Commun. Comput. Phys.
doi: 10.4208/cicp.190411.301111a

Vol. **12**, No. 4, pp. 981-1005
October 2012

Boosted Hybrid Method for Solving Chemical Reaction Systems with Multiple Scales in Time and Population Size

Yucheng Hu[1,*], Assyr Abdulle[2] and Tiejun Li[1]

[1] *Laboratory of Mathematics and Applied Mathematics and School of Mathematical Sciences, Peking University, Beijing 100871, China.*
[2] *Mathematics Section, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.*

Received 19 April 2011; Accepted (in revised version) 30 November 2011

Communicated by Pingwen Zhang

Available online 28 March 2012

Abstract. A new algorithm, called boosted hybrid method, is proposed for the simulation of chemical reaction systems with scale-separation in time and disparity in species population. For such stiff systems, the algorithm can automatically identify scale-separation in time and slow down the fast reactions while maintaining a good approximation to the original effective dynamics. This technique is called boosting. As disparity in species population may still exist in the boosted system, we propose a hybrid strategy based on coarse-graining methods, such as the tau-leaping method, to accelerate the reactions among large population species. The combination of the boosting strategy and the hybrid method allow for an efficient and adaptive simulation of complex chemical reactions. The new method does not need *a priori* knowledge of the system and can also be used for systems with hierarchical multiple time scales. Numerical experiments illustrate the versatility and efficiency of the method.

AMS subject classifications: 65C05, 65C20

Key words: Chemical reaction, multiscale, boosting, hybrid method.

1 Introduction

Advances in experimental and computational methods over the last decades have made a quantitative, systematic understanding of cellular processes in molecular level possible [1–6]. For micro-scale biochemical systems, such as one single living cell, consider-

*Corresponding author. *Email addresses:* huyc@pku.edu.cn (Y. Hu), assyr.abdulle@epfl.ch (A. Abdulle), tieli@pku.edu.cn (T. Li)

able evidence indicates that stochasticity plays an important role, especially when low-molecular-number reactant species are being considered [2,3]. The usefulness of the traditional deterministic approach, based on reaction rate equations, is limited in such situation. In turn, many stochastic biochemical reaction networks have been built to take into account the randomness in biological processes. Because of the disparity of time scales and species population, the simulation of such systems is often challenging and the development of new numerical techniques for multiscale chemical reactions has become an active research field [4,7,8].

One fundamental method in simulating chemical reaction systems is Gillespie's Stochastic Simulating Algorithm (SSA) [9,10]. It can generate statistically exact trajectories of the system state by randomly sampling each reaction event. In principle, SSA applies to any chemical reaction system, but the method become computationally costly when reaction events occur very frequently in the system. This often happens because of the co-existence of fast and slow dynamics in a system, or reactions involving species with very large populations, or both.

On one hand, the co-existence of fast and slow dynamics in a system leads often to severe step-size restriction for standard methods. Such systems are called stiff and need a special numerical treatments. For stiff ordinary or stochastic differential equations, implicit methods or stabilized explicit methods (called Chebyshev methods) can be efficient [11–13]. But fast variables in chemical reaction system often fluctuate quickly around a "slow manifold", and implicit or stabilized method usually fail to capture the right stationary distribution of the fast variables [12, 14, 15]. Rao and Arkin [16] first formalized the quasi-equilibrium approximation in chemical reaction system and implemented it in SSA. Their idea was further extended by Cao et al. in developing the slow-scale SSA [17]. Both methods require explicit form of the stationary distribution for the fast variables which in general is difficult to get. To remove this restriction, E et al. developed the nested-SSA [18]. In this method, the averaged rates of the slow reactions are sampled by inner SSA, as the micro-solver acting on fast reactions only, during a period of time that is much larger than the fast time scale and at the same time much smaller than the slow time scale. Then the average rates of the slow reactions will be used by the outer SSA, as the macro-solver acting on slow reactions only, to march the system forward.

On the other hand, reactions involving species with large population size fire very frequently which also make the SSA computationally inefficient. To overcome this difficulty, Haseltine and Rawlings proposed a hybrid method for solving chemical reaction systems with disparity in species population [8]. The main idea is to apply a coarse-graining approximation (based on stochastic or ordinary differential equations) for species with large population size and SSA for species with small population size. Many variants of hybrid method have been proposed [6, 19–25]. Based on the system size, the τ-leaping method [7], chemical Langevin equations or reaction rate equations [26] are often used as the coarse solver.

Numerical algorithm that can handle both the multiple time scales and the disparity

in species population is currently a challenging research direction. Cao et al. [27] demonstrate that hybrid method can be combined with slow-scale SSA and implicit τ-leaping to handle stiff separations. Samant et al. [28] designed a more general algorithm that can identify scale-separation and partition the reactions on-the-fly. Another improvement in this work is the generalization of the nested-SSA to nest-tau-leaping to handle fast reactions involving large population species.

Compared with the nested-SSA by taking the heterogeneous multi-scale method (HMM) approach [18] and slow-scale SSA based on taking the explicit quasi-equilibrium approximation [17], another nice idea for multi-scale methods which is called *boosting* has never been investigated in the chemical reaction literature [29]. The idea of boosting for accelerating the multi-scale simulation is simply to rescale the scale separation parameter $\lambda \ll 1$ to an artificial constant λ_0 such that $\lambda \ll \lambda_0 \ll 1$, which can speed up the simulation of fast reactions and keep the computational accuracy at the same time. The idea can be dated to Carr-Parrinello's molecular dynamics and Chorin's artificial compressibility method in fluid mechanics [29].

In this paper we present a new general algorithm, called boosted hybrid method, that can efficiently simulate complex chemical reaction systems with time-scale separation and disparity in species population. The algorithm combines the idea of boosting and hybrid simulations. It collects the system state information on-the-fly and automatically partition the reactions into fast and slow groups if certain criteria is satisfied. Then the original system is approximated by a *boosted system*, in which the fast reactions are "slowed-down" by a change of time-unit. The new system, which poses less scale-separation in time, is simulated using a hybrid method.

The schematic overview of our method is shown in Fig. 1. The hybrid solver is built in Block B, whose details will be given in Section 2. It is similar to that of Haseltine et al.'s approach [8] but has several crucial improvements. Block C is the monitor block. It does not directly update the system but only collects information. With the collected statistical information over a short time window, the monitor will generate an approximate system as a replacement of the original one. The main tool used in maintaining a reasonable approximation of the system is boosting [29, 30]. It relies on the following idea: in presence of large time-scale separation, it is possible to slow down the fast reactions of a system while maintaining a good approximation to the original effective dynamics. A boosted system becomes less stiff without destroying the effective dynamics. It can then be solved by a hybrid method, so that the reactions among species with large population size can be efficiently simulated. Basically speaking, the time-scale separation is dealt with boosting, and the disparity in the species population is taken care by hybrid method.

The paper is organized as follows. In Section 2 we discuss the criteria for partitioning the reactions and present a improved hybrid method. The boosting strategy is introduced in Section 3. Its combination with the boosting method is presented in Section 4. Various numerical experiments are given in Section 5 illustrating the versatility and efficiency of the method. Finally in Section 6 we summarize our findings and discuss unsolved issues and future works.

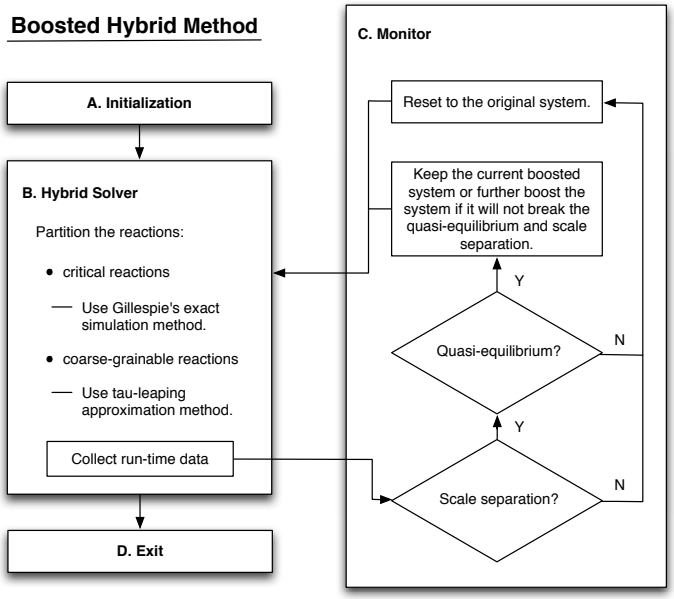

Figure 1: Schematic overview of the boosted hybrid method. It consists of two major components: the monitor and the hybrid solver. The monitor collects information, including the average rates and system state change. With this statistical information, it will decide if quasi-equilibrium is reached. If so, a boosted system will be proposed to replace the original system to be simulated by the hybrid method.

2 Adaptive hybrid method

In this section we start by describing how to adaptively partition the reactions in a complex chemical reaction systems. A hybrid method is then introduced which allows to use different numerical strategies for chemical species with different population size.

2.1 Partition of the system

Consider a well stirred chemical reaction system consisting of N different species interacting through M reaction channels. Let $X_i(t)$ be the population number of the i-th specie at time t, and the system state vector be $\boldsymbol{X}(t)=(X_1(t),X_2(t),\cdots,X_N(t))\in\mathbb{N}^N$. Each reaction channel j has a propensity function, or reaction rate, $a_j(\boldsymbol{x}),\boldsymbol{x}\in\mathbb{N}^N$. The probability that the j-th reaction fires during an infinitesimal time dt is $a_j(\boldsymbol{X}(t))dt$, independent of the other reactions. If the j-th reaction fires, the system is updated as $\boldsymbol{X}(t)\to\boldsymbol{X}(t)+\boldsymbol{\nu}_j$, where $\boldsymbol{\nu}_j=(\nu_{1j},\cdots,\nu_{Nj})^T\in\mathbb{N}^N$ is the state-change vector corresponding to this reaction. Given the initial state $\boldsymbol{X}(0)$ and the aforementioned evolutionary law, we can use Gillespie's SSA [9] to simulate trajectories of $\boldsymbol{X}(t)$ and study its statistical properties. But for systems with multiple well separated time-scales and disparity in species population, this procedure often becomes too costly. The aim of this work is to provide a more efficient simulation strategy for such systems.

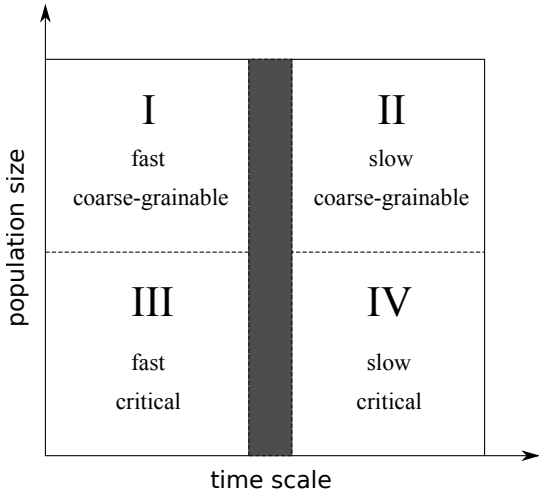

Figure 2: After partition, the reactions are divided into four non-overlapping groups. (I) Fast coarse-grainable reactions, suitable for boosting and coarse-graining approximations. (II) Slow coarse-grainable reactions, suitable for coarse-graining approximation. (III) Fast critical reactions, suitable for boosting approximation. (IV) Slow critical reactions, suitable for SSA. To apply quasi-equilibrium approximation, we need time-scale separation between fast and slow reactions, which is represented by the gray zone in the figure.

The first step is to partition the system. We divide the reactions into different groups according to their property (see Fig. 2). Each group will then be treated with different algorithms. As shown in Fig. 2, we partition all the reactions into four groups based on the species population numbers and the reaction rates. Reactions with fast dynamics are simply called fast reactions, the remaining reactions are called slow reactions. We also distinguish reactions involving species with large population size, called *coarse-grainable* reactions, from reactions involving species with small population size, called *critical* reactions. We give below the implementation details for realizing such a partition.

Adaptive selection of critical and non-critical reactions. Given the system state $\boldsymbol{X}$, we define the *bottleneck specie number* z_j for each reaction j as,

$$z_j(\boldsymbol{X}) = \min_{i=1,\cdots,N;\nu_{ij}\neq 0} \left\{ \frac{X_i}{|\nu_{ij}|} \right\}.$$

Note that if reaction j fires once, at least one specie will be changed by a proportion of $1/z_j$. If z_j is large, this will be just a small change to the system. For these reactions coarse-graining approximation such as tau-leaping can apply. The idea of tau-leaping is to assume the system state to be fixed for a small period of time so that many reactions can be updated in one time. But if z_j is small, the system will experience a significant change, and exact method like the SSA need to be used. We choose a small value ϵ (say, $\epsilon = 0.1$, the same ϵ will be used later in the tau-leaping method). If $z_j < 1/\epsilon$, then reaction j will be called as critical reaction, otherwise as coarse-grainable reaction.

Adaptive selection of fast and slow reactions. We further partition all the reactions as being either fast or slow reactions. Define the *characteristic rate* for reaction j as

$$A_j(\boldsymbol{X}) = \frac{a_j(\boldsymbol{X})}{\max\{\epsilon z_j(\boldsymbol{X}),1\}}, \tag{2.1}$$

where ϵ is a small positive parameter. Note that for critical reactions $A_j(\boldsymbol{X}) = a_j(\boldsymbol{X})$, while for coarse-grainable reactions $A_j(\boldsymbol{X}) < a_j(\boldsymbol{X})$.

To apply quasi-equilibrium approximations, time-scale separation between the averaged characteristic rates is required. Then we can partition all the reactions into four non-overlapping groups as shown in Fig. 2.

2.2 Hybrid method

The hybrid method allows efficient simulation of chemical reaction systems with disparity in species population. It is also a major building block of the boosted hybrid method relying on the use of boosting to handle the time-scale separation and a hybrid treatment of the resulting reactions groups (II) and (IV). The idea is to use SSA for critical reactions and at the same time use tau-leaping method for coarse-grainable reaction. The difficulty is the coupling of time-steps between those used by tau-leaping and SSA. There are a lot of works in this direction [6, 19–25]. The coupling strategy that we use here relies on [8]. We however propose some improvements of the aforementioned work related to an adaptive partitions of the reactions as explained below.

Remark 2.1. In [8], the authors use SSA combined with stochastic differential equation integrator or ordinary differential equation integrator. Here we will discuss the hybridization of SSA and tau-leaping but the framework also apply to other coarse-graining solvers.

Consider a system with only slow reactions. The critical reactions (group (IV)) are labeled as $1,\cdots,q$ and the coarse-grainable reactions (group (II)) are labeled as $q+1,\cdots,M$. Define $a_\Lambda(\boldsymbol{X})$ as the sum of rates of all the critical reactions,

$$a_\Lambda(\boldsymbol{X}) \equiv \sum_{j=1}^{q} a_j(\boldsymbol{X}).$$

The waiting time for the next critical reaction to fire is a random variable τ with probability density function

$$p(\tau) = a_\Lambda(\boldsymbol{X}(t+\tau))\exp\left(-\int_t^{t+\tau} a_\Lambda(\boldsymbol{X}(s))ds\right). \tag{2.2}$$

In order to generate such a random variable, one can first generate a standard uniformly distributed random variable u and then solve

$$\int_{t_0}^{t_0+\tau} \sum_{j=1}^{q} a_j(\boldsymbol{X}(s))ds + \ln u_1 = 0. \tag{2.3}$$

During $(t_0,t_0+\tau)$ the coarse-grainable reactions may change the system state $\boldsymbol{X}(t)$. But under the leaping condition [31]

$$|X_j(t_0+t_{leap})-X_j(t_0)|\leq \epsilon X_j, \quad j=q+1,\cdots,M, \tag{2.4}$$

we can find $t_{leap}>0$ so that before time t_0+t_{leap} the system-change caused by the coarse-grainable reactions can be neglected. Assuming the system state remains unchanged to be $\boldsymbol{X}(t_0)$, Eq. (2.3) predicts $dt_1=-\ln u_1/a_\Lambda(\boldsymbol{X}(t_0))$. If we have $dt_1<t_{leap}$, the leaping condition is not violated and we choose a critical reaction to fire at time t_0+dt_1. We also use this dt_1 as the leaping step-size to update the coarse-grainable reactions in the tau-leaping method. If, however, we have $dt_1>t_{leap}$, then we need to update the coarse-grainable reactions at the time t_0+dt_1. So we update the coarse-grainable reactions using the tau-leaping method with step-size t_{leap} and define $dt_1=t_{leap}$ instead. Then we update the time to $t_0'\equiv t_0+dt_1$. Since no critical reaction has fired, we update Eq. (2.3) as

$$\int_{t_0'}^{t_0+\tau}\sum_{j=1}^{q}a_j(\boldsymbol{X}(s))ds=-\ln u_1-\sum_{j=1}^{q}a_j(\boldsymbol{X}(t_0))dt_1>0.$$

Now we let $t_0\leftarrow t_0'$ and repeat the above procedures to get dt_i $(i=2,\cdots)$ in each step. As the left-hand-side of the above equation will decrease each time, eventually a critical reaction will fire and Eq. (2.3) can be solved. Thus the final $\tau=\sum_i dt_i$. Then, using SSA, the critical reaction that will fire can be randomly chosen by finding a l that satisfies

$$\sum_{j=1}^{l-1}a_j(\boldsymbol{X}(t_0+\tau))<u_2a_\Lambda(\boldsymbol{X}(t_0+\tau))\leq\sum_{j=1}^{l}a_j(\boldsymbol{X}(t_0+\tau)), \tag{2.5}$$

where u_2 is a random variable uniformly distributed in $[0,1]$.

Remark 2.2. Note that after several steps of the numerical method for the coarse-grainable reactions, the classification of the critical and non-critical reactions may no longer be valid. In this case, we have to partition the system again.

Remark 2.3. Eqs. (2.2) and (2.3) are the same as Eqs. (17) and (20a) in [8]. But the way of solving them is different here. Instead of using "no-reaction", which does not change the system state but only makes critical reactions to happen at a more frequent basis in order to scale stochastic time step, we search for the next zero-crossing time point for Eq. (2.3) under the constrain of the leaping condition. The no-reaction approach is easy to use and still applicable here, but it may result in computational overhead because more critical reactions (including the no-reactions) need to be handled. Lastly, we point out that our approach is similar with the work in [20], but the later is based on the *first reaction* variant of Gillespie's SSA method. Generally speaking, there is no significant difference between the applied principles.

Finally we discuss how to generate t_{leap}. There already exist a couple of procedures to select a reasonable t_{leap} that meets the leaping condition Eq. (2.4), such as those in [7, 31, 32]. They can be used directly in our hybrid method. Here we propose another step-size selection strategy, which is simpler to be implemented in our algorithm. We let

$$t_{leap} = \min_{j=q+1,\cdots,M} \frac{1}{A_j(\boldsymbol{X})},$$

where $A_j(\boldsymbol{X})$ is the characteristic rate of the reaction j as given in Eq. (2.1). Note that within a time-step $1/A_j(\boldsymbol{X})$, reaction j fires on average $a_j(\boldsymbol{X})/A_j(\boldsymbol{X}) = \epsilon z_j(\boldsymbol{X})$ times, and cause a change of $\epsilon z_j(\boldsymbol{X})\nu_j \leq \epsilon X_i$ $(i=1,\cdots,N)$, which is consistent with the leaping condition (2.4).

The hybrid method is given in Algorithm 1. Note that we need to do a new partition (step 3 below) each time after we update the system state because the critical reaction set may have changed.

Algorithm 1. Hybrid method.

Initialization: Let $t=0$, give $\boldsymbol{X}(0)$, and set $\epsilon=0.1$, $\gamma=0$.

1. If $\gamma \geq 0$, generate a standard uniform random variables u_1 and let $\gamma = \log u_1 < 0$. Otherwise continue.
2. Compute the rates $a_j(\boldsymbol{X}(t))$, $j=1,\cdots,M$.
3. For each reaction j, compute its bottleneck specie number as $z_j(\boldsymbol{X}(t)) = \min_{\nu_{ij}\neq 0}\{X_i/|\nu_{ij}|\}$, $j=1,\cdots,M$. If $\epsilon z_j(\boldsymbol{X}(t)) \leq 1$, label it as critical reaction, otherwise as coarse-grainable reaction and compute its characteristic rate $A_j(\boldsymbol{X}(t)) = a_j(\boldsymbol{X}(t))/\epsilon z_j(\boldsymbol{X}(t))$.
4. Choose the leaping step-size $t_{leap} = 1/\max(A_j(\boldsymbol{X}(t)))$, for all coarse-grainable reaction j. Compute $\tau = -\gamma/a_\Lambda(\boldsymbol{X}(t))$.
5. If $t_{leap} > \tau$, fire one critical reaction using SSA as in Eq. (2.5). Apply tau-leaping with step-size τ for all coarse-grainable reactions. Update time $t = t+\tau$. Otherwise, apply tau-leaping with step-size t_{leap} for all coarse-grainable reactions, and do not fire critical reactions.
6. Update time $t \leftarrow t + t_{leap}$ and $\gamma \leftarrow \gamma + a_\Lambda(\boldsymbol{X}(t))t_{leap}$. Repeat from step 1.

3 Boosting method for chemical reaction systems

In chemical reaction systems, it is very common to have large time-scale separations. A widely used strategy is quasi-equilibrium approximation. There exist a number of ways of implementing this idea in chemical reaction systems, such as slow-scale SSA, nested SSA. Here we consider another approach called *boosting*, first introduced by Vanden-Eijnden [29]. This approach, as we will see, has the advantage of allowing for an easy implementation. The idea of boosting has been originally proposed in dynamical systems with scale-separation in time, such as stiff ODEs and SDEs. We will see here that it is also a good strategy for stochastic systems arising in chemical reactions.

A simple example. First let us consider a simple chemical reaction system

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

Suppose $c_1=200, c_2=100 \gg c_3=c_4=0.5$, and initially $\boldsymbol{X}(0)=(1,0,0)$. One can think of S_1, S_2 and S_3 as being different forms of a protein. This protein will switch between state S_1 and S_2 at a very fast rate, and it can also switch between state S_1 and S_3 at a much slower rate. Suppose that we are interested in the slow variable S_3. In fact the rate of the third reaction $S_1 \to S_3$ only depends on the amount of time that the protein stays at state S_1, which is determined by the ratio of c_1 and c_2. The idea of boosting for this example is to slow down the fast reaction while maintaining correct behavior of the slow dynamics. We can simply dividing c_1 and c_2 both by 10, so that the protein still oscillate faster between S_1 and S_2. But the system is less stiff since we have decreased the scale separation and it is easier to solve with a standard solver.

3.1 Boosted-SSA

Consider a chemical reaction system with only critical reactions. The slow reactions in group (III) are labeled as $1,2,\cdots,q$, and the fast reactions in group (IV) are labeled as $q+1,q+2,\cdots,M$. Scale separation between fast and slow reactions requires

$$\min(a_{q+1},\cdots,a_M) \gg \max(a_1,\cdots,a_q).$$

If we simulate this system with SSA, fast reactions will fire frequently and involve a high computational cost. But it is only the slow dynamics that we want to capture. If we restrict our attention to slow reactions, the waiting time for the next slow reaction in the system, τ, has the probability density function

$$p(\tau)=\sum_{j=1}^{q} a_j(\boldsymbol{X}(\tau))\exp\left(-\int_0^\tau \sum_{j=1}^{q} a_j(\boldsymbol{X}(s))ds\right). \tag{3.1}$$

The above equation is exact, but $\boldsymbol{X}(t)$ contains the information of the fast reactions which we do not want to resolve. If the fast reactions quickly drive $\boldsymbol{X}(t)$ into quasi-equilibrium, then we can approximate the effective rate of slow reaction j b_j as

$$b_j \equiv \frac{1}{\tau}\int_0^\tau a_j(\boldsymbol{X}(s))ds \approx \frac{1}{h}\int_0^h a_j(\boldsymbol{X}(s))ds, \quad j=1,\cdots,q. \tag{3.2}$$

Here h a parameter chosen so that it is large in microscopic sense, meaning quasi-equilibrium is well established for $\boldsymbol{X}(t)$, and small in macroscopic sense, meaning h is

far less than the waiting time of the next slow reaction τ. Assume $h=\kappa\tau$, where $\kappa\ll 1$. In (3.2) we make the transformation

$$b_j \approx \frac{1}{h}\int_0^h a_j(\boldsymbol{X}(s))ds = \frac{1}{\tau}\int_0^\tau a_j(\boldsymbol{X}(\kappa s))ds, \tag{3.3}$$

for $j=1,\cdots,q$. The above equation implies that we can approximate the effective rate b_j by "slowing down" the fast reactions by a factor of κ, which can be done simply by setting the rate constants as $c_j'=\kappa c_j$, $j=1,\cdots,q$ to form a modified system. The boosted-SSA as given by Algorithm 2 is nothing but to solve a modified system with SSA.

Algorithm 2. Boosted-SSA.

1. Divide all reactions into slow and fast reactions.
2. Choose parameter $\kappa\ll 1$ (we will discuss how to choose κ in Section 4). Modify the system by rescalling the rate constants of all the fast reactions as $c_j'=\kappa c_j$.
3. Simulate the modified system using SSA.

3.2 Comparison with slow-scale SSA and nested SSA

The slow-scale SSA [17] and nested SSA [18] are also based on quasi-equilibrium approximation of the fast reactions in the system. In the slow-scale SSA, the system state is divided into fast and slow variables as $\boldsymbol{X}=(\boldsymbol{Z},\boldsymbol{Y})$. Conditioned on the current state of the slow variable $\boldsymbol{Y}=\boldsymbol{y}$, it calculates the conditional probability distribution function of the fast variables $p(\boldsymbol{z}|\boldsymbol{y})$, analytically or approximately. Then it computes the effective rate $b_j(\boldsymbol{y})$ of the slow reaction j by using

$$b_j(\boldsymbol{y}) = \sum_{\boldsymbol{z}} a_j(\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{y}). \tag{3.4}$$

This amount to integrate out the fast variable $\boldsymbol{z}$ in the system, which leads to a reduced system with state $\boldsymbol{Y}$ and slow reactions with rates $b_j(\boldsymbol{y})$. Then Gillespie's SSA is applied to the reduced slow-scale system and $\boldsymbol{Y}$ gets updated. The slow-scale SSA is very efficient if one can obtain the conditional distribution $p(\boldsymbol{z}|\boldsymbol{y})$. However, for complex system there is no systematic way to do so.

The nested SSA does not require an explicit form of $p(\boldsymbol{z}|\boldsymbol{y})$, not even the partition of slow and fast variables. It only partitions the reactions into slow and fast reactions, which is relatively easier. It then uses an inner-SSA subroutine to simulate only the fast reactions for a small amount of time h to compute the effective rate of slow reactions j as

$$b_j = \frac{1}{h}\int_0^h a_j(\boldsymbol{X}(s))ds.$$

The above equation is exactly (3.2). With the effective rates, an outer-SSA subroutine is called to simulate one single event among the slow reactions and update the system state and the time.

The nested SSA and boosted SSA share similar theoretical basis: both need to partition the reactions, and share the same quasi-equilibration time h. The major difference is in the implementation: in the former, the fast and slow reactions are simulated separately and only the slow reactions get updated, but in the latter, the fast reactions are first slowed down, then all the reactions are updated together. Boosting is easier to implement, especially for multiscale systems with a hierarchy of time scales. In such a situation, if using nested-SSA, we need to identify and order the hierarchy of time scales and implement successively a hierarchy of fast solvers for the different scales involved with the averaged rates for one level coming from the quasi-equilibrium of the fast reaction at the previous level. In the boosting framework, one does not need any additional subroutine to handle the time scale separations, we just boost the reaction rates of the fast dynamics, and use some single scale hybrid solver in hand to perform the time integration.

3.3 Boosted tau-leaping

Boosting can also be used in coarse-grainable reactions, so to get the *boosted tau-leaping*. Because in tau-leaping the step-size τ is deterministic, the conventional procedure of boosting that is used in solving stiff ODEs and SDEs can be applied here. The following algorithm can be obtained by following [30].

Algorithm 3. Seamless tau-leaping with fixed step-size.

1. Divide all reactions into slow and fast reactions.
2. Select the step-size δt for the fast reactions and Δt for the slow reactions. Choose a parameter K, which is the estimated number of steps that the fast reactions require to reach quasi-equilibrium. Because of the time-scale separation between fast and slow reactions, we have $K\delta t \ll \tau$. Repeat the following procedure K times:
   (a) Integrate the fast reactions with step-size δt using tau-leaping.
   (b) Integrate the slow reactions with step-size $\Delta t' = \Delta t/K$ using tau-leaping.
3. Repeat from step 1.

While the above scheme works for tau-leaping, we do not know how to generalize it into SSA because of the randomness of the step-size. The boosted tau-leaping algorithm is as follows:

Algorithm 4. Boosted tau-leaping with modified rates.

1. Choose $\kappa = K\delta t/\Delta t$ and modify the rate constants of the fast reactions as $c' = \kappa c$.
2. Integrate the whole system with step-size $\Delta t/K$.

In fact, Algorithms 3 and 4 are theoretically equivalent, but the latter is more flexible, and more importantly, it works for both SSA and tau-leaping.

4 Boosted hybrid method

In this section we introduce the boosted hybrid method as an adaptive solver for multiscale chemical reaction system. As Fig. 1 shows, it has two major components. The monitor will maintain an approximating system. It will monitor the system state during the simulation to see if there exists quasi-equilibrium for fast reactions and decide whether it is possible to boost the system to reduce stiffness. The approximating system is simulated using the hybrid method.

4.1 The monitor

We use the hybrid method to simulate the chemical system for a time of length Δt, which is called a monitoring window. Information such as reaction rates and system state is collected during the monitoring window, with which the monitor can identify time-scale separation between fast and slow reactions, and if so, check if quasi-equilibrium is reached for the fast reactions. Based on these judgments, the method is able to maintain a reasonable approximation of the original system.

We say that time-scale separation between the fast reactions group Ω and its complementary, the slow reactions group, Ω^c, exist if

$$\min_{j\in\Omega} A_j > 10^q \times \max_{j\in\Omega^c} A_j > \omega, \tag{4.1}$$

where A_j is the characteristic rates in Eq. (2.1). There are two parameters in the above formula, q and ω. The parameter q represents the degree of scale separation, for example, $q=2$ corresponds to a 100 times scale difference between fast and slow reactions. The parameter ω indicates the time-scale we are interested in. In other words, we only apply boosting to reactions whose rates are greater than ω. It may vary from system to system, based on the particular dynamics we want to learn. Note that in out algorithm we use the averaged characteristic rates in Eq. (4.1). This is because for some fast reactions their rates may vary rapidly. For simplicity, in the rest of the paper we still denote the averaged rates as A_j.

Time-scale separation between the fast reactions Ω and slow reactions Ω^c does not guarantee quasi-equilibrium. We claim quasi-equilibrium for reactions in Ω is reached if, during the last monitoring window, the total flux due to reactions in Ω is much larger than the net-change of the system state (a similar idea as used in [27]). More specifically, in a monitoring window, we compute the total flux as

$$\boldsymbol{X}^{flux} = \sum_j r_j |\boldsymbol{\nu}_j|,\ j\in\Omega,$$

where r_j is the total number that reaction j fired, and record X_i^{min} and X_i^{max}, the minimum and the maximum state reached during the monitoring window, respectively. We then

compute the so-called *redundancy coefficient* defined by,

$$\zeta = \min_{i=1,\cdots,N} \frac{\max(X_i^{flux},1)}{\max(X_i^{max}-X_i^{min},1)}. \tag{4.2}$$

We take $\max(X_i^{flux},1)$ and $\max(X_i^{max}-X_i^{min},1)$ to avoid having vanishing terms. We will assume that the fast reactions have reached their quasi-equilibrium if $\zeta \geq 10$, and not if $\zeta < 10$.

In the algorithm, we keep a vector called *boosting vector* $\boldsymbol{\kappa} = (\kappa_1,\cdots,\kappa_M)$. The approximating system has rate constant $c_j' = c_j\kappa_j$. If $\boldsymbol{\kappa} = \mathbf{1}$, we just have the original system. If some $\kappa_j < 1$, it means that in the boosted system reaction j has been slowed down. Now we introduce two parameters $\zeta_{crit,1}$, $\zeta_{crit,2}$ related to critical values of ζ. Above the value $\zeta_{crit,2}$ the reactions will be considered fast enough to be slowed down. We implement the following strategy:

- If $\zeta < \zeta_{crit,1}$ we reset the approximating system to the original one by setting $\boldsymbol{\kappa} = \mathbf{1}$.
- If $\zeta > \zeta_{crit,2}$ we boost the system by let $\kappa_j = 0.75\kappa_j$, $j \in \Omega$.
- If $\zeta_{crit,1} \leq \zeta \leq \zeta_{crit,2}$, we just hold $\boldsymbol{\kappa}$ unchanged.

The parameters $\zeta_{crit,1}$, $\zeta_{crit,2}$ are chosen in an empirical way and different values leads to different performance of the algorithm. Clearly more investigation is needed to set these values, which is left as a future work. In the numerical experiments we set $\zeta_{crit,1} = 10$, $\zeta_{crit,2} = 30$.

Note that the boosting is done gradually based on the current system state. When boosting is turned on, we only multiply κ_j by 0.75 for the fast reactions. Keep in mind the rates of all the fast reactions should be multiplied by the same constant other wise the effective dynamics will be altered. At the next monitoring window, we may keep reducing some κ_j, hold the current $\boldsymbol{\kappa}$ unchanged, or even reset $\boldsymbol{\kappa} = 1$. The above adaptive process can thus automatically handle systems with multiple scale separations hierarchically. For example, if a system consists of three groups of reactions with disparate rates, the algorithm will first slow down the fastest reactions. Then, it will consider the two fastest group of reactions as long as quasi-equilibrium approximation is valid.

The length of the monitoring window should be large enough to let the fast reactions reach equilibrium, but not too large, otherwise both the efficiency and the accuracy of the system may be compromised. We choose Δt to be 100 times the minimum characteristic rate among all the fast reactions:

$$\Delta t = 100/\min_{j\in\Omega} A_j.$$

4.2 The overall algorithm

A. Initialization Set the initial molecule number $x_i = x_i(0)$ $(i=1,\cdots,N)$, the boosting coefficient $\kappa_j=1$ $(j=1,\cdots,M)$, the initial time $t=0$, $\gamma=0$, the leaping parameter $\epsilon=0.1$, the initial window length $\Delta t=0.2$, and the fast reaction set Ω to empty set. Choose $q=2$ and a slow scale ω for the system. Invoke hybrid solver **Subroutine B**.

B. Hybrid Solver (κ_j, Δt, Ω as input)

1. Let $t_{monitor}=t+\Delta t$.
2. If $\gamma\geq 0$, generate a uniformly distributed random variables u_1 and let $\gamma=\log u_1<0$.
3. Compute propensity functions a_j and do the boosting $a_j=\kappa_j a_j$ $(j=1,\cdots,M)$.
4. For each reaction j, compute its bottleneck-specie number $z_j(\boldsymbol{X})=\min_{\nu_{ij}\neq 0}\{X_i/|\nu_{ij}|\}$, $j=1,\cdots,M$. If $\epsilon z_j(\boldsymbol{X})\leq 1$, label it as critical reaction, otherwise coarse-grainable reaction with coarse-grained reaction rate $A_j(\boldsymbol{X}(t))=a_j(\boldsymbol{X}(t))/\epsilon z_j(\boldsymbol{X}(t))$.
5. Choose step-size: first choose the leaping step-size $t_{leap}=1/\max(A_j(\boldsymbol{X}(t)))$, where j runs over all coarse-grainable reactions. Then compute the waiting time for the next critical reaction $\tau=\gamma/a_\Lambda$ using $a_\Lambda=\sum a_j$, where j runs over all critical reactions. The step-size is chosen as $dt=\min(t_{leap},\tau)$.
6. Apply τ-leaping with step-size dt for all coarse-grainable reactions. Let $\gamma=\gamma+a_\Lambda dt$, update time $t=t+dt$. If $t_{leap}>\tau$, simulate one reaction events among the critical reactions using SSA.
7. Invoke **Subroutine D** if exit conditions are met, otherwise collect information:
$$\begin{aligned}&\boldsymbol{X}^{max}=\max(\boldsymbol{X}(t),\boldsymbol{X}^{max}),\\&\boldsymbol{X}^{min}=\min(\boldsymbol{X}(t),\boldsymbol{X}^{min}),\\&\boldsymbol{X}^{flux}=\boldsymbol{X}^{flux}+\sum_j r_j|\boldsymbol{\nu}_j|,\quad j\in\Omega.\end{aligned}$$
8. If $t>t_{monitor}$, invoke **Subroutine C**. Otherwise go back to Step 2.

C. Monitor

1. Compute the redundancy coefficient
$$\zeta=\min_i \frac{\max(X_i^{flux},1)}{\max(X_i^{max}-X_i^{min},1)}.$$
2. If there exists a non empty set Ω_{new} such that
$$\min_{j\in\Omega_{new}} A_j > 10^q \times \max_{j\in\Omega_{new}^c} A_j > \omega, \tag{4.3}$$
then there exists a scale separation in reaction rates between Ω_{new} and Ω_{new}^c and we define
$$\Delta t_{new}=100/\max_{j\in\Omega_{new}^c} A_j.$$

Table 1: Rules that decide whether to apply boosting or not in the boosted hybrid method. The symbol "$\surd$", "$\times$", or "—" indicates whether a specific condition is satisfied, not satisfied, or does not matter. "s.s." is short for scale separation. "ζ" is the redundancy coefficient given by Eq. (4.2). "Δt and Δt_{new}" is the time length of the previous and next monitor window, respectively. "Ω and Ω_{new}" is the fast reactions set in the previous and next monitor window, respectively. Under different conditions, we will take different "actions", with the method of choosing the new window length Δt, the boosting coefficient $\boldsymbol{\kappa}$ and the fast reaction table Ω for the next monitor window. Some rational of doing this is given in the appendix.

s.s.	ζ	$\Delta t_{new}<5\Delta t$	$\Omega_{new}=\Omega$	action
$\times$	<10	—	—	$\kappa_j=1, j=1,\cdots,M$, keep the old Δt and Ω
$\times$	≥ 10	—	—	keep the old κ_j, Δt and Ω
$\surd$	—	$\times$	—	keep the old κ_j for this and the next monitoring windows, use the old Δt, use Ω_{new}
$\surd$	—	$\surd$	$\times$	keep the old κ_j, use Ω_{new} and Δt_{new}
$\surd$	>30	$\surd$	$\surd$	$\kappa_j=0.75\kappa_j, j\in\Omega$, use Ω_{new} and Δt_{new}
$\surd$	[10,30]	$\surd$	$\surd$	keep the old κ_j, use Ω_{new} and Δt_{new}
$\surd$	<10	$\surd$	$\surd$	$\kappa_j=1, j=1,\cdots,M$, use Ω_{new} and Δt_{new}
If there exists $\kappa_j<1$ and $\min_{\kappa_j<1}(A_j)<5\max_{\kappa_j=1}(A_j)$				$\kappa_j=1, j=1,\cdots,M$, use Ω_{new} and Δt_{new}

3. Set the window length Δt, the boosting coefficient $\boldsymbol{\kappa}$ and the fast reaction set Ω according to the rules given by Table 1. Invoke **Subroutine B**.

D. Exit

5 Numerical examples

In this section we test the efficiency and the versatility of our method on five numerical examples. We will see that for the first three examples, our method is efficient and can handle numerically the chemical systems in robust and adaptive way. For the last two examples, the method shows limited applicability. The comparison of our method with other algorithms will be discussed in a future work.

5.1 System 1

This system is a toy model to test the method. It has five species and seven reactions.

$$\begin{aligned}
1,2:&\quad \mathrm{S}_1+\mathrm{L}_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} \mathrm{S}_2+\mathrm{L}_2,\\
3,4:&\quad \mathrm{S}_1 \underset{c_4}{\overset{c_3}{\rightleftharpoons}} \mathrm{S}_3,\\
5:&\quad \mathrm{L}_1 \xrightarrow{c_5} 0,\\
6,7:&\quad \mathrm{L}_1 \underset{c_7}{\overset{c_6}{\rightleftharpoons}} \mathrm{L}_2.
\end{aligned}$$

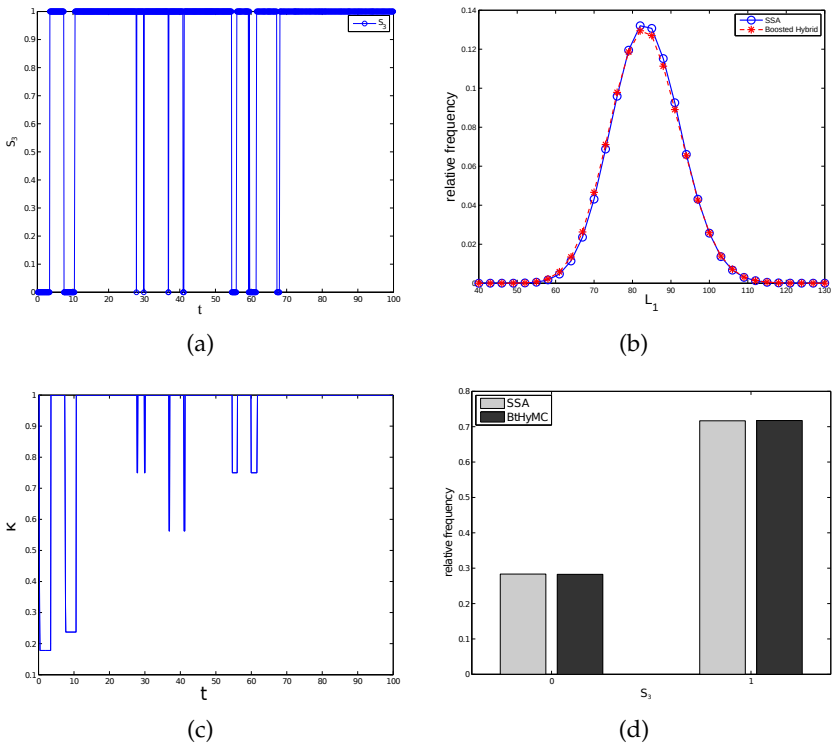

Figure 3: Numerical results for System 1. (a) A trajectory of S_3 obtained by the boosted hybrid method (when $S_3=0$ the reactions 1 and 2 are fast); (b) Histogram of L_1 sampled from SSA (circle, solid line) and the boosted hybrid method (star, dashed line) at final time, with sample size 10^6; (c) Evolution of κ_1 and κ_2 in the boosted hybrid method (the two are identical in the figure), the other components of $\boldsymbol{\kappa}$ are all equal to 1; (d) Histogram of S_3 sampled from SSA and the boosted hybrid method at final time, with sample size 10^6.

The initial condition is $L_1=L_2=1000$, $S_1=1$, $S_2=S_3=0$. The rate constants are $c_1=10$, $c_2=10$, $c_3=1$, $c_4=0.2$, $c_5=0.05$, $c_6=1$,$c_7=1$. The system is simulated in time $[0,100]$.

Fig. 3 shows the numerical results of the boosted hybrid method on System 1. When

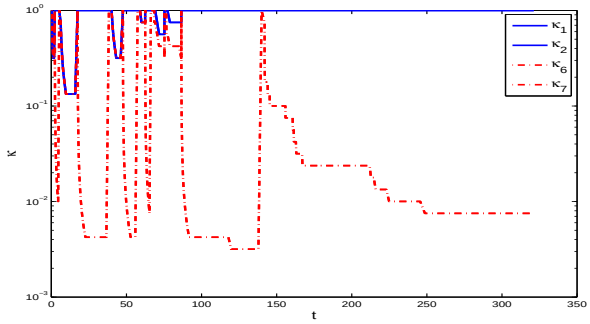

Figure 4: One sampled trajectory of κ for the modified system 1. κ_1 (solid line) and κ_6 (dashed line) coincide with κ_2 and κ_7. The other entries of $\boldsymbol{\kappa}$ equal to 1 all the time (log scale for the y axis).

S_3=0, κ_1 and κ_2 are less than 1. This means that the algorithm identifies quasi-equilibrium of the fast reactions 1 and 2, and applies boosting to them. More interestingly, as $L_{1,2}$ decrease, this time-scale separation gets weaker and the algorithm automatically adjusts the boosting coefficients $\boldsymbol{\kappa}$. Eventually, the fast reactions 1 and 2 are no longer fast enough to support any boosting. When this happens, κ_1 and κ_2 are equal to 1 (even though $S_3 = 0$), which demonstrates the adaptivity of the algorithm. Concerning the accuracy of the method, we can see from the histograms of L_1 and S_3 that the results of the boosted hybrid method approximate those of the SSA very well.

Next, we make some modifications to the system 1 by letting c_3=0.2, c_4=0.1, c_6=1000, $c_7 = 1000$, and simulate the modified system in the time interval [0,400]. The remaining parameters are unchanged. The modified system contains hierarchical scale separation in time, namely, reactions 6, 7 are "very fast", reactions 1, 2 are "fast", and the remaining ones are "slow". One trajectory of the boosting coefficient κ is given in Fig. 4. The algorithm first recognizes reactions 6 and 7 as fast group and boosts the system by reducing κ_6 and κ_7. After the first round of boosting, reactions 6, 7, 1, 2 are treated as fast group and the system can be further boosted. Note also that when t is close to 150, L_1 drops below 10, and reactions 6, 7 become critical and must be simulated using SSA. $\boldsymbol{\kappa}$ is set to **1** here when some reaction change from coarse-grainable to critical type. However, critical reactions 6 and 7 are still fast enough for new boosting and κ_6 and κ_7 decrease again. We can see from this example that the new algorithm is capable of treating system with hierarchical time scales whose hierarchy can change over time.

5.2 System 2

This system is adapted from [28] and was originally proposed in [33], in which it was used to model stochastic gene regulation. It has four reactant species θ_0,θ_1,M,D and seven reactions.

$$
\begin{aligned}
1,2:&\quad 2\mathrm{M} \underset{c_2}{\overset{c_1}{\rightleftharpoons}} \mathrm{D},\\
3,4:&\quad \theta_0 + \mathrm{D} \underset{c_4}{\overset{c_3}{\rightleftharpoons}} \theta_1,\\
5:&\quad \theta_0 \xrightarrow{c_5} \theta_0 + \mathrm{M},\\
6:&\quad \theta_1 \xrightarrow{c_6} \theta_1 + \mathrm{M},\\
7:&\quad \mathrm{M} \xrightarrow{c_7} \emptyset.
\end{aligned}
$$

Initially we have $\theta_0 = 1, \theta_1 = 0, \mathrm{M} = 500, \mathrm{D} = 100$. The rate constants are $c_1 = 10, c_2 = 10000, c_3 = 0.02, c_4 = 1.5, c_5 = 50, c_6 = 1000, c_7 = 1$. Fig. 5 shows a trajectory of M and κ_1 in time interval [2000,4000]. In this system, M appears to be have a bistable pattern. When M is relatively large, reactions 1, 2 are very fast, and hence κ_1 and κ_2 are very small. When M is small, reactions 1and 2 are not as fast as before, but still fast enough to be boosted (with a larger κ). Also note that M and D sometimes get very unstable, for example

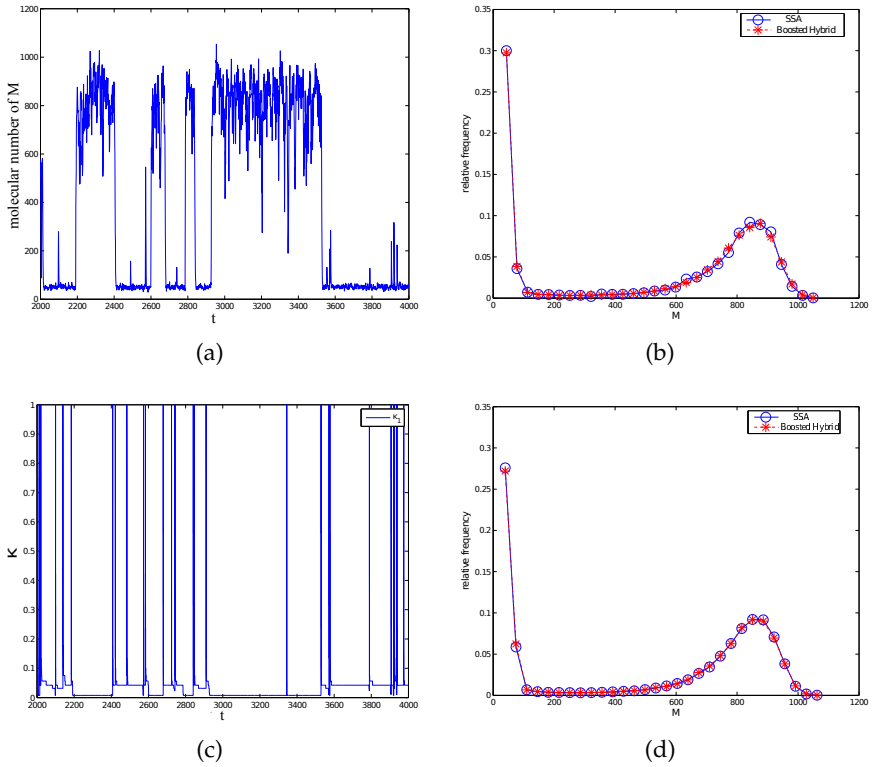

Figure 5: Numerical results for System 2. (a) A typical trajectory of M obtained by the boosted hybrid method; (b) Histograms of M at time $t=10$ obtained by using 10^5 samples with the boosted hybrid method (star, dashed line) and SSA (circle, solid line), respectively. (c) A typical trajectory of κ_1 in the boosted hybrid method. κ_2 is identical with κ_1 and others are equal to 1; (d) Histograms of M over time $[1000,200000]$ obtained by using the boosted hybrid method (star, dashed line) and SSA (circle, solid line), respectively.

around $t=3340$, at this time the approximating system is reset to the original system by the algorithm. This is exactly what we need, because such transition period are usually very important and interesting and quasi-equilibrium approximation is not valid there.

In order to compare the accuracy of the boosted hybrid method with SSA, we first obtain ensemble histograms for specie M using 10^5 samples at time $t=10$ ((b) in Fig. 5). We also simulate a trajectory up-to time 200, 000 and draw a sample of specie M at each integer value of time starting from t = 1000, from which we get the time histograms ((d) in Fig. 5). It shows the result of the boosted hybrid method matches with SSA quite well.

To test efficiency, we let $c_1=10k$, $c_2=10000k$, with $k=0.001,0.01,0.1,1$ to see the performance of the boosted hybrid method for varying time-scale separation (the larger the k is, the larger the time-scale separation will be in this system). The time taken for each simulation of one trajectory during $[0,20000]$ using SSA and boosted hybrid method, respectively, is listed in Table 2. SSA becomes dramatically slow as k increases, but the boosted hybrid method is still fast because of its adaptivity.

Table 2: Simulation time using SSA and boosted hybrid for system 2 with different k. As k increases, the system becomes more stiff.

k	SSA	boosted hybrid
0.001	22s	5s
0.01	147s	30s
0.1	1435s	35s
1	19559s	44s

5.3 System 3

This model is adapted from [27]. There are seven species and ten reactions.

$$
\begin{aligned}
1:&\quad D* \to D+M+R,\\
2:&\quad M \to M+P,\\
3:&\quad M \to 0,\\
4:&\quad P \to 0,\\
5:&\quad D+R \to D*,\\
6:&\quad D* \to D+R,\\
7:&\quad P+P \to P2,\\
8:&\quad P2 \to P+P,\\
9:&\quad D+P2 \to Q,\\
10:&\quad Q \to D+P2.
\end{aligned}
$$

Here D and D^* represents the activated and deactivated states of a DNA molecule, respectively. M is the mRNA, R is the RNA, P is a protein, and P_2 a protein dimmer. Initially we have $D^*=1$, $R=30$, the other specie populations are all zeros. The rate constants are $c_1=0.0078$, $c_2=0.043$, $c_3=0.0039$, $c_4=0.0007$, $c_5=0.38$, $c_6=3$, $c_7=0.5$, $c_8=5$, $c_9=0.12$, $c_{10}=9$. The simulation time is $[0,50000]$. In [27], reactions 5, 6, 9, 10 are approximated by slow-scale SSA. Reactions 7, 8 are first solved by explicit τ-leaping then implicit τ-leaping when the system becomes more stiff.

For this system, we choose different scale-separation threshold $q=1,2$ (see Eq.(4.1)). $q=1$ means there must be at least one order of magnitude scale-separation between the fast and slow reaction rates to try boosting, and $q=2$ means two orders of magnitude scale-separation. So $q=1$ is a more aggressive boosting strategy. Fig. 6 (a) and (b) show a typical trajectory of $\boldsymbol{\kappa}$ using different q. (c) and (d) show the histograms sampled from 10^5 trajectories simulated by using SSA and the boosted hybrid method to $t=50000$. We can see by choosing more aggressive boosting strategy, $q=1$, $\boldsymbol{\kappa}$ is smaller and hence the algorithm is faster (about 0.7 second for one trajectory for $q=1$, 16.2 seconds for $q=2$). But doing so also increases numerical errors as shown in (d).

We also modified other parameters in the algorithm, such as ϵ that controls the coarse-grained time-step. We change ϵ from 0.1 to 0.05. So now critical reaction is more frequent

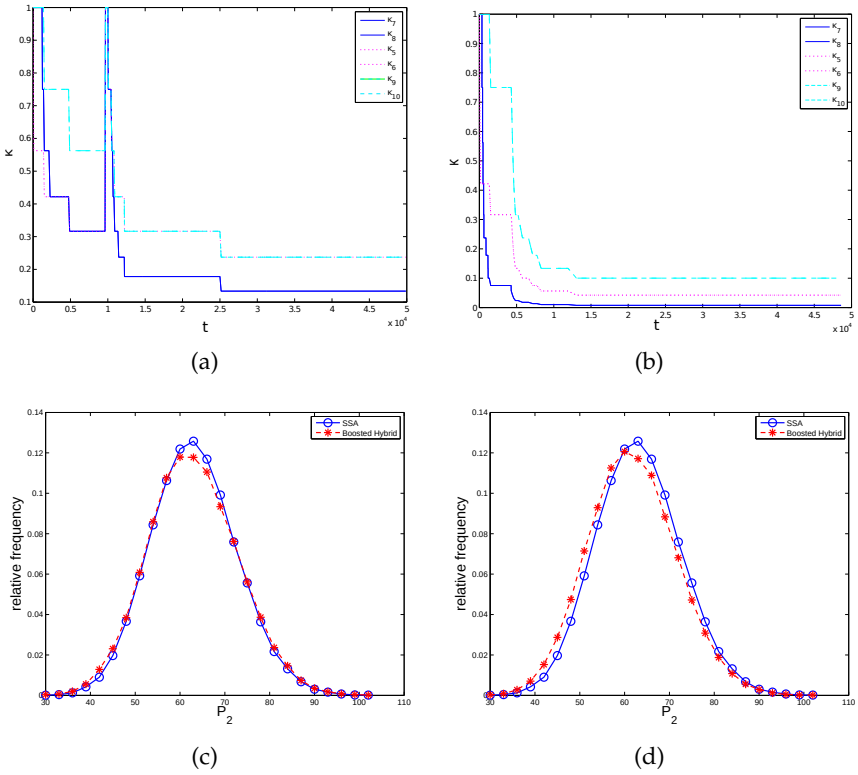

Figure 6: Numerical results for system 3. (a) A typical trajectory of $\boldsymbol{\kappa}$ in the boosted hybrid method for $q=2$. In this case, there are always two elements of κ are identical; (b) A typical trajectory of $\boldsymbol{\kappa}$ in the boosted hybrid method for $q=1$; (c) Histograms of P_2 obtained by using the boosted hybrid method (circle, solid line) and SSA (star, dashed line) at $t=50000$, $q=2$, sample size 10^5; (d) Histograms of P_2 obtained by using the boosted hybrid method (circle, solid line) and SSA (star, dashed line) at $t=50000$, $q=1$, sample size 10^5.

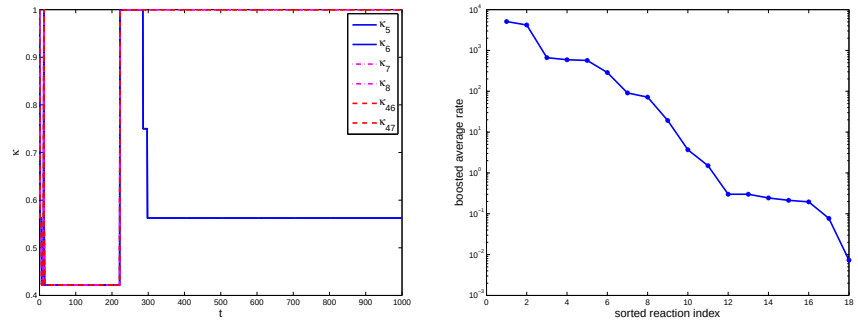

Figure 7: Numerical results for system 4. (a) A typical trajectory of $\boldsymbol{\kappa}$. The entries of κ not shown in the picture are equal to one; (b) Boosted average coarse-grain rates $\kappa_j A_j$ (log scale in the y-axis). It shows that, after boosting, the rates still spread over a large interval without obvious scale-separation.

and the leaping step-size is smaller. The algorithm will be much slower but no appreciable improvement in accuracy is observed for this system (results are not shown).

5.4 System 4

This system describes the heat shock response of the E. Coli bacteria [34, 35]. It consists of 28 species and 61 reactions as given in the Appendix B.

The initial amount of species are $s_1=s_2=s_3=s_4=0$, $s_5=1$, $s_6=4645670$, $s_7=1324$, $s_8=80$, $s_9=16$, $s_{10}=3413$, $s_{11}=29$, $s_{12}=584$, $s_{13}=1$, $s_{14}=22$, $s_{15}=0$, $s_{16}=171440$, $s_{17}=9150$, $s_{18}=2280$, $s_{19}=6$, $s_{20}=596$, $s_{21}=0$, $s_{22}=13$, $s_{23}=3$, $s_{24}=3$, $s_{25}=7$, $s_{26}=0$, $s_{27}=260$, $s_{28}=0$.

The numerical results are shown in Fig. 7. During the simulation, six components of $\boldsymbol{\kappa}$ are less than 1, but they are not very small (see Fig. 7 (a)). This means that the effect of boosting is limited. Moreover, even after boosting, Fig. 7 (b) shows that the rates A_j still occupy quite a large scope, which means the system is still very stiff. Since here the scale separation is not large enough we can not boost the system further. For this system, the boosted hybrid method is even slower than SSA because of the widely and "continuously" distributed reaction rates.

5.5 System 5

Another limitation of the new method (and most of the current simulation method) is illustrated by the following simple example,

$$\begin{aligned}
&1:\ \emptyset \to \mathrm{A},\\
&2:\ \mathrm{A}+\mathrm{E} \to \mathrm{EA},\\
&3:\ \mathrm{EA} \to \mathrm{B}+\mathrm{E},\\
&4:\ \mathrm{B} \to \emptyset.
\end{aligned}$$

A is a transcription factor, E is DNA, EA is the protein-DNA binding complex, and B is protein. Usually the species A and B have large populations and specie E is only one in molecular number. Assume the rate constant of reactions 2 and 3 are much faster than reactions 1 and 4. The overall effect is that A is constantly converted into B. In this system, reactions 1 and 4 are suitable for coarse-grained approximation while reactions 2 and 3 are critical. Moreover, the number of E will quickly reach to a stationary distribution. However, we can not apply boosting here because no partial equilibrium holds for the fast reactions 2 and 3 as there is a non-zero flux from A to B.

A possible approximation to simulate the above system fast may be done by combining the two fast critical reactions into one slow coarse-grained reactions. This gives a modified system

$$\begin{aligned}
&1:\ \emptyset \to \mathrm{A},\\
&2:\ \mathrm{A} \to \mathrm{B}, \text{ or } \emptyset \to \mathrm{B},\\
&3:\ \mathrm{B} \to \emptyset.
\end{aligned}$$

Note that the reaction 2 in the above system does not exist in the original system. This amounts to modify the reaction network itself, rather then treating different reactions differently. One should be very careful when using the above approximation because the model itself has been changed. Some progress for handling this kind of problems are reported in [36] but they remain, in general, difficult to simulate due to the difficulty of dealing with these systems in a systematic way.

6 Discussion

We proposed a new numerical method to simulate chemical reaction systems with time-scale separation and disparity in population species. The method relies on boosting, which is a strategy to decrease the stiffness of a system in presence of time-scale separation. After stiffness reduction, the system may still exhibit disparity in species and we suggest the use of a hybrid method for its simulation in which certain reactions are coarse-grained. We showed that both stiffness reduction and coarse-graining approximation can be adapted in an automatic way in time. This adaptivity is suitable in practice. We notice that works remain to be done to optimize some parameters of the algorithm. Even so, by combining boosting and hybrid methods it is possible to save substantial computational cost for many complex systems, when compared to SSA. This has been illustrated numerically for various chemical systems. We showed how the main source of error in the algorithm, the error from boosting and the error from coarse-graining, can be controlled by tuning the scale separation threshold at which the boosting is turned on or by tuning the parameter threshold to set the number of reactions chosen for coarse-graining.

The new algorithm has many advantages. First, for chemical systems with multiple time scales, it can hierarchically slow down (boost) the system which can then be numerically integrated by a single-scale solver. It avoids the use of multiscale solver such as the slow-scale SSA or nested SSA solver which makes its coding easier for complex systems. Second, it does not need an *a priori* knowledge of the features of the chemical system (such as reaction rates and population size). Third, it allows for a systematic control of the main sources of the numerical errors coming from the boosting in time and the coarse graining procedure for species with large population size. Numerical experiments show good performance for the long-time simulation of systems with multiple scales in time and population size.

We found two situations in which the boosted hybrid method is not very efficient. First, when the reaction rates are spread over a large interval without obvious scale separation, boosting strategy is difficult to be applied. In this situation we do not know of any other method capable of handling efficiently this kind of system. We note that thanks to the adaptivity in the boosted hybrid method, the algorithm remains robust (i.e. gives a reasonable approximation of the original system) even though not it is not efficient when compared to SSA. The second situation, illustrated in Section 5.5, is concerned with a

system where an efficient simulation requires to change the original model. This remains a challenging and interesting task for future work. Refinement and further applications of the boosted hybrid method is also currently under investigation.

Acknowledgments

Hu and Li are supported by the National Science Foundation of China under grant 10871010. The authors also thank the referees for their constructive comments and suggestions to improve the current paper.

7 Appendix

7.1 Some implementation details

We collect here a few details for a practical implementation of the boosted hybrid algorithm.

1. Initially, we just let the fast reaction set Ω to be empty, and pick up a small window length for monitoring, for example, $\Delta t = 0.2$.
2. After boosting, since the fast reaction rates tend to decrease, the new fast reactions set Ω_{new} may become empty. We do not reset $\boldsymbol{\kappa}$ to $\mathbf{1}$ here, but keep using the old Ω coming from the last step. But as soon as $\zeta < 10$ (see Section 4.2), we rest $\boldsymbol{\kappa}$ to $\mathbf{1}$.
3. If $\min_{\kappa_j<1}(A_j) < 5\max_{\kappa_j=1}(A_j)$ we reset $\boldsymbol{\kappa}$ to $\mathbf{1}$. This guarantees enough time-scale separation between boosted reactions and non-boosted reactions.
4. Some care has to be taken whether or not to accept the new set a fast reaction Ω_{new}. In our implementation, when Δt_{new} given by the algorithm has changed by magnitude from the old Δt, we will keep the current partition for the system and do another monitoring step with time-step Δt_{new} to see if Ω_{new} have reached quasi-equilibrium state.

7.2 List of chemical reactions for system 4

There are totally 28 species and 61 reactions for system 4 as given below.

$$S_1+S_2 \underset{c_2=1}{\overset{c_1=2.54}{\rightleftharpoons}} S_3; \qquad S_1+S_4 \underset{c_4=1}{\overset{c_3=0.254}{\rightleftharpoons}} S_5; \qquad S_1+S_6 \underset{c_6=10}{\overset{c_5=0.0254}{\rightleftharpoons}} S_7;$$

$$S_4+S_{14} \underset{c_8=10000}{\overset{c_7=254}{\rightleftharpoons}} S_{15}; \qquad S_{14}+S_{16} \underset{c_{10}=0.01}{\overset{c_9=0.000254}{\rightleftharpoons}} S_{17}; \qquad S_3+S_6 \underset{c_{12}=1}{\overset{c_{11}=0.000254}{\rightleftharpoons}} S_8;$$

$$S_5+S_6 \underset{c_{14}=1}{\overset{c_{13}=0.000254}{\rightleftharpoons}} S_9; \qquad S_3+S_{10} \underset{c_{16}=1}{\overset{c_{15}=2.54}{\rightleftharpoons}} S_{12}; \qquad S_5+S_{11} \underset{c_{18}=1000}{\overset{c_{17}=2540}{\rightleftharpoons}} S_{13};$$

$$S_{15}+S_{18} \underset{c_{20}=1}{\overset{c_{19}=0.0254}{\rightleftharpoons}} S_{19}; \quad \emptyset \underset{c_{22}=0.5}{\overset{c_{21}*s_{13}=6.62*s_{13}}{\rightleftharpoons}} S_{22}{}^{\dagger}; \quad \emptyset \underset{c_{24}=0.03}{\overset{c_{23}*s_{22}=20*s_{22}}{\rightleftharpoons}} S_{14};$$

$$S_{17} \xrightarrow{c_{25}=0.03} S_{16}; \quad S_{15} \xrightarrow{c_{26}=0.03} S_4; \quad S_{19} \xrightarrow{c_{27}=0.03} S_4+S_{18};$$

$$S_{28} \xrightarrow{c_{28}=0.03} S_4+S_{27}; \quad \emptyset \underset{c_{30}=0.5}{\overset{c_{29}*s_{13}=1.67*s_{13}}{\rightleftharpoons}} S_{23}; \quad \emptyset \underset{c_{32}=0.03}{\overset{c_{31}*s_{23}=20*s_{23}}{\rightleftharpoons}} S_{18};$$

$$S_{19} \xrightarrow{c_{33}=0.03} S_{15}; \quad \emptyset \underset{c_{35}=0.5}{\overset{c_{34}*s_{12}=0.00625*s_{12}}{\rightleftharpoons}} S_{25}; \quad \emptyset \underset{c_{37}=0.03}{\overset{c_{36}*s_{25}=7*s_{25}}{\rightleftharpoons}} S_4;$$

$$S_{19} \xrightarrow{c_{38}=3} S_{14}+S_{18}; \quad S_{21} \xrightarrow{c_{39}=0.7} S_{20}; \quad S_{28} \xrightarrow{c_{40}=0.5} S_{14}+S_{27};$$

$$\emptyset \underset{c_{42}=0.5}{\overset{c_{41}*s_{13}=1*s_{13}}{\rightleftharpoons}} S_{24}; \quad \emptyset \underset{c_{44}=0.03}{\overset{c_{43}*s_{24}=20*s_{24}}{\rightleftharpoons}} S_{20}; \quad S_{21} \xrightarrow{c_{45}=0.03} S_4;$$

$$S_4+S_{20} \underset{c_{47}=10000}{\overset{c_{46}=2.54}{\rightleftharpoons}} S_{21}; \quad \emptyset \underset{c_{49}=0.5}{\overset{c_{48}*s_{13}=0.43333*s_{13}}{\rightleftharpoons}} S_{26}; \quad \emptyset \underset{c_{51}=0.03}{\overset{c_{50}*s_{26}=20*s_{26}}{\rightleftharpoons}} S_{27};$$

$$S_{28} \xrightarrow{c_{52}=0.03} S_{15}; \quad S_{15}+S_{27} \underset{c_{54}=10000}{\overset{c_{53}=2.54}{\rightleftharpoons}} S_{28}; \quad S_5 \xrightarrow{c_{55}=0.03} S_1;$$

$$S_{13} \xrightarrow{c_{56}=0.03} S_1+S_{11}; \quad S_9 \xrightarrow{c_{57}=0.03} S_7; \quad S_{15} \xrightarrow{c_{58}=0.03} S_{14};$$

$$S_{19} \xrightarrow{c_{59}=0.03} S_{14}+S_{18}; \quad S_{20} \xrightarrow{c_{60}=0.03} S_{14}+S_{27}; \quad S_{21} \xrightarrow{c_{61}=0.03} S_{20}.$$

References

[1] N. Guido, X. Wang, D. Adalsteinsson, D. McMillen, J. Hasty, C. Cantor, T. Elston, and J. Collins, Nature 439, 856 (2006).
[2] M. Elowitz, A. Levine, E. Siggia, and P. Swain, Science 297, 1183 (2002).
[3] D. Endy and R. Brent, Nature 409, 391 (2001).
[4] H. Li, Y. Cao, L. Petzold, and D. Gillespie, Biotech. Prog. 24, 56 (2008).
[5] A. Arkin, J. Ross, and H. McAdams, Genetics 149, 1633 (1998).
[6] D. Adalsteinsson, D. McMillen, and T. Elston, BMC Bioinformatics 5, 24 (2004).
[7] D. Gillespie, J. Chem. Phys. 115, 1716 (2001).
[8] E. Haseltine and J. Rawlings, J. Chem. Phys. 117, 6959 (2002).
[9] D. Gillespie, J. Comput. Phys 22, 403 (1976).
[10] D. Gillespie, J. Phys. Chem. 81, 2340 (1977).
[11] Y. Cao, L. Petzold, and M. Rathinam, J. Chem. Phys. 121, 12169 (2004).
[12] A. Abdulle and T. Li, Comm. Math. Sci. 6, 845 (2008).
[13] A. Abdulle, Y. Hu, and T. Li, J. Comp. Math. 28, 195 (2010).
[14] T. Li, A. Abdulle, and W. E, Comm. Comp. Phys. 3, 295 (2008).
[15] W. E and B. Engquist, Comm. Math. Sci. 1, 87 (2003).
[16] C. Rao and A. Arkin, J. Chem. Phys. 118, 4999 (2003).
[17] Y. Cao, L. Petzold, and D. Gillespie, J. Chem. Phys. 122, 14116 (2005).
[18] W. E, D. Liu, and E. Vanden-Eijnden, J. Chem. Phys. 123, 194107 (2005).
[19] K. Takahashi, K. Kaizu, B. Hu, and M. Tomita, Bioinformatics 20, 538 (2004).
[20] H. Salis and Y. Kaznessis, J. Chem. Phys. 122, 054103 (2005).
[21] A. Bracciali, M. Brunelli, E. Cataldo, and P. Degano, BMC Bioinformatics 9, S7 (2008).

[22] K. Vasudeva and U. Bhalla, Bioinformatics 20, 78 (2004).
[23] K. Burrage, T. Tian, and P. Burrage, Prog. Biophys. Mol. Bio. 85, 217 (2004).
[24] A. Samant and D. Vlachos, J. Chem. Phys. 123, 144114 (2005).
[25] H. Salis and Y. Kaznessis, J. Chem. Phys. 123, 214106 (2005).
[26] D. Higham, SIAM Rev. 50, 347 (2008).
[27] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. 126, 224101 (2007).
[28] A. Samant, B. Ogunnaike, and D. Vlachos, BMC Bioinformatics 8, 175 (2007).
[29] E. Vanden-Eijnden, Comm. Math. Sci. 5, 495 (2007).
[30] W. Ren, Comm. Math. Sci. 5, 1027 (2007).
[31] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. 124, 44109 (2006).
[32] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. 123, 054104 (2005).
[33] T. Kepler and T. Elston, Biophys. J. 81, 3116 (2001).
[34] Y. Cao, H. Li, and L. Petzold, J. Chem. Phys. 121, 4059 (2004).
[35] J. McCollum, G. Peterson, C. Cox, M. Simpson, and N. Samatova, Comput. Biol. Chem. 30, 39 (2006).
[36] N. Sinitsyn, N. Hengartner, and I. Nemenman, Proc. Natl. Acad. Sci. 106, 10546 (2009).