned} \tag{A.25}$$

This is the time dependent solution for $\boldsymbol{x}(t)$ and $\boldsymbol{r}_O(t)$

For the simulations in Section III in the main paper, the mean and variance of $\boldsymbol{x}(t)$ have also been used. It would be convenient to have formulas for these values. It seems that we can compute them from (A.25), however (A.25) is the formula when $\boldsymbol{r}_I$ has already been sampled. If we need the mean and variance before $\boldsymbol{r}_I$ has been sampled, we must replace (A.15) by

the Poisson random variables (A.12), yielding

$$\mathbb{E}\left(x_i\left(t\right)\right) = \sum_{j=1}^{n} x_j\left(0\right) p_i^j\left(t\right) + \lambda_i \tag{A.26}$$

$$\mathrm{Var}\left(x_i\left(t\right)\right) = \sum_{j=1}^{n} x_j\left(0\right) p_i^j\left(t\right)\left(1 - p_i^j\left(t\right)\right) + \lambda_i. \tag{A.27}$$

In section III we also need to use the solutions for $n = 1$ and $n = 2$. The solutions for these two cases are given below.

$n = 1$: In this case, $A = -B = c_O$, and $\lambda^A = -\lambda^B = c_O$. Equations (A.10) and (A.11) give

$$\lambda = \frac{a_I}{c_O}\left(1 - e^{-c_O t}\right),\ \lambda_O = a_I t - \lambda,$$

and (A.22) and (A.23) yield

$$p\left(t\right) = e^{-c_O t},\ p_O\left(t\right) = 1 - e^{-c_O t}.$$

Thus the time dependent solution of $x\left(t\right)$ and $r_O\left(t\right)$ given by (A.25) is

$$\begin{aligned}\left(x\left(t\right),\ r_O\left(t\right)\right) &\sim \mathcal{M}\left(x\left(0\right),\ e^{-c_O t},\ 1 - e^{-c_O t}\right) \\ &+ \mathcal{M}\left(r_I,\ \frac{1 - e^{-c_O t}}{c_O t},\ 1 - \frac{1 - e^{-c_O t}}{c_O t}\right).\end{aligned}$$

$n = 2$: Assume the two species are E (enzyme) and ES (enzyme-substrate compound) as shown in Figure A.2. The population of S (substrate) is very large ($x_S(0) \gg x_E(0),\ x_{ES}(0)$). The reactions in the system are

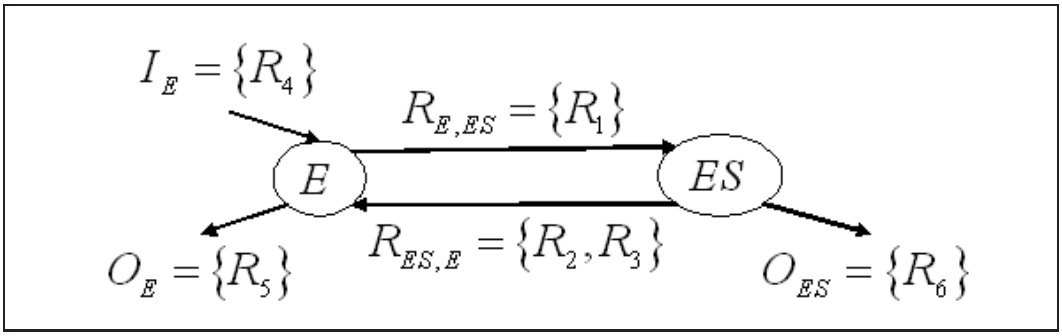

Figure A.2: R_4 is the input reaction for E, and R_5 and R_6 are the output reactions for E and ES respectively. R_1 converts E to ES, R_2 and R_3 convert ES to E.

$$R_1 : \ E + S \xrightarrow{c_1} ES, \quad R_2 : \ ES \xrightarrow{c_2} E + S, \quad R_3 : \ ES \xrightarrow{c_3} E + P$$
$$R_4 : \ \phi \xrightarrow{a_4} E, \quad R_5 : \ E \xrightarrow{c_5} \phi, \quad R_6 : \ ES \xrightarrow{c_6} \phi.$$

During a stepsize of S, equation (A.25) in this case has the form

$$\begin{aligned}
&\left(x_E(t),\ x_{ES}(t),\ r_O^E(t),\ r_O^{ES}(t)\right) \\
&\sim \mathcal{M}\left(x_E\left(0\right),\ p_1^E(t),\ p_2^E(t),\ p_{O1}^E(t),\ p_{O2}^E(t)\right) \\
&+ \mathcal{M}\left(x_{ES}\left(0\right),\ p_1^{ES}(t),\ p_2^{ES}(t),\ p_{O1}^{ES}(t),\ p_{O2}^{ES}(t)\right) \\
&+ \mathcal{M}\left(r_I^E, \frac{\lambda_1(t)}{a_I^E t},\ \frac{\lambda_2(t)}{a_I^E t}\ \frac{\lambda_{O1}(t)}{a_I^E t}\ \frac{\lambda_{O2}(t)}{a_I^E t}\right),
\end{aligned} \tag{A.28}$$

where

$$
\begin{aligned}
\begin{pmatrix}\lambda_1 & \lambda_2\end{pmatrix} &= \begin{pmatrix}a_I^E & a_I^{ES}\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_+^A & \boldsymbol{v}_-^A\end{pmatrix} \\
&\qquad \operatorname{diag}\left(\frac{1-e^{-\lambda_+^A t}}{\lambda_+^A},\ \frac{1-e^{-\lambda_-^A t}}{\lambda_-^A}\right)\begin{pmatrix}\boldsymbol{v}_+^A & \boldsymbol{v}_-^A\end{pmatrix}^{-1} \\
\begin{pmatrix}\lambda_{O1} & \lambda_{O2}\end{pmatrix} &= \left(\begin{pmatrix}a_I^E & a_I^{ES}\end{pmatrix} t - \begin{pmatrix}\lambda_1 & \lambda_2\end{pmatrix}\right)\boldsymbol{A}^{-1} \\
&\qquad \operatorname{diag}\begin{pmatrix}c_O^E & c_O^{ES}\end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\begin{pmatrix}p_1^E \\ p_2^E\end{pmatrix} &= \begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}\operatorname{diag}\left(e^{\lambda_+^B t},\ e^{\lambda_-^B t}\right)\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}^{-1}\begin{pmatrix}1 \\ 0\end{pmatrix} \\
\begin{pmatrix}p_{O1}^E \\ p_{O2}^E\end{pmatrix} &= \operatorname{diag}\begin{pmatrix}c_O^E,\ c_O^{ES}\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix} \\
&\qquad \operatorname{diag}\left(\frac{e^{\lambda_+^B t}-1}{\lambda_+^B},\ \frac{e^{\lambda_-^B t}-1}{\lambda_-^B}\right)\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}^{-1}\begin{pmatrix}1 \\ 0\end{pmatrix} \\
\begin{pmatrix}p_1^{ES} \\ p_2^{ES}\end{pmatrix} &= \begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}\operatorname{diag}\left(e^{\lambda_+^B t},\ e^{\lambda_-^B t}\right)\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}^{-1}\begin{pmatrix}0 \\ 1\end{pmatrix} \\
\begin{pmatrix}p_{O1}^{ES} \\ p_{O2}^{ES}\end{pmatrix} &= \operatorname{diag}\begin{pmatrix}c_O^E,\ c_O^{ES}\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix} \\
&\qquad \operatorname{diag}\left(\frac{e^{\lambda_+^B t}-1}{\lambda_+^B},\ \frac{e^{\lambda_-^B t}-1}{\lambda_-^B}\right)\begin{pmatrix}\boldsymbol{v}_+^B & \boldsymbol{v}_-^B\end{pmatrix}^{-1}\begin{pmatrix}0 \\ 1\end{pmatrix}.
\end{aligned}
$$

Here,

$$
\begin{aligned}
&a_I^E = a_4, \quad a_I^{ES} = 0, \quad c_O^E = c_5, \quad c_O^{ES} = c_6 \\
&c_{E,ES} = c_1 x_S(0), \quad c_{ES,E} = c_2 + c_3 \\
&\boldsymbol{A} = -\boldsymbol{B}^T = \begin{pmatrix} c_{E,ES} + c_O^E & -c_{E,ES} \\ -c_{ES,E} & c_{ES,S} + c_O^{ES} \end{pmatrix},
\end{aligned} \tag{A.29}
$$

where λ_+^A, λ_-^A, $\boldsymbol{v}_+^A$, $\boldsymbol{v}_-^A$ are the eigenvalues and corresponding eigenvectors of $\boldsymbol{A}$, and λ_+^B, λ_-^B, $\boldsymbol{v}_+^B$, $\boldsymbol{v}_-^B$ are the eigenvalues and corresponding eigenvectors of $\boldsymbol{B}$.

Appendix B. The mean and variance of $\boldsymbol{Y = \mathcal{P}(X)}$

Suppose that we sample two random variables X and Y. Y depends on X in such a way that after we have sampled the value x of X, we will sample Y as a Poisson random variable with parameter x, i.e. $Y = \mathcal{P}(x)$. The purpose of this section is to compute the mean and variance of Y, and show that if we approximate Y by $\mathcal{P}(\mathbb{E}X)$, the approximation will give us the correct mean value but a smaller variance than the true $\mathrm{Var}(Y)$.

Let us begin with the expectation of Y. Using the conditional expectation, we have

$$\mathbb{E}Y = \mathbb{E}(\mathbb{E}(Y|X)).$$

When X is given, Y is a Poisson random number with parameter X, so the conditional expectation $\mathbb{E}(Y|X)$ is actually the expectation of a Poisson

random variable with the given parameter X. Thus,

$$\mathbb{E}\left(Y\,|X\right) = X$$

and

$$\mathbb{E}Y = \mathbb{E}\left(\mathbb{E}\left(Y\,|X\right)\right) = \mathbb{E}X. \tag{B.1}$$

For the variance of Y, we have

$$\mathrm{Var}\left(Y\right) = \mathbb{E}\left(Y^2\right) - \left(\mathbb{E}Y\right)^2. \tag{B.2}$$

For $\mathbb{E}\left(Y^2\right)$ we also use the conditional expectation

$$\mathbb{E}\left(Y^2\right) = \mathbb{E}\left(\mathbb{E}\left(Y^2\,|X\right)\right). \tag{B.3}$$

Here

$$\mathbb{E}\left(Y^2\,|X\right) = \mathrm{Var}\left(Y\,|X\right) + \left(\mathbb{E}\left(Y\,|X\right)\right)^2 = X + X^2. \tag{B.4}$$

The last step in the previous equation uses the fact that when X is given, Y is a Poisson random variable with parameter X so both the mean and the variance of Y are equal to X. Inserting (B.4) in (B.3) yields

$$\mathbb{E}\left(Y^2\right) = \mathbb{E}\left(\mathbb{E}\left(Y^2\,|X\right)\right) = \mathbb{E}\left(X + X^2\right) = \mathbb{E}X + \mathbb{E}\left(X^2\right).$$

Inserting this into (B.2) and using (B.1), we obtain the variance of Y,

$$\begin{aligned}\mathrm{Var}\left(Y\right) &= \mathbb{E}X + \mathbb{E}\left(X^2\right) - \left(\mathbb{E}Y\right)^2 \\ &= \mathbb{E}X + \mathbb{E}\left(X^2\right) - \left(\mathbb{E}X\right)^2 \\ &= \mathbb{E}X + \mathrm{Var}\left(X\right). \end{aligned} \tag{B.5}$$

Now we can compare this with the approximation $Y' = \mathcal{P}\left(\mathbb{E}X\right)$. As $\mathbb{E}X$ is a real number, Y' is actually a Poisson random variable with

$$\mathbb{E}\left(Y'\right) = \mathrm{Var}\left(Y'\right) = \mathbb{E}X.$$

Comparing this with (B.1) and (B.5), we can see that the approximation has the same mean value but a smaller variance.

Appendix C. The mean and variance of the number of firings in a reaction channel

Consider the Example System from Appendix Appendix A. For any species in $\hat{S}$, we know its time dependent solution. Thus there is no stepsize requirement associated with this species, as long as the species not belonging to $\hat{S}$ can be considered as constant. In another words, we need only to compute the stepsize for species not in $\hat{S}$.

We use the following inequalities to bound the change of a species.

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

where g_i is a constant that depends on the highest order of the reactions

which involve S_i as a reactant. In the current situation r_i may no longer be a Poisson random variable. The purpose of this section is to find the mean and variance for such reactions.

For the system in Appendix Appendix A, we can partition the reactions into three groups:

1. Reactions whose reactants do not belong to $\hat{S}$ (e.g. all the input channels). As the reactants for these reactions can be considered constant during the step, these reactions can be sampled by Poisson random variables as in tau leaping.
2. Reactions corresponding to output channels. In the Example System of Appendix Appendix A, the output reactions are $R_O^i,\ i = 1, \ldots, n$, however, generally speaking a species $S_i \in \hat{S}$ could have several output reactions, i.e. R_O^i in not just one reaction but a set of reactions. These reactions should compete with each other for a share of r_O^i. Now the rate constant c_O^i for R_O^i is the sum of all the rate constants for reactions in R_O^i. Supposing that $R_k : S_i \to \phi$ is in R_O^i with reaction rate c_k. Then the probability that R_k is responsible for a firing of R_O^i is c_k/c_O^i.
   Now let us compute the mean and variance of r_k. In the Example System of Appendix Appendix A, there are r_O^i molecules consumed by R_O^i. These molecules come from either the input channels (denoted by r_P^i) or the initial molecules of species in $\hat{S}$ (denoted by r_B^i). Thus

$$r_O^i = r_P^i + r_B^i.$$

   It is shown in Appendix Appendix A that r_P^i is a Poisson random

number with parameter λ_{Oi} (see (A.12)),

$$r_P^i \sim \mathcal{P}\left(\lambda_{Oi}\right).$$

r_B^i is the sum of n binomial random variables with parameters $\left(x_j\left(0\right),\ p_{Oi}^j\right)$, $j = 1, \ldots, n.$ (see (A.24)),

$$r_B^i \sim \sum_{j=1}^{n} \mathcal{B}\left(x_j\left(0\right),\ p_{Oi}^j\right).$$

We want to distribute these molecules to the output channels in R_O^i. The probability that a molecule goes through reaction channel R_k is c_k/c_O^i. To distribute the first part, we make use of the following theorem.

Theorem 3. Let N be a Poisson random number with parameter λ. Then the sum of N i.i.d Bernoulli variables with parameter p is also a Poisson random variable with parameter λp.

The proof can be found in a probability textbook (see example (27) in [1]).

In our case, $r_P^i \sim \mathcal{P}\left(\lambda_{Oi}\right)$, and each molecule in r_P^i has a probability c_k/c_O^i to go through channel R_k. By Theorem 3, the number of molecules that choose R_k is a Poisson random number

$$\mathcal{P}\left(\frac{c_k}{c_O^i}\lambda_{Oi}\right).$$

Now let us distribute the second part r_B^i. r_B^i is the sum of n independent binomial random numbers. Each molecule in r_B^i also has a probability

c_k/c_O^i to choose channel R_k, so in this case the number of molecules R_k consumed is also the sum of n binomial random variables

$$\sum_{j=1}^{n} \mathcal{B}\left(x_j\left(0\right),\ \frac{c_k}{c_O^i} p_{Oi}^j\right).$$

Adding the two parts together, we obtain

$$r_k \sim \mathcal{P}\left(\frac{c_k}{c_O^i}\lambda_{Oi}\right) + \sum_{j=1}^{n} \mathcal{B}\left(x_j\left(0\right),\ \frac{c_k}{c_O^i} p_{Oi}^j\right).$$

The mean and variance of r_k can be calculated by

$$\begin{aligned}
\mathbb{E}r_k &= \frac{c_k}{c_O^i}\left(\lambda_{Oi} + \sum_{j=1}^{n} x_j\left(0\right) p_{Oi}^j\right) \\
\mathrm{Var}\left(r_k\right) &= \frac{c_k}{c_O^i}\left(\lambda_{Oi} + \sum_{j=1}^{n} x_j\left(0\right) p_{Oi}^j\left(1 - \frac{c_k}{c_O^i} p_{Oi}^j\right)\right).
\end{aligned}$$

3. Reactions which convert one species in $\hat{S}$ to another species in $\hat{S}$. In Appendix Appendix A, the $R_{ij},\ i,j = 1,\ldots,n$ are of this type. In a more general case, R_{ij} can contain several reactions as well. Suppose that $R_k : S_i \to S_j$ is one of them, with rate constant c_k.

   Now we want to compute the mean and variance of r_k. Since we use species S_i as the reactant and its population is a random variable during the step, we may not have an exact formula for r_k. Here we use the following approximation,

$$r_k \approx \mathcal{P}\left(c_k\left(\int_0^\tau \mathbb{E}x_i\left(t\right)dt + \frac{\tau}{2}\left(x_i\left(\tau\right) - \mathbb{E}\left(x_i\left(\tau\right)\right)\right)\right)\right). \tag{C.1}$$

The mean and variance of r_k can be computed using (B.1) and (B.5) as follows:

$$\begin{aligned}
&\mathbb{E}r_k \approx \mathbb{E}\left(c_k\left(\int_0^\tau \mathbb{E}x_i\left(t\right)dt + \frac{\tau}{2}\left(x_i\left(\tau\right) - \mathbb{E}\left(x_i\left(\tau\right)\right)\right)\right)\right) \\
&= c_k\int_0^\tau \mathbb{E}\left(x_k\left(t\right)\right)dt \\
&\mathrm{Var}\left(r_k\right) \approx c_k\int_0^\tau \mathbb{E}\left(x_i\left(t\right)\right)dt \\
&+ \mathrm{Var}\left(c_k\left(\int_0^\tau \mathbb{E}x_i\left(t\right)dt + \frac{\tau}{2}\left(x_i\left(\tau\right) - \mathbb{E}\left(x_i\left(\tau\right)\right)\right)\right)\right) \\
&= c_k\int_0^\tau \mathbb{E}\left(x_i\left(t\right)\right)dt + \frac{\tau^2}{4}\mathrm{Var}\left(x_i\left(\tau\right)\right).
\end{aligned}$$

Here the formulas for $\mathbb{E}\left(x_i\left(t\right)\right)$ and $\mathrm{Var}\left(x_i\left(t\right)\right)$ are given by (A.26) and (A.27).

Appendix D. Sampling a feasible flow in the network

Consider each species in $\hat{S}$ as a vertex. Vertices i and j are connected if there are reactions which convert species S_i to S_j or S_j to S_i. On each edge we define the flow

$$f_{ij} = r_{ij} - r_{ji}, \tag{D.1}$$

where f_{ij} indicates the number of molecules that go from S_i to S_j. If its value is negative, there are more firings of R_{ji} than R_{ij}.

Using the result in Appendix Appendix A, we can sample all the input reactions R_I^i, all the output reactions R_O^i and the population vector $\boldsymbol{x}$. However sometimes we also need to sample the reactions R_{ij}. If we do this, we should make sure that we only sample the flow for a proper set of edges.

Here 'a proper set' means that after sampling the flow values for this set, the flow values of other edges can be uniquely determined by mass conservation equations.

For each vertex i, the mass conservation equation is given by,

$$r_I^i + x_i(0) = x_i(t) + r_O^i + \sum_{j \neq i} f_{ij}. \tag{D.2}$$

Consider a connected subgraph $G = (V,\ E)$, where V is the set of vertices in G and E is the set of edges in G. Each vertex provides a mass conservation equation and each edge provides an unknown. If the subgraph contains no loop, then the number of vertices is one more than the number of edges, which means that the number of equations is one more than the number of unknowns. However, these equations are not independent. Summing (D.2) up over all vertices in V, we obtain

$$\sum_{i \in V} \left(r_I^i + x_i(0) \right) = \sum_{i \in V} \left(x_i(t) + r_O^i \right).$$

Here the flows completely cancel out. This equation simply shows the total mass conservation of the system and it is automatically satisfied by (A.25). Thus the number of independent equations is one less than the total number of vertices in V. For the connected subgraph G we have the same number of equations and unknowns, thus the flow can be determined.

After obtaining a flow value f_{ij} from the mass conservation equation, we can go on sampling r_{ij} and r_{ij} in the following manner such that (D.1) is satisfied:

If $f_{ij} \geq 0$, sample r_{ji} using (C.1) and compute r_{ij} as $r_{ij} = r_{ji} + f_{ij}$. If $f_{ij} < 0$, sample r_{ij} using (C.1) and compute r_{ji} as $r_{ji} = r_{ij} - f_{ij}$.

If G has loops, the number of unknowns will be more than the number of equations. In this case, we need to sample the flow value (by sampling r_{ij} and r_{ji} using (C.1) and computing f_{ij} using (D.1)) of some edges to decrease the number of unknown. The following is a simple algorithm to determine the edges we are going to sample.

1. Create an empty list L. Arbitrarily pick a start vertex i in V and push it into L. Create a pointer and let it point to the first element in the list, which at the beginning is i.
2. Push all the vertices connected to i into the list.
3. Move the pointer to the next element in the list (suppose the second element is j).
4. Collect all the vertices connected to j except the one that caused j to have been pushed into the list, i.e. the vertex i. Denote these vertices by V_j.
5. Compare every vertex in V_j with the elements in the list. If a vertex $k \in V_j$ is not in the list, push it into the list. If it is already in the list, this implies that there is a loop in the system. This is because we already have a path from i to k and now we have found another one. It is obvious that edge e_{jk} is in the loop, so we sample the value of f_{jk} and cut the edge e_{jk}. Now we have removed the loop. Continue comparing other vertices until all the vertices in V_j are treated as we do for vertex k.
6. Move the pointer to the next element in the list and do the same as we

did for vertex j. Stop the process when the pointer has walked through the whole list.

After applying the above algorithm to the graph G, the unsampled edges contain no loops. Thus we have the same number of independent equations and unknowns, and the flow in the graph can be uniquely determined.

[1] G. R. Grimmett, D. R. Stirzaker, Probabiliy and Random Processes, Oxford University Press Inc., New York, third edition, p. 154.