THE JOURNAL OF CHEMICAL PHYSICS **129**, 204109 (2008)

Efficient stochastic sampling of first-passage times with applications to self-assembly simulations

Navodit Misra[1] and Russell Schwartz[2,a)]
[1]Department of Physics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA
[2]Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

(Received 1 April 2008; accepted 21 October 2008; published online 25 November 2008)

Models of reaction chemistry based on the stochastic simulation algorithm (SSA) have become a crucial tool for simulating complicated biological reaction networks due to their ability to handle extremely complicated networks and to represent noise in small-scale chemistry. These methods can, however, become highly inefficient for stiff reaction systems, those in which different reaction channels operate on widely varying time scales. In this paper, we develop two methods for accelerating sampling in SSA models: an exact method and a scheme allowing for sampling accuracy up to any arbitrary error bound. Both methods depend on the analysis of the eigenvalues of continuous time Markov models that define the behavior of the SSA. We show how each can be applied to accelerate sampling within known Markov models or to subgraphs discovered automatically during execution. We demonstrate these methods for two applications of sampling in stiff SSAs that are important for modeling self-assembly reactions: sampling breakage times for multiply connected bond networks and sampling assembly times for multisubunit nucleation reactions. We show theoretically and empirically that our eigenvalue methods provide substantially reduced sampling times for a large class of models used in simulating self-assembly. These techniques are also likely to have broader use in accelerating SSA models so as to apply them to systems and parameter ranges that are currently computationally intractable. © *2008 American Institute of Physics*. [DOI: 10.1063/1.3026595]

I. INTRODUCTION

Stochastic simulation methods have become increasingly widespread as a means of simulating and analyzing biochemical reaction kinetics.[1] The chemical master equation, which governs the reaction kinetics for well-mixed systems, forms the basis for the stochastic simulation algorithm (SSA), proposed by Gillespie.[2,3] SSA models a reaction system as a continuous time Markov model (CTMM) in which states of the system are defined by counts of reactants present at a given point in time and transitions between states correspond to individual reaction events. This SSA approach is valuable in part because it provides a model of reaction noise, which can become significant for reaction networks on cellular scales.[4] Furthermore, SSA models can provide significant computational advantages over continuum models for networks characterized by extremely large sets of possible reaction intermediates. The computational value of the SSA approach lies in the fact that for a large class of networks, the random walk visits only a small fraction of the state space before equilibrium is established. As a result, kinetics on complicated networks can be simulated "on the fly," requiring explicit construction of the CTMM network only in the immediate vicinity of those states visited on a given trajectory. This property is an essential requirement for any feasible simulation algorithm, since the size of the state space describing the master equation is astronomical even for modest system sizes. Successful applications of SSA include gene regulatory networks[4] and self-assembly of complicated structures, such as virus capsids.[5,6] Furthermore, the SSA approach has now been adopted by several approaches for whole-cell modeling[7,8] and modeling generic complex reaction networks.[9,10]

The relaxation time of the SSA can, however, be extremely sensitive to the transition rates controlling the reaction kinetics. A pure SSA model has difficulty with stiff reaction systems, i.e., those where important events occur in parallel on very different time scales. In such cases, a simulation can become bogged down by sampling fast events to the exclusion of the slow events. Hybrid discrete/stochastic models[11–13] can resolve this problem in some domains, but not when the fast reactions make use of too many intermediates to allow them to be modeled continuously. One important example of such a stiff reaction system is the breaking of bond networks, where individual bonds may break and repair repeatedly before a sufficiently large bond group is broken to fracture the network. Another form of stiff SSA network occurs near the critical concentration of a self-assembly system, where high-order nucleation events can be orders of magnitude slower than individual binding reactions. In these stiff systems, a SSA model can become "trapped" for many steps in a small subset of the state space, resulting in negligible simulation progress for long periods of time.

To understand these trapped systems, it is useful to consider the graph theoretic representation of the SSA method. A SSA model is represented by a graph in which each node corresponds to one possible state of the full model. Edges

---

[a)]Electronic mail: russells@andrew.cmu.edu.

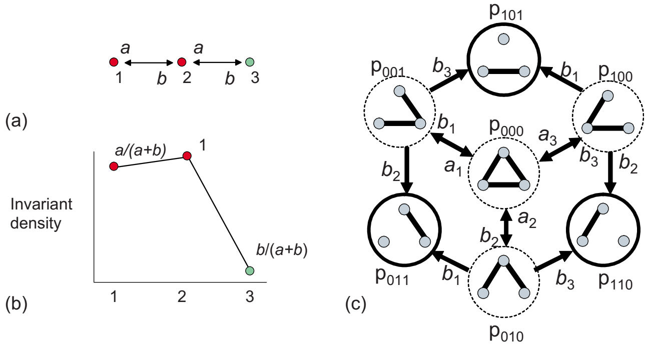

FIG. 1. (Color online) Illustration of trapped subgraphs in SSA models. (a) A simple CTMM on a three-path with transition rates a and b. (b) The probability landscape for the model. SSA is slow whenever the invariant density for the corresponding Markov chain is irregular. Here, the SSA takes $O(a/b)$ steps to reach vertex 3. (c) CTMM model of a trimer assembly system with three subunits. Graph of possible configurations joined by reaction rates. States in which the trimer is broken are surrounded by solid lines and others by dashed lines.

connect nodes whose states can be reached from one another by a single reaction event, e.g., two molecules binding to one another. At each simulation step, SSA considers only the immediate neighbors of the current state. As a result, the simulation is prone to traps that can result from irregularities in the invariant density of the embedded Markov chain (EMC) implemented by SSA for a given CTMM. For example, consider a three-state CTMM represented by a simple path [Fig. 1(a)], where the backward transition rate a is much larger than the forward transition rate b. The average number of SSA steps to reach state 3 from initial state 1 is $O(a/b)$ because once SSA visits state 2, it will jump to state 3 only a b/a fraction of the time. Nodes 1 and 2 collectively define a trapped subgraph from which the model must escape. In general, for an N-path, where each forward rate b is smaller than the backward rate a, SSA takes $O(a/b)^{N-2}$ steps to traverse the path (see Theorem III.1 for an analogous problem). One way such a trapped subgraph can arise in a physical system is through models of the breakage of bond networks. Figure 1(c) shows the graph arising from a model of the breakage of a three-cycle bond network, which behaves similarly to the three-state CTMM by establishing a trapped inner graph of four states—the unbroken state and three states with a single broken bond—from which the model must escape to reach any broken network state. We can alternatively understand the trapping problem in terms of a probability landscape view of a reaction system. The SSA is sluggish whenever its equilibrium landscape is irregular, consisting of valleys and hills. The broader and deeper these are, the slower SSA becomes.

To overcome the presence of traps or landscape irregularity, we propose two *nonlocal* simulation algorithms that rely on the spectral decomposition of the Kolmogorov matrix (for a CTMM) or the transition matrix (for the EMC). These eigenvalues and their associated eigenvectors describe global modes of relaxation of the full graph or any of its subgraphs. Since eigenvalues are global properties of a graph, spectral methods are much less sensitive to local landscape traps. These methods can be applied to quickly sample first-passage times on small CTMM graphs such as those in Fig. 1 or to sample escape times from trapped subgraphs when the full model is prohibitively large.

Previous attempts at simulating rare events include the forward flux sampling (FFS) technique of Allen *et al.*[14] and related methods.[15,16] The approach breaks a rare event into a series of relatively more probable stages and uses estimates of waiting times for the successive stages to develop an aggregate transition rate for the full event. This aggregate rate can then be used to approximate the first-passage time density as a single exponential random variable. However, while the exponential tail dominates the density for stiff systems and is therefore a highly accurate approximation in many cases, the true probability density has a peak at short times followed by a mixed exponential tail. The methods developed in the present work, in contrast to the FFS-like methods, sample first-passage times from the entire density to within an arbitrary error bound. Recently, another method called the slow-scale SSA was proposed by Cao *et al.*,[17,18] which relies on a technique called the partial equilibrium approximation (PEA). PEA essentially assumes that the set of fast reactions is always in equilibrium and the method approximates transition rates between slowly varying reactant species by their expected value in the partial equilibrium state. While these methods can provide significant benefits for some CTMMs, there are several limitations in using PEA or similar approximations for arbitrary graphs. First, a clear distinction between fast and slow species may not be obvious in a given problem. For example, in rule-based simulation of bond networks, stiffness is built in through the association/dissociation rates of individual bonds rather than being species dependent. Second, these methods always need to be supplemented with approximations involved in computing the mean values of the reaction propensities. Furthermore, PEA will be inaccurate whenever fluctuations in the reaction propensities within the partial equilibrium state are comparable to their mean values.

The goal of the present work is to develop efficient methods for some important classes of stiff SSA model for which the above techniques are unsuitable, with a particular emphasis on models important to simulations of self-assembly reactions. The methods proposed in this paper can be applied to Markov processes on arbitrary graphs. Furthermore, they can be made accurate to within arbitrary error bounds. The remainder of this paper is organized as follows. Section II A sets up some basic notation and a description of the sampling problem for general CTMM. In Sec. II B we introduce a spectral method which relies on the eigendecomposition of the master equation describing the CTMM (Fig. 2). We use a complete spectral decomposition of the first-passage time density and rejection sampling to return sample first-passage times for arbitrary CTMMs. In Sec. II C we introduce another spectral method which works as a hybrid between the purely local SSA and the completely nonlocal master equation method. The latter method proceeds by adaptively constructing a basis in which to simulate the Markov chain until the system state has relaxed to its slowest eigenvector (Figs. 3 and 4). If first passage out of the trapped subgraphs does not occur by that time, we use the appropriate eigenvalue to sample the time to first passage as an ex-

```
Algorithm: Prepare Discrete Mixture
Input: First passage time probability density $\rho$
Output: Envelope $g(t) = \sum_i S_i f_i(t)$, rejection ratio $R$

Sort the list $\{c_i\}$ in increasing order of $\lambda_i$;
Initialize $R \leftarrow 0$;
for $i \in \{1, ...N\}$ do
    Compute the partial sums $p_i \leftarrow \sum_{n=1}^{i} c_n$;
    if $p_i > 0$ then
        $S_i \leftarrow p_i * (\lambda_{i+1} - \lambda_i)/(\lambda_i \lambda_{i+1})$;
        $R \leftarrow R + S_i$;
    end
end
Return $\{S_i\}, R$;
```

```
Algorithm: Sample first passage time
Input: Probability density $\rho(t) = \sum_i c_i e^{\lambda_i t}$
Output: Sample time $T$ distributed according to $\rho$

Set $\{S_i\}, R \leftarrow$ Prepare Discrete Mixture($\rho$);
$reject1 \leftarrow 0$;
while $reject1 < 1$ do
    Generate a uniform $[0, 1]$ random variate $U$;
    $prob \leftarrow 0$;
    $mix \leftarrow 1$;
    while $prob < U$ do
        $prob \leftarrow prob + S_{mix}/R$;
        $mix \leftarrow mix + 1$;
    end
    $reject2 \leftarrow 0$;
    Compute an envelope $f_{mix} \leq R_{mix} e^{-g_{mix} t}$:
        Set $g_{mix} \leftarrow (\lambda_{mix} * \lambda_{mix+1})/(\lambda_{mix} + \lambda_{mix+1})$;
        Compute $R_{mix}(\lambda_{mix}, \lambda_{mix+1})$;
    while $reject2 < 1$ do
        Generate an exponential random variate $T$, with mean $1/g_{mix}$;
        Generate a uniform random variate [0,1] $X$;
        $reject2 \leftarrow \frac{f_{mix}(T)}{\exp(-g_{mix} T) * R_{mix} * X}$;
    end
    Generate a uniform random variate [0,1] $Y$;
    $reject1 \leftarrow \frac{\rho(T)}{g(T) * R * Y}$;
end
Return $T$;
```

FIG. 2. Pseudocode for spectral method 1.

ponential random variable. In Sec. II D we introduce a method for automated discovery (AD) of trapped regions in stiff Markov models. This technique allows efficient implementation of spectral methods for large state spaces by isolating regions repeatedly visited by a given random trajectory and using spectral sampling to escape any such subgraph. In Sec. III we present theoretical results on the time complexity of SSA for bond networks followed by experiments on some special classes of bond networks to compare the simulation efficiency of each method discussed. In Sec. IV we evaluate the AD variants of the method by applying them to models of a nucleation-limited assembly system with a state space too large to explicitly construct. Section V concludes the paper with a discussion of results and directions for future research.

II. THEORY

A. The chemical master equation and the SSA

The SSA identifies reaction kinetics for networks of biochemical subunits as a Markov process governed by an appropriate *Chapman–Kolmogorov* equation or, equivalently, its differential version—the master equation. Let $S=\{1,2,\ldots,N_S\}$ be the state space for the CTMM, each node representing a possible state for the simulated system. The time evolution of probability densities is governed by a *Kolmogorov* matrix W, which specifies the transition rates W_{nm} from the state m to n,

$$\frac{dp_n}{dt} = \sum_{m \in S} W_{nm} p_m(t) - W_{mn} p_n(t), \tag{1}$$

where $p_n(t)$ denotes the probability to be in state n at time t. The matrix elements W_{nm} satisfy two necessary conditions:

(1) $W_{nm} \geq 0$ for $n \neq m$;
(2) $\Sigma_m W_{nm} = 0$.

Under these conditions, it is well known that the matrix has a steady state solution $|\Pi\rangle = \Sigma_n \pi_n |n\rangle$ that is an eigenvector of W with eigenvalue zero and that all initial distributions relax to $|\Pi\rangle$ in the limit of long times.[19] In addition, we will require W to satisfy the *detailed balance* condition, which states that at equilibrium, the sum of probability current exchanged between any pair of states (n,m) is zero, i.e., $W_{nm}\pi_m = W_{mn}\pi_n$. This in turn allows one to define a scalar product on the state space such that W is *self-adjoint*,

$$\langle n|m\rangle \equiv \delta_{nm} \frac{1}{\pi_m}. \tag{2}$$

This condition ensures that we can construct an orthogonal *eigenbasis* and compute time-evolved versions of any given initial probability distribution using spectral decomposition.

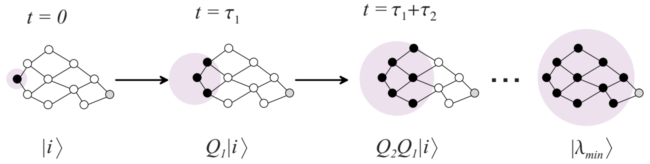

FIG. 3. (Color online) Schematic of the EMC-based spectral method. Filled vertices are the currently occupied nodes. Simulation advances the system state as a discrete mixture until such time as the state has relaxed to its slowest eigenstate $|\lambda_{\min}\rangle$. At each step, direct transitions to the absorbing vertex (gray) are computed according to the Kolmogorov matrix.

B. Spectral sampling 1: Master equation approach

Given a Kolmogorov matrix W on a state space S and an arbitrary initial state $i \in V \subset S$, the first-passage time $T_F(i)$ is a random variable which gives the time at which the trajectory first reaches any state in some subset of the state space $F = S - V$. The standard method of solving a first-passage problem is to set up the master equation for V with an absorbing boundary over F (zero Dirichlet boundary condition).[19] Let P_V be a projection operator onto the subspace V and let N be the cardinality of V. Then, $M = P_V W P_V$ is the effective Kolmogorov matrix that governs time evolution over V. From detailed balance, M is self-adjoint over $L^2_{\pi^{-1}}$. Hence, the eigenvectors of M form a complete basis $\{|\psi_\alpha\rangle\}$. A consequence of the spectral theorem is

```
Algorithm: Next State
Input: Current state vector $|\psi\rangle = \sum_i \psi_i |i\rangle$
Output: Next state $|\phi\rangle = \sum_i \phi_i |i\rangle$ and rate $r$

for $i \in V$ do
    if $\psi_i > 0$ AND $r < -M_{ii}$ then
        $r \leftarrow -M_{ii}$;
    end
end
Compute the next state $\phi_i \leftarrow \sum_j (\delta_{ij} + \frac{1}{r} M_{ij}) \psi_j$;
Return $\{|\phi\rangle \leftarrow \sum_i \phi_i |i\rangle, r\}$;
```

```
Algorithm: Check Convergence
Input: Next state $|\phi\rangle$, present state $|\psi\rangle$, rate $r$
Output: The first passage time $t$

$EigenMode \leftarrow$ Yes;
$P \leftarrow 0$;
for $i \in V$ do
    $P \leftarrow P + \phi_i$;
    if $|\phi_i - \psi_i| > \epsilon \psi_i$ then
        $EigenMode \leftarrow$ No;
    end
end
Generate a uniform $[0,1]$ random variable $U$;
if $EigenMode = Yes$ then
    $\lambda_{min} \leftarrow r * (1 - P)$ ;
    Return $t \leftarrow t - \frac{1}{\lambda_{min}} \ln U$;
end
else
    Compute the next time step $\tau \leftarrow -\frac{1}{r} \ln U$;
    $t \leftarrow t + \tau$;
    Generate a uniform $[0,1]$ random variable $X$;
    if $X < P$ then
        $\{|\psi\rangle, r\} \leftarrow$ Next State($\frac{1}{P}|\phi\rangle$);
        Check Convergence($|\psi\rangle$,$|\phi\rangle$,r);
    end
    else
        Return $t$;
    end
end
```

FIG. 4. Pseudocode for spectral method 2.

the completeness relation for the properly normalized eigenbasis, i.e., $\langle \psi_\alpha | \psi_\beta \rangle = \delta_{\alpha\beta}$. Given any vector $|\eta\rangle$,

$$|\eta\rangle = \sum_{\alpha=1}^{N} |\psi_\alpha\rangle\langle\psi_\alpha|\eta\rangle. \tag{3}$$

1. Spectral decomposition of the first-passage time distribution

In terms of the vertex set basis, the completeness relation over $L^2_{\pi^{-1}}$ is $I = \Sigma_{n \in V} P_n$, where $P_n = \pi_n |n\rangle\langle n|$ is the projector onto vertex state $|n\rangle$. Given an initial probability density $p_n(t=0) = \delta_{ni}$, the probability for state $n \in V$ evolves as

$$p_n(t)|n\rangle = P_n e^{tM}|i\rangle = \pi_n |n\rangle\langle n| \sum_{\alpha=0}^{N} \langle\psi_\alpha|i\rangle e^{-\lambda_\alpha t}|\psi_\alpha\rangle \Rightarrow p_n(t)$$
$$= \sum_{\alpha=1}^{N} \pi_n \psi_{\alpha,n} \psi_{\alpha,i} \exp[-\lambda_\alpha t]. \tag{4}$$

The transition to an element $f \in F$ outside of V is governed by the following equation:

$$\frac{dp_f}{dt} = \pi_f \langle f|(W-M) \sum_{n \in V} p_n(t)|n\rangle = \sum_{\alpha=1}^{N} c_{\alpha,f} \exp[-\lambda_\alpha t]. \tag{5}$$

The probability for a first passage to the state f between t and $t+dt$ is hence given by $\rho(T_f = t)dt = \Sigma_{\alpha=1}^{N} c_{\alpha,f} e^{-\lambda_\alpha t} dt$.

2. Exact sampling for the first-passage time distribution

In this section we describe a method for returning a sample time from the computed first-passage density $\rho(t) = \Sigma_{i=1}^{N} c_i e^{-\lambda_i t}$ to any state $f \in F$. A general method for sampling from complicated distributions is to use the method of rejection sampling, which first chooses a random variable from a convenient envelope density and accepts or rejects the sample based on a second random sample that depends on the tightness of the envelope fit. The rejection rate is low if the envelope curve closely approximates the given curve. A simple envelope curve is provided by a pure exponential of the most slowly decaying eigenvalue, with a coefficient equal to the sum of all positive terms $\Sigma_{c_i > 0} c_i$ in the computed density $\rho(t)$. However, there is no guarantee that the rejected part is small. Since each eigenmode encloses an area c_i / λ_i, cancellations between near-degenerate eigenvalues can in principle lead to a high rejection ratio. We therefore present a method for choosing an envelope curve $g(t)$ which eliminates these cancellations. Furthermore, in Sec. III B we show that the envelope curve is exact for bond networks generated by cycle graphs C_N. We sample from $g(t)$ using a decomposition into a discrete mixture of densities $f_\alpha(t) = d_\alpha (e^{-\lambda_\alpha t} - e^{-\lambda_{\alpha+1} t})$ and an efficient rejection step. Here d_α are constants, one for each component f_α of the envelope curve $g(t)$. The next theorem proves that the density $f_\alpha(t)$ can be sampled efficiently using a rejection method.

Theorem II.1: *The expected rejection ratio for $f_\alpha(t)$ is bounded from above by 1.5.*

Proof: We will use a simple exponential $h_\alpha(t) = C\lambda \exp[-\lambda t]$ as the envelope function. In order to minimize C we choose $h_\alpha(t)$ such that

$$h_\alpha(t_*) = f_\alpha(t_*) \tag{6}$$

and

$$\left.\frac{dh_\alpha}{dt}\right|_{t=t_*} = \left.\frac{df_\alpha}{dt}\right|_{t=t_*}, \tag{7}$$

where t_* is defined implicitly by the condition

$$\left.\frac{d^2 f_\alpha}{dt^2}\right|_{t=t_*} = 0. \tag{8}$$

These constraints yield a unique solution $t_* = 2\ln(\lambda_{\alpha+1}/\lambda_\alpha)/\Delta\lambda$. Since $d^2 f_\alpha / dt^2 > 0$ for $t > t_*$, the slope of $\ln[f_\alpha]$ monotonically increases to $-\lambda_\alpha$ as $t \to \infty$. The corresponding envelope rate then satisfies $\lambda = \lambda_\alpha \lambda_{\alpha+1}/(\lambda_\alpha + \lambda_{\alpha+1}) \leq \lambda_\alpha$. The rejection ratio is given by

$$\begin{aligned} C &= \frac{1}{\lambda}\frac{\lambda_\alpha\lambda_{\alpha+1}}{\lambda_{\alpha+1}-\lambda_\alpha}\exp[(\lambda-\lambda_\alpha)t_*]\left(1-\left(\frac{\lambda_\alpha}{\lambda_{\alpha+1}}\right)^2\right) \\ &= \left[\frac{\lambda_\alpha+\lambda_{\alpha+1}}{\lambda_{\alpha+1}}\right]^2\exp\left[-\frac{\lambda_\alpha^2}{\lambda_\alpha+\lambda_{\alpha+1}}t_*\right] \\ &\leq 4\exp\left[2\frac{x^2\ln[x]}{(1-x^2)}\right], \end{aligned} \tag{9}$$

where $x=\lambda_\alpha/\lambda_{\alpha+1}\in(0,1)$. To upper-bound C, note that the exponent increases monotonically with x and its maximum is $\lim_{x\to 1-}(2x^2\ln[x])/(1-x^2)=-1$. This bound finally gives us $C\leq 4/e\approx 1.47$. □

As a final comment, we note that for general graphs the average time complexity of this algorithm is dominated by the computation of the eigenvectors and eigenvalues, which gives us the following theorem.

Theorem II.2: *The average time complexity for spectral decomposition of the master equation is* $O(N^3)$ *for a graph of* N *vertices.*[20]

C. Spectral sampling 2: Modified EMC method

The efficiency of the SSA is dependent on the relaxation time of the EMC. We use this observation to modify the basis in which the EMC is simulated. The standard method of executing a random walk is to consider the transition between adjacent states, each of which is localized at a vertex of the CTMM. However, correct simulation only requires that these states form a basis, not that they are orthogonal. If we can choose a set of states which are increasingly likely to appear during the simulation of the Markov chain, we are unlikely to make repeated visits to the same state. In order to identify such a basis starting from an initial state $|i\rangle$, we first identify the transition matrix for an EMC that correctly describes the given CTMM. Consider the vertex set $V=\{1,2,\ldots,N\}$ and the basis constructed from V, $B=\{|1\rangle,|2\rangle,\ldots,|N\rangle\}$. At any given time t, let the state of the time-evolved Markov chain be $|\psi(t)\rangle=\Sigma_{i=1}^N\psi_i|i\rangle$. Let $V_t=\{i\in V|\psi_i\neq 0\}$ be the vertex subset populated by the current state vector. We construct the EMC for the subgraph induced by V_t at each step of the algorithm. Given the projection of the Kolmogorov matrix M over the vertex set V, choose $r=\max(-M_{ii}|\psi_i\neq 0)$ to be the effective rate of transition to the next state and choose an exponentially distributed random time step τ with mean waiting time $1/r$. Then, $L_t=-(1/r)\times M$ is the Laplacian governing the EMC and $Q_t=I-L_t$ is the effective transition matrix at that time step. The next state vector is chosen to be $|\phi\rangle=Q_t|\psi(t)\rangle$. The reason for choosing this particular value of r is to ensure that no term in Q_t becomes negative, a necessary condition for a transition matrix.

Theorem II.3: *The choice of next state is consistent with the master equation governing the CTMM.*

Proof: Rewrite the master equation in terms of the $\{|\psi\rangle,|\phi\rangle\}$ basis (where the other $N-2$ linearly independent basis vectors can be chosen arbitrarily),

$$\frac{d|\psi\rangle}{dt}=r\left(I+\frac{1}{r}M-I\right)|\psi\rangle=r(Q_t-I)|\psi\rangle=r(|\phi\rangle-|\psi\rangle). \tag{10}$$

Since there is a unique decomposition for any vector in terms of a linearly independent basis set, Eq. (10) proves that starting from $|\psi\rangle$ the next state is uniquely determined to be $|\phi\rangle$. □

In general, the next state $|\phi\rangle$ will have a total probability $P=\Sigma_i\phi_i\leq 1$ due to possible transitions out of the subgraph. We check if that is the case by generating a $[0,1]$ random variable X to compare with P. If $X<P$, the next state is still trapped inside the subgraph and we normalize it as $\psi(t+\tau)=1/P|\phi\rangle$. $|\psi(t+\tau)\rangle$ is used to generate the next state in the simulation. This sequence will continue until the state has relaxed to its slowest eigenvector $|\lambda_{\min}\rangle$, such that $M|\lambda_{\min}\rangle=-\lambda_{\min}|\lambda_{\min}\rangle$ to within a user-defined relative error ϵ. Once that state is achieved, we just need one more exponentially distributed random sample time τ with mean $1/\lambda_{\min}$ to escape the network.

SSA chooses a stochastic trajectory by sampling both the next neighbor and the time for the next step at random. The EMC method, on the other hand, evolves deterministically in our modified basis and only the time between transitions is stochastic. At each time step, transition to the absorbing boundary states is governed by the matrix elements connecting each of the transient states to the absorbing boundary. The advantage of such an approach is that it allows us to automatically compute the most slowly decaying eigenvector during the simulation. For completeness, we note the following result.

Theorem II.4: *For a graph of degree bounded by* d *and* V *of cardinality* N *, each step of this algorithm takes* $O(N\times d)$ *time.*

D. AD of trapped subgraphs

As previously mentioned, stiffness in Markov model graphs results from repeated visits by a typical random trajectory to a small subset of vertices of the entire graph. Since the performance of spectral methods is sensitive to the size of the vertex set, it would prove useful if we could somehow identify these trapped subgraphs for stiff Markov models and apply spectral methods directly to those. In this section we present one such method, which we call "automated discovery" (AD) and which we show to be formally applicable to arbitrary bounded-degree graphs.

Let there be a state space S over which a CTMM is defined and consider a subgraph $G(V,E)$ with vertex set $V\subset S$ and edge set E. Starting from an initial state $i\in V$, we are interested in the time $T_F(i)$ to first passage out of V. Consider the subgraph H_i induced by the vertex set $U_i\subseteq V$ visited by a random trajectory executing the SSA random walk before it escapes V and let $N_i=|U_i|$ be the cardinality of U_i. If T_{Fi} is the number of steps a SSA random walk takes to escape V, then a Markov model will show simulation stiffness whenever the expected values satisfy

```
Algorithm: Automated Discovery
Input: Discovered vertex set $K$, current state $i$, time elapsed $t$
Output: Final state vertex $i \notin V$, first passage time $t$

while $i \in V$ do
    $size \leftarrow |K|$;
    $steps \leftarrow 0$;
    if $steps < c * size^3$ then
        Generate SSA next state $i$;
        Update $t$;
        $steps \leftarrow steps + 1$;
        if $i \notin K$ then
            $K \leftarrow K \bigcup \{i\}$;
            $size \leftarrow size + 1$;
        end
    end
    else
        Do Spectral Sampling over $K$;
        $i \leftarrow$ Next state outside $K$;
        $t \leftarrow$ updated time;
        $K \leftarrow K \bigcup \{i\}$;
    end
end
Return $i, t$;
```

FIG. 5. Pseudocode for AD.

$$E[N_i] \ll E[T_{Fi}], \qquad (11)$$

since this would imply that certain vertices in U_i are being visited repeatedly. AD works by progressively sampling larger regions of V until it identifies a subgraph K_i induced by a vertex set $W_i \subseteq V$ such that $U_i \subseteq W_i$. Once K_i is identified either of the spectral methods can be used directly over K_i. The method will be efficient as long as $|W_i| \sim |U_i|$ and the number of steps taken to identify K_i is comparable to the computational cost of using spectral sampling over K_i. Formally, H_i can be exactly discovered by repeatedly enlarging the discovered graph to include the last vertex outside K_i visited by the trajectory. If spectral sampling for a graph of vertex set size N works in time $f(N)$, this procedure would ensure that implementing spectral sampling in conjunction with AD takes $O(N_i \times f(N_i))$ steps. The stiffness condition [Eq. (11)] would usually ensure that this procedure is still efficient. Another method for discovering the trapped subgraph would be to implement the SSA random walk for a specified number of steps $S(N)$ (depending on the size of the vertex set N). Since eigenvalue methods are, in general, $O(N^3)$, we can implement SSA until $S(N) \leq C \times N^3$ for some constant C to discover the trapped subgraph K and then use spectral sampling to escape the discovered graph. This alternative approach could be less efficient in some circumstances, but would guarantee that the overhead for spectral sampling is not more than a constant factor beyond that of the standard SSA. Figure 5 shows the pseudocode for implementing AD for a given graph by this method. The algorithm generates a sample trajectory using SSA until such time that the trajectory spends $O(N^3)$ steps within a trapped graph K of vertex set cardinality $N = |K|$. Then either of the spectral methods described in Sec. II B or Sec. II C are used to sample the first passage outside K to a vertex i. In general, the state of the system at the time of first passage outside K will be a discrete probability mixture of more than one vertices. In such cases, the vertex i is randomly selected in accordance with the appropriate probability weight. The algorithm then resumes SSA execution over the enlarged graph $K \cup \{i\}$. Further investigation is, however, required to search for algorithms that may further improve the performance of AD. In Sec. II B we prove that for at least one important class of graphs, namely, models of chemically reacting species, we can indeed reduce the time complexity to its optimal value to within a constant factor, i.e., $O(f(N_i))$.

III. APPLICATION WHEN THE SUBGRAPH IS KNOWN: FRACTURING BOND NETWORKS

A. Stiffness in SSA for bond networks

In order to validate the methods, we instantiate them for some specific challenging systems. We begin by demonstrating the non-AD variants of the methods for the problem of sampling the time required to break a network of bonds. This problem is an example of a stiff SSA on a generally small graph. It is also of independent interest because of its importance in modeling self-assembly processes on long time scales. Given such a system, we are interested here in the first-passage time to the subset of states corresponding to disconnected graphs $V_b \subset S$. Since each bond can occur in two states, intact or broken, a network of d bonds can be represented as a vertex on a unit hypercube in d dimensions. The state space generated by the bond network before it becomes disconnected will usually be a truncated unit hypercube. An N-cycle C_N generates the simplest nontrivial example, where the absorbing boundary is placed at all points on the hypercube at distance 2 from the fully connected state. Figure 1(c) illustrates this absorbing boundary for C_3. Given a d-bond network, we will represent the μth bond-breaking rate by b_μ and association or binding rate by a_μ. It is convenient to represent a vertex on this hypercube by a binary d-tuple $\boldsymbol{i} = \{i_d, \ldots, i_\mu, \ldots, i_1\}$, where $i_\mu = 0$ implies that the μth bond is intact [see Fig. 1(c) for the graph corresponding to a trimer]. From here on, we will use the notation $\hat{\boldsymbol{\mu}} = \{\delta_{d\mu}, \ldots, \delta_{1\mu}\}$ for the vector describing a state of the model with only the μth bond broken. For such a graph, the time complexity of each SSA step is $O(d)$. In the rest of this paper we will use this model of truncated hypercubes to represent bond networks. Morris and Sinclair[21] proved that in the case of unweighted graphs, a random walk on a hypercube truncated by a hyperplane relaxes to equilibrium in polynomial time bounded by $O(d)^{9/2+\epsilon}$ for any $\epsilon > 0$. However, as we have argued in Sec. I, the mean *hitting time*, i.e., the number of random walk steps between a pair of vertices, can be extremely sensitive to the parameters governing the walk. We formalize this observation in Theorem III.1 below, which bounds the expected number of SSA steps before the network is disconnected. Let $r \equiv \min(a_\mu / b_\nu | \mu, \nu \in \{1, \ldots, d\})$.

Theorem III.1: *The expected number of SSA steps required to break a* k-connected network with $k > 1$ *and* $r > 1$ *is* $\Omega(r^{k-1})$.

A detailed proof of the theorem is provided in the Appendix. Figures 6 and 7 provide an empirical demonstration of the theorem. Figure 6 analyzes the number of steps required in 100 trials of the SSA algorithm for simulating the breakage of a set of cycle graphs C_N ranging in size from 3

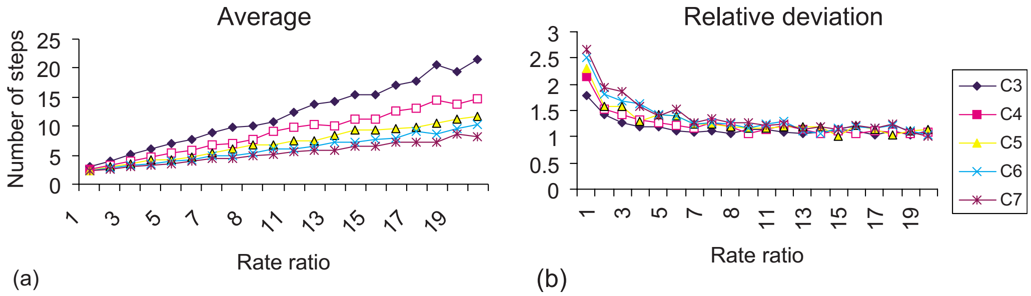

FIG. 6. (Color online) Number of SSA steps until first passage for the network generated by an N-cycle C_N. (a) Average number of steps $\langle s\rangle$. (b) Relative deviation $\frac{\sqrt{\langle \delta s^2\rangle}}{\langle s\rangle}$.

to 7. Each model was examined using ratios of forward to backward rate from 1 to 20 in increments of 1. Breakage times for the cycle graphs increase linearly with rate ratio, although they also fall monotonically with cycle size [Fig. 6(a)]. Figure 7 analyzes the number of steps required to break k-connected hypercube graphs of dimensions $k=\{2,3,4,5\}$. Figure 7(a) also shows that the slope of a log-log plot approaches the predicted exponent $k-1$. Figures 6(b) and 7(b) suggest why a spectral approach might be effective—as the reaction rate increases, steps to first passage behave more like a geometric random variable (as mean $\rightarrow\infty$, standard deviation $\rightarrow$ mean), as expected for a slowly decaying eigenmode of the transition matrix. More detailed explanations of the simulation protocol for these figures are provided in Sec. III C.

B. Master equation for bond networks

1. Spectral analysis for breaking the C_N network

We illustrate the master equation spectral method using the cycle graph C_N as an example. This is a graph of N vertices and N edges, connected together in a loop such that exactly two edges need to be removed to disconnect the graph (called a *separation pair*). The state space is $S=\{\mathbf{0}\}\cup_{\mu=1}^{N}\{\hat{\boldsymbol{\mu}}\}\cup_{\mu=2}^{N}\cup_{\nu<\mu}\{\hat{\boldsymbol{\mu}}+\hat{\boldsymbol{\nu}}\}$. In this case, the subspace $V_b=\cup_{\mu=2}^{N}\cup_{\nu<\mu}\{\hat{\boldsymbol{\mu}}+\hat{\boldsymbol{\nu}}\}$ defines the absorbing boundary and the subspace $V_c=S-V_b$ defines the space of *transient* states. We begin with the most general form for M, the projection of W onto the subspace V_c.

$$M=\begin{pmatrix} -\sum_{\mu} b_\mu & a_1 & a_2 & \dots & a_N \\ b_1 & -\left(a_1+\sum_{\mu\neq 1} b_\mu\right) & 0 & \dots & 0 \\ b_2 & 0 & -\left(a_2+\sum_{\mu\neq 2} b_\mu\right) & 0 & \dots \\ \dots & \dots & 0 & \dots & 0 \\ b_N & 0 & \dots & 0 & -\left(a_N+\sum_{\mu\neq N} b_\mu\right) \end{pmatrix}. \qquad (12)$$

In what follows, we assume that all the eigenvalues of M are negative (as they must be over the subset of *transient states* since $\Sigma_n M_{nm}\leq 0$ ensures any positive probability density decays to zero) and that the set of rates $\{a_i\}$ and $\{b_j\}$ is positive (ensured by property 1 of W). For economy of notation, let us define $k_n=a_n+\Sigma_{m\neq n}b_m$. Also, in what follows we assume that the bond indices have been labeled such that $k_1\leq k_2,\ldots,k_\alpha\leq k_{\alpha+1},\ldots,\leq k_N$. In the case of a C_N network, the $T_f(i)$ distribution can be efficiently sampled due to certain properties of the eigenvalue distribution and the form of the eigenvectors. Since the sampling technique for a general CTMM will be an extension of this special case, it will be helpful to illustrate the method by investigating the spectral properties of C_N. The next few results establish bounds on the eigenvalues of M as a special case of the interlacing eigenvalue theorem.[22]

Theorem III.2: *The* $N+1$ *eigenvalues* $\{-\lambda_0>-\lambda_1>\cdots-\lambda_N\}$ *of the matrix* M *in Eq. (12) satisfy the following.*

(1) *If* $k_i=k_{i+1}$ *then* $-k_i$ *is an eigenvalue of* M*. If* n *such diagonal elements are identical then the eigenvalue is* $(n-1)$*-fold degenerate.*

(2) *There is at least one eigenvalue of* M *in the interval* $\epsilon_i\equiv(-k_i,-k_{i+1})$.

Proof: The *eigenvalue* condition $\mathrm{Det}|M-\lambda I|=0$ implies that the eigenvalues λ are the zeros of an $(N+1)$th order polynomial,

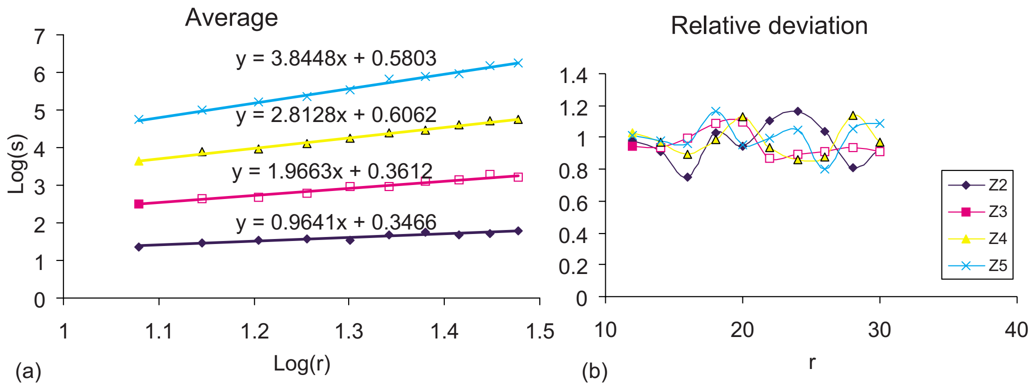

FIG. 7. (Color online) Number of SSA steps until first passage on an N-dimensional unit hypercube Z_N. (a) A $\log_{10}$-$\log_{10}$ plot of the average number of steps s vs rate ratio r. (b) Relative deviation $\frac{\sqrt{\langle \delta s^2 \rangle}}{\langle s \rangle}$.

$$f(\lambda) = \left(\sum_{\mu} b_\mu + \lambda \right) \prod_{i=1}^{N} (k_i + \lambda) - \sum_{i=1}^{N} a_i b_i \prod_{j \neq i} (k_j + \lambda) = 0. \tag{13}$$

We establish bounds on the roots by calculating the sign of $f(\lambda)$ over the set of points $\{-k_1, \ldots, -k_N\}$.

(1) Each term inside the summation sign in $f(\lambda)$ contains $n-1$ factors of $(k_i+\lambda)$. Hence $-k_i$ is an $(n-1)$-fold degenerate eigenvalue. In what follows we assume that the remaining k_j are all distinct.

(2) The sign of the function $f(\lambda)$ at $\lambda=-k_i$ is $(-1)^i$. Hence ϵ_i encloses at least one root of $f(\lambda)$. □

The eigenvectors of M $\{|\psi_\alpha\rangle\}$ are mutually orthogonal for the set of nondegenerate eigenvalues. In the case of nondegenerate eigenvalues $(-\lambda_\alpha \neq -k_m)$, these eigenvectors are

$$\psi_{\alpha,n} = \langle n | \psi_\alpha \rangle = N_\alpha \frac{a_n}{(k_n - \lambda_\alpha)}, \tag{14}$$

$$\psi_{\alpha,0} = \langle 0 | \psi_\alpha \rangle = N_\alpha, \tag{15}$$

where N_α is a normalization constant. For degenerate eigenvalues, an orthogonal basis can always be chosen using the *Gram–Schmidt* procedure. As will become apparent later, however, these eigenvalues do not contribute to the sampling in the case of a first-passage problem beginning with the unbroken loop $i=0$.

Theorem III.3: *The envelope curve $g(t)$ defined by our method is identical to the first-passage density for a C_N network.*

Proof: Beginning with an unbroken state at $t=0$, the probability the model occupies a given state n at time t is given by

$$p_n(t) = \pi_n \sum_{\alpha=0}^{N} \psi_{\alpha,n} \psi_{\alpha,0} \exp[-\lambda_\alpha t] = \sum_{\alpha=0}^{N} c_{\alpha,n} \exp[-\lambda_\alpha t], \tag{16}$$

where $\psi_{\alpha,i}$ is an eigenvector of $\mathcal{M}$ with eigenvalue $-\lambda_\alpha$. Note that only those $\lambda_\alpha \neq k_i$ contribute; otherwise $\psi_{\alpha,0}=0$. Assuming $\lambda_\alpha < \lambda_{\alpha+1}$, the coefficients satisfy $c_{\alpha,n} < 0$ for $\alpha > n$. Since the partial sum $S_{N,n} = \Sigma_{\alpha=0}^{N} c_{\alpha,n} = \Sigma_{\alpha=0}^{N} \pi_n \psi_{\alpha,n} \psi_{\alpha,0} = \pi_n \langle n | 0 \rangle = 0$, all other partial sums satisfy $S_{\beta,n} = \Sigma_{\alpha=0}^{\beta} c_{\alpha,n} \geq 0$. These observations provide a means of decomposing the probability density into the following discrete mixture with positive coefficients:

$$p_n(t) = \sum_{\alpha=1}^{N} S_{\alpha,n} (\exp[-\lambda_\alpha t] - \exp[-\lambda_{\alpha+1} t]). \tag{17}$$

Since $b_n > 0$, the combined rate of decay to any one of the broken states is given by

$$\frac{dp^B}{dt} \equiv \sum_{(n,m)} \frac{dp_{(n,m)}}{dt} = \sum_{\alpha=0}^{N-1} S_\alpha f_\alpha(t), \tag{18}$$

where

$$S_\alpha = \left(\frac{\lambda_{\alpha+1} - \lambda_\alpha}{\lambda_\alpha \lambda_{\alpha+1}} \right) \sum_{m} \sum_{n \neq m} \pi_{(n,m)} (b_m S_{\alpha,n}) > 0 \tag{19}$$

and

$$f_\alpha(t) = \frac{\lambda_\alpha \lambda_{\alpha+1}}{\lambda_{\alpha+1} - \lambda_\alpha} (\exp[-\lambda_\alpha t] - \exp[-\lambda_{\alpha+1} t]). \tag{20}$$

□

C. Simulation models used for bond networks

Although our methods can in principle sample escape times from any subnetwork of a CTMM graph, we have validated them here for the specific case of breaking networks of bonds due to the importance of this problem for self-assembly modeling. In rule-based models of self-assembly, a simulation is initialized with a set of assembly subunits, each with a complement of prespecified binding sites. As the simulation progresses, the system evolves into a state with an assembly of disjoint networks. The binding interactions between two disconnected pieces of the network usually occurs on a slower scale than individual bond-breaking reactions.[6] For biconnected networks, however, the association rate within a connected network is much larger than the bond-breaking rate since there is no entropy penalty in associating bonds between constituent subunits. Such models allow for a natural partitioning of the state space into subgraphs corresponding to the biconnected components of

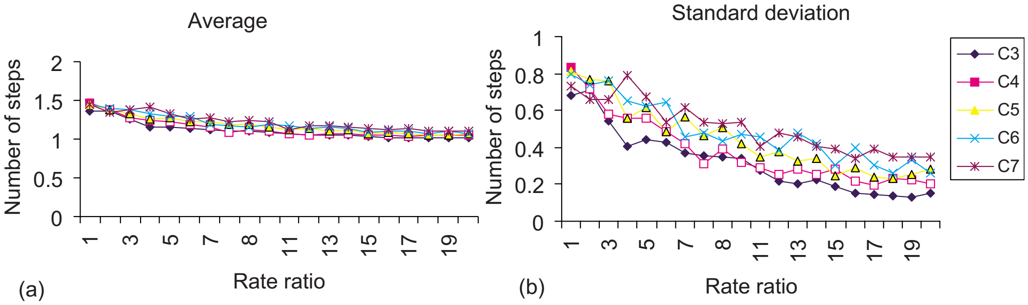

FIG. 8. (Color online) Number of rejection steps for the master equation method until first passage for the network generated by C_N. (a) Average number of steps $\langle s \rangle$. (b) Standard deviation $\sqrt{\langle \delta s^2 \rangle}$.

the entire network. The first set of experiments that we performed was on such biconnected networks. The simplest nontrivial example of a biconnected bond network is the graph generated by an N-cycle (C_N). More complicated networks of N bonds can be viewed as special cases of a truncated unit hypercube in N dimensions. We therefore carried out simulations for the network generated by C_N as well as the full hypercube (Z_N). Theorem III.1 guarantees that the expected number of SSA steps for a k-connected network of d bonds is $P(d,k)\Omega(r^{k-1})$, where $P(d,k)$ is some combinatorial function dependent on the topology of the network.

Each model is parametrized by a rate of bond formation, a, and a rate of bond breaking, b. These values were varied in different simulations. Each of the bonds had different binding/breaking rates but the ratio was maintained at the same order of magnitude for each simulation. Specifically, for a d bond network $b_\mu = b(1.0+0.05\mu/d)$ and $a_\mu = a$. These slight variations in rates from bond to bond were used to avoid giving our methods an unfair advantage, as they will generally be more efficient when the transition matrix has degenerate eigenvalues.

D. Experiments

We conducted a series of simulations to determine the performance of the SSA, master equation, and EMC methods for bond network first-passage times. All simulations were implemented in MATHEMATICA. Run time simulations were executed on a Macintosh machine with a 1.8 GHz G5 processor and 512 Mbyte random access memory. For the EMC based spectral method, we allowed each component of the state vector to converge within a relative error of $\epsilon = 0.01$. Each data point reported was the average over 500 simulations except for run time data, which were averaged over 100 simulations.

We first examined the efficiency of the master equation method by assessing the number of rejection steps needed to sample each first-passage time. We carried out simulations for cycle graphs (C_N) varying the cycle length from 3 to 7 and the rate ratio a/b from 1 to 20 in increments of 1. These experiments were then repeated for unit hypercubes (Z_N) with dimension varied from 2 to 5 and rate ratio a/b from 1 to 10 in increments of 1. For each condition, we recorded the number of rejection steps required for each of 500 simulations and computed the mean and standard deviation across the 500 trials.

We next examined the number of steps required by the EMC method for sampling times to network breakage. We examined the same models as those used to validate the master equation method: cycles of length of 3–7 with rate ratios from 1 to 20 in increments of 1 and hypercubes of dimensions 2–5 with rate ratios from 1 to 10 in increments of 1. We similarly recorded the number of EMC steps required for each of 500 simulations and computed the mean and standard deviation across the 500 trials. We also computed the fraction of models that reached the first-passage time before relaxing to the slowest decay mode.

We next tested the total run time of each of the three methods on a broader set of parameter ranges. We evaluated run times for each method for cycle networks of sizes of 3–7. We performed two sets of evaluations for each. The first set varied the rate ratio a/b from 500 to 5000 in increments of 500 to provide a broad view of the relative run times of the three methods. These numbers span ranges of values likely for protein assemblies. For example, Zlotnick *et al.*[23] estimated a binding free energy of $\Delta G = 4.2$ kcal/mole for ordinary differential equation based simulation of the kinetics of the Hepatitis B virus, which yields $a/b = \exp(\Delta G/RT) \sim 1200$. We then examined ratios of SSA to master equation and SSA to EMC run times for each data point based on averages over 100 simulations per parameter set. In a second set of experiments, designed to give a finer view of where each method is dominant in parameter space, we varied the rate ratio a/b from 30 to 300 in increments of 30. We then identified the most efficient of the three methods for each point, again using averaged run times over 100 trials per data point.

We then performed analogous experiments for hypercube graphs in order to test performance on networks with higher connectivity. For each graph Z_2 to Z_5, we carried out simulations for rate ratio a/b from 3 to 30 in increments of 3. We were limited to small ratios because the SSA method becomes prohibitively costly for high-connectivity networks at higher ratios. Each simulation was repeated 100 times to yield average run times for each parameter set and for each of the three methods. For each parameter set, we computed the ratio of run times for SSA versus master equation and SSA versus EMC. We further evaluated which of the three methods produced the shortest average run time for each parameter set.

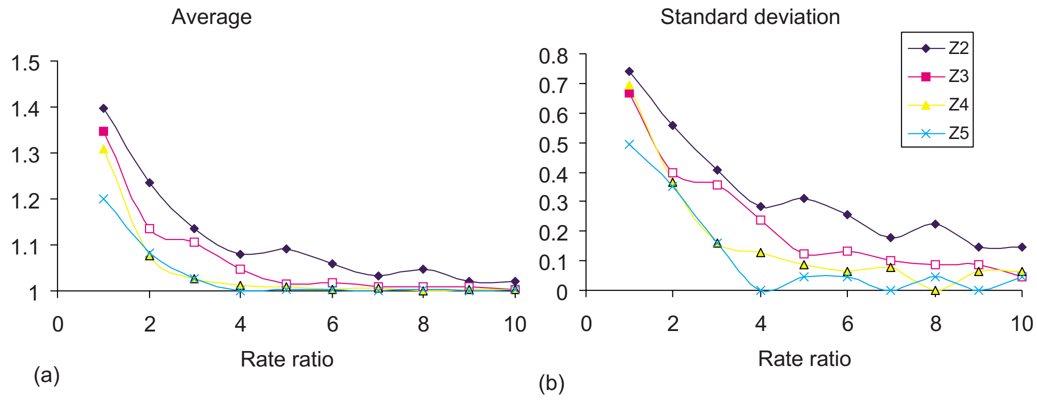

FIG. 9. (Color online) Number of rejection steps for the master equation method until first passage for Z_N. (a) Average number of steps $\langle s\rangle$. (b) Standard deviation $\sqrt{\langle\delta s^2\rangle}$.

E. Results

We first present results on the efficiency of the rejection sampling scheme for the master equation method. The expected run time of the method is proportional to the expected number of trials needed to produce a successful sample. A low number of steps is therefore preferable, with a value of 1 being ideal. Figure 8(a) shows the rejection ratio for cycle graphs C_3 to C_7. The mean number of rejection steps is consistently below 1.5, as expected from Theorems III.3 and II.1. The number of rejection steps drops with increasing rate ratio but increases with increasing cycle length. These results together establish the efficiency of the method. Figure 8(b) shows that the method is also robust, with standard deviation consistently below 0.9 for the experiments shown here. The standard deviation also decreases with increasing rate ratio but increases with cycle size.

Figure 9(a) shows mean numbers of rejection steps for hypercube graphs. Since the envelope curve for hypercubes is not exact, these experiments provide information about how well the method performs for more general networks. The hypercube graphs also yield mean numbers of rejection steps consistently below 1.5. The number of steps generally falls with increasing rate ratio. Figure 9(b) shows the method also to be robust for hypercube graphs, with standard deviations consistently below 1.0 and following similar trends to the means.

Next, we performed identical experiments to study the performance of the EMC-based spectral method. Figure 10(a) shows mean numbers of EMC steps for cycle graphs. The number of steps remains consistently below 6. The values rise sharply at the lowest rate ratios but quickly level off to approximately 4–5 depending on the cycle length. Figures 10(b) and 10(c) provide the explanation for this feature. For small rate ratio, multiple eigenmodes are responsible for the decay [see part (c)], which corresponds to increasing EMC steps before first passage, similar to SSA. However, as rate ratio increases further, relaxation time to the slowest eigenmode becomes smaller than the average first-passage time and the method automatically samples breaking times according to the slowest eigenmode [Fig. 10(c)]. This feature is evident in part (b) of the figures, which measure the standard deviation. At high rate ratio the "trajectory" is almost deterministic, i.e., it always takes the same number of steps to break the network. This happens because the state almost always relaxes to the slowest eigenmode before escaping the subgraph, hence giving a low value for σ at high rate ratio.

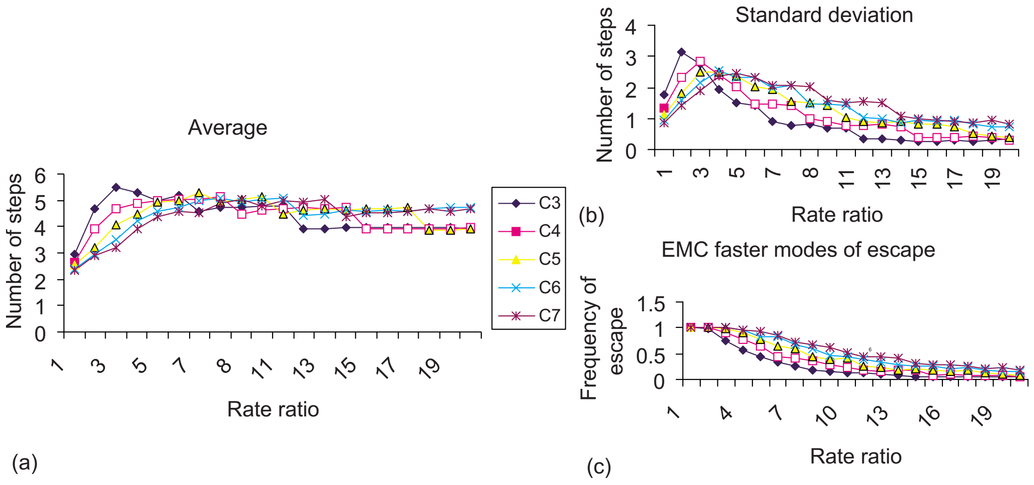

FIG. 10. (Color online) Number of EMC steps until first passage for the network generated by C_N. (a) Average number of steps $\langle s\rangle$. (b) Standard deviation $\sqrt{\langle\delta s^2\rangle}$. (c) Fraction of times the trajectory escapes before relaxing to the slowest decay mode.

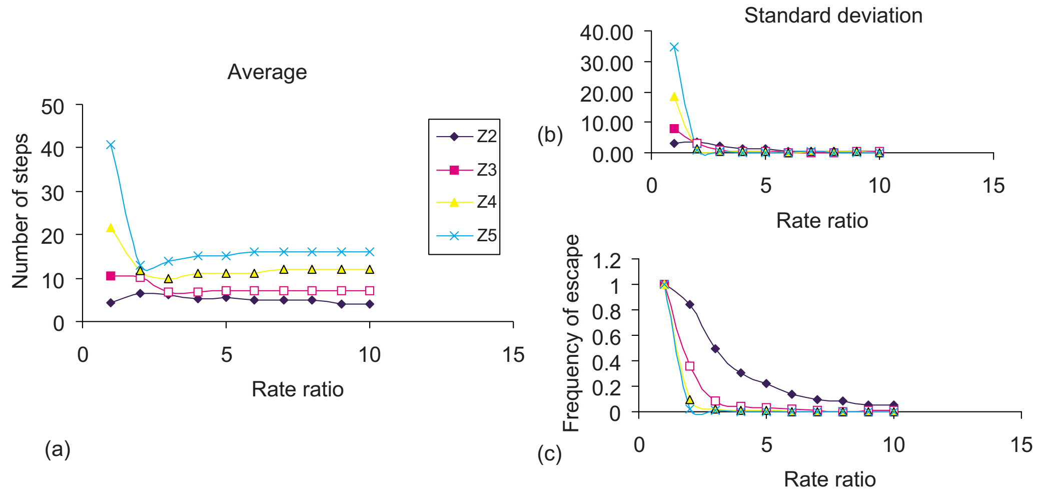

FIG. 11. (Color online) Number of EMC steps until first passage on Z_N. (a) Average number of steps $\langle s \rangle$. (b) Standard deviation $\sqrt{\langle \delta s^2 \rangle}$. (c) Fraction of times the trajectory escapes before relaxing to the slowest decay mode.

Figure 11 shows comparable results for hypercube graphs. Figure 11(a) shows that mean numbers of steps drop substantially between ratios of 1 and 2 but quickly level off to an apparent constant for each graph. The number of steps increases with increasing hypercube dimension. Figures 11(b) and 11(c) again show that the method has high variability for low rate ratios, where multiple eigenmodes contribute significantly to the time distribution and the method must behave similarly to the standard SSA. At higher ratios, though, the slowest mode quickly dominates and the number of steps required becomes highly reproducible.

We next examined total run times of the three methods, beginning with the cycle graphs C_3 to C_7. Figure 12 plots results of the EMC and master equation methods relative to the basic SSA. Figure 12(a) shows ratios of run times for standard SSA to the master equation method. The ratio grows rapidly with increasing rate ratio, although it falls with increasing cycle size. Figure 12(b) shows the comparison of SSA to the EMC method. The SSA:EMC ratio likewise peaks for large rate ratios and small cycle sizes. The EMC method appears generally superior to the master equation method, beginning to dominate at a lower rate ratio and reaching a higher peak. Figure 12(c) shows for a narrower rate range where each of the three methods dominates. The EMC method is the fastest for most of the range examined, with the standard SSA superior at the extreme of low ratios and large cycle sizes.

We then examined run times on the hypercube graphs Z_2 to Z_5. Figure 13(a) shows run time ratios for SSA versus the master equation method and Fig. 13(b) for SSA versus the EMC method. Both spectral methods show sizable improvements over the pure SSA method for larger rate ratios and higher hypercube dimensions. SSA appears much more sensitive to rate ratio as compared to the spectral methods. Even for a rate ratio of 30, the spectral methods were more than

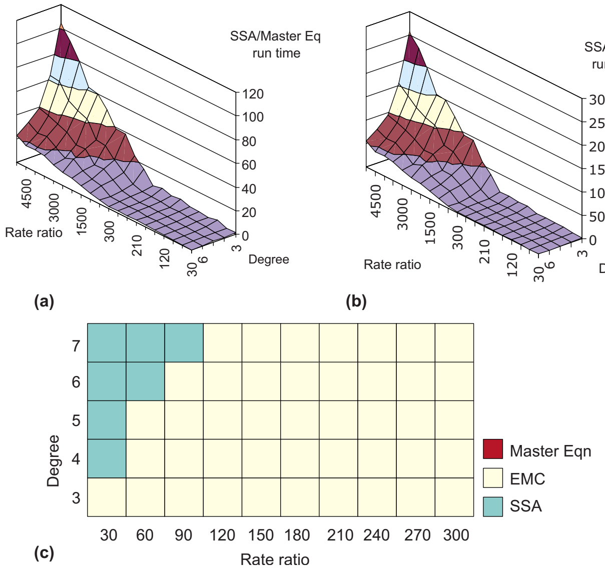

FIG. 12. (Color online) Comparative run times for the network generated by C_N. (a) Ratio of SSA to master equation run times. (b) Ratio of SSA to EMC run times. (c) Region in two dimensional (2D) parameter space where each method is optimal.

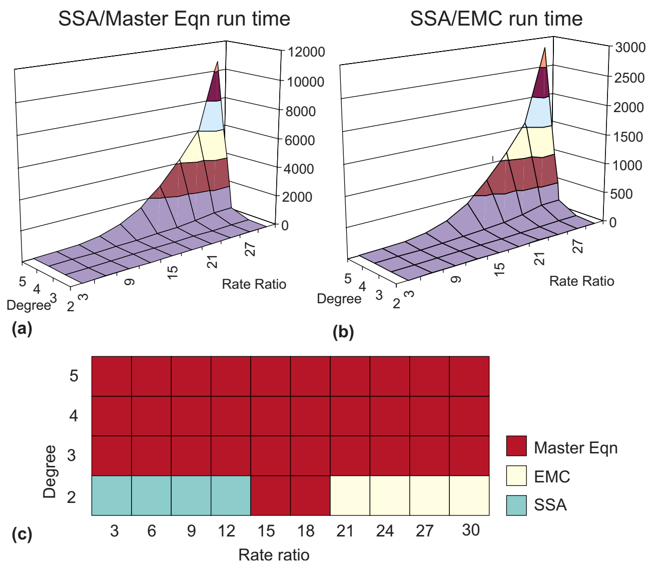

FIG. 13. (Color online) Comparative run times for first passage on Z_N. (a) Ratio of SSA to master equation run times. (b) Ratio of SSA to EMC run times. (c) Region in 2D parameter space where each method is optimal.

three orders of magnitude more efficient than SSA for Z_5. For hypercubes, unlike cycle graphs, the master equation method appears generally more efficient than the EMC-based method, even for small rate ratios. Figure 13(c) shows where each method is dominant. The master equation method is dominant for most of the parameter range examined, with the SSA method superior at the limit of lowest degree and smallest rate ratios and the EMC dominant for low degree and higher rate ratios. This result is expected from Fig. 7(a), since the average number of steps seems to increase monotonically with the connectivity of the graph for the EMC-based method. The efficiency of the master equation method, on the other hand, depends primarily on the size of the complete graph, since matrix diagonalization is the eventual efficiency bottleneck.

IV. APPLICATION WITH AD: NUCLEATION-LIMITED ASSEMBLY

In this section, we apply the AD variants of the methods to a different system type also motivated by self-assembly modeling. The rate of self-assembly processes is often limited by the time required to build the first stable multisubunit complex, called a nucleus, which then acts as a seed for assembly of the rest of a larger structure. Because partially formed nuclei are unstable, considerable trial and error may be needed before one reaches completion. The time to complete a single nucleus can thus be orders of magnitude longer than the intersubunit binding rate. These nucleation-limited assembly systems are one example of the broader class of stiff models for chemically reacting species. The state space of any such system can be represented as a lattice corresponding to the population individual species. These models are similar to those treated in earlier studies of accelerated SSA methods.[14,18] We apply one such model, representing the formation of simple trimeric nuclei, to demonstrate and evaluate the AD variants of the spectral methods.

A. Integer lattice models

The second model we consider is again an assembly of bond networks where monomers (m) with two identical binding sites combine to form dimers (d) and trimers (t). In order to show stiffness with respect to a single parameter, the trimers were assumed to be completely stable. If the total number of monomer subunits is N, the state space is the intersection of the plane $N_m+N_d+N_t=N$ with the positive octant of the three dimensional lattice formed by integer counts of the monomer (N_m), dimer (N_d), and trimer (N_t) populations. Let us represent each vertex of this graph by the pair (N_t,N_d). The reaction propensities $\alpha_{N_t,N_d}^{N_t',N_d'}$ to reach the vertex (N_t',N_d') from (N_t,N_d) are (to within an overall constant)

$$\alpha_{N_t,N_d}^{N_t,N_d+1} = N_m(N_m-1)/v, \tag{21}$$

$$\alpha_{N_t,N_d}^{N_t,N_d-1} = N_d, \tag{22}$$

$$\alpha_{N_t,N_d}^{N_t+1,N_d-1} = N_m N_d/v, \tag{23}$$

$$\alpha_{N_t,N_d}^{N_t-1,N_d+1} = 0, \tag{24}$$

where $1/v$ is an entropy penalty due to the finite volume of the system. We initialize the system at the state (0,0) for a given monomer count N and sample the first-passage time until the trimer count reaches a given value. This system will show stiffness if the parameter $\rho \equiv N/v$ is small. For small ρ, which corresponds to low concentration and/or small binding energy, trimer formation will be much slower than dimer breaking/binding reactions.

```
Algorithm:Automated Discovery
Input: Discovered vertex set $K$, current state $i$, time elapsed $t$
Output: Final state vertex $i \notin V$, first passage time $t$

Do Spectral Sampling over $K$;
$i \leftarrow$ Next state outside $K$;
$t \leftarrow$ updated time;
if $i \in V$ then
    $K \leftarrow$ Enlarge Graph($K, i$);
    Automated Discovery($K, i, t$);
end
else
    Return $i, t$;
end

Algorithm:Enlarge Graph
Input: Discovered vertex set $K = \{0, 1, \ldots N_2\}$, current state $i$
Output: Enlarged vertex set $K$

$N_2 \leftarrow 1 + [m * N_2]$;
Return $K \leftarrow \{0, 1, \ldots N_2\}$;
```

FIG. 14. Pseudocode for AD for a simple path.

B. AD for integer lattice

Efficient simulation over an integer lattice, where one pair of species reacts on a much faster timescale than the others, requires a partitioning of the entire lattice into subgraphs with fixed trimer count (since trimer formation occurs on a much longer timescale than monomer-dimer reactions). These subgraphs are simple paths with vertex set $V(N_t)=\{(N_t,0),(N_t,1),\ldots,(N_t,[(N-3N_t)/2])\}$, where N_t represents the fixed trimer count and square brackets represent the largest integer smaller than the enclosed expression. Figure 14 presents a procedure for implementing AD on such graphs which works in optimal time to within a small constant factor (the vertices are represented by dimer count for simplicity). At each step of AD the method enlarges the graph by a factor proportional to its present length. The scale factor m can be optimized for any given sampling algorithm to optimize run time. For example, if a given spectral sampling algorithm works in time $f(N)=N^\alpha$, where N is the cardinality of the vertex set; the total number of steps used in AD of a graph sized N_i can be bounded from above by the following quantity S_{N_i}:

$$S_{N_i} = \sum_{n=1}^{1+\log_m[N_i]} m^{n\alpha} \le \frac{N_i^\alpha m^{2\alpha}-1}{m^\alpha-1} \le C N_i^\alpha \qquad \text{for } m = 2^{1/(\alpha+1)}, \tag{25}$$

where $C=2^{2\alpha/\alpha+1}/(2^{\alpha/\alpha+1}-1)$. In general, the master equation based method works in $O(N^3)$, however, for simple paths the

Kolmogorov matrix is sparse and effective power is expected to be more like $\alpha=2$. Since S_{N_i} is more sensitive to deviations for smaller values of m, we chose $m=1.3>2^{1/3}$ in the experiments reported here.

The method reported here can in principle be generalized for arbitrary lattice graphs in d dimensions. The size of the discovered graph in such cases would overestimate the actual trapped graph by a factor of m^d for a scaling factor m. For small dimensions, this may still be more efficient than the method discussed in Sec. II D which exactly samples the trapped subgraph.

C. Experiments

We performed two sets of experiments to compare the performance of spectral methods with SSA. The first set of experiments compared the master equation method implemented in conjunction with AD for the trimer model with SSA. Each experiment compared the ratio of run times for sampling first-passage times to reach a trimer count $N_t=100$, starting from an initial monomer count N. The state space was partitioned into subgraphs corresponding to fixed trimer counts and AD was used to identify the trapped regions for spectral decomposition. We then performed a total of 50 comparative run time simulations varying N from 1000 to 9000 in steps of 2000 and varying ρ from 10^{-5} to 1.9×10^{-4} in steps of 2.0×10^{-5}. All run times were averaged over 50 samples. The scale factor for AD was set at 1.3. The second set of experiments compared the run time ratio for the EMC based method and SSA for first-passage time to reach a trimer count $N_t=100$, starting from an initial monomer count N. The state space was again partitioned into subgraphs of fixed trimer counts. AD was not required for these simulations since the method automatically selects the trapped region of the subgraph according to the evolving probability distribution. We then performed a total of 25 comparative run time simulations varying N from 1000 to 9000 in steps of 2000 and varying ρ from 10^{-4} to 0.9×10^{-3} in steps of 2.0×10^{-4}. All run times were averaged over 50 samples. For the spectral method, each component of the slowest eigenvector was allowed to relax to within a relative error of 0.01 and an absolute error of 1.0×10^{-6}.

D. Results

We first present results for the master equation method run times as we vary the stiffness parameter ρ. Figure 15(a) shows the behavior for five different initial monomer counts N. Small ρ values correspond to a stiff model, since the average dimer count varies approximately as $N_d\sim\rho N_m$ and the ratio of the rate of dimer formation to trimer formation varies as $\sim N_m/N_d$. Figure 15(a) demonstrates the efficiency of the master equation method. The method shows large gains in the domain of small ρ and small N, with relative performance dropping rapidly with increasing ρ and more slowly with increasing N. Next we present results for the ratio of SSA/EMC method run times as we vary ρ and N. Figure 15(b) shows the behavior for five different initial monomer counts N. The EMC method is effective at substantially larger values of ρ than is the master equation method.

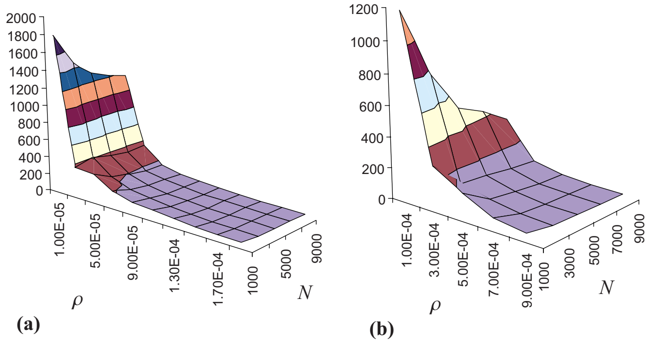

FIG. 15. (Color online) (a) Comparative run times for first passage for ten trimer counts. Ratio of SSA to ME-AD run times. (b) Comparative run times for first passage for 100 trimer counts. Ratio of SSA to EMC-AD run times.

As is to be expected, the spectral methods do not scale as well as the usual SSA method with increasing N. Even for relatively large networks, though, the performance gain obtained by spectral sampling is appreciable. The reason for this is that for most cases the slowest eigenmode is reasonably well approximated by a vector populating only a small fraction of the subgraph vertices. As a result we can look at the EMC method as a generalization of other accelerated sampling schemes which only use one vertex, the mean value of the slowest decay mode, as in the PEA based methods.

V. DISCUSSION

We have investigated the problem of efficiently simulating stochastic reaction models and introduced two methods for accelerating sampling on problems characterized by multiple time scales. Both methods are based on spectral analysis of CTMMs equivalent to the SSA model. We have applied these methods in the present work to two special cases of these models that are important to simulations of molecular self-assembly: sampling times to break multiply connected bond networks and simulating growth in nucleation-limited assembly systems. Collectively, these two applications demonstrate the use of the proposed spectral methods on small CTMM graphs known *a priori* and on automatically discovered subgraphs of large CTMMs. We have shown theoretically and empirically that the new methods are substantially more robust to variations in the ratios of reaction rates than is the basic SSA method for these problems.

While we have applied these methods here to models used in self-assembly simulations, the basic methods can be expected to have much broader application. Both methods can be applied to sample first-passage times for any arbitrary subset of states of any SSA CTMM graph. Both can also be applied to sample escape times from any subgraph of such a graph, using AD to identify trapped regions of the CTMM graph. The latter distinction is important because CTMM graphs for complicated biological systems are generally far too large to represent explicitly. These spectral methods might be extended to incorporate on the fly graph construction techniques, like those used by rule-based methods widely used for SSA simulations.[9,10] The EMC method, especially, would seem to be a candidate for such an extension. For example, if at each iteration, instead of adding all the possible next neighbors to the system state, we add only a subset of them depending on their transition probabilities, then we will get a natural, nonlocal generalization of the SSA. Such an approach could provide a precise and general method for pruning full SSA graphs to achieve more efficient pathway sampling in extremely large state spaces.

ACKNOWLEDGMENTS

This work was supported in part by U.S. National Science Foundation Award No. 0346981.

APPENDIX: TIME COMPLEXITY OF SSA FOR BOND NETWORKS

In this appendix, we prove Theorem III.1, which helps us establish the relative efficiency of the spectral methods to the standard SSA. Before we prove the theorem, we need to establish some preliminary results. Let $r \equiv \min(a_\mu / b_\nu | \mu, \nu \in \{1, \dots, d\})$. To construct the transition matrix Q, SSA identifies the negative of the diagonal element of the Kolmogorov matrix $-W_{\boldsymbol{n},\boldsymbol{n}} = (\Sigma_\beta a_\beta n_\beta + (1-n_\beta) b_\beta)$ as the inverse of the mean waiting time at each SSA step and the matrix $L_{\boldsymbol{m},\boldsymbol{n}} = -W_{\boldsymbol{m},\boldsymbol{n}} / W_{\boldsymbol{n},\boldsymbol{n}}$ as the graph Laplacian. Therefore, $Q_{\boldsymbol{m},\boldsymbol{n}} = \delta_{\boldsymbol{m},\boldsymbol{n}} - L_{\boldsymbol{m},\boldsymbol{n}}$. Since SSA simulates a periodic Markov chain Q, the graph is bipartite and the two step chain Q^2 is reducible into $Q^2_{\text{even}} \oplus Q^2_{\text{odd}}$. Here, Q^2_{even} is the projection of Q^2 over the subspace of states with an even number of bonds broken V_{even} and Q^2_{odd} is the projection over $V_{\text{odd}} = S - V_{\text{even}}$. Since both Q^2_{even} and Q^2_{odd} are irreducible and aperiodic, the ergodic theorem applies to each one separately, and if $|\Pi\rangle = \Sigma_{i \in S} \pi_{\boldsymbol{i}} |\boldsymbol{i}\rangle$ is the eigenvector of Q with eigenvalue of 1, the vectors $|\Pi_e\rangle = \Sigma_{\boldsymbol{i} \in V_{\text{even}}} \pi_{\boldsymbol{i}} |\boldsymbol{i}\rangle$ and $|\Pi_o\rangle = \Sigma_{\boldsymbol{i} \in V_{\text{odd}}} \pi_{\boldsymbol{i}} |\boldsymbol{i}\rangle$ are the equilibrium distributions for Q^2_{even} and Q^2_{odd}, respectively, up to a normalization constant. To bound the mean hitting time $T_{\boldsymbol{b}0}$ from $\boldsymbol{0}$ to the set V_b, we first apply the common technique of constructing another graph with vertex set $\bar{V} = V_c \cup \{\boldsymbol{b}\}$, where all vertices in V_b are truncated to a single vertex $\boldsymbol{b}$ and $V_c = S - V_b$. The edge weights for edges from $\boldsymbol{i} \in V_c$ to $\boldsymbol{b}$ are chosen as $Q_{\boldsymbol{b},\boldsymbol{i}} = \Sigma_{\boldsymbol{j} \in V_b} Q_{\boldsymbol{j},\boldsymbol{i}}$, which will leave $T_{\boldsymbol{b}0}$ unchanged from that of the original graph. We must further specify the edge weights from $\boldsymbol{b}$ to any states with $k-1$ broken bonds. In order to ensure that the Markov chain still obeys detailed balance, we require that $Q_{\boldsymbol{i},\boldsymbol{b}} / Q_{\boldsymbol{j},\boldsymbol{b}} = (Q_{\boldsymbol{b},\boldsymbol{i}} \pi_{\boldsymbol{i}}) / (Q_{\boldsymbol{b},\boldsymbol{j}} \pi_{\boldsymbol{j}})$ and $\Sigma_{\boldsymbol{i} \neq \boldsymbol{b}} Q_{\boldsymbol{i},\boldsymbol{b}} = 1$. The resulting modified graph will then have the same hitting time $T_{\boldsymbol{b0}}$ as the original graph.

We next need three auxiliary results about properties of the resulting graph in order to prove our main theorem.

Lemma A.1: *For an ergodic Markov chain, the cover time* $C_{\boldsymbol{ij}} \equiv T_{\boldsymbol{ij}} + T_{\boldsymbol{ji}}$ *between any two states* $\boldsymbol{i}$ *and* $\boldsymbol{j}$ *satisfies*[24]

$$E[C_{\boldsymbol{ij}}] = E[T_{\boldsymbol{ij}}] + E[T_{\boldsymbol{ji}}] = 1/(\pi_{\boldsymbol{j}} Pr[T_{\boldsymbol{jj}} > T_{\boldsymbol{ij}}]). \tag{A1}$$

Lemma A.2: *The transition matrix Q satisfies the following conditions.*

(1) *If* $\boldsymbol{i}$ *and* $\boldsymbol{j} = \boldsymbol{i} + \hat{\boldsymbol{\mu}}$ *are two neighboring states with n and* $n+1$ *bonds broken, respectively, then* $Q_{\boldsymbol{j},\boldsymbol{i}} \leq (nr)^{-1}$ *for any* $n > 0$.

(2) *For any initial state* $\boldsymbol{i}$ *containing n broken bonds, the* n *-step transition probability to* $\boldsymbol{0}$ *is bounded from below by* $Q^n_{\boldsymbol{0},\boldsymbol{i}} \geq (1 + d/r)^{-n}$.

(3) *Let T be any stopping time for the transition matrix Q with expectation value* $E[T] = \Sigma_{n=1} n Pr[T = n] = \Sigma_{n=0} Pr[T > n]$. *For any integer* $l > 1$ *consider the expectation value of T for the* l *-step transition matrix* Q^l *defined as* $E^{(l)}[T] = \Sigma_n n Pr[(n-1)l < T \leq nl] = \Sigma_{n=0} Pr[T > nl]$. *Then,* $l(E^{(l)}[T] - 1) \leq E[T] \leq l E^{(l)}[T]$.

*Proof***:**

(1) The transition probability corresponding to the matrix element connecting $\boldsymbol{i}$ to $\boldsymbol{j}$ is

$$Q_{\boldsymbol{j},\boldsymbol{i}} = \frac{b_\mu}{\left(\sum_\beta a_\beta i_\beta + (1 - i_\beta) b_\beta\right)} \leq \begin{cases} (nr)^{-1} & \forall \boldsymbol{i} \neq \boldsymbol{0} \\ 1 & \text{if } \boldsymbol{i} = \boldsymbol{0}. \end{cases}$$

(2) First consider any state $\hat{\boldsymbol{\mu}}$ with one bond broken,

$$Q_{\boldsymbol{0},\hat{\boldsymbol{\mu}}} = \frac{a_\mu}{a_\mu + \sum_{\nu \neq \mu} b_\nu} \geq (1 + d/r)^{-1}. \tag{A2}$$

Assume $Q^n_{\boldsymbol{0},\Sigma^n_{i=1}\hat{\boldsymbol{\mu}}_i} \geq (1 + d/r)^{-n}$ for all n broken bond states $\Sigma^n_{i=1} \hat{\boldsymbol{\mu}}_i$, then

$$\begin{aligned} Q^{n+1}_{\boldsymbol{0},\Sigma^{n+1}_{i=1}\hat{\boldsymbol{\mu}}_i} &= \sum_{\nu \in \{\mu_1, \dots, \mu_{n+1}\}} Q^n_{\boldsymbol{0},\Sigma_{\mu_i \neq \nu}\hat{\boldsymbol{\mu}}_i} Q_{\sum_{\mu_i \neq \nu} \hat{\boldsymbol{\mu}}_i, \Sigma^{n+1}_{i=1}\hat{\boldsymbol{\mu}}_i} \\ &\geq (1 + d/r)^{-n} \sum_{\nu \in \{\mu_1, \dots, \mu_{n+1}\}} \frac{a_\nu}{\sum^{n+1}_{i=1} a_{\mu_i} + \sum_{\eta \neq \{\mu_i\}} b_\eta} \\ &\geq (1 + d/r)^{-n-1}. \end{aligned} \tag{A3}$$

Since $Q_{\boldsymbol{0},\hat{\boldsymbol{\mu}}} > (1 + d/r)^{-1}$, the assertion holds for all $n > 1$ by induction.

(3) We can prove the upper bound as follows:

$$\begin{aligned} E[T] &= \sum_{n=1} \left(\sum^{l}_{m=1} ((n-1)l + m) Pr[T = ((n-1)l + m)] \right) \\ &\leq \sum_{n=1} nl \left(\sum^{l}_{m=1} Pr[T = ((n-1)l + m)] \right) \\ &\leq l \sum_{n=1} n Pr[(n-1)l < T \leq nl] \leq l E^{(l)}[T]. \end{aligned} \tag{A4}$$

We can similarly prove the lower bound,

$$\begin{aligned} E[T] &= \sum_{n=1} \left(\sum^{l}_{m=1} ((n-1)l + m) Pr[T = ((n-1)l + m)] \right) \\ &\geq \sum_{n=1} (n-1) l \left(\sum^{l}_{m=1} Pr[T = ((n-1)l + m)] \right) \\ &\geq l \sum_{n=1} n \, Pr[(n-1)l < T \leq nl] - l \geq l(E^{(l)}[T] - 1). \end{aligned} \tag{A5}$$

□

Lemma A.3: *The expected hitting time from the vertex* $\boldsymbol{b}$

to $\mathbf{0}$ *is bounded by*

$$k \leq E[T_{\mathbf{0}\boldsymbol{b}}] \leq k(1+d/r)^k. \tag{A6}$$

Proof: The lower bound is trivial since at least k bonds must be repaired before any disconnected state can reach $\mathbf{0}$. Consider the nk step probability for transition from $\boldsymbol{b}$ to $\mathbf{0}$. Let $\tilde{Q}$ be the transition matrix restricted to the set $\tilde{V}=\bar{V}-\{\mathbf{0}\}$, i.e.,

$$\tilde{Q}_{\boldsymbol{i},\boldsymbol{j}} = Q_{\boldsymbol{i},\boldsymbol{j}}(1-\delta_{\mathbf{0}\boldsymbol{j}}-\delta_{\boldsymbol{i}\mathbf{0}}+\delta_{\mathbf{0}\boldsymbol{j}}\delta_{\boldsymbol{i}\mathbf{0}}). \tag{A7}$$

The probability of a trajectory starting at $\boldsymbol{i}\in\tilde{V}$ reaching $\mathbf{0}$ in k steps or less is given by

$$\begin{aligned} Pr[T_{\mathbf{0}\boldsymbol{i}} \leq k] &= 1 - \sum_{\boldsymbol{j}\in\tilde{V}} \tilde{Q}^k_{\boldsymbol{j},\boldsymbol{i}} \\ &= \sum_{n=1}^{k}\left(\sum_{\boldsymbol{l}\in\tilde{V}} Q_{\mathbf{0},\boldsymbol{l}}\tilde{Q}^{n-1}_{\boldsymbol{l},\boldsymbol{i}}\right) \geq Q^k_{\mathbf{0},\boldsymbol{b}} > (1+d/r)^{-k}, \end{aligned} \tag{A8}$$

where we have used Lemma A.2 part (2). Let us define $p=(1+d/r)^{-k}$. In terms of p, the previous inequality and Lemma A.2 part (3) imply

$$\begin{aligned} &Pr[T_{\mathbf{0}\boldsymbol{b}} > nk] = 1 - Pr[T_{\mathbf{0}\boldsymbol{b}} \leq nk] \leq (1-p)^n \\ &\qquad\Rightarrow E^{(k)}[T_{\mathbf{0}\boldsymbol{b}}] \leq \sum_{n=1}^{\infty}(1-p)^n = 1/p, \\ &E^{(k)}[T_{\mathbf{0}\boldsymbol{b}}] \leq (1+d/r)^k \\ &\qquad\Rightarrow E[T_{\mathbf{0}\boldsymbol{b}}] \leq k(1+d/r)^k. \end{aligned} \tag{A9}$$

□

An immediate consequence of the previous lemma is that for k even

$$k/2 \leq E^{(2)}[T_{\mathbf{0}\boldsymbol{b}}] \leq (k/2)(1+d/r)^k + 1. \tag{A10}$$

Similarly, if k is odd, using the fact that $Pr[T_{\mathbf{0}\boldsymbol{b}}<T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}]=0$, we get

$$\frac{k-1}{2} \leq E^{(2)}[T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}] \leq E[T_{\mathbf{0}\boldsymbol{b}}]/2 + 1 \leq \frac{k}{2}(1+d/r)^k + 1. \tag{A11}$$

Let us define the equilibrium probability for $\boldsymbol{b}$ as $\tilde{\pi}_{\boldsymbol{b}}$; then

$$\begin{aligned} \tilde{\pi}_{\boldsymbol{b}} &= \frac{\pi_{\boldsymbol{b}}}{\sum_{\boldsymbol{i}\in V_{\text{even}}}\pi_{\boldsymbol{i}}} \quad \text{if } k \text{ is even} \\ &= \frac{\pi_{\boldsymbol{b}}}{\sum_{\boldsymbol{i}\in V_{\text{odd}}}\pi_{\boldsymbol{i}}} \quad \text{if } k \text{ is odd.} \end{aligned} \tag{A12}$$

We can finally compute upper (U) and lower (L) bounds on the hitting time $T_{\boldsymbol{b}\mathbf{0}}$ which are asymptotically equivalent in the limit $r\to\infty$. The following theorem implies that $\Delta(r)\equiv U(r)-L(r)$ is monotonically decreasing in r and $\lim_{r\to\infty}\Delta(r)/L(r)=0$:[25]

Theorem A.1: *The expected number of SSA steps before first passage on a k-connected graph is bounded within*

$$\begin{aligned} 2\frac{(1-d/r)^{-1}-((k-1)/2)\tilde{\pi}_{\boldsymbol{b}}}{\tilde{\pi}_{\boldsymbol{b}}} &\geq E[T_{\boldsymbol{b}\mathbf{0}}] \\ &\geq 2\frac{1-(k/2(1+d/r)^k+2)\tilde{\pi}_{\boldsymbol{b}}}{\tilde{\pi}_{\boldsymbol{b}}}. \end{aligned} \tag{A13}$$

Proof: In order to apply Lemma A.1 to bound the hitting time, we need to look at graphs with k odd or even separately. If k is even we can apply Lemma A.1 directly to $C_{\mathbf{0}\boldsymbol{b}}$ for the two-step chain Q^2_{even}. However, if k is odd, we need to consider the cover time between $\boldsymbol{b}$ and each state $\hat{\boldsymbol{\mu}}$ with exactly one broken bond. Then, using the fact $Q|\mathbf{0}\rangle=(1/\Sigma_\nu b_\nu)\Sigma_\mu b_\mu|\hat{\boldsymbol{\mu}}\rangle$, we get

$$E[T_{\boldsymbol{b}\mathbf{0}}] = 1 + \frac{1}{\sum_\nu b_\nu}\sum_\mu b_\mu E[T_{\boldsymbol{b}\hat{\boldsymbol{\mu}}}]. \tag{A14}$$

Since $Pr[T_{\boldsymbol{bb}}>T_{\mathbf{0}\boldsymbol{b}}]=\Sigma_{n>m}Pr[T_{\boldsymbol{bb}}=n]Pr[T_{\mathbf{0}\boldsymbol{b}}=m]$, for the k-step chain discussed in Lemma A.3, we get

$$\begin{aligned} &Pr[T_{\boldsymbol{bb}} < T_{\mathbf{0}\boldsymbol{b}}] \leq \sum_{n=1}^{\infty}\left(\frac{d}{r(1+d/r)}\right)^n = d/r \\ &\qquad\Rightarrow Pr[T_{\boldsymbol{bb}} > T_{\mathbf{0}\boldsymbol{b}}] \geq 1-d/r. \end{aligned} \tag{A15}$$

Also, $Pr[T_{\boldsymbol{bb}}>T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}]\geq Pr[T_{\boldsymbol{bb}}>T_{\mathbf{0}\boldsymbol{b}}]\geq 1-d/r$. Suppose k is even. Then we can estimate the cover time $C_{\mathbf{0}\boldsymbol{b}}=T_{\boldsymbol{b}\mathbf{0}}+T_{\mathbf{0}\boldsymbol{b}}$ using Lemma A.1,

$$\begin{aligned} E[T_{\boldsymbol{b}\mathbf{0}}] &\geq 2E^{(2)}[T_{\boldsymbol{b}\mathbf{0}}] - 2 \\ &= 2\left(\frac{1}{\tilde{\pi}_{\boldsymbol{b}}Pr[T_{\boldsymbol{bb}}>T_{\mathbf{0}\boldsymbol{b}}]} - E^{(2)}[T_{\mathbf{0}\boldsymbol{b}}] - 1\right) \leq 2E^{(2)} \\ &\times[T_{\boldsymbol{b}\mathbf{0}}] = 2\left(\frac{1}{\tilde{\pi}_{\boldsymbol{b}}Pr[T_{\boldsymbol{bb}}>T_{\mathbf{0}\boldsymbol{b}}]} - E^{(2)}[T_{\mathbf{0}\boldsymbol{b}}]\right). \end{aligned} \tag{A16}$$

An analogous argument for odd k on using Eq. (A14) gives

$$\begin{aligned} E[T_{\boldsymbol{b}\mathbf{0}}] &\geq 1 + \frac{1}{\sum_\nu b_\nu}\sum_\mu b_\mu\left(\frac{2}{\tilde{\pi}_{\boldsymbol{b}}Pr[T_{\boldsymbol{bb}}>T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}]} - 2E^{(2)}\right. \\ &\left.\times[T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}] - 2\right) \leq 1 \\ &+ \frac{1}{\sum_\nu b_\nu}\sum_\mu b_\mu\left(\frac{2}{\tilde{\pi}_{\boldsymbol{b}}Pr[T_{\boldsymbol{bb}}>T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}]} - 2E^{(2)}\right. \\ &\left.\times[T_{\hat{\boldsymbol{\mu}}\boldsymbol{b}}]\right). \end{aligned} \tag{A17}$$

Finally, using Lemmas A.1 and A.3 we get for all k

$$\begin{aligned} E[T_{\boldsymbol{b}\mathbf{0}}] &\geq 2\frac{1-(k/2(1+d/r)^k+2)\tilde{\pi}_{\boldsymbol{b}}}{\tilde{\pi}_{\boldsymbol{b}}} \\ &\leq 2\frac{(1-d/r)^{-1}-((k-1)/2)\tilde{\pi}_{\boldsymbol{b}}}{\tilde{\pi}_{\boldsymbol{b}}}. \end{aligned} \tag{A18}$$

□

As a corollary to the preceding theorem we get the result stated in Sec. III.

Theorem III.1: The expected number of SSA steps required to break a k-connected network with $k>1$ and $r>1$ is $\Omega(r^{k-1})$.

Proof: Let $\boldsymbol{i}$ and $\boldsymbol{j}=\boldsymbol{i}+\hat{\boldsymbol{\mu}}+\hat{\boldsymbol{\nu}}$ be two graphs with c and $c+2$ bonds broken, respectively. Since we are interested in computing the invariant distribution for the irreducible components Q^2_{even} and Q^2_{odd}, we first compute each matrix element connecting $\boldsymbol{i}$ to $\boldsymbol{j}$,

$$Q^2_{\boldsymbol{j},\boldsymbol{i}}=\sum_{p=\mu,\nu}Q_{\boldsymbol{j},\boldsymbol{i}+\hat{\boldsymbol{p}}}Q_{\boldsymbol{i}+\hat{\boldsymbol{p}},\boldsymbol{i}}=\sum_{p=\mu,\nu}\left(\frac{b_\mu b_\nu}{\left(\sum_\alpha a_\alpha(i_\alpha+\delta_{p\alpha})+(1-i_\alpha-\delta_{p\alpha})b_\alpha\right)\left(\sum_\beta a_\beta i_\beta+(1-i_\beta)b_\beta\right)}\right)$$

$$=\frac{b_\mu b_\nu}{W_{\boldsymbol{i},\boldsymbol{i}}}\left(\frac{1}{W_{\boldsymbol{i}+\hat{\boldsymbol{\mu}},\boldsymbol{i}+\hat{\boldsymbol{\mu}}}}+\frac{1}{W_{\boldsymbol{i}+\hat{\boldsymbol{\nu}},\boldsymbol{i}+\hat{\boldsymbol{\nu}}}}\right). \qquad (A19)$$

Similarly,

$$Q^2_{\boldsymbol{i},\boldsymbol{j}}=\sum_{p=\mu,\nu}Q_{\boldsymbol{i},\boldsymbol{j}-\hat{\boldsymbol{p}}}Q_{\boldsymbol{j}-\hat{\boldsymbol{p}},\boldsymbol{j}}=\sum_{p=\mu,\nu}\left(\frac{a_\mu a_\nu}{\left(\sum_\alpha a_\alpha(j_\alpha-\delta_{p\alpha})+(1-j_\alpha+\delta_{p\alpha})b_\alpha\right)\left(\sum_\beta a_\beta j_\beta+(1-j_\beta)b_\beta\right)}\right)$$

$$=\frac{a_\mu a_\nu}{W_{\boldsymbol{j},\boldsymbol{j}}}\left(\frac{1}{W_{\boldsymbol{i}+\hat{\boldsymbol{\mu}},\boldsymbol{i}+\hat{\boldsymbol{\mu}}}}+\frac{1}{W_{\boldsymbol{i}+\hat{\boldsymbol{\nu}},\boldsymbol{i}+\hat{\boldsymbol{\nu}}}}\right). \qquad (A20)$$

Detailed balance then implies that

$$\frac{\pi_{\boldsymbol{j}}}{\pi_{\boldsymbol{i}}}=\frac{Q^2_{\boldsymbol{j},\boldsymbol{i}}}{Q^2_{\boldsymbol{i},\boldsymbol{j}}}=\frac{b_\mu}{a_\mu}\frac{b_\nu}{a_\nu}\frac{W_{\boldsymbol{j},\boldsymbol{j}}}{W_{\boldsymbol{i},\boldsymbol{i}}}=\frac{b_\mu}{a_\mu}\frac{b_\nu}{a_\nu}\left(1+\frac{(a_\mu+a_\nu)-(b_\mu+b_\nu)}{-W_{\boldsymbol{i},\boldsymbol{i}}}\right)$$

$$\leq\frac{c+2}{c}r^{-2}\quad\text{if }\ c\neq 0. \qquad (A21)$$

Since $\pi_{\hat{\boldsymbol{\mu}}}=\pi_{\boldsymbol{0}}b_\mu(a_\mu+\Sigma_{\nu\neq\mu}b_\nu)/(b_\mu+\Sigma_{\nu\neq\mu}b_\nu)a_\mu<\pi_{\boldsymbol{0}}$ we can deduce that for any state $\boldsymbol{i}$ with c bonds broken, with $k-1\geq c\geq 1$, the invariant probability $\pi_{\boldsymbol{i}}\leq cr^{-c+1}\pi_{\boldsymbol{0}}$. Let, $\boldsymbol{l}$ be the state with $k-1$ bonds broken for which π is maximized. The choice of matrix elements imposed by detailed balance implies $Q_{\boldsymbol{l},\boldsymbol{b}}\geq 1/\binom{d}{k-1}$. Also, since Lemma A.2 implies $\Sigma_\mu\pi_{\hat{\boldsymbol{\mu}}}\leq\pi_{\boldsymbol{0}}(1+d/r)$, we get for all values of k

$$\frac{\pi_{\boldsymbol{b}}}{\pi_{\boldsymbol{l}}}=\frac{Q_{\boldsymbol{b},\boldsymbol{l}}}{Q_{\boldsymbol{l},\boldsymbol{b}}}\leq\frac{(d-k+1)\binom{d}{k-1}}{(k-1)r}$$

$$\Rightarrow\tilde{\pi}_{\boldsymbol{b}}\leq\frac{\pi_{\boldsymbol{b}}}{\pi_{\boldsymbol{0}}}\leq(d-k+1)\binom{d}{k-1}r^{-k+1}. \qquad (A22)$$

Finally, using the lower bound on $E[T_{\boldsymbol{b}\boldsymbol{0}}]$ computed in preceding theorem, we get

$$E[T_{\boldsymbol{b}\boldsymbol{0}}]\geq 2\frac{1-(k/2(1+d/r)^k+2)\tilde{\pi}_{\boldsymbol{b}}}{\tilde{\pi}_{\boldsymbol{b}}}$$

$$\geq P(d,k)r^{k-1},\qquad\forall\, r>r_0, \qquad (A23)$$

where $P(d,k)=\left[2/(d-k+1)\binom{d}{k-1}\right]\left(1-(1+(k2^{k-2})^{-1})k(d-k+1)\binom{d}{k-1}/(2/d)^{k-1}\right)$ and $r_0=d$. ☐

[1] C. V. Rao, D. M. Wolf, and A. P. Arkin, Nature (London) **420**, 231 (2002).
[2] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[3] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[4] H. H. McAdams and A. Arkin, Proc. Natl. Acad. Sci. U.S.A. **94**, 814 (1997).
[5] B. Berger, P. W. Shor, L. Tucker-Kellog, and J. King, Proc. Natl. Acad. Sci. U.S.A. **91**, 7732 (1994).
[6] T. Zhang and R. Schwartz, Biophys. J. **90**, 57 (2006).
[7] C. J. Morton-Firth and D. Bray, J. Theor. Biol. **192**, 117 (1998).
[8] K. Takahashi, T. Sakurada, K. Kaizu, T. Kitamaya, S. Arjunam, T. Ishida, G. Bereczki, D. Ito, M. Sugimoto, T. Komori, O. Seiji, and M. Tomita, Genome Informatics **14**, 294 (2003).
[9] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek, Bioinformatics **20**, 3289 (2004).
[10] L. Lok and R. Brent, Nat. Biotechnol. **23**, 131 (2005).
[11] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[12] E. Haseltine and J. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[13] P. Ceres and A. Zlotnick, Biochemistry **41**, 11525 (2002).
[14] R. J. Allen, P. B. Warren, and P. R. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005).
[15] G. E. Crooks and D. Chandler, Phys. Rev. E **64**, 026109 (2001).
[16] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. **123**, 1 (2002).
[17] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **122**, 014116 (2005).
[18] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Comput. Phys. **206**, 395 (2005).
[19] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland, Amsterdam, 1981).
[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2007).
[21] B. Morris and A. Sinclair, SIAM J. Comput. **34**, 195 (2004).
[22] R. A. Horn and C. A. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, 1985).
[23] A. Zlotnick, J. M. Johnson, P. W. Wingfield, S. J. Stahl, and D. Endres, Biochemistry **38**, 14644 (1999).
[24] D. Aldous and J. A. Fill, *Reversible Markov Chains and Random Walks on Graphs*, http://www.stat.berkeley.edu/~aldous/RWG/book.html.
[25] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers* (McGraw-Hill, New York, 1978).