Stochastic Simulation of Chemical Kinetics

Daniel T. Gillespie

Dan T Gillespie Consulting, Castaic, California 91384;
email: GillespieDT@mailaps.org

Key Words

tau-leaping, master equation, Langevin equation, stiff systems

Abstract

Stochastic chemical kinetics describes the time evolution of a well-stirred chemically reacting system in a way that takes into account the fact that molecules come in whole numbers and exhibit some degree of randomness in their dynamical behavior. Researchers are increasingly using this approach to chemical kinetics in the analysis of cellular systems in biology, where the small molecular populations of only a few reactant species can lead to deviations from the predictions of the deterministic differential equations of classical chemical kinetics. After reviewing the supporting theory of stochastic chemical kinetics, I discuss some recent advances in methods for using that theory to make numerical simulations. These include improvements to the exact stochastic simulation algorithm (SSA) and the approximate explicit tau-leaping procedure, as well as the development of two approximate strategies for simulating systems that are dynamically stiff: implicit tau-leaping and the slow-scale SSA.

1. INTRODUCTION

Reaction-rate equation (RRE): the set of coupled, first-order, ordinary differential equations traditionally used to describe the time evolution of a well-stirred chemical system

The most accurate way to depict the time evolution of a system of chemically reacting molecules is to do a molecular dynamics simulation, meticulously tracking the positions and velocities of all the molecules, and changing the populations of the species appropriately whenever a chemical reaction occurs. Chemical reactions in this approach are viewed as distinct, essentially instantaneous physical events, and they are generally of two elemental types: unimolecular, occurring as a result of processes internal to a single molecule, and bimolecular, occurring as a result of a collision between two molecules. (Trimolecular and higher order reactions are approximate representations of sequences of two or more elemental reactions.)

From a classical mechanics viewpoint, one might suppose that such a system is deterministic, in that a given initial state always leads to the same state at some specified later time. There are three reasons why this is not so: First, even if the system evolved deterministically with respect to the positions, velocities, and molecular populations of the species, it would not evolve deterministically with respect to the species populations alone. Second, quantum indeterminacy unavoidably enters; e.g., in a unimolecular reaction we can never know exactly when a molecule will transform itself into a different isomeric form. Third, chemical systems are usually not mechanically isolated; rather, they are in contact with a heat bath, whose essentially random perturbations keep the system in thermal equilibrium at some temperature.

In view of the fact that molecular populations in a chemically reacting system are integer variables that evolve stochastically, it is remarkable that chemical kinetics has traditionally been analyzed using a mathematical formalism in which continuous (real) variables evolve deterministically: Traditional chemical kinetics holds that in a well-stirred, thermally equilibrated chemical system, the number of molecules X_i of each chemical species S_i ($i = 1, \ldots, N$) evolves in time according to a set of coupled ordinary differential equations (ODEs) of the form

$$\frac{dX_i}{dt} = f_i(X_1, \ldots, X_N) \quad (i = 1, \ldots, N) \tag{1}$$

where the functions f_i are inferred from the specifics of the various reactions. This set of equations is called the reaction-rate equation (RRE). It is usually expressed in terms of the concentration variables $Z_i \equiv X_i/\Omega$, where Ω is the system volume, but that scalar transformation is not important here. Even more remarkable is that for systems of test-tube size or larger, the RRE seems to work quite well. But if the system is small enough that the molecular populations of at least some of the reactant species are not too many orders of magnitude larger than one, discreteness and stochasticity may play important roles. Whenever that happens, and it often does in cellular systems in biology (1–7), Equation 1 does not accurately describe the system's true behavior.

Stochastic chemical kinetics attempts to describe the time evolution of a well-stirred chemically reacting system in a way that takes honest account of the system's discreteness and stochasticity. In this chapter I briefly review the theoretical foundations of stochastic chemical kinetics and then describe some recent advances in numerical-simulation strategies that are supported by this theory.

2. STOCHASTIC CHEMICAL KINETICS: THE CHEMICAL MASTER EQUATION AND THE STOCHASTIC SIMULATION ALGORITHM

Let us consider a well-stirred system of molecules of N chemical species $\{S_1, \ldots, S_N\}$, which interact through M chemical reactions $\{R_1, \ldots, R_M\}$. We assume that the system is confined to a constant volume Ω and is in thermal (but not chemical) equilibrium at some constant temperature. We let $X_i(t)$ denote the number of molecules of species S_i in the system at time t. Our goal is to estimate the state vector $\mathbf{X}(t) \equiv (X_1(t), \ldots, X_N(t))$, given that the system was in state $\mathbf{X}(t_0) = \mathbf{x}_0$ at some initial time t_0.

The justification for the tacit assumption that we can describe the system's state by specifying only the molecular populations, ignoring the positions and velocities of the individual molecules, lies in the conditions responsible for the system being well stirred. The fundamental assumption being made is that the overwhelming majority of molecular collisions that take place in the system are elastic (nonreactive), and further that the net effect of these elastic collisions is twofold: First, the positions of the molecules become uniformly randomized throughout Ω; second, the velocities of the molecules become thermally randomized to the Maxwell-Boltzmann distribution. To the extent that this happens, we can ignore the nonreactive molecular collisions, the simulation of which would occupy most of the computation time in a molecular dynamics simulation, and concern ourselves only with events that change the populations of the chemical species. This simplifies the problem enormously.

The changes in the species populations are of course a consequence of the chemical reactions. Each reaction channel R_j is characterized mathematically by two quantities. The first is its state-change vector $\mathbf{v}_j \equiv (v_{1j}, \ldots, v_{Nj})$, where v_{ij} is the change in the S_i molecular population caused by one R_j reaction, so if the system is in state $\mathbf{x}$ and one R_j reaction occurs, the system immediately jumps to state $\mathbf{x} + \mathbf{v}_j$. The other characterizing quantity for R_j is its propensity function a_j, which is defined so that

$$a_j(\mathbf{x})\,dt \stackrel{\Delta}{=} \text{the probability, given } \mathbf{X}(t) = \mathbf{x}, \text{ that one } R_j \text{ reaction will occur}$$

$$\text{somewhere inside } \Omega \text{ in the next infinitesimal time interval } [t, t + dt). \qquad (2)$$

Definition 2 can be regarded as the fundamental premise of stochastic chemical kinetics because everything else in the theory follows from it via the laws of probability. The physical rationale for Definition 2 for unimolecular and bimolecular reactions can be briefly summarized as follows.

If R_j is the unimolecular reaction $S_1 \rightarrow$ product(s), the underlying physics, which is usually quantum mechanical, dictates the existence of some constant c_j, such that $c_j dt$ gives the probability that any particular S_1 molecule will so react in the next infinitesimal time dt. It then follows from the laws of probability that if there are currently x_1 S_1 molecules in the system, the probability that some one of them will undergo the R_j reaction in the next dt is $x_1 \cdot c_j dt$. Thus the propensity function in Equation 2 is $a_j(\mathbf{x}) = c_j x_1$.

If R_j is a bimolecular reaction of the form $S_1 + S_2 \rightarrow$ product(s), kinetic theory arguments and the well-stirred condition together imply the existence of a constant c_j,

State-change vector: the change in the vector of the species' molecular populations induced by a single occurrence of a particular reaction

Propensity function: the function whose product with dt gives the probability that a particular reaction will occur in the next infinitesimal time dt

such that $c_j dt$ gives the probability that a randomly chosen pair of S_1 and S_2 molecules will react according to R_j in the next infinitesimal time dt (8–11). The probability that some one of the $x_1 x_2$ S_1-S_2 pairs inside Ω will react according to R_j in the next dt is therefore $x_1 x_2 \cdot c_j dt$, and that implies that the propensity function in Equation 2 is $a_j(\mathbf{x}) = c_j x_1 x_2$. If instead the bimolecular reaction had been $S_1 + S_1 \rightarrow$ product(s), we would have reckoned the number of distinct S_1 molecular pairs as $\frac{1}{2} x_1 (x_1 - 1)$, and so obtained for the propensity function $a_j(\mathbf{x}) = c_j \frac{1}{2} x_1 (x_1 - 1)$.

Evaluating c_j completely from first principles is a challenging task, requiring specific assumptions to be made about how the reaction R_j physically occurs. Unimolecular c_j's and bimolecular c_j's are quite different from each other. For example, whereas a unimolecular c_j is independent of the system volume Ω, a bimolecular c_j is inversely proportional to Ω, reflecting the fact that two reactant molecules will have a harder time finding each other inside a larger volume. It turns out that for a unimolecular reaction, c_j is numerically equal to the reaction-rate constant k_j of conventional deterministic chemical kinetics, whereas for a bimolecular reaction, c_j is equal to k_j / Ω if the reactants are different species, or $2k_j / \Omega$ if they are the same species (8–11). However, these results should not be taken to imply that the mathematical forms of the propensity functions are just heuristic extrapolations of the reaction rates of deterministic chemical kinetics. The propensity functions are grounded in molecular physics, and the formulas of deterministic chemical kinetics are approximate consequences of the formulas of stochastic chemical kinetics, not the other way around.

Although the probabilistic nature of Equation 2 precludes making an exact prediction of $\mathbf{X}(t)$, we might hope to infer the probability

$$P(\mathbf{x}, t \mid \mathbf{x}_0, t_0) \overset{\Delta}{=} \text{Prob}\{\mathbf{X}(t) = \mathbf{x}, \text{ given } \mathbf{X}(t_0) = \mathbf{x}_0\}. \qquad (3)$$

It is not difficult to derive a time-evolution equation for $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ by applying the laws of probability to the fundamental premise (Equation 2). The result is the chemical master equation (CME) (10–12):

$$\frac{\partial P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)}{\partial t} = \sum_{j=1}^{M} [a_j(\mathbf{x} - \mathbf{v}_j) P(\mathbf{x} - \mathbf{v}_j, t \mid \mathbf{x}_0, t_0) - a_j(\mathbf{x}) P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)]. \qquad (4)$$

In principle, the CME completely determines the function $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$. But a close inspection reveals that the CME is actually a set of coupled ODEs, with one equation for every possible combination of reactant molecules. It is therefore not surprising that the CME can be solved analytically for only a few simple cases, and even numerical solutions are prohibitively difficult in other cases.

It is also difficult to infer anything about the behavior of averages such as $\langle b(\mathbf{X}(t)) \rangle \equiv \sum_{\mathbf{x}} b(\mathbf{x}) P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ if any of the reaction channels are bimolecular. For example, if we multiply the CME (Equation 4) through by $\mathbf{x}$ and then sum over all $\mathbf{x}$, we get

$$\frac{d \langle \mathbf{X}(t) \rangle}{dt} = \sum_{j=1}^{M} \mathbf{v}_j \langle a_j(\mathbf{X}(t)) \rangle. \qquad (5)$$

If all the reactions were unimolecular, the propensity functions would all be linear in the state variables, and we would have $\langle a_j(\mathbf{X}(t)) \rangle = a_j(\langle \mathbf{X}(t) \rangle)$, so Equation 5 would

reduce to a closed ODE for the first moment $\langle \mathbf{X}(t) \rangle$. But if any reaction is bimolecular, the right-hand side of Equation 5 will contain at least one quadratic moment of the form $\langle X_i(t)X_{i'}(t) \rangle$, and Equation 5 would then be merely the first of an infinite, open-ended set of equations for all the moments.

In the hypothetical case in which there are no fluctuations—i.e., if $\mathbf{X}(t)$ were a deterministic or sure process—we would have $\langle h(\mathbf{X}(t)) \rangle = h(\mathbf{X}(t))$ for all functions h, and Equation 5 would then reduce to

$$\frac{d\mathbf{X}(t)}{dt} = \sum_{j=1}^{M} \mathbf{v}_j a_j(\mathbf{X}(\mathbf{t})). \tag{6}$$

This is precisely the RRE (Equation 1), with the functions $f_i(\mathbf{X}) \equiv \sum_j v_{ij} a_j(\mathbf{X})$ now explicitly rendered. As a set of coupled ODEs, Equation 6 characterizes $\mathbf{X}(t)$ as a continuous, deterministic process. But while this shows that the RRE would be valid if all fluctuations could be ignored, it does not provide any justification for doing that. Below I discuss how, and under what conditions, the discrete, stochastic CME description approximately gives rise to the continuous, deterministic RRE description.

Because the CME (Equation 4) can rarely be solved for the probability density function of $\mathbf{X}(t)$, perhaps we should look for a way to construct numerical realizations of $\mathbf{X}(t)$, i.e., simulated trajectories of $\mathbf{X}(t)$ versus t. This is not the same as solving the CME numerically, as that would give us the probability density function of $\mathbf{X}(t)$ instead of a random sample of $\mathbf{X}(t)$. The key to generating simulated trajectories of $\mathbf{X}(t)$ is not the function $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$, but rather a new probability function $p(\tau, j \mid \mathbf{x}, t)$, which is defined as follows:

$$p(\tau, j \mid \mathbf{x}, t)\,d\tau \;\overset{\Delta}{=}\; \text{the probability, given } \mathbf{X}(t) = \mathbf{x}, \text{ that the next reaction in the}$$
$$\text{system will occur in the infinitesimal time interval}$$
$$[t+\tau, t+\tau+d\tau), \text{ and will be an } R_j \text{ reaction.} \tag{7}$$

Formally, this function is the joint probability density function of the two random variables time to the next reaction (τ) and index of the next reaction (j), given that the system is currently in state $\mathbf{x}$. It is not difficult to derive an exact formula for $p(\tau, j \mid \mathbf{x}, t)$ by applying the laws of probability to the fundamental premise (Equation 2). The result is (8–11)

$$p(\tau, j \mid \mathbf{x}, t) = a_j(\mathbf{x})\ \exp(-a_0(\mathbf{x})\,\tau), \tag{8}$$

where

$$a_0(\mathbf{x}) \overset{\Delta}{=} \sum_{j'=1}^{M} a_{j'}(\mathbf{x}). \tag{9}$$

Equation 8 is the mathematical basis for the stochastic simulation approach. It implies that τ is an exponential random variable with mean (and standard deviation) $1/a_0(\mathbf{x})$, while j is a statistically independent integer random variable with point probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$. There are several exact Monte Carlo procedures for generating samples of τ and j according to these distributions. Perhaps the simplest is the so-called direct method, which follows by applying the standard inversion generating method of Monte Carlo theory (11): We draw two random numbers r_1 and r_2 from the uniform

distribution in the unit interval, and take

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right), \qquad (10a)$$

$$j = \text{the smallest integer satisfying } \sum_{j'=1}^{j} a_{j'(\mathbf{x})} > r_2 \, a_0(\mathbf{x}). \qquad (10b)$$

Stochastic simulation algorithm (SSA): a Monte Carlo procedure for numerically generating time trajectories of the molecular populations in exact accordance with the CME

With this generating method (or any mathematically equivalent one), we have the following stochastic simulation algorithm (SSA) for constructing an exact numerical realization of the process $\mathbf{X}(t)$ (8, 9):

  0. Initialize the time $t = t_0$ and the system's state $\mathbf{x} = \mathbf{x}_0$.
  1. With the system in state $\mathbf{x}$ at time t, evaluate all the $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x})$.
  2. Generate values for τ and j using Equations 10a,b (or their equivalent).
  3. Effect the next reaction by replacing $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}_j$.
  4. Record $(\mathbf{x}, t)$ as desired. Return to Step 1, or else end the simulation.

The $\mathbf{X}(t)$ trajectory produced by the SSA may be thought of as a stochastic version of the trajectory that would be obtained by solving the RRE (Equation 6). But note that the time step τ in the SSA is exact and not a finite approximation to some infinitesimal dt, as is the time step in a typical ODE solver. If it is found that every SSA-generated trajectory is practically indistinguishable from the RRE trajectory, then we may conclude that microscale randomness is ignorable. But if the SSA trajectories are found to deviate significantly from the RRE trajectory, or from each other, then we must conclude that microscale randomness is not ignorable, and the deterministic RRE does not provide an accurate description of the system's true behavior.

Because the SSA and the CME are each derived without approximation from the fundamental premise (Equation 2), they are logically equivalent to each other. But even when the CME is intractable, the SSA is easy to implement; indeed, as a numerical procedure, the SSA is simpler than most procedures that are used to numerically solve the RRE (Equation 6). The catch is that the SSA is often very slow, essentially because it insists on simulating every individual reaction event. The mathematical reason for this slowness can be traced to the factor $1/a_0(\mathbf{x})$ in Equation 10a, which will be small if any reactant population is large.

To illustrate the foregoing ideas, consider the simple reaction

$$S \xrightarrow{c} \varnothing. \qquad (11)$$

The propensity function for this reaction is $a(x) = c\,x$, and the state-change vector is $v = -1$. The RRE (Equation 6) reads $dX/dt = -c\,X$, and the solution to this ODE for the initial condition $X(0) = x_0$ is

$$X(t) = x_0 e^{-ct} \text{(RREsolution)}. \qquad (12)$$

The CME (Equation 4) reads

$$\frac{\partial}{\partial t} P(x, t \mid x_0, 0) = a(x + 1) P(x + 1, t \mid x_0, 0) - a(x) P(x, t \mid x_0, 0).$$
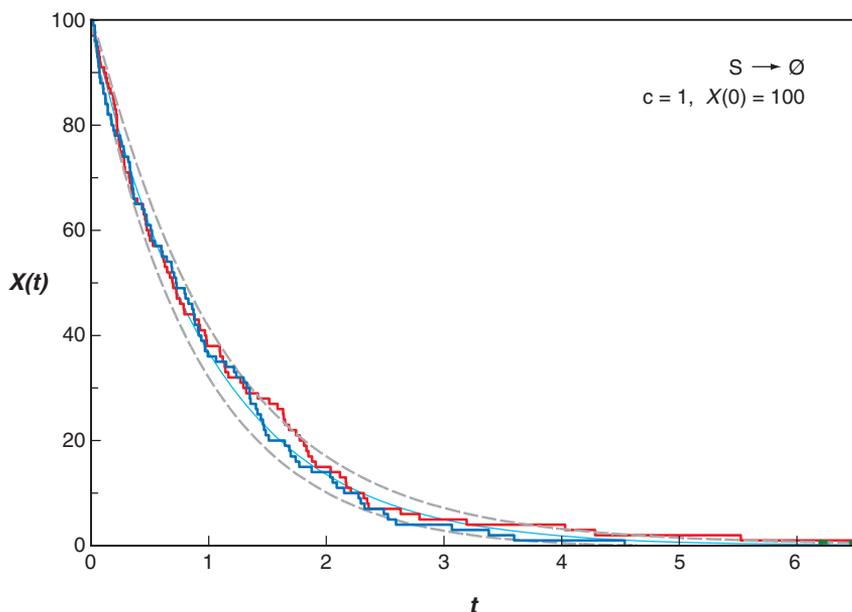
Figure 1

Simulating the simple isomerization reaction (Equation 11). The thin light blue line shows the solution (Equation 12) of the reaction-rate equation (RRE). The two dashed gray lines show the one-standard-deviation envelope $\langle X(t) \rangle \pm \text{sdev}\{X(t)\}$ of Equations 14a,b as predicted by the solution of the chemical master equation (CME) (Equation 13). The red and blue jagged curves show the trajectories produced by two separate runs of the stochastic simulation algorithm (SSA).

Because $P(x_0 + 1, t \mid x_0, 0) \equiv 0$, we can solve this equation exactly by successively putting $x = x_0, x_0 - 1, \ldots, 0$. The result is

$$P(x, t \mid x_0, 0) = \frac{x_0!}{x!(x_0 - x)!} e^{-cxt} (1 - e^{-ct})^{x_0 - x} \quad (x = 0, \ldots, x_0), \qquad (13)$$

which we recognize as the probability density function for the binomial random variable with mean and standard deviation

$$\langle X(t) \rangle = x_0 e^{-ct}, \qquad (14a)$$

$$\text{sdev}\{X(t)\} = \sqrt{x_0 e^{-ct}(1 - e^{-ct})}. \qquad (14b)$$

Note that $\langle X(t) \rangle$ is identical to the solution (Equation 12) of the RRE; this will always be so if all the propensity functions are linear in the populations, but not otherwise. The SSA for this reaction is simple: In state x at time t, we draw a unit-interval uniform random number r, increase t by $\tau = (1/ax) \ln(1/r)$, decrease x by 1, and then repeat. **Figure 1** shows numerical results for $c = 1$ and $x_0 = 100$.

3. ELABORATIONS ON AND IMPROVEMENTS TO THE STOCHASTIC SIMULATION ALGORITHM

The version of the SSA described above is the one originally presented in References 8 and 9. A number of earlier works applied similar if not equivalent procedures to specific model systems (13–19) but paid little attention to developing the supporting theory. In this section I focus on some later elaborations on and improvements to the SSA.

One elaboration of the SSA already noted in Reference 8 is an alternative to the direct method (Equation 10) for generating values of τ and j. Called the first-reaction method, it begins by drawing M random numbers $r_1, \ldots, r_M$ from the unit-interval uniform distribution and computing

$$\tau_{j'} = \frac{1}{a_{j'}(\mathbf{x})} \ln\left(\frac{1}{r_{j'}}\right) \quad (j' = 1, \ldots, M); \tag{15a}$$

then it takes

$$\left.\begin{array}{l} \tau = \text{the smallest of the } \{\tau_{j'}\} \\ j = \text{the index of the smallest } \{\tau_{j'}\} \end{array}\right\}. \tag{15b}$$

Heuristically, $\tau_1, \ldots, \tau_M$ are putative times to the next firings of the respective reaction channels; however, we accept only the earliest of those and discard the rest. It can be proved (8) that this procedure, like the direct method, generates values for τ and j in exact accord with the joint density function (Equation 8). However, if the system has many reaction channels, this method will be computationally less efficient than the direct method.

A generalization of the direct and first-reaction methods that includes both as special cases is L. Lok's (unpublished manuscript) first-family method. The M reactions are partitioned into L families $\{F_1, \ldots, F_L\}$ and then relabeled so that the M_l reactions in family F_l are $\{R_1^l, \ldots, R_{M_l}^l\}$. Each family F_l is then viewed as a pseudoreaction with propensity function $a_0^l(\mathbf{x}) \equiv \sum_{j=1}^{M_l} a_j^l(\mathbf{x})$. To generate the time τ to the next reaction event and the index pair (l, j) that identifies that reaction, we draw $L + 1$ random numbers $r_1, \ldots, r_{L+1}$ from the uniform distribution in the unit interval. We use the first L of these to compute

$$\tau_{l'} = \frac{1}{a_0^{l'}(\mathbf{x})} \ln\left(\frac{1}{r_{l'}}\right) \quad (l' = 1, \ldots, L); \tag{16a}$$

then we take

$$\left.\begin{array}{l} \tau = \text{the smallest of the } \{\tau_{l'}\} \\ l = \text{the index of the smallest } \{\tau_{l'}\} \end{array}\right\}, \tag{16b}$$

and finally

$$j = \text{the smallest integer satisfying } \sum_{j'=1}^{j} a_{j'}^l(\mathbf{x}) > r_{L+1}\, a_0^l(\mathbf{x}). \tag{16c}$$

Heuristically, Equations 16a,b generate the time step τ to, and the index l of, the next firing family, and Equation 16c then decides which reaction in F_l actually fires. It can be proved (L. Lok, unpublished manuscript) that this procedure generates values for τ and $j = (l, j)$ in exact accord with the joint density function (Equation 8). It reduces to the direct method if all the reactions are taken to be members of one family, and it reduces to the first-reaction method if each reaction is taken to be a family unto itself. For intermediate partitionings, the method may afford bookkeeping advantages when M is large.

A reformulation of the SSA that, for large N and M, significantly increases its speed as compared with the direct method is Gibson & Bruck's (20) next-reaction method.

Essentially a heavily revised version of the first-reaction method, the next-reaction method saves the putative next firing times of all reaction channels in an indexed binary tree priority queue, which is constructed so that the firing time of each parent node is always earlier than the firing times of its two daughter nodes. The time and index of the next occurring reaction are therefore always available at the top node of the queue. The indexing scheme and the binary-tree structure of the queue facilitate updating the queue as a result of changes caused by occurring reactions. With added effort, we can even make the next-reaction method consume only one uniform random number per additional reaction event, in contrast to the two required by the direct method. Although the next-reaction method can be significantly faster than the direct method, it is much more challenging to code. For more details, see References 20 and 21.

Lok & Brent (22) have developed a stochastic simulation software package called Moleculizer that uses a slightly simplified version of the next-reaction method, but with a unique twist: Reaction channels and species are introduced only when they are needed, and removed when they are not needed. The target application for Moleculizer is the simulation of the pheromone-response mechanism in yeast, a chemical system that entails a potentially enormous number of species and reaction channels. Lok showed that, at least when using the SSA, it is not necessary to introduce all the reactions and species at the beginning of a simulation, and for the yeast system in particular, a just-in-time introduction of the reactions and species makes feasible an otherwise infeasible simulation. Lok also observes that this just-in-time strategy cannot be used with the RRE, which is noteworthy because the RRE too becomes unwieldy when enormous numbers of species and reaction channels are involved; thus we have yet another reason for using stochastic simulation instead of deterministic simulation on biological systems. For further discussion of these points, see Reference 22.

Cao et al. (21) recently introduced a modified direct method that often makes the direct method competitive in speed with the next-reaction method. These authors observed that if the reaction channels are indexed so that reactions R_j with larger propensity functions are assigned lower index values j, then the average number of terms summed in the computation (Equation 10b) is minimized. The consequent gain in speed can be significant for systems with many reactions and a wide range of propensity function values, a common circumstance in biological models. The modified direct method starts off with a relatively short prerun using the direct method in which the average sizes of the propensity functions are assessed. Then it reindexes the reactions accordingly and resumes the simulation, at a usually much-greater speed. See Reference 21 for details.

McCollum et al. (23) have recently proposed a further improvement on the direct method in what they call the sorting direct method. Similar to the modified direct method, the sorting direct method seeks to index the reaction channels in order of decreasing values of their propensity functions so as to optimize the search in Equation 10b. But the sorting method does this dynamically, and without the need for a prerun, by using a simple bubble-up tactic: Whenever a reaction channel fires, the index of the firing channel is interchanged with the index of the next lower indexed channel

(if there is one). Doing this repeatedly tends to establish the desired index ordering. This tactic not only eliminates the prerun of the modified direct method, but it also accommodates any changes in the relative sizes of the propensity functions that might develop as the simulation proceeds.

H. Li & L.R. Petzold (unpublished manuscript) have recently proposed the logarithmic direct method, another novel twist on the direct method. Its strategy is to collect and store the partial sums of the propensity functions during the computation of the full sum a_0 in Equation 9. The value of j in Equation 10b can then be found rapidly by conducting a binary search over those partial sums. Although the logarithmic direct method (24) may not always result in as great a speed gain as the sorting direct method (23) or the optimized direct method (21), it can avoid a potential accuracy problem that may afflict those other two methods. Arranging the reaction indices in order of decreasing size of the propensity functions does make the linear search Equation 10b go faster, but it also makes that search potentially less accurate; e.g., if we were carrying k decimals in the sum on the left side of Equation 10b and the highest-indexed propensity function happened to be k orders of magnitude smaller than a_M, then numerical truncation results in R_M never firing at all. For maximal numerical accuracy in executing Equation 10b, reactions with smaller propensity functions should be assigned lower index values—just the opposite of what the modified and sorting direct methods do. The logarithmic direct method is not susceptible to this problem because it does not depend on any ordering scheme for the reaction indices; indeed, with the logarithmic direct method, the ordering could deliberately be arranged to achieve maximum accuracy. It might also be possible to overcome the inaccuracy problem by using Lok's first-family method, grouping reactions with very small propensities together into a family F_1 and all the other reactions into a family F_2: Most of the time the selection in Equation 16b falls to family F_2, but on those rare occasions when it falls to F_1, the subsequent search in Equation 16c is accomplished without involving the larger-valued propensity functions, so no truncation errors arise.

Improvements to the SSA along the lines described above are certainly beneficial, but any procedure that simulates every reaction event one at a time, no matter how efficiently it does that, will simply be too slow for many practical applications. This prompts us to look for ways to sacrifice some of the exactness of the SSA in return for greater simulation speed. One way of doing that is to use an approximate simulation strategy called tau-leaping.

4. TAU-LEAPING: THE BRIDGE TO THE REACTION-RATE EQUATION

With the system in state $\mathbf{x}$ at time t, let us suppose there exists a $\tau > 0$ that satisfies the following leap condition: During $[t, t + \tau)$, no propensity function is likely to change its value by a significant amount. With $a_j(\mathbf{x})$ remaining essentially constant during $[t, t + \tau)$, it then follows from the fundamental premise (Equation 2) that the number of times reaction channel R_j fires in $[t, t + \tau)$ is a Poisson random variable with mean (and variance) $a_j(\mathbf{x})\,\tau$. Therefore, to the degree that the leap condition is

Tau-leaping: an approximate way of accelerating the SSA in which each time step τ advances the system through possibly many reaction events

Leap condition: an accuracy-assuring restriction on tau-leaping that requires τ to be small enough that no propensity function changes by a significant amount

satisfied, we can approximately leap the system ahead by a time τ by taking (25, 26)

$$\mathbf{X}(t + \tau) \doteq \mathbf{x} + \sum_{j=1}^{M} \mathcal{P}_j(a_j(\mathbf{x})\tau)\mathbf{v}_j, \qquad (17)$$

where $\mathbf{x} = \mathbf{X}(t)$, and $\mathcal{P}_j(m_j)$ is a statistically independent Poisson random variable with mean (and variance) m_j.

Equation 17 is the basic tau-leaping formula. In the next section below I discuss how we can use it to perform faster stochastic simulations. But for now, let us suppose that τ is not only small enough to satisfy the leap condition, but also large enough that the expected number of firings of each reaction channel R_j during τ is $\gg 1$:

$$a_j(\mathbf{x})\,\tau \gg 1 \text{ for all } j = 1, \ldots, M. \qquad (18)$$

Then, denoting the normal (Gaussian) random variable with mean m and variance σ^2 by $\mathcal{N}(m, \sigma^2)$, and invoking the mathematical fact that a Poisson random variable with a mean and variance that is $\gg 1$ can be approximated as a normal random variable with that same mean and variance, we can further approximate Equation 17 as

$$\mathbf{X}(t + \tau) \doteq \mathbf{x} + \sum_{j=1}^{M} \mathcal{N}_j(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau)\,\mathbf{v}_j = \mathbf{x} + \sum_{j=1}^{M} \left[a_j(\mathbf{x})\tau + \sqrt{a_j(\mathbf{x})\tau}\,\mathcal{N}_j(0, 1) \right]\mathbf{v}_j.$$

The last step here invokes the well-known property of the normal random variable that $\mathcal{N}(m, \sigma^2) = m + \sigma\mathcal{N}(0, 1)$. Collecting terms gives us what is known as the chemical Langevin equation (CLE) or Langevin leaping formula (25),

$$\mathbf{X}(t + \tau) \doteq \mathbf{x} + \sum_{j=1}^{M} \mathbf{v}_j a_j(\mathbf{x})\tau + \sum_{j=1}^{M} \mathbf{v}_j \sqrt{a_j(\mathbf{x})}\,\mathcal{N}_j(0, 1)\sqrt{\tau}, \qquad (19)$$

where $\mathbf{x} = \mathbf{X}(t)$, and each $\mathcal{N}_j(0, 1)$ is a statistically independent normal random variable with mean 0 and variance 1. Again, this equation is valid only to the extent that during τ, no propensity function changes its value significantly, yet every reaction channel fires many more times than once. It is usually possible to find a τ that satisfies these opposing conditions if all the reactant populations are sufficiently large.

In the theory of continuous Markov processes, it can be shown that the CLE (Equation 19) can also be written in the white-noise form (25, 27)

$$\frac{d\mathbf{X}(t)}{dt} \doteq \sum_{j=1}^{M} \mathbf{v}_j\, a_j(\mathbf{X}(t)) + \sum_{j=1}^{M} \mathbf{v}_j \sqrt{a_j(\mathbf{X}(t))}\,\Gamma_j(t). \qquad (20)$$

Here the $\Gamma_j(t)$ are statistically independent Gaussian white-noise processes, satisfying $\langle \Gamma_j(t)\,\Gamma_{j'}(t') \rangle = \delta_{jj'}\,\delta(t - t')$, where the first delta function is Kronecker's and the second is Dirac's. Equation 20 is just another way of writing Equation 19; the two equations are mathematically equivalent. Equations of the form of Equation 20, with the right side the sum of a deterministic drift term and a stochastic diffusion term proportional to Gaussian white noise, are known as Langevin equations or stochastic differential equations. In most occurrences of Langevin equations in science and engineering applications, the form of the stochastic diffusion term is postulated ad hoc; here, however, it has been derived. Continuous Markov process theory also implies that

the probability density function $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ of the random variable $\mathbf{X}(t)$ in the CLE Equations 19 and 20 obeys a well-defined partial differential equation called the chemical Fokker-Planck equation (CFPE). References 25, 27, and 28 give a derivation and discussion of the CFPE.

The thermodynamic limit is defined as the limit in which the species populations X_i and the system volume Ω all approach infinity, but in such a way that the species concentrations X_i/Ω stay constant. As this limit is approached, all propensity functions grow in direct proportion to the size of the system. This is obvious for a unimolecular propensity function of the form $c_j x_i$; for a bimolecular propensity function of the form $c_j x_i x_{i'}$, this follows because c_j is inversely proportional to Ω. Therefore, as the thermodynamic limit is approached, the term on the left side of the CLE (Equation 20) and the first term on the right side both grow like the system size, whereas the second term on the right grows more slowly as the square root of the system size. In the full thermodynamic limit, the last term becomes negligibly small compared with the other terms, and the CLE (Equation 20) reduces to the RRE (Equation 6). Thus we have derived the RRE from the fundamental stochastic premise (Equation 2). The approximations made in this derivation are schematized in **Figure 2**, which summarizes the theoretical structure of stochastic chemical kinetics (29).

5. THE EXPLICIT TAU-LEAPING SIMULATION ALGORITHM

The basic tau-leaping formula (Equation 17) suggests an obvious strategy for approximately doing stochastic simulations: In the current state $\mathbf{x}$, we first choose a value for τ that satisfies the leap condition. Next, we generate for each j a sample k_j of the Poisson random variable with mean $a_j(\mathbf{x})\,\tau$, by using, for example, the numerical procedure described in Reference 30. (Because the Poisson random numbers $k_1, \ldots, k_M$ are statistically independent, they could be generated simultaneously on M parallel processors, which would result in a substantial gain in computational speed.) Finally, we update the state from $\mathbf{x}$ to $\mathbf{x} + \sum_j k_j \mathbf{v}_j$. If the values generated for the k_j are sufficiently large, this approximate procedure will be faster than the exact SSA. But several practical issues need to be resolved to effectively implement this strategy: First, how can we estimate in advance the largest value of τ that satisfies the leap condition? Second, how can we ensure that the generated k_j-values do not cause some R_j to fire so many times that the population of some reactant is driven negative? Finally, how can we arrange it so that tau-leaping segues efficiently to the SSA?

The method originally suggested in Reference 26 for estimating the largest value of τ that satisfies the leap condition has undergone two successive refinements (31, 32). The latest τ-selection procedure (32) is not only more accurate than the earlier procedures, but also faster, especially if M is large. It computes the largest value of τ for which the estimated fractional change $\Delta_\tau a_j/a_j$ in each propensity function during τ is bounded by a user-specified accuracy-control parameter ε ($0 < \varepsilon \ll 1$). However, it does this in an indirect way: It chooses τ so that the estimated fractional change $\Delta_\tau x_i/x_i$ in each reactant population is bounded by an amount $\varepsilon_i = \varepsilon_i(\varepsilon, x_i)$ (except that no x_i is required to change by an amount less than one), where the functions
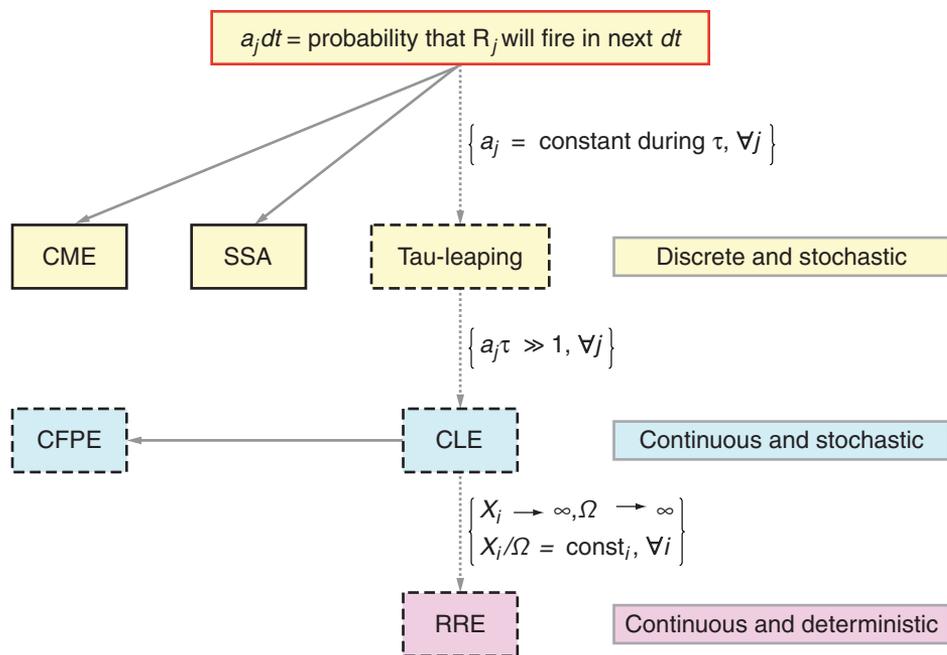
Thermodynamic limit:
the infinite-population, infinite-volume, finite-concentration limit in which the stochastic CLE reduces (usually) to the deterministic RRE

$a_j dt$ = probability that R_j will fire in next dt

$\left\{ a_j = \text{constant during } \tau, \forall j \right\}$

| CME | SSA | Tau-leaping | Discrete and stochastic |

$\left\{ a_j \tau \gg 1, \forall j \right\}$

| CFPE | CLE | Continuous and stochastic |

$\left[\begin{array}{l} X_i \longrightarrow \infty, \Omega \longrightarrow \infty \\ X_i/\Omega = \text{const}_i, \forall i \end{array} \right]$

| RRE | Continuous and deterministic |

Figure 2

Logical structure of stochastic chemical kinetics. Everything follows from the fundamental premise at the top via the laws of probability theory. Inference routes that are exact are shown by solid arrows. Inference routes that are approximate are shown by dotted arrows, with the condition justifying the approximation indicated in braces immediately to the right. Solid-outlined boxes are exact results: the chemical master equation (CME) and the stochastic simulation algorithm (SSA). Dashed-outlined boxes are approximate results: the tau-leaping formula, the chemical Langevin equation (CLE), the chemical Fokker-Planck equation (CFPE), and the reaction-rate equation (RRE). The condition justifying the arguments leading from the fundamental premise to tau-leaping is called the leap condition, and the condition justifying the arguments leading from the CLE to the RRE is called the thermodynamic limit.

ε_i have been chosen so that $\Delta_\tau a_j/a_j$ for every j is then bounded by the stipulated amount ε. As is shown in Reference 32, the algebraic forms of the functions $\varepsilon_i(\varepsilon, x_i)$ that accomplish this are quite simple, and they can easily be inferred by inspection at the outset of the simulation. Enforcing the bound

$$|\Delta_\tau x_i| \leq \max\{\varepsilon_i x_i, 1\} \tag{21}$$

is accomplished by first noting from the basic tau-leaping formula (Equation 17) that

$$\Delta_\tau x_i \doteq \sum_j \mathcal{P}_j(a_j \tau) v_{ij}.$$

Because the statistically independent Poisson random variables $\mathcal{P}_j(a_j \tau)$ have means and variances $a_j \tau$, the means and variances of $\Delta_\tau x_i$ are

$$\langle \Delta_\tau x_i \rangle \doteq \sum_j v_{ij}(a_j \tau), \quad \text{var}\{\Delta_\tau x_i\} \doteq \sum_j v_{ij}^2(a_j \tau). \tag{22}$$

Condition 21 is deemed to be adequately fulfilled if it is satisfied by both $\langle \Delta_\tau x_i \rangle$ and $\sqrt{\text{var}\{\Delta_\tau x_i\}}$. The resulting set of inequalities yields an efficient, explicit formula for the largest permissible value of τ (32).

As for keeping the generated random numbers k_j from driving the R_j reactant populations negative, several strategies have been proposed. Tian & Burrage (33)—and, independently, Chatterjee et al. (34)—proposed approximating the unbounded Poisson random numbers k_j with bounded binomial random numbers. But it turns out that it is usually not the unboundedness of the Poisson k_j's that produces negative populations, but rather the lack of coordination in tau-leaping between different reaction channels that separately decrease the population of a common species. Cao et al. (35) have proposed a different approach that resolves this difficulty and also establishes a smooth connection with the SSA. In their approach, we first identify as critical all those reactions with nonzero propensities that are currently within n_c firings of exhausting one of its reactants, n_c being a user-specified integer. All other reactions are called noncritical. The noncritical reactions are handled by the regular Poisson tau-leaping method, and a maximum leap time τ' is computed for them using the procedure described in the preceding paragraph. For the critical reactions, we use the direct method formulas Equations 10a,b to estimate the time τ'' to, and index j_c of, the next critical reaction. The actual time step τ is then taken to be the smaller of τ' and τ''; if the former, no critical reaction fires, and if the latter, only one critical reaction (R_{j_c}) fires. Because the total number of critical reactions firing during τ is never greater than one, it is impossible for any critical reaction to drive any population negative.

If n_c is taken so large that every reaction becomes critical, the foregoing procedure reduces to the exact SSA. This is fortunate because although tau-leaping theoretically becomes exact (and hence equivalent to the SSA) as $\tau \to 0$, it also becomes grossly inefficient in that limit. This is because when τ is small, the Poisson random numbers $k_j = \mathcal{P}_j(a_j \tau)$ usually all are zero, resulting in a small tau-leap with no reactions firing. It is not efficient to do tau-leaping when τ is less than a few multiples of $1/a_0(\mathbf{x})$, the expected next time step in the SSA. But by using a reasonable value for n_c (e.g., between 5 and 50), along with a reasonable value of ε (e.g., between 0.01 and 0.06), large leaps are taken whenever possible, and a gradual transition to the SSA occurs automatically as needed for accuracy. Finally, if we write the code for generating a Poisson random number with mean (and variance) m so that when $m \gg 1$, it generates instead a normal random number with mean and variance m, then a smooth transition from tau-leaping to the more computationally efficient Langevin leaping will occur automatically.

Reference 32 gives a more detailed description of the current explicit tau-leaping procedure. Tests indicate that for many systems in which the molecular populations of at least some of the reactant species are large, it produces significantly faster simulations than the SSA with only a slight loss of accuracy.

6. SIMULATING STIFF SYSTEMS

A system of ODEs is said to be stiff if is characterized by well-separated fast and slow dynamical modes, the fastest of which is stable. The solution space of a stiff ODE has a slow manifold, on which the state point moves slowly, and off which the state point

moves rapidly toward the slow manifold. Researchers have devoted much effort over the years to understanding and overcoming the computational problems posed by stiff ODEs (36) because such equations arise in many practical contexts. Stiff RREs in particular are quite commonplace.

Stiffness is just as computationally troublesome in the stochastic context. When the SSA simulates a stiff system, it moves along as usual, one reaction at a time, oblivious to the stiffness. But because the great majority of the reactions are the usually uninteresting fast ones, the simulation proceeds slowly from a practical point of view. The explicit tau-leaping algorithm also performs as advertised on stiff systems. But because the τ-selection procedure that keeps the algorithm accurate restricts τ to the timescale of the system's fastest mode, then even those leaps seem frustratingly small. I conclude by describing two recently developed strategies for simulating stiff chemical systems: the implicit tau-leaping algorithm and the slow-scale SSA (ssSSA).

Slow-scale SSA (ssSSA): an approximation to the SSA for systems with fast and slow reactions in which only the latter are explicitly simulated

6.1. The Implicit Tau-Leaping Algorithm

A well-known strategy for numerically solving a stiff ODE $dx/dt = f(x)$ is to replace the explicit updating formula $x_{t+\Delta t} = x_t + f(x_t)\Delta t$ with an implicit formula, such as $x_{t+\Delta t} = x_t + f(x_{t+\Delta t})\Delta t$ (36). The latter equation of course has to be solved to obtain $x_{t+\Delta t}$ at each time step. But even when that can only be done numerically (Newton iteration is usually used), the extra effort is usually more than compensated by the ability to use much-larger values of Δt for the same degree of accuracy.

The tau-leaping formula (Equation 17), where $\mathbf{x} = \mathbf{X}(t)$, is obviously an explicit updating formula. To make it implicit by replacing $\mathbf{x}$ in the argument of the Poisson random variable with $\mathbf{X}(t+\tau)$, however, raises some serious questions in the context of Markov process theory, where updates are supposed to be past-forgetting; moreover, even if that replacement could be justified theoretically, there appears to be no way to solve the resulting equation for $\mathbf{X}(t + \tau)$. Rathinam et al. (37) have proposed a partial implicitization, in the following implicit tau-leaping formula:

$$\mathbf{X}(t + \tau) \doteq \mathbf{x} + \sum_{j=1}^{M} [\mathcal{P}_j(a_j(\mathbf{x})\tau) - a_j(\mathbf{x})\tau + a_j(\mathbf{X}(t + \tau))\tau]\mathbf{v}_j. \qquad (23)$$

In this formula, the mean of the Poisson random variable $\mathcal{P}_j(a_j(\mathbf{x})\tau)$ is subtracted out and replaced by its value at the later time $t + \tau$, but the variance has been left unchanged. The advantage of this formula is that, once the Poisson random numbers have been generated using the current state $\mathbf{x}$, the equation can then be solved for $\mathbf{X}(t + \tau)$ using the same (deterministic) numerical techniques developed for implicit ODE solvers (36). Noninteger values for the components of $\mathbf{X}(t+\tau)$ can be avoided by rounding the quantity in brackets in Equation 23 to the nearest nonnegative integer and then recomputing $\mathbf{X}(t + \tau)$ directly from Equation 23 (37).

Tests of this implicit tau-leaping strategy show that it produces significantly faster simulations than the explicit tau-leaping formula (Equation 17) for stiff systems, but with one major qualification: Formula 23 excessively damps the fluctuations in the fast components of $\mathbf{X}(t)$. Rapid fluctuations of the state point transverse to the slow manifold naturally occur in a stochastically evolving system, as can be seen in

simulations made with the exact SSA. But Equation 23 suppresses these fluctuations. However, we can restore the properly fluctuating fast variables whenever desired by taking a succession of much shorter explicit tau-leaps or SSA steps, a tactic called downshifting. For more details, see Reference 37.

Cao & Petzold (38) subsequently proposed a trapezoidal implicit tau-leaping formula, which has the same form as Equation 23 except that a factor of $1/2$ appears in front of each of the last two terms in the brackets. Tests suggest that the trapezoidal formula often gives more accurate results than Equation 23 for a given value of τ.

6.2. The Slow-Scale Stochastic Simulation Algorithm

A different approach to stochastically simulating stiff chemical systems is one that was inspired by the well-known Michaelis-Menten approximation in deterministic chemical kinetics (39). Several different ways of realizing this approach have been proposed (40–48), but all are rooted in the same basic idea of eliminating the fast entities through a kind of quasi-steady-state approximation (41). Arguably the clearest articulation of this approach, at least as regards its theoretical justification within stochastic chemical kinetics, is the ssSSA of References 42 and 45.

The ssSSA proceeds in a series of steps, the first of which is to make a provisional partitioning of the reaction channels $R = \{R_1, \ldots, R_M\}$ into fast and slow subsets, R^f and R^s. Assigned to R^f are those reactions whose propensity functions tend to have the largest values. All the other reactions are assigned to R^s. If it is not obvious how to make this partitioning, then it may be that the system is not really stiff, and therefore not a candidate for the ssSSA. In any case, this provisional partitioning of the reactions will later be subjected to an acceptance test.

The second step is to partition the species $S = \{S_1, \ldots, S_N\}$ into fast and slow subsets, S^f and S^f, according to the following rule: Any species whose population gets changed by a fast reaction is classified as a fast species; all other species (if there are any) are classified as slow. This rule induces a partitioning of the process $\mathbf{X}(t)$ into a fast process $\mathbf{X}^f(t)$ and a slow process $\mathbf{X}^s(t)$. Note the subtle but important asymmetry that a fast species can get changed by a slow reaction, but a slow species cannot get changed by a fast reaction.

The third step defines the virtual fast process $\hat{\mathbf{X}}^f(t)$ as the fast species populations evolving under only the fast reactions R^f; i.e., $\hat{\mathbf{X}}^f(t)$ is $\mathbf{X}^f(t)$ with all the slow reactions switched off. The virtual fast process $\hat{\mathbf{X}}^f(t)$ is a Markov process, whereas the real fast process $\mathbf{X}^f(t)$ is generally non-Markovian, and hence practically intractable.

Next we require that two stochastic stiffness conditions be satisfied: First, $\hat{\mathbf{X}}^f(t)$ must be stable, in that it approaches as $t \to \infty$ a well-defined time-independent random variable $\hat{\mathbf{X}}^f(\infty)$. This is the counterpart to the deterministic stiffness requirement that the fastest dynamical mode be stable. Second, the approach $\hat{\mathbf{X}}^f(t) \to \hat{\mathbf{X}}^f(\infty)$ must be effectively accomplished in a time that is small compared with the expected time to the next slow reaction. This is a more precise specification of the degree of separation that must exist between the timescales of the fast and slow reactions. If we find these two stiffness conditions are satisfied, then our original classification of the fast

reactions is deemed acceptable; otherwise we must try another set of fast reactions, or else we must conclude that the system is not really stiff, and the ssSSA cannot be applied.

With the stochastic stiffness conditions satisfied, we now invoke the slow-scale approximation—a result that can be mathematically derived from the fundamental premise (Equation 2) (42). The slow-scale approximation states, in essence, that we can ignore the fast reactions and simulate the system one slow reaction at a time, provided we replace the propensity function of each slow reaction by its average with respect to the asymptotic virtual fast process $\hat{\mathbf{X}}^f(\infty)$. More precisely, if $\hat{P}(\mathbf{y}^f, \infty \mid \mathbf{x}^f, \mathbf{x}^s)$ is the probability that $\hat{\mathbf{X}}^f(\infty) = \mathbf{y}^f$ given that $\mathbf{X}(t) = (\mathbf{x}^f, \mathbf{x}^s)$, then the propensity function $a_j^s(\mathbf{x}^f, \mathbf{x}^s)$ of each slow reaction R_j^s at time t can be approximated on the timescale of the slow reactions by

$$\bar{a}_j^s(\mathbf{x}^f, \mathbf{x}^s) \equiv \sum_{\mathbf{y}^f} \hat{P}(\mathbf{y}^f, \infty \mid \mathbf{x}^f, \mathbf{x}^s)\, a_j^s(\mathbf{y}^f, \mathbf{x}^s). \tag{24}$$

The ssSSA thus proceeds by simulating, in the manner of the SSA, the slow reactions using the propensity functions (Equation 24) and ignoring the fast reactions. We can exhibit the populations of the fast species whenever desired by Monte Carlo sampling the probability function $\hat{P}$.

The approaches of References 43, 44, 47 and 48 differ from the ssSSA approach described above in the way the averages (Equation 24) are computed and the way in which the fast-species populations are generated. All rely on making relatively short SSA runs of the virtual fast process between the slow reactions.

Although the ssSSA can be challenging to implement, it has been successfully applied to a number of simple stiff systems (42), as well as the prototypical Michaelis-Menten system (45) that is so ubiquitous in enzymatic reactions. These applications showed increases in simulation speed over the exact SSA of two to three orders of magnitude with no perceptible loss of simulation accuracy.

7. OUTLOOK

The robustness and efficiency of both the SSA and the explicit tau-leaping algorithm have been considerably improved in the past few years, and those procedures seem to be nearing maturity. However, there is still room for improvement on the stiffness problem. Such improvement may come in the form of refinements to the implicit tau-leaping procedure and the ssSSA, and a clarification of the theoretical connection between those two approaches for dealing with stiff systems. Also needed are robust, adaptive strategies for deciding during a simulation when to use which simulation method. There is also a need for a better understanding of several foundational issues, such as how the reaction constants c_j are to be derived in the context of diffusional kinetics, and what are the effects of the molecular-crowding conditions usually present in living cells. Finally, the problem of how best to simulate systems that are not well stirred, a problem that is not addressed in this review, holds a great many challenges.

SUMMARY POINTS

1. The SSA is a procedure for numerically simulating well-stirred chemically reacting systems by stepping in time to successive molecular reaction events in exact accord with the premises of the CME.

2. The ability of the SSA to take proper account of the discrete, stochastic nature of chemical reactions makes it better suited to cellular chemical kinetics than the traditional RRE because in cellular systems the small numbers of molecules of some key reactants can amplify the effects of discreteness and randomness.

3. Because the SSA simulates every successive molecular reaction event that occurs in the system, it is often too slow for practical simulation of realistic cellular systems.

4. An approximate speedup to the SSA is provided by tau-leaping, in which time is advanced by a preselected amount τ and the numbers of firings of the individual reaction channels are approximated by Poisson random numbers.

5. If the expected number of firings of each reaction channel during a tau-leap is much greater than one, the Poisson random numbers are well approximated by normal random numbers, and the result is equivalent to a Langevin-type equation called the CLE.

6. In the thermodynamic (macroscopic) limit, the noise terms in the CLE become negligibly small and the CLE reduces to the conventional RRE, thereby establishing deterministic chemical kinetics in the context of stochastic chemical kinetics.

7. For stiff systems—which evolve on both fast and slow timescales with the fastest modes being stable—accuracy in tau-leaping requires τ to be small on the fastest timescale, which makes even tau-leaping seem too slow.

8. Two acceleration procedures for stiff systems are implicit tau-leaping, which mirrors the implicit Euler method in ODE theory, and the ssSSA, in which the fast reactions are skipped over and only the slow reactions are directly simulated using specially modified propensity functions.

LITERATURE CITED

1. **McAdams HH, Arkin AP. 1997. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA* 94:814–19**

2. Arkin AP, Ross J, McAdams HH. 1998. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *E. coli* cells. *Genetics* 149:1633–48

3. Elowitz MB, Levine AJ, Siggia ED, Swain PS. 2002. Stochastic gene expression in a single cell. *Science* 297:1183–86

4. Blake WJ, Kaern M, Cantor CR, Collins JJ. 2003. Noise in eukaryotic gene expression. *Nature* 422:633–37

5. Raser JM, O'Shea EK. 2004. Control of stochasticity in eukaryotic gene expression. *Science* 304:1811–14

6. Weinberger LS, Burnett JC, Toettcher JE, Arkin AP, Schaffer DV. 2005. Stochastic gene expression in a lentiviral positive-feedback loop: HIV-1 Tat fluctuations drive phenotypic diversity. *Cell* 122:169–82

7. **Samoilov M, Plyasunov S, Arkin AP. 2005. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proc. Natl. Acad. Sci. USA* 102:2310–15**

8. **Gillespie DT. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 22:403–34**

9. Gillespie DT. 1977. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* 81:2340–61

10. **Gillespie DT. 1992. A rigorous derivation of the chemical master equation. *Physica A* 188:404–25**

11. Gillespie DT. 1992. *Markov Processes: An Introduction for Physical Scientists*. San Diego: Academic

12. **McQuarrie D. 1967. Stochastic approach to chemical kinetics. *J. Appl. Probab.* 4:413–78**

13. Bharucha-Reid AT. 1960. *Elements of the Theory of Markov Processes and Their Applications*. New York: McGraw-Hill

14. Nakanishi T. 1972. Stochastic analysis of an oscillating chemical reaction. *J. Phys. Soc. Japan* 32:1313–22

15. Solc M, Horsak I. 1972. Simulation of first-order chemical reaction as a stochastic process on a digital computer. *Collect. Czech. Chem. Commun.* 37:2994–3000

16. Horsak I, Solc M. 1973. Simulation study of a stochastic model of reversible first-order reaction equilibrium. *Collect. Czech. Chem. Commun.* 38:2200–3

17. Bunker DL, Garret B, Kleindienst T, Long GS III. 1974. Discrete simulation methods in combustion kinetics. *Combust. Flame* 23:373–79

18. Horsak I, Solc M. 1975. Simulation study of a stochastic models of higher order reactions. *Collect. Czech. Chem. Commun.* 40:321–25

19. Nakanishi T. 1976. Time dependent fluctuation in a nonlinear chemical system. *J. Phys. Soc. Japan* 40:1232–39

20. Gibson MA, Bruck J. 2000. Exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.* 105:1876–89

1. Earliest application of the SSA to a real biological system demonstrating that stochasticity can play a critically important role.

7. Demonstrates the surprising ability of external noise to cause a monostable system to become bistable.

8. Introduced the SSA.

10. Clarified the physical and mathematical foundations of the CME and the SSA.

12. Introduced the CME.

21. Cao Y, Li H, Petzold LR. 2004. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.* 121:4059–67

22. Lok L, Brent R. 2005. Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat. Biotech.* 23:131–36

23. McCollum JM, Peterson GD, Cox CD, Simpson ML, Samatova NF. 2006. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Bio. Chem.* 30:39–49

24. Deleted in proof

25. Gave a conceptually transparent derivation of the CLE from the CME that clarified its form and the approximations involved.

25. Gillespie DT. 2000. The chemical Langevin equation. *J. Chem. Phys.* 113:297–306

26. Gillespie DT. 2001. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.* 115:1716–33

26. Introduced tau-leaping and a variation called k-leaping.

27. Gillespie DT. 1996. The multivariate Langevin and Fokker-Planck equations. *Am. J. Phys.* 64:1246–57

28. Gillespie DT. 2002. The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction. *J. Phys. Chem. A* 106:5063–71

27. A tutorial on Langevin equations, Fokker-Planck equations, Gaussian white noise, Brownian motion, and related subjects.

29. Gillespie DT. 2005. Stochastic chemical kinetics. In *Handbook of Materials Modeling*, ed. S Yip, pp. 1735–52. Dordrecht: Springer

30. Press W, Flannery B, Teukolsky S, Vetterling W. 1986. *Numerical Recipes: The Art of Scientific Computing*. Cambridge, UK: Cambridge Univ. Press

31. Gillespie DT, Petzold LR. 2003. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.* 119:8229–34

32. Cao Y, Gillespie DT, Petzold LR. 2006. Efficient stepsize selection for the tau-leaping simulation method. *J. Chem. Phys.* 124:044109

32. Describes the latest, most efficient version of explicit tau-leaping.

33. Tian T, Burrage K. 2004. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.* 121:10356–64

34. Chatterjee A, Vlachos D, Katsoulakis M. 2005. Binomial distribution based τ-leap accelerated stochastic simulation. *J. Chem. Phys.* 122:024112

35. Cao Y, Gillespie DT, Petzold LR. 2005. Avoiding negative populations in explicit Poisson tau-leaping. *J. Chem. Phys.* 123:054104

36. Ascher UM, Petzold LR. 1998. *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*. Philadelphia: Soc. Ind. Appl. Math.

37. Rathinam M, Petzold LR, Cao Y, Gillespie DT. 2003. Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. *J. Chem. Phys.* 119:12784–94

38. Cao Y, Petzold LR. 2005. Trapezoidal tau-leaping formula for the stochastic simulation of chemically reacting systems. *Proc. Found. Syst. Biol. Eng. (FOSBE 2005)*, pp. 149–52

39. Plowman KM. 1971. *Enzyme Kinetics*. New York: McGraw-Hill

40. Haseltine EL, Rawlings JB. 2002. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.* 117:6959–69

41. Rao C, Arkin AP. 2003. Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.* 118:4999–5010

42. Introduced the ssSSA.

42. Cao Y, Gillespie DT, Petzold LR. 2005. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.* 122:014116

43. Cao Y, Gillespie DT, Petzold LR. 2005. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *J. Comput. Phys.* 206:395–411

44. Samant A, Vlachos DG. 2005. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: a multiscale Monte Carlo algorithm. *J. Chem. Phys.* 123:144114

45. Cao Y, Gillespie DT, Petzold LR. 2005. Accelerated stochastic simulation of the stiff enzyme-substrate reaction. *J. Chem. Phys.* 123:144917

46. Haseltine EL, Rawlings JB. 2005. On the origins of approximations for stochastic chemical kinetics. *J. Chem. Phys.* 123:164115

47. Weinan E, Liu D, Vanden-Eijnden E. 2005. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *J. Chem. Phys.* 123:194107

48. Salis H, Kaznessis Y. 2005. Equation-free probabilistic steady-state approximation: dynamic application to the stochastic simulation of biochemical reaction networks. *J. Chem. Phys.* 123:214106

Contents