R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps

Anne Auger, Philippe Chatelain, and Petros Koumoutsakos[a)]
Computational Laboratory (CoLab), ETH, Zurich CH-8092, Switzerland

(Received 14 February 2006; accepted 5 June 2006; published online 23 August 2006)

A novel algorithm is proposed for the acceleration of the exact stochastic simulation algorithm by a predefined number of reaction firings (*R-leaping*) that may occur across several reaction channels. In the present approach, the numbers of reaction firings are correlated binomial distributions and the sampling procedure is independent of any permutation of the reaction channels. This enables the algorithm to efficiently handle large systems with disparate rates, providing substantial computational savings in certain cases. Several mechanisms for controlling the accuracy and the appearance of negative species are described. The advantages and drawbacks of *R-leaping* are assessed by simulations on a number of benchmark problems and the results are discussed in comparison with established methods. © *2006 American Institute of Physics*.
[DOI: 10.1063/1.2218339]

I. INTRODUCTION

The stochastic simulation algorithm[1] (SSA) is an exact method for computing the time evolution of well-stirred chemically reacting systems. The SSA simulates every reaction event, one at a time, and it may become inefficient for systems involving species in large numbers. Over the past five years several methods have been proposed, aiming to accelerate the SSA, at the expense of its accuracy. In 2001, Gillespie[2] introduced the so-called τ-leaping algorithm that entails leaps over several reaction events during a preselected time increment τ. In τ-leaping the number of firings, during τ, for each one of the M reaction channels, is sampled from a Poisson distribution. An essential aspect of the method is the requirement that the propensity changes are small during each step (the *τ-leap condition*).

The τ-leaping algorithm can provide significant gains in simulation speed over the SSA, but as originally outlined by Gillespie[2] it is also faced with challenges such as the occurrence of negative numbers of species. A number of works have addressed this issue. Tian and Burrage[3] and Chatterjee *et al.*[4] proposed the *binomial* τ-leaping which approximates the unbounded distributions by bounded ones but it may lead to significant loss of accuracy in some cases.[5] In turn Cao *et al.*[5] modified the original τ-leaping by distinguishing the types of the involved reactions. This method and its recent variant[6] represent, to the best of our knowledge, the current state of the art in the framework of τ-leaping algorithms.

The present work proposes an acceleration of the exact SSA by reaction leaps (*R-leaping*). Here, a simulation step is characterized by a preselected number of reaction firings L. Assuming—as for τ-leaping—that the propensities remain essentially constant during one simulation step, we demonstrate that the number of firings in each reaction channel follows a binomial distribution. These M random variables are correlated and we propose a sampling procedure requiring at most $M-1$ samples. In the case of large and stiff systems, the number of random samples can be reduced significantly yielding appreciable computational savings. Following the works presented in Refs. 2, 6, and 7 we present three conditions to control the accuracy of the approximation by bounding the parameter L. The resulting approximate algorithm becomes the exact SSA when the leap size is equal to 1 and is related to the k_α-leaping algorithm.[2]

The present algorithm uses the number of reactions as the leap parameter, allowing for direct control over the appearance of negative species. Two control mechanisms are proposed, starting with a strict enforcement of positive species. Alternatively this requirement is relaxed, which allows larger leaps and the occurrence of negative species at a controlled probability. The relaxation mechanism thus offers a tunable compromise between accuracy and efficiency when some reactant populations are low.

In Sec. III, we examine the performance of *R*-leaping in benchmark problems such as decaying-dimerizing reactions, chained decaying reactions with disparate species populations, and the biochemical reaction model of LacZ/LacY of Kierzek.[8] We compare the results of *R*-leaping with several variants of τ-leaping and assess its advantages and drawbacks. We conclude with a summary in Sec. IV.

II. REACTION LEAPING

We first outline the SSA and discuss the need for accelerated schemes. Subsequently we introduce the *R*-leaping algorithm.

A. Background

1. Exact stochastic simulation algorithm

The SSA[1] is an exact method for computing the time evolution of well-stirred chemically reacting systems. We consider a system of N species S_i, $i=1,\ldots,N$, that can react

[a)]Electronic mail: petros@ethz.ch

through M reaction channels R_j, $j=1,\ldots,M$. The number of molecules of species S_i at time t is a random variable $\mathbf{X}(t)=(X_1(t),\ldots,X_N(t))$, as random molecular collisions give rise to chemical transformations described by the reaction channels. We denote as $\mathbf{x}(t)$ a realization of $\mathbf{X}(t)$. A propensity function $a_j(\mathbf{x})$ and a state-change vector $\boldsymbol{\nu}_j=(\nu_{1j},\ldots,\nu_{Nj})$ specify the dynamics of a reaction channel R_j. The quantity $a_j(\mathbf{x})dt$ represents the probability that a reaction of type R_j occurs in the infinitesimal time interval $[t,t+dt)$ and ν_{ij} denotes the change induced on the population of species S_i after a reaction R_j has occurred, i.e.,

$$X_i(t+dt)=X_i(t)+\nu_{ij}.$$

At each step in the "direct-reaction" version of the SSA, the time increment τ is sampled from an exponential distribution with parameter $1/a_0(\mathbf{x})$,

$$\tau\sim\mathcal{E}(1/a_0(\mathbf{x})),$$

where $a_0(\mathbf{x})$ is the sum of all propensity functions, i.e.,

$$a_0(\mathbf{x})=\sum_{j=1}^{M}a_j(\mathbf{x}). \tag{1}$$

The choice of which reaction R_j occurs at $t+\tau$ is sampled independently from the pointwise probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$,

$$P(j=l)=\frac{a_l(\mathbf{x})}{a_0(\mathbf{x})}\quad\text{for } l=1,\ldots,M. \tag{2}$$

The populations are then updated according to

$$\mathbf{X}(t+\tau)=\mathbf{X}(t)+\nu_j.$$

2. Approximate accelerated algorithms: τ-leaping

In order to speed up the SSA, Gillespie[2] introduced the τ-leaping method. In τ-leaping, instead of sampling a time increment and the reaction that occurred within this interval, the algorithm samples the number of reaction $K_j^{\mathcal{P}}$ of type j occurring during a specified time step of length τ. Under the *τ-leaping assumption* that the propensities $(a_j(\mathbf{x}))_{1\leqslant j\leqslant M}$ remain approximately constant over τ, the $K_j^{\mathcal{P}}$'s follow Poisson distributions and are being sampled *independently*,

$$K_j^{\mathcal{P}}\sim\mathcal{P}(a_j(\mathbf{x})\tau).$$

The update of $\mathbf{X}(t)$ is then performed according to

$$\mathbf{X}(t+\tau)=\mathbf{X}(t)+\sum_{j=1}^{M}K_j^{\mathcal{P}}\boldsymbol{\nu}_j. \tag{3}$$

The parameter τ is chosen such that the a_j's do not change significantly during one step and a number of control mechanisms for τ have been proposed.[2,6,7]

A drawback of the original τ-leaping is that the Poisson drawn $K_j^{\mathcal{P}}$'s are unbounded and may lead to negative populations. Several modifications have been introduced to circumvent this difficulty. Tian and Burrage[3] and Chatterjee *et al.*[4] proposed a binomial τ-leaping which approximates Poisson distributions with bounded binomial ones. This remedy, however, can lead to a loss of accuracy[5] and, as the parameters are chosen depending on the types of reactions, generalization of this method seems to be a challenging task.

More recently, Cao *et al.*[5] introduced a modified version of the original Poisson τ-leaping. In their approach, a reaction is tagged as critical whenever the number of firings allowed by its reactants drops below a certain threshold. A critical reaction is then handled with the standard SSA while the noncritical reactions are advanced with the regular τ-leaping.

B. *R*-leaping: The reaction leaping approach

We propose an acceleration of the exact SSA by a predefined number of firings L that may span several reaction channels. We assume that we can find a positive integer L such that, in the current state $\mathbf{x}$, the next L reactions are not likely to substantially change the value of any of the propensity functions. Under the approximation that the propensities are constant the time span τ_L of these L reactions is a gamma random variable (as in k_α-leaping[2]) with parameters $(L,1/a_0(\mathbf{x}))$,

$$\tau_L\sim\Gamma(L,1/a_0(\mathbf{x})).$$

Over this time interval, we will have K_m firings of reaction channel R_m with $\Sigma_{m=1}^{M}K_m=L$ and we demonstrate that these K_m's follow correlated binomial distributions. We can distinguish two approaches for the sampling of the K_m's, as well as a variant that improves the capability of the method for systems with disparate rates.

1. R_0-sampling: L independent pointwise variables

The first method consists in sampling the indices $(j_1,\ldots,j_L)$ of the L reactions according to Eq. (2). The total number K_m of reactions R_m occurring over a step is

$$K_m=\sum_{k=1}^{L}\delta_{j_k,m}, \tag{4}$$

where $\delta_{j_k,m}$ is the Kronecker delta. The species update is given by

$$\mathbf{X}(t+\tau_L)=\mathbf{X}(t)+\sum_{j=1}^{M}K_j\boldsymbol{\nu}_j. \tag{5}$$

Note that Eq. (5) exhibits the same structure as Eq. (3) but for the $K_j^{\mathcal{P}}$'s and K_j's which follow different distributions. In this formulation the sole gain over the SSA arises from the less frequent computation of the propensities $(a_j(\mathbf{x}))_{1\leqslant j\leqslant M}$.

Algorithm 1: R_0-sampling

In state $\mathbf{x}$ at time t,

- Initialize $k_m=0$, for $1\leqslant m\leqslant M$

FOR $i=1:L$ *DO*

- Sample j_i according to a pointwise distribution with probability $a_j(\mathbf{x})/a_0(\mathbf{x})$
- $k_{j_i}=k_{j_i}+1$

ENDDO

The R_0-sampling scales with L and, in particular, when compared with τ-leaping that scales with M, the method is inefficient for large leap sizes, $L\gg M$.

2. R_1-sampling: $\mathcal{O}(M)$ correlated binomial variables

We now propose a remedy to the drawbacks of the above algorithm and present an algorithm that scales as $\mathcal{O}(M)$. The algorithm performs the L reactions by sampling the K_m's [Eq. (4)] in a loop over the M reaction channels. This sampling is akin to the τ-leaping procedure and relies on the following theorem (see proof in the Appendix).

Theorem. *The distribution of K_1 is a binomial distribution $\mathcal{B}(L,a_1(\mathbf{x})/a_0(\mathbf{x}))$ and for every $m\in\{2,\ldots,M\}$, the conditional distribution of K_m given the event $\{(K_1,\ldots,K_{m-1})=(k_1,\ldots,k_{m-1})\}$ is*

$$\mathcal{B}\left(L-\sum_{i=1}^{m-1}k_i,\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})-\sum_{i=1}^{m-1}a_i(\mathbf{x})}\right).$$

In addition, this result is invariant under any permutation of the indices.

From this theorem we construct the following sampling procedure for the K_m's.

Algorithm 2: R_1-sampling

In state $\mathbf{x}$ at time t,

- Initialize $k_m=0$, for $1\leqslant m\leqslant M$; $S=0$, $m=1$

WHILE $S<L$ and $m<M$ *DO*

- Sampling the firings of reaction channel R_m according to the binomial distribution

$$k_m=\mathcal{B}\left(L-S,\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})-\sum_{i=1}^{m-1}a_i(\mathbf{x})}\right)$$

- $S=S+k_m$; $m=m+1$

ENDDO

IF $m=M$ *THEN*

- Set $k_M=L-S$

ENDIF

We remark that $K_M=L-\Sigma_{m=1}^{M-1}K_m$ and at most $M-1$ random numbers are needed to sample the K_m's. Note that the sampling procedure may also terminate when the L reaction firings are exhausted. Hence, the number of binomial samples needed $N_{\mathcal{B}}$ is equal to the minimum between $M-1$ and M', where M' is defined as the smallest integer such that $\Sigma_{m=1}^{M'}k_m=L$ i.e.,

$$N_{\mathcal{B}}=\min\left(M-1,\min\left\{M'\in\mathbb{N}\Big|\sum_{m=1}^{M'}k_m=L\right\}\right). \tag{6}$$

This is a notable difference from τ-leaping where exactly M Poisson samples are required. We note, however, that in R-leaping, we need to perform the additional sampling of the time step τ from a gamma distribution. This brings to M the maximum number of random variable samples for R-leaping.

3. R_1-sampling acceleration for systems with disparate rates

We may exploit the invariance of the sampling procedure under permutation of the reaction channel indices (see theorem) and the condition for an early exit from the sampling loop in algorithm 2 ($S=L$) to propose a sampling algorithm which harnesses the stiffness of the simulated system.

The invariance under permutation implies that the indices can be reordered before the sampling without changing the overall distribution of the vector $(K_1,\ldots,K_m)$. In particular, the indices can be reordered such that the largest values of the K_m's are sampled at the beginning. Hence, every p_s steps, we sort the indices such that the sequence $a_m(\mathbf{x})/a_0(\mathbf{x})$ is decreasing. This arrangement increases the probability that M' is small (compared to $M-1$).

Algorithm 3: R_1-sampling for systems with disparate rates

In state $\mathbf{x}$ at time t, step n

IF mod $(n,p_s)=0$ *THEN*

- The indices of the reactions are stored in an array I_m
- Sort the sequence $A_m=a_m(\mathbf{x})/a_0(\mathbf{x})$ in decreasing order
- Arrange the reaction indices I_m as they correspond to the sorted A_m's

ENDIF

- Initialize $k_m=0$, for $1\leqslant m\leqslant M$; $S=0$, $m=1$

WHILE $S<L$ and $m<M$ *DO*

- Sample the firings of reaction channel R_{I_m} according to the binomial distribution

$$k_{I_m}=\mathcal{B}\left(L-S,\frac{a_{I_m}(\mathbf{x})}{a_0(\mathbf{x})-\sum_{m'=1}^{m-1}a_{I_{m'}}(\mathbf{x})}\right)$$

- $S=S+k_{I_m}$, $m=m+1$

ENDDO

IF $m=M$ *THEN*

- Set $k_{I_M}=L-S$

ENDIF

The sorting of the reaction channels is an additional task that we carry out periodically during the simulation. Our implementation uses an insertion algorithm as at each time we start from a nearly sorted list.

This rearrangement can bring appreciable improvement in the case of large and stiff systems where the propensities can span several orders of magnitude. Our approach exploits the stiffness of the system with little additional cost (the sorting) and without affecting the accuracy of the method.

4. Discussion

The R_0- and R_1-sampling algorithms scale with L and M, respectively. In practice the algorithm uses R_0 whenever $L \leqslant M$ and R_1 otherwise.

We note that for R-leaping, firings may occur across all the reaction channels instead of one predetermined channel as in k_α-leaping.[2] In the latter, the channel α fires k_α times and the remaining ones advance by a τ leap. We remark that a unique feature of k_α-leaping is that we can bring the system to the next firing of a particular channel. This property is not offered by τ- or R-leaping. A generalization of k_α-leaping is made possible by the R-leaping methodology. We introduce k_A-leaping where one advances a subset $A=\{\alpha_1,\alpha_2,\ldots\}$ of the reaction channels (instead of a single one for k_α) by a predefined number of firings and then proceeds with τ leaping for the remaining ones.

By the nature of the sampled variables, R-leaping is also reminiscent of the binomial τ-leap method. There are outstanding differences, however. In the R_1-sampling procedure, the K_m's are not independent, but correlated as discussed above. In addition, binomial τ-leaping approximates unbounded distributions with bounded ones, thus introducing a bias. The binomial distributions of R-leaping are exact under the approximation of constant propensities.

C. Control of L

The number L of reactions simulated at each time step determines the degree of approximation of the algorithm. The R-leaping algorithm relies on the assumption that the propensity functions do not change significantly during each step. This assumption is the same as in τ-leaping algorithms where several conditions for choosing τ have been proposed.[6,7] We follow these works and derive three corresponding conditions for R-leaping. The first condition is based on the work presented in Ref. 7, where the change in propensities is bounded by a fraction ϵ of the sum $a_0(\mathbf{x})$. This condition was subsequently refined in Ref. 6, where it is proposed to bound the relative change in the propensities. A further refinement reduces the computational cost, from M^2 to M, by bounding the change in the molecular populations.

1. Bounding the propensity changes by a fraction of $a_0(\mathbf{x})$

Following Ref. 7, we determine L such that during $[t,t+\tau_L)$, the change for each propensity function is bounded by $\epsilon a_0(\mathbf{x})$,

$$|a_j(\mathbf{X}(t+\tau_L))-a_j(\mathbf{x})| \leqslant \epsilon a_0(\mathbf{x}) \quad \text{for } j=1,\ldots,M,$$

which may be expressed using the definition of K_m [Eq. (4)] as

$$\left| a_j\left(\mathbf{x}+\sum_{m=1}^{M} K_m \boldsymbol{\nu}_m\right) - a_j(\mathbf{x}) \right| \leqslant \epsilon a_0(\mathbf{x}) \quad \text{for } j=1,\ldots,M. \tag{7}$$

The interpretation of this inequality can lead to different control mechanisms. In the context of τ-leaping, Gillespie[2] proposed the requirement that the expected value of the left-hand side of Eq. (7) is smaller than $\epsilon a_0(\mathbf{x})$. This condition was further enhanced in Ref. 7 requiring that, in addition to the expected value, the standard deviation of the left-hand side be smaller than $\epsilon a_0(\mathbf{x})$,

$$\left| \mathbb{E}\left(a_j\left(\mathbf{x}+\sum_{m=1}^{M} K_m \boldsymbol{\nu}_m\right) - a_j(\mathbf{x}) \right) \right| \leqslant \epsilon a_0(\mathbf{x}), \tag{8}$$

$$\mathrm{var}\left(a_j\left(\mathbf{x}+\sum_{m=1}^{M} K_m \boldsymbol{\nu}_m\right) - a_j(\mathbf{x}) \right) \leqslant (\epsilon a_0(\mathbf{x}))^2. \tag{9}$$

For R-leaping, those conditions are satisfied at first order under the following condition.

R-leaping condition 1.

$$L \leqslant a_0(\mathbf{x}) \min_{j=1,\ldots,M} \left\{ \frac{\epsilon a_0(\mathbf{x})}{|\mu_j(\mathbf{x})|}, \frac{\epsilon^2 a_0^2(\mathbf{x})}{\sigma_j^2(\mathbf{x}) - [\mu_j^2(\mathbf{x})/a_0(\mathbf{x})]} \right\}, \tag{10}$$

where

$$\mu_j(\mathbf{x}) = \sum_{m=1}^{M} f_{jm}(\mathbf{x}) a_m(\mathbf{x}), \tag{11}$$

$$\sigma_j^2(\mathbf{x}) = \sum_{m=1}^{M} f_{jm}^2(\mathbf{x}) a_m(\mathbf{x}), \tag{12}$$

with

$$f_{jm}(\mathbf{x}) = \sum_{l=1}^{N} \frac{\partial a_j(\mathbf{x})}{\partial x_l} \nu_{lm}. \tag{13}$$

Proof. The propensity change in the left-hand side of Eq. (7) can be approximated to first order by $\Sigma_{l=1}^{N} (\Sigma_{m=1}^{M} K_m \nu_{lm}) \times [\partial a_j(\mathbf{x})/\partial x_l]$. Using the definition of f_{jm} this can be simplified further to

$$a_j\left(\mathbf{x}+\sum_{m=1}^{M} K_m \boldsymbol{\nu}_m\right) - a_j(\mathbf{x}) \approx \sum_{m=1}^{M} f_{jm}(\mathbf{x}) K_m. \tag{14}$$

The problem is now reduced to deriving the expectation and the variance of the right-hand side. The expectation of a binomial random variable $\mathcal{B}(n,p)$ is np. We then have

$$\mathbb{E}\left(\sum_{m=1}^{M} f_{jm}(\mathbf{x}) K_m \right) = \sum_{m=1}^{M} f_{jm}(\mathbf{x}) L \frac{a_m(\mathbf{x})}{a_0(\mathbf{x})} = L \frac{\mu_j(\mathbf{x})}{a_0(\mathbf{x})}. \tag{15}$$

The variance of the right-hand side in Eq. (14) reads

$$\mathrm{var}\left(\sum_{m=1}^{M} f_{jm}(\mathbf{x}) K_m \right) = \sum_{m=1}^{M} f_{jm}^2(\mathbf{x}) \mathrm{var}(K_m) + \sum_{\substack{(m,m')=1 \\ m \neq m'}}^{M} f_{jm}(\mathbf{x}) f_{jm'}(\mathbf{x}) \mathrm{cov}(K_m, K_{m'}). \tag{16}$$

The variance $\mathrm{var}(K_m)$ is that of a binomial random variable $\mathcal{B}(n,p)$ and it is equal to $np(1-p)$, while the remaining nontrivial terms are the covariances $\mathrm{cov}(K_m,K_{m'})$. Note that

these covariance terms do not vanish, as in the τ-leaping,[2] because of the correlated K_m 's.

$$\mathrm{cov}(K_m,K_{m'}) = \mathbb{E}\left(\sum_{k=1}^{L}\mathbf{1}_{j_k=m}\sum_{k'=1}^{L}\mathbf{1}_{j'_{k'}=m'}\right) - \mathbb{E}(K_m)\mathbb{E}(K_{m'})$$

$$= \mathbb{E}\left(\sum_{k,k'=1}^{L}\mathbf{1}_{j_k=m}\mathbf{1}_{j'_{k'}=m'}\right) - \mathbb{E}(K_m)\mathbb{E}(K_{m'}).$$

For $k=k'$, the product $\mathbf{1}_{j_k=m'}\mathbf{1}_{j'_{k'}=m'}$ is zero since $m\neq m'$ and the previous equation reduces to

$$\mathrm{cov}(K_m,K_{m'}) = \sum_{\substack{(k,k')=1\\ k\neq k'}}^{L}\mathbb{E}(\mathbf{1}_{j_k=m})\mathbb{E}(\mathbf{1}_{j'_{k'}=m'}) - \mathbb{E}(K_m)\mathbb{E}(K_{m'})$$

$$= (L^2-L)\frac{a_m(\mathbf{x})a_{m'}(\mathbf{x})}{a_0^2(\mathbf{x})} - L^2\frac{a_m(\mathbf{x})a_{m'}(\mathbf{x})}{a_0^2(\mathbf{x})}$$

$$= -L\frac{a_m(\mathbf{x})a_{m'}(\mathbf{x})}{a_0^2(\mathbf{x})}.$$

Finally, we obtain that

$$\mathrm{var}\left(\sum_{m=1}^{M}f_{jm}(\mathbf{x})K_m\right) = L\left(\sum_{m=1}^{M}f_{jm}^2(\mathbf{x})\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})}\left(1-\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})}\right) - \sum_{\substack{(m,m')=1\\ m\neq m'}}^{M}f_{jm}(\mathbf{x})f_{jm'}(\mathbf{x})\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})}\frac{a_{m'}(\mathbf{x})}{a_0(\mathbf{x})}\right)$$

or

$$\mathrm{var}\left(\sum_{m=1}^{M}f_{jm}(\mathbf{x})K_m\right) = L\left(\frac{\sigma_j^2(\mathbf{x})}{a_0(\mathbf{x})} - \frac{\mu_j^2(\mathbf{x})}{a_0^2(\mathbf{x})}\right). \tag{17}$$

Substituting Eqs. (15) and (17), respectively, into Eqs. (8) and (9) yields the control condition on L. □

We remark that the bounds in Eq. (10) differ slightly from the ones of the τ-leaping approach[7]

$$\tau \leqslant \min_{j=1,\ldots,M}\left\{\frac{\epsilon a_0(\mathbf{x})}{|\mu_j(\mathbf{x})|}, \frac{\epsilon^2 a_0^2(\mathbf{x})}{\sigma_j^2(\mathbf{x})}\right\}, \tag{18}$$

by the factor $a_0(\mathbf{x})$, a reaction per unit time rate and by the smaller denominator of the variance criterion. The variance per unit leap size [Eq. (17)] is smaller in the case of *R*-leaping due to the correlations between the K_m variables. In practice, however, this difference does not lead to any noticeable increase of the leap size. We observe the same expected number of time steps for τ-leaping and *R*-leaping for a given ϵ. We note that this additional term has a marginal effect on the computational cost of the method.

2. Bounding the relative change in the propensities

Following Ref. 6, we derive a bound on L where instead of Eq. (7) the relative change in propensities is considered

$$\left|a_j\left(\mathbf{x}+\sum_{m=1}^{M}K_m\boldsymbol{\nu}_m\right) - a_j(\mathbf{x})\right| \leqslant \max\{\epsilon a_j(\mathbf{x}), c_j\} \tag{19}$$

for $j=1,\ldots .M$.

The derivation of Eq. (19) uses the same arguments as in the development of condition 1 [Eq. (10)]. This leads to the following condition:

R-leaping condition 2.

$$L \leqslant a_0(\mathbf{x})\min_{j=1,\ldots,M}\left\{\frac{\max\{\epsilon a_j(\mathbf{x}),c_j\}}{|\mu_j(\mathbf{x})|}, \frac{(\max\{\epsilon a_j(\mathbf{x}),c_j\})^2}{\sigma_j^2(\mathbf{x})-[\mu_j^2(\mathbf{x})/a_0(\mathbf{x})]}\right\}. \tag{20}$$

The computational cost of conditions 1 and 2 scales as M^2 because of the calculation of the tensor f_{ij} necessary to determine the denominators in Eqs. (10) and (20).

3. Bounding the relative change in the molecular population

In order to make the condition 2 computationally efficient, Cao *et al.*[6] proposed a new selection procedure for the τ-leaping which bounds the relative change in the molecular populations. Equation (19) is now replaced by

$$|X_i(t+\tau)-x_i| \leqslant \max\left\{\epsilon\frac{x_i}{g_i}, 1\right\}, \quad \forall\, i\in I_{\mathrm{rs}}, \tag{21}$$

where $\mathbf{x}=(x_1,\ldots,x_N)$ and the set I_{rs} denotes the indices of all reactant species. The factor g_i takes into account the highest order of reaction (abbreviated as HOR) in which species S_i appears as a reactant.

(i) If HOR(i)=1, set $g_i=1$.
(ii) If HOR(i)=2, set $g_i=2$, except if any second-order reaction requires two S_i molecules, set instead $g_i=(2+1/(x_i-1))$.
(iii) If HOR(i)=3, set $g_i=3$, except if some third-order reaction requires two S_i molecules, set instead $g_i=\frac{3}{2}(2+1/(x_i-1))$ except if some third-order reaction requires three S_i molecules, set instead $g_i=(3+1/(x_i-1)+2/(x_i-2))$.

This leads to the following condition.

R-leaping condition 3.

$$L \leqslant a_0(\mathbf{x})\min_{i\in I_{\mathrm{rs}}}\left\{\frac{\max\{\epsilon x_i/g_i,1\}}{|\hat{\mu}_i(\mathbf{x})|}, \frac{(\max\{\epsilon x_i/g_i,1\})^2}{\hat{\sigma}_i^2(\mathbf{x})-[\hat{\mu}_i^2(\mathbf{x})/a_0(\mathbf{x})]}\right\}, \tag{22}$$

where

$$\hat{\mu}_i(\mathbf{x}) = \sum_{j=1}^{M}\nu_{ij}a_j(\mathbf{x}), \quad \forall\, i\in I_{\mathrm{rs}} \tag{23}$$

and

$$\hat{\sigma}_i^2(\mathbf{x}) = \sum_{j=1}^{M}\nu_{ij}^2a_j(\mathbf{x}), \quad \forall\, i\in I_{\mathrm{rs}}. \tag{24}$$

The conditions 2 and 3 [Eqs. (20) and (22)] interpret more accurately the *leap condition* than condition 1 [Eq. (10)] as they limit the relative changes in the propensity functions. Furthermore, condition 3 scales as M, resulting in significant computational savings and usually would be the best choice.

D. Control of negative species

Approximate algorithms offer a computationally efficient method over the SSA but are often hindered by the appearance of negative species. In this section we complement R-leaping with bounding conditions for L in order to control the appearance of negative species.

We denote as L_j the maximum number of reaction R_j that can occur before running out of any reactants that participate in R_j,

$$L_j = \min_{i\in[1,N]:\nu_{ij}<0}\left[\frac{x_i}{|\nu_{ij}|}\right]. \tag{25}$$

If we neglect the fact that reactions may be coupled by having common reactants or by producing each other's reactants, we can derive a simple and strict bound: negative species are avoided by bounding each sampled K_j by L_j. As the total number of reactions is equal to L, one way to achieve this condition is by requiring that each K_j is not larger than L. In particular, the condition $K_j \leqslant L_j$ is satisfied for each j if L is bounded by L_j for all j, i.e.,

$$L \leqslant L_j \quad \text{for all } j, \tag{26}$$

which can be rewritten as

$$L \leqslant \min_{j\in[1,M]:a_j>0} L_j. \tag{27}$$

This scheme is straightforward to implement and ensures the nonnegativity of the species since the total number of reactions fired is smaller than the smallest number of reactions that can introduce negative species. The bound it imposes, however, on the total number of reactions is rather restrictive. Indeed, the expected value of K_j is $L[a_j(\mathbf{x})/a_0(\mathbf{x})]$. This can be considerably smaller than L for reactions that are unlikely to be fired off, i.e., for reactions for which $a_j(\mathbf{x})/a_0(\mathbf{x})$ is small. Therefore, imposing $L \leqslant L_j$ in such cases heavily restricts L and considerably slows down the computation of systems including a slow reaction with scarce reactants.

To circumvent this drawback we propose a relaxation of the condition in Eq. (27). A bound allowing negative species, with a small probability, is introduced. In other words, $K_j \leqslant L_j$ will no longer be satisfied with probability one. If K_j was always equal to its expected value $L[a_j(\mathbf{x})/a_0(\mathbf{x})]$ then a condition could be

$$L\frac{a_j(\mathbf{x})}{a_0(\mathbf{x})} \leqslant L_j \quad \text{for all } j,$$

or equivalently

$$L \leqslant \frac{a_0(\mathbf{x})}{a_j(\mathbf{x})}L_j \quad \text{for all } j. \tag{28}$$

Note, however, that this condition may result in large numbers of negative species as several samples will be above the expected value of K_j. Alternatively, we consider the following tradeoff between conditions (27) and (28):

$$L \leqslant (1-\theta)L_j + \theta\frac{a_0(\mathbf{x})}{a_j(\mathbf{x})}L_j \text{ for all } j,$$

or equivalently

$$L \leqslant \min_{j=1,\ldots,M}\left(1-\theta\left(1-\frac{a_0(\mathbf{x})}{a_j(\mathbf{x})}\right)\right)L_j. \tag{29}$$

For $\theta=0$, we recover the conservative scheme of Eq. (27), while for larger values, the bound allows to fire more reactions than allowed by the reactants. We emphasize that for a fixed value of θ, the probability of introducing negative species varies depending on the vector $(a_1(\mathbf{x})/a_0(\mathbf{x}),\ldots,a_M(\mathbf{x})/a_0(\mathbf{x}))$ [cf. Eq. (29)]. Prescribing this probability in advance requires the inversion of the cumulative density function of the K_m's. This is an expensive computational task even for low M. The scheme of Eq. (29) can thus be considered as a crude control mechanism over this probability.

In addition to the control scheme provided by Eq. (29), we reject vectors K_m that result in negative species. We note here that τ-leaping algorithms also recur to this procedure.[5] A rejection modifies the effective distribution law for K_m and affects the accuracy of our scheme. As θ offers a limited control over the probability of negative species, we divide the leap length L by 2 when a rejection occurs. This additional caution guarantees that the probability of generating negative species decreases at each attempt. Moreover, the probability of generating valid K_m's in a finite number $(\mathcal{O}(\log_2(L)))$ of attempts is one.

This property is not verified for the Poisson τ-leaping or the modified Poisson τ-leaping approaches[5] since the Poisson distribution is unbounded.

E. *R*-leaping: Summary

One time step of the R-leaping algorithm is summarized below.

In state $\mathbf{x}$ at time t, step n.

(1) Compute a candidate reaction leap L' controlling the accuracy with a chosen value for ϵ using either of the following.

- R-leaping condition 1,

$$L' = a_0(\mathbf{x}) \min_{j=1,\ldots,M}\left\{\frac{\epsilon a_0(\mathbf{x})}{|\mu_j(\mathbf{x})|}, \frac{\epsilon^2 a_0^2(\mathbf{x})}{\sigma_j^2(\mathbf{x}) - [\mu_j^2(\mathbf{x})/a_0(\mathbf{x})]}\right\}.$$

- R-leaping condition 2,

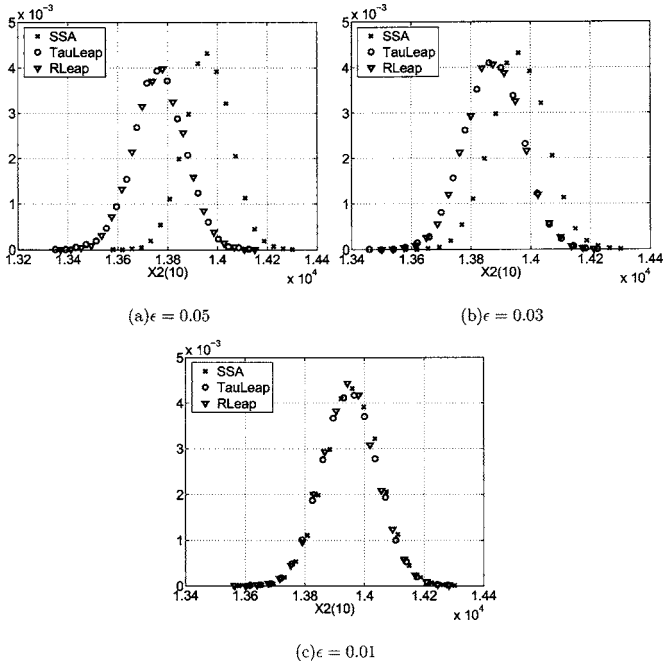


FIG. 1. Decaying-dimerizing reaction set [Eq. (30)]: histogram of the distribution of the species X_2 for $t=10$ for several values of the accuracy parameter ϵ.

$$L' = a_0(\mathbf{x}) \min_{j=1,\ldots,M} \left\{ \frac{\max\{\epsilon a_j(\mathbf{x}), c_j\}}{|\mu_j(\mathbf{x})|}, \frac{(\max\{\epsilon a_j(\mathbf{x}), c_j\})^2}{\sigma_j^2(\mathbf{x}) - [\mu_j^2(\mathbf{x})/a_0(\mathbf{x})]} \right\}.$$

- *R*-leaping condition 3,

$$L' = a_0(\mathbf{x}) \min_{i \in I_{\mathrm{rs}}} \left\{ \frac{\max\{\epsilon x_i/g_i, 1\}}{|\hat{\mu}_i(\mathbf{x})|}, \frac{(\max\{\epsilon x_i/g_i, 1\})^2}{\hat{\sigma}_i^2(\mathbf{x}) - [\hat{\mu}_i^2(\mathbf{x})/a_0(\mathbf{x})]} \right\}.$$

(2) Compute a candidate reaction leap L'' controlling the occurrence of negative species with a chosen value for θ.

$$L'' = \min_{j=1,\ldots,M} \left(1 - \theta \left(1 - \frac{a_0(\mathbf{x})}{a_j(\mathbf{x})} \right) \right) L_j,$$

where

$$L_j = \min_{i=1,\ldots,N:\nu_{ij}<0} \left\lfloor \frac{x_i}{|\nu_{ij}|} \right\rfloor.$$

(3) Set $L = \min(L', L'')$.

(4) Sample $(K_1, \ldots, K_M) = (k_1, \ldots, k_M)$ according to the following.

- If $L < M$, R_0-*sampling* (algorithm 1).
- Otherwise, R_1-*sampling* (algorithm 2).

(5) If $\mathrm{mod}(n, p_s) = 0$, reorder the indices such that $(a_m(\mathbf{x})/a_0(\mathbf{x}))_{1 \le m \le M}$ is in decreasing order by applying an insertion algorithm.

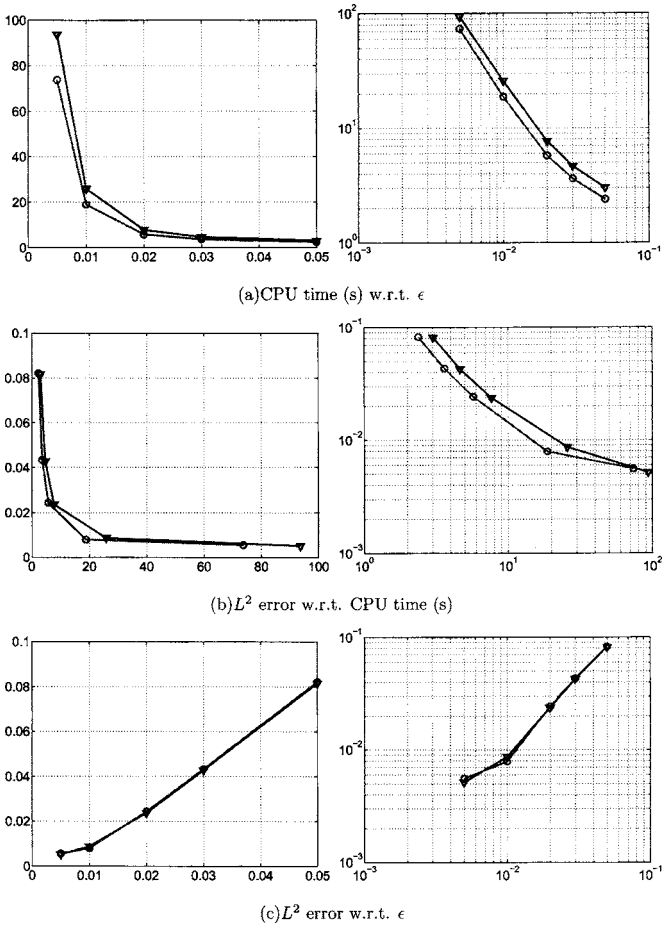


FIG. 2. Decaying-dimerizing reaction set [Eq. (30)]: Cross plots of the computation time, L^2 histogram error and control parameter ϵ for τ-leaping ("○" markers) and *R*-leaping ("▽" markers).

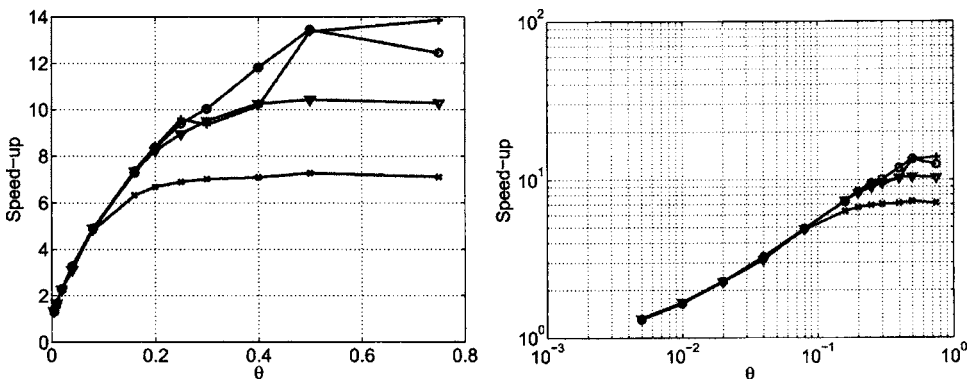


FIG. 3. Low species reactions [Eq. (32)] speedup of *R*-leaping relative to the SSA vs the relaxation parameter θ for $\epsilon = 0.005$ (×), 0.01 (▽), 0.03 (○), and 0.05 (+) in linear and logarithmic coordinates.

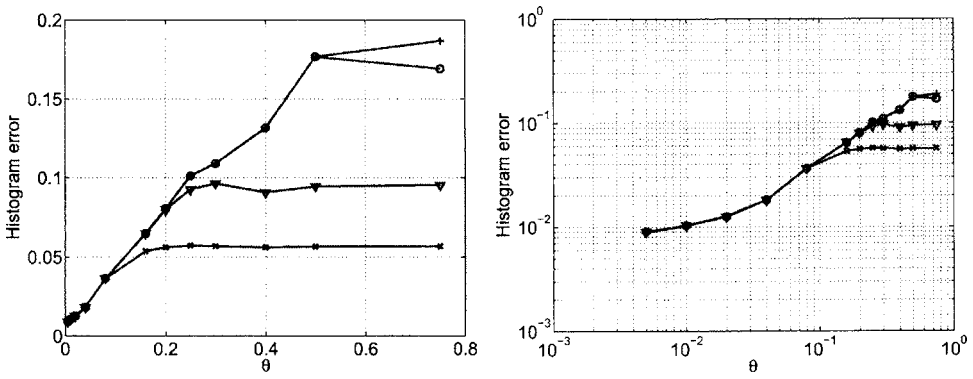


FIG. 4. Low species reactions [Eq. (32)]: L^2 error on the histogram of species 1 for the modified R-leaping relative to the SSA vs the relaxation parameter θ for $\epsilon=0.005$ ($\times$), 0.01 (∇), 0.03 ($\bigcirc$), and 0.05 ($+$) in linear and logarithmic coordinates.

(6) If there is a negative component in $\mathbf{x}+\sum_{j=1}^{M}k_j\boldsymbol{\nu}_j$ reduce L by half return to step (4), otherwise,

$$\mathbf{x}=\mathbf{x}+\sum_{j=1}^{M}k_j\boldsymbol{\nu}_j.$$

(7) Sample τ_L according to a gamma distribution $\Gamma(L,1/a_0(\mathbf{x}))$ and do

$$t=t+\tau_L.$$

III. NUMERICAL SIMULATIONS

A. Accuracy of *R*-leaping

We first test the accuracy of the R-leaping algorithm and its dependence on the parameter ϵ as prescribed in Eq. (10). Following similar tests as for τ-leaping,[2,7] we consider the decaying-dimerizing reaction, a problem that does not involve scarce reactants,

$$S_1 \xrightarrow{c_1} 0, \quad S_1+S_1 \xrightarrow{c_2} S_2,$$
$$S_2 \xrightarrow{c_3} S_1+S_1, \quad S_2 \xrightarrow{c_4} S_3, \tag{30}$$

where

$$c_1=1, \quad c_2=0.002, \quad c_3=0.5, \quad c_4=0.04. \tag{31}$$

The initial conditions $X_1(0)=4150$, $X_2(0)=39\,565$, and $X_3(0)=3445$ are imposed. The R-leaping condition 1 is used to select L' and we set L'' to infinity to test solely the dependency on ϵ. Step (5) of the R-leaping algorithm is omitted (no reordering of the indices is used). Results were obtained using 10 000 samples for ϵ values of 0.01, 0.03, and 0.05.

We compare the histogram of the distribution of $X_2(t=10)$ for the SSA, τ-leaping (with condition of Ref. 7 to choose τ), and R-leaping. This particular species displays a nonzero bias for a large ϵ, as shown in Fig. 1(a). We observe that, in a similar fashion to τ-leaping,[7] the accuracy of the R-leaping algorithm increases for smaller values of ϵ. The accuracy for a given ϵ and the convergence with respect to decreasing ϵ appear to be the same for the τ-leaping and the R-leaping. This is also confirmed in Fig. 2 where we consider the relationships between the L^2 error of the histograms, the computation time (on a 2.53 GHz Pentium 4 workstation), and the parameter ϵ. The convergence rate and the complexity of both leaping algorithms have the same dependency on ϵ. We also observe that R-leaping is slightly more expensive than τ-leaping.

B. Control of negative species

We now investigate the control of negative species using Eqs. (27) and (29) on a system where some of the reactants appear in small numbers.

$$S_1 \xrightarrow{c_1} S_2, \quad S_2 \xrightarrow{c_2} S_3, \tag{32}$$

with $c_1=10$ and $c_2=0.1$. The initial populations are $X_1(0)=9$, $X_2(0)=2\times10^4$, and $X_3(0)=0$. This system has been investigated in Ref. 5 for the binomial, the original Poisson, and modified Poisson τ-leaping algorithms. The simulations were run for 10^5 samples from time 0–0.1 for the SSA and the R-leaping algorithm for a range of ϵ and θ. The step (5) of the R-leaping algorithm is omitted and the R-leaping condition 1 is used to select L'.

The speedup defined as the ratio of the CPU time of the SSA and the CPU time for the accelerated scheme is shown in Figs. 3 and 4, along with the L^2 error versus the relaxation parameter θ for several values of ϵ. We observe that the parameter θ allows us to tune the compromise between speed and accuracy. On one hand, θ controls the probability of introducing negative species or equivalently the frequency of rejections which in turn affects accuracy. On the other hand, it allows to take longer leaps. For small values of θ and a fixed ϵ, the θ bounds on the leap become gradually more stringent than the ϵ ones. Inversely, for large values of θ, the accuracy condition is the tighter one; the speedup and the error become independent of θ. For the sake of completeness, we present the combined influences of ϵ and θ in the contour plots of Fig. 5.

In Table I, we compare a subset of these results to the results in Ref. 5. We note that the R-leaping algorithm does some excess work in this case. Contrary to τ-leaping, the time step is nondeterministic; it cannot bring the system exactly to $t=0.1$ but has to go beyond. As anticipated, for $\theta=0$, the bound of Eq. (27) is quite restrictive and forces the R-leaping algorithm to do low-L leaps. For $\theta=0.4$, we match the speedup of the original Poisson and maintain a better accuracy. For $\theta=0.04-0.08$, we match the speedup of the modified Poisson algorithm and the resulting distribution is close to the exact one given by the SSA, as can be seen in Fig. 6.

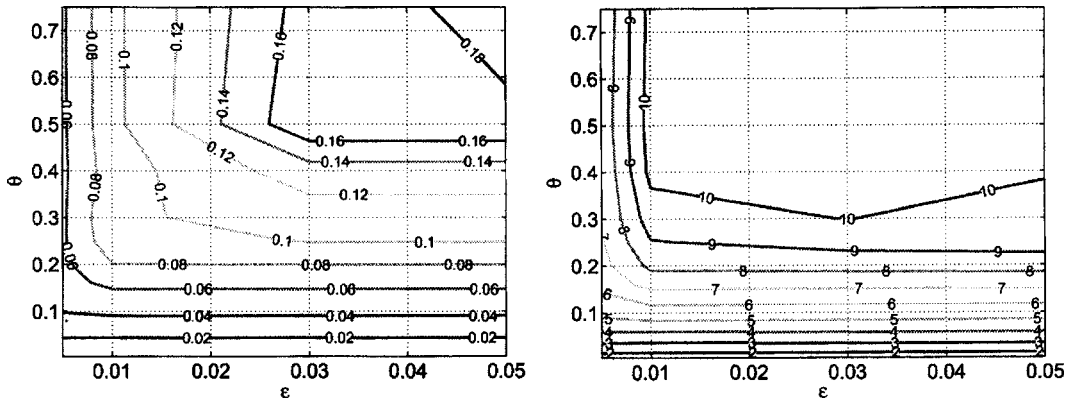


FIG. 5. Low species reactions [Eq. (32)]: contours of the speedup and L^2 error of *R*-leaping relative to the SSA vs the relaxation parameters ϵ and θ.

C. LacZ/LacY model

We then consider the LacZ/LacY reaction model of Kierzek[8] with the reactions and reaction rates presented in Table II. This system helps demonstrate the capabilities of *R*-leaping in handling moderately large and stiff systems of reactions.

The initial conditions of this system are the following. All the species are initially at 0 except PLac=1 and the species RNAP and ribosome which are drawn from random pools with normal distributions during the whole simulation: RNAP$\sim\mathcal{N}(350,35^2)$ and ribosome$\sim\mathcal{N}(35,3.5^2)$. The reaction volume increases linearly with time and the propensities of the reactions with an order larger than 1 have to be rescaled accordingly.[3,8]

We carry out tests similar to those in Ref. 3. Table III shows the speedups relative to the SSA for 100 simulations of the first cell generation ($t=2100$). The second test taken from Ref. 3 considers the trajectories of several species. Runs span the time interval of 300–330 and have an initial condition given by one SSA simulation. Figure 7 presents the mean trajectories of four species and their standard deviation for the SSA and the *R*-leaping method with $\epsilon=0.03$, condition 1 for the selection of L' and $\theta=0.1$. The *R*-leaping trajectories are indistinguishable from the SSA ones.

To test the effect of the sorting of the $a_j(\mathbf{x})/a_0(\mathbf{x})$'s on the number of random numbers sampled at each time step, we run the *R*-leaping algorithm using condition 1 for the selection of L' for one cell generation ($t=0$ to time $t=2100$) with $\theta=0.1$ and $\epsilon=0.03$. We apply the sorting [step (5)] every 10 000 time steps (run spans $\sim 3\times 10^6$ time steps). Figure 8 shows the evolutions of the number of binomial samples $N_{\mathcal{B}}$ [Eq. (6)], with and without the sorting. Without the sorting, $N_{\mathcal{B}}$ is always equal to $M-1=21$, which is due to the fact that R_{22} is a fast reaction. The average value of $N_{\mathcal{B}}$ drops to 4.1 when the sorting is applied. From $t=0$ to $t=275$, the R_0-sampling algorithm is used and therefore $N_{\mathcal{B}}$ is equal to 0 on the plots. We recall that in the case of the τ-leaping, 22 Poisson samples are always needed. We appraise the effect of $N_{\mathcal{B}}$ on the overall timing in the next experiment.

We now consider the performance of *R*- and τ-leaping with a leap condition on the relative population changes [leap condition 3 for L' and Eq. (33) of Ref. 6 for τ]. We carry out the same experiment as in Ref. 6. We first run a single SSA sample from time $t=0$ to time $t=1000$. The state at $t=1000$ is then used as an initial condition for runs of the SSA, *R*- and τ-leaping up to $t=1001$. The modified Poisson τ-leaping is run for several values of ϵ with $n_c=10$, as in Ref. 6. Several values of ϵ and θ are used for the *R*-leaping; the reaction sorting is also turned on or off. The evolution of the histogram distance error for LacZlactose with the CPU time is shown in Fig. 9. As observed in Sec. III B, θ and ϵ both govern the accuracy of the algorithm with θ becoming the dominant parameter when scarce reactants are involved. Therefore, on the plots of Fig. 9 each *R*-leaping curve is for a fixed ϵ while θ goes from 0.025 to 0.75. *R*-leaping without permutations displays a behavior similar to the modified τ-leaping. For a fixed error, it is faster than the modified τ by 10%–20%. In a second set of runs, the reaction indices are sorted according to their propensities [step (5)] every $p_s=100$ time steps. The gain from the sorting is quite consequent in this case and *R*-leap becomes twice as fast as the modified τ-leap. This large difference between the unsorted and the sorted cases is explained by the relatively light cost of the leap condition 3. The savings in the binomial samplings (from 21 to 4) thus affect the total cost greatly.

TABLE I. Low species reactions [Eq. (32)]: speedup relative to the SSA and average number of step per run for the τ- and *R*-leaping alogrithms with $\epsilon=0.03$. The values for the original Poisson (OP) and modified Poisson (MP) are taken from (Ref. 5).

	OP	MP	*R*-leaping			
θ			0.0	0.04	0.08	0.4
Speedup	7.84	3.46	1.36	3.36	4.90	11.81
Avg Steps Per Run	2.0	6.7	44.6	17	10.4	3.1
Avg Rej. Per Run	0.13	0.0	0	$5E^{-4}$	$1.4E^{-3}$	$5.7E^{-2}$

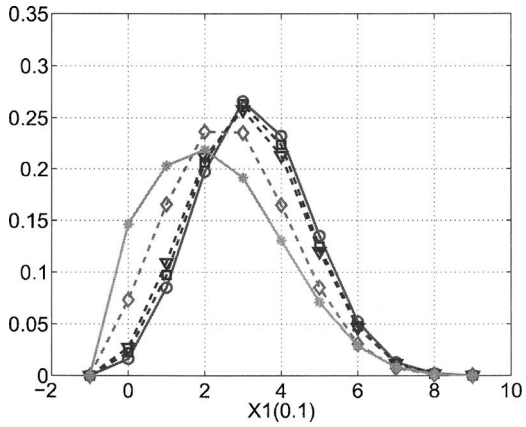


FIG. 6. Low species reactions [Eq. (32)]: histogram of species 1 for the SSA (○), τ-leaping (∗), and R-leaping for $\theta=0.04$ (□), 0.08 (▽), and 0.4 (◇).

IV. CONCLUSION

In this paper we have introduced the R-leaping algorithm for the acceleration of the SSA by predefined numbers of reaction firings (L) that may span several reaction channels.

The key feature of R-leaping relies on our demonstration that the number of reaction firings is correlated binomial distributions and that they can be sampled independently of any permutation of the reaction channels. By periodically applying a sorting of the reactions, such that the propensities are in decreasing order, we minimize the number of necessary binomial samples without affecting the accuracy of the algorithm.

Similarly to the τ-leaping approaches, the algorithm assumes that the reaction propensities remain constant over several firings. The control of this approximation for R-leaping relies on related procedures proposed for its τ-leaping counterparts.[2,6,7] We propose three leap conditions for L: one controlling the absolute change in the propensities, another one controlling the relative change, and a third one controlling the relative change in the molecular population.

Maintaining performance, while at the same time enforcing positive molecular populations, is a challenging task for approximate algorithms, in particular, for systems with scarce reactants. We introduce two bounding mechanisms over L in an effort to control the occurrence of negative species. One mechanism strictly enforces positive populations albeit at a large computational cost, while the second offers a tunable compromise between efficiency and accuracy.

We present validations of the R-leaping algorithm on benchmark chemical and biochemical systems. For the moderately large LacZ/LacY system, we observe efficiency improvements up to a factor of 2 with respect to the modified Poisson τ-leaping. Most of R-leaping's advantage originates from the efficient reaction sampling procedure which exploits the size and the stiffness of the simulated system.

Future work directions involve the extension of the R-leaping algorithm to systems with multiple disparate rates and its inclusion in spatial models of biochemical systems.

APPENDIX A: ADDITIONAL METHOD PROOFS

Theorem. *The distribution of K_1 is a binomial distribution $\mathcal{B}(L,a_1(\mathbf{x})/a_0(\mathbf{x}))$ and for every $m\in\{2,\ldots,M\}$, the conditional distribution of K_m given the event $\{(K_1,\ldots,K_{m-1})=(k_1,\ldots,k_{m-1})\}$ is*

$$\mathcal{B}\left(L-\sum_{i=1}^{m-1}k_i,\frac{a_m(\mathbf{x})}{a_0(\mathbf{x})-\sum_{i=1}^{m-1}a_i(\mathbf{x})}\right).$$

In addition, this result is invariant under any permutation of the indices.

To prove this theorem we need the two following lemmas.

Lemma 1. *For every $m=1,\ldots,M$, the random variables K_m's follow a binomial distribution with parameters L and $a_j(\mathbf{x})/a_0(\mathbf{x})$, i.e.,*

$$K_m\sim\mathcal{B}(L,a_j(\mathbf{x})/a_0(\mathbf{x})).$$

Proof. Assume that the indices $(j_1,\ldots,j_L)$ such as defined in Eq. (2) have been drawn from pointwise distributions and fix an integer $m\in[1,M]$. Consider now for each index $(j_1,\ldots,j_L)$ the following events: either it is equal to m

TABLE II. LacZ/LacY model [Kierzek (Ref. 8)]: reaction channels and rates.

	Reaction channel	Reaction rate
R_1	PLac+RNAP→PLacRNAP	0.17
R_2	PLacRNAP→PLac+RNAP	10
R_3	PLacRNAP→TrLacZ1	1
R_4	TrLacZ1→RbsLacZ+PLac+TrLacZ2	1
R_5	TrLacZ2→TrLacY2	0.015
R_6	TrLacY1→RbsLacY+TrLacY2	1
R_7	TrLacY2→RNAP	0.36
R_8	Ribosome+RbsLacZ→RbsribosomeLacZ	0.17
R_9	Ribosome+RbsLacY→RbsribosomeLacY	0.17
R_{10}	RbsribosomeLacZ→Ribosome+RbsLacZ	0.45
R_{11}	RbsribosomeLacY→Ribosome+RbsLacY	0.45
R_{12}	RbsribosomeLacZ→TrRbsLacZ+RbsLacZ	0.4
R_{13}	RbsribsomeLacY→TrRbsLacY+RbsLacY	0.4
R_{14}	TrRbsLacZ→LacZ	0.015
R_{15}	TrRbsLacZ→LacY	0.036
R_{16}	LacZ→dgrLacZ	6.42×10^{-5}
R_{17}	LacY→dgrLacY	6.42×10^{-5}
R_{18}	RbsLacZ→dgrLacY	0.3
R_{19}	RbsLacZ→dgrRbsLacY	0.3
R_{20}	LacZ+lactose→LacZlactose	9.52×10^{-5}
R_{21}	LacZlactose→product+LacZ	431
R_{22}	LacY→lactose+LacY	14

TABLE III. LacZ/LacY model [Kierzek (Ref. 8)]: speedup relative to the SSA (CPU time SSA/CPU time) for the computation of the first generation ($t=2100$).

Method	Speedup
Binomial τ leap ($\epsilon=0.05$)	70.53*
Binomial τ leap ($\epsilon=0.03$)	56.50*
Binomial τ leap ($\epsilon=0.01$)	16.70*
R leap ($\epsilon=0.05,\theta=0.1$)	64.0
R leap ($\epsilon=0.03,\theta=0.1$)	56.6
R leap ($\epsilon=0.01,\theta=0.1$)	20.0

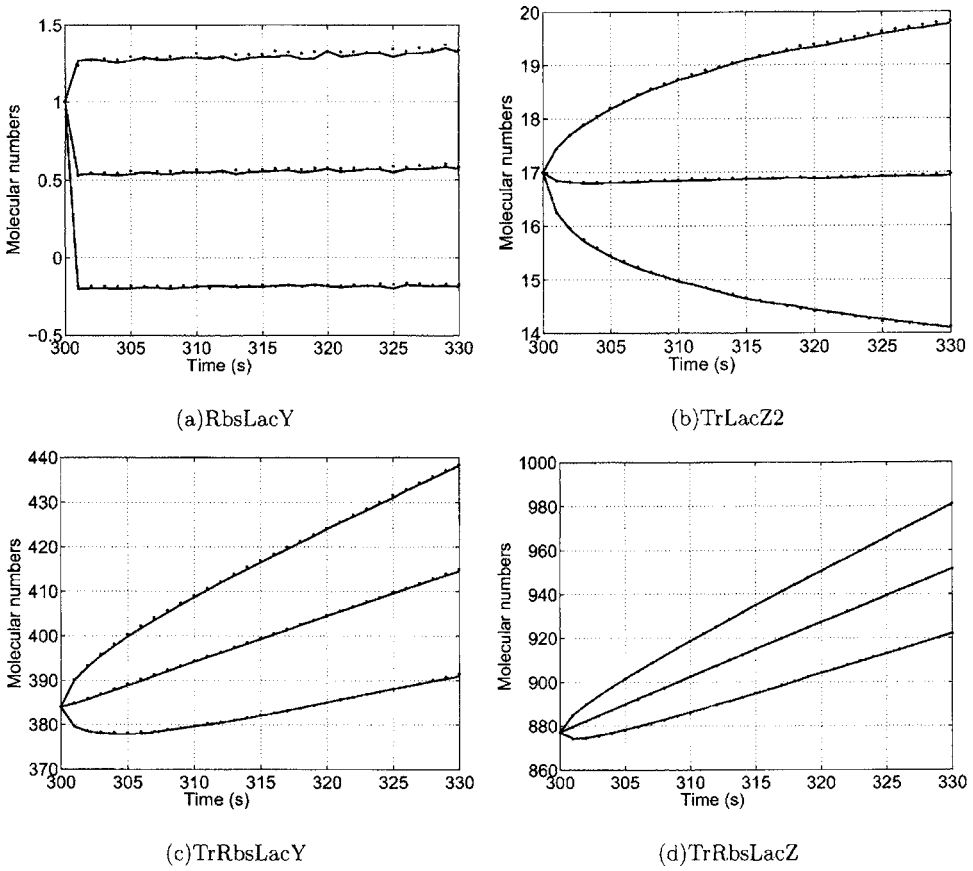


FIG. 7. LacZ/LacY model [Kierzek (Ref. 8)]: mean trajectories and standard deviations of four molecular species for the SSA (solid line) and the *R*-leap method with $\epsilon=0.03$, $\theta=0.1$ (dots).

or not, i.e., there are only two possible outcomes for an index; equal to m and this with probability $a_m(\mathbf{x})/a_0(\mathbf{x})$ or not with probability $1-a_m(\mathbf{x})/a_0(\mathbf{x})$. Each event $\{j_k=m\}$ is then Bernoulli distributed with probability $a_m(\mathbf{x})/a_0(\mathbf{x})$. Because K_m is the number of successes in these L independent Bernoulli sampling, it will follow the distribution $\mathcal{B}(L,a_m(\mathbf{x})/a_0(\mathbf{x}))$. □

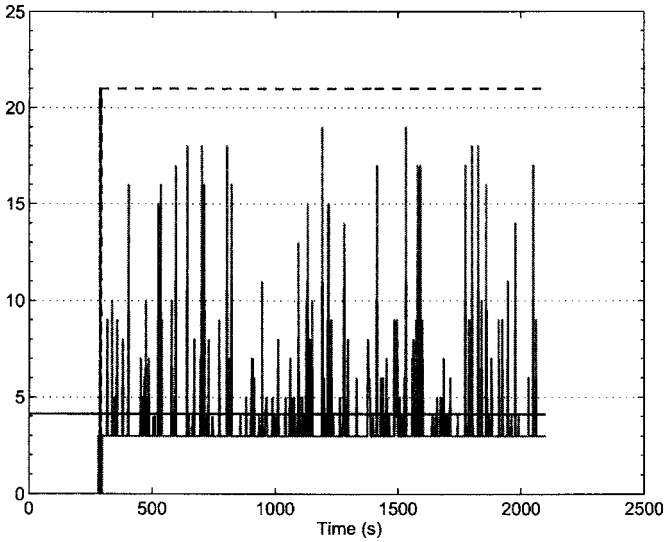


FIG. 8. LacZ/LacY model, first cell generation ($t=0$ to $t=2100$): plot of the evolution of the number of binomial samples N_B for one *R*-leaping run ($\epsilon=0.03$, $\theta=0.1$, condition 1 for L' selection) with (solid) and without (dashed) sorting [step (5)]. When applied, the sorting is made every 10 000 time steps. The straight line indicates the mean value of N_B (equal to 4.1) when the sorting is applied. The value of N_B is monitored every 2500 time steps.

Lemma 2. *The following holds*

$$P(j_k=2|j_k>1)=\frac{a_2(\mathbf{x})}{a_0(\mathbf{x})-a_1(\mathbf{x})}, \tag{A1}$$

$$P(j_k\neq 2|j_k>1)=1-\frac{a_2(\mathbf{x})}{a_0(\mathbf{x})-a_1(\mathbf{x})}. \tag{A2}$$

Proof.

$$P(j_k=2|j_k>1)=\frac{P(j_k=2\cap j_k>1)}{P(j_k>1)}$$

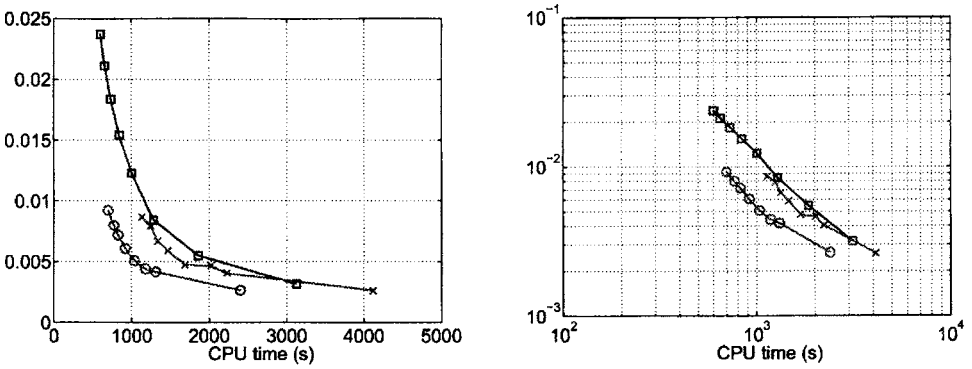
$$=\frac{P(j_k=2)}{P(j_k>1)}=\frac{a_2(\mathbf{x})/a_0(\mathbf{x})}{1-a_1(\mathbf{x})/a_0(\mathbf{x})}=\frac{a_2(\mathbf{x})}{a_0(\mathbf{x})-a_1(\mathbf{x})},$$

$$P(j_k\neq 2|j_k>1)=1-\frac{a_2(\mathbf{x})}{a_0(\mathbf{x})-a_1(\mathbf{x})}.$$ □

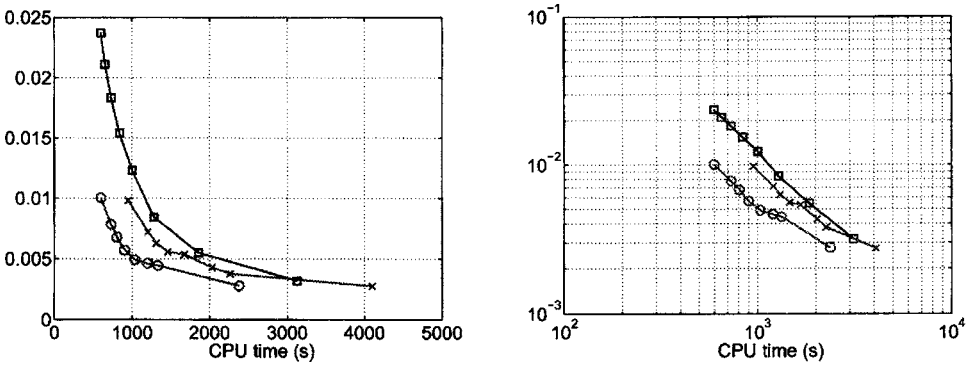
Proof of theorem. From Lemma 1 we have that $K_1\sim\mathcal{B}(L,a_j(\mathbf{x})/a_0(\mathbf{x}))$. Let us now consider the conditional distribution of K_2 with respect to the event $\{K_1=k_1\}$,

$$K_2|\{K_1=k_1\}=\sum_{k=1}^{L}\mathbf{1}_{\{j_k=2|k_1 \text{ indices among } L \text{ are equal to } 1\}},$$

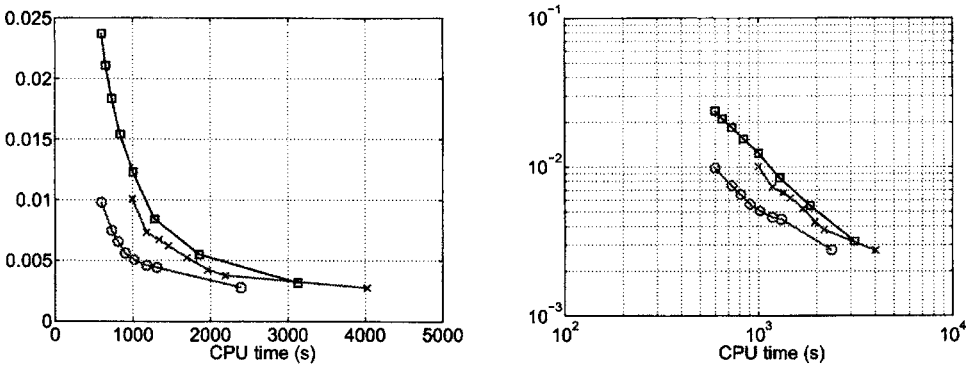
where $\mathbf{1}_{\{A\}}$ is the indicator function of an event A. Let us introduce now the set of indices $(\kappa_1^1,\ldots,\kappa_{k_1}^1)$ for which $j_k=1$, i.e., $j_{\kappa_1^1}=1,\ldots,j_{\kappa_{k_1}^1}=1$. The previous equation can be rewritten as



(a) $\epsilon = 0.05$ for R-leaping



(b) $\epsilon = 0.1$ for R-leaping



(c) $\epsilon = 0.2$ for R-leaping

FIG. 9. LacZ/LacY model ($t=1000$ to $t=1001$): histogram distance errors with respect to CPU times for 2×10^5 runs. The modified Poisson (□) algorithm with τ-selection formula of Ref. 6 [Eq. (3)] is plotted for $\epsilon=0.03$ to $\epsilon=0.1$ by steps of 0.01. The R-leaping with condition 3 is plotted for $\theta=0.025$, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, and 0.75, for fixed values of ϵ (indicated under the figures), with (○) and without (×) sorting.

$$K_2|\{K_1=k_1\} = \sum_{k\in\{\kappa_1^1\ldots\kappa_{k_1}^1\}} \mathbf{1}_{\{j_k=2|k_1 \text{ indices among } L \text{ are equal to } 1\}} + \sum_{k\notin\{\kappa_1^1\ldots\kappa_{k_1}^1\}} \mathbf{1}_{\{j_k=2|k_1 \text{ indices among } L \text{ are equal to } 1\}}.$$

But for $k\in\{\kappa_1^1\cdots\kappa_{k_1}^1\}$, $\mathbf{1}_{\{j_k=2|k_1 \text{ indices among } L \text{ are equal to } 1\}}=0$ and for $k\notin\{\kappa_1^1\cdots\kappa_{k_1}^1\}$, it is equal to $\mathbf{1}_{\{j_k=2|j_k>1\}}$. Then

$$K_2|\{K_1=k_1\} = \sum_{k\notin\{\kappa_1^1\ldots\kappa_{k_1}^1\}} \mathbf{1}_{\{j_k=2|j_k>1\}}.$$

From Eqs. (A1) and (A2), we deduce that K_2 conditioned to $K_1=k_1$ obeys a binomial distribution $\mathcal{B}(L-k_1,[a_2(\mathbf{x})/a_0(\mathbf{x})-a_1(\mathbf{x})])$.

By induction we then prove that conditionally to the event $\{(K_1,\ldots,K_{m-1})=(k_1,\ldots,k_{m-1})\}$ the distribution of K_m is

$$\mathcal{B}\left(L-\sum_{i=1}^{m-1}k_i, \frac{a_m(\mathbf{x})}{a_0(\mathbf{x})-\sum_{i=1}^{m-1}a_i(\mathbf{x})}\right).$$

The previous steps do not depend on the ordering of the K_m and the proof would have been the same applying first a permutation π to the indices and redefining

$$K_i := K_{\pi(i)} \quad \text{for all } 1\leqslant i\leqslant M. \qquad \square$$

[1] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[2] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[3] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).

[4] A. Chatterjee, D. Vlachos, and M. Katsoulakis, J. Chem. Phys. **122**, 024112 (2005).
[5] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **123**, 054104 (2005).
[6] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **124**, 044109 (2006).
[7] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[8] A. M. Kierzek, Bioinformatics **18**, 470 (2002).