Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions

Howard Salis and Yiannis Kaznessis

I. INTRODUCTION

With the observation that biological systems are inherently stochastic, the topic of stochastic chemical kinetics has received wider interest. By representing the chemically reacting system as a jump Markov process, one captures the significant fluctuations that occur when dilute chemical species are discretely altered by discontinuously occurring reactions. Many of the existing numerical methods for the solution of well-mixed chemical kinetic systems are based on the work of Gillespie, who developed the stochastic simulation algorithm (SSA) in two variations, the Direct and First Reaction variants. Using two special data structures and efficient use of random numbers, Gibson and Bruck created the Next Reaction variant. More recently, Cao and co-workers returned to the Direct Method implementation of the SSA and demonstrated that for certain classes of systems the method can be faster than the Next Reaction variant. While the stochastic simulation algorithm produces an exact solution to the dynamics of the chemical system, the algorithm is impractical for systems where numerous reaction events occur. The method simulates each molecular reaction event, causing the computational intensity to be directly proportional to the number of reaction occurrences. For systems with fast reversible or enzymatic reactions, numerous reaction events will occur, causing the stochastic simulation algorithm to become computationally slow.

Several approximations have been proposed to address this obstacle. Rao and Arkin apply the quasisteady state approximation to intermediate molecular species of the Master equation, numerically simulating a reduced system of reactions with some success. By approximating fast reactions as Poisson distributed, Gillespie describes the tau-leap and kappa-leap numerical methods that jump through fast reaction occurrences. By making approximations on the continuity of reaction events, Gillespie changes the representation of the system to a continuous Markov process and derives a chemical Langevin equation (CLE), which is a multiplicative Itô stochastic differential equation. These approximations describe the system as a single mathematical representation; in effect, the system is either discrete-stochastic (SSA) or continuous-stochastic (CLE). However, if the approximation is not valid for a subset of the system, the numerical accuracy will suffer. By representing subsets of the system by separate mathematical descriptions and merging them self-consistently, one creates a hybrid method that reproduces an accurate solution while decreasing the computational cost.

Haseltine and Rawlings propose a stochastic hybrid method that partitions a system into subsets of fast and slow reactions, representing the fast and slow subsets as con-
tuous and jump Markov processes, respectively. They specifically proposed two variants of the direct method for hybrid simulation: one that exactly determines the slow reaction waiting times, and one that approximately determines the slow reaction waiting times. Their methods numerically integrate the chemical Langevin equation of the system, as affected by the fast reactions, while proposing to compute the slow reaction waiting times by evaluating a constraint, derived from the time-dependent probability density of the Direct variant of the stochastic simulation algorithm. In the presented implementation these authors approximated the slow reaction waiting times using the time-independent constraint and were required to introduce a propensity of “no slow reaction” to ensure accuracy.

Puchalka and Kierzek propose another hybrid method, the “maximal time step” method, which partitions the system into fast and slow reactions, simulates the slow reactions using the time-independent Next Reaction variant of the stochastic simulation algorithm, and approximates the fast reactions from a Poisson distribution. The time step of the simulation is the minimum of the next reaction time and a user-selected maximal time step.

In this paper, we describe a hybrid method that produces an accurate approximation to the solution of a well-mixed system of chemical or biochemical reactions whose kinetics partly operate in the stochastic regime. Similar to previous work, we partition the system of reactions into “fast” and “slow” subsystems and approximate the subsystem of fast reactions as a continuous Markov process, using the chemical Langevin equation to describe the fast dynamics. We propose using the time-dependent probability density of the Next Reaction variant of the stochastic simulation algorithm and introduce a new method to compute the waiting times of the slow reactions, constrained by a system of integral algebraic equations, which we will refer to as “jump equations.” By equating the jump equations to a system of residuals, we may compute the reaction times of each reaction by simply monitoring the zero crossings of the residuals over time. We also propose an approximation that allows for the execution of multiple slow reactions within a single numerical integration of the chemical Langevin equation, increasing computational efficiency with only a marginal decrease in accuracy. Event times sampled from other time-dependent probability densities may be computed, but we will restrict ourselves to the exponential distribution used in stochastic chemical kinetics. The resulting system of stochastic differential and jump equations respectively describes the occurrences of the fast and slow reactions and is simultaneously solvable using a variety of stochastic numerical integrators, including implicit, adaptive and higher order methods. Here, we will restrict ourselves to the Euler–Maruyama method for simplicity.

The authors expect this numerical method to be useful when simulating any chemical or physical system operating in the stochastic kinetic regime and containing both fast and slow reactions or other transitions. These types of systems are common in biology, where fast enzymatic reactions often modify regulatory proteins, which in turn discretely adhere to DNA binding sites to initiate a critical event. Whole cell simulators will require a stochastic hybrid method in order to simulate the hundreds of thousands of reactions, both fast and slow, whose molecular reactant species may exist at high or extremely dilute concentrations.

We should stress here that for our hybrid method, which combines representations of a coupled jump and continuous Markov process, the transitions in the jump Markov process may be described by any computable distribution, including, but not limited to, the Direct or Next Reaction exponential distributions. We have not tied the theory behind our method to any probability distribution, whether it is the “Direct,” “First Reaction,” “Next Reaction,” or any other exponential distribution. In fact, all of these distributions are mathematically equivalent. One might want to use the Direct probability distribution in the event where it can result to computational gains, as Cao and co-workers discussed. Practically, we implemented the method using the Next Reaction distribution because it has properties that are useful to us. Specifically as described in Methods, we use gamma-distributed reactions to simulate transcriptional and translational elongation steps in gene expression and their implementation is not straightforward with the Direct method.

The paper is organized into three major sections: First, we discuss the fundamental theory behind the partitioning of the system, the approximation of the fast reactions, and the formulation of the jump equations. Next, we outline the algorithms of the proposed hybrid stochastic methods, both with and without the multiple slow reaction approximation. We then provide multiple examples, first validating the accuracy of the proposed methods, demonstrating their usefulness in a highly dynamic biological system, and finally comparing them to Haseltine and Rawling’s direct hybrid stochastic method in both a simple crystallization example and a set of large-scale system benchmark models.

II. THEORY

We define our system as a well-mixed volume, \( V \), containing \( N \) unique chemical species that participate in \( M \) reactions. The state of the system, \( \mathbf{X} \), (a \( N \)-vector) consists of the numbers of molecules of the chemical species. The stoichiometric matrix, \( \mathbf{v} \), (an \( M \times N \) vector) defines the reactants and products of reactions. The reaction propensities, \( a_i \), (a \( M \)-vector) are the probabilistic rates of the reactions, where \( a_i \, dt \) is the probability of the \( j \)th reaction occurring in a small time increment. The propensities may be computed using different rate laws, such as mass action kinetics or Michaelis Mentin type kinetics. The time increment for the numerical integration of the chemical Langevin equation is \( \Delta t \).

A. System partitioning

The system is dynamically partitioned into two subsets, respectively, containing all of the “fast” or “slow” reactions, where the number of fast and slow reactions is \( M^{\text{fast}} \) and \( M^{\text{slow}} \). Prior to partitioning, the entire system is described as a jump Markov process, governed by a chemical Master equation. One can classify a reaction as “fast” if it may be accurately approximated as a continuous Markov process. This approximation becomes accurate when two conditions are satisfied.
(1) The reaction occurs many times in a small time interval;
(2) The effect of each reaction on the numbers of reactants and products species is small, when compared to the total numbers of reactant and product species.

One can quantify these conditions so that, for the $j$th reaction to be classified as “fast,” the following must be true:

$$a_j(t) \Delta t \gg \lambda \gg 1,$$

(1)

and

$$X_i(t) > \varepsilon \cdot |v_{ji}| i = \{\text{reactant or product of } j\text{th reaction}\}.$$  

(2)

The two parameters, $\lambda$ and $\varepsilon$, respectively define how many reactions occur within a $\Delta t$ and how fine grained the reactant and product species must be in order for it to appear continuous-valued and not discrete-valued. As $\varepsilon$ and $\lambda$ increase towards infinity, we reach the thermodynamic limit of the chemical Master equation, where this approximation becomes exact. However, practical values for $\varepsilon$ and $\lambda$ are 100 and 10, respectively. Both the reaction propensities and state vector change over time, requiring these conditions to be reevaluated multiple times throughout the simulation and each reaction to be reclassified as “fast” or “slow.”

The theory behind the partitioning of the chemical Master equation into multiple subsets has been previously covered. Here, we briefly outline the procedure and illustrate the mathematical consequences of restricting the simulation to executing one or multiple slow reactions per $\Delta t$. One constructs the chemical Master equation of a system, which describes the joint probability density of the number of occurrences of both fast and slow reactions, respectively $r^f$ and $r^s$. The joint probability density is separated into two components: the conditional probability of a slow reaction occurring, $P(r_s^t)$, and the marginal probability of a fast reaction occurring, $P(r^f|t)$, where

$$P(r^s,r^f|t) = P(r^f|t)P(r^s).$$  

(3)

Differentiating Eq. (3) with respect to time yields

$$\frac{dP(r^s,r^f|t)}{dt} = \frac{dP(r^f|t)}{dt}P(r^s|t) + \frac{dP(r^s|t)}{dt}P(r^f|t).$$  

(4)

By restricting ourselves to executing one slow reaction at a time, we find that, for the slow reaction’s waiting time, the probability of a slow reaction, conditioned on a constant number of fast reaction occurrences, does not change over time. The derivative of the conditional probability vanishes and Eq. (4) simplifies to

$$\frac{dP(r^s,r^f|t)}{dt} = \frac{dP(r^s|t)}{dt}P(r^f|t).$$  

(5)

Substituting Eq. (5) back into the chemical Master equation produces the governing equation on how the probability density of the number of fast reaction occurrences changes over time, constrained by the occurrences of the slow reactions.

If we allow multiple slow reactions to occur within a small time increment, updating the conditional probability $P(r^f|t)$ after each slow reaction occurrence, but ignoring the changes in $[dP(r^s|t)]/dt$, then Eq. (5) is only an approximation. For small time increments, the effect of the slow reactions on the probabilities of fast reaction occurrences is small. In the example section, we will ascertain the accuracy of this approximation using differently sized time increments.

B. The chemical Langevin equation

We approximate the subset of fast reactions as a continuous Markov process and use the chemical Langevin equation to simulate the stochastic dynamics of the system as affected by only the fast reactions. The chemical Langevin equation has been derived and justified previously by approximating the occurrences of fast reactions as a Gaussian distribution. One may also derive the chemical Langevin equation by rewriting the chemical Master equation in terms of extensive variables, using a volume, and then proceeding to take the thermodynamic limit, where only the first two moments of the resulting probability distribution are retained. We dynamically evaluate the validity of this approximation using Eqs. (1) and (2).

With either derivation, the result is a system of Ito stochastic differential equations with multiple multiplicative noises, written as

$$dX_i = \sum_{j=1}^{M^f} v_{ji}a_j^f(X(t))dt + \frac{1}{2} \sum_{j=1}^{M^f} v_{ji}a_j^f(X(t))dW_j,$$

(6)

where $a^f$ is now only the fast reaction propensities, the stoichiometric matrix, $v$, is correspondingly altered, and $W$ is an $M^f$ dimensional Wiener process, responsible for producing the Gaussian noise within the system. Whereas the solution to the chemical Master equation describes the probability density of the system existing at a certain state at an instance of time, the solution to the chemical Langevin equation describes one possible trajectory of the numbers of chemical species over time.

Like ordinary differential equations, stochastic differential equations may be numerically integrated using a variety of numerical methods. However, the differences between the respective numerical methods are great. For an overview of the differences and the derivation of stochastic numerical integrators, there is an excellent reference. In addition to the common Euler, or Euler–Maruyama, method, there exist numerous higher order schemes that require the sampling of multidimensional Ito integrals. There are also derivative-free stochastic Runge Kutta, fully implicit and adaptive methods. These more advanced methods will surely become the workhorse of future applications. For now, we limit ourselves to the Euler method, but note that existing and newly proposed hybrid stochastic methods should be constructed to take advantage of the increased time increments that become acceptable when using more advanced methods.
C. The jump equations

Having treated the simulation of the fast subset of reactions, we now turn to simulating the occurrences of the slow subset by constructing the system of integral algebraic constraints, or “jump equations.” We refer to them as jump equations because their solution produces the time at which the system jumps from one state to another. We then propose a fast mechanism for determining when slow reactions occur that enables one to capture the time-dependence of the slow reaction propensities while remaining computationally efficient.

In order to sample the possible times at which the slow reactions may occur, we use a Monte Carlo technique that equates the integration of a time-dependent probability density to a uniform random number. In general, one can formulate a jump equation from any probability density, but not all probability densities may be amenable to producing a solution. For the \( j \)th slow reaction in a system, a general probability density may be defined as

\[
P_j(\tau_j|a_j(t),X(t)) = \int_t^{t_0 + \tau_j} dx a_j(t')X(t) dt = \]

The probability that the \( j \)th slow reaction occurs within the interval \([t_0 + \tau_j, t_0 + \tau_j + dt]\), given that the slow reaction occurs with probabilistic rate \( a_j'(t) \), and conditioned on the entire state history.

The corresponding jump equation, governing the reaction times, is then

\[
\int_{t_0}^{t_0 + \tau_j} P_j(\tau_j|a_j'(t'),X(t)) = x_n \ldots X(t_0) = x_0, t_0) dt' - \text{URN}_j = 0,
\]  

(7)

where \( t_0 \) is the time in which the reaction last occurred, \( \tau_j \) is the reaction time for the \( j \)th slow reaction, and \( \text{URN}_j \) is the \( j \)th uniform random number within \((0,1)\). To solve Eq. (7), we rearrange in terms of a residual and set a variable upper bound on the integral, so that

\[
\int_{t_0}^{t_0 + \tau_j} P_j(\tau_j|a_j'(t'),X(t)) = x_n \ldots X(t_0) = x_0, t_0) dt' - \text{URN}_j = \text{R}_j|t'|
\]

(8)

where the initial condition of the residual is \( \text{R}_j|t' = -\text{URN}_j \). If the probability density, abbreviated \( P(\tau_j) \), is monotonically nondecreasing over a small time increment, \( \Delta t \), such that

\[
\int_{t_0}^{t_0 + \tau_j + \Delta t} P(\tau_j) dt' \geq \int_{t_0}^{t_0 + \tau_j} P(\tau_j) dt' \quad \forall k > 0, \]

(9)

then one can determine whether the \( j \)th slow reaction has occurred by simply monitoring the sign of the residual, according to

\[
\text{R}_j|t' < 0 \rightarrow \tau_j, \quad \text{R}_j|t' > 0 \rightarrow \tau_j, \quad \text{R}_j|t' = 0 \rightarrow t = \tau_j,
\]

(10)

where \( t \) is the current simulation time.

Equation (10) suggests a relatively simple mechanism for determining whether a slow reaction has occurred within a given time increment. Starting from a simulation time, \( t \), with state vector \( X(t) \), we increment the time by \( \Delta t \), compute \( X(t + \Delta t) \), assuming that no slow reactions occur in \([t, t + \Delta t]\), and evaluate the integral and residual in Eq. (8), using \( X(t) \) and \( X(t + \Delta t) \). If the residual has performed a zero crossing, then a slow reaction has occurred within the interval \((t, t + \Delta t)\). We can then more exactly solve for the slow reaction time, \( \tau_j \), using the state vector \( X(t) \), the integral of the probability density, and the residual evaluated at time \( t \). In addition, when solving a system of multiple jump equations, we know that if only the \( j \)th residual performs a zero crossing in the interval \((t, t + \Delta t)\), then only the \( j \)th slow reaction has occurred within the interval \((t, t + \tau_j)\) and no others. This fact may help us to solve Eq. (8) more exactly.

Now we will define the previously used “Next Reaction” probability density,\(^{12}\) insert into Eq. (8), and simplify. The Next Reaction probability density for the \( j \)th slow reaction, or

\[
P_j(\tau_j) = a_j'(t + \tau_j) \exp \left( - \int_{t_0}^{t_0 + \tau_j} a_j'(t') dt' \right)
\]

(11)

is a time-dependent exponential distribution with rate \( a_j'(t) \). There is one for each slow reaction in the system. Substituting Eq. (11) into Eq. (8) and simplifying yields

\[
\int_{t_0}^{t_0 + \tau_j} a_j'(t') dt' + \text{log}(\text{URN}_j) = 0 \quad j = 1, \ldots, M^\text{slow},
\]

(12)

which forms the system of jump equations we will solve in order to compute the slow reaction times. Note that if the slow reaction propensities are time-independent, then Eq. (12) simplifies to

\[
\tau_j = \frac{-\text{log}(\text{URN}_j)}{a_j'}, \quad j = 1, \ldots, M^\text{slow},
\]

(13)

which is the same algebraic expression used to compute the reaction times in the First or Next Reaction variants of the stochastic simulation algorithm.

Again, we rearrange the jump equations in terms of a system of residuals and change the integral’s upper bound to a variable so that we now have

\[
\int_{t_0}^{t_0 + \tau_j} a_j'(t') dt' + \text{log}(\text{URN}_j) = \text{R}_j|t' \quad j = 1, \ldots, M^\text{slow},
\]

(14)

where, by using Eq. (10), we can determine within what time increment the \( j \)th reaction has occurred by monitoring the zero crossing of the \( j \)th residual. We can evaluate the integral of the slow reaction propensities using a Riemann sum. We
can also convert Eq. (14) from an integral algebraic equation to a differential one by taking the derivative with respect to time, yielding
\[
\frac{dR_j}{dt} = a_j(t)dt, \quad IC: R_j|_{t^0} = \log(URN_j), \quad j = 1, \ldots, M_{\text{slow}}.
\]  
(15)

This is actually an Ito stochastic differential equation because the slow reaction propensities are a function of the state of the system, which is a stochastic process. Equivalently, to using a Riemann sum to evaluate the integral in Eq. (14), we may use the Euler method to numerically integrate Eq. (15). However, if we wish to evaluate the jump equations with more accuracy, we must use a higher order stochastic numerical integrator.

If we restrict the method to executing only one slow reaction event per time increment, then we may not allow two or more residuals to perform a zero crossing within the same time increment. If this occurs, we must rewind the state of the system to a previous one, decrease the \( \Delta t \) of the stochastic numerical integrator, and repeat the method until only one residual performs a zero crossing. By carefully selecting the \( \Delta t \), one may aim to reduce the number of rewrites of the system. On the other hand, when using the multiple slow reaction approximation, multiple residuals may perform zero crossings and their corresponding slow reactions are executed in the order of occurrence.

Once a single residual has performed a zero crossing, the reaction time of the corresponding slow reaction must be computed. In order to do this, we may expand Eqs. (14) or (15) through an Ito–Taylor series and approximate the slow reaction time from a time \textit{just prior} to the zero crossing. In this manner, we retain the time-dependence of the slow reaction propensities on the interval \((t_0, t_0 + t')\) and approximate the slow reaction propensity within the interval \((t_0 + t', t_0 + \tau_j)\), where \( t' \) is the time just prior to the \( j \)th residual’s zero crossing. The difference \( \tau_j - t' \) is always less than \( \Delta t \), the time increment of the numerical integration of the chemical Langevin equation. Because this difference is typically small, we retain only the first term, yielding an algebraic expression that solves for the slow reaction times, \( \tau_j \), at time \( t' \), which is
\[
\tau_j = -\frac{R_j}{a_j} + t'.
\]  
(16)

The residual is negative at time \( t' \) because the zero crossing occurs in the following time increment. We will use this expression to compute the slow reaction times after having monitored the occurrence of a slow reaction. If the time increment, \( \Delta t \), becomes large, additional terms of the Ito–Taylor expansion may be retained to compute a more accurate slow reaction time.

Another common exponential distribution is the one used by the Direct method.\(^{10}\) One may substitute the Direct probability distribution into Eq. (8) and create a single Jump equation. When any reaction in the system has occurred, the residual will perform a zero crossing in the interval. One then randomly selects the occurring reaction, weighted by the reaction propensities. In addition to the exponential distribution, other useful time-dependent probability densities may be used. The advantage of representing each slow reaction by its own jump equation is that multiple different probability densities may be simultaneously used, both time independent and dependent. The probability densities may also be functions of temperature, volume, or other physical or environmental quantities. For example, the usage of the gamma distribution in biological systems has been previously covered.\(^{12}\)

III. NUMERICAL IMPLEMENTATION

The “Next Reaction” hybrid algorithms with and without the “Multiple Slow Reaction” approximation are implemented in Fortran95 in a Sun Solaris environment, using the NetCDF file format. All simulations were performed on a Sun UltraSparc workstation with a 950 MHz processor and 6 GB of RAM. Haseltine and Rawlings’ Direct hybrid algorithm,\(^{17}\) including the scaled stochastic time step and the propensity of “no slow reaction,” is also implemented for comparison, using the same data structures and file format as the newly proposed methods. We use the Next Reaction variant of the stochastic simulation algorithm as a baseline “exact” method.

The Next Reaction hybrid algorithms consist of a two-part series of computations and decision-making. We briefly outline them: The chemical Langevin equation is numerically integrated forward from \( X(t) \) to \( X(t + \Delta t) \) and the residuals of the jump equations are evaluated using \( X(t) \). If one or more residuals perform a zero crossing, then, using the multiple slow reaction approximation, all of the slow reactions that occur within \((t, t + \Delta t)\) are executed in order, their reaction times computed and the state of the system updated. Without the approximation, if two residuals perform a zero crossing then the system is rewound and the \( \Delta t \) is divided by some constant. If only one residual performs a zero crossing then the reaction is executed at its computed reaction time, updating the system correspondingly. The algorithm then repeats until a desired time is reached.

For compactness, \( X(t) \) is the state vector evaluated at time \( t \), \( a(t) \) is a reaction propensity evaluated at time \( t \), and \( a(X) \) is a reaction propensity evaluated when the state vector is \( X \). The initial number of chemical species is \( X_0 \). Variables ending in last are the save points for possible rewinding. The words stochastic simulation algorithm, stochastic differential equation, chemical Langevin equation, and uniform random number are abbreviated as SSA, SDE, CLE, and URN, respectively.

A. Next reaction hybrid algorithm

1. Initialize the system:
   \begin{enumerate}
   \item \( X = X_0 \), \( X_{\text{last}} = X_0 \), \( t = t_{\text{Start}} \), \( t_{\text{last}} = t_{\text{Start}} \), \( \Delta t = \Delta t_{\text{SDE}} \),
   \item \( R = \log(URN) \), \( R_{\text{last}} = \log(URN) \).
   \end{enumerate}
2. Time iterative loop, stop when \( t = t_{\text{End}} \)
   \begin{enumerate}
   \item Classify the reactions as fast or slow using Eqs. (1) and (2).
   \item If no fast reactions exist, perform the SSA and go to step (2a).
   \end{enumerate}
(b) Compute slow and fast reaction propensity... 
(c) Numerically integrate the CLE from \( t \) to \( t + \Delta t \), using only \( a^f(t) \).
(d) Evaluate the residuals of the jump Eqs. (14):
(i) For all slow reactions, \( j, R^\text{new}_j = R_j + a^f_j(t) \Delta t \)
(e) Count the number of zero crossings of the residuals:
(i) \( ZC = \text{Count}(R^\text{new} > 0) \)
(f) Perform an action based on the number of zero crossings:
(i) If \( ZC = 0 \), then \( X = X(t + \Delta t), \ t = t + \Delta t \)
(ii) If \( ZC = 1 \), then the \( \mu \)th slow reaction has occurred:
1. Compute \( \tau_\mu = -R_\mu/a^s_\mu(t) \).
2. Numerically integrate the CLE from \( t \) to \( t + \tau_\mu \), using the same random numbers as in step (c).
3. \( X = X(t + \tau_\mu) + v_\mu, \ t = t + \tau_\mu \)
4. For all slow reactions where \( i \neq \mu, \ R_i = R_i + a^s_i(t) \tau_\mu \)
5. Reset the \( \mu \)th residual, \( R_\mu = \log(\text{URN}_\mu) \)
6. Save the state of the system for possible rewind
7. Reset \( \Delta t \) to \( \Delta t^{\text{SDE}} \) or a more optimal value
8. If \( ZC > 2 \), then rewind the state of the system and decrease \( \Delta t \)
9. \( X = X_{\text{last}}, \ t = t_{\text{last}}, \ R = R_{\text{last}}, \ \Delta t = \Delta t/K \)
(g) Save the state of the system as often as desired. Loop to step (2).

### B. Next reaction hybrid algorithm with approximation

The Next Reaction hybrid algorithm with the multiple slow reaction approximation proceeds as before except for step (f). Instead, step (f) proceeds as follows:

(f) Perform an action based on the number of zero crossings:
(i) If \( ZC = 0 \), then \( X = X(t + \Delta t), \ t = t + \Delta t \)
(ii) If \( ZC > 0 \), then execute all slow reactions within \( (t, t + \Delta t) \)
1. Initialize: \( t^\text{sum} = 0, \ dX = 0 \)
2. For all slow reactions, \( j, \ \tau_j = -R_j/a^s_j(t) \)
3. \( \mu = \min_j(\tau_j), \ t^\text{sum} = t^\text{sum} + \mu \)
4. While \( (t^\text{sum} \leq \Delta t) \), do the following:
(a) For all slow reactions where \( i \neq \mu, \ R_i = R_i + a^s_i(t) \mu \)
(b) Reset the \( \mu \)th residual, \( R_\mu = \log(\text{URN}_\mu) \)
(c) \( dX = dX + v_\mu \)
(d) For all slow reactions, \( j, \ \tau_j = -R_j/a^s_j(X + dX) \)
(e) \( \mu = \min_j(\tau_j), \ t^\text{sum} = t^\text{sum} + \mu \)
5. For all slow reactions, \( j, \ R_j = R_j + a^s_j(X + dX) \mu(\Delta t - t^\text{sum} + \mu) \)
6. \( X = X(t + \Delta t) + dX, \ t = t + \Delta t \)

In effect, step (f.4) performs the Next Reaction variant of the stochastic simulation algorithm for the interval \( (t, t + \Delta t) \).

### IV. EXAMPLES, ERROR ANALYSIS, AND CRITICAL COMPARISONS

#### A. The cycle test

The first example will be a simple one that tests the numerical accuracy of the proposed algorithms and demonstrates the benefits of using a hybrid method when simulating a system with multiple time scales. The system of reactions is called the “Cycle Test” and is detailed in Table I. The test is run at multiple “system sizes,” \( \Theta \), where the initial conditions of the species and the kinetic constants of the slow reactions are varied in order to increase the separation between the fast and slow reaction rates. The fast reactions are numbered 1–3 and the slow reactions are numbered 4 and 5. The two slow reactions have kinetic coefficients that, for all system sizes, give them an average rate of 0.75 and 1 molecules per second, respectively. The rates of the fast reactions increase as the system size is increased.

The Cycle Test, at system sizes of 100, 1000, 10 000, and 100 000, is simulated with an end time of 100 s, using the stochastic simulation algorithm (SSA), the Next Reaction (NRH) hybrid method, and the Next Reaction hybrid method with the multiple slow reaction approximation (ANRH). The stochastic numerical integrator utilizes a maximum time step of 0.1 s. The average computational run times per trial, normalized with respect to the SSA, are shown in Table II. Using the three methods, 10 000 independent trials are performed for system sizes of 100, 1000, and 10 000. Using the Next Reaction hybrid and SSA methods, the mean and variance of the solution for a system size of 100 is shown in Fig. 1. Using all three methods, the weak mean and variance errors of representative species A and D are shown in Fig. 2.
The weak mean and variance errors are defined as

$$
\begin{align*}
\epsilon_{\text{mean}}(t) &= \left| E(X_{\text{SSA}}(t)) - E(X_{\text{Hybrid}}(t)) \right|, \\
\epsilon_{\text{variance}}(t) &= \left| \text{var}(X_{\text{SSA}}(t)) - \text{var}(X_{\text{Hybrid}}(t)) \right|.
\end{align*}
$$

The probability distributions of the species of the Cycle Test with a system size of 100 at times of 5 and 40 s are shown in Figs. 3 and 4, comparing the SSA and Next Reaction hybrid methods. It is advantageous to compare probability distributions of the solution when the solution does not

FIG. 1. Comparison of the mean (left) and variance (right) of the Cycle Test with a system size of 100, using the stochastic simulation algorithm (lines) and the Next Reaction hybrid method (dots).

FIG. 2. The weak mean (left) and variance (right) errors of species A and D of the Cycle Test at system sizes 100 (light/dotted), 1 000 (dark/dashed), and 10 000 (dark/solid), using the Next Reaction hybrid method (top) and Next Reaction hybrid method with the Multiple Slow Reaction approximation (bottom).
follow a Gaussian distribution. For species D and E, it is shown that neither have a Gaussian distribution, yet the proposed hybrid method still captures their heavy tail behavior. This becomes important when simulating the occurrences of rare, but critical events.

As the system size of the Cycle Test increases, the stochastic simulation algorithm requires more time to simulate the increasing number of reaction occurrences. However, the proposed hybrid methods approximate the increasingly fast reactions as a continuous Markov process and have the same running times for all system sizes. Therefore, as the system size increases, the Next Reaction hybrid method becomes more efficient. Figures 1 and 2 demonstrate the accuracy of the Next Reaction hybrid method, with and without the approximation, when considering only the difference between the moments of the probability distribution. When comparing the full probability distributions, Figs. 3 and 4 show that the distributions may be accurately reconstructed, within reasonable limits. These include non-Gaussian distributions such as the distributions of species D and E at both times, but especially at a time of 5 s where the distribution is noticeably heavy-tailed. These accurate results are produced with much less computational burden and a many order of magnitude speed up. By allowing reactions 4 and 5 to occur multiple times in a single time increment rather than rewinding the state of the system, the multiple slow reaction approximation increases efficiency by about 21%.

B. The pulse generator

The Pulse Generator is a simplified model of the molecular mechanisms responsible for the circadian rhythm in Drosophila fruit flies.\textsuperscript{26,27} It consists of two genes, labeled G1 and G2, each producing a monomer protein, labeled P1 and P2. The monomer proteins form a dimer, P1:P2, which

FIG. 3. Probability distributions of species A, B, C (left) and D, E (right) of the Cycle Test with a system size of 100 and a time of 5 s, comparing the SSA (lines) and Next Reaction hybrid method (dots).

FIG. 4. Probability distributions of species A, B, C (left) and D, E (right) of the Cycle Test with a system size of 100 and a time of 40 s, comparing the SSA (lines) and Next Reaction hybrid method (dots).
may bind to operator sites in genes G1 and G2 and repress expression. There are three operator sites on each gene, each capable of binding a single dimer. The operator sites feature cooperativity, where the second dimer binds faster than the first and the third dimer binds faster than the second. Expression of a gene occurs when no dimers have bound to any of its operator sites. A kinase enzyme, labeled E, may bind to both the monomer and dimer proteins and phosphorylate them, which targets them for fast degradation by proteolytic machinery in the cell. However, the kinase enzyme has a significantly lower rate of catalysis on the dimer than the monomer, causing a common Michaelis Mentin competitive inhibition scenario. The degradation step is considered immediate after a dimer or monomer is phosphorylated. The $k_{\text{cat}}$ and $K_m$ of the monomer and dimer phosphorylation is $10^4$ and $10^{-2}$ [s]$^{-1}$ and 20 mM and 10 $\mu$M, respectively. The reactions and kinetic coefficients that describe this molecular mechanism are explicitly listed in the Appendix. The resulting dynamics consist of oscillatory expression of genes G1 and G2. The amplitude of the oscillations is constant, while the period fluctuates due to the probabilistic binding and unbinding of dimers to the operator sites of both genes.

In biological systems, such as the Pulse Generator, enzymatic and dimerization reactions contribute the majority of reaction occurrences in the system. The SSA method spends most of its time resolving the individual occurrences of these types of reactions, whereas the proposed hybrid methods approximate these reactions as “fast” and describe their dynamics using stochastic differential equations. In the Pulse Generator, concurrent production of both monomers leads directly to dimerization and fast repression of both genes. While both monomers exist for only short periods of time, the SSA method consumes the majority of its computational time resolving the occurrences of the dimerization reactions. The proposed hybrid methods, however, dynamically partition the dimerization reactions into the fast reaction subsystem, where the time increment of the numerical integration of the representative stochastic differential equation is much larger than the reaction times of these reactions. Figure 5 shows the oscillatory behavior of the monomer species alongside the computational running time for both the hybrid and SSA methods. Sharp increases in the running time for the SSA method are caused by the frequent occurrences of the dimerization reactions that immediately precede the rapid degradation of both the monomers and dimers. Typical running times for the SSA, the Next Reaction hybrid method, and the Next Reaction hybrid method with the multiple slow reaction approximation are 1898, 160, and 147 s, respectively. Using the proposed hybrid methods increase efficiency by a factor of 11.9 and 12.9.

The Pulse Generator reaction system demonstrates the effectiveness of dynamic partitioning into fast and slow subsets. Only a small portion of the simulation time features fast dimerization reactions, yet the SSA method consumes the majority of its running time resolving these reaction occurrences. Instead, the proposed hybrid algorithms automatically partition reactions into fast and slow subsets and use the SSA if no fast reactions exist and the Next Reaction hybrid method otherwise. However, because the Next Reaction hybrid method is able to quickly simulate the system when fast dimerization reactions do exist, the overall running time for it is much less than the SSA.

C. Comparison with the Direct hybrid method

The Direct hybrid method of Haseltine and Rawlings similarly uses the chemical Langevin equation to approximate the occurrences of the fast reactions and allows only one slow reaction occurrence per time step, but it utilizes the probability distribution of the “Direct” variant of the stochastic simulation algorithm. We use their numerical implementation involving the usage of a scaled stochastic time step with a propensity of “no slow reaction” of 10.0. The first comparison uses a previously treated simple model of
crystallization where we analyze both the accuracy and efficiency of each hybrid method. The second comparison uses a benchmark model of a large-scale system, containing both fast and slow reactions, to analyze how each hybrid method performs with more realistic systems.

1. A crystallization example

The crystallization reaction system is depicted in Table III. It is a system with one fast and one slow reaction. To examine the effect of the stochastic numerical integrator time step on the efficiency and accuracy of each method, 10,000 trials of the crystallization example are performed while, for each simulation, setting the time step of the stochastic numerical integrator to values of 0.01, 0.05, 0.1, 0.2, or 0.5 s. The resulting weak mean and variance errors for each method are shown in Fig. 6. The time step determines the maximum time increment the stochastic numerical integrator may use, with both the Direct and Next Reaction hybrid methods sometimes utilizing a lesser time step to resolve the occurrence of a single reaction event. However, the Next Reaction method with the multiple slow reaction approximation always uses the maximum time step when numerically integrating the system of stochastic differential equations. This difference has a significant effect on the computational running times of the respective methods. Table IV lists the running times and the average number of calls to the stochastic numerical integrator function when simulating the crystallization example using different time steps.

By increasing the time step, in general, each method calls the stochastic numerical integrator less often, but the Next Reaction hybrid algorithms call the integrator less often than the Direct hybrid. Conversely, the usage of the Direct hybrid algorithm is less computationally costly per iteration,

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Rate Constant</th>
<th>Initial Concentration</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A→B</td>
<td>$k_1 = 30.11 \text{ [M s}^{-1}]$</td>
<td>$A_0 = 1e6, #C_0 = 10, #B_0 = #D_0 = 0$</td>
<td>$10^{15} \text{ L}$</td>
</tr>
<tr>
<td>2. A+C→D</td>
<td>$k_2 = 60.22 \text{ [M s}^{-1}]$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table III. The crystallization reaction system and kinetic parameters.

FIG. 6. The weak mean (top) and variance (bottom) errors of species A (left) and C (right) of the crystallization example, using the Direct (DH), Next Reaction (NRH), and Next Reaction with Multiple Slow Reaction approximation (ANRH) hybrid methods with time steps of 0.01 (triangled pointed up), 0.05 (squares), 0.1 (diamonds), 0.2 (triangled pointed down), and 0.5 (circles) seconds.
TABLE IV. Effect of time step on computational efficiency of hybrid methods.

<table>
<thead>
<tr>
<th>SDE numerical time step, Δt (s)</th>
<th>0.01</th>
<th>0.05</th>
<th>0.10</th>
<th>0.20</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct hybrid: Time (s)</td>
<td>0.141</td>
<td>0.05</td>
<td>0.032</td>
<td>0.0265</td>
<td>0.0255</td>
</tr>
<tr>
<td>SDE calls</td>
<td>10.513</td>
<td>2547</td>
<td>1588</td>
<td>1164</td>
<td>1037</td>
</tr>
<tr>
<td>Next Reaction hybrid: Time (s)</td>
<td>0.16</td>
<td>0.04</td>
<td>0.024</td>
<td>0.0195</td>
<td>0.0125</td>
</tr>
<tr>
<td>SDE calls</td>
<td>10.005</td>
<td>2005</td>
<td>1005</td>
<td>505</td>
<td>205</td>
</tr>
<tr>
<td>Next Reaction hybrid: Time (s)</td>
<td>0.142</td>
<td>0.0355</td>
<td>0.0225</td>
<td>0.015</td>
<td>0.011</td>
</tr>
<tr>
<td>MSR approximation SDE calls</td>
<td>10.000</td>
<td>2000</td>
<td>1000</td>
<td>500</td>
<td>200</td>
</tr>
</tbody>
</table>

resulting in it being faster when the number of SDE numerical integrator calls is similar to when using the Next Reaction hybrid algorithm. However, for larger time steps, the balance between decreasing the number of iterations and the higher cost per iteration of the algorithm tips in favor of the Next Reaction hybrid methods. By increasing the time step, the error of the solution of the fast reaction occurrences also increases; at a time step of 0.1 s the error is about 1%. For other, more stiff systems, where the stochastic differential equations themselves have disparate time scales, the time step must be decreased or a more advanced numerical integrator must be used.

2. A benchmark for large-scale systems

The examples previously stated are relatively simple models of more interesting systems. We now propose a straightforward benchmark to represent a large-scale system in order to analyze the ability of a hybrid method to simulate a system of many fast and slow reactions. The benchmark model is described in Table V. Here, we set $R_F$, the number of fast reactions, to 3 and vary $R_S$, the number of slow reactions, between 6 and 1000. The largest system then contains 1003 reactions and 2009 unique chemical species. We set the kinetic constants of the fast reactions to 60.22, 33.11, and 15.06 [M s$^{-1}$] respectively. For each slow reaction, we set a kinetic constant of 60.22 [M s$^{-1}$]. The initial rates of the fast reactions are 100000, 55000, and 25000 molecules per second, respectively. The initial rates of the slow reactions are each 10 molecules per second. We use a numerical time step of 0.1 s for all benchmark simulations. For $T$=200 s of simulated time, Table VI displays the typical computational running times and the number of calls to the stochastic numerical integrator, per trial, using the three hybrid methods and the stochastic simulation algorithm for increasing values of $R_S$. We include the number of rewinds the Next Reaction hybrid method performs each trial. The number of stochastic numerical integrator function calls for the Next Reaction hybrid method with the approximation is always $T/\Delta t$, which is 2000 for these simulations. The stochastic simulation algorithm executed approximately 2.7–2.8 million reaction events in each trial.

As the number of slow reactions increases, the computational cost of computing their propensities increases as well. This has dramatic effects on the Direct and Next Reaction hybrid methods (without the approximation) because each method must compute the slow reaction propensities at least once after each reaction event. Practically, this cost becomes so high that the stochastic simulation algorithm becomes faster at some critical number of slow reactions. However, using a previously described dependency graph,\textsuperscript{12} the Next Reaction hybrid method with the multiple slow reaction approximation allows one to compute the slow reaction propensities only when they have changed. For systems where many slow reactions occur within a time increment, $\Delta t$, this optimization is a significant improvement.

In addition, both the Direct and Next Reaction hybrid methods (without the approximation) must numerically integrate the chemical Langevin equation at least once per slow reaction event. As the numbers of slow reaction events increase, the Next Reaction hybrid method will more frequently rewind the state of the system in order to resolve individual occurrences. Because of these computational costs, the running times for both the Direct and Next Reaction hybrid methods (without the approximation) increase sharply with an increasing number of slow reactions.

The multiple slow reaction approximation is required in order to reduce computational running times to practical durations. Using either the Next Reaction or Direct probability distributions, inserted into Eq. (8), resulted in similar computational running times for all large-scale benchmark models. Comparing optimal implementations of the two slightly different methods, it is likely that should some slow reactions occur much more often than others, the Next Reaction probability distribution would be more efficient due to the nature of the data structures utilized. A description and comparison of optimal implementations of these two methods have been previously covered.\textsuperscript{12} In the large-scale benchmark models, each slow reaction occurs at a similar frequency, possibly leading to the insignificant difference between the usage of the Next Reaction and Direct distributions.

V. DISCUSSION

This paper proposes a novel “Next Reaction” hybrid method with an optional approximation that relieves the restriction of one slow reaction event per time step. This stochastic hybrid method generates an accurate solution to a
TABLE VI. Simulations statistics of the benchmark models of large-scale systems.

\begin{tabular}{|c|c|c|c|}
\hline
$R_s$ & SSA (s) & Direct (s) & Direct (SDE calls) \\
\hline
6 & 45 & 0.111 & 3 667 \\
30 & 83 & 0.650 & 5 801 \\
100 & 188 & 16.082 & 12 365 \\
200 & 341 & 185.05 & 21 984 \\
500 & 811 & 6290 & 50 698 \\
1000 & 1640 & $>10 000$ & $>75 000$ \\
\hline
\end{tabular}

There are two sources of error in the proposed method; the simulation of the fast reactions as a continuous Markov process and the computation of the slow reaction times, although the two are coupled if a slow reaction propensity depends on an accurate solution to the fast reaction dynamics. The former source of error has two components. The first is the validity of the initial approximation of the fast reactions as a continuous Markov process, which converts the reactions into continuously occurring events living in a continuous phase space. As the rate and the numbers of chemical species participating in the fast reactions increase, this approximation becomes more exact. The second component is the inaccuracy of the numerical integration of the chemical Langevin equation, a stochastic differential equation. By using the Euler–Maruyama method with a fixed time step, we are guaranteed to obtain weak convergence to the Ito solution with order 1.0. If we instead use the strong definition of convergence, the Euler–Maruyama method yields only order 0.5 accuracy. By choosing a more advanced stochastic numerical integrator, we may decrease this source of the error.

The second source of error arises from the inaccurate evaluation of the jump equations and the resulting error in the computation of the slow reaction times. We first consider the execution of a single slow reaction per time step. The error in the slow reaction times is directly related to the inaccuracy in the integral of the reaction propensities in the jump equation. By better evaluating this integral, we decrease the error in the slow reaction times. Examining the differential form of the jump equation in Eq. (15), we notice that we may augment the $M^{\text{fast}}$ stochastic differential equations governing the fast reactions with the differential jump equations, a system of $M^{\text{slow}}$ equations of the same type. The differential jump equations are stochastic differential equations, but do not contain a Wiener process. We may solve both sets simultaneously using the same stochastic numerical integrators we have mentioned thus far. We then quantify the accuracy of the residuals of the jump equations, and correspondingly the accuracy of the slow reaction times, in terms of the stochastic numerical integrator we use and the time step. Again, if we desire more accuracy in the slow reaction times, we may use a more advanced stochastic numerical integrator.

When executing multiple reactions per time step, the deviation of Eq. (5) from exactness will determine the extent of the error in the slow reaction times. If the state of the system, as affected by the fast reactions, changes rapidly over the time increment $\Delta t$ then the slow reaction propensities computed within the interval $(t, t + \Delta t)$ will be inaccurate. One way to decrease this error is to compute the derivative of the state of the system and estimate the state at a given time between the same interval. Because the chemical Langevin equation is not strictly differential in the normal sense, the derivative could only be an approximate one. As a final measure, the time increment may always be decreased with a corresponding decrease in efficiency.

One may also simplify this hybrid method by ignoring the fluctuations in the fast reaction dynamics and using ordinary differential equations instead of stochastic ones. The differences in the algorithm outline are minor. Instead of using a numerical integrator for a stochastic differential equation, one uses a more familiar method for ordinary differential equations, such as a Runge–Kutta method. Unless the number of molecules participating in a fast reaction is exceedingly numerous, this approximation will generate more error in the solution to the fast reaction dynamics and, consequently, in the computation of the slow reaction times.

In addition to systems of chemical or biochemical reactions, many physical processes may be quantitatively described using a combination of ordinary and stochastic differential and Master equations. This stochastic hybrid method may be used to simulate these systems by replacing the chemical Langevin equation with any system of differential equations, defining the probability densities governing each discrete transition, and rearranging to form the system of jump equations. However, this paper does not address the problem of stiffness in differential equations nor the difficulty in integrating certain probability densities.

We expect the proposed stochastic hybrid methods to be useful in the prediction and design of physical, chemical, and biological systems that contain both continuous and discrete events and interactions. With growing interest in the “small” world, these numerical tools will facilitate scientific understanding and future engineering.
ACKNOWLEDGMENTS

We thank Chetan Gadgil for discussion of the manuscript and the Supercomputing Institute of Minnesota for computational resources. This work is supported by an NIH training grant (No. GM08347), an IBM faculty award, and the Army High Performance Computing Research Center under the auspices of the U.S. Army Research Laboratory (Contract No. DAAD10-01-2-0014).

APPENDIX: THE PULSE GENERATOR REACTION SYSTEM

See Table VII.

**Table VII.** Pulse generator system of reactions and parameters.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Kinetic coeff.</th>
<th>Reaction</th>
<th>Kinetic coeff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) G1 → G1*</td>
<td>5.0</td>
<td>1st</td>
<td>(15) G1:P1:P2 → G1 + P1:P2</td>
</tr>
<tr>
<td>(2) G1* → G1 + 20 P1</td>
<td>100</td>
<td>1st</td>
<td>(16) G2:3P1:P2 → G2:2P1:P2 + P1:P2</td>
</tr>
<tr>
<td>(3) G2 → G2*</td>
<td>5.0</td>
<td>1st</td>
<td>(17) G2:2P1:P2 → G2:P1:P2 + P1:P2</td>
</tr>
<tr>
<td>(4) G2* → G2 + 20 P2</td>
<td>100</td>
<td>1st</td>
<td>(18) G2:P1:P2 → G2 + P1:P2</td>
</tr>
<tr>
<td>(5) P1:P2 → P1 + P2</td>
<td>2e8</td>
<td>2nd</td>
<td>(19) P1 + E → P1:E</td>
</tr>
<tr>
<td>(6) P1:P2 → P1 + P2</td>
<td>2.0</td>
<td>1st</td>
<td>(20) P1:E → P1 + E</td>
</tr>
<tr>
<td>(7) G1 + P1:P2 → G1:P1:P2</td>
<td>1e4</td>
<td>2nd</td>
<td>(21) P1:E → E</td>
</tr>
<tr>
<td>(8) G1:P1:P2 + P1:P2 → G1:2P1:P2</td>
<td>1.5e4</td>
<td>2nd</td>
<td>(22) P2 + E → P2:E</td>
</tr>
<tr>
<td>(9) G1:2P1:P2 + P1:P2 → G1:3P1:P2</td>
<td>3e4</td>
<td>2nd</td>
<td>(23) P2:E → P2:E</td>
</tr>
<tr>
<td>(10) G2 + P1:P2 → G2:P1:P2</td>
<td>1e4</td>
<td>2nd</td>
<td>(24) P2:E → E</td>
</tr>
<tr>
<td>(13) G1:3P1:P2 → G1:P1:P2 + P1:P2</td>
<td>1e−3</td>
<td>1st</td>
<td>(27) P1:P2:E → E</td>
</tr>
<tr>
<td>(14) G1:2P1:P2 → G1:P1:P2 + P1:P2</td>
<td>1.5e−3</td>
<td>1st</td>
<td>Volume=1e−15 L s−1</td>
</tr>
</tbody>
</table>

Initial conditions: #G1 = #G2 = 1, #E = 100 [molecules] (zero otherwise)

---