Approximate accelerated stochastic simulation of chemically reacting systems

Daniel T. Gillespie

Citation: J. Chem. Phys. **115**, 1716 (2001); doi: 10.1063/1.1378322
View online: http://dx.doi.org/10.1063/1.1378322
View Table of Contents: http://jcp.aip.org/resource/1/JCPSA6/v115/i4
Published by the American Institute of Physics.

Approximate accelerated stochastic simulation of chemically reacting systems

Daniel T. Gillespie[a)]
Research Department, Code 4T4100D, Naval Air Warfare Center, China Lake, California 93555

(Received 29 December 2000; accepted 19 April 2001)

The stochastic simulation algorithm (SSA) is an essentially exact procedure for numerically simulating the time evolution of a well-stirred chemically reacting system. Despite recent major improvements in the efficiency of the SSA, its drawback remains the great amount of computer time that is often required to simulate a desired amount of system time. Presented here is the "τ-leap" method, an approximate procedure that in some circumstances can produce significant gains in simulation speed with acceptable losses in accuracy. Some primitive strategies for control parameter selection and error mitigation for the τ-leap method are described, and simulation results for two simple model systems are exhibited. With further refinement, the τ-leap method should provide a viable way of segueing from the exact SSA to the approximate chemical Langevin equation, and thence to the conventional deterministic reaction rate equation, as the system size becomes larger.

I. INTRODUCTION

The stochastic simulation algorithm (SSA) allows one to numerically simulate the time evolution of a well-stirred chemically reacting system in a way that takes proper account of the randomness that is inherent in such a system.[1,2] The SSA is exact in the sense that it is rigorously based on the same microphysical premise that underlies the chemical master equation (CME);[3,4] thus, a history or "realization" produced by the SSA gives a more realistic representation of the system's evolution than would a history inferred from the conventional deterministic reaction rate equation (RRE). The RRE can be particularly misleading if the molecular population of some critical reactant species becomes so small that microscopic fluctuations can conspire with reaction channel feedback loops to produce macroscopic effects. It has been shown that this can happen with dramatic consequences in the genetic/enzymatic reactions that go on inside a living cell.[5,6]

Two mathematically equivalent recipes for implementing the SSA were originally proposed.[1] Dubbed the "Direct method" and the "First Reaction method," both are exact and straightforward to program. Since the Direct method is usually more efficient, it is usually the method employed. But the Achilles' heel of either method has always been computing speed: The computer times required to simulate reasonable system times tend to be prohibitively long if the molecular populations of at least *some* of the reactant species are very large, and even in cellular systems that is nearly always the case.

Recently, substantial improvements have been made in stochastic simulation methodology. Lukkien *et al.*[7] have improved the Direct method for the special but important case of surface reactions. And Gibson and Bruck[8] have transformed the First Reaction method into a clever new scheme called the Next Reaction method; although more challenging to program, it is significantly faster than even the Direct method when many species and many reaction channels are involved. At present, the Next Reaction method appears to be the most computationally efficient way to make *exact* stochastic simulations of complex volumetric chemical systems. But modelers, especially of cellular systems,[9] are increasingly feeling the need for even faster methods. It therefore seems prudent to ask if major gains in simulation speed can be obtained by making minor sacrifices in simulation accuracy. That is the question that will be addressed in this paper.

We shall begin in Sec. II by establishing our notation and briefly reviewing the chemical master equation, the stochastic simulation algorithm, the chemical Langevin equation, and the reaction rate equation. In Sec. III we shall consider what kinds or degrees of simulation detail we might be willing to sacrifice in return for greater simulation speed; these considerations will lead us to propose an *approximate* acceleration procedure called the *τ-leap method*. In Sec. IV we shall show how the τ-leap method simplifies, given sufficiently large molecular population levels, to a *Langevin method* that is equivalent to the chemical Langevin equation; the Langevin method in turn usually reduces, in the limit of infinitely large molecular populations, to an updating algorithm that is equivalent to the deterministic reaction rate equation. In Sec. V we shall present a simple strategy for choosing appropriate values of the parameters that govern the τ-leap method—a very modest first step toward a robust optimal control strategy. In Sec. VI we shall describe a refinement that in at least some cases will reduce the errors in τ leaping. In Sec. VII we shall demonstrate τ leaping on two simple model systems. In Sec. VIII we shall briefly describe an alternate but essentially equivalent leaping strategy called *k_α-leaping*. Finally, in Sec. IX, we shall offer some tentative conclusions.

II. STOCHASTIC CHEMICAL KINETICS

We shall be concerned here a well-stirred mixture of $N \geqslant 1$ molecular species $\{S_1,\ldots,S_N\}$ that chemically interact,

a)Electronic mail: GillespieDT@mailaps.org

inside some fixed volume Ω at a constant temperature, through $M \geqslant 1$ reaction channels $\{R_1,\ldots,R_M\}$. We specify the dynamical state of this system by $\mathbf{X}(t) \equiv (X_1(t),\ldots,X_N(t))$, where

$$X_i(t) \equiv \text{the number of } S_i \text{ molecules in the system at time } t \quad (i=1,\ldots,N). \tag{1}$$

Our goal will be to describe the evolution of $\mathbf{X}(t)$ from some given initial state $\mathbf{X}(t_0)=\mathbf{x}_0$. (All boldface vectors in this paper are species indexed, with N components.)

The molecular populations $X_i(t)$ will actually be *random* variables, because we choose not to track the positions and velocities of all the molecules in the system. Indeed, we deliberately rely on the occurrence of many *nonreactive* molecular collisions to "stir" the system between successive *reactive* collisions. Under these conditions, it can be proved using elementary kinetic theory arguments[4] that there will exist for each reaction channel R_j a well defined function a_j, called the *propensity function* for R_j, which is such that

$$a_j(\mathbf{x})dt \equiv \text{the probability, given } \mathbf{X}(t)=\mathbf{x}, \text{ that one } R_j \text{ reaction will occur somewhere inside } \Omega \text{ in the next infinitesimal time interval } [t,t+dt) \quad (j=1,\ldots,M). \tag{2}$$

The function a_j and the *state-change vector* $\boldsymbol{\nu}_j$, whose ith component is defined by

$$\nu_{ji} \equiv \text{the change in the number of } S_i \text{ molecules produced by one } R_j \text{ reaction} \quad (j=1,\ldots,M;\ i=1\ldots,N), \tag{3}$$

together completely characterize reaction channel R_j.

The physical rationale for Eq. (2) has been detailed elsewhere,[1,4] and is briefly summarized in Ref. 10. Here it should suffice to give a couple of illustrative examples: If R_1 is the reaction $X_1+X_2 \rightarrow 2X_1$, then $a_1(\mathbf{x})=c_1x_1x_2$ and $\boldsymbol{\nu}_1=(+1,-1,0,\ldots,0)$, with the "specific reaction probability rate constant" c_1 being algebraically related to the conventional deterministic rate constant k_1 by $c_1=k_1/\Omega$. And if R_2 is the inverse of that reaction, then $a_2(\mathbf{x})=c_2x_1(x_1-1)/2$ and $\boldsymbol{\nu}_2=-\boldsymbol{\nu}_1$, with $c_2=2k_2/\Omega$. For the present, we do not adopt any specific forms for the propensity functions $\{a_1(\mathbf{x}),\ldots,a_M(\mathbf{x})\}$ and the state-change vectors $\{\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_M\}$; we simply assume that those quantities are specified, and hence that the chemically reacting system is defined. Equations (2) and (3) together then imply that $\mathbf{X}(t)$ is a jump Markov process on the N-dimensional non-negative integer lattice.

One rigorous consequence of Eqs. (2) and (3) is a time-evolution equation for the probability $P(\mathbf{x},t|\mathbf{x}_0,t_0)$ that $\mathbf{X}(t)$ will equal $\mathbf{x}$ given that $\mathbf{X}(t_0)=\mathbf{x}_0$ (for $t \geqslant t_0$). This is the *chemical master equation* (CME):[3,4]

$$\frac{\partial}{\partial t}P(\mathbf{x},t|\mathbf{x}_0,t_0)=\sum_{j=1}^{M}\left[a_j(\mathbf{x}-\boldsymbol{\nu}_j)P(\mathbf{x}-\boldsymbol{\nu}_j,t|\mathbf{x}_0,t_0)-a_j(\mathbf{x})P(\mathbf{x},t|\mathbf{x}_0,t_0)\right]. \tag{4}$$

But it is rarely possible to solve the CME, either analytically or numerically, for any but the simplest of chemical systems.

Another rigorous consequence of Eqs. (2) and (3) is the existence and form of the *next-reaction density function* $p(\tau,j|\mathbf{x},t)$.[1,4,11] By definition, $p(\tau,j|\mathbf{x},t)d\tau$ 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. It follows from Eqs. (2) and (3) that this function takes the form

$$p(\tau,j|\mathbf{x},t)=a_j(\mathbf{x})\exp(-a_0(\mathbf{x})\tau) \quad (\tau \geqslant 0;\ j=1,\ldots,M), \tag{5}$$

where

$$a_0(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x}), \tag{6}$$

and this provides the basis for the SSA. The SSA uses rigorous Monte Carlo techniques to generate random pairs (τ,j) according to the joint density function (5), then augments the time t by τ and the system state $\mathbf{x}$ by $\boldsymbol{\nu}_j$, and finally recalculates the propensity functions as necessary in order to repeat these steps until a sufficiently long time span has been simulated. The resultant trajectory constitutes an "unbiased realization" of the process $\mathbf{X}(t)$.

One way of generating random pairs (τ,j) according to the joint density function Eq. (5) is the so-called Direct method. For it, we first write p in the "conditioned" form

$$p(\tau,j|\mathbf{x},t)=p_1(\tau|\mathbf{x},t)p_2(j|\tau,\mathbf{x},t), \tag{7}$$

and we then generate τ according to p_1 and j according to p_2. It follows from Eq. (5) that the functions p_1 and p_2 are given by

$$p_1(\tau|\mathbf{x},t)=a_0(\mathbf{x})\exp(-a_0(\mathbf{x})\tau) \quad (\tau \geqslant 0), \tag{8a}$$

$$p_2(j|\tau,\mathbf{x},t)=\frac{a_j(\mathbf{x})}{a_0(\mathbf{x})} \quad (j=1,\ldots,M), \tag{8b}$$

and from this we may conclude that τ is a sample of $\mathcal{E}(a_0(\mathbf{x}))$, the exponential random variable with decay constant $a_0(\mathbf{x})$, while j is an independent sample of the integer random variable on $[1,M]$ with point probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$. The standard Monte Carlo inversion generating rule then dictates the following recipe for generating random pairs (τ,j): Draw two independent samples r_1 and r_2 of $\mathcal{U}(0,1)$, the unit-interval uniform random variable, and take

$$\tau=\frac{1}{a_0(\mathbf{x})}\ln\left(\frac{1}{r_1}\right), \tag{9a}$$

$$j=\text{the smallest integer satisfying } \sum_{j'=1}^{j} a_{j'}(\mathbf{x}) > r_2a_0(\mathbf{x}). \tag{9b}$$

Another method of generating values for τ and j is the First Reaction method. It generates a *tentative* reaction time for each reaction channel R_l according to

$$\tau_l=\frac{1}{a_l(\mathbf{x})}\ln\left(\frac{1}{r_l}\right) \quad (l=1,\ldots,M), \tag{10}$$

where $r_1,\ldots,r_M$ are M statistically independent samplings of $\mathcal{U}(0,1)$, and then takes

$$\tau=\text{the smallest of } \{\tau_1,\ldots,\tau_M\}, \tag{11a}$$

$$j = \text{the index of the smallest of } \{\tau_1,\ldots,\tau_M\}. \qquad (11b)$$

That the random pairs (τ,j) produced by this procedure are actually distributed according to the joint density function (5), and hence that this procedure is fully equivalent to the Direct method (9), is proved in Ref. 1. Since the $M-1$ unused tentative reaction times in Eq. (10) are discarded after the selections in Eqs. (11) are made, the First Reaction method is usually less efficient than the Direct method, and consequently is rarely used. But the mere fact that the First Reaction method works carries an important lesson: Whenever M reaction events with respective propensities $a_1,\ldots,a_M$ are in ''competition'' with each other, one can resolve the question of which of those events actually occurs next by imagining that each event occurs independently on its own—i.e., at the times given by Eq. (10)—and then *allowing* the occurrence of *only* that event with the *earliest* occurrence time.

A third way of generating random pairs (τ,j) according to the joint density function (5) is the recently developed Next Reaction method of Gibson and Bruck.[8] It is possible to view that method as an extension of the First Reaction method in which the *unused* reaction times (10) are suitably modified for reuse. The Next Reaction method also employs clever data storage structures for efficiently accomplishing step (11). The overall result is a procedure for stepping from one reaction to the next that is significantly faster than the Direct method for large M and N, requiring (asymptotically) only one random number per reaction event. Reference 8 describes the Next Reaction method in detail.

The Direct, First Reaction, and Next Reaction methods all produce exact realizations of $\mathbf{X}(t)$ by essentially generating random pairs (τ,j) rigorously according to the joint density function (5). The three methods can therefore be regarded as different but mathematically equivalent ways of implementing the SSA.

In addition to the foregoing *exact* consequences of Eqs. (2) and (3), we shall also require here a recently identified *approximate* consequence:[10] If the system possesses a *macroscopically infinitesimal time scale*, in the sense that during any time increment dt on that scale *all* the reaction channels fire many more times than once yet *none* of the propensity functions changes appreciably, then the jump Markov process $\mathbf{X}(t)$ can be approximated by the *continuous* Markov process defined by the standard form *chemical Langevin equation* (CLE)

$$X_i(t+dt) = X_i(t) + \sum_{j=1}^{M} \nu_{ji} a_j(\mathbf{X}(t))\,dt + \sum_{j=1}^{M} \nu_{ji} a_j^{1/2}(\mathbf{X}(t)) N_j(t) (dt)^{1/2} \quad (i=1,\ldots,N). \qquad (12)$$

Here, $N_1(t),\ldots,N_M(t)$ are M temporally uncorrelated, statistically independent normal random variables with mean 0 and variance 1, and dt is a ''macroscopically infinitesimal'' time increment in the sense just described. (Associated with this standard form Langevin equation is a white noise form Langevin equation as well as a Fokker–Planck equation,[10] but we shall not need either of those equations here.)

Finally, as discussed in Ref. 10, in the limit of infinitely large molecular populations of the reactant species, the cumulative contribution of each term in the second summation on the right hand side of Eq. (12) usually becomes vanishingly small compared to that of the correspondingly indexed term in the first summation. Therefore, in that ''thermodynamic limit,'' Eq. (12) usually reduces to

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ji} a_j(\mathbf{X}(t)) \quad (i=1,\ldots,N). \qquad (13)$$

This equation is, apart from an inconsequential scaling factor of Ω^{-1}, the well known *reaction rate equation* (RRE) of conventional chemical kinetics. As a set of coupled ordinary differential equations, it describes $\mathbf{X}(t)$ as a *continuous deterministic* process. For most macroscopic systems encountered in practice, Eq. (13) suffices. But it is important to recognize that the RRE (13) is actually a limiting approximation of the CLE (12), and the CLE in turn is an approximate consequence of the premises (2) and (3) which rigorously underlie the CME and the SSA.

III. THE τ-LEAP METHOD

As the time evolution of $\mathbf{X}(t)$ unfolds from some initial state $\mathbf{x}_0$ at some initial time t_0, let us suppose the history of the system to be recorded by marking on a time axis the successive instants $t_1,t_2,t_3,\ldots$ at which the first, second, third,... reaction events occur, and also appending to those points the indices $j_1,j_2,j_3,\ldots$ of the respective reaction channels R_j that ''fire'' at those instants. This ''history axis'' completely describes a realization of $\mathbf{X}(t)$; indeed, we could imagine it being constructed by simply monitoring the (τ,j)-generating procedure of the SSA as it dutifully steps us along from each t_n to t_{n+1}. We note that this ''stepping along the history axis'' approach of the SSA is both its strength and its weakness: The meticulous construction of every individual reaction event gives us a complete and detailed history of $\mathbf{X}(t)$, but that construction is usually a very time-consuming task for systems of practical interest, because of the enormous number of reaction events that take place in real systems.

It is probably so that much of the detail on the history axis of the system is neither useful nor necessary. In particular, it is conceivable that the system's history axis could be divided into a set of contiguous subintervals in such a way that, if we could determine only how many times each reaction channel fired in each subinterval, we could forego knowing the precise instants at which those firings took place. Such a circumstance would allow us to *leap* along the system's history axis from one *subinterval* to the next, instead of stepping along from one reaction event to the next. And if enough of the subintervals contained many individual reaction events, the gain in simulation speed could be substantial, provided of course that each subinterval leap could be done expeditiously.

To get a mathematical handle on such a leaping strategy, consider the probability function Q, defined by

$$Q(k_1,\ldots,k_M|\tau;\mathbf{x},t) \equiv \text{the probability, given } \mathbf{X}(t)=\mathbf{x}, \text{ that in the time interval } [t,t+\tau) \text{ exactly } k_j \text{ firings of reaction channel } R_j \text{ will occur, for each } j=1,\ldots,M. \tag{14}$$

Q is evidently the joint probability density function of the M integer random variables

$$K_j(\tau;\mathbf{x},t) \equiv \text{the number of times, given } \mathbf{X}(t)=\mathbf{x}, \text{ that reaction channel } R_j \text{ will fire in the time interval } [t,t+\tau) \ (j=1,\ldots,M). \tag{15}$$

To determine $Q(k_1,\ldots,k_M|\tau;\mathbf{x},t)$ for *arbitrary* $\tau>0$ would be a task at least as formidable as solving the master equation (4) for $P(\mathbf{x},t|\mathbf{x}_0,t_0)$ for arbitrary $t>t_0$. But we can get a simple *approximate* form for $Q(k_1,\ldots,k_M|\tau;\mathbf{x},t)$ if we impose the following condition on τ.

Leap Condition: Require τ to be small enough that the change in the state during $[t,t+\tau]$ will be so slight that no propensity function will suffer an appreciable (i.e., macroscopically noninfinitesimal) change in its value.

Assuming this condition is satisfied, then during the entire interval $[t,t+\tau)$ contemplated in Eqs. (14) and (15), the propensity function for each channel R_j will remain *essentially constant* at the value $a_j(\mathbf{x})$. This means that $a_j(\mathbf{x})dt$ will give the probability that channel R_j will fire during *any* infinitesimal interval dt inside $[t,t+\tau)$, regardless of what the other reaction channels are doing. In that case, as we remind ourselves in the Appendix, $K_j(\tau;\mathbf{x},t)$ will be the *Poisson* random variable

$$K_j(\tau;\mathbf{x},t)=\mathcal{P}(a_j(\mathbf{x}),\tau) \quad (j=1,\ldots,M). \tag{16}$$

And since these M random variables $K_1(\tau;\mathbf{x},t),\ldots,K_M(\tau;\mathbf{x},t)$ will be statistically independent, the joint density function (14) will simply be the product of the density functions of the individual Poisson random variables:

$$Q(k_1,\ldots,k_M|\tau;\mathbf{x},t)=\prod_{j=1}^{M} P_{\mathcal{P}}(k_j;a_j(\mathbf{x}),\tau). \tag{17}$$

As noted in the Appendix, reliable numerical techniques exist for generating sample values of the Poisson random variable $\mathcal{P}(a,\tau)$. So, provided the Leap Condition is satisfied, we can leap down the history axis of the system by the amount τ from state $\mathbf{x}$ at time t by proceeding as follows. First we generate for each reaction channel R_j a sample value k_j of the Poisson random variable $\mathcal{P}(a_j(\mathbf{x}),\tau)$; k_j will be the number of times reaction channel R_j fires in $[t,t+\tau)$. Since each firing of R_j changes the S_i population by ν_{ji} molecules, the net change in the state of the system in $[t,t+\tau)$ will be

$$\boldsymbol{\lambda}=\sum_{j=1}^{M} k_j\boldsymbol{\nu}_j. \tag{18}$$

Thus we arrive at the following procedure.

Basic τ-Leap Method: Choose a value for τ that satisfies the Leap Condition; i.e., a temporal leap by τ will result in a state change $\boldsymbol{\lambda}$ which is such that, for every reaction channel R_j, $|a_j(\mathbf{x}+\boldsymbol{\lambda})-a_j(\mathbf{x})|$ is "effectively infinitesimal." Generate for each $j=1,\ldots,M$ a sample value k_j of the Poisson random variable $\mathcal{P}(a_j(\mathbf{x}),\tau)$, and compute $\boldsymbol{\lambda}=\Sigma_j k_j\boldsymbol{\nu}_j$. Finally, effect the leap by replacing t by $t+\tau$ and $\mathbf{x}$ by $\mathbf{x}+\boldsymbol{\lambda}$.

The accuracy of τ leaping will depend upon how well the Leap Condition is satisfied. In the trivial case where none of the propensity functions depend on $\mathbf{x}$, the Leap Condition would be satisfied exactly for any τ, and τ leaping would be exact. Much more commonly, the propensity functions will depend linearly or quadratically on the molecular populations, and τ leaping will not be exact. But since each reaction event changes the reactant populations by no more than one or two molecules, then *if the reactant molecule populations are very large*, it will take a very large number of reaction events to change the propensity functions "noticeably." So, if we have large molecular populations, and the exact SSA is therefore slow, we should be able to satisfy the Leap condition with a choice for τ that allows *many* reaction events to occur in $[t,t+\tau]$; that of course will result in a "leap" down the history axis of the system that is much longer than the single reaction "step" of the exact SSA.

If, on the other hand, satisfying the Leap Condition turns out to require τ to be so small that only a very few reactions are leaped over, then it would be faster to forego leaping and use the exact SSA. For example, if we were to take τ to be the relatively small value $1/a_0(\mathbf{x})$ [see Eq. (6)], then the resultant leap would be the *expected* size of the next time step in the SSA [see Eqs. (8a) and (9a)], and very likely one of the generated k_js would be 1 and all the others would be 0. Still smaller choices for τ would result in leaps in which *all* of the k_js would likely be 0, a circumstance that clearly would gain us nothing. But, although it would be *inefficient* to use the τ-leap method when τ is less than or equal to $1/a_0(\mathbf{x})$, it would *not* be incorrect; indeed, leaps with no more than one reaction event should be virtually exact. We may therefore expect that as τ decreases to $1/a_0(\mathbf{x})$ or smaller, the results produced by τ leaping will segue smoothly to results that would be produced by the exact SSA.

In order to successfully employ τ leaping in a practical situation, we obviously need some way of quickly determining the *largest* value of τ that is compatible with the Leap Condition. We shall make an initial assault on this critical problem in Sec. V, although it must be stated in advance that a completely satisfactory solution seems not yet to be in hand. But before we address that problem, let us show how the τ-leap method segues, in the limit of *very* large reactant populations, to an even faster simulation method.

IV. THE LANGEVIN METHOD

Suppose conditions are such that, starting in state $\mathbf{x}$ at time t, we can leap down the history axis of the system by an amount τ that spans a *very large* number of firings of *every* reaction channel and *still* satisfy the Leap Condition. More quantitatively, suppose the *expected* number of firings of each reaction channel in $[t,t+\tau)$ obeys (see Appendix)

$$\langle \mathcal{P}(a_j(\mathbf{x}),\tau)\rangle = a_j(\mathbf{x})\tau \gg 1 \quad (\forall j = 1,...,M), \tag{19}$$

yet all those firings induce only miniscule changes in the values of all the propensity functions. Then the following simplification can be made in the τ-leap method: Since the Poisson random variable $\mathcal{P}(a,t)$ will, when $at \gg 1$, be well approximated by a *normal* random variable with the same mean and variance [see Eq. (A5)], then the number of firings of channel R_j in $[t,t+\tau)$ can be approximated by

$$\begin{aligned} K_j(\tau;\mathbf{x},t) &= \mathcal{P}_j(a_j(\mathbf{x}),\tau) \\ &\approx \mathcal{N}_j(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau) \\ K_j(\tau;\mathbf{x},t) &= a_j(\mathbf{x})\tau + (a_j(\mathbf{x})\tau)^{1/2}\mathcal{N}_j(0,1) \quad (j=1,...,M). \end{aligned} \tag{20}$$

Here, the first line follows from the Leap Condition, with the subscript j on $\mathcal{P}$ reminding us that a different, statistically independent Poisson random variable is used for each reaction channel; the second line follows by virtue of the approximation induced by condition (19); and the last line follows from the normal random variable property $\mathcal{N}(m,\sigma^2) = m + \sigma\mathcal{N}(0,1)$. Computationally, the third line in Eq. (20) is an improvement over the first line (albeit an approximate one), because normal random numbers can be generated more quickly than Poisson random numbers. The result (20) provides the basis for a special case of the τ-leap method that we shall call the Langevin method.

Langevin Method: Suppose it is possible to choose τ so that (i) the Leap Condition is satisfied, *and* (ii) $\tau \gg \mathrm{Max}_j\{1/a_j(\mathbf{x})\}$. Then for each $j=1,...,M$, generate a sample value n_j of the "unit normal" random variable $\mathcal{N}(0,1)$ and put $k_j = a_j(\mathbf{x})\tau + (a_j(\mathbf{x})\tau)^{1/2} n_j$. Finally, compute $\boldsymbol{\lambda} = \Sigma_j k_j \boldsymbol{\nu}_j$, and effect the leap by replacing t by $t+\tau$ and $\mathbf{x}$ by $\mathbf{x}+\boldsymbol{\lambda}$.

We call this the Langevin method because it is entirely equivalent to the chemical Langevin equation (12), with dt replaced by τ. To see this, observe that the Langevin method computes the change $\lambda_i \equiv X_i(t+\tau) - X_i(t)$ in the S_i population as

$$\begin{aligned} \lambda_j &= \sum_{j=1}^{M} k_j \nu_{ji} = \sum_{j=1}^{M} [a_j(\mathbf{x})\tau + (a_j(\mathbf{x})\tau)^{1/2} n_j]\nu_{ji} \\ &= \sum_{j=1}^{M} \nu_{ji} a_j(\mathbf{x})\tau + \sum_{j=1}^{M} \nu_{ji} a_j^{1/2}(\mathbf{x}) n_j \tau^{1/2}, \end{aligned}$$

and this is precisely the updating recipe dictated by CLE (12). In fact, the derivation of Eq. (12)[10] requires that both the Leap Condition and condition (19) be satisfied, and the lines leading to Eq. (20) essentially trace the logic of that derivation.

The Langevin method will be fast not only because the unit normal random numbers n_j are relatively easy to generate, but also because condition (19) implies that each k_j will be large compared to 1, and hence that the leap will encompass many reaction events. But it should be clearly understood that, in order to validly use the Langevin method, *both* the Leap Condition and condition (19) need to be satisfied.

If the τ-leap method is used when all M conditions (19) happen to be satisfied, the resultant leap will be entirely equivalent to a Langevin method leap, since all the *Poisson* random numbers used in the τ leap will then be practically indistinguishable from *normal* random numbers. This means that the τ-leap method smoothly transitions to the Langevin method as conditions (19) become satisfied. The Langevin method in turn smoothly transitions to the deterministic RRE (13) whenever the inequalities (19) become *strongly* satisfied. This is because the limiting case $a_j(\mathbf{x})\tau \to \infty$ of condition (19) implies that, in the Langevin method formula $k_j = a_j(\mathbf{x})\tau + (a_j(\mathbf{x})\tau)^{1/2} n_j$, the second term becomes negligibly small compared to the first term; hence, the increment $\lambda_i \equiv X_i(t+\tau) - X_i(t)$ in the S_i population becomes, in the limit $a_j(\mathbf{x})\tau \to \infty$,

$$\lambda_i = \sum_{j=1}^{M} k_j \nu_{ji} = \sum_{j=1}^{M} [a_j(\mathbf{x})\tau]\nu_{ji} = \sum_{j=1}^{M} \nu_{ji} a_j(\mathbf{x})\tau,$$

and this is nothing more than the Euler formula for numerically solving the RRE (13). The Langevin method therefore plays the important conceptual role of showing how the stochastic simulation methods (the exact SSA and its approximating τ-leap method) are related to the deterministic RRE of traditional chemical kinetics.

V. A PROCEDURE FOR SELECTING TAU

To successfully apply τ leaping, we obviously need a procedure for quickly determining the largest value of τ that is compatible with the Leap Condition. One way to do that might be to make a postleap check of the differences $|a_j(\mathbf{x}+\boldsymbol{\lambda}) - a_j(\mathbf{x})|$ for each j from 1 to M, and then try either a smaller value of τ if any of those differences is too large, or a larger value of τ if larger differences could be tolerated. But that procedure would probably be time consuming; moreover, it might engender a bias against infrequent but nonetheless legitimate large fluctuations.

A *preleap* check on the acceptability of τ might be carried out as follows: Since the mean or expected value of k_j will be $\langle \mathcal{P}(a_j(\mathbf{x}),\tau)\rangle = a_j(\mathbf{x})\tau$, then the *expected* net change in state in $[t,t+\tau)$ will be

$$\overline{\boldsymbol{\lambda}} \equiv \overline{\boldsymbol{\lambda}}(\mathbf{x},\tau) = \sum_{j=1}^{M} [a_j(\mathbf{x})\tau]\boldsymbol{\nu}_j = \tau\boldsymbol{\xi}(\mathbf{x}), \tag{21}$$

where we have defined

$$\boldsymbol{\xi}(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x})\boldsymbol{\nu}_j. \tag{22}$$

$\boldsymbol{\xi}(\mathbf{x})$ can be interpreted as the mean or expected state change in a *unit* of time. We observe that $\overline{\boldsymbol{\lambda}}$ can be calculated fairly easily for any τ. So, let us simply require that the *expected* changes in the propensity functions in time τ, namely the

differences $|a_j(\mathbf{x}+\overline{\boldsymbol{\lambda}})-a_j(\mathbf{x})|$, be bounded by some specified fraction ϵ $(0<\epsilon<1)$ of the *sum* of all the propensity functions:

$$|a_j(\mathbf{x}+\overline{\boldsymbol{\lambda}})-a_j(\mathbf{x})|\leqslant \epsilon a_0(\mathbf{x}) \quad (j=1,...,M). \tag{23}$$

We can estimate the difference on the left side of Eq. (23) by a first-order Taylor expansion:

$$a_j(\mathbf{x}+\overline{\boldsymbol{\lambda}})-a_j(\mathbf{x})\approx \overline{\boldsymbol{\lambda}}\cdot\nabla a_j(\mathbf{x})=\sum_{i=1}^{N}\tau\xi_i(\mathbf{x})\frac{\partial}{\partial x_i}a_j(\mathbf{x}).$$

So, defining

$$b_{ji}(\mathbf{x})\equiv\frac{\partial a_j(\mathbf{x})}{\partial x_i} \quad (j=1,...,M;i=1,...,N), \tag{24}$$

the requirement Eq. (23) becomes, to a reasonably good approximation,

$$\tau\left|\sum_{i=1}^{N}\xi_i(\mathbf{x})b_{ji}(\mathbf{x})\right|\leqslant\epsilon a_0(\mathbf{x}) \quad (j=1,...,M). \tag{25}$$

The largest value of τ that is consistent with this condition, and hence the optimal choice for τ given the value chosen for ϵ, is

$$\tau=\min_{j\in[1,M]}\left\{\epsilon a_0(\mathbf{x})\Big/\left|\sum_{i=1}^{N}\xi_i(\mathbf{x})b_{ji}(\mathbf{x})\right|\right\}. \tag{26a}$$

There will clearly be some computational overhead in selecting τ according to Eq. (26a): We shall have to evaluate the N functions $\xi_i(\mathbf{x})$ in Eq. (22) and the MN functions $b_{ji}(\mathbf{x})$ in Eq. (24) (but note that the derivatives in the latter may be computed beforehand and they will usually be quite simple), and we shall then have to compute and find the smallest of the M ratios on the right side of Eq. (26a). Moreover, the τ value thus found should *not* be used unconditionally; because, if τ turns out to be less than a few multiples of the time required for the SSA to make an *exact* time step, then it would be better to use the SSA instead. So, since the expected time to the next reaction in the SSA is $1/a_0(\mathbf{x})$, we supplement the τ selection rule (26a) with the proviso

$$\text{Use exact SSA instead if } \tau\leqslant\frac{2}{a_0(\mathbf{x})}, \tag{26b}$$

where the numerator 2 could arguably be replaced by anything between 1 and 10.

The foregoing τ-selection procedure should be viewed as only a first step towards a more robust control strategy for optimally using the τ-leap method in conjunction with the SSA and the Langevin method. Note that the segue from τ leaping to the Langevin method will occur almost invisibly if the computer routine that generates samples of the Poisson random variable $\mathcal{P}(a_j,\tau)$ is written to return samples of the *normal* random variable $\mathcal{N}(a_j\tau,a_j\tau)$ whenever $a_j\tau$ is "sufficiently large." When normal random variables are being returned for *all* the reaction channels R_j, the τ-leap method will have become the Langevin method.

VI. THE ESTIMATED-MIDPOINT TECHNIQUE

The Leap Condition requires that none of the propensity functions changes "appreciably" in the course of a leap. But taking leaps that are large enough to produce a faster simulation than the SSA will practically always result in *some* changes in the propensity functions, and those changes will inevitably give rise to computational errors.

A similar difficulty arises when numerically solving an ordinary differential equation of the form $dX/dt=f(X)$ by the simple Euler method, where a leap down the t axis by Δt according to $X(t+\Delta t)=X(t)+f(X(t))\Delta t$ will produce errors whenever the function f changes during that Δt increment. As is well known, one way to reduce those errors is to use the estimated-midpoint (or second-order Runge–Kutta) procedure instead: With $\Delta_1 X\equiv f(X(t))\Delta t$, take $X(t+\Delta t)=X(t)+f(X(t)+\frac{1}{2}\Delta_1 X)\Delta t$; in other words, use the simple Euler method to estimate the "midpoint" value of X during $[t,t+\Delta t)$, and then calculate the actual increment in X by evaluating the slope function f at that estimated midpoint.

In an attempt to adapt this estimated-midpoint strategy to the τ-leap method, let us take as the analogue of the simple Euler increment $\Delta_1 X$ the *expected* state change $\overline{\boldsymbol{\lambda}}$ in Eq. (21). More precisely, with $[z]$ denoting the largest integer in z, let us take $\mathbf{x}+[\overline{\boldsymbol{\lambda}}/2]$ to be the "estimated midpoint state" during the leap, and then let us generate the Poisson random numbers k_j for the leap using density functions evaluated at $\mathbf{x}+[\overline{\boldsymbol{\lambda}}/2]$ instead of at $\mathbf{x}$. So, our estimated-midpoint method for τ leaping from state $\mathbf{x}$ at time t will be as follows.

Estimated-Midpoint τ-Leap Method: For the selected leaping time τ (which satisfies the Leap Condition), compute the *expected* state change $\overline{\boldsymbol{\lambda}}=\tau\Sigma_j a_j(\mathbf{x})\boldsymbol{\nu}_j$ during $[t,t+\tau)$. Then, with $\mathbf{x}'\equiv\mathbf{x}+[\overline{\boldsymbol{\lambda}}/2]$, generate for each $j=1,...,M$ a sample value k_j of the Poisson random variable $\mathcal{P}(a_j(\mathbf{x}'),\tau)$. Compute the actual state change, $\boldsymbol{\lambda}=\Sigma_j k_j\boldsymbol{\nu}_j$, and effect the leap by replacing t by $t+\tau$ and $\mathbf{x}$ by $\mathbf{x}+\boldsymbol{\lambda}$.

To examine the legitimacy of this strategy, let us consider its effect on a reaction set that is simple enough to solve exactly in the stochastic formalism. The $N=M=1$ isomerization reaction

$$X\overset{c}{\rightarrow}Y \tag{27}$$

has propensity function $a(x)=cx$ and state change vector $\nu=-1$. The solution to its CME can be shown to be

$$P(x-k,t+\tau|x,t)=\frac{x!}{k!(x-k)!}[e^{-c\tau}]^{x-k}[1-e^{-c\tau}]^k \quad (0\leqslant k\leqslant x;\tau\geqslant 0). \tag{28}$$

The correctness of this formula can be seen by noting that the second factor is the probability that a specified group of $x-k$ molecules will *not* isomerize in $[t,t+\tau)$, the third factor is the probability that a specified group of k molecules *will* isomerize in $[t,t+\tau)$, and the first factor is the number of distinct ways of dividing x molecules into two groups of k and $x-k$ molecules.

To effect a leap $\lambda=k\nu=-k$ for this reaction from state x at time t using the *plain* τ-leap method, we would first choose a leaping time τ, and then obtain the number k of reactions (27) that occur in $[t,t+\tau)$ by sampling the Poisson density function

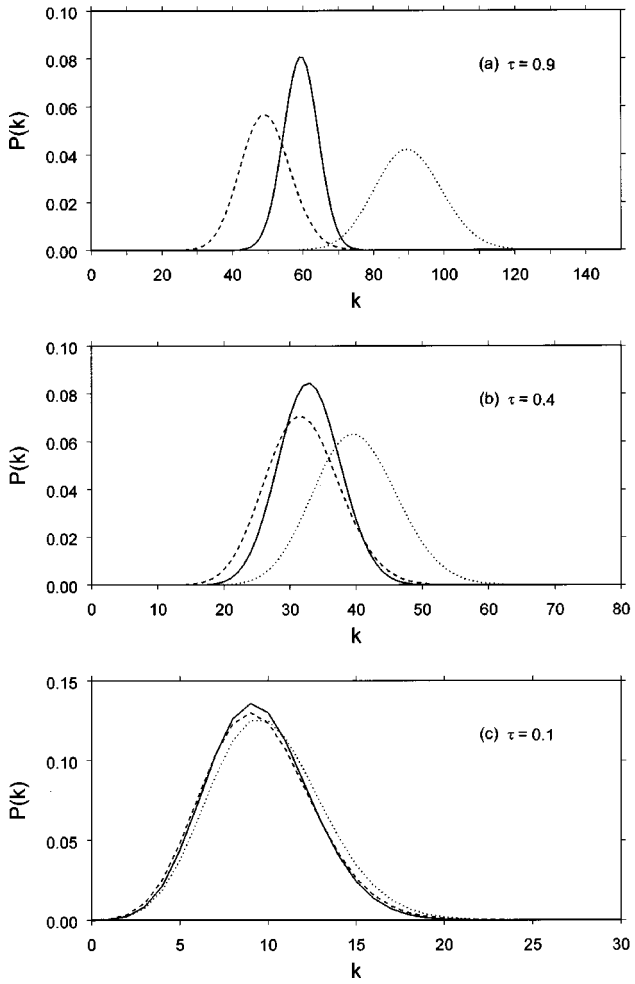

FIG. 1. Probability density functions for the number k of isomerizations (27) occurring in a given time τ, with $c=1$ and $x=100$. (a) has $\tau=0.9$, (b) has $\tau=0.4$, and (c) has $\tau=0.1$. In each case, the *solid* curve is the exact density function (28), the *dotted* curve is the density function (29a) predicted by the plain τ-leap method, and the *dashed* curve is the density function (29b) predicted by the estimated-midpoint τ-leap method.

$$P_{\mathcal{P}}(k;cx,\tau)=\frac{e^{-cx\tau}(cx\tau)^k}{k!}\quad(k=0,...,\infty). \tag{29a}$$

To leap using the *estimated-midpoint* τ-leap method, we would first compute for the chosen τ value the expected state change in $[t,t+\tau)$, $\overline{\lambda}=\tau a(x)\nu=-\tau cx$. Then, with $x'\equiv x+[\frac{1}{2}\overline{\lambda}]=x-[\frac{1}{2}\tau cx]$, we would obtain k by sampling the Poisson density function

$$P_{\mathcal{P}}(k;cx',\tau)=\frac{e^{-cx'\tau}(cx'\tau)^k}{k!}\quad(k=0,...,\infty). \tag{29b}$$

The *exact* way to choose k would be to sample the binomial density function (28), since that function is precisely the probability that exactly k isomerizations will occur in $[t,t+\tau)$.

In Fig. 1 we compare the exact k density function (28) with the proposed approximations (29a) and (29b) for $c=1$, $x=100$, and three different values of τ. Figure 1(a) has $\tau=0.9$, a time leap that is "large" since the average time to isomerization for any individual X molecule is $c^{-1}=1$; indeed, the exact (solid) curve in Fig. 1(a) shows that between 40 and 80 of the 100 X molecules should isomerize in that time leap. In this case, the plain τ-leap (dotted) curve significantly overestimates k, while the estimated-midpoint τ-leap (dashed) curve, to a much lesser extent, underestimates k. Note also that plain τ leaping with $\tau=0.9$ predicts a substantial probability of producing a k value greater than 100, which of course is physically unrealistic. For $\tau=0.4$ [Fig. 1(b)], the estimated-midpoint k distribution provides an arguably acceptable approximation to the true k distribution. The plain k distribution is still too high, although it no longer predicts k values that are unphysically large. For $\tau=0.1$ [Fig. 1(c)], the estimated-midpoint k distribution matches the exact k distribution extremely well, and the plain k distribution has finally become arguably acceptable.

Figure 1 suggests that, at least for the simple isomerization reaction, the estimated-midpoint technique allows a roughly fourfold increase in the leap size for the same degree of accuracy. But it remains to be seen how effective this technique will be for other kinds of reactions. We also see from Fig. 1 that the estimated-midpoint technique overcorrects the plain τ-leap k distribution, at least insofar as peak placement is concerned, so it might be better to take $\mathbf{x}'=\mathbf{x}+f\overline{\boldsymbol{\lambda}}$, where f is a bit *less* than 1/2; however, we shall not pursue that ad hoc refinement at this stage.

The estimated-midpoint technique should also be applicable to the Langevin method. That would give us the following procedure.

Estimated-Midpoint Langevin Method: Choosing τ so that (i) the Leap Condition is satisfied, *and* (ii) $\tau\gg\mathrm{Max}_j\{1/a_j(\mathbf{x})\}$, compute $\overline{\boldsymbol{\lambda}}=\tau\Sigma_j a_j(\mathbf{x})\boldsymbol{\nu}_j$. Then, with $\mathbf{x}'\equiv\mathbf{x}+[\overline{\boldsymbol{\lambda}}/2]$, for each $j=1,...,M$ generate a sample value n_j of the "unit normal" random variable $\mathcal{N}(0,1)$ and put $k_j=a_j(\mathbf{x}')\tau+(a_j(\mathbf{x}')\tau)^{1/2}n_j$. Finally, compute $\boldsymbol{\lambda}=\Sigma_j k_j\boldsymbol{\nu}_j$, and effect the leap by replacing t by $t+\tau$ and $\mathbf{x}$ by $\mathbf{x}+\boldsymbol{\lambda}$.

In the special case that condition (ii) here happens to be satisfied so strongly that the second term in the above formula for k_j is for every j negligibly small compared to the first term, this method reduces to the second-order Runge–Kutta algorithm for the deterministic reaction rate equation (13). This plausible result must, however, be viewed in the light of a rather more disquieting one: For any Langevin equation whose diffusion functions are state dependent—as are the diffusion functions $\nu_{ij}a_j^{1/2}(\mathbf{x})$ in the chemical Langevin equation (12)—the estimated-midpoint logic produces an updating formula that is demonstrably *wrong* in the limit $\tau\to 0$.[12] But we can optimistically hope that this will not pose a problem with our estimated-midpoint Langevin method here because condition (ii) serves to keep us away from the limit $\tau\to 0$; i.e., since the CME (12) is valid *only* if dt is a *macroscopic* infinitesimal, then the limit $dt\to 0$ for that particular Langevin equation is effectively precluded.

VII. TWO ELEMENTARY EXAMPLES

As our first application of the τ-leap method we shall take the simplest of all chemical reactions, the irreversible isomerization

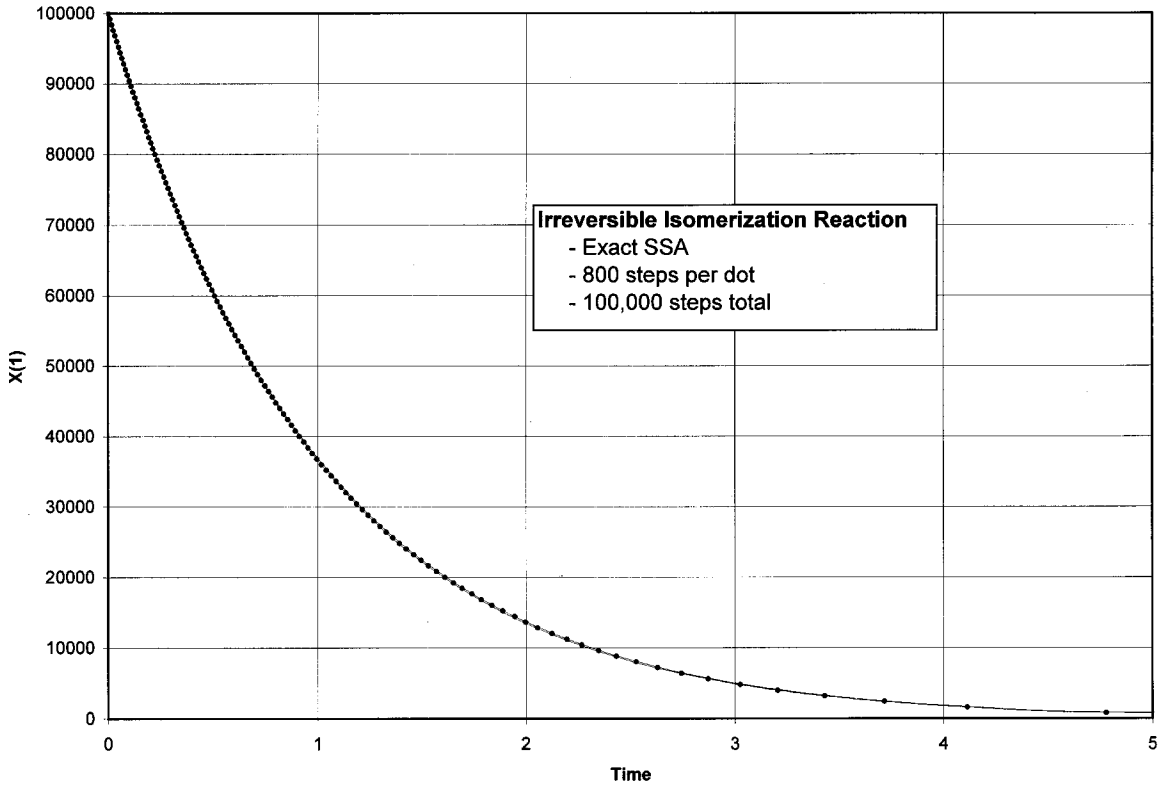

FIG. 2. Exact stochastic simulation of the simple isomerization reaction (30) with $c_1=1$. Each of the 100 000 reactions is simulated, and the state is plotted as a dot after every 800 reactions. The last reaction in this run occurred at $t=12.76$. The two solid lines show the 1 sd envelope predicted by the chemical master equation.

$$S_1 \xrightarrow{c_1} 0, \tag{30}$$

for which $N=M=1$, $a_1(\mathbf{x})=c_1x_1$, and $\boldsymbol{\nu}_1=-1$. We shall let $c_1=1$, and assume that there are 10^5 S_1 molecules at time 0. The solution $P(x_1,t|10^5,0)$ to the CME (4) for this reaction can be read off from Eq. (28).

Figure 2 shows the result of an exact stochastic simulation of reaction (30) in which (t,X_1) has been plotted after every 800 reactions. What might appear to be a single solid line in Fig. 2 is actually two lines, which demarcate the 1 sd envelope $\langle X_1(t)\rangle \pm \mathrm{sdev}\{X_1(t)\}$ as computed from the solution to the CME. Reaction (30) evidently exhibits little stochasticity on the population scale of Fig. 2; nevertheless, the exact SSA dutifully generates the precise (stochastic) instant at which each of the 10^5 S_1 molecules isomerizes. The final reaction occurs in this particular run at $t=12.76$, a value that will fluctuate considerably from run to run.

Figures 3(a)–3(c) show the results of three simulation runs of reaction (30) using the τ-leap method. For each of these runs the leap size τ was chosen using the ϵ-control strategy described in Sec. V, i.e., according to Eqs. (26a) and (26b). So each leap advances time by the amount that makes the expected fractional decrease in the propensity function equal to ϵ, except that the final few reactions are generated using the exact SSA. The plain τ-leap run in Fig. 3(a) has $\epsilon=0.03$. The dots show the state of the system after every second leap, and the entire run uses 305 leaps, as compared to the 100 000 steps in the exact run of Fig. 2. The accuracy appears to be good, although a close inspection reveals that the trajectory is slightly biased to the low side of the 1 sd envelope. Increasing the accuracy parameter ϵ by a factor of 5 gives the run in Fig. 3(b), where now the state has been plotted after every leap. The run is of course faster, but the low bias has become unacceptably large. Repeating this run using the estimated-midpoint technique yields the trajectory in Fig. 3(c). The low bias appears to have been eliminated, and the run uses only 70 leaps.

Figures 3(a) and 3(c) together suggest that the estimated-midpoint strategy allows the average leap size in a simulation of reaction (30) to be increased by a factor of roughly 4 while maintaining the same degree of accuracy, consistent with our findings in Sec. VI. Additional simulations with larger values of ϵ revealed that the slight high bias in the estimated-midpoint method can be greatly reduced by decreasing midpoint fraction from 0.5 to 0.45. But other testing should be done before adopting such an *ad hoc* refinement. For instance, we should plot histograms of X_1 at various fixed times (e.g., $t=0.5$, 2.0, and 8.0) for a thousand or so repeated runs of each of the three cases in Fig. 3, and then compare those histograms with plots of $P(x_1,t|10^5,0)$ as given by Eq. (28) to see how accurately the *dispersion* in X_1 about its mean is being replicated in these leaping simulations. For now, though, we shall simply take these initial

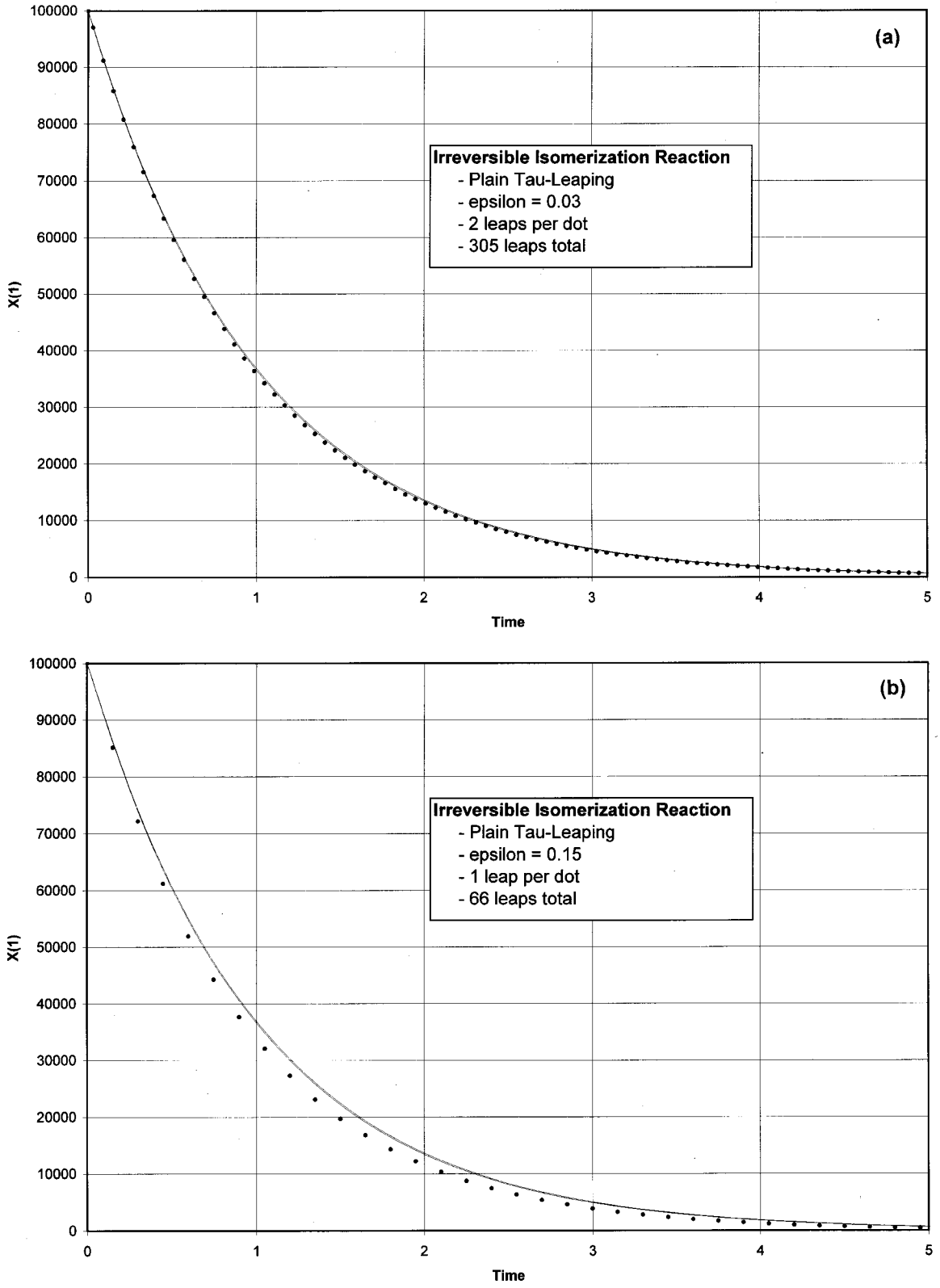

FIG. 3. Three τ-leap simulations of the reaction in Fig. 2, using in each case the control strategy (26). The solid lines show the 1 sd envelope predicted by the chemical master equation. In (a) the error control parameter $\epsilon = 0.03$, and the state is plotted after every second leap; a total of 305 leaps were needed to complete the simulation, as compared to the 100 000 steps in Fig. 2. In (b) ϵ has been increased by a factor of 5, and the state is plotted after every leap; the trajectory now falls off too rapidly. The problem is corrected in (c) by using the estimated-midpoint technique, which evidently allows an acceptable simulation to be accomplished in only 70 leaps.

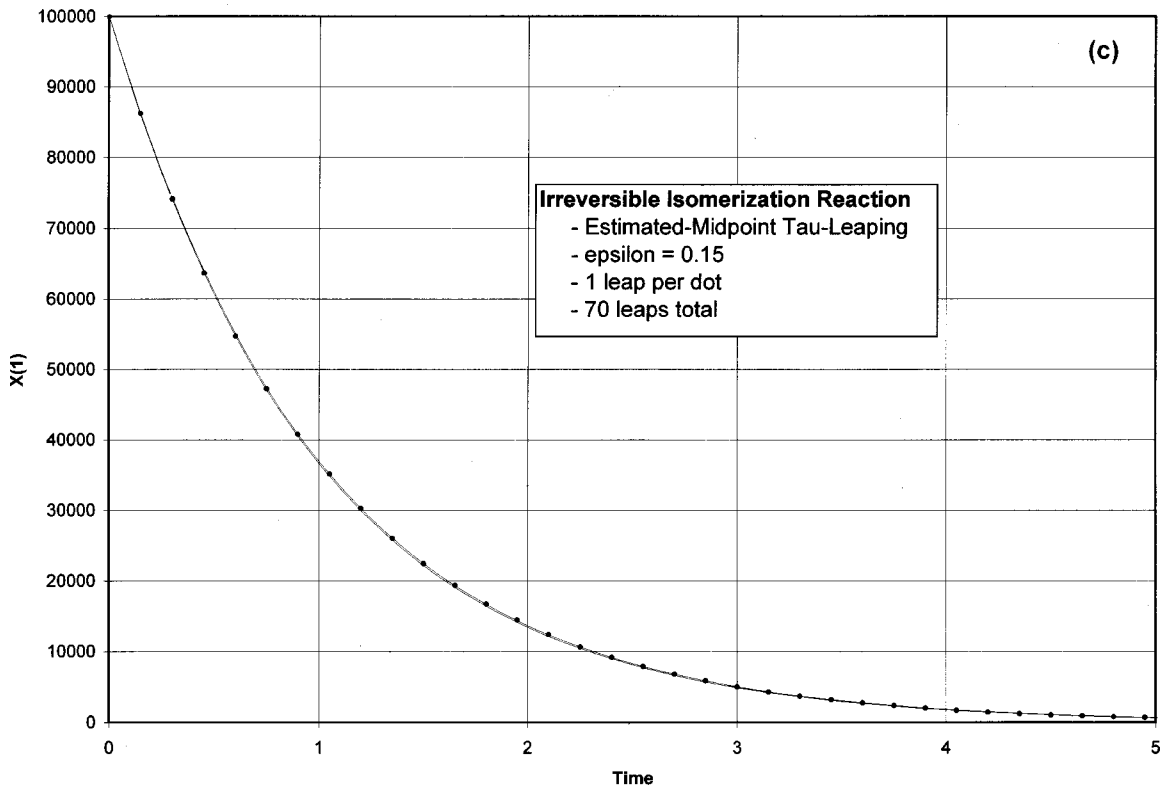

FIG. 3. (*Continued.*)

results for reaction (30) as ''encouraging,'' and move on to a slightly more complicated set of reactions.

We shall augment reaction (30) by adding two more molecular species and three more reaction channels as follows:

$$\begin{aligned} &S_1 \xrightarrow{c_1} 0 \\ &S_1 + S_1 \xrightarrow{c_2} S_2 \\ &S_2 \xrightarrow{c_3} S_1 + S_1 \\ &S_2 \xrightarrow{c_4} S_3. \end{aligned} \tag{31}$$

When c_2 and c_3 are sufficiently large, the disappearance of S_1 molecules through reaction R_1 is superimposed on a fast, reversible dimerization of the ''monomer'' S_1 into an ''unstable dimer'' S_2, which in turn can convert to a stable form S_3. We shall simulate these reactions using the rate constant values

$$c_1 = 1, \quad c_2 = 0.002, \quad c_3 = 0.5, \quad c_4 = 0.04 \tag{32a}$$

and the initial conditions

$$X_1 = 10^5, \quad X_2 = X_3 = 0. \tag{32b}$$

Figures 4(a) and 4(b) show the results of an exact stochastic simulation of reactions (31), in which the species populations have been plotted every 2000 reactions. We observe that the initial monomer population first plummets sharply [for comparison we show in Fig. 4(a) the pure isomerization 1 sd envelope of Fig. 2] as reaction channel R_2 rapidly builds up the unstable dimer population. At about $t \approx 0.2$, a quasiequilibrium is achieved between species S_1 and S_2. Thereafter, those two volatile species are slowly depleted through reaction channels R_1 and R_4, respectively, with all reactions ceasing when $X_1 = X_2 = 0$. The final value of X_3 gives the number of stable dimers that were created from the initial pool of unstable monomers. The run shown in Fig. 4 actually ended at $t = 43.06$ with $X_3 = 17\,027$, after a total of 526 692 reactions. Of course, these terminal values will vary from run to run: In 20 independent simulations, the 1 sd ranges of those terminal values were found to be $t = 46.4 \pm 2.0$, $X_3 = 17\,066 \pm 106$, and reaction count $= 526\,009 \pm 1908$.

Figures 5(a) and 5(b) show the results of a simulation using the plain τ-leap method, with the leap size τ being chosen according to the strategy of Eqs. (26) with $\epsilon = 0.03$. The species populations here have been plotted after every leap. The run ended after 459 leaps, with $t = 43.67$ and $X_3 = 17\,045$. Simple overlays of these trajectories on those in Figs. 4(a) and 4(b) show good agreement, although detailed statistical comparison tests were not performed. A repetition of this run using the estimated-midpoint technique produced the results shown in Figs. 6(a) and 6(b), and these results are a little disappointing: After successfully negotiating the dynamical transition that occurs around $t \approx 0.2$, the simulation seems to ''get lost'' momentarily around $t \approx 1$. But the simu-

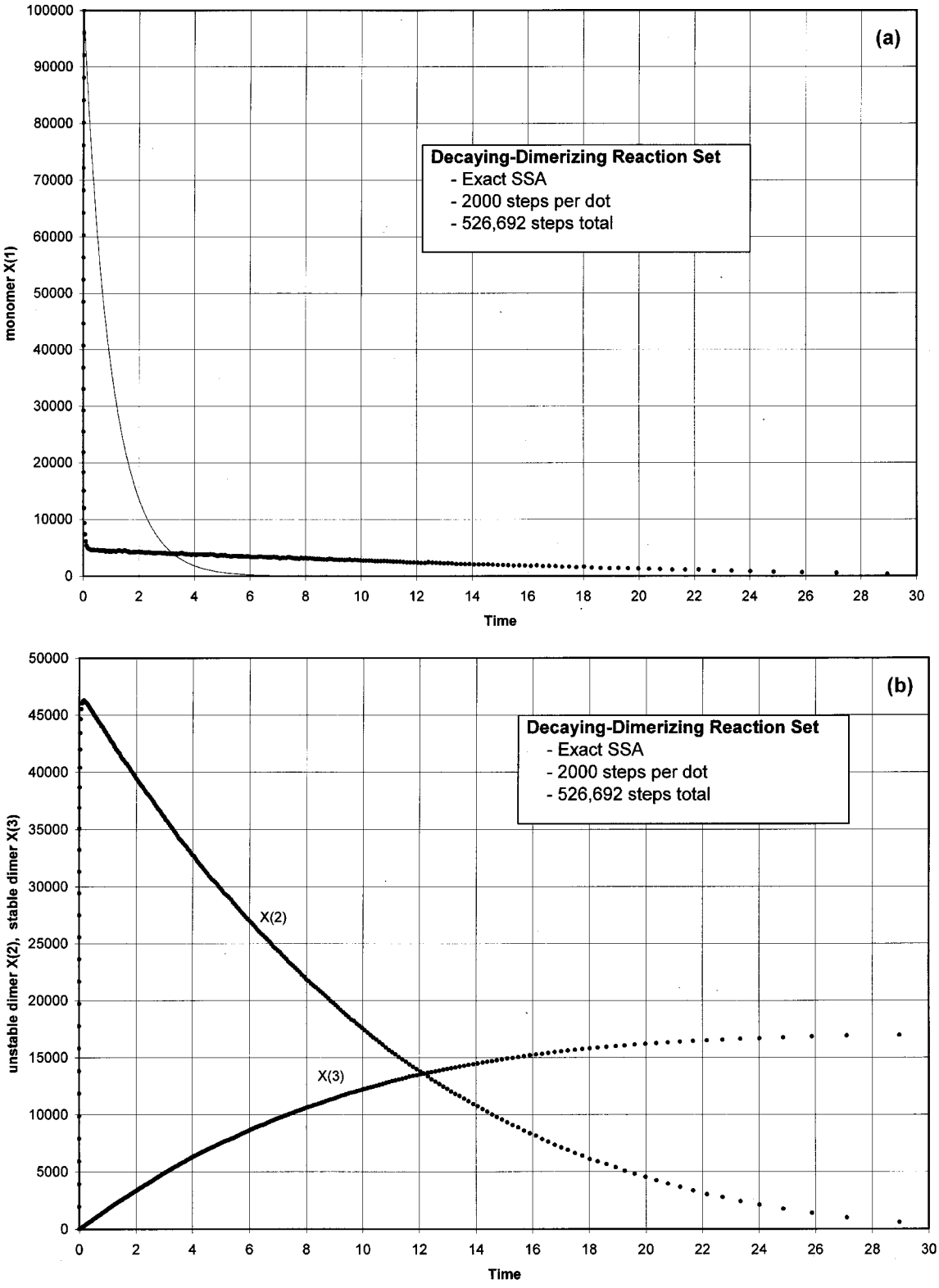

FIG. 4. Exact stochastic simulation of reaction set (31) for the rate constants (32a) and the initial condition (32b). The state is plotted after every 2000 reactions. (a) shows the evolution of the unstable monomer population X_1; the solid lines show the 1 sd envelope of Fig. 2, which would be obtained in the absence of the last three reactions in (31). (b) shows the evolutions of the unstable dimer population X_2 and the stable dimer population X_3. All reactions end when X_1 and X_2 both reach zero; that happened in this simulation at $t=43.06$, after a total of 526 692 reactions had occurred.

lation seems to recover itself quickly with no serious consequences, as overlay comparisons with Figs. 4(a) and 4(b) show good agreement elsewhere; indeed, the run ends with the quite acceptable terminal values $t=43.26$ and $X_3=17\,093$. Nevertheless, it appears that the plain τ-leap run of Fig. 5 provides a better simulation than the estimated-midpoint τ-leap run of Fig. 6, which is in sharp contrast to what we found for the simple isomerization reaction (30).

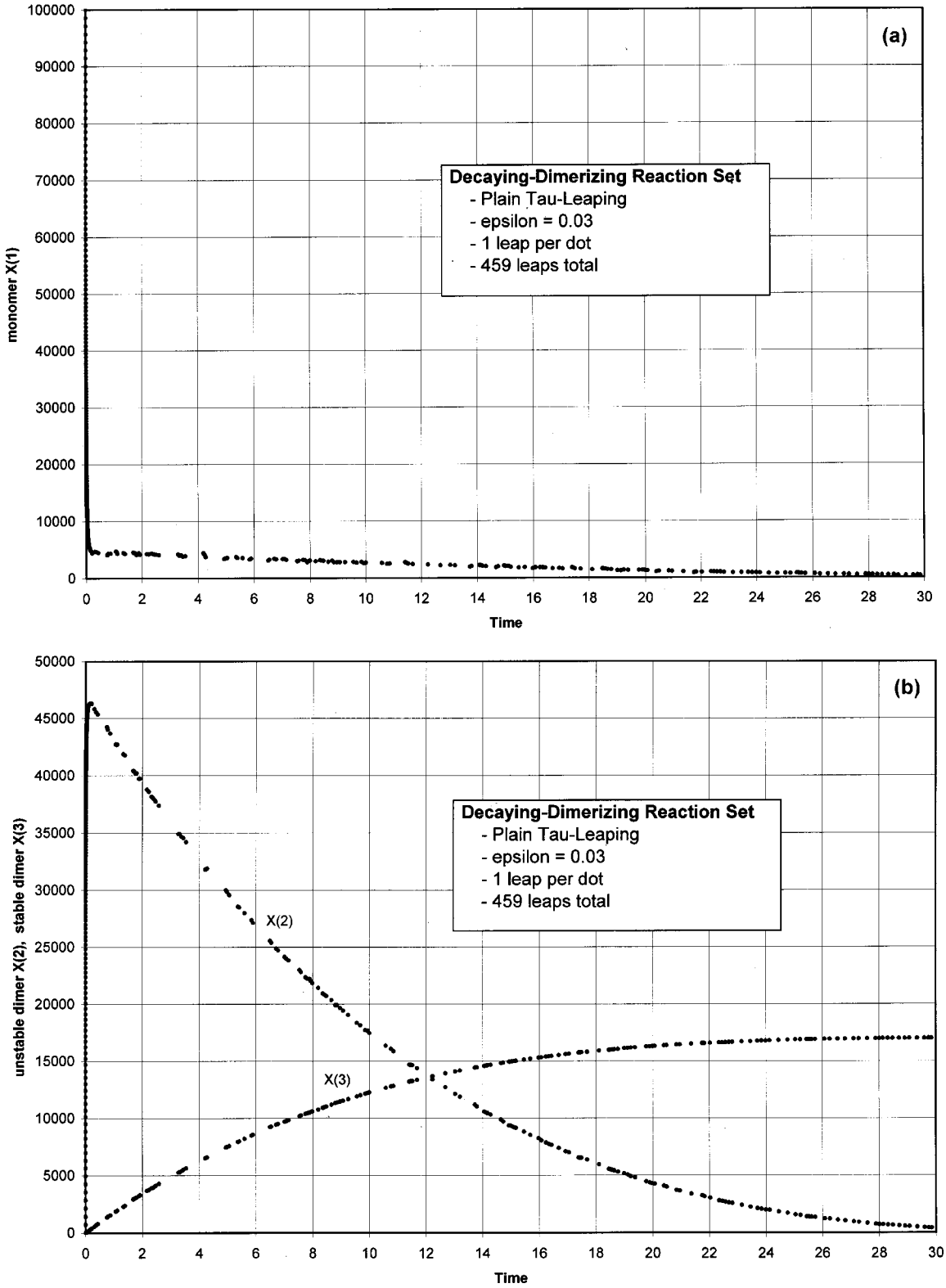

FIG. 5. A plain τ-leap simulation of the reactions in Figs. 4, using the control strategy (26) with $\epsilon = 0.03$. A dot is plotted after every leap, and only 459 leaps were required to complete the run.

Perhaps we should apply the estimated-midpoint technique to reaction (31) in a more selective way. An attempt to do that led to the simulation shown in Fig. 7: This run used the estimated-midpoint technique with $\epsilon = 0.2$ only when $a_2(\mathbf{x}) > 0.6 a_0(\mathbf{x})$, a condition that is obtained only during the early moments when the system is rapidly building up a population of unstable dimers; otherwise, ordinary τ leaping was used with $\epsilon = 0.03$. The results of this simulation are a little more encouraging. The simulation seems to track well with the exact SSA run in Fig. 4, but it uses only 289 leaps, as compared to 526 692 steps; moreover, it ends with values $t = 47.31$ and $X_3 = 17\,091$, which compare favorably with the

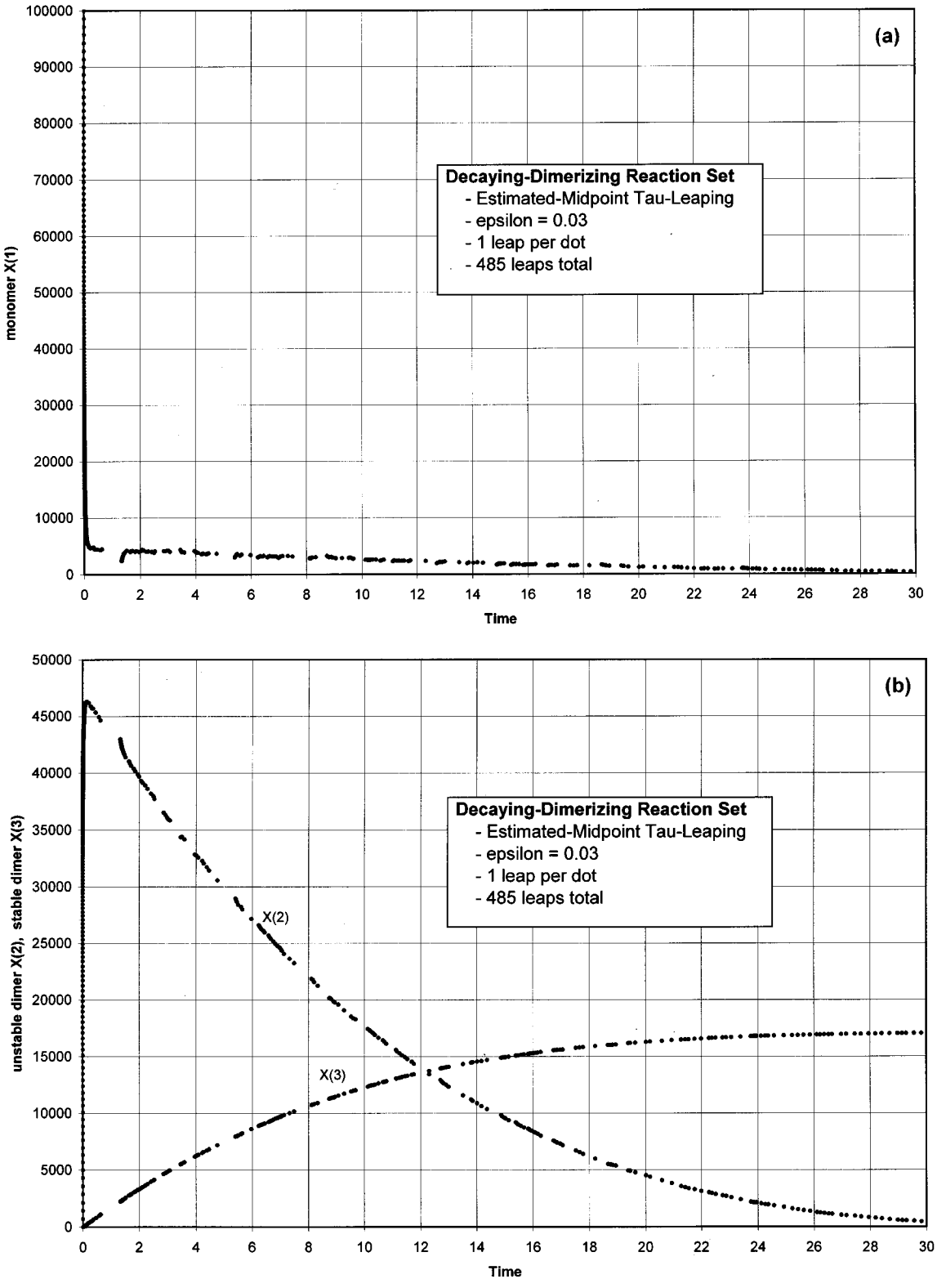

FIG. 6. A repetition of the τ-leap simulation in Figs. 5, but now using the estimated-midpoint technique. Notice the spurious momentary instability that occurs around $t=1$.

ending values found in the run in Fig. 4. But the *ad hoc* nature of this modestly improved simulation shows that more remains to be understood about τ leaping before an efficient, robust control strategy can be devised.

VIII. THE k_α-LEAP METHOD

We now describe an alternative to the τ-leap method that might, under some circumstances, be more convenient. Sup-

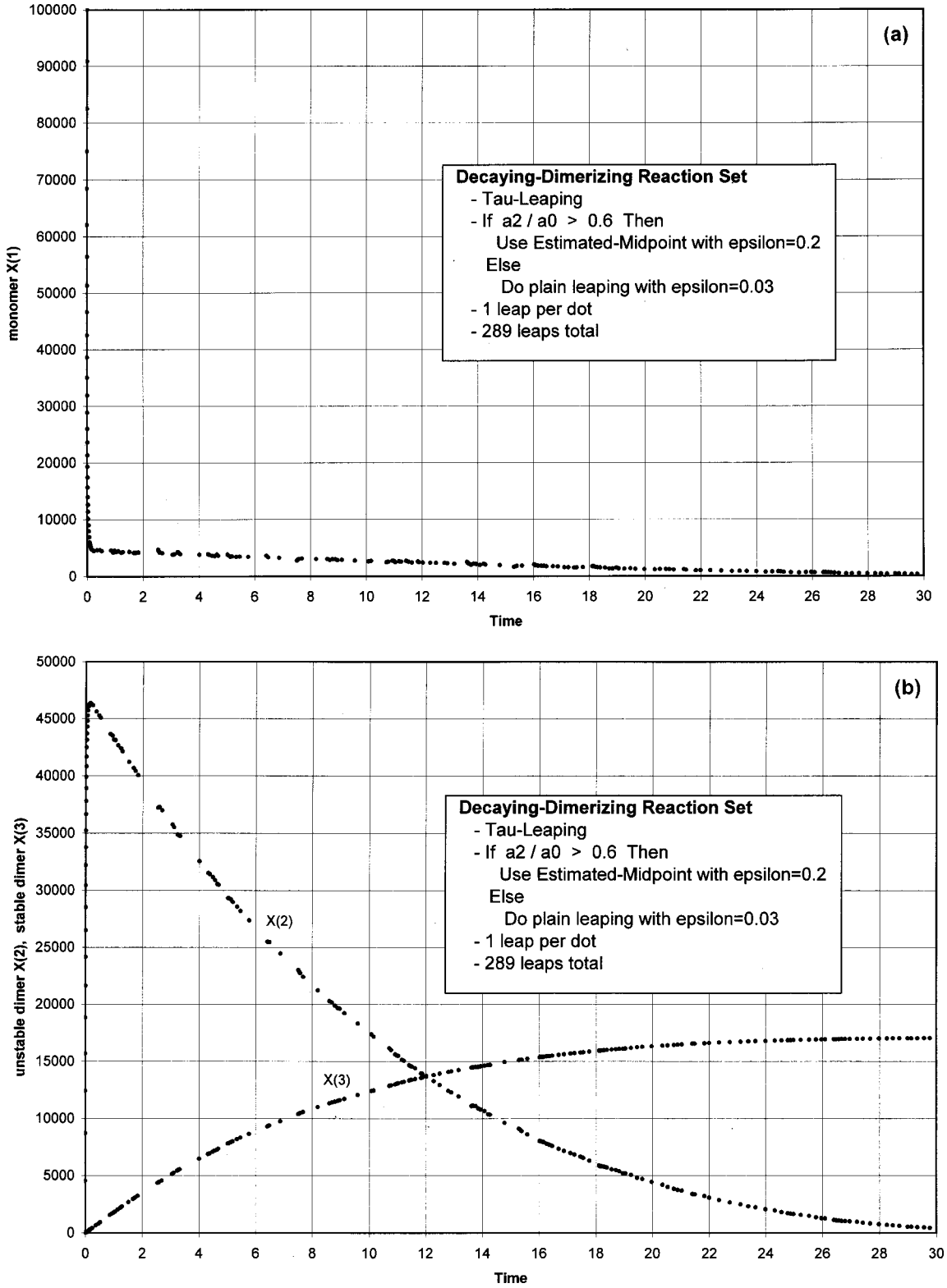

FIG. 7. A τ-leap simulation of the reactions in Figs. 4 using a hybrid strategy: When $a_2(\mathbf{x})/a_0(\mathbf{x})>0.6$, the estimated-midpoint technique is used in conjunction with the control strategy (26) with $\epsilon=0.2$; otherwise, the control strategy (26) is used with $\epsilon=0.03$ without the estimated-midpoint technique. A dot is plotted after every leap, and only 289 leaps were required to complete the run.

pose that instead of leaping down the history axis by a predetermined time τ, we leap by a predetermined number of firings k_α of a specified reaction channel R_α. With the system as before in state $\mathbf{x}$ at time t, the task then would be to determine the time $t+\tau$ at which the k_αth firing of channel R_α occurs, and also the numbers of contemporaneous firings of all the other reaction channels $R_{j\neq\alpha}$. Whereas in τ leaping we generate values for the M random variables $\{K_j(\tau;\mathbf{x},t)\}$ in Eq. (15), in k_α leaping we generate values for the M random variables $\{T(k_\alpha;\mathbf{x},t),K_{j\neq\alpha}(k_\alpha;\mathbf{x},t)\}$, where

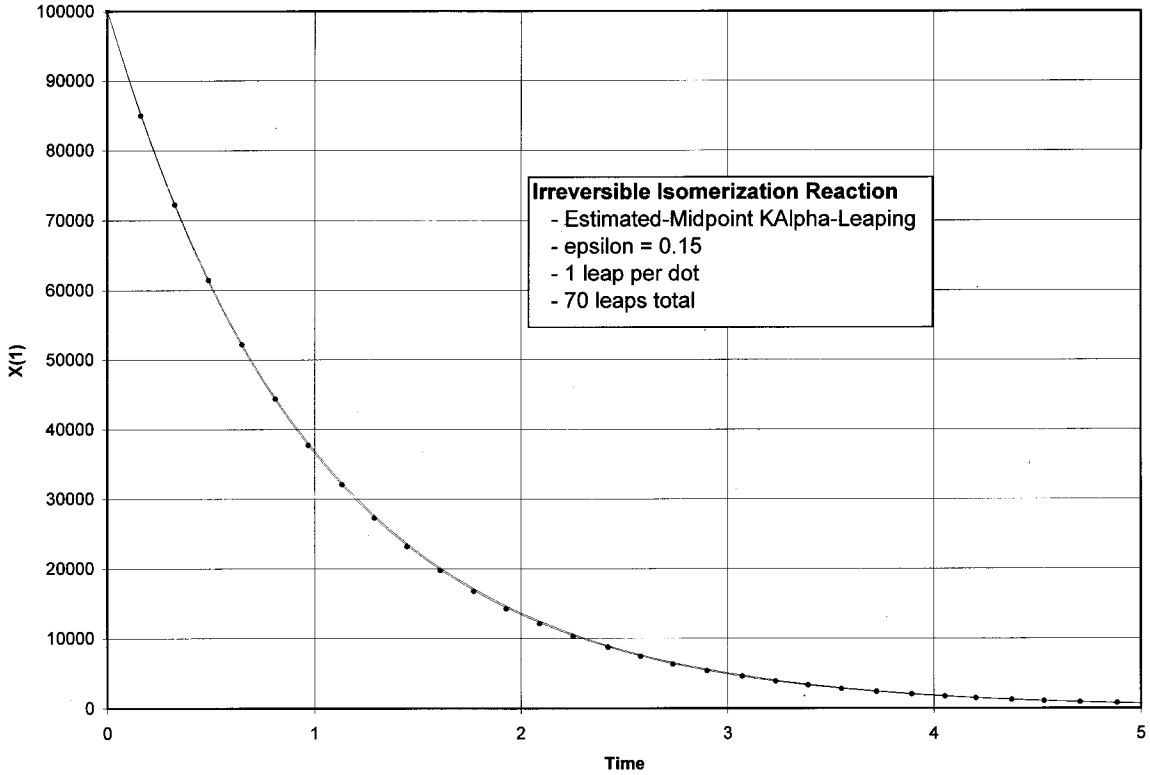

FIG. 8. A k_α-leap simulation of the reaction in Fig. 2 using the control strategy (34) in conjunction with the estimated-midpoint technique. This simulation, which uses *gamma* random numbers, is to be compared with the τ-leap simulation in Fig. 3(c) which uses *Poisson* random numbers.

$T(k_\alpha;\mathbf{x},t)$ is the time required for exactly k_α firings of channel R_α. Assuming as before that the Leap Condition is satisfied, it can be shown that $T(k_\alpha;\mathbf{x},t)$ will be the *gamma* random variable $\Gamma(a_\alpha(\mathbf{x}),k_\alpha)$ (see the Appendix). And once a value τ has been *assigned* to $T(k_\alpha;\mathbf{x},t)$, each $K_j(k_\alpha;\mathbf{x},t)$ for $j\neq\alpha$ will then be the *Poisson* random variable $\mathcal{P}(a_j(\mathbf{x}),\tau)$. We thus arrive at the following procedure.

Basic k_α-Leap Method: Choose a value for k_α that satisfies the Leap Condition; i.e., a leap by k_α of the R_α reactions will result in a state change $\boldsymbol{\lambda}$ which is such that, for every reaction channel R_j, $|a_j(\mathbf{x}+\boldsymbol{\lambda})-a_j(\mathbf{x})|$ is "effectively infinitesimal." Generate a sample value τ of the gamma random variable $\Gamma(a_\alpha(\mathbf{x}),k_\alpha)$, and then generate for each $j\neq\alpha$ a sample value k_j of the Poisson random variable $\mathcal{P}(a_j(\mathbf{x}),\tau)$. Compute $\boldsymbol{\lambda}=\Sigma_j k_j\boldsymbol{\nu}_j$, and effect the leap by replacing t by $t+\tau$ and $\mathbf{x}$ by $\mathbf{x}+\boldsymbol{\lambda}$.

Since the *average* value of τ will be (see Appendix) $\langle\Gamma(a_\alpha(\mathbf{x}),k_\alpha)\rangle=k_\alpha/a_\alpha(\mathbf{x})$, then the average or *expected* change in state in a k_α leap will be the following simple variation on Eq. (21):

$$\overline{\boldsymbol{\lambda}}\equiv\overline{\boldsymbol{\lambda}}(\mathbf{x},k_\alpha)=\frac{k_\alpha}{a_\alpha(\mathbf{x})}\boldsymbol{\xi}(\mathbf{x}). \tag{33}$$

Using this formula, we can easily adapt both the leap size selection procedure of Sec. V and the estimated-midpoint technique of Sec. VI to k_α leaping. In particular, we have in place of Eqs. (26) the following formula for the optimal k_α:

$$k_\alpha=\left[\min_{j\in[1,M]}\left\{\epsilon\alpha_\alpha(\mathbf{x})a_0(\mathbf{x})\Big/\left|\sum_{i=1}^{N}\xi_i(\mathbf{x})b_{ji}(\mathbf{x})\right|\right\}\right], \tag{34a}$$

where $[z]$ denotes the greatest integer in z. And we add the proviso

$$\text{Use exact SSA instead if}\quad k_\alpha<1\quad\text{or}\quad\frac{k_\alpha a_0(\mathbf{x})}{a_\alpha(\mathbf{x})}<2, \tag{34b}$$

where the 2 could arguably be replaced by anything between 1 and 10.

In principle, the k_α-leap method is no more nor no less accurate or efficient than the τ-leap method. For example, Fig. 8 shows a simulation of the simple isomerization reaction (30) using the k_α-leap method in conjunction with the estimated midpoint technique, and the results are quite on a par with the corresponding τ-leap simulation in Fig. 3(c). The mathematical difference between those two runs is that the τ-leap run in Fig. 3(c) was generated using only *Poisson* random numbers (selecting τ and generating k), whereas the k_α-leap run in Fig. 8 was generated using only *gamma* random numbers (selecting k and generating τ). Figures 9(a) and 9(b) show a k_α-leap simulation of reactions (31) using the same auxiliary strategies as the τ-leap simulation in Figs. 7(a) and 7(b), and again the results are quite comparable. But we shall discuss in Sec. IX a reason for believing that the τ-leap method will usually be more convenient than the k_α-leap method.

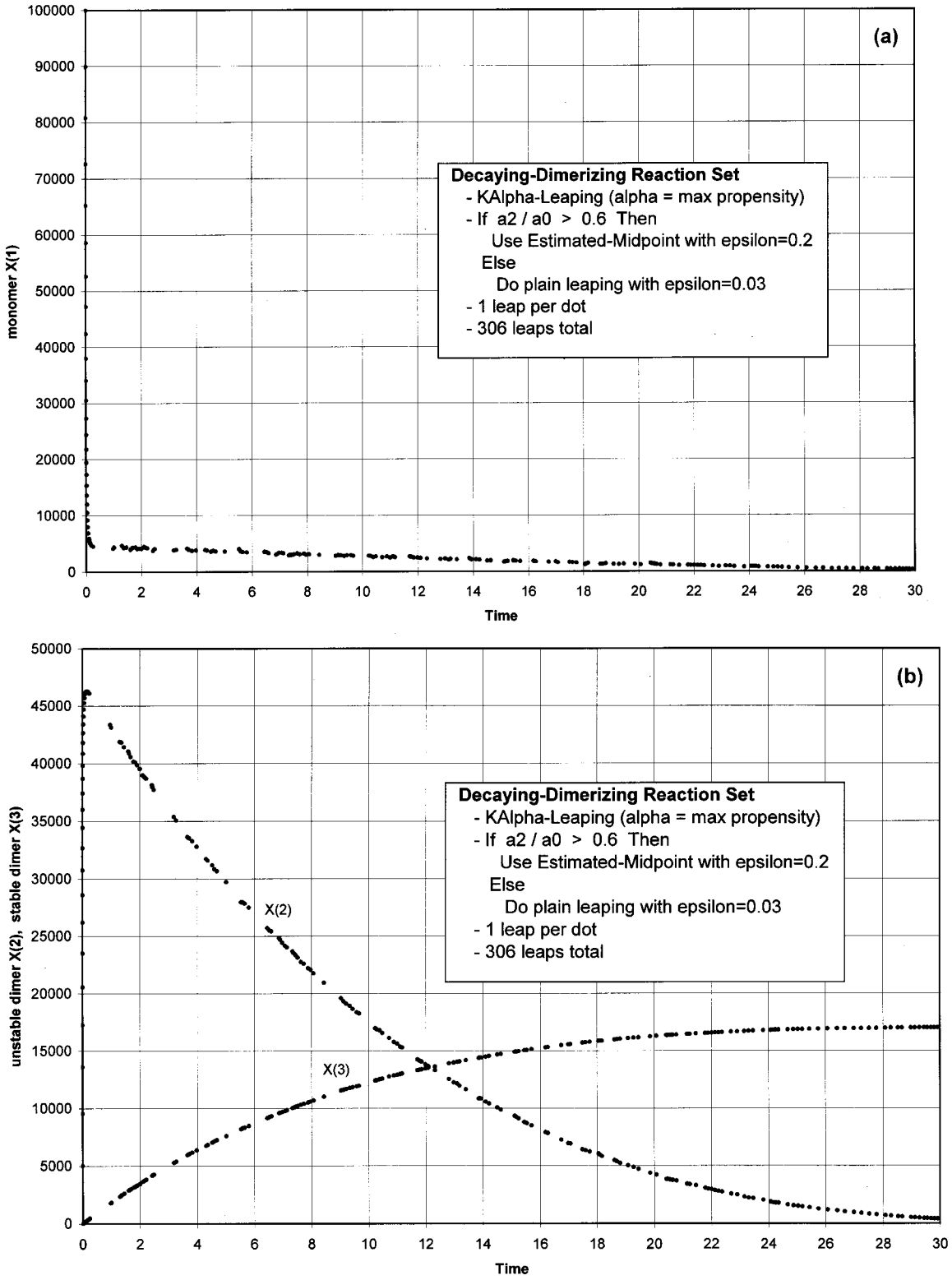

FIG. 9. A k_α-leap simulation of the reactions in Figs. 4, with the pacing channel R_α chosen at each leap to be the channel with the largest propensity function. When $a_2(\mathbf{x})/a_0(\mathbf{x}) > 0.6$, the estimated-midpoint technique is used in conjunction with the control strategy (34) with $\epsilon = 0.2$; otherwise, the control strategy (34) is used with $\epsilon = 0.03$ without the estimated-midpoint technique.

IX. CONCLUSIONS AND PROSPECTS

The simulation results reported in Sec. VII are preliminary and limited in scope, but they strongly suggest that the τ-leap method introduced in Sec. III can be made to work. The τ-leap runs in Figs. 3(c) and 7 are reasonably good approximations to the SSA runs in Figs. 2 and 4, respectively, with the ratio of number of leaps to number of steps being less than 1/1000. Of course, this cannot be interpreted as a real acceleration factor of 10^3, since it takes longer to execute a leap than to take a step. Since these runs were not

made with great attention to programming efficiency, it seems premature to try to estimate real acceleration factors. Suffice it for now to say that the leaping runs were noticeably faster than the SSA runs.

The τ-leap method fills a critical gap in our spectrum of tools for numerically simulating chemically reacting systems. At the exact end of that spectrum we have the method of molecular dynamics, which simulates every molecular collision that occurs in the system. Next we have the SSA, which simulates only those molecular collisions that are *reactive*; this is an approximating simplification which is valid only for systems that are well stirred. If the well-stirred system is large enough that we can approximately satisfy the Leap Condition, we can speed up the SSA by using the τ-leap method. If, further, each τ leap encompasses a *very* large number of firings of *every* reaction channel, the τ-leap method becomes the Langevin method. Finally, in the limit of infinitely large systems, the Langevin method typically approaches the deterministic RRE of traditional chemical kinetics.

An alternative to the τ-leap method is the k_α-leap method, described in Sec. VIII. It appears to work just as well as the τ-leap method, but the following considerations lead us to expect that the τ-leap method will usually be more convenient. It often happens in a simulation that a specified event becomes scheduled to occur at some future instant t', and that event will influence the subsequent evolution of the system. An example would be a determination that the translation of a certain portion of an mRNA chain by a ribosome will conclude at time t', resulting in the release of a reactant enzyme molecule into the system at that instant. If we were doing τ leaping, we could easily accommodate the t' event by proceeding as follows: Simulate as usual until we reach a time t when our τ-selection algorithm *suggests* a next-τ value satisfying $t+\tau>t'$. Leap instead by the *smaller* value $\tau=t'-t$, which is always permitted, and which brings us up to the instant t'. Now introduce the scheduled event, and then resume ordinary τ leaping. But it would be much less easy to accommodate the scheduled t' event when doing k_α leaping. Note that the *effect* of a k_α leap can always be *approximated* by doing a τ leap with $\tau=k_a/a_a(x)$, which is the *expected* time for k_α firings of reaction channel R_α. But of course, if ever the need should arise to step by an *exact* number of firings of a particular reaction channel, then we should do a k_α leap for that step.

The shortcomings of the τ-leap method as presented here are most evident in the general raggedness of the trajectories in Figs. 5, 6, and 7 in the time interval $1<t<20$, which shows that the procedure described in Sec. V for selecting optimal values for τ needs improvement. Also, we need to understand why the estimated-midpoint technique described in Sec. VI does not work in all situations. More generally, we need a *robust control strategy* for dynamically deciding when to step exactly and when to leap approximately, and if leaping then with what parameter values and with what auxiliary error-reduction scheme. *Until such a robust control strategy is developed, the τ-leap method cannot be considered ready for practical application.*

We may hope that such a robust control strategy will arise out of future efforts to apply τ leaping to more complicated reaction schemes, such as model chemical oscillators, bistable systems, and simple genetic regulatory reactions. Such applications should also try to determine, through test runs made in conjunction with exact SSA runs, the nature and extent of the errors that are introduced by leaping. For example, whereas we show in Figs. 2 and 3(c), respectively, *single* SSA and estimated-midpoint τ-leap runs, we really need to generate, say, 1000 runs of each, and then compare in detail the resultant SSA and τ-leap histograms of $X(t)$ at various values of t. And if it is found that the τ-leap histograms differ substantially from the SSA histograms in either peak placement or peak shape, then ways of reducing those differences should be sought.

Finally, there are several purely computer science issues that deserve attention. First, since τ leaping relies heavily on Poisson random numbers, then any improvements in the efficiency of the standard method[13] for generating Poisson random numbers will naturally lead to faster leaping simulations. Also in that connection, when we can we reliably invoke the simplifying *normal approximation* (A5) for the Poisson random variable? Second, instead of generating k values according to the estimated-midpoint formula (29b), would it be feasible and preferable to generate k values according to the binomial formula (28)? Finally, in view of the vector nature of the variables k_j, $\mathbf{\lambda}$, and $\overline{\mathbf{\lambda}}$, and also the vector-matrix nature of the τ-selection formula (26a), we may expect τ leaping to lend itself naturally to vectorized or parallel computation, especially when the numbers of chemical species and reaction channels are large.

In summary, τ leaping looks promising, but the present work is only a beginning.

ACKNOWLEDGMENTS

The author thanks Harley McAdams, Adam Arkin, Michael Gibson, and Mark Ettinger for many helpful discussions. The author also thanks Carol Gillespie for assistance with the numerical computations. This work was supported by the Office of Naval Research, Grant No. N0001499WX20544, and the NavAir/ONR ILIR program.

APPENDIX: THE POISSON AND GAMMA RANDOM VARIABLES

The *Poisson* random variable $\mathcal{P}(a,t)$ is defined to be the number of ''events'' that occur in a time t, given that adt is the probability for an event to occur in any next infinitesimal time interval dt. The parameters a and t can be any positive real numbers; however, the random variable $\mathcal{P}(a,t)$ itself is a non-negative integer.

Letting $P_{\mathcal{P}}(k;a,t)$ denote the probability that $\mathcal{P}(a,t)=k$, it is easy to show that $P_{\mathcal{P}}(0;a,t)=\exp(-at)$, and by the laws of probability, we have for any integer $k\geqslant 1$,

$$P_{\mathcal{P}}(k;a,t)=\int_{t'=0}^{t} P_{\mathcal{P}}(k-1;a,t')\times a\,dt'\times P_{\mathcal{P}}(0;a,t-t').$$

Using this recursion relation and the $k=0$ formula, one can establish by induction that

$$P_{\mathcal{P}}(k;a,t)=\frac{e^{-at}(at)^k}{k!} \qquad (k=0,1,2,...). \tag{A1}$$

It can be shown from this result that the mean and variance of $\mathcal{P}(a,t)$ are

$$\langle \mathcal{P}(a,t)\rangle = \mathrm{var}\{\mathcal{P}(a,t)\} = at. \tag{A2}$$

Equation (A2) is the basis for the well known rule-of-thumb that, for random events occurring at a rate a, or more precisely with mean time per event a^{-1}, the number of events expected in a time t is $at \pm \sqrt{at}$.

The *gamma* random variable $\Gamma(a,k)$ is defined to be the sum of k statistically independent exponential random variables with common decay constant a; so, in particular, $\Gamma(a,1)=\mathcal{E}(a)$. The parameter a can be any positive real number, and the parameter k can be any positive integer; however, the random variable $\Gamma(a,k)$ itself is a non-negative real.

To deduce the form of the probability density function $P_\Gamma(t;a,k)$ of $\Gamma(a,k)$, we observe from the foregoing definition that $\Gamma(a,k)=\Sigma_{i=1}^{k} T_i$, where the random variables $T_1,\ldots,T_k$ have joint density function $\Pi_{i=1}^{k}(a\exp(-at_i))$. Therefore, by the random variable transformation theorem,[14] the density function of the sum is

$$P_\Gamma(t;a,k) = \int_0^\infty dt_1 \cdots \int_0^\infty dt_k \prod_{i=1}^{k} (a\exp(-at_i))\,\delta\left(t-\sum_{j=1}^{k} t_j\right),$$

where δ is the Dirac delta function. Evaluation of this integral gives

$$P_\Gamma(t;a,k)=ae^{-at}\frac{(at)^{k-1}}{(k-1)!}\,(t\geqslant 0). \tag{A3}$$

It can be shown form this result that the mean and variance of $\Gamma(a,k)$ are given by

$$\langle \Gamma(a,k)\rangle = \frac{k}{a}, \quad \mathrm{var}\{\Gamma(a,k)\} = \frac{k}{a^2}. \tag{A4}$$

Equations (A4) also follow from the fact that the mean and variance of the sum of k statistically independent random variables are simply the sums of the k means and the k variances, those being in this case $\langle \mathcal{E}(a)\rangle = a^{-1}$ and $\mathrm{var}\{\mathcal{E}(a)\} = a^{-2}$.

Both the Poisson and the gamma random variables become *normal* random variables for suitable limiting values of their parameters. In the case of $\mathcal{P}(a,t)$, one can use the Stirling factorial approximation together with the small-ϵ approximation for $\ln(1+\epsilon)$ to show from the density function formula (A1) that

$$\mathcal{P}(a,t)\rightarrow \mathcal{N}(at,at) \quad \text{as} \quad at\rightarrow\infty, \tag{A5}$$

$\mathcal{N}(m,\sigma^2)$ being the normal random variable with mean m and variance σ^2. And in the case of $\Gamma(a,k)$, its definition as a sum of k statistically independent random variables with mean a^{-1} and variance a^{-2} allows us to conclude from the central limit theorem that

$$\Gamma(a,k)\rightarrow \mathcal{N}\left(\frac{k}{a},\frac{k}{a^2}\right) \quad \text{as} \quad k\rightarrow\infty. \tag{A6}$$

Computer algorithms for generating Poisson and gamma random numbers are given in Press *et al.*[13] Function poidev(x,*iseed*) of Ref. 13 generates a sample of a Poisson random variable with mean x, so a sample of $\mathcal{P}(a,t)$ may be calculated as poidev(at,*iseed*). And function gamadev(k,*iseed*) of Ref. 13 generates a sample of $\Gamma(1,k)$; so, since it follows from the random variable transformation theorem that $\Gamma(a,k)=a^{-1}\Gamma(1,k)$, then a sample of $\Gamma(a,k)$ may be calculated as a^{-1}gamdev(k,*iseed*). These are the methods for generating Poisson and gamma random numbers that were used for all simulations reported in this paper.

[1] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[2] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[3] D. A. McQuarrie, J. Appl. Probab. **4**, 413 (1967).
[4] D. T. Gillespie, Physica A **188**, 404 (1992).
[5] H. H. McAdams and A. P. Arkin, Proc. Natl. Acad. Sci. U.S.A. **94**, 814 (1997).
[6] A. P. Arkin, J. Ross, and H. H. McAdams, Genetics **149**, 1633 (1998).
[7] J. J. Lukkien, J. P. L. Segers, P. A. J. Hilbers, R. J. Gelten, and A. P. J. Jansen, Phys. Rev. E **58**, 2598 (1998).
[8] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).
[9] D. Endy and R. Brent, Nature (London) **409**, 391 (2001).
[10] D. T. Gillespie, J. Chem. Phys. **113**, 297 (2000).
[11] D. T. Gillespie, *Markov Processes: An Introduction for Physical Scientists* (Academic, San Diego, 1992).
[12] See Ref. 11, Appendix E.
[13] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press, New York, 1986).
[14] See Ref. 11, Chap. 1.