Chapter 1

Numerical Simulation for Biochemical Kinetics

Daniel T. Gillespie and Linda R. Petzold

In chemical systems formed by living cells, the small numbers of molecules of a few reactant species can result in dynamical behavior that is discrete and stochastic, rather than continuous and deterministic (McAdams & Arkin, 1999; McAdams & Arkin, 1997; Arkin et al., 1998; Elowitz et al., 2002; Fedoroff & Fontana, 2002). By "discrete", we mean the integer-valued nature of small molecular populations, which makes their representation by real-valued (continuous) variables inappropriate. By "stochastic", we mean the random behavior that arises from the lack of total predictability in molecular dynamics. In this chapter we introduce some concepts and techniques that have been developed for mathematically describing and numerically simulating chemical systems that take proper account of discreteness and stochasticity. Throughout, we shall make the simplifying assumption that the system is *well-stirred* or spatially homogeneous. In practice this assumption is often justified, and it allows us to specify the state of the system simply by giving the molecular populations of the various chemical species. But in some circumstances the well-stirred assumption will *not* be justified; then the locations of the molecules and the dynamics of their movement must also be considered. Some approaches to this more computationally challenging situation are described in Chapter **??**.

1.1 Chapter Overview

We begin in Section 1.2 by outlining the foundations of “stochastic chemical kinetics” and deriving the *chemical master equation* (CME), the time-evolution equation for the probability function of the system’s state. Unfortunately, the CME cannot be solved, either analytically or numerically, for any but the simplest of systems. But we can generate numerical realizations (sample trajectories in state space) of the stochastic process defined by the CME by using a Monte Carlo strategy called the *stochastic simulation algorithm* (SSA). The SSA is derived and discussed in Section 1.3. Although the SSA is an ideal algorithm in the sense that it provides exact realizations of the CME, there is a computational price for this: Because the SSA simulates *every* reaction event, it will be painfully slow for systems that involve enormous numbers of such events, which most real chemical systems do. This has motivated a search for algorithms that give up some of the exactness of the SSA in return for greater simulation speed.

One such approximate accelerated algorithm is known as *tau-leaping*, and it is described in Section 1.4. In tau-leaping, instead of advancing the system to the time of the next reaction event, the system is advanced by a pre-selected time τ which typically encompasses more than one reaction event. The number of times each reaction fires in time τ is approximated by a Poisson random variable, and we explain why that can be done in 1.4. In Section 1.5 we show how, under certain conditions, tau-leaping further approximates to a stochastic differential equation called the *chemical Langevin equation* (CLE), and then how the CLE can in turn sometimes be approximated by an ordinary differential equation called the *reaction rate equation* (RRE). Tau-leaping, the CLE, and the RRE are successively coarser-grained approximations which usually become appropriate as the molecular populations of the reacting species are made larger and larger.

In the past, virtually all chemically reacting systems were analyzed using the deterministic RRE, even though that equation is accurate only in the limit of infinitely large molecular populations. Near that limit though, the RRE practically always provides the most efficient description. One reason for this is the extensive theory that has been developed over the years for efficiently solving ordinary differential equations, especially those that are *stiff.* A stiff system of ordinary differential equations is one that involves processes occurring on vastly different time scales, the fastest of which is stable. Stiff RREs arise for chemical systems that contain a mixture of fast and slow reactions, and many if not most cellular systems are of this type. The practical consequence of stiffness is that, even though the system itself is stable, naive simulation techniques will be unstable unless they proceed in extremely small time steps. In Section 1.6 we describe the problem of stiffness in a deterministic (RRE) context, along with its standard numerical resolution: implicit methods.

Given the connections described above between tau-leaping, the CLE, and the RRE, it should not be surprising that stiffness is also an issue for tau-leaping and the CLE. In Section 1.7 we describe an *implicit tau-leaping algorithm* for stochastically simulating stiff chemical systems. We conclude in Section 1.8 by describing and illustrating yet another promising algorithm for dealing with stiff stochastic chemical systems, which we call the *slow-scale SSA*.

1.2 Foundations of Stochastic Chemical Kinetics and the Chemical Master Equation

We consider a well-stirred system of molecules of N chemical species $\{S_1, \ldots, S_N\}$ interacting through M chemical reaction channels $\{R_1, \ldots, R_M\}$. The system is assumed to be confined to a constant volume Ω, and to be in thermal (but not necessarily chemical) equilibrium at some constant temperature. With $X_i(t)$ denoting the number of molecules of species S_i in the system at time t, we want to study the evolution of the state vector $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$, given that the system was initially in some state $\mathbf{X}(t_0) = \mathbf{x}_0$.

Each reaction channel R_j is assumed to be "elemental" in the sense that it describes a distinct physical event which happens essentially instantaneously. Elemental reactions are either *unimolecular* or *bimolecular*; more complicated chemical reactions (including trimolecular reactions) are actually coupled sequences of two or more elemental reactions.

Reaction channel R_j is characterized mathematically by two quantities. The first is its *state-change vector* $\boldsymbol{\nu}_j = (\nu_{1j}, \ldots, \nu_{Nj})$, where ν_{ij} is defined to be the change in the S_i molecular population caused by one R_j reaction; thus, if the system is in state $\mathbf{x}$ and an R_j reaction occurs, the system immediately jumps to state $\mathbf{x} + \boldsymbol{\nu}_j$. The array $\{\nu_{ij}\}$ is commonly known as the stoichiometric matrix.

The other characterizing quantity for reaction channel R_j is its *propensity* *function* a_j. It is defined so that $a_j(\mathbf{x})\,dt$ gives the *probability*, given $\mathbf{X}(t) = \mathbf{x}$, that one R_j reaction will occur somewhere inside Ω in the next infinitesimal time interval $[t, t + dt)$. This probabilistic definition of the propensity function finds its justification in physical theory (Gillespie, 1992a; Gillespie, 1992b). If R_j is the unimolecular reaction $S_i \rightarrow$ products, the underlying physics is quantum mechanical, and implies the existence of some constant c_j such that $a_j(\mathbf{x}) = c_j x_i$. If R_j is the bimolecular reaction $S_i + S_{i'} \rightarrow$ products, the underlying physics implies a different constant c_j, and a propensity function $a_j(\mathbf{x})$ of the form $c_j x_i x_{i'}$ if $i \neq i'$, or $c_j \frac{1}{2} x_i(x_i - 1)$ if $i = i'$. The stochasticity of a bimolecular reaction stems from the fact that we do not know the precise positions and velocities of all the molecules in the system, so we can predict only the probability that an S_i molecule and an $S_{i'}$ molecule will collide in the next dt and then react according to R_j.

It turns out that c_j for a unimolecular reaction is numerically equal to the reaction rate constant k_j of conventional deterministic chemical kinetics, while c_j for a bimolecular reaction is equal to k_j/Ω if the reactants are different species, or $2k_j/\Omega$ if they are the same (Gillespie, 1976; Gillespie, 1992a; Gillespie, 1992b). But it would be wrong to infer from this that the propensity functions are simple heuristic extrapolations of the rates used in deterministic chemical kinetics; in fact, the inference flow actually goes the other way. The existence and forms of the propensity functions follow directly from molecular physics and kinetic theory, and not from deterministic chemical kinetics.

The probabilistic nature of the dynamics described above implies that the most we can hope to compute is the *probability* $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$ that $\mathbf{X}(t)$ will equal $\mathbf{x}$, given that $\mathbf{X}(t_0) = \mathbf{x}_0$. We can deduce a time-evolution equation for this function by using the laws of probability to write $P(\mathbf{x}, t + dt \,|\, \mathbf{x}_0, t_0)$ as:

$$\begin{aligned} P(\mathbf{x}, t + dt \,|\, \mathbf{x}_0, t_0) = P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0) \times [1 - \sum_{j=1}^{M} a_j(\mathbf{x})dt] \\ + \sum_{j=1}^{M} P(\mathbf{x} - \boldsymbol{\nu}_j, t \,|\, \mathbf{x}_0, t_0) \times a_j(\mathbf{x} - \boldsymbol{\nu}_j)dt. \end{aligned}$$

The first term on the right is the probability that the system is already in state $\mathbf{x}$ at time t and no reaction of any kind occurs in $[t, t + dt)$, and the generic second term is the probability that the system is one R_j reaction removed from state $\mathbf{x}$ at time t and one R_j reaction occurs in $[t, t + dt)$. That these $M + 1$

routes from time t to state $\mathbf{x}$ at time $t+dt$ are mutually exclusive and collectively exhaustive is ensured by taking dt so small that no more than one reaction of any kind can occur in $[t, t+dt)$. Subtracting $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ from both sides, dividing through by dt, and taking the limit $dt \to 0$, we obtain (McQuarrie, 1967; Gillespie, 1992a)

$$\begin{aligned}
&\frac{\partial P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)}{\partial t} \\
&\qquad = \sum_{j=1}^{M} [a_j(\mathbf{x} - \boldsymbol{\nu}_j) P(\mathbf{x} - \boldsymbol{\nu}_j, t \mid \mathbf{x}_0, t_0) - a_j(\mathbf{x}) P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)]. \qquad (1.1)
\end{aligned}$$

This is the *chemical master equation* (CME). In principle, it completely determines the function $P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$. But the CME is really a set of nearly as many coupled ordinary differential equations as there are combinations of molecules that can exist in the system! So it is not surprising that the CME can be solved analytically for only a very few very simple systems, and numerical solutions are usually prohibitively difficult.

One might hope to learn something from the CME about the behavior of averages like $\langle f(\mathbf{X}(t)) \rangle \equiv \sum_{\mathbf{x}} f(\mathbf{x}) P(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$, but this too turns out to pose difficulties if any of the reaction channels are bimolecular. For example, it can be proved from equation (1.1) that

$$\frac{d \langle X_i(t) \rangle}{dt} = \sum_{j=1}^{M} \nu_{ij} \langle a_j(\mathbf{X}(t)) \rangle \quad (i = 1, \ldots, N).$$

If all the reactions were unimolecular, the propensity functions would all be linear in the state variables, and we would have $\langle a_j(\mathbf{X}(t)) \rangle = a_j(\langle \mathbf{X}(t) \rangle)$. The above equation would then become a closed set of ordinary differential equations for the first moments, $\langle X_i(t) \rangle$. But if any reaction is bimolecular, the right hand side will contain at least one quadratic moment of the form $\langle X_i(t) X_{i'}(t) \rangle$, and the equation then becomes merely the first of an infinite, open-ended set of coupled quations for all the moments.

In the hypothetical case that there are *no fluctuations*, we would have $\langle f(\mathbf{X}(t)) \rangle = f(\mathbf{X}(t))$ for all functions f. The above equation for $\langle X_i(t) \rangle$ would then reduce to

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ij} \, a_j(\mathbf{X}(t)) \quad (i = 1, \ldots, N). \qquad (1.2)$$

This is the *reaction rate equation* (RRE) of traditional deterministic chemical kinetics — a set of N coupled first-order ordinary differential equations for the $X_i(t)$, which are now continuous (real) variables. The RRE is more commonly written in terms of the concentration variables $X_i(t)/\Omega$, but that scalar transformation is inconsequential for our purposes here. Examples of RREs in a biological context abound in Chapter **??**.

Although the deterministic RRE (1.2) would evidently be valid in the absence of fluctuations, it is not clear what the justification and penalty might be for ignoring fluctuations. We shall later see how the RRE follows more deductively from a series of physically transparent approximating assumptions to the stochastic theory.

1.3 The Stochastic Simulation Algorithm

Since the CME (1.1) is rarely of much use in computing the probability density function $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$ of $\mathbf{X}(t)$, we need another computational approach. One approach that has proven fruitful is 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 samplings of that random variable. However, much the same effect can be achieved by either histogramming or averaging the results of many realizations. The key to generating simulated trajectories of $\mathbf{X}(t)$ is not the CME or even the function $P(\mathbf{x}, t \,|\, \mathbf{x}_0, t_0)$, but rather a new function, $p(\tau, j \,|\, \mathbf{x}, t)$ (Gillespie, 1976). It is defined so that $p(\tau, j \,|\, \mathbf{x}, t)\, d\tau$ is the probability, given $\mathbf{X}(t) = \mathbf{x}$, that the *next* reaction in the system will occur in the infinitesimal time interval $[t + \tau, t + \tau + d\tau)$, *and* will be an R_j reaction. 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).

To derive an analytical expression for $p(\tau, j \,|\, \mathbf{x}, t)$, we begin by noting that if $P_0(\tau \,|\, \mathbf{x}, t)$ is the probability, given $\mathbf{X}(t) = \mathbf{x}$, that no reaction of any kind occurs in the time interval $[t, t + \tau)$, then the laws of probability imply the relation

$$p(\tau, j \,|\, \mathbf{x}, t)\, d\tau = P_0(\tau \,|\, \mathbf{x}, t) \times a_j(\mathbf{x}) d\tau.$$

The laws of probability also imply

$$P_0(\tau + d\tau \,|\, \mathbf{x}, t) = P_0(\tau \,|\, \mathbf{x}, t) \times [1 - \sum_{j'=1}^{M} a_{j'}(\mathbf{x}) d\tau].$$

An algebraic rearrangement of this last equation and passage to the limit $d\tau \to 0$ results in a differential equation whose solution is easily found to be $P_0(\tau \,|\, \mathbf{x}, t) = \exp\left(-a_0(\mathbf{x})\,\tau\right)$, where

$$a_0(\mathbf{x}) \equiv \sum_{j'=1}^{M} a_{j'}(\mathbf{x}). \tag{1.3}$$

When we insert this result into the equation for p, we get

$$p(\tau, j \,|\, \mathbf{x}, t) = a_j(\mathbf{x})\ \exp\left(-a_0(\mathbf{x})\,\tau\right). \tag{1.4}$$

Equation (1.4) is the mathematical basis for the stochastic simulation approach. It implies that the joint density function of τ and j can be written as the product of the τ-density function, $a_0(\mathbf{x})\ \exp\left(-a_0(\mathbf{x})\tau\right)$, and the j-density function, $a_j(\mathbf{x})/a_0(\mathbf{x})$. We can generate random samples from these two density functions by using the inversion method of Monte Carlo theory (Gillespie, 1992b): Draw two random numbers r_1 and r_2 from the uniform distribution in the unit-interval, and select τ and j according to

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right), \tag{1.5a}$$

$$j = \text{ the smallest integer satisfying } \sum_{j'=1}^{j} a_{j'}(\mathbf{x}) > r_2\, a_0(\mathbf{x}). \tag{1.5b}$$

Thus we arrive at the following version of the *stochastic simulation algorithm* (SSA) (Gillespie, 1976; Gillespie, 1977):

1. Initialize the time $t = t_0$ and the system's state $\mathbf{x} = \mathbf{x}_0$.

2. With the system in state $\mathbf{x}$ at time t, evaluate all the $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x})$.

3. Generate values for τ and j according to equations (1.5).

4. Effect the next reaction by replacing $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\nu}_j$.

5. Record $(\mathbf{x}, t)$ as desired. Return to Step 2, or else end the simulation.

The $\mathbf{X}(t)$ trajectory that is produced by the SSA might be thought of as a "stochastic version" of the trajectory that would be obtained by solving the RRE (1.2). But note that the time step τ in the SSA is *exact*, and is *not* a finite approximation to some infinitesimal dt, as is the time step in most numerical solvers for the RRE. If it is found that every SSA-generated trajectory is practically indistinguishable from the RRE trajectory, then we may conclude that microscale fluctuations are ignorable. But if the SSA trajectories deviate noticeably from the RRE trajectory, then we must conclude that microscale fluctuations are not ignorable, and the deterministic RRE does not provide an accurate description of the system's real behavior.

The SSA and the CME are logically equivalent to each other; yet even when the CME is completely intractable, the SSA is quite straightforward to implement. The problem with the SSA is that it is often very slow. The source of this slowness can be traced to the factor $1/a_0(\mathbf{x})$ in the τ equation (1.5a): $a_0(\mathbf{x})$ can be very large if the population of one or more reactant species is large, and that is often the case in practice.

There are variations on the above method for implementing the SSA that make it more computationally efficiency (Gibson & Bruck, 2000; Cao et al., 2004). But *any* procedure that simulates *every* reaction event one at a time will inevitably be too slow for most practical applications. This prompts us to look for ways of giving up some of the exactness of the SSA in return for greater simulation speed.

1.4 Tau-Leaping

One approximate accelerated simulation strategy is *tau-leaping* (Gillespie, 2001). It advances the system by a *pre-selected* time τ which encompasses more than one reaction event. In its simplest form, tau-leaping requires that τ be chosen small enough that the following *Leap Condition* is satisfied: The expected state change induced by the leap must be sufficiently small that no propensity function changes its value by a significant amount.

We recall that the Poisson random variable $\mathcal{P}(a, \tau)$ is by definition the number of events that will occur in time τ given that $a\,dt$ is the probability that an event will occur in any infinitesimal time dt, where a can be any positive constant. Therefore, if $\mathbf{X}(t) = \mathbf{x}$, and if we choose τ small enough to satisfy the

Leap Condition, so that the propensity functions stay approximately constant during the leap, then reaction R_j should fire approximately $\mathcal{P}_j\left(a_j(\mathbf{x}), \tau\right)$ times in $[t, t+\tau)$. Thus, to the degree that the Leap Condition is satisfied, we can leap by a time τ simply by taking

$$\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j \, \mathcal{P}_j\left(a_j(\mathbf{x}), \tau\right). \tag{1.6}$$

Doing this evidently requires generating M Poisson random numbers for each leap (Press et al., 1986). It will result in a faster simulation than the SSA to the degree that the total number of reactions leapt over, $\sum_{j=1}^{M} \mathcal{P}_j\left(a_j(\mathbf{x}), \tau\right)$, is large compared to M.

In order to use this simulation technique efficiently, we obviously need a way to estimate the largest value of τ that is compatible with the Leap Condition. One possible way of doing that (Gillespie & Petzold, 2003) is to estimate the largest value of τ for which no propensity function is likely to change its value during τ by more than $\varepsilon\, a_0(\mathbf{x})$, where ε $(0 < \varepsilon \ll 1)$ is some pre-chosen accuracy-control parameter. Whatever the method of selecting τ, the (explicit) *tau-leaping simulation procedure* goes as follows (Gillespie, 2001; Gillespie & Petzold, 2003):

1. In state $\mathbf{x}$ at time t, choose a value for τ that satisfies the Leap Condition.

2. For each $j = 1, \ldots, M$, generate the number of firings k_j of reaction R_j in time τ as a sample of the Poisson random variable $\mathcal{P}\left(a_j(\mathbf{x}), \tau\right)$.

3. Leap, by replacing $t \leftarrow t+\tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \sum\limits_{j=1}^{M} k_j \boldsymbol{\nu}_j$.

In the limit that $\tau \to 0$, tau-leaping becomes mathematically equivalent to the SSA. But tau-leaping also becomes very inefficient in that limit, because all the k_j's will approach zero, giving a very small time step with usually no reactions firing. As a practical matter, tau-leaping should not be used if the largest value of τ that satisfies the Leap Condition is less than a few multiples of $1/a_0(\mathbf{x})$, the expected time to the next reaction in the SSA, since it would then be more efficient to use the SSA.

Tau-leaping has been shown to significantly speed up the simulation of *some* systems (Gillespie, 2001; Gillespie & Petzold, 2003). But it is not as foolproof as the SSA. If one takes leaps that are too large, bad things can happen; e.g., some species populations might be driven negative. If the system is “stiff”, meaning that it has widely varying dynamical modes with the fastest mode being stable, the Leap Condition will generally limit the size of τ to the time scale of the fastest mode, with the result that large leaps cannot be taken. Stiffness is very common in cellular chemical systems, and will be considered in more detail later.

It is tempting to try to formulate a “higher-order” tau-leaping formula by extending higher-order ODE methods in a straightforward manner for discrete stochastic simulation. However, doing this correctly is much harder than it might at first appear. Most such extensions are not even first order accurate for the stochastic part of the system. An analysis of the consistency, order, and convergence of tau-leaping methods is given in (Rathinam et al., 2005), where it is shown that the tau-leaping method defined above, and the “implicit” tau-leaping method to be described in Section 7, are both first-order accurate as $\tau \to 0$.

1.5 Transitioning to the Macroscale: The Chemical Langevin Equation and the Reaction Rate Equation

Suppose we can choose τ small enough to satisfy the Leap Condition, so that approximation (1.6) is good, but nevertheless large enough that

$$a_j(\mathbf{x})\,\tau \gg 1 \quad \text{for all } j = 1, \ldots, M. \tag{1.7}$$

Since $a_j(\mathbf{x})\tau$ is the mean of the random variable $\mathcal{P}_j\left(a_j(\mathbf{x}), \tau\right)$, the physical significance of condition (1.7) is that each reaction channel is expected to fire many more times than once in the next τ. It will not always be possible to find a τ that satisfies *both* the Leap Condition and condition (1.7), but it usually will be if the populations of all the reactant species are sufficiently large.

When condition (1.7) does hold, we can make a useful approximation to the tau-leaping formula (1.6). This approximation stems from the purely mathematical fact that the Poisson random variable $\mathcal{P}(a, \tau)$, which has mean and variance $a\tau$, can be well approximated when $a\tau \gg 1$ by a *normal* random variable with the same mean and variance. Denoting the normal random variable with mean m and variance σ^2 by $\mathcal{N}(m, \sigma^2)$, it thus follows that when condition

(1.7) holds,

$$\mathcal{P}_j\left(a_j(\mathbf{x}),\tau\right) \doteq \mathcal{N}_j\left(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau\right) = a_j(\mathbf{x})\tau + \left(a_j(\mathbf{x})\tau\right)^{1/2}\mathcal{N}_j(0,1),$$

the last step following from the fact that $\mathcal{N}(m,\sigma^2) = m + \sigma\mathcal{N}(0,1)$. Inserting this approximation into equation (1.6) gives (Gillespie, 2000; Gillespie, 2002)

$$\mathbf{X}(t+\tau) \doteq \mathbf{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{x})\tau + \sum_{j=1}^{M} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{x})}\mathcal{N}_j(0,1)\sqrt{\tau}\,, \tag{1.8}$$

where the $\mathcal{N}_j(0,1)$ are statistically independent normal random variables with means 0 and variances 1. Equation (1.8) is called the *Langevin leaping formula.* It evidently expresses the state increment $\mathbf{X}(t+\tau) - \mathbf{x}$ as the sum of two terms: a deterministic *drift term* proportional to τ, and a fluctuating *diffusion term* proportional to $\sqrt{\tau}$. It is important to keep in mind that equation (1.8) is an *approximation*, which is valid only to the extent that τ is (i) *small* enough that no propensity function changes its value significantly during τ, yet (ii) *large* enough that every reaction fires many more times than once during τ. The approximate nature of (1.8) is underscored by the fact that $\mathbf{X}(t)$ therein is now a continuous (real-valued) random variable instead of a discrete (integer-valued) random variable; we lost discreteness when we replaced the integer-valued Poisson random variable with a real-valued normal random variable. The Langevin leaping formula (1.8) gives faster simulations than the tau-leaping formula (1.6) not only because condition (1.7) implies that very many reactions get leapt over at each step, but also because the normal random numbers that are required by (1.8) can be generated much more easily than the Poisson random numbers that are required by (1.6) (Press et al., 1986).

The "small-but-large" character of τ in equation (1.8) marks that variable as a "macroscopic infinitesimal". If we subtract $\mathbf{x}$ from both sides and then divide through by τ, the result can be shown to be the following (approximate) *stochastic differential equation*, which is called the *chemical Langevin equation* (CLE) (Gillespie, 2000; Gillespie, 2002):

$$\frac{d\mathbf{X}(t)}{dt} \doteq \sum_{j=1}^{M} \boldsymbol{\nu}_j\, a_j\left(\mathbf{X}(t)\right) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \sqrt{a_j\left(\mathbf{X}(t)\right)}\,\Gamma_j(t)\,. \tag{1.9}$$

The $\Gamma_j(t)$ here 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. The CLE (1.9) is mathematically equivalent to the Langevin leaping formula (1.8), and is subject to the same conditions for validity. Stochastic differential equations arise in many areas of physics, but the usual way of obtaining them is to start with a macroscopically inspired drift term (the first term on the right side of the CLE) and then *assume* a form for the diffusion term (the second term on the right side of the CLE) with an eye to obtaining some pre-conceived outcome. So it is noteworthy that our derivation here of the CLE did *not* proceed in that *ad-hoc* manner; instead, we used careful mathematical approximations to infer the forms of *both* the drift and diffusion terms from the premises underlying the CME/SSA.

Molecular systems become "macroscopic" in what physicists and chemists call the *thermodynamic limit*. This limit is formally defined as follows: The system volume Ω and the species populations X_i all approach ∞, but in such a way that the species concentrations X_i/Ω all remain constant. The large molecular populations in chemical systems near the thermodynamic limit generally mean that such systems will be well described by the Langevin formulas (1.8) and (1.9). To discern the implications of those formulas in the thermodynamic limit, we evidently need to know the behavior of the propensity functions in that limit. It turns out that all propensity functions grow *linearly* with the system size as the thermodynamic limit is approached. For a unimolecular propensity function of the form $c_j x_i$ this behavior is obvious, since c_j will be independent of the system size. For a bimolecular propensity function of the form $c_j x_i x_{i'}$ this behavior is a consequence of the fact that bimolecular c_j's are always inversely proportional to Ω, reflecting the fact that two reactant molecules have a harder time finding each other in larger volumes.

It follows that, as the thermodynamic limit is approached, the deterministic drift term in (1.8) grows like the size of the system, while the fluctuating diffusion term grows like the *square root* of the size of the system, and likewise for the CLE (1.9). This establishes the well known rule-of-thumb in chemical kinetics that *relative fluctuation effects in chemical systems typically scale as the inverse square root of the size of the system.*

In the *full thermodynamic limit*, the size of the second term on the right side of (1.9) will usually be negligibly small compared to the size of the first term, in which case the CLE (1.9) reduces to the RRE (1.2). Thus we have *derived* the RRE as a series of limiting approximations to the stochastic theory that underlies the CME and the SSA. The tau-leaping and Langevin-leaping formulas evidently provide a conceptual bridge between stochastic chemical kinetics (the CME and SSA) and conventional deterministic chemical kinetics (the RRE), enabling us to see how the latter emerges as a limiting approximation of the former.

1.6 Stiffness in Deterministic Reaction Rate Equations

Stiffness can be defined roughly as the presence of widely varying time-scales in a dynamical system, the fastest of which is stable. It poses special problems for the numerical solution of both deterministic ordinary differential equations (ODEs) and stochastic differential equations (SDEs), particularly in the context of chemical kinetics. Stiffness also impacts both the SSA and the tau-leaping algorithm (1.6). In this section we will describe the phenomenon of stiffness for deterministic systems of ODEs, and show how it restricts the timestep size for all "explicit" methods. Then we will show how the use of "implicit" methods overcomes this restriction.

Consider the deterministic ODE system

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}). \tag{1.10}$$

In simplest terms, this system is said to be "stiff" if it has a strongly damped, or "superstable" mode. To get a feeling for this concept, consider the solutions $\mathbf{x}(t)$ of an ODE system starting from various initial conditions. For a typical *nonstiff* system, if we plot a given component of the vector $\mathbf{x}$-versus-t we might get a family of curves resembling those shown in Figure (1.1a): The curves either remain roughly the same distance apart as t increases, as in the figure, or they might show a tendency to merge very slowly. But when such a family of curves is plotted for a typical *stiff* system, the result looks more like what is shown in Figure (1.1b): The curves merge rapidly to one or more smoother curves, with the deviation from the smoother curves becoming very small as t increases.

Stiffness in a system of ODEs corresponds to a strongly stable behavior of the physical system being modeled. At any given time the system will be in a sort of equilibrium, although not necessarily a static one, and if some state variable is perturbed slightly, the system will respond rapidly to restore the equilibrium. Typically, the true solution $\mathbf{x}(t)$ of the ODE system does not show any rapid

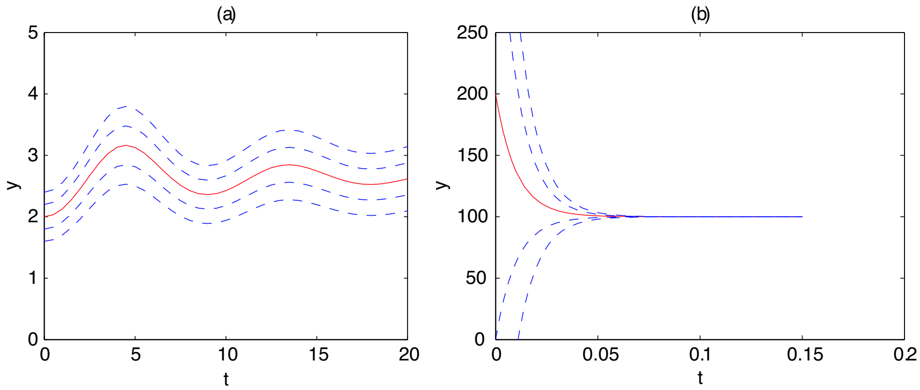

Figure 1.1: A system of ODEs is said to be "stiff" if its solutions show strongly damped behavior as a function of the initial conditions. The family of curves shown in (a) represents the behavior of solutions to a nonstiff system for various initial conditions. In contrast, solutions to the stiff system shown in (b) tend to merge quickly.

variation, except possibly during an initial transient phase. But the potential for rapid response is always present, and will manifest itself if we perturb $\mathbf{x}$ out of equilibrium. A stiff system has (at least) two time scales. There is a long (slow) time scale for the quasi-equilibrium phase, and a short (fast) time scale for the transient phase following a perturbation. The more different these two time scales are, the "stiffer" the system is said to be.

The smallest (fastest) time scale in a stiff system manifests itself in another way when we try to carry out a numerical solution of the system. Solution by an explicit time stepping method, such as the simple *explicit Euler method*

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \tau \mathbf{f}(t_{n-1}, \mathbf{x}_{n-1}), \tag{1.11}$$

where $t_n = t_{n-1} + \tau$ and $\mathbf{x}_n$ is the numerical approximation to $\mathbf{x}(t_n)$, will produce very inaccurate results unless the time stepsize τ is kept smaller than the smallest time scale in the system.

To see why this is so, let us consider a simple example: the reversible isomerization reaction, $S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2$. Let x_{T} denote the (constant) total number of molecules of the two isomeric species, and $x(t)$ the time-varying number of S_1

molecules. The deterministic RRE for this system is the ODE

$$\frac{dx}{dt} = -c_1 x + c_2(x_{\mathrm{T}} - x) = -(c_1 + c_2)x + c_2 x_{\mathrm{T}}. \tag{1.12}$$

The solution to this ODE for the initial condition $x(0) = x_0$, is given by

$$x(t) = \frac{c_2 x_{\mathrm{T}}}{c_1 + c_2} + \left(x_0 - \frac{c_2 x_{\mathrm{T}}}{c_1 + c_2} \right) e^{-(c_1+c_2)t}.$$

¿From the form of this solution, we can see that if the initial value x_0 differs from the asymptotic value $\frac{c_2 x_{\mathrm{T}}}{c_1+c_2}$, the solution will relax to that asymptotic value in a time of order $(c_1 + c_2)^{-1}$; therefore, if $(c_1 + c_2)$ is very 1arge, this system will be stiff. In Figure (1.2) we show the exact solution of the reversible isomerization reaction (1.12) along with numerical solutions obtained using the explicit Euler method (1.11) with two different stepsizes τ. Note that the smaller stepsize Euler solution is accurate, but the larger stepsize solution is unstable.

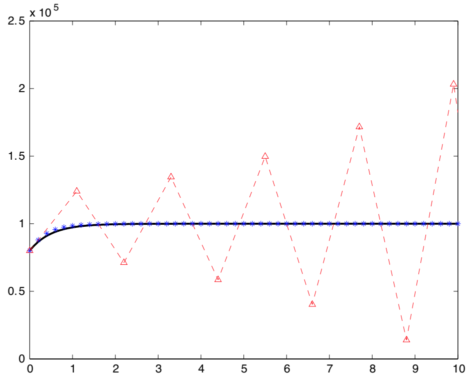

Figure 1.2: Exact solution of (1.12) (solid line) and its explicit Euler approximations for stepsizes 0.2 (asterisks) and 1.1 (triangles) with $c_1 = c_2 = 1$ and $x_{\mathrm{T}} = 2 \times 10^5$. The fast time constant for this problem is $(c_1 + c_2)^{-1} = 0.5$.

To see why this instability arises, let us write down the explicit Euler method (1.11) with stepsize τ for the ODE (1.12):

$$x_n = x_{n-1} - \tau(c_1 + c_2)x_{n-1} + \tau c_2 x_{\mathrm{T}}. \tag{1.13}$$

If we expand the true solution $x(t)$ in a Taylor series about t_{n-1}, we get

$$x(t_n) = x(t_{n-1}) - \tau(c_1 + c_2)x(t_{n-1}) + \tau c_2 x_{\mathrm{T}} + O(\tau^2). \tag{1.14}$$

Subtracting (1.14) from (1.13), and defining the error $e_n = x_n - x(t_n)$, we obtain

$$e_n = e_{n-1} - \tau(c_1 + c_2)e_{n-1} + O(\tau^2). \tag{1.15}$$

Thus, e_n is given by the recurrence formula

$$e_n = (1 - \tau(c_1 + c_2))\, e_{n-1} + O(\tau^2). \tag{1.16}$$

If $\tau > 2(c_1 + c_2)^{-1}$, then $|1 - \tau(c_1 + c_2)|$ will be greater than 1, and we will have $|e_n| > |e_{n-1}|$. The recurrence will then be unstable. In general, to ensure the stability of an explicit method, we must restrict the stepsize to the timescale of the fastest mode, even though much larger stepsizes might seem perfectly acceptable for getting an adequate resolution of the solution curve.

The restriction of the explicit Euler method (1.11) to timesteps τ that are on the order of the short (fast) time scale makes the method very slow for stiff systems. So it is natural to ask if there are other solution methods for which the timesteps are not restricted by stability, but only by the need to resolve the solution curve. It is now widely recognized that a general way of doing this is provided by "implicit" methods (Ascher & Petzold, 1998), the simplest of which is the *implicit Euler method*. For the ODE (1.10), it reads

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \tau\mathbf{f}(t_n, \mathbf{x}_n). \tag{1.17}$$

In contrast to the explicit Euler formula (1.11), this method is implicit because $\mathbf{x}_n$ is not defined entirely in terms of past values of the solution; instead, it is defined implicitly as the solution of the (possibly nonlinear) system of equations (1.17). We can write this system abstractly as

$$\mathbf{F}(\mathbf{u}) = 0, \tag{1.18}$$

where $\mathbf{u} = \mathbf{x}_n$ and $\mathbf{F}(\mathbf{u}) = \mathbf{u} - \mathbf{x}_{n-1} - \tau\mathbf{f}(t_n, \mathbf{u})$. Usually, the most efficient way to numerically solve the system of equations (1.18) is by *Newton iteration*: One iterates the formula

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right)[\mathbf{u}^{(m+1)} - \mathbf{u}^{(m)}] = -\mathbf{F}(\mathbf{u}^{(m)}) \tag{1.19}$$

over m, where $\mathbf{u}^{(m)}$ is the m^{th} iterated approximation to the exact root of F, and the Jacobian matrix $\partial\mathbf{F}/\partial\mathbf{u}$ is evaluated at $\mathbf{u}^{(m)}$. This is a *linear* system of equations, which is to be solved at each iteration for $\mathbf{u}^{m+1}$. Newton's method converges in one iteration for linear systems, and the convergence is quite rapid for most nonlinear systems given a good initial guess. The initial guess is usually obtained by evaluating a polynomial that coincides with recent past solution values at t_n. In practice, the Jacobian matrix is usually not reevaluated at each iteration; also, it is often approximated by numerical difference quotients rather than evaluated exactly. The use of an approximate Jacobian that is fixed throughout the iteration is called *modified Newton* iteration. On first glance, it might seem that the expense of solving the nonlinear system at each time step would outweigh the advantage of increased stability; however, this is usually not so. For stiff systems, implicit methods are usually able to take timesteps that are so much larger than those of explicit methods that the implicit methods wind up being much more efficient.

To see why the implicit Euler method does not need to restrict the step size to maintain stability for stiff systems, let us consider again the reversible isomerization reaction (1.12). For it, the implicit Euler method reads [cf. (1.13)]

$$x_n = x_{n-1} - \tau(c_1 + c_2)x_n + \tau c_2 x_{\mathrm{T}}. \tag{1.20}$$

Expanding the true solution in a Taylor series about t_n, we get [cf. (1.14)]

$$x(t_n) = x(t_{n-1}) - \tau(c_1 + c_2)x(t_n) + \tau c_2 x_{\mathrm{T}} + O(\tau^2). \tag{1.21}$$

Subtracting (1.21) from (1.20), we find that the error $e_n = x_n - x(t_n)$ now satisfies [cf. (1.15)]

$$e_n = e_{n-1} - \tau(c_1 + c_2)e_n + O(\tau^2). \tag{1.22}$$

Solving this for e_n, we get

$$e_n = \frac{e_{n-1}}{1 + \tau(c_1 + c_2)} + O(\tau^2). \tag{1.23}$$

In contrast to the error (1.16) for the *explicit* Euler method, the error for the implicit Euler method remains small for arbitrarily large values of $\tau(c_1 + c_2)$, as seen in Figure 1.3.

For the general ODE system (1.10), the negative eigenvalues of the matrix $J = \partial\mathbf{f}/\partial\mathbf{x}$ play the role of $(c_1 + c_2)$. For stiff systems, the eigenvalues λ of J will include at least one with a relatively large negative real part, corresponding in the case of an RRE to the fastest reactions. The set of complex numbers $\tau\lambda$ satisfying $|1 + \tau\lambda| < 1$ is called the *region of absolute stability* of the explicit

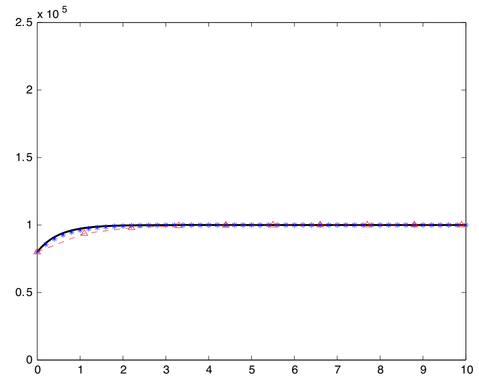

Figure 1.3: The implicit Euler method overcomes a weakness of the explicit Euler method in that it does not need to restrict the step size to provide stable solutions for stiff systems. The figure shows the true solution of the deterministic reversible isomerization reaction (1.12) (solid line), and the numerical solution by the implicit Euler method for stepsizes 0.2 (asterisks) and 1.1 (triangles) with $c_1 = c_2 = 1$ and $x_{\mathrm{T}} = 2 \times 10^5$. Compare with Figure (1.2).

Euler method. The corresponding region for the implicit Euler method is given by $1/|1 - \tau\lambda| < 1$, and it will be much larger.

A great deal of work has gone into the numerical solution of stiff systems of ODEs (and of ODEs coupled with nonlinear constraints, called differential algebraic equations (DAEs)). There is extensive theory and highly efficient and reliable software which adapts both the method order and the timestep to the given problem. See (Ascher & Petzold, 1998) for more details.

1.7 Stiffness in Stochastic Chemical Kinetics: The Implicit Tau-Leaping Method

When stochasticity is introduced into a chemical system that has fast and slow time scales, with the fast mode being stable as before, we may still expect there to be a slow manifold corresponding to the equilibrium of the fast reactions. But stochasticity changes the picture in a fundamental way: Once the system reaches the slow manifold, naturally occurring fluctuations will drive it back off, leading to *persistent random fluctuations transverse to the slow manifold.*

If these fluctuations are negligibly small, then an implicit scheme which takes large steps (on the time scale of the slow mode) will do just fine. But if the fluctuations off the slow manifold are noticeable, then an implicit scheme that takes steps much larger than the time scale of the fast dynamics will *dampen* the fluctuations, and thus fail to reproduce them correctly. Fortunately, this failing can usually be corrected by using a procedure called *down-shifting*, which we will describe shortly.

The original tau-leaping method (1.6) is explicit because the propensity functions a_j are evaluated at the current (known) state, so the future (unknown) random state $\mathbf{X}(t+\tau)$ is given as an explicit function of $\mathbf{X}(t)$. It is this explicit nature of (1.6) that leads to stability problems when stiffness is present, just as with ordinary differential equations. One way of making the explicit tau-leaping formula (1.6) *implicit* is to modify it as follows (Rathinam et al., 2003):

$$\begin{aligned}\mathbf{X}(t+\tau) \doteq \mathbf{X}(t) &+ \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}(t+\tau))\,\tau \\ &+ \sum_{j=1}^{M} \boldsymbol{\nu}_j \left[\mathcal{P}_j(a_j(\mathbf{X}(t)),\tau) - a_j(\mathbf{X}(t))\,\tau\right]. \qquad (1.24)\end{aligned}$$

Since the random variables $\mathcal{P}_j(a_j(\mathbf{X}(t),\tau)$ here can be generated without knowing $\mathbf{X}(t+\tau)$, then once values for those random variables are set, (1.24) becomes an ordinary implicit equation for the unknown state $\mathbf{X}(t+\tau)$, and $\mathbf{X}(t+\tau)$ can then be found by applying Newton iteration to (1.24).

Just as the explicit-tau method segues to the explicit Euler methods for SDEs and ODEs, the implicit-tau method segues to the implicit Euler methods for SDEs and ODEs. In the SDE regime we get, approximating Poissons by normals, the implicit version of the Langevin leaping formula (1.8):

$$\mathbf{X}(t+\tau) \doteq \mathbf{X}(t) + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}(t+\tau)) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{X}(t))}\mathcal{N}_j(0,1)\sqrt{\tau}. \quad (1.25)$$

Here, the $\mathcal{N}_j(0,1)$ are, as in (1.8), independent normal random variables with mean zero and variance 1. And in the thermodynamic limit, where the random terms in (1.25) may be ignored, it reduces to the implicit Euler method

$$\mathbf{X}(t+\tau) \doteq \mathbf{X}(t) + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}(t+\tau)) \qquad (1.26)$$

for the deterministic RRE (1.2).

We noted earlier that the implicit tau method, when used with a relatively large timestep, will damp out the natural fluctuations of the fast variables. Thus, although the implicit tau-leaping method computes the *slow* variables with their correct distributions, it computes the *fast* variables with the correct means but with spreads about those means that are too narrow. Fortunately, a time-stepping strategy called *down-shifting* can restore the overly-damped fluctuations in the fast variables. The idea is to interlace the implicit tau-leaps, each of which is on the order of the time scale of the slow variables and hence "large", with a sequence of much smaller time steps on the time scale of the fast variables, these being taken using either the explicit-tau method or the SSA. This sequence of smaller steps "regenerates" the correct statistical distributions of the fast variables. Further details on implicit tau-leaping and down-shifting can be found in (Rathinam et al., 2003).

1.8 Stiffness in Stochastic Chemical Kinetics: The Slow-Scale SSA

Another way to deal with stiffness in stochastic systems is to use the recently developed (Cao et al., 2005) *Slow-Scale SSA* (ssSSA). The first step in setting up the ssSSA is to divide (and re-index) the M reaction channels $\{R_1, \ldots, R_M\}$ into fast and slow subsets, $\{R_1^{\mathrm{f}}, \ldots, R_{M_{\mathrm{f}}}^{\mathrm{f}}\}$ and $\{R_1^{\mathrm{s}}, \ldots, R_{M_{\mathrm{s}}}^{\mathrm{s}}\}$, where $M_{\mathrm{f}} + M_{\mathrm{s}} = M$. We initially do this *provisionally* (subject to possible later change) according to the following criterion: the propensity functions of the fast reactions, $a_1^{\mathrm{f}}, \ldots, a_{M_{\mathrm{f}}}^{\mathrm{f}}$, should usually be very much larger than the propensity functions of the slow reactions, $a_1^{\mathrm{s}}, \ldots, a_{M_{\mathrm{s}}}^{\mathrm{s}}$. The broad result of this partitioning will be that the time to the occurrence of the next fast reaction will usually be very much smaller than the time to the occurrence of the next slow reaction.

Next we divide (and re-index) the N species $\{S_1, \ldots, S_N\}$ into fast and slow subsets, $\{S_1^{\mathrm{f}}, \ldots, S_{N_{\mathrm{f}}}^{\mathrm{f}}\}$ and $\{S_1^{\mathrm{s}}, \ldots, S_{N_{\mathrm{s}}}^{\mathrm{s}}\}$, where $N_{\mathrm{f}} + N_{\mathrm{s}} = N$. This gives rise to a like partitioning of the state vector $\mathbf{X}(t) = \left(\mathbf{X}^{\mathrm{f}}(t), \mathbf{X}^{\mathrm{s}}(t)\right)$, and also the generic state space variable $\mathbf{x} = \left(\mathbf{x}^{\mathrm{f}}, \mathbf{x}^{\mathrm{s}}\right)$, into fast and slow components. The criterion for making this partitioning is simple: A fast species is any species whose population *gets changed* by some *fast* reaction; all the other species are slow. Note the asymmetry in this definition: a slow species cannot get changed by a fast reaction, but a fast species can get changed by a slow reaction. Note also that a_j^{f} and a_j^{s} can both depend on both fast and slow variables. The state-change vectors can now be re-indexed

$$\boldsymbol{\nu}_j^{\mathrm{f}} \equiv \left(\nu_{1j}^{\mathrm{ff}}, \ldots, \nu_{N_{\mathrm{f}}j}^{\mathrm{ff}}\right), \quad j = 1, \ldots, M_{\mathrm{f}},$$

$$\boldsymbol{\nu}_j^{\mathrm{s}} \equiv \left(\nu_{1j}^{\mathrm{fs}}, \ldots, \nu_{N_{\mathrm{f}}j}^{\mathrm{fs}}, \nu_{1j}^{\mathrm{ss}}, \ldots, \nu_{N_{\mathrm{s}}j}^{\mathrm{ss}}\right), \quad j = 1, \ldots, M_{\mathrm{s}},$$

where $\nu_{ij}^{\sigma\rho}$ denotes the change in the number of molecules of species S_i^{σ} ($\sigma =$ f, s) induced by one reaction R_j^{ρ} ($\rho =$ f, s). We can regard $\boldsymbol{\nu}_j^{\mathrm{f}}$ as a vector with the same dimensionality (N_{f}) as $\mathbf{X}^{\mathrm{f}}$, because $\nu_{ij}^{\mathrm{sf}} \equiv 0$ (slow species do not get changed by fast reactions).

The next step in setting up the ssSSA is to introduce the *virtual fast process* $\hat{\mathbf{X}}^{\mathrm{f}}(t)$. It is composed of the same fast species state variables as the real fast process $\mathbf{X}^{\mathrm{f}}(t)$, but it evolves *only* through the fast reactions; i.e., $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ is $\mathbf{X}^{\mathrm{f}}(t)$ with all the slow reactions switched off. To the extent that the slow reactions don't occur very often, we may expect $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ and $\mathbf{X}^{\mathrm{f}}(t)$ to be very similar to each other. But from a mathematical standpoint there is an profound difference: $\mathbf{X}^{\mathrm{f}}(t)$ by itself is not a Markov (past-forgetting) process, whereas $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ is. Since the evolution of $\mathbf{X}^{\mathrm{f}}(t)$ depends on the evolving slow process $\mathbf{X}^{\mathrm{s}}(t)$, $\mathbf{X}^{\mathrm{f}}(t)$ is not governed by a master equation of the simple Markovian form (1.1); indeed, the easiest way to find $\mathbf{X}^{\mathrm{f}}(t)$ would be to solve the Markovian master equation for the full process $\mathbf{X}(t) \equiv \left(\mathbf{X}^{\mathrm{f}}(t), \mathbf{X}^{\mathrm{s}}(t)\right)$, which is something we have tacitly assumed cannot be done. But for the *virtual* fast process $\hat{\mathbf{X}}^{\mathrm{f}}(t)$, the slow process $\mathbf{X}^{\mathrm{s}}(t)$ stays fixed at some constant initial value $\mathbf{x}_0^{\mathrm{s}}$; therefore, $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ evolves according to the Markovian master equation,

$$\begin{aligned}&\frac{\partial \hat{P}(\mathbf{x}^{\mathrm{f}}, t \,|\, \mathbf{x}_0, t_0)}{\partial t} \\ &\quad = \sum_{j=1}^{M_{\mathrm{f}}} \left[a_j^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}} - \boldsymbol{\nu}_j^{\mathrm{f}}, \mathbf{x}_0^{\mathrm{s}}) \hat{P}(\mathbf{x}^{\mathrm{f}} - \boldsymbol{\nu}_j^{\mathrm{f}}, t \,|\, \mathbf{x}_0, t_0) - a_j^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}}, \mathbf{x}_0^{\mathrm{s}}) \hat{P}(\mathbf{x}^{\mathrm{f}}, t \,|\, \mathbf{x}_0, t_0) \right],\end{aligned}$$

wherein $\hat{P}(\mathbf{x}^{\mathrm{f}}, t \,|\, \mathbf{x}_0, t_0)$ is the probability that $\hat{\mathbf{X}}^{\mathrm{f}}(t) = \mathbf{x}^{\mathrm{f}}$, given that $\mathbf{X}(t_0) = \mathbf{x}_0$. This master equation for $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ will be simpler than the master equation for $\mathbf{X}(t)$ because it has fewer reactions and fewer species.

Finally, in order to apply the ssSSA, we require that *two conditions* be satisfied. The first condition is that the virtual fast process $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ be *stable*, in the sense that it approaches a well defined, time-independent random variable $\hat{\mathbf{X}}^{\mathrm{f}}(\infty)$ as $t \to \infty$; thus, we require the limit

$$\lim_{t\to\infty} \hat{P}(\mathbf{x}^{\mathrm{f}}, t \,|\, \mathbf{x}_0, t_0) \equiv \hat{P}(\mathbf{x}^{\mathrm{f}}, \infty \,|\, \mathbf{x}_0)$$

to exist. $\hat{P}(\mathbf{x}^{\mathrm{f}}, \infty \,|\, \mathbf{x}_0)$ can be calculated from the stationary form of the time-dependent master equation,

$$0 = \sum_{j=1}^{M_{\mathrm{f}}} \Big[a_j^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}} - \boldsymbol{\nu}_j^{\mathrm{f}}, \mathbf{x}_0^{\mathrm{s}}) \hat{P}(\mathbf{x}^{\mathrm{f}} - \boldsymbol{\nu}_j^{\mathrm{f}}, \infty \,|\, \mathbf{x}_0) - a_j^{\mathrm{f}}(\mathbf{x}^{\mathrm{f}}, \mathbf{x}_0^{\mathrm{s}}) \hat{P}(\mathbf{x}^{\mathrm{f}}, \infty \,|\, \mathbf{x}_0) \Big],$$

which will be easier to solve since it is purely algebraic. The second condition we impose is that the relaxation of $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ to its stationary asymptotic form $\hat{\mathbf{X}}^{\mathrm{f}}(\infty)$ happen *very quickly* on the time scale of the *slow* reactions. More precisely, we require that the relaxation time of the virtual fast process be very much less than the expected time to the next slow reaction.

These two conditions will generally be satisfied if the system is stiff. If satisfying them can be accomplished only by making some changes in the way we originally partitioned the reactions into fast and slow subsets, then we do that, regardless of propensity function values. But if these conditions cannot be satisfied, we must conclude that the ssSSA is not applicable.

Given the forgoing definitions and conditions, it is possible to prove the *Slow-Scale Approximation*:(Cao et al., 2005) If the system is in state $(\mathbf{x}^{\mathrm{f}}, \mathbf{x}^{\mathrm{s}})$ at time t, and if Δ_{s} is a time increment that is *very large* compared to the relaxation time of $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ but *very small* compared to the expected time to the next slow reaction, then the probability that one R_j^{s} reaction will occur in the time interval $[t, t + \Delta_{\mathrm{s}})$ can be well approximated by $\bar{a}_j^{\mathrm{s}}(\mathbf{x}^{\mathrm{s}}; \mathbf{x}^{\mathrm{f}})\, \Delta_{\mathrm{s}}$, where

$$\bar{a}_j^{\mathrm{s}}(\mathbf{x}^{\mathrm{s}}; \mathbf{x}^{\mathrm{f}}) \triangleq \sum_{\mathbf{x}^{\mathrm{f}'}} \hat{P}(\mathbf{x}^{\mathrm{f}'}, \infty \,|\, \mathbf{x}^{\mathrm{f}}, \mathbf{x}^{\mathrm{s}})\; a_j^{\mathrm{s}}(\mathbf{x}^{\mathrm{f}'}, \mathbf{x}^{\mathrm{s}}). \tag{1.28}$$

We call $\bar{a}_j^{\mathrm{s}}(\mathbf{x}^{\mathrm{s}}; \mathbf{x}^{\mathrm{f}})$ the *slow-scale propensity* function for reaction channel R_j^{s} because it serves as a propensity function for R_j *on the timescale of the slow reactions*. Mathematically, it is the average of the regular R_j^{s} propensity function over the fast variables, treated as though they were distributed according to the asymptotic virtual fast process $\hat{\mathbf{X}}^{\mathrm{f}}(\infty)$.

The *slow-scale SSA* is an immediate consequence of this Slow-Scale Approximation. The idea is to move the system forward in time in the manner of the

SSA *one slow reaction at a time*, updating the fast variables after each step by randomly sampling $\hat{\mathbf{X}}^{\mathrm{f}}(\infty)$. (Cao et al., 2005)

To illustrate how the ssSSA works, consider the simple reaction set

$$S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2 \xrightarrow{c_3} S_3 \tag{1.29}$$

under the condition

$$c_2 \gg c_3. \tag{1.30}$$

Here, an S_2 molecule is most likely to change into an S_1 molecule, a change that is relatively unimportant since it will eventually be reversed. On rare occasions though, an S_2 molecule will instead change into an S_3 molecule, a potentially more important change since it is irreversible. As an example, S_2 might be the active form of an enzyme which either becomes deactivated via reaction R_2 (and subsequently reactivated via reaction R_1), or binds to a DNA promoter site via reaction R_3 thereby allowing the transcription of some important gene. For this example we might be particularly interested in situations where the *average* number of S_2 molecules is very small, even less than 1, since that sometimes happens in practice.

We shall take the fast reactions to be R_1 and R_2, and the slow reaction to be R_3. Then the fast species will be S_1 and S_2, and the slow species S_3. The virtual fast process $\hat{\mathbf{X}}^{\mathrm{f}}(t)$ will be the S_1 and S_2 populations undergoing *only* the fast reactions R_1 and R_2. Unlike the *real* fast process, which gets affected whenever R_3 fires, the *virtual* fast process obeys the conservation relation

$$\hat{X}_1(t) + \hat{X}_2(t) = x_{\mathrm{T}} \quad \text{(constant)}. \tag{1.31}$$

This relation greatly simplifies the analysis of the virtual fast process, since it reduces the problem to a single independent state variable.

Eliminating $\hat{X}_2(t)$ in favor of $\hat{X}_1(t)$ by means of equation (1.31), we see that given $\hat{X}_1(t) = x_1'$, $\hat{X}_1(t+dt)$ will equal $x_1' - 1$ with probability $c_1 x_1' dt$, and $x_1' + 1$ with probability $c_2(x_{\mathrm{T}} - x_1')dt$. $\hat{X}_1(t)$ is therefore what is known mathematically as a "bounded birth-death" Markov process. It can be shown (Gillespie, 2002) that this process has, for any initial value $x_1 \in [0, x_{\mathrm{T}}]$, the asymptotic stationary distribution

$$\hat{P}(x_1', \infty \,|\, x_{\mathrm{T}}) = \frac{x_{\mathrm{T}}!}{x_1'!\,(x_{\mathrm{T}} - x_1')!} q^{x_1'} (1-q)^{x_{\mathrm{T}} - x_1'}, \quad (x_1' = 0, 1, \ldots, x_{\mathrm{T}}) \tag{1.32}$$

where $q \equiv c_2/(c_1+c_2)$. This tells us that $\hat{X}_1(\infty)$ is the binomial random variable $\mathcal{B}(q, x_{\mathrm{T}})$, whose mean and variance are given by

$$\left\langle \hat{X}_1(\infty) \right\rangle = x_{\mathrm{T}} q = \frac{c_2 x_{\mathrm{T}}}{c_1+c_2}, \tag{1.33a}$$

$$\operatorname{var}\left\{ \hat{X}_1(\infty) \right\} = x_{\mathrm{T}} q(1-q) = \frac{c_1 c_2 x_{\mathrm{T}}}{(c_1+c_2)^2}. \tag{1.33b}$$

It can also be shown (Cao et al., 2005) that $\hat{X}_1(t)$ relaxes to $\hat{X}_1(\infty)$ in a time of order $(c_1+c_2)^{-1}$.

The slow scale propensity function for the slow reaction R_3 is, according to the result (1.28), the average of $a_3(\mathbf{x}) = c_3 x_2$ with respect to $\hat{\mathbf{X}}^{\mathrm{f}}(\infty)$. Therefore, using equations (1.31) and (1.33a),

$$\bar{a}_3(x_3; x_1, x_2) = c_3 \left\langle \hat{X}_2(\infty) \right\rangle = \frac{c_3 c_1 (x_1+x_2)}{c_1+c_2}. \tag{1.34}$$

Since the reciprocal of $\bar{a}_3(x_3; x_1, x_2)$ estimates the average time to the next R_3 reaction, the condition that the relaxation time of the virtual fast process be very much smaller than the mean time to the next slow reaction is

$$c_1 + c_2 \gg \frac{c_3 c_1 (x_1+x_2)}{c_1+c_2}. \tag{1.35}$$

This condition will be satisfied if the inequality (1.30) is sufficiently strong. In that case, the Slow-Scale SSA for reactions (1.29) goes as follows:

1. Given $\mathbf{X}(t_0) = (x_{10}, x_{20}, x_{30})$, set $t \leftarrow t_0$ and $x_i \leftarrow x_{i0}$ $(i = 1, 2, 3)$.
2. In state (x_1, x_2, x_3) at time t, compute $\bar{a}_3(x_3; x_1, x_2)$ from equation (1.34).
3. Draw a unit-interval uniform random number r, and compute

$$\tau = \frac{1}{\bar{a}_3(x_3; x_1, x_2)} \ln\left(\frac{1}{r}\right).$$

4. Advance to the next R_3 reaction by replacing $t \leftarrow t + \tau$ and

$$x_3 \leftarrow x_3 + 1,\ x_2 \leftarrow x_2 - 1,$$

$$\begin{cases} \text{With } x_{\mathrm{T}} = x_1 + x_2, \\ x_1 \leftarrow \text{ sample of } \mathcal{B}\left(\dfrac{c_2}{c_1+c_2}, x_{\mathrm{T}}\right), \\ x_2 \leftarrow x_{\mathrm{T}} - x_1. \end{cases}$$

5. Record (t, x_1, x_2, x_3) if desired. Then return to Step 2, or else stop.

In Step 4, the x_3 update and the first x_2 update actualize the R_3 reaction. The bracketed procedure then "relaxes" the fast variables in a manner consistent with the stationary distribution (1.32) and the new value of x_T. See (Press et al., 1986) for a way to generate samples of the binomial random variable $\mathcal{B}(q, x_\mathrm{T})$.

Figure 1.4 (a) shows the results of an exact SSA run of reactions (1.29) for the parameter values

$$c_1 = 10, \;\; c_2 = 4 \times 10^4, \;\; c_3 = 2; \quad x_{10} = 2000, \;\; x_{20} = x_{30} = 0. \qquad (1.36)$$

The S_1 and S_3 populations here are plotted out immediately after each R_3 reaction. The S_2 population, which is shown on a separate scale, is plotted out at a like number of *equally spaced time intervals*; this gives a more typical picture of the S_2 population than plotting it *immediately* after each R_3 reaction because R_3 reactions are more likely to occur when the S_2 population is larger.

For the parameter values (1.36), condition (1.35) is satisfied by 4 orders of magnitude initially, and even more so as the total population of S_1 and S_2 declines; therefore, this reaction set should be amenable to simulation using the Slow-Scale SSA. Figure 1.4 (b) shows the results of such a simulation, plotted after each R_3 reaction. We note that all the species trajectories in this *approximate* ssSSA run agree very well with those in the *exact* SSA run of Figure 1.4 (a); even the behavior of the sparsely populated species S_2 is accurately replicated by the ssSSA. But whereas the SSA run in Figure 1.4 (a) had to simulate over 23 million reactions, the Slow-Scale SSA run in Figure 1.4 (b) simulated only 587 reactions, with commensurate differences in their computation times.

1.9 Concluding Remarks

In this chapter we have discussed two broad themes. The first is the "logical bridge" that connects the chemical master equation (CME) and stochastic simulation algorithm (SSA) on one side with the reaction rate equation (RRE) on the other side. Under the well-stirred (spatially homogeneous) assumption, the CME/SSA provides a mathematical description that is *exact*, *discrete*, and *stochastic*. If the system is such that the Leap Condition can be satisfied, the CME/SSA can be approximated by the Poissonian tau-leaping formula (1.6) to obtain a description that is *approximate*, discrete, and stochastic. If further the reactant populations are large enough that the Poissonian tau-leaping formula (1.6) can be approximated by the Gaussian tau-leaping formula (1.8), which in turn is equivalent to the chemical Langevin equation (CLE) (1.9), we obtain a description that is approximate, *continuous*, and stochastic. And finally, in the thermodynamic limit of an infinitely large system, the random terms in the CLE usually become negligibly small compared to the deterministic terms, and the CLE reduces to the RRE, which is approximate, continuous, and *deterministic*.

This progression – from the CME to the tau-leaping approximation to the CLE to the RRE – in which each successive level is an approximation of the preceeding level, would, along with the corresponding numerical methods at each level, give us all the tools we need to efficiently simulate spatially homogeneous systems were it not for the *multiscale nature* of most biochemical systems: Both the species populations and the rates of the various chemical reactions typically span many orders of magnitude. As a consequence in many cases the system as a whole does not fit efficiently into one level of description exclusive of all the others. The second theme of our development in this chapter has been to describe two strategies for coping with multiscale problems: *implicit* tau-leaping, and the *slow-scale* SSA. Much more remains to be done on the problem of multiscale.

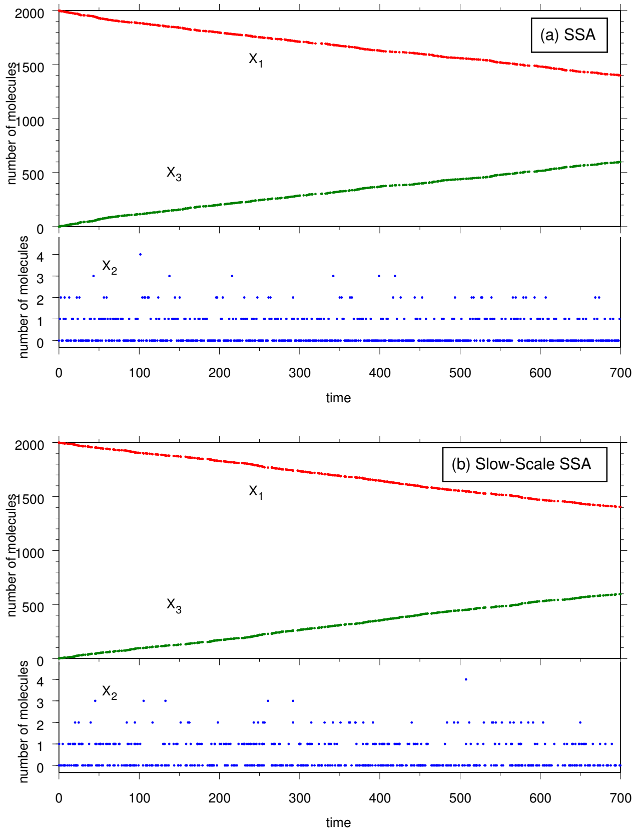

Figure 1.4: (a) Two simulations of reactions (1.29) using the parameter values (1.36). (a) shows an exact SSA run in which the populations are plotted essentially after each R_3 reaction (see text for details). Over 23 million reactions make up this run, the overwhelming majority of which are R_1 and R_2 reactions. (b) shows an approximate ssSSA run in which only R_3 reactions, which totaled 587, were directly simulated, and the populations are plotted after each of those. The ssSSA simulation ran over 1000 times faster than the SSA simulation.

Bibliography

Arkin, A., Ross, J. & McAdams, H. (Aug 1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *e. coli* cells. Genetics 149, 1633–1648.

Ascher, U. M. & Petzold, L. R. (1998). Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM.

Cao, Y., Gillespie, D. T. & Petzold, L. R. (2005). The slow-scale stochastic simulation algorithm. J. Chem. Phys. 122, 014116.

Cao, Y., Li, H. & Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. J. Chem. Phys. 121, 4059–4067.

Elowitz, M. B., Levine, A. J., Siggia, E. D. & Swain, P. S. (16 Aug 2002). Stochastic gene expression in a singel cell. Science 297, 1183–1186.

Fedoroff, N. & Fontana, W. (16 Aug 2002). Small numbers of big molecules. Science 297, 1129–1131.

Gibson, M. A. & Bruck, J. (2000). Exact stochastic simulation of chemical systems with many species and many channels. J. Phys. Chem. 105, 1876–1889.

Gillespie, D. & Petzold, L. (2003). Improved leap-size selection for accelerated stochastic simulation. J. Chem. Phys. 119, 8229–8234.

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Comp. Phys. 22, 403–434.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81, 2340–2361.

Gillespie, D. T. (1992a). A rigorous derivation of the chemical master equation. Physica A 188, 404–425.

Gillespie, D. T. (1992b). Markov Processes: An Introduction for Physical Scientists. Academic Press.

Gillespie, D. T. (2000). The chemical Langevin equation. J. Chem. Phys. 113, 297–306.

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115, 1716–1733.

Gillespie, D. T. (2002). The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction. J. Phys. Chem. A 106, 5063–5071.

McAdams, H. & Arkin, A. (1997). Stochastic mechanisms in gene expression. Proc. Natl. Acad. Sci. USA 94, 814–819.

McAdams, H. & Arkin, A. (Feb 1999). It's a noisy business! Trends in Genetics 15, 65–69.

McQuarrie, D. (1967). Stochastic approach to chemical kinetics. J. Applied Probability 4, 413–478.

Press, W., Flannery, B., Teukolsky, S. & Vetterling, W. (1986). Numerical Recipes: The Art of Scientific Computing. Cambridge University Press.

Rathinam, M., Cao, Y., Petzold, L. & Gillespie, D. (2003). Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. J. Chem. Phys. 119, 12784–12794.

Rathinam, M., Cao, Y., Petzold, L. & Gillespie, D. (2005). Consistency and stability of tau-leaping schemes for chemical reaction systems. SIAM Multiscale Modeling & Simulation . submitted.