A rigorous framework for multiscale simulation of stochastic cellular networks

Michael W. Chevalier[a] and Hana El-Samad[b]
Department of Biochemistry and Biophysics, California Institute for Quantitative Biosciences, University of California San Francisco, 1700, 4th Street, San Francisco, California 94143-2542, USA

(Received 22 March 2009; accepted 8 July 2009; published online 3 August 2009)

Noise and stochasticity are fundamental to biology and derive from the very nature of biochemical reactions where thermal motion of molecules translates into randomness in the sequence and timing of reactions. This randomness leads to cell-cell variability even in clonal populations. Stochastic biochemical networks are modeled as continuous time discrete state Markov processes whose probability density functions evolve according to a chemical master equation (CME). The CME is not solvable but for the simplest cases, and one has to resort to kinetic Monte Carlo techniques to simulate the stochastic trajectories of the biochemical network under study. A commonly used such algorithm is the stochastic simulation algorithm (SSA). Because it tracks every biochemical reaction that occurs in a given system, the SSA presents computational difficulties especially when there is a vast disparity in the timescales of the reactions or in the number of molecules involved in these reactions. This is common in cellular networks, and many approximation algorithms have evolved to alleviate the computational burdens of the SSA. Here, we present a rigorously derived modified CME framework based on the partition of a biochemically reacting system into restricted and unrestricted reactions. Although this modified CME decomposition is as analytically difficult as the original CME, it can be naturally used to generate a hierarchy of approximations at different levels of accuracy. Most importantly, some previously derived algorithms are demonstrated to be limiting cases of our formulation. We apply our methods to biologically relevant test systems to demonstrate their accuracy and efficiency. © *2009 American Institute of Physics*. [DOI: 10.1063/1.3190327]

I. BACKGROUND AND MOTIVATION

In populations of organisms and cells, phenotypic differences between individuals are caused by a combination of genetic and environmental factors. However, significant variability persists between individuals of the same genotype in highly similar environments. For example, studies of gene expression variability in inbred mouse strains revealed as much interindividual variability within the same genotype as interstrain variability. In human genetic studies, there is substantial variability in the penetrance and expressivity of phenotypes of individuals who share a genotype. While subtle genetic or environmental differences might be at play, stochastic factors account for a significant portion of phenotypic variability. Such variability impacts many biological processes. In Drosophila embryos, variability in the gradient of the bicoid protein results in undesirable developmental alterations, and studies suggest that the system critically depends on molecular devices that are dedicated to filtering noise.[1,2] Other biological systems use heterogeneity productively to allow populations to "hedge their bets" with respect to future environmental shifts. This is thought to explain the ability of subpopulations of bacteria to resist antibiotics and promote latency of infection by viruses.[3,4] As a consequence, the thorough investigation of the principles underlying such nongenetic individuality is essential for understanding the differential susceptibility of cells and organisms to diseases, drugs, and pathogens.

Molecular noise, an important contributor to nongenetic individuality, is a direct consequence of the stochastic nature of biomolecular reactions. Indeed, biochemical reactions are probabilistic collisions of biological molecules whose occurrence depends on the channels of possible interactions between these molecules (structure of the interaction network), the interaction affinity (reaction rates), as well as the number of molecules available for such interactions. This inevitably results in fluctuations in the levels of the various molecular species and translates into cell-to-cell phenotypic differences in cellular behaviors.

Much experimental work has recently focused on the importance of fluctuations and noise in cell dynamics. Investigations ranged from probing the stochastic behaviors of the steps involved in gene expression[5–9] to the effects of post-translational modifications. In some cases, modulating stochastic gene expression has been shown to percolate through the cellular networks, generating wide cell-to-cell differences in the chemotactic behaviors of genetically identical *E. coli*[10] and influencing the excursion into competence in *B. subtilis*.[11] Notably, in most of these studies, computational modeling has emerged as a key step for discovering, describing, understanding, and ultimately generalizing the observed stochastic phenomena.

The traditional description of the stochastic chemical re-

[a]Electronic mail: michael.chevalier@ucsf.edu.
[b]Electronic addresses: hana.el-samad@ucsf.edu and helsamad@biochem.ucsf.edu.

actions that implement biological functions relies on the formulation of the chemical master equation (CME), a differential equation for the time evolution of probabilities.[12–14] The CME corresponds to a continuous time discrete state Markov process model, the sample paths of which can be simulated exactly using a simple Monte Carlo procedure known as the stochastic simulation algorithm (SSA) developed by Gillespie.[15] The SSA has been used with great success for the study of a number of cellular networks. However, one of its most important drawbacks is computational inefficiency; since every reaction that occurs in the system is tracked, dramatic increases in simulation times are frequent when the system contains large concentrations and/or fast reaction rates for some or all of the species involved. This is rather the norm than the exception in biological networks. The number of molecules that interact in biological networks spans orders of magnitudes. For example, genes in unicellular eukaryotes (such as the yeast *S. cerevisiae*) are present in one to two copies per cell while mRNA and protein molecules can, on average, range between less than one to multiple thousands. Furthermore, typical biological timescales range from seconds/minutes for protein-protein interactions to minutes/hours for gene expressions. As a consequence, simulating realistic biological dynamics can rapidly become prohibitive, and efficient computational algorithms need to be devised to exploit the full predictive power of computational stochastic modeling.

A number of approaches, such as time-leaping methods and system-partitioning methods, have been developed to reduce the bottlenecks inherent in the SSA simulations of stochastic biological dynamics. All of these methods sacrifice some of the accuracy of the SSA for faster approximate Monte Carlo schemes.

Time-leaping methods, such as τ-leaping, originally developed by Gillespie,[16] allows for numerous reactions to occur between each successive simulation times. These methods begin with picking a leaping-time τ over which the propensity functions are assumed to remain constant. In this case, the number of times a given reaction occurs in τ can be approximated by a *Poisson* random variable. The work by Gillespie[16] gives indications as to the maximum value of τ that satisfies the constant-propensity assumption. A simple method developed by Rathinam,[17] which accounts for the changing propensities of the system, results in an implicit τ-leaping algorithm. More recently, Rathinam and El-Samad[18] proposed a τ-leaping algorithm for addressing small number and stiff stochastic chemical systems.

Another set of approximation techniques relies on multiscale methods in which the reacting system is typically partitioned into slow reactions and fast reactions. Notably, the work of Haseltine and Rawlings[19] separates the joint probability of the CME into a conditional probability function of the slow/small populations reactions multiplied by a marginal probability density of the fast/large populations reactions. Based on the timescale of the slow reactions, the authors derive a CME for the fast reactions, which is then approximated by Langevin equations.[20] These trajectories then serve as an input into an SSA algorithm for the slow reactions. The work of Rao and Arkin[21] and Cao *et al.*[22] develops similar algorithms, focusing on very stiff systems where the fast reactions typically reach an equilibrium before a slow reaction occurs. Similar to the work of Haseltine and Rawlings,[19] they derive a CME involving only the fast reactions but set the time derivative of this CME to zero, allowing for stationary distribution calculations to be made. The propensities of the slow reactions, calculated from the average values of the stationary distribution for the fast reactions, are then used in an SSA for the slow reactions. More recently Lötstedt and Hellander[23] proposed a hybrid method in which deterministic equations are derived for fast reactions with small variances and the SSA is applied to the slow reactions. The two partitions are coupled via Monte Carlo and quasi-Monte-Carlo summations. To maintain accuracy, the appearance of fast reactions with large variances causes the algorithm to revert to a nested-loop SSA algorithm.[24] The method developed by Weinan *et al.*[24] proposes a nested-SSA algorithm in which an inner SSA can be applied to the fast reactions by sampling a certain number of fast reaction trajectories. The approximate statistical properties of these trajectories are then used as an input to drive the SSA algorithm for the outer loop, i.e., the slow reactions. The nested-SSA is not exact as is the case for the SSA, but the authors provide error and convergence estimates for their algorithm.

In this paper, we devise a general multiscale approach to derive an algorithm that efficiently and accurately calculates stochastic realizations of the CME. We take an approach similar to the work of Haseltine and Rawlings,[19] Rao and Arkin,[21] and Cao *et al.*[22] by partitioning our reactions into two groups, which we call restricted (low propensity) and unrestricted (high propensity). Based on this partition, we derive an exact modified CME for the unrestricted reactions, which explicitly includes time-dependent effects corresponding to the probability of occurrence of the restricted reactions. Although this modified CME decomposition is as analytically difficult as the original CME, it lends itself to natural approximations. We exploit this property to devise an algorithm that considers various approximations at different levels of accuracy. Using our results, we examine some specific test systems, all of which are biologically relevant, clearly demonstrating the accuracy and efficiency of our proposed methods.

II. MARKOV PROCESS DESCRIPTION OF CHEMICAL REACTIONS AND THE CME

In this section we describe the discrete state, continuous time Markov process model for well-stirred chemical reaction systems, as well as an exact simulation algorithm for this model known as the SSA.[15] Throughout the paper, $\mathbf{Z}_+$ denotes the set of nonnegative integers and $\mathbf{R}_+$ the set of nonnegative real numbers.

The formulation we consider consists of a system of well-stirred chemical reaction with N molecular species. We use the state $X(t) \in \mathbf{Z}_+^N$ to denote the vector whose elements $X_i(t)$ are the number of molecules of the ith species at time t. If there are M elementary chemical reactions that can occur among these N species, then we associate with each reaction r_j $(j=1,\ldots,M)$ a non-negative *propensity function* $a_j: \mathbf{Z}_+^N$

$\rightarrow \mathbf{R}_+$ defined such that $a_j(X(t))\tau + o(\tau)$ is the probability that reaction r_j will happen in the next small time interval $(t, t+\tau]$, as $\tau \rightarrow 0$. The polynomial form of the propensities $a_j(x)$ may be derived from fundamental principles under certain assumptions.[15] In this paper, the propensities will be at most quadratic, that is, $a_j(x) \in \{c_0, c_1 x_i, c_2 x_i(x_i-1)/2!, c_3 x_i x_k\}$, where c_0, c_1, c_2, c_3 are rate constants. The occurrence of a reaction r_j leads to a change in $\nu_j \in \mathbf{Z}^N$ for the state X. ν_j is therefore a stoichiometric vector that reflects the integer change in reactant species due to a reaction r_j.

Based on these premises, it can be shown that the probability density function (PDF) for the waiting time τ for the next reaction is given by $a_0(x)e^{-a_0(x)\tau}$, where x is the state and $a_0(x) = \Sigma_{j=1}^{M} a_j(x)$. Also, the probability that the next reaction is r_j, is given by $a_j(x)/a_0(x)$, and is independent of τ. Knowing these two probability densities for the next reaction time and type, we can simulate the system one reaction event at a time. This method is known as the SSA and belongs to a wider class of numerical techniques known as kinetic Monte Carlo algorithms.

Given the initial conditions n_0, t_0, the evolution of all possible SSA trajectories can be described by the joint PDF $P(x,t|n_0,t_0)$, whose evolution is given by

$$\frac{\partial P(x,t|n_0,t_0)}{\partial t} = \sum_{j=1}^{M} [a_j(x-\nu_j)P(x-\nu_j,t|n_0,t_0) - a_j(x)P(x,t|n_0,t_0)]. \quad (1)$$

Equation (1) is the so-called CME for a single state initial condition, that is, $P(x=n_0, t=t_0|n_0,t_0)=1$. The CME is also generally derived using the Markov property by writing the Chapman–Kolmogorov equation, an identity that must be obeyed by the transition probability of any Markov process.[12,13] Using stationarity and taking the limit for infinitesimally vanishing time intervals, one obtains the CME as the differential form of the Chapman–Kolmogorov equation.

III. PARTITIONING A BIOCHEMICALLY REACTING SYSTEM INTO RESTRICTED AND UNRESTRICTED REACTIONS

Much of the computational inefficiency of the SSA derives from the interplay of propensities spanning a wide range of magnitudes. This can be the result of different biochemical rate constants and/or number of molecules. Biological networks always operate in this scenario. For example, transcription factors present at low molecular counts interact routinely with other abundant proteins. Furthermore, reactions involving interactions (association and dissociation) between biological molecules are typically very fast compared to gene expression. Irrespective, differences in propensity values will generally lead to disproportionate simulation time being devoted to the reactions with large propensities. This can result in severe degradation in computational efficiency. In this work, we present a method to circumvent this difficulty by leaping from one low propensity reaction to the next while estimating the change in the state due to the high propensity reactions.

Our starting point is a partitioning of the system of M biochemical reactions into two sets: K *unrestricted* reactions (typically high propensity reactions) $U=\{U_1,\ldots,U_K\}$ with propensity set $\{a_{u_1},\ldots,a_{u_K}\}$ and $M-K$ *restricted* reactions (typically low propensity reactions) $R=\{R_1,\ldots,R_{M-K}\}$ with corresponding propensity set $\{a_{r_1},\ldots,a_{r_{M-K}}\}$. Here, we will assume such a decomposition and defer the more detailed discussion of reaction partitioning to Sec. VII. When applying the SSA, one would expect that in any sequence of reactions, the relative frequency of occurrence of the restricted (R) and unrestricted (U) reactions will generally depend on the relative magnitude of the sum of each set's propensity functions. To first order, the ratio of the average time between unrestricted reactions $\tau_u = 1/\Sigma_i a_{u_i}$ relative to the average time between restricted reactions $\tau_r = 1/\Sigma_i a_{r_i}$ approximates the relative average occurrence of each reaction type. For example, a system with $\tau_r/\tau_u = 1$ might yield a reaction sequence of $\ldots URUURRURRUURURR\ldots$, while one with $\tau_r/\tau_u = 5$ might yield $\ldots RUUUUURUUUUUURUUUUR\ldots$ and so on. The larger the value of τ_r/τ_u is, the larger the sequence of unrestricted reactions would be before the occurrence of a restricted reaction. It is precisely in these situations that the SSA can become computationally cumbersome since it spends most of its time calculating the occurrence of the unrestricted reactions. In many cases of biological interest, it is essential to capture the slower-scale behavior of the restricted reactions while still properly accounting for the faster-scale unrestricted reaction statistics and their effect on the dynamics of the restricted reactions. For example, reactions involving transcription factor binding and unbinding to gene promoters occur fast and frequently. While it might not be important to account for every single one of these events, it is crucial to capture accurately the binding event that leads to productive gene expression, a rare and slow reaction.

Starting from a system that is in state n_s at time t_s, our general approach will be based on characterizing all possible SSA with restricted reaction (SSA-RR) trajectories. A SSA-RR trajectory is obtained by running the SSA starting from the state n_s at time t_s until the first occurrence of a restricted reaction. This results in a time-dependent trajectory due to a sequence of unrestricted reactions ending with a single restricted reaction. Analogous to the manner in which the CME describes the time evolution of all possible trajectories of the SSA in the form of a PDF, we will derive a similar equation for all possible SSA-RR trajectories. We will then use our derivations to form an algorithm that accurately samples a new state, n_{s+1} at time t_{s+1} given the current sampled state n_s at time t_s while skipping over the details of the many unrestricted reactions that might occur between times t_s and t_{s+1}.

IV. A MODIFIED CME FORMULATION

Given the partition of a biochemically reacting system into restricted and unrestricted, we can define the function $W(x,t|n_s,t_s)$ as the joint probability that the state assumes the value $X(t)=x$ at time t starting from the initial state $X(t_s)=n_s$ at time t_s *and* that no restricted reaction has occurred. The time evolution equation for this function can be

derived by using the laws of probability. Specifically,

$$W(x,t+dt|n_s,t_s) = W(x,t|n_s,t_s)\left[1-\sum_{j=1}^{K} a_{u_j}(x)dt-\sum_{\ell=1}^{M-K} a_{r_\ell}(x)dt\right] + \sum_{j=1}^{K} W(x-\nu_{u_j},t|n_s,t_s)a_{u_j}(x-\nu_{u_j})dt. \quad (2)$$

The first term on the right side of the equation describes the probability that the trajectory remains in the state x over the interval dt, i.e., no reaction occurs. The second term describes the probability of the trajectories entering the state x from the states $x-\nu_{u_j}$ $(j=1,\ldots,K)$ through unrestricted reactions. Note that there is no term describing the probability of entering x from the states $x-\nu_{r_\ell}$ $(\ell=1,\ldots,M-K)$ through restricted reactions. It is absent in Eq. (2) since the definition of the SSA-RR trajectory requires that the trajectory terminates when a restricted reaction occurs. Rearranging the equation, then dividing by dt, and taking the limit as $dt\to 0$ yields

$$\frac{\partial W(x,t|n_s,t_s)}{\partial t} = \sum_{j=1}^{K}\left[a_{u_j}(x-\nu_{u_j})W(x-\nu_{u_j},t|n_s,t_s)-a_{u_j}(x)W(x,t|n_s,t_s)\right] - \sum_{\ell=1}^{M-K} a_{r_\ell}(x)W(x,t|n_s,t_s). \quad (3)$$

We will refer to Eq. (3) as the CME with restricted reactions (CME-RR). Since $W(x,t|n_s,t_s)$ is a joint PDF, it can be written as the product of the conditional probability $P_u(x,t|n_s,t_s)$ of being in state $X(t)=x$ at time t given that no restricted reaction has occurred and the probability $P_{\mathrm{nr},r}(t)$ that no restricted reaction has occurred by time t. Furthermore, $P_{\mathrm{nr},r}(t)\leq 1$ is a monotonically decreasing function given by

$$P_{\mathrm{nr},r}(t)=\sum_{x'} W(x',t|n_s,t_s). \quad (4)$$

Therefore,

$$W(x,t|n_s,t_s)=\sum_{x'} W(x',t|n_s,t_s)P_u(x,t|n_s,t_s) = P_{\mathrm{nr},r}(t)P_u(x,t|n_s,t_s). \quad (5)$$

Replacing identity (5) into Eq. (3) generates the two coupled equations (for details see Appendix A),

$$P_{\mathrm{nr},r}(t)=\exp\left(-\int_{t_s}^{t}\sum_{x}\sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t'|n_s,t_s)dt'\right) \quad (6)$$

and

$$\frac{\partial P_u(x,t|n_s,t_s)}{\partial t} = \sum_{j=1}^{K}\left[a_{u_j}(x-\nu_{u_j})P_u(x-\nu_{u_j},t|n_s,t_s)-a_{u_j}(x)P_u(x,t|n_s,t_s)\right] - \sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t|n_s,t_s) + \left[\sum_{x'}\sum_{\ell=1}^{M-K} a_{r_\ell}(x')P_u(x',t|n_s,t_s)\right]P_u(x,t|n_s,t_s). \quad (7)$$

Equations (6) and (7) bear some resemblance to, but are more general than, those derived in other works.[19–22] In fact, the exact equations derived here are generalizations of these previous works. Specifically, the last two terms in Eq. (7) are time-dependent effects due to the state-dependent restricted reaction probability flux that alters the shape of the distribution function. In the limit of very large timescale separations between the restricted and unrestricted reactions, the last two terms of Eq. (7) become negligible. In this case, Eq. (7) reduces to the equations used in the derivation of the slow-scale SSA (ssSSA) algorithm.[22] Therefore, these equations recapitulate the formalism of the ssSSA as a limiting case when the propensity functions in the system are clearly separated in magnitude.

V. A SIMULATION ALGORITHM

Equations (6) and (7) form the basis for a general simulation procedure. Specifically, for a defined partition of a system into restricted and unrestricted reactions and given that $X(t_s)=n_s$ at time t_s, Eqs. (6) and (7) are solved for until time $t_s+\tau_s$, where τ_s is typically on the order of the expected time for a restricted reaction. $P_{nr,r}(t_s+\tau_s)$ is then used to determine whether or not a restricted reaction has occurred. If such a restricted reaction has not occurred, the state of the system at time $t_s+\tau_s$ is sampled from $P_u(x,t_s+\tau_s|n_s,t_s)$. However, if a restricted reaction has occurred, the restricted reaction type μ, the time t' at which it occurred, and the state from which it occurred need to be sampled. The time and type of restricted reaction can be sampled from $P(t',\mu|t'<t_s+\tau_s)$, the joint PDF that a restricted reaction occurred at time t' and that it is a μ reaction given that $t'<t_s+\tau_s$. It can be proven that $P(t',\mu|t'<t_s+\tau_s)$ is given by (see Appendix C)

$$P(t',\mu|t'<t_s+\tau_s)=\sum_{x} a_{r_\mu}(x)P_u(x,t'|n_s,t_s)\frac{P_{\mathrm{nr},r}(t')}{1-P_{\mathrm{nr},r}(t_s+\tau_s)}. \quad (8)$$

This is simply the expected value of the propensity function of the μth restricted reaction at time t', multiplied by the probability that a restricted reaction has not occurred by time t', given that $t'<t_s+\tau_s$. The expression in Eq. (8) has the same functional form and rationale as the expression for the joint probability used to sample the time and type of reaction derived for the full SSA,[15] but with two important differences. The first difference reflects the fact that the restricted

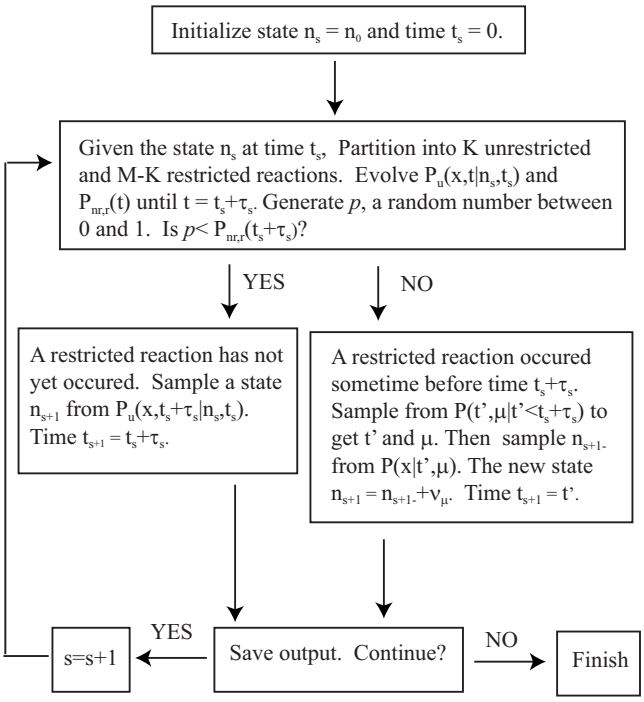

FIG. 1. Simplified flow diagram of the RRA.

reaction μ can occur from any state x which the system can assume at time t', necessitating summation over x. The second difference is the normalization factor $1-P_{\mathrm{nr},r}(t_s+\tau_s)$, which reflects the condition that the time for the restricted reaction is bounded by τ_s and in this case, is not infinite.

To sample the state x from which the restricted reaction occurred, we can use the conditional PDF that the restricted reaction occurred from state x, given it occurred at time t' and is a μ reaction,

$$P(x|t',\mu) = \frac{a_{r_\mu}(x)P_u(x,t'|n_s,t_s)}{\Sigma_{x'}a_{r_\mu}(x')P_u(x',t'|n_s,t_s)}. \tag{9}$$

We will call this procedure the restricted reaction algorithm (RRA). A block diagram illustrating the general workflow of the algorithm is depicted in Fig. 1. The algorithm, based on propagating $P_{\mathrm{nr},r}$ and P_u, then sampling from (8) and (9), will yield identical statistics to the SSA since they share the same physical underpinnings. However, solving for Eqs. (6) and (7) exactly is generally as computationally expensive as solving the CME. Nonetheless, the same procedure can be implemented if an efficient proxy of P_u is used instead. Below, we propose to use the moments of P_u to that effect and present an approximation of these moments, which can be used with a defined error.

VI. MOMENT EXPRESSIONS FOR $P_{\mathrm{nr},r}(t)$ AND $P_u(x,t|n_s,t_s)$

Let $z(t)$ be the time-dependent mean of $P_u(x,t|n_s,t_s)$. For the derivations presented below, it is convenient to express the propensities as a Taylor series around this mean of the form

$$a_k(x) = a_k(z(t)) + \sum_{i=1}^{N} \left.\frac{\partial a_k(x)}{\partial x_i}\right|_{x=z(t)} [x_i - z_i(t)] + \sum_{i,i'}^{N} \left.\frac{\partial^2 a_k(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t)} \frac{[x_i - z_i(t)][x_{i'} - z_{i'}(t)]}{2}. \tag{10}$$

In general, propensity functions for elementary reactions are at most quadratic (e.g., binding reactions). In this case, Eq. (10) is exact and can be used to express $P_{\mathrm{nr},r}(t)$ in terms of the central moments of $P_u(x,t|n_s,t_s)$. Replacing Eq. (10) into Eq. (6) yields

$$P_{\mathrm{nr},r}(t) = \exp\left(-\int_{t_s}^{t} \sum_{x} \sum_{\ell=1}^{M-K} \left[a_{r_\ell}(z(t')) + \sum_{i=1}^{N} \left.\frac{\partial a_{r_\ell}(x)}{\partial x_i}\right|_{x=z(t')} \times [x_i - z_i(t')] + \sum_{i,i'}^{N} \left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t')} \times \frac{[x_i - z_i(t')][x_{i'} - z_{i'}(t')]}{2} \right] P_u(x,t'|n_s,t_s)dt' \right).$$

Therefore,

$$P_{\mathrm{nr},r}(t) = \exp\left(-\int_{t_s}^{t} \sum_{\ell=1}^{M-K} \left[a_{r_\ell}(z(t')) + \sum_{i,i'}^{N} \left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t')} \frac{C_{ii'}(t')}{2} \right] dt' \right). \tag{11}$$

Equation (11) states that for linear and quadratic propensities, $P_{\mathrm{nr},r}(t)$ is a function of the central moments of $P_u(x,t|n_s,t_s)$ up through the covariances. Furthermore, this result is exact, and up to this point, has not involved any approximations. If the propensities are at most linear, the equations for means and covariances of $P_u(x,t|n_s,t_s)$ are self-contained and can be easily solved to generate $P_{\mathrm{nr},r}(t)$. However, for quadratic propensity functions, the first two time-dependent central moments for $P_u(x,t|n_s,t_s)$ are given by[25]

$$\frac{\partial z_k(t)}{\partial t} = \sum_{j=1}^{K} \nu_{jk} \left(a_{u_j}(x) + \sum_{i,i'}^{N} \left.\frac{\partial^2 a_{u_j}(z(t))}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t)} \frac{C_{ii'(t)}}{2} \right) - \sum_{\ell=1}^{M-K} \sum_{i=1}^{N} \left.\frac{\partial a_{r_\ell}(x)}{\partial x_i}\right|_{x=z(t)} C_{ki} - \sum_{x} \sum_{\ell=1}^{M-K} \sum_{i,i'}^{N} \times \left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t)} [x_k - z_k(t)] \times \frac{[x_i - z_i(t)][x_{i'} - z_{i'}(t)]}{2} P_u(x,t|n_s,t_s), \tag{12}$$

$$\frac{\partial C_{kk'}(t)}{\partial t} = \sum_{j=1}^{K} \left(\nu_{jk} \left[\sum_{i} \left. \frac{\partial a_{u_j}(x)}{\partial x_i} \right|_{x=z(t)} C_{ik'} + \sum_{x} \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{u_j}(x)}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} [x_{k'} - z_{k'}(t)] \frac{[x_i - z_i(t)][x_{i'} - z_{i'}(t)]}{2} P_u(x,t|n_s,t_s) \right] \right.$$
$$\left. + \nu_{jk'} \left[\sum_{i'} \left. \frac{\partial a_{u_j}(x)}{\partial x_{i'}} \right|_{x=z(t)} C_{ki'} + \sum_{x} \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{u_j}(x)}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} [x_k - z_k(t)] \frac{[x_i - z_i(t)][x_{i'} - z_{i'}(t)]}{2} \times P_u(x,t|n_s,t_s) \right] \right)$$
$$+ \sum_{j=1}^{K} \nu_{jk} \nu_{jk'} \left(a_{u_j}(x) + \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{u_j}(x)}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} \frac{C_{ii'}}{2} \right) - \sum_{x} \sum_{\ell=1}^{M-K} \left(\sum_{i=1}^{N} \left. \frac{\partial a_{r_\ell}(x)}{\partial x_i} \right|_{x=z(t)} [x_k - z_k(t)][x_{k'} - z_{k'}(t)][x_i - z_i(t)] \right.$$
$$\left. + \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{r_\ell}(x)}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} [x_k - z_k(t)][x_{k'} - z_{k'}(t)] \frac{[x_i - z_i(t)][x_{i'} - z_{i'}(t)]}{2} \right) P_u(x,t|n_s,t_s)$$
$$+ \sum_{r_\ell=1}^{M-K} \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{r_\ell}(x)}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} \frac{C_{ii'}(t)}{2} C_{kk'}(t). \tag{13}$$

In these equations, the third and fourth order central moments are presented in summation form (see Appendix B for derivation). One can see that a difficulty arises when the covariances and means of $P_u(x,t|n_s,t_s)$ depend on higher order moments in the presence of quadratic propensities, i.e., nonzero terms involving the second derivatives of the unrestricted propensities. It is common practice in the literature to truncate the system after the Nth moment (typically $N=2$) either by setting the higher than N moments to zero or by approximating their functional relationship to the first N moments. For example, a Gaussian approximation assumes that all odd central moments are zero and the even moments are functions of the second moment. In general, other moment truncation approximations such as those proposed by Singh and Hespanha[26] could be used.

For time-dependent moments, the duration that a particular moment truncation method remains accurate is uncertain. Here, we can take advantage of the fact that by construction $P_u(x=n_s,t=t_s|n_s,t_s)=1$ since the system is in a single deterministic state at the time t_s when it is sampled. At this point, the first moment is nonzero, while all other central moments are zero. Evidently, as $P_u(x,t|n_s,t_s)$ evolves over time, dependence on higher order central moments increases. Therefore, in principle, one strategy is to ignore higher order moments with a quantifiable error until some time τ_s at which their contribution becomes important. In that sense, τ_s could be the maximum time-step for which moment truncation error is below a satisfactory tolerance. To quantify the error, we can exploit the functional form of Eq. (10) where the difference between the linear and quadratic terms reflects the relative contributions to the first order and second order unrestricted propensity derivative terms in Eqs. (12) and (13). When the distribution is in a locally linear unrestricted reaction propensity landscape, the summation of the terms involving the second derivative of the unrestricted propensities will be negligible. These terms include covariances, third and higher central moments. We therefore truncate our equations at the covariances with the assumption that at time τ_s, the error is less or equal than some error bound ϵ_s. The benefit of this approach is that it ensures controllable accuracy in the moments of P_u without resorting to more complicated truncation methods. Indeed, the error made at every step is bounded by ϵ_s. This is reminiscent of the choice of error tolerance in ordinary differential equation (ODE) solvers. It is important to note here, however, that the choice of ϵ_s presents an intrinsic compromise between speed and accuracy; if ϵ_s is required to be very small, τ_s will be restricted to small values, and one might lose the benefits of this method as compared to the SSA.

A natural definition of the moment error e_m is

$$e_m = \max \left[\frac{\Sigma_{i,i'}^{N} \left. \frac{\partial^2 a_{u_1}(z(t))}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} \frac{C_{ii'(t)}}{2}}{a_{u_1}(x)_{x=z(t)}}, \ldots, \right.$$
$$\left. \frac{\Sigma_{i,i'}^{N} \left. \frac{\partial^2 a_{u_K}(z(t))}{\partial x_i \partial x_{i'}} \right|_{x=z(t)} \frac{C_{ii'(t)}}{2}}{a_{u_K}(x)_{x=z(t)}} \right], \tag{14}$$

which is the maximum of the ratios of the second order propensity, covariance term to the mean propensity for each of the unrestricted reactions. The "max" function takes into account the worst error, which is a conservative measure. In our evaluation of e_m, we are only focused on the local nonlinear effects of the unrestricted propensities since, due to their high propensities, their contribution to the error will be the largest. Exploring other representations of e_m is a topic of future research.

With this approximation of the moments of P_u, the general simulation procedure outlined in the previous section can now be implemented. For τ_s that satisfies the prescribed tolerance ϵ_s, an expression for $P(t',\mu|t'<t_s+\tau_s)$, which is easily obtained by replacing Eq. (10) into Eq. (8), is given by

$$P(t',\mu|t' < t_s + \tau_s) = \left[a_{r_\mu}(z(t')) + \sum_{i,i'}^{N} \left. \frac{\partial^2 a_{r_\mu}(x)}{\partial x_i \, \partial x_{i'}} \right|_{x=z(t')} \frac{C_{ii'}(t')}{2} \right] \times \frac{P_{\mathrm{nr},r}(t')}{1 - P_{\mathrm{nr},r}(t_s + \tau_s)}. \quad (15)$$

Likewise, a similar procedure can be used to derive an approximate expression for $P(x|t',\mu)$ as

$$P(x|t',\mu) = \frac{a_{r_\mu}(x)\bar{P}_u(x,t'|n_s,t_s)}{a_{r_\mu}(z(t')) + \Sigma_{i,i'}^{N} \left. \frac{\partial^2 a_{r_\mu}(x)}{\partial x_i \, \partial x_{i'}} \right|_{x=z(t')} \frac{C_{ii'}(t')}{2}}. \quad (16)$$

The numerator in Eq. (16) is the same as that in Eq. (9) with the difference that we have replaced P_u with the approximate version $\bar{P}_u$ fit to the calculated moments. The details of sampling from these distributions are given in Appendix D.

VII. A NOTE ABOUT PARTITIONING

The partitioning strategy for a system of reactions into restricted and unrestricted will necessarily affect the efficiency of the algorithm. Optimizing this partitioning beyond rules of thumb and intuition about a given system is a difficult but interesting problem, which we do not tackle in this work. Instead, here we discuss qualitatively a few typical partitioning examples, which will be further illustrated in the numerical examples.

A. Large separation in propensity values

If the magnitudes of propensities are clearly separated by orders of magnitude, then an obvious partition would assign the low propensity values to be restricted reactions. In this case, the RRA effectively converges to the ssSSA algorithm. Specifically, the moments for P_u will reach an equilibrium well before the time at which a restricted reaction is expected to occur. An efficient ODE solver can still be used to determine both the dynamics and the convergence to steady-state of the moments in a few implicit iterations. The ssSSA, in which the stationary moments are iteratively solved for, can also be efficiently applied.

B. Similar propensity values

If reaction propensities have similar magnitudes, one can choose to revert to SSA (all restricted reactions) or jump state through the evolution of the moments (all unrestricted reactions). The decision should depend on the efficiency of the SSA; if the concentrations of species and/or the propensities are low, the SSA will be optimal. On the other hand, if the concentrations and/or propensities are high, jumping to a new state by evolving the moments may be most efficient. Switching between these two regimes dynamically might also be required, for example, in a bistable system where high and low equilibriums are visited by stochastic trajectories.

C. Spectrum of propensity values

Many biological systems assume a widespread spectrum of propensities, making separation between "fast" and "slow" timescales difficult. Consider, for example, a system of ten reactions and eight species. If there are four reactions with propensities near one, then $t_{\mathrm{slow}} \simeq 1/(4\times 1) = 1/4$. If there are two reactions with propensities near 50, $t_{\mathrm{middle}} \simeq 1/(2\times 50) = 1/100$. Finally, if the remaining four reactions each have propensities near 1000, then $t_{\mathrm{fast}} \simeq 1/(4\times 1000) = 1/4000$. The timescales ratio are $t_{\mathrm{slow}}/t_{\mathrm{middle}} = 25$ and $t_{\mathrm{middle}}/t_{\mathrm{fast}} = 40$. If the four lowest propensity reactions are considered to be restricted, then the ODE solver needs to integrate the moments for a time, on average, of t_{slow} seconds where the moment system has a stiffness of approximately $t_{\mathrm{middle}}/t_{\mathrm{fast}} = 40$. In this case it would be most efficient to apply an implicit ODE solver which can take much larger timesteps than explicit solvers, especially for stiff systems. If, instead, both low and middle propensity reactions are considered to be restricted, then the update time for a restricted reaction is t_{middle} seconds and the moment system stiffness is approximately 1. Here, it would be most efficient to apply an explicit ODE solver. However, there will be 25 restricted reactions before the reacting system has evolved through a time of t_{slow} seconds. For the two strategies described, the most computationally efficient strategy would be to partition both the middle and fast reactions as unrestricted and apply an implicit ODE solver. In general, one has to be aware of the fact that the largest computational expense of the RRA stems from the ODE solution of the moment equations and, therefore, the choice of implicit versus explicit integration strategies will either enhance or inhibit efficiency.

For the numerical results in this paper, we will only use the implicit ODE solver CVODE[27,28] from the SUNDIALS (Ref. 29) package, which can be interfaced with the Portable Extensible Toolkit for Scientific Computation (PETSc).[30,31] Throughout our numerical examples below, we will investigate the issues associated with partitioning in the three cases presented above and explore the tradeoffs between numerical efficiency and accuracy.

VIII. BIOLOGICAL EXAMPLES

In this section, we apply the RRA algorithm to three biological network examples, a simple linear system, a genetic oscillator, and a bistable system. The examples are chosen to scrutinize various properties of the method and allow us to gain some practical understanding of the tradeoffs between its accuracy and computational expediency. We therefore compare the performance of a number of approximate algorithms, all based on Eqs. (6) and (7) and the general strategy outline in Fig. 1, but which involve different accuracy in the approximation of P_u. These include the following.

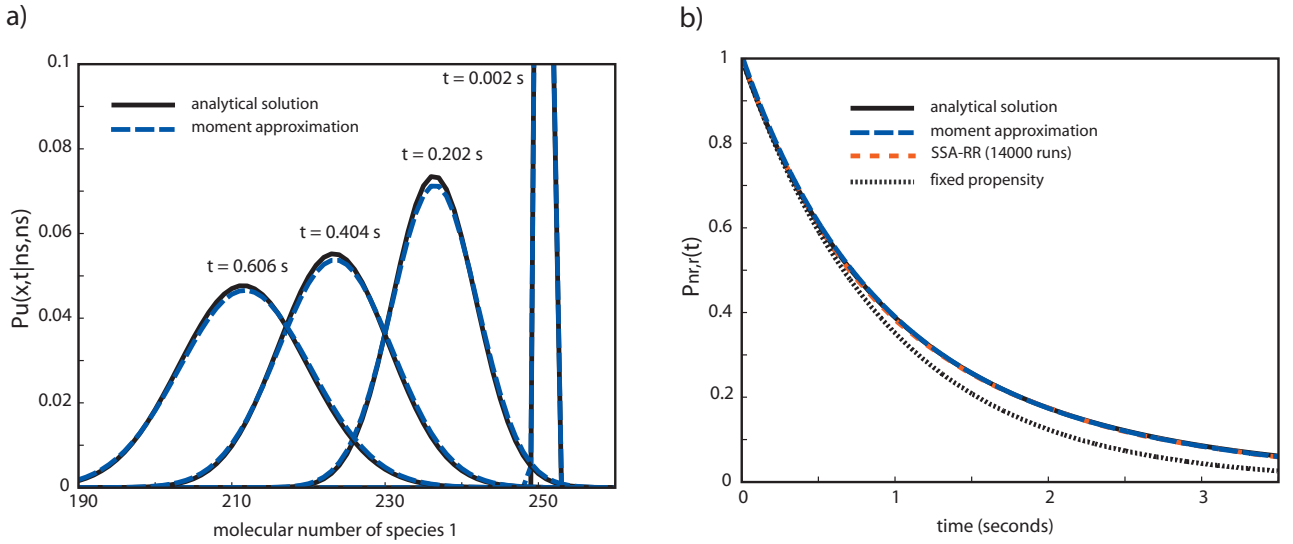

FIG. 2. $P_u(x,t|n_s,t_s)$ and $P_{\mathrm{nr},r}(t)$ for a simple reacting system. (a) Plot of $P_u(x,t|n_s,t_s)$ at different times. (b) Plot of $P_{\mathrm{nr},r}(t)$. Analytic method, moment method, and the results from the compilation of 14 000 SSA-RR runs all agree very closely. Solid black line approximates $P_{\mathrm{nr},r}(t)$ when the propensities are held constant at $a(n_s)$.

- RRA-exact is an exact simulation algorithm using the propagation of Eqs. (6) and (7).
- RRA-exact-m2 algorithm uses the propagation of Eqs. (6) and (7) while neglecting the last two terms in Eq. (7).
- RRA-cov algorithm uses the covariance representation, Eqs. (11)–(13).
- RRA-mean algorithm uses the mean representation, Eqs. (11) and (12) neglecting the covariance.

A. A simple linear reacting system

For this example we will examine the time evolution of $P_{\mathrm{nr},r}(t)$ and $P_u(x,t|n_s,t_s)$ for a simple linear reacting system. The system consists of $N=2$ species and $M=4$ reaction pathways given by

$$\varnothing \xrightarrow{k_1} X_1 \xrightarrow{\alpha_1 x_1} \varnothing\,, \tag{17a}$$

$$\varnothing \xrightarrow{k_2 x_1} X_2 \xrightarrow{\alpha_2 x_2} \varnothing\,. \tag{17b}$$

The propensities are given by $a(x)=[k_1,\alpha_1 x_1,k_2 x_1,\alpha_2 x_2]$. We define the unrestricted reaction propensities to be $[k_1,\alpha_1 x_1]$ and the restricted reaction propensities to be $[k_2 x_1,\alpha_2 x_2]$. Here, a numerical solution for Eq. (3) can directly be obtained using the finite state projection.[32] This is possible and accurate since in our case $(K=2)$, $P_{\mathrm{nr},r}(t)$, converges toward zero quickly, allowing us to represent the state space with only 400 states. This method will provide a solution used to gauge the accuracy of the RRA. The explicit formulas for $P_{\mathrm{nr},r}(t)$ and the moments are given by

$$P_{\mathrm{nr},r}(t) = \exp\left(-\int_{t_s}^{t} [k_2 z_1(t') + \alpha_2] dt'\right), \tag{18a}$$

$$\frac{\partial z_1}{\partial t} = k_1 - \alpha_1 z_1 - \alpha_2 C_{11}, \tag{18b}$$

$$\frac{\partial C_{11}}{\partial t} = -2\alpha_1 C_{11} + k_1 + \alpha_1 z_1. \tag{18c}$$

Equations (18b) and (18c) form a closed set of equations that can be solved for mean and variance $[z_1(t),C_{11}(t)]$ for species one. For the initial state $n_s=[256;5]$ and parameter values $[k_1=49,\ \alpha_1=.5,\ k_2=.0031,\ \alpha_2=.055]$, we evolve both the analytical system and the moment system [Eqs. (18a)–(18c)] over time. Snapshots of $P_u(x,t|n_s,t_s)$ for both the analytical method (solid) and the moment method (dashed) are plotted for different time points in Fig. 2(a). For this example, the Gaussian function approximation $P_u(x,t|n_s,t_s) = (1/(2\pi C_{11}(t)))^{1/2}\exp(-(x_1-z_1(t))^2/(2C_{11}(t)))$ agrees very well with the analytical solution.

$P_{\mathrm{nr},r}(t)$, plotted in Fig. 2(b), shows perfect overlap between the analytic method and the moment approximation. The combined results from 14 000 SSA-RR runs also show excellent agreement. Figure 2(b) also shows an approximation of $P_{\mathrm{nr},r}(t)$ in the case where the restricted reaction propensities are held constant at $a_{r_\ell}(n_s)$. This approximations result in a purely exponential curve, which differs from the exact solution. This clearly underscores the importance of accounting for the restricted reaction propensity changes due to the unrestricted reactions distribution, $P_u(x,t|n_s,t_s)$.

B. A genetic oscillator

In this example, we simulate a model of a system developed by Vilar *et al.*[33] to approximate the essential functionality of a circadian oscillator. The system is described by the following biochemical reactions:

TABLE I. Parameters for oscillator example.

Parameter	α_A	α'_A	α_R	α'_R	β_A	β_R	δ_{MA}	δ_{MR}
Case 1	50	500	0.01	50	50	5	10	0.5
Case 2	1000	10 000	0.2	1000	50	5	10	0.5
Case 3	10 000	100 000	2	10 000	50	5	10	0.5

Parameter	δ_A	δ_R	γ_A	γ_R	γ_C	θ_A	θ_R
Case 1	1	0.05	1	1	2	50	100
Case 2	1	0.05	1	1	2	50	100
Case 3	1	0.05	1	1	2	50	100

$$D_a + A \underset{\theta_a}{\overset{\gamma_a A}{\rightleftharpoons}} D'_a, \tag{19a}$$

$$D_r + A \underset{\theta_r}{\overset{\gamma_r A}{\rightleftharpoons}} D'_r, \tag{19b}$$

$$\varnothing \overset{\alpha'_a D'_a + \alpha_a D_a}{\longrightarrow} M_a \overset{\delta_{ma} M_a}{\longrightarrow} \varnothing, \tag{19c}$$

$$\varnothing \overset{\alpha'_r D'_r + \alpha_r D_r}{\longrightarrow} M_r \overset{\delta_{mr} M_r}{\longrightarrow} \varnothing, \tag{19d}$$

$$\varnothing \overset{\beta_a M_a}{\longrightarrow} A \overset{\delta_a A}{\longrightarrow} \varnothing, \tag{19e}$$

$$\varnothing \overset{\beta_r M_r}{\longrightarrow} R \overset{\delta_r R}{\longrightarrow} \varnothing, \tag{19f}$$

$$A + R \overset{\alpha'_c AR}{\longrightarrow} C \overset{\delta_a C}{\longrightarrow} R. \tag{19g}$$

Briefly, transcription factor A positively regulates both itself and the repressor protein R, resulting in nested positive and negative feedback loops. For the parameters used in this paper, A is produced at a faster rate than R but has a shorter half-life. As R is produced, it negatively regulates A by combining with A to form the complex C. Eventually, the population of A is reduced to near zero, thereby downregulating the transcription of A and R. As the population of R decays toward zero, as long as the reactions are stochastic, the resulting fluctuations in the concentration of A and R enable the process to repeat (oscillate). It is during the periods of near-zero numbers of A that the transcription factor binding of A and consequently the transcription of A and R are rare events. We simulate the oscillator model for three sets of parameter values, where time is measured in hours, given in Table I.

For case 1, the deterministic (mean path) behavior of the system plotted as the dashed line in Fig. 3(a) indicates that repressor protein R converges to a stable steady state. However, accounting for the stochasticity in the reactions by applying the SSA results in oscillations [Fig. 3(a), solid curve for R and inset for A]. We define the period of oscillation as a random variable τ that captures the time difference between two successive peaks. As a metric for comparison, we will use histograms of τ produced by the different algorithms. All histograms are calculated from the results of 300 simulations where each simulation is run for 6000 h.

In a first set of tests whose purpose is to demonstrate the equivalence of the SSA and RRA-exact algorithms and to compare the performance of RRA-mean and RRA-cov, we investigate static partitioning (partitioning is preassigned and remains constant throughout the simulation). We model the reactions that dictate the transcription factor-bound states of the genes, i.e., (19a) and (19b), as unrestricted reactions and consider all other reactions to be restricted. RRA-exact and RRA-exact-m2 are easily implemented since for one copy of gene promoters, there are only four states in $P_u(x,t|n_s,t_s)$ to be solved for over time. This particular restricted reaction partitioning also allows all of the algorithms to use the exact discrete sampling techniques given in Appendix D, therefore, eliminating sources of error for the RRA-cov and RRA-mean algorithms. Figure 3(b) shows PDFs of τ, the period of oscillation. As expected, the SSA and the RRA-exact overlap. The RRA-exact-m2, which neglects the last two terms of P_u in Eq. (7), results in a thinner distribution and a lower mean. The RRA-mean algorithm has a similar distribution width to the SSA but is shifted to a higher mean. In contrast, the RRA-cov algorithm generates a PDF that agrees very well with the SSA and RRA-exact. This test again demonstrates that neglecting the effect of restricted reactions on P_u can generate error. It also demonstrates that, at least for this particular static partitioning, accounting for the covariances is both necessary and sufficient to reproduce the PDF of τ.

Next, we applied dynamic partitioning of propensities to the RRA-cov for the three difference cases described in Table I. We tested two simple partitioning methods. The first partitioning method (type I) uses a propensity threshold value a_t, below which a reaction is partitioned as restricted and above which it is considered unrestricted. We set the ODE integration time for the moments of P_u to be $\tau_s = 2.0(1/\Sigma_\ell a_{r_\ell}(n_s))$ and the error threshold to $\epsilon_s = 0.15$. To gauge the performance of the approximation, we measure the maximum error produced by the nonlinear terms within the moments equations as a function of state update/time. A threshold value of $a_t = 18\ \text{h}^{-1}$ was chosen since it optimized the run times for the three test cases. For comparison, the second partitioning method (type II) is based on the value of the protein A. When $A < 10$, we partition all the reactions as restricted; otherwise all reactions are partitioned as unrestricted.

Figure 3(c) shows the results for case 1. Period distribu-

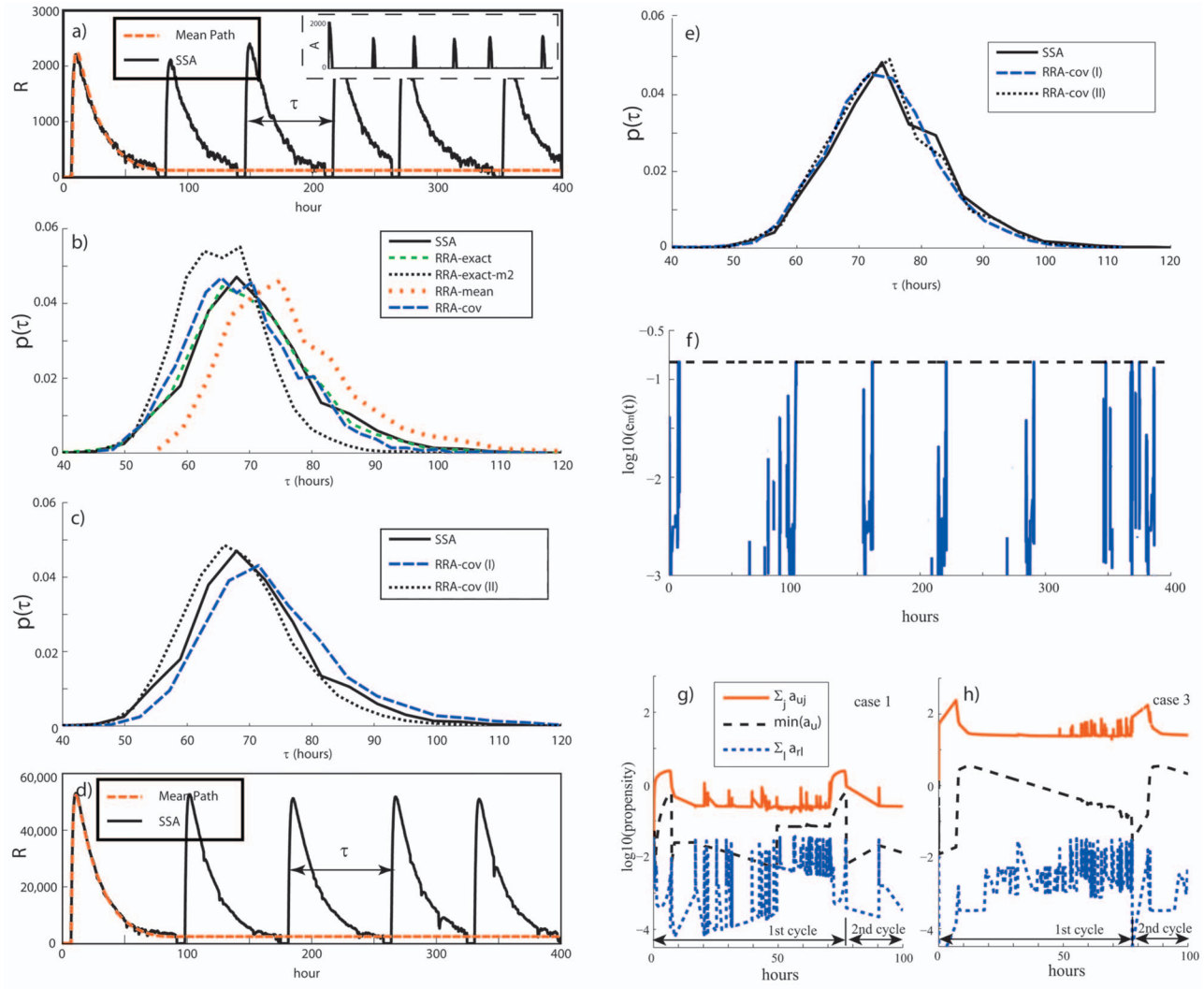

FIG. 3. Biological oscillator. Case 1: (a)–(c). (a) Plot of the repressor protein R for the mean path equations and one SSA simulation. (b) Histrograms of τ for a fixed partitioning of the *unrestricted* reactions (gene state transitions). (c) Histrograms of τ for dynamic partitioning of the reactions. Case 2: (d)–(e). (d) Plot of the repressor protein R for the mean path equations and one SSA simulation. (e) Histrograms of τ for dynamic partitioning of the reactions. (f) Plot of moment error, $e_m(t)$ (solid), and error threshold, ϵ_s=0.15 (dashed). [(g) and (h)] Plots vs time for the mean values of $\Sigma_j a_{u_j}(z(t))$, $\min(a_u(z(t))$, and $\Sigma_\ell a_{r_\ell}(z(t))$. (g) Case 1. (h) Case 3.

tions for RRA-cov (I) and RRA-cov (II) exhibit equally accurate distributions compared to the SSA. The differences in the distributions between the SSA vs. the RRA-cov algorithms are now mostly due to the sampling technique for P_u, specifically to the extraction of a value for A mRNA. In this case, when protein A concentration is at near-zero levels, the mRNA also exists in low numbers. At these low values, the assumptions made for the sampling technique will be inaccurate (see Appendix D for further details). In general, sampling from a multivariate distribution P_u is a challenging problem and a focus of much current research.

In case 2, the transcription rate for A and R was increased by a factor of 20. A plot of the repressor protein R as a function of time is shown in Fig. 3(d). In this case, protein levels are about a factor of 20 higher than in case 1. The results of the mean-path equations are also plotted in Fig. 3(d) and still converge to a unique stable equilibrium as in case 1. Period distributions for RRA-cov (I) and RRA-cov (II) are plotted in Fig. 3(e). Here, distributions generated by the RRA-cov algorithms agree exquisitely well with the SSA, indicating that the algorithms are still applicable even for this substantially "stiffer" problem.

For case 3, the transcription rate for A and R was increased from case 1 by a factor of 200. The distributions (not shown) for the RRA-cov algorithms show a similar remarkable accuracy as in case 2.

The log of the moment error $e_m(t)$, which was found to be similar for the RRA-cov algorithms in all three cases, is plotted in Fig. 3(f) for a portion of a trajectory of the RRA-cov (I). The error is greatest during the pulses of the oscillator and only on occasion reaches the error threshold ϵ_s =0.15. One notices that there are regions in between the pulses where the error is vanished. These regions are where all unrestricted reactions are at most linear propensities, i.e., all bimolecular reactions are partitioned as restricted. An interesting feature, emphasized by these results, is that the number of state updates and the run time duration for both SSA and RRA-cov (II) scaled directly with the change in

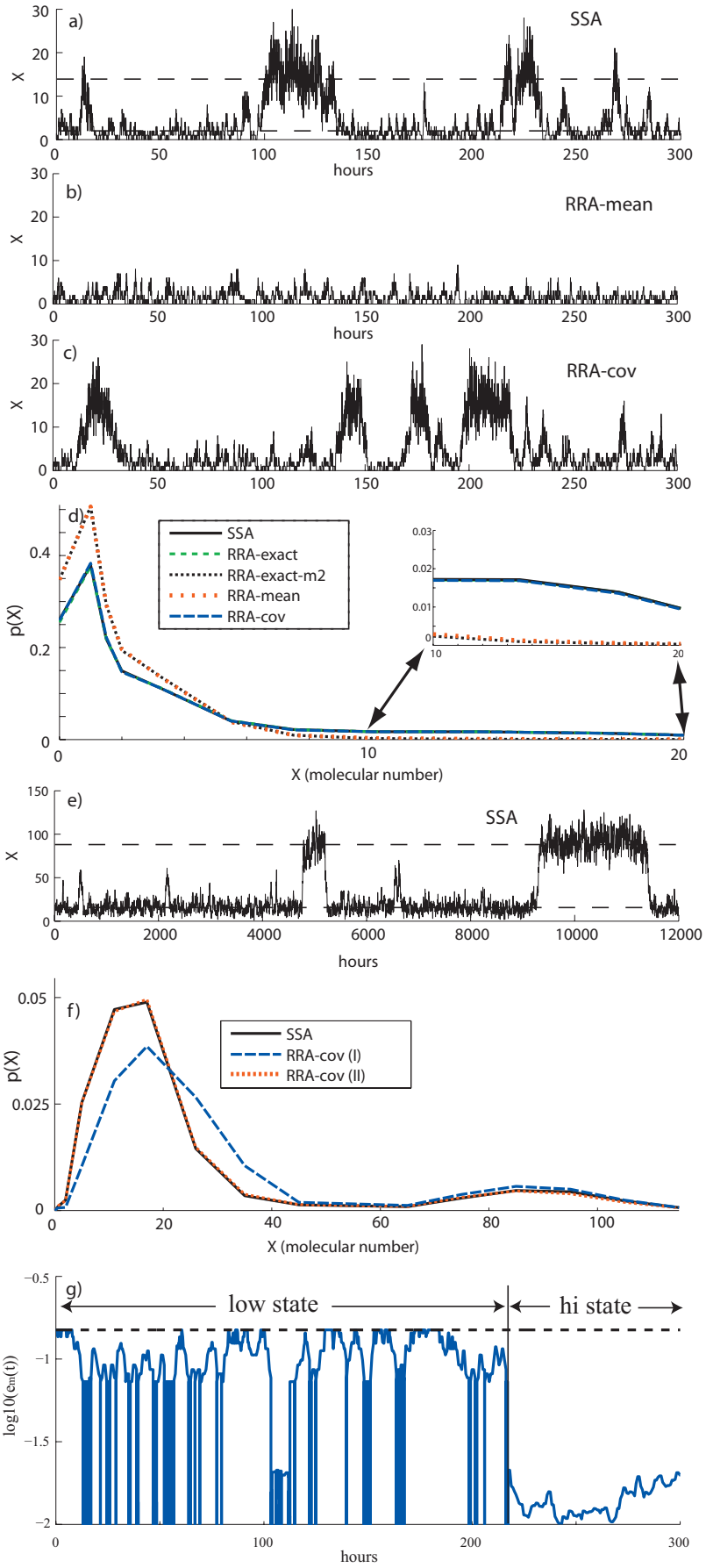

FIG. 4. Bistable pathway. Case 1: (a)–(d). (a) Stochastic trajectory of SSA (dashed lines represent the fixed points). (b) Stochastic trajectory of RRA-mean, fixed partition. (c) Stochastic trajectory of RRA-cov, fixed partition. (d) Plot of stationary distribution for case 1, fixed partition case. Case 2: (e) and (f). (e) Stochastic trajectory of SSA (dashed lines represent the fixed points). (f) Plot of stationary distribution for case 2. (g) Plot of moment error (log), $e_m(t)$ (solid), and error threshold (log), $\epsilon_s=0.15$ (dashed).

TABLE II. Run time results for the oscillator example.

Algorithm	Run time (s)	Relative speed vs SSA	Total state updates
Case 1			
SSA	12.5	⋯	9 100 000
RRA-cov (I)	112	0.116	47 000
RRA-cov (II)	11.2	1.17	5 300 000
Case 2			
SSA	240	⋯	185 000 000
RRA-cov (I)	120	2.0	39 000
RRA-cov (II)	144	1.66	108 000 000
Case 3			
SSA	2410	⋯	1 800 000 000
RRA-cov (I)	120	20	39 000
RRA-cov (II)	1340	1.78	950 000 000

transcription rates (see Table II). In contrast, the number of state updates and run times for the RRA-cov (I) algorithm remained relatively constant as parameters changed. This is not unexpected since spreading of the timescales in the system results mainly in an increase in the number of unrestricted reactions, which are handled efficiently by the implicit ODE solver. For the less-stiff system in case 1, the gain from the reduction in the number of reactions is outweighed by the cost of the implicit ODE solver. However, as the problem becomes stiffer, the computational cost of the SSA scales up, while that of RRA, particularly RRA (I), stays relatively constant. Optimizing the interface between the RRA and the ODE solver and improving the dynamic partitioning, which was not attempted in this paper, should make this contrast all the more salient.

To probe further the computational efficiency of the RRA (I), we plotted the log of $\Sigma_j a_{u_j}(z(t))$, $\min(a_u(z(t))$, and $\Sigma_j a_{r_\ell}(z(t))$ versus time as the simulation proceeded for cases 1 and 3 [see Figs. 3(g) and 3(h)]. These quantities represent the fastest timescale of the unrestricted reactions, the slowest timescale of the unrestricted reactions, and the fastest timescale of the restricted reactions, respectively. The difference in the log plots of $\min(a_u(z(t))$ (long dashed) versus $\Sigma_\ell a_{r_\ell}(z(t))$ (short dashed) represents a measure of the minimum separation of timescales between the unrestricted and the restricted partitions. More precisely, the value of the log difference represents the order of magnitude in the ratios of the quantities. For case 1, the difference is between two and three for the early part of the oscillation and is very small for the rest of the cycle. In contrast, for case 3, the separation is two and three for most of the cycle. Furthermore, the difference in the log plots of $\min(a_u(z(t))$ (long dashed) versus $\Sigma_j a_{u_j}(z(t))$ (solid) represents a measure of the maximum separation of timescales within the unrestricted partition and is an approximate gauge for the stiffness within the unrestricted partition. For case 1, the difference is one or less, while for case 3, the difference is two to four. These results are in clear agreement with our numerical studies indicating that the SSA should be very efficient for case 1 and that the RRA should more efficient in case 3 with the use of an implicit ODE solver. These results also show that the oscillator, especially in case 3, exhibits a spectrum of timescales throughout the cycle for which the RRA is ideally suited.

C. A bistable switch

Next, we use the RRA to simulate a bistable system developed by Hasty *et al.*[34] as an example to discuss noise-based switches and amplifiers for gene expression. The biochemical reactions are given by

$$2X \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} X_2, \tag{20a}$$

$$D + X_2 \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} DX_2, \tag{20b}$$

$$D + X_2 \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} DX_2^*, \tag{20c}$$

$$DX_2 + X_2 \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} DX_2X_2, \tag{20d}$$

$$0 \overset{r}{\rightarrow} nX, \tag{20e}$$

$$DX_2 + P \overset{k_t}{\rightarrow} DX_2 + P + nX, \tag{20f}$$

$$X \overset{k_d}{\rightarrow} \varnothing, \tag{20g}$$

$$X_2 \overset{k_d}{\rightarrow} X. \tag{20h}$$

For this system, the transcription/translation of a protein X is aggregated into one step. It can be shown that for a deterministic mass action model, the equation for the protein abundance x at steady-state reduces to

$$\frac{p_0 k_t n d_t R_2 f(x)}{1 + (R_2 + R_3) f(x) + R_2 R_4 f^2(x)} - k_d x + r = 0, \tag{21}$$

where $f(x) = (k_1/(k_{-1}+k_d))x(x-1)/2$, p_0 is the RNA polymerase concentration, d_t is the total number of DNA sites, $R_2 = k_2/k_{-2}$, $R_3 = k_3/k_{-3}$, $R_4 = k_4/k_{-4}$, and $n=2$. Solving Eq. (21) yields the fixed points for the system for parameter values, where time is measured in seconds, given in Table III.

For case 1, the stable fixed points are $x_{ss} = [1.45, 13.23]$. In the first set of tests for this case, we adopt a static partitioning where reactions that dictate the states of the genes (20b)–(20d) are considered to be unrestricted, while all other reactions are restricted. Figure 4(a) shows an example of a time plot of protein X for the SSA where the dashed lines represent the fixed points. Figures 4(b) and 4(c) show typical results for the RRA-mean and RRA-cov algorithms, respectively. Obviously, RRA-mean generates fluctuations of X around its low steady-state and does not exhibit the switching seen either by SSA or RRA-cov. This is further corroborated in Fig. 4(d) where the stationary distribution of the protein X is plotted. The distributions calculated from the SSA, RRA-exact, and RRA-cov all agree very closely.

TABLE III. Parameters for the bistable system.

Parameter	k_1	k_{-1}	k_2	k_{-2}	k_3	k_{-3}	k_4	k_{-4}
Case 1	0.0301	0.5	0.0199	0.5	0.0199	0.5	0.994	0.5
Case 2	0.0043	0.5	0.0028	0.5	0.0028	0.5	0.142	0.5

Parameter	$p_0 k_t n$	d_t	k_d	r
Case 1	0.0954	1	0.0007	0.000 954
Case 2	0.0954	1	0.0001	0.000 954

Therefore, in order to capture the noise induced switching in this example, it is necessary to include the higher order moments. Importantly, this strongly points to the fact that such switching behavior is inherently dependent on the noise in the binding of X_2 to DNA since switching is eliminated when only the mean behavior of the process is accounted for. As in the oscillator example, the results for the RRA-exact-m2 are again very poor, and the stationary distribution does not display bistable behavior [distribution is near zero for $X>10$ (Fig. 4(d) inset)].

Next, we applied two simple partitioning methods for the RRA-cov to the two cases given in Table III. For the first method (type I), we partition all reactions as unrestricted and set the maximum integration time to be $\tau_s=10$ h and an error threshold of $\epsilon_s=0.15$. The second method (type II) is based on the value of the protein A; when $A<10$, we partition the reactions as all restricted, while for $A>=10$ all reactions are partitioned as unrestricted. The maximum integration time is still $\tau_s=10$ h.

For case 1, the SSA is efficient and there is no gain in computational expediency from using RRA-cov (I) or RRA-cov (II) (see Table IV).

In case 2, we scale k_1, k_2, k_3, k_4, and k_d by α where $\alpha=1/7$ (Table III). The stable fixed points are $x_{ss}=[15.4, 90.4]$. Figure 4(e) shows a time plot of the SSA where dashed lines represent the fixed points. Figure 4(f) shows the stationary distribution for the SSA, RRA-cov (I), and RRA-cov (II). All methods capture the noise-based switching. Distributions generated by SSA and RRA-cov (II) match perfectly, while that generated by RRA-cov (I) shows a perfect match to the high-state distribution but an offset in the distribution of the low state. Error in RRA-cov (I) is likely due to the moment error, as well as from the sampling of the state from P_u while the system at low concentrations.

TABLE IV. Run time results for the bistable system.

Algorithm	Run time (s)	Relative speed vs SSA	Total stated updates
		Case 1	
SSA	66	⋯	72 000 000
RRA-cov (I)	180	0.37	120 000
RRA-cov (II)	90	0.73	6 000 000
		Case 2	
SSA	460	⋯	450 000 000
RRA-cov (I)	30	15.3	30 000
RRA-cov (II)	50	9.20	30 000 000

The log of the moment error $e_m(t)$ generated by RRA-cov (I) for case 2 is plotted in Fig. 4(f). It can easily be seen that the error in the low state reaches or is near $\epsilon_s=0.15$ much of the time. The error in the high state is an order of magnitude lower. Reducing the error threshold will decrease the speed of the RRA-cov (I) algorithm. This compromise is captured by the performance differences between RRA-cov (I) (faster and less accurate) and RRA-cov (II) (slower and more accurate) partitioning schemes (see Table IV).

IX. SUMMARY AND FUTURE WORK

In this paper, we introduce a general formulation that decomposes a system of biochemical reactions into restricted and unrestricted sets, then rigorously derives evolution equations for both. This representation gives rise to a modified form of the CME, the CME-RR, which untangles the probability of occurrence for a restricted reaction [Eq. (6)] from the probability distribution due to the unrestricted reactions [Eq. (7)]. Importantly, the result of this decomposition unfolds as a set of equations that generalize some previous approximations[19,21] and converge to others as a limiting case.[22] In that sense, we have succeeded in rigorously deriving the exact equations and have illustrated how and when previous approximations arise.

Some of the previous work has ignored the coupling of the restricted reactions in Eq. (7). Our framework accounts for this coupling, a feature that was demonstrated through example to be important. However, we do recognize that in the limit of a large separation in timescales, the contribution of this coupling becomes negligible and the cost of computing it might not outweigh the gain in accuracy. In this case, using established algorithms such as the ssSSA may be more efficient.

Starting from the exact derivation, we devised the RRA algorithm for the exact system and the approximate moment system. The accuracy and speed of the moment system are presented in a series of examples. We found that the algorithm is crucially dependent on accurate sampling from P_u. In the case where P_u is analytically or numerically solvable, the accuracy of the algorithm is excellent, i.e., the static partition tests for the oscillator and bistable examples. Our process of sampling from the distribution P_u knowing only its moments (Appendix D) has not been optimized, and we are currently investigating more general methods. We are also investigating the applicability of alternative moment truncation approximations based on log-normal closure methods[26] while still truncating at the covariances.

The bistable and oscillator examples show many regions of parameter space where the RRA algorithm is more efficient than, but has comparable accuracy, to the SSA. There are parameter regimes, however, where it seems to be less efficient. This is strongly dependent on the optimization of the partitioning. For the oscillator example, the simple propensity threshold method (type I) was the most robust in terms of accuracy and efficiency with respect to the reaction rate scaling. In the bistable system, partitioning all reactions as unrestricted (type I) was the most efficient, while type II partitioning method provided the most accuracy. Ultimately, for a given level of accuracy, optimizing the dynamic partitioning by minimizing the number of computations/operations per unit time will be the most efficient.

Despite these sources of inefficiency in the implementation of the RRA, there are two important features of the algorithm that we would like to emphasize. First, the specific formulation makes the RRA method dynamically applicable, limiting to the SSA on one extreme and the ssSSA on the other but also being able to tackle situations where timescales are spread over multiple scales. Second, while the computational cost of the SSA seems to scale directly with the increase in the number of molecular species and stiffness of the problem, the cost of the RRA algorithm stays relatively constant by construction. These are valuable properties that underlie the applicability of our method to the study of realistic complex biological problems.

In this work, we have not attempted to compare our results to those of τ-leaping algorithms. We believe that there will be cases when these algorithms may be more efficient/accurate and vice versa. Ultimately, different simulations algorithms should be coupled and utilized depending on the specifics of any given problem. However, we would like to point out a crucial distinction that differentiates our framework. While τ-leaping algorithms jump from one state to the next and generate a single trajectory, the RRA algorithm evolves the distribution P_u between any two states, therefore generating information that can be exploited to derive various characteristics of the system.

ACKNOWLEDGMENTS

This work was supported by the UCSF Program for breakthrough biomedical research and the NIH to H.E. We would like to thank members of the El-Samad Laboratory for critical reading of the manuscript.

APPENDIX A: DERIVATION OF EQUATIONS (6) and (7)

In this section we derive Eqs. (6) and (7). Since $W(x,t|n_s,t_s)=P_{\mathrm{nr},r}(t)P_u(x,t|n_s,t_s)$, then

$$\frac{\partial W(x,t|n_s,t_s)}{\partial t}=\frac{\partial P_{\mathrm{nr},r}(t)}{\partial t}P_u(x,t|n_s,t_s)+P_{\mathrm{nr},r}(t)\frac{\partial P_u(x,t|n_s,t_s)}{\partial t}$$
$$=P_{\mathrm{nr},r}(t)\left(\sum_{j=1}^{K}[a_{u_j}(x-\nu_{u_j})P_u(x-\nu_{u_j},t|n_s,t_s)-a_{u_j}(x)P_u(x,t|n_s,t_s)]-\sum_{\ell=1}^{M-K}a_{r_\ell}(x)P_u(x,t|n_s,t_s)\right), \tag{A1}$$

where we have replaced $W(x,t|n_s,t_s)$ in Eq. (3) with $P_{\mathrm{nr},r}(t)P_u(x,t|n_s,t_s)$ throughout.

Summing over x (Σ_x) and using the fact that $\Sigma_x P_u(x,t|n_s,t_s)=1$, we obtain

$$\frac{\partial P_{\mathrm{nr},r}(t)}{\partial t}=P_{\mathrm{nr},r}(t)\left(\sum_x\sum_{j=1}^{K}[a_{u_j}(x-\nu_{u_j})P_u(x-\nu_{u_j},t|n_s,t_s)-a_{u_j}(x)P_u(x,t|n_s,t_s)]-\sum_x\sum_{\ell=1}^{M-K}a_{r_\ell}(x)P_u(x,t|n_s,t_s)\right)$$
$$=-P_{\mathrm{nr},r}(t)\sum_x\sum_{\ell=1}^{M-K}a_{r_\ell}(x)P_u(x,t|n_s,t_s). \tag{A2}$$

Equation (6) can be easily be seen as the solution to Eq. (A2). Substituting the explicit expression for $\partial P_{\mathrm{nr},r}(t)/\partial t$ from Eq. (A2) into Eq. (A1) yields

$$-P_{\mathrm{nr},r}(t)\sum_{x'}\sum_{\ell=1}^{M-K}a_{r_\ell}(x')P_u(x',t|n_s,t_s)P_u(x,t|n_s,t_s)+P_{\mathrm{nr},r}(t)\frac{\partial P_u(x,t|n_s,t_s)}{\partial t}$$
$$=P_{\mathrm{nr},r}(t)\left(\sum_{j=1}^{K}[a_{u_j}(x-\nu_{u_j})P_u(x-\nu_{u_j},t|n_s,t_s)-a_{u_j}(x)P_u(x,t|n_s,t_s)]-\sum_{\ell=1}^{M-K}a_{r_\ell}(x)P_u(x,t|n_s,t_s)\right). \tag{A3}$$

Dividing through by $P_{\mathrm{nr},r}(t)$ and rearranging the terms yields Eq. (7). The effect of the final term in Eq. (7) is essentially to renormalize $P_u(x,t|n_s,t_s)$ and enforce that $\Sigma_x P_u(x,t|n_s,t_s)=1$. This term also makes Eq. (7) a nonlinear equation.

APPENDIX B: MOMENT DERIVATIONS

To calculate the time-dependent equations for moments, we use the method by Engblom[25] where moment equations

for the CME were derived by assuming quadratic propensities. Time-dependent moment equations can be computed using test functions $T(x)$ and a summation over x in Eq. (7). The general form of the time-dependent moment equation is

$$\begin{aligned}
&\sum_x T(x)\frac{\partial P_u(x,t|n_s,t_s)}{\partial t} \\
&\quad = \sum_x T(x)\sum_{j=1}^{K}[a_{u_j}(x-\nu_{u_j})P_u(x-\nu_{u_j},t|n_s,t_s) \\
&\quad - a_{u_j}(x)P(x,t|n_s,t_s)] - \sum_x T(x)\sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t|n_s,t_s) \\
&\quad + \left[\sum_{x'}\sum_{\ell=1}^{M-K} a_{r_\ell}(x')P_u(x',t|n_s,t_s)\right] \\
&\quad \times \sum_x T(x)P_u(x,t|n_s,t_s). \qquad \text{(B1)}
\end{aligned}$$

For our purposes $T(x)$ will either be $T(x)=x_k$, the first moment of the kth species, or $T(x)=[x_k-z_k(t)][x_{k'}-z_{k'}(t)]$, the second central moment of the kth and k'th species. Obtaining a given time-dependent moment equation involves transforming the propensities in Eq. (B1) to their Taylor series form from Eq. (10). After some manipulation, one obtains terms for moments of various orders. Terms of the form $\Sigma_x x_j P_u(x,t|n_s,t_s)$ are then represented by $z_j(t)$. Terms of the form $\Sigma_x[x_j-z_j(t)]P_u(x,t|n_s,t_s)$ are zero. Second order central moments $\Sigma_x[x_j-z_j(t)][x_{j'}-z_{j'}(t)]P_u(x,t|n_s,t_s)$ can be represented by the covariance $C_{jj'}$. Any third or fourth order terms are left in their integral form. The terms in Eqs. (12) and (13) involving *unrestricted* reactions are taken directly from the work of Engblom.[25] However, the terms involving *restricted* reactions, i.e., the last two terms on the right in Eq. (B1), have to be derived for each time-dependent moment of interest. We first analyze the term $\Sigma_x T(x)\Sigma_{\ell=1}^{M-K}a_{r_\ell}(x)P_u(x,t|n_s,t_s)$. When $T(x)=x_k$, we have

$$\begin{aligned}
&\sum_x x_k \sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t|n_s,t_s) \\
&\quad = \sum_{\ell=1}^{M-K} a_{r_\ell}(z(t))z_k(t) + \sum_{i=1}^{N}\left.\frac{\partial a_{r_\ell}(x)}{\partial x_i}\right|_{x=z(t)} C_{ik} \\
&\quad + \sum_x\sum_{i,i'}^{N}\left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t)}[x_k-z_k(t)+z_k(t)] \\
&\quad \times \frac{[x_i-z_i(t)][x_{i'}-z_{i'}(t)]}{2}P_u(x,t|n_s,t_s), \qquad \text{(B2)}
\end{aligned}$$

which contributes to Eq. (12). When $T(x)=[x_k-z_k(t)]\times[x_{k'}-z_{k'}(t)]$, the same term becomes

$$\begin{aligned}
&\sum_x [x_k-z_k(t)][x_{k'}-z_{k'}(t)]\sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t|n_s,t_s) \\
&\quad = \sum_{\ell=1}^{M-K} a_{r_\ell}(z(t))C_{kk'} + \sum_x\left(\sum_{i=1}^{N}\left.\frac{\partial a_{r_\ell}(x)}{\partial x_i}\right|_{x=z(t)}[x_k-z_k(t)]\right. \\
&\quad \times[x_{k'}-z_{k'}(t)][x_i-z_i(t)] + \sum_{i,i'}^{N}\left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t)} \\
&\quad \left.\times[x_k-z_k(t)][x_{k'}-z_{k'}(t)]\frac{[x_i-z_i(t)][x_{i'}-z_{i'}(t)]}{2}\right) \\
&\quad \times P_u(x,t|n_s,t_s), \qquad \text{(B3)}
\end{aligned}$$

which contributes to Eq. (13). For second term on the right hand side of Eq. (B1) we have

$$\begin{aligned}
&\left[\sum_{x'}\sum_{\ell=1}^{M-K} a_{r_\ell}(x')P_u(x',t|n_s,t_s)\right]\sum_x T(x)P_u(x,t|n_s,t_s) \\
&\quad = \left[\sum_{\ell=1}^{M-K}\left(a_{r_\ell}(z(t)) + \sum_{i,i'}^{N}\left.\frac{\partial^2 a_{r_\ell}(x')}{\partial x_i'\,\partial x_{i'}'}\right|_{x=z(t)}\frac{C_{ii'}}{2}\right)\right] \\
&\quad \times \sum_x T(x)P_u(x,t|n_s,t_s), \qquad \text{(B4)}
\end{aligned}$$

which then contributes to the time-dependent moment equation for a particular $T(x)$. Equations (12) and (13) represent the final contributions of adding up all the derived moment terms.

APPENDIX C: DERIVATION OF $P(t',\mu|t'<t_s+\tau_s)$

The probability that a restricted reaction occurred before t' would simply be $P_{r,r}(t')=1-P_{\mathrm{nr},r}(t')$, both of which are cumulative distribution functions. Therefore the PDF for a restricted reaction to occur as a function of t' would simply by $P(t')=\partial P_{r,r}(t')/\partial t'=-\partial P_{\mathrm{nr},r}(t')/\partial t'$. Directly applying Eq. (A2) yields

$$\begin{aligned}
P(t') &= \exp\left(-\int_{t_s}^{t'}\sum_x\sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t|n_s,t_s)dt\right) \\
&\quad \times \sum_x\sum_{\ell=1}^{M-K} a_{r_\ell}(x)P_u(x,t'|n_s,t_s). \qquad \text{(C1)}
\end{aligned}$$

Next, the conditional PDF for a restricted reaction to occur as a function of t', given that it occurred before $t_s+\tau_s$, would be

$$P(t'|t'<t_s+\tau_s) = \frac{P(t')}{1-P_{\mathrm{nr},r}(t_s+\tau_s)}. \qquad \text{(C2)}$$

Finally, the probability that a restricted reaction of type μ occurs given a restricted reaction occurred at time t', by construction, is

$$P(\mu|t') = \frac{\Sigma_{x'} a_{r_\mu}(x') P_u(x',t'|n_s,t_s)}{\Sigma_{x'}\Sigma_{\ell=1}^{M-K} a_{r_\ell}(x') P_u(x',t'|n_s,t_s)}, \tag{C3}$$

which represents the weighted sum over all possible states from which a restricted reaction of type μ could occur, divided by the weighted sum over all possible states from which a restricted reaction could occur. Therefore $P(t',\mu|t'<t_s+\tau_s)=P(\mu|t')P(t'|t'<t_s+\tau_s)$, yielding Eq. (8). To sample $P(t',\mu|t'<t_s+\tau_s)$, we first sample from the time of the restricted reaction $P(t'|t'<t_s+\tau_s)$ and then we sample the restricted reaction type from $P(\mu|t')$.

APPENDIX D: SAMPLING FROM THE VARIOUS DISTRIBUTIONS

Below we discuss sampling from all the distributions involved in the RRA algorithm.

1. Sampling the time of the restricted reaction

To sample t', we must sample from the cumulative distribution function of $P(t'|t'<t_s+\tau_s)$, which we know to be $(1-P_{\mathrm{nr},r}(t'))/(1-P_{\mathrm{nr},r}(t_s+\tau_s))$. We therefore generate random number p and solve $P_{\mathrm{nr},t}$ and P_u or its moments until the time t' that satisfies $p=(1-P_{\mathrm{nr},r}(t'))/(1-P_{\mathrm{nr},r}(t_s+\tau_s))$.

2. Sampling the restricted reaction

We discuss sampling from $P(\mu|t')$. Let $G(\mu|t')$ be the cumulative distribution function of $P(\mu|t')$. We generate a random number p from a uniform distribution between zero and one and then choose μ that satisfies $G(\mu-1|t')<p<=G(\mu|t')$. For the exact case

$$G(\mu|t') = \frac{\Sigma_{x'}\Sigma_{\ell=1}^{\mu} a_{r_\ell}(x') P_u(x',t'|n_s,t_s)}{\Sigma_{x'}\Sigma_{\ell=1}^{M-K} a_{r_\ell}(x') P_u(x',t'|n_s,t_s)}, \tag{D1}$$

and for the moment system

$$G(\mu|t') = \frac{\sum_{\ell=1}^{\mu}\left[a_{r_\ell}(z(t')) + \sum_{i,i'} \left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t')} \frac{C_{ii'}(t')}{2}\right]}{\sum_{\ell=1}^{M-K}\left[a_{r_\ell}(z(t')) + \sum_{i,i'} \left.\frac{\partial^2 a_{r_\ell}(x)}{\partial x_i\,\partial x_{i'}}\right|_{x=z(t')} \frac{C_{ii'}(t')}{2}\right]}. \tag{D2}$$

3. Sampling a state from $P(x|t',\mu)$ for the discrete distribution

Given the time of the restricted reaction t' and the restricted reaction type μ, we now can sample the state n_{s+1-} from $P(x|t',\mu)$. We define $G(x|t',\mu)$ as the cumulative distribution function of $P(x|t',\mu)$. This is applicable when $P_u(x,t'|n_s,t_s)$ can be represented by a finite number of N discrete states. To sample a state we generate a random number p from a uniform distribution between zero and one and then choose the state with index I that satisfies the inequality $G(I-1|t',\mu)<p<=G(I|t',\mu)$, where

$$G(I|t',\mu) = \frac{\Sigma_{x=1}^{I} a_{r_j}(x) P_u(x,t'|n_s,t_s)}{\Sigma_{x'=1}^{N} a_{r_j}(x') P_u(x',t'|n_s,t_s)}. \tag{D3}$$

4. Sampling a state from $P(x|t',\mu)$ when there is only moment information for P_u

Sampling from $P(x|t',\mu)$ when only the moment information for P_u is available is a difficult problem and an important topic of current research. We apply a heuristic method. The first step is to impose constraints on the distribution based on the restricted reaction that occurred. For instance, let us assume that the restricted reaction is a transcription of mRNA. Let us also assume that the transcription could only occur from the gene state J of N possible gene states and that the transition between gene states are unrestricted reactions. We would then impose that the gene is in the state J at time t'.

For this paper, the genes that we model use positively regulating transcription factors. The transitions between gene states are also modeled as unrestricted reactions only when transcription factor numbers are relatively abundant. When these conditions hold, the states of a given gene are independent of the states of another gene. This is because each gene is relatively independent of fluctuations in the transcription factor. This can be verified by checking the covariances between the various genes and transcription factors, ensuring they are near zero. Given that these conditions hold, we can justifiably sample the state of each gene independently. For each gene, the probability that the gene is in a particular state is simply that state's mean value. Accordingly, we can setup a cumulative distribution function for each gene and sample from it.

For the remaining species with their corresponding mean and covariances, we assume that they fit a multivariate Gaussian distribution $P_r(x)$. Technically, we must sample from the distribution, $a_{r_\mu}(x)P_r(x)$ (or rather a normalized version of it). However, the function $a_{r_\mu}(x)$ is up to quadratic and complicates the sampling. To simplify the sampling procedure, in this paper, we make an assumption that $a_{r_\mu}(x)$ is relatively constant over the range of $P_r(x)$ allowing one to sample just from the multivariate Gaussian distribution

$P_r(x)$. For the multivariate Gaussian, we apply the standard sampling through the Cholesky factoring of the covariance matrix.[35] Each species sampled from $P_r(x)$ is then rounded to the nearest non-negative integer to ensure that $n_{s+1-} \in \mathbf{Z}_+^N$. For future work, we aim to devise methods to sample the distribution more systematically.

5. Sampling a state from P_u for the discrete distribution

Given that no restricted reaction occurred by time $t_s + \tau_s$, we must sample a state from P_u. We define G_u as the cumulative distribution function of P_u. This is applicable when $P_u(x, t_s+\tau_s | n_s, t_s)$ can be represented by a finite number of N discrete states. To sample a state from P_u we generate a random number p from a uniform distribution between zero and one and than choose the state with index I that satisfies the inequality $G_u(I-1, t_s+\tau_s | n_s, t_s) < p <= G_u(I, t_s+\tau_s | n_s, t_s)$, where

$$G_u(I, t_s+\tau_s | n_s, t_s) = \frac{\sum_{x=1}^{I} P_u(x, t_s+\tau_s | n_s, t_s)}{\sum_{x'=1}^{N} P_u(x', t_s+\tau_s | n_s, t_s)}. \quad \text{(D4)}$$

6. Sampling a state from P_u when there is only moment information

Given that no restricted reaction occurred by time $t_s + \tau_s$, we must sample a state from an approximate version of P_u, i.e., $\bar{P}_u$, given only that we have only solved through the covariances. The basic procedure is the same as that from Appendix D 4, but where we do not weight the distribution by the restricted reaction propensity since a restricted reaction did not occur.

[1] T. Gregor, E. Wieschaus, A. McGregor, W. Bialek, and D. Tank, Cell **130**, 141 (2007).
[2] T. Gregor, D. Tank, E. Wieschaus, and W. Bialek, Cell **130**, 153 (2007).
[3] N. Balaban, J. Merrin, R. Chait, L. Kowalik, and S. Leibler, Science **305**, 1622 (2004).
[4] M. Kaern, T. Elston, W. Blake, and J. Collins, Nat. Rev. Genet. **6**, 451 (2005).
[5] H. McAdams and A. Arkin, Proc. Natl. Acad. Sci. U.S.A. **94**, 814 (1997).
[6] T. Kepler and T. Elston, Biophys. J. **81**, 3116 (2001).
[7] P. Swain, M. Elowitz, and E. Siggia, Proc. Natl. Acad. Sci. U.S.A. **99**, 12795 (2002).
[8] W. Blake, N. Kærn, C. Cantor, and J. Collins, Nature (London) **422**, 633 (2003).
[9] J. Raser and E. O'Shea, Science **304**, 1811 (2004).
[10] M. Levin, FEBS Lett. **550**, 135 (2003).
[11] G. Suel, R. Kulkarni, J. Dworkin, J. Garcia-Ojalvo, and M. Elowitz, Science **315**, 1716 (2007).
[12] D. McQuarrie, J. Appl. Probab. **4**, 413 (1967).
[13] D. Gillespie, Physica A **188**, 404 (1992).
[14] N. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland, Amsterdam, 1992).
[15] D. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[16] D. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[17] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie, J. Chem. Phys. **119**, 12784 (2003).
[18] M. Rathinam and H. El-Samad, J. Comput. Phys. **224**, 897 (2007).
[19] E. Haseltine and J. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[20] D. Gillespie, J. Chem. Phys. **113**, 297 (2000).
[21] C. Rao and A. Arkin, J. Comput. Phys. **227**, 100 (2003).
[22] Y. Cao, D. Gillespie, and L. Petzold, J. Chem. Phys. **122**, 014116 (2005).
[23] P. Lötstedt and A. Hellander, J. Comput. Phys. **227**, 100 (2007).
[24] E. Weinan, D. Liu, and E. Vanden-Eijnden, J. Comput. Phys. **221**, 158 (2007).
[25] S. Engblom, Appl. Math. Comput. **180**, 498 (2006).
[26] A. Singh and J. Hespanha, Bull. Math. Biol. **69**, 1909 (2007).
[27] G. Byrne and A. Hindmarsh, Int. J. High Perform. Comput. Appl. **13**, 354 (1999).
[28] S. Cohen and A. Hindmarsh, Comput. Phys. **10**, 138 (1996).
[29] A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker, and C. Woodward, ACM Trans. Math. Softw. **31**, 363 (2005).
[30] S. Balay, W. Gropp, L. McInnes, and B. Smith, *Modern Software Tools in Scientific Computing* (Birkhäuser, Basel, 1997).
[31] S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang, PETSc User's Manual, ANL-95/11-Revision 2.1.5, Argonne National Laboratory (2004).
[32] B. Munsky and M. Khammash, J. Chem. Phys. **124**, 044104 (2006).
[33] J. Vilar, H. Kueh, N. Barkai, and S. Leibler, Proc. Natl. Acad. Sci. U.S.A. **99**, 5988 (2002).
[34] J. Hasty, J. Pradines, M. Dolnik, and J. Collins, Proc. Natl. Acad. Sci. U.S.A. **97**, 2075 (2000).
[35] D. Barr and N. Slezak, Commun. ACM **15**, 1048 (1972).